بسم الله الرحمن الرحیم

اَللّٰهُمَّ صَلِّ عَلٰی مُحَمَّدٍ وَ آلِ مُحَمَّدٍ وَ عَجِّلْ فَرَجَهُمْ

طراح سایت

رشتهٔ شبکه و نرم افزار رایانه

گروه برق و رایانه

شاخه فنی و حرفه‌ای

پایهٔ یازدهم دورهٔ دوم متوسطه

دانشچی
DANESHCHI
دانلود کتاب های این مقطع

وزارت آموزش و پرورش
سازمان پژوهش و برنامه‌ریزی آموزشی

نام کتاب: طراح سایت ـ ۲۱۱۲۹۱
پدیدآورنده: سازمان پژوهش و برنامه‌ریزی آموزشی
مدیریت برنامه‌ریزی درسی و تألیف: دفتر تألیف کتاب‌های درسی فنی و حرفه‌ای و کاردانش
شناسه افزوده برنامه‌ریزی و تألیف: فاطمه درفشی‌نبی‌کندی، حسن جعفریه، علی‌اصغر صالحی‌سلیمان‌آبادی، سالار دل‌انگیزان، بیتا رهنمای‌زربیجاری، مریم سلامتی، سحر رستمیان، آمنه آب‌زن، فاطمه نوروزی هریس، مریم شبانکاره، رامین مولاناپور و سید حمید طباطبایی‌مطلق (اعضای شورای برنامه‌ریزی)
حسین حریریان، آمنه آب‌زن، فاطمه درفشی‌نبی‌کندی، فاطمه حسینی‌صفا، مسیب حاتمی‌علم و علی‌اصغر صالحی‌سلیمان آبادی (اعضای گروه تألیف)
مدیریت آماده‌سازی هنری: ادارهٔ کلّ نظارت بر نشر و توزیع مواد آموزشی
شناسه افزوده آماده‌سازی: جواد صفری (مدیر هنری) ـ مریم دهقان‌زاده (صفحه‌آرا) ـ مریم کیوان (طراح جلد) ـ زهرا ایمانی‌نصر، نیلوفر صفادل، یوسف گلی، زینت بهشتی شیرازی و حمید ثابت کلاچاهی (امور آماده‌سازی)
نشانی سازمان: تهران: خیابان ایرانشهر شمالی ـ ساختمان شمارهٔ ۴ آموزش و پرورش (شهید موسوی)
تلفن: ۹ـ۸۸۸۳۱۱۶۱، دورنگار: ۸۸۳۰۹۲۶۶، کد پستی: ۱۵۸۴۷۴۷۳۵۹
وب‌گاه: www.chap.sch.ir و www.irtextbook.ir
ناشر: شرکت چاپ و نشر کتاب‌های درسی ایران: تهران ـ کیلومتر ۱۷ جادهٔ مخصوص کرج ـ خیابان ۶۱ (دارو پخش)
تلفن: ۵ـ۴۴۹۸۵۱۶۱، دورنگار: ۴۴۹۸۵۱۶۰، صندوق پستی: ۱۳۹ـ ۳۷۵۱۵
چاپخانه: شرکت چاپ‌ونشر کتاب‌های درسی ایران «سهامی خاص»
سال انتشار و نوبت چاپ: چاپ اول ۱۴۰۵

کلیه حقوق مادی و معنوی این کتاب متعلق به سازمان پژوهش و برنامه‌ریزی آموزشی وزارت آموزش و پرورش است و هرگونه استفاده از کتاب و اجزای آن به‌صورت چاپی و الکترونیکی و ارائه در پایگاه‌های مجازی، نمایش، اقتباس، تلخیص، تبدیل، ترجمه، عکس‌برداری، نقاشی، تهیه فیلم و تکثیر به هر شکل و نوع بدون کسب مجوز از این سازمان ممنوع است و متخلفان تحت پیگرد قانونی قرار می‌گیرند.

شابک ۳- ۳۸۳۳ - ۰۵ - ۹۶۴ - ۹۷۸ **ISBN 978 – 964 – 05 – 3833–3**



ملت شریف ما اگر در این انقلاب بخواهد پیروز شود باید دست از آستین برآرد و به‌کار بپردازد. از متن دانشگاه‌ها تا بازارها و کارخانه‌ها و مزارع و باغستان‌ها تا آنجا که خودکفا شود و روی پای خود بایستد.

امام خمینی (قُدِّسَ سِرُّهُ)

فهرست

پودمان اول :برنامه نویسی پایتون ۱

واحد یادگیری ۱ـ مدل‌سازی داده ۲

واحد یادگیری ۲ـ کار با توابع و ماژول‌ها ۴۸

پودمان دوم : طراحی صفحات وب ایستا ۸۳

واحد یادگیری ۱ـ طراحی ساختار صفحات وب با html ۸۴

واحد یادگیری ۲ـ طراحی ظاهر صفحات وب با CSS ۱۱۶

پودمان سوم : تولید صفحات وب تعاملی ۱۴۵

واحد یادگیری ۱ـ توسعه صفحات وب تعاملی با بوت استرپ ۱۴۶

واحد یادگیری ۲ـ طراحی صفحات وب تعاملی با جاوا اسکریپت ۱۹۲

پودمان چهارم : توسعه وب پایتون ۲۳۳

واحد یادگیری ۱ـ تولید برنامه شیءگرا ۲۳۴

واحد یادگیری ۲ـ توسعه برنامه شیءگرا ۲۵۴

پودمان پنجم : برنامه‌نویسی جنگو ۲۷۵

واحد یادگیری ۱ـ ایجاد وب اپلیکیشن ۲۷۶

واحد یادگیری ۲ـ توسعه وب اپلیکیشن ۲۹۵

منابع: ۳۳۰

این کتاب در سال‌های ۱۴۰۲ و ۱۴۰۳ از منظر چهار گروه از دنیای کار (کارفرمایان بزرگ، خبرگان حرفه‌ای، فروشندگان خدمات و کالا و انجمن‌های حرفه‌ای) و چهار گروه از دنیای آموزش (هنرآموزان خبره، مربیان سازمان فنی و حرفه‌ای، هنرجویان پایه دوازدهم و فارغ‌التحصیلان شاغل) مورد ارزشیابی کشوری قرار گرفت و نتایج آن در محتوا اعمال گردید.

سخنی با هنرآموزان گرامی

در راستای تحقق اهداف سند تحول بنیادین آموزش و پرورش و برنامه درسی ملی جمهوری اسلامی ایران و نیازهای متغیر دنیای کار و مشاغل، برنامه درسی رشته شبکه و نرم‌افزار رایانه طراحی و بر اساس آن محتوای آموزشی نیز تألیف گردید. کتاب حاضر از مجموعه کتاب‌های کارگاهی می‌باشد که برای سال یازدهم تدوین و تألیف گردیده است. این کتاب دارای پنج پودمان است و شرط لازم برای کسب شایستگی در هر پودمان کسب شایستگی در هر دو واحد یادگیری آن پودمان است. در صورت احراز نشدن شایستگی در هر یک از واحدهای یادگیری پس از ارزشیابی اول، فرصت جبران و ارزشیابی دوباره تا پایان سال تحصیلی وجود دارد. ارزشیابی هر واحد یادگیری نیز از دو بخش نمره مستمر و نمره شایستگی فنی تشکیل شده است و شرط لازم برای محاسبه این دو بخش، کسب شایستگی در بخش شایستگی‌های غیر فنی است. از ویژگی‌های دیگر این کتاب طراحی فعالیت‌های یادگیری ساخت‌یافته در ارتباط با شایستگی‌های فنی و غیرفنی از جمله مدیریت منابع، اخلاق حرفه‌ای و مباحث زیست‌محیطی است. این کتاب جزئی از بسته آموزشی تدارک دیده شده برای هنرجویان است که لازم است از سایر اجزای بستهٔ آموزشی مانند کتاب همراه هنرجو، نرم‌افزار و فیلم آموزشی در فرایند یادگیری استفاده شود. کتاب همراه هنرجو در هنگام یادگیری، ارزشیابی و انجام کار واقعی مورد استفاده قرار می‌گیرد. شما می‌توانید برای آشنایی بیشتر با اجزای بسته یادگیری، روش‌های تدریس کتاب، شیوه ارزشیابی مبتنی بر شایستگی، مشکلات رایج در یادگیری محتوای کتاب، بودجه‌بندی زمانی، نکات آموزشی شایستگی‌های غیرفنی، آموزش ایمنی و بهداشت و دریافت راهنما و پاسخ فعالیت‌های یادگیری و تمرین‌ها به کتاب راهنمای هنرآموز این درس مراجعه کنید. رعایت ایمنی و بهداشت، شایستگی‌های غیرفنی و مراحل کلیدی بر اساس استاندارد از ملزومات کسب شایستگی می‌باشند.

کتاب «طراح سایت» شامل پنج پودمان آموزشی است که هر پودمان دارای دو واحد یادگیری می‌باشد. هنرجویان در طول دوره، مهارت‌های نظری و عملی متنوعی در زمینه طراحی، توسعه و پیاده‌سازی وب‌سایت‌های پویا کسب می‌کنند.

پودمان اول : برنامه‌نویسی پایتون؛ این پودمان شامل دو واحد یادگیری است : مدل‌سازی داده و کار با توابع و ماژول‌ها.

در این بخش هنرجویان با مفاهیم بنیادین برنامه‌نویسی، تعریف توابع، استفاده از کتابخانه‌ها و مدیریت داده‌ها به‌صورت عملی آشنا می‌شوند.

پودمان دوم : طراحی صفحات وب ایستا؛ این پودمان شامل دو واحد یادگیری است : طراحی ساختار صفحات وب با html و طراحی ظاهر صفحات وب با CSS :در این بخش، اصول ساخت و نمایش عناصر صفحه، رنگ‌بندی، چینش محتوا و طراحی رابط‌های کاربری به‌صورت کاربردی آموزش داده می‌شود.

پودمان سوم : تولید صفحات وب تعاملی؛ شامل دو واحد یادگیری است : توسعه صفحات وب تعاملی با بوت‌استرپ و طراحی صفحات وب تعاملی با جاوا اسکریپت : در این پودمان، هنرجویان به شکل عملی مهارت‌های لازم برای افزودن تعامل، پویایی به صفحات وب را تمرین می‌کنند.

پودمان چهارم : توسعه وب پایتون؛ این پودمان شامل دو واحد یادگیری است : تولید برنامه شیءگرا و توسعه برنامه شیءگرا. در این پودمان، مبانی طراحی شیءگرا و پیاده‌سازی پروژه‌های کوچک با استفاده از ساختارهای کلاس و شیء آموزش داده می‌شود.

پودمان پنجم : برنامه‌نویسی جنگو (Django)؛این پودمان پایانی شامل دو واحد یادگیری است : ایجاد وب اپلیکیشن و توسعه وب اپلیکیشن. در این بخش، هنرجویان با چارچوب Django کار می‌کنند و به‌صورت عملی وب‌سایت کامل را طراحی و پیاده‌سازی می‌نمایند. مطالب به‌صورت گام‌به‌گام از نصب و پیکربندی محیط تا اجرای نهایی پروژه آموزش داده می‌شود. امید است که با تلاش و کوشش شما همکاران گرامی اهداف پیش‌بینی شده برای این درس محقق گردد.

دفتر برنامه‌ریزی و تألیف کتاب‌های درسی فنی و حرفه‌ای و کاردانش

نظرسنجی کتاب درسی

سخنی با هنرجویان عزیز

شرایط در حال تغییر دنیای کار در مشاغل گوناگون، توسعه فناوری‌ها و تحقق توسعه پایدار، ما را بر آن داشت تا برنامه‌های درسی و محتوای کتاب‌های درسی را در ادامه تغییرات پایه‌های قبلی براساس نیاز کشور و مطابق با رویکرد سند تحول بنیادین آموزش و پرورش و برنامه درسی ملی جمهوری اسلامی ایران در نظام جدید آموزشی بازطراحی و تألیف کنیم. مهم‌ترین تغییر در کتاب‌ها، آموزش و ارزشیابی مبتنی بر شایستگی است. شایستگی، توانایی انجام کار واقعی به‌طور استاندارد و درست تعریف شده است. توانایی شامل دانش، مهارت و نگرش می‌شود. در رشته تحصیلی ـ حرفه‌ای شما، چهار دسته شایستگی در نظر گرفته است :

شایستگی‌های فنی برای ورود و رشد در بازار کار مانند طراحی و توسعه وب‌سایت‌های پویا و کاربردی.

۱ـ شایستگی‌های غیرفنی برای موفقیت حرفه‌ای مانند مسئولیت‌پذیری، اخلاق کاری و ارتباط مؤثر.

۲ـ شایستگی‌های فناوری اطلاعات و ارتباطات مانند کار با ابزارهای توسعه، نگارش کد و طراحی رابط کاربری.

۳ـ شایستگی‌های یادگیری مادام‌العمر مانند جست‌وجو و کسب دانش از منابع جدید در حوزه فناوری.

بر این اساس دفتر تألیف کتاب‌های درسی فنی و حرفه‌ای و کاردانش مبتنی بر اسناد بالادستی و با مشارکت متخصصان برنامه‌ریزی درسی فنی و حرفه‌ای و خبرگان دنیای کار مجموعه اسناد برنامه درسی رشته‌های شاخه فنی و حرفه‌ای را تدوین نموده‌اند که مرجع اصلی و راهنمای تألیف کتاب‌های درسی هر رشته است.

این درس، اولین درس شایستگی‌های فنی و کارگاهی است که ویژه رشته شبکه و نرم‌افزار رایانه تألیف شده است. کسب شایستگی‌های این کتاب برای موفقیت آینده شغلی و حرفه‌ای شما بسیار ضروری است. هنرجویان عزیز سعی نمایید؛ تمام شایستگی‌های آموزش داده شده دراین کتاب را کسب و در فرایند ارزشیابی به اثبات رسانید.

کتاب درسی طراح سایت شامل پنج پودمان است و هر پودمان دارای دو واحد یادگیری است و هر واحد یادگیری از چند مرحله کاری تشکیل شده است. شما هنرجویان عزیز پس از یادگیری هر پودمان می‌توانید شایستگی‌های مربوط به آن را کسب نمایید. شرط لازم برای کسب شایستگی در هر پودمان، کسب شایستگی در هر دو واحد یادگیری آن پودمان است. در صورت احراز نشدن شایستگی در یک یا هر دو واحد یادگیری پس از ارزشیابی اول، فرصت جبران و ارزشیابی دوباره تا پایان سال تحصیلی وجود دارد. این درس شامل پنج پودمان است. شرط لازم برای کسب شایستگی در هر پودمان کسب شایستگی در هر دو واحد یادگیری آن پودمان است. در صورت احراز نشدن شایستگی در هر یک از واحدهای یادگیری پس ارزشیابی اول، فرصت جبران و ارزشیابی دوباره تا پایان سال تحصیلی وجود دارد. ارزشیابی هر واحد یادگیری نیز از دو بخش نمره مستمر و نمره شایستگی فنی تشکیل شده است و شرط لازم برای محاسبه این دو بخش، کسب شایستگی در بخش شایستگی‌های غیر فنی است.

همچنین علاوه بر کتاب درسی شما امکان استفاده از سایر اجزای بسته آموزشی که برای شما طراحی و تألیف شده است، وجود دارد. یکی از این اجزای بسته آموزشی کتاب همراه هنرجو می‌باشد که برای انجام فعالیت‌های موجود در کتاب درسی باید استفاده نمایید. کتاب همراه خود را می‌توانید هنگام آزمون و فرایند ارزشیابی نیز همراه داشته باشید. سایر اجزای بسته آموزشی دیگری نیز برای شما در نظر گرفته شده است که با مراجعه به وبگاه رشته خود با نشانی www.oerp.ir می‌توانید از عناوین آن مطلع شوید.

فعالیت‌های یادگیری در ارتباط با شایستگی‌های غیرفنی از جمله مدیریت منابع، اخلاق حرفه‌ای، حفاظت از محیط‌زیست و شایستگی‌های یادگیری مادام‌العمر و فناوری اطلاعات و ارتباطات همراه با شایستگی‌های فنی طراحی و در کتاب درسی و بسته آموزشی ارائه شده است. شما هنرجویان عزیز کوشش نمایید این شایستگی‌ها را در کنار شایستگی‌های فنی آموزش ببینید، تجربه کنید و آنها را در انجام فعالیت‌های یادگیری به کار گیرید.

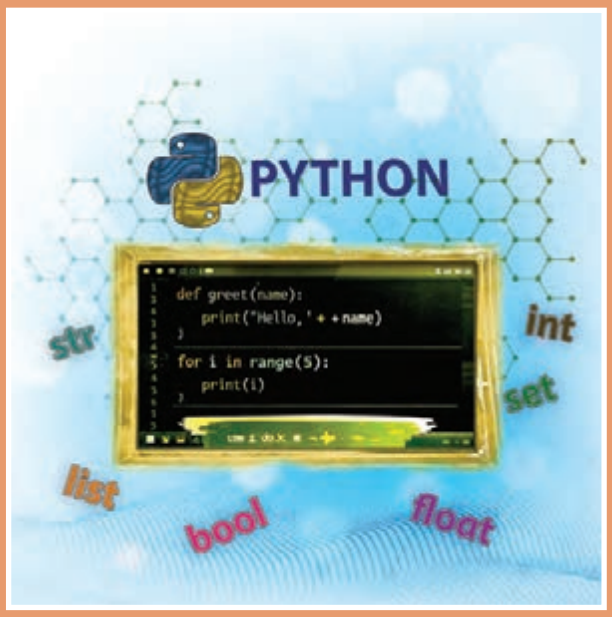
رعایت نکات ایمنی، بهداشتی و حفاظتی از اصول انجام کار است؛ لذا توصیه‌های هنرآموز محترمتان در خصوص رعایت مواردی که در کتاب آمده است، در انجام کارها جدی بگیرید.

امیدواریم با تلاش و کوشش شما هنرجویان عزیز و هدایت هنرآموزان گرامی، گام‌های مؤثری در جهت سربلندی و استقلال کشور و پیشرفت اجتماعی و اقتصادی و تربیت مؤثری شایسته جوانان برومند میهن اسلامی برداشته شود.

دفتر برنامه‌ریزی و تألیف کتاب‌های درسی فنی و حرفه‌ای و کاردانش

پودمان اول

برنامه‌نویسی پایتون



پایتون یکی از زبان‌های برنامه‌نویسی سطح بالا و پرکاربرد است که به دلیل سادگی در یادگیری و قدرت بالا در پیاده‌سازی برنامه‌ها، در حوزه‌های مختلف فناوری اطلاعات مورد استفاده قرار می‌گیرد. آشنایی با این زبان، زمینه مناسبی برای ورود هنرجویان به دنیای برنامه‌نویسی و حل مسائل رایانه‌ای فراهم می‌کند.

در این پودمان، هنرجویان با مفاهیم پایه برنامه‌نویسی به زبان پایتون آشنا می‌شوند و مهارت لازم برای **مدل‌سازی داده‌ها** و **سازمان‌دهی برنامه‌ها** را کسب می‌کنند. واحد یادگیری اول به «مدل‌سازی داده» اختصاص دارد که در آن هنرجو با انواع داده‌ها، متغیرها و ساختارهای داده‌ای آشنا شده و می‌آموزد چگونه اطلاعات را به‌صورت مناسب در برنامه‌ها ذخیره و پردازش کند. در واحد یادگیری دوم، «کار با توابع و ماژول‌ها»، هنرجویان با نحوه تعریف و استفاده از توابع آشنا می‌شوند و می‌آموزند با تقسیم برنامه به بخش‌های کوچک‌تر و استفاده از ماژول‌ها، کدی منظم، خوانا و قابل توسعه تولید کنند.

واحد یادگیری ۱
مدل‌سازی داده

آیا تابه‌حال اندیشیده‌اید

- اگر بخواهید اطلاعات خودتان مثل نام، سن و رنگ مورد علاقه‌تان را در یک برنامه ذخیره کنید، چگونه این اطلاعات را سازماندهی می‌کنید؟
- چگونه می‌توانید یک پیام خوشامدگویی به کاربر نمایش دهید و نام او را از کاربر دریافت کنید؟
- اگر بخواهید برنامه‌ای بسازید که فقط افراد دارای شرایط سنی خاص اجازه استفاده داشته باشند، چه شرطی باید بنویسید؟
- فرض کنید یک سبد خرید دارید که چند محصول در آن وجود دارد، چگونه می‌توانید همه محصولات را یکی‌یکی نمایش دهید؟
- چگونه می‌توانید عددی که کاربر به‌صورت رشته وارد کرده است را برای محاسبات ریاضی آماده کنید؟

در پایان از هنرجو انتظار می‌رود

۱ توضیح دهد که پایتون چیست و چه کاربردهایی دارد و بتواند برنامه‌های ساده را در محیط IDLE اجرا کند.

۲ قوانین نوشتن کد در پایتون (syntax) را رعایت کند، تورفتگی (indentation) را درست اعمال کند و بتواند چند دستور را در یک برنامه قرار دهد.

۳ رشته‌ها و اعداد را در پایتون چاپ کند، داده‌ها را از کاربر دریافت کرده و نوع داده‌ها را در صورت نیاز تبدیل کند.

۴ متغیر تعریف کند، انواع داده پرکاربرد را بشناسد و طبق قواعد PEP و نام‌گذاری استاندارد، نام مناسب برای متغیرها انتخاب کند.

۵ از انواع عملگرها (ریاضی، منطقی، مقایسه‌ای) استفاده کند و شرط‌های ساده با if ایجاد کند تا تصمیم‌گیری در برنامه انجام شود.

۶ لیست‌ها را تعریف و پیمایش کند، از حلقه‌های for و while برای تکرار عملیات استفاده کند، داده‌ها را در لیست‌ها و رشته‌ها پیمایش کند و از دستورات break و continue برای کنترل حلقه‌ها بهره ببرد.

استاندارد عملکرد

هنرجو بتواند داده‌ها را در پایتون تعریف و مدیریت کند، عملیات پایه روی داده‌ها انجام دهد، از ساختارهای شرطی و حلقه‌ها استفاده کند و داده‌ها را به‌صورت سازمان‌یافته پردازش کند.

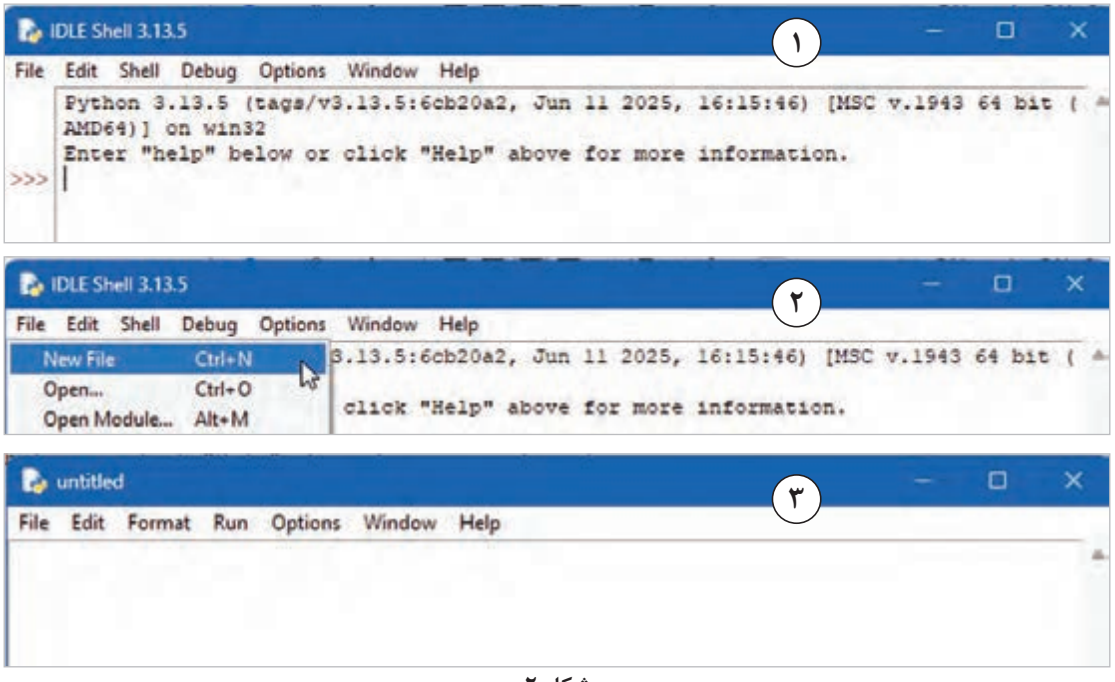
آشنایی با پایتون و محیط برنامه‌نویسی

پایتون روی پلتفرم‌های مختلف (ویندوز، مک، لینوکس و...) کار می‌کند. این زبان برنامه‌نویسی ساختار دستوری (syntax) ساده‌ای شبیه به زبان انگلیسی دارد و به توسعه‌دهندگان اجازه می‌دهد، برنامه‌هایی با تعداد خطوط کمتر نسبت به برخی زبان‌های برنامه‌نویسی دیگر بنویسند. پایتون یک زبان برنامه‌نویسی مفسری است؛ به این معنا که کدها خط‌به‌خط توسط مفسر ترجمه و اجرا می‌شوند. این زبان از شیوه‌های برنامه‌نویسی رویه‌ای، شیءگرا و تابعی پشتیبانی می‌کند؛ می‌توان برنامه‌های پایتون را در قالب فایل‌هایی با پسوند py. در هر ویرایشگر متنی ایجاد کرد و سپس آنها را برای اجرا به مفسر پایتون ارسال نمود.



شکل ۱

برای شروع کدنویسی با پایتون (بعد از نصب پایتون آخرین نسخه) با استفاده از محیط IDLE، ابتدا منوی Start را باز کنید، سپس تایپ کنید IDLE یا Python IDLE و روی IDLE (Python 3.x) کلیک کنید. پنجره محیط تعاملی (پنجره IDLE Shell) باز می‌شود. در این پنجره از منوی file، یک فایل جدید ایجاد کنید. محیط ویرایشگر متنی پایتون باز شده و می‌توانید دستورات خود را در آن وارد کنید (شکل ۲).



شکل ۲

برای شروع کار، کد نوشته در شکل ۳ را در ویرایشگر متنی پایتون وارد کنید و فایل را با کمک کلید ترکیبی ctrl + s با نام prog01 _ hello.py در یک‌پوشه ذخیره کنید. در مرحله ذخیره‌سازی دقت کنید علاوه بر نام فایل باید مسیر ذخیره‌سازی فایل نیز با دقت تعیین شوند. توصیه می‌شود از قبل یک پوشه در یکی از درایوهای D: یا E: با نام PyProg ایجاد و فایل‌های خود را در این پوشه ذخیره نمایید.
بهتر است نام فایل‌ها را با یک الگوی دلخواه ذخیره کنید. مثلاً نام فایل اول که کلمه hello را چاپ می‌کند به‌صورت زیر باشد:

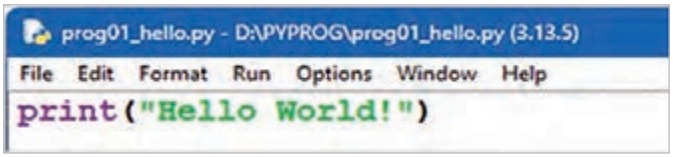
prog01_hello.py

و فایل دوم که برای کار با متغیر هاست به‌صورت زیر باشد.

prog02_variable

نکته

به هیچ عنوان در نام فایل‌ها و پوشه‌ها از حروف فارسی استفاده نشود چرا که در آینده به مشکلات زیادی بر خواهید خورد.
مثلاً این نام کاملاً اشتباه است: سلام py.

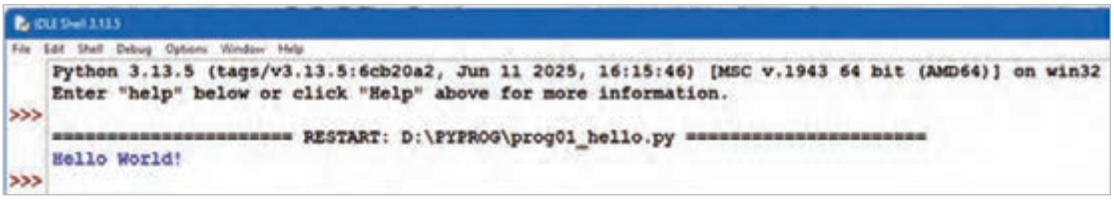


شکل ۳

نکته

پایتون از زبان فارسی پشتیبانی می‌کند؛ ولی حروف در آن به‌هم‌ریخته نشان داده می‌شوند. بهتر است از کاراکترهای انگلیسی در همه موارد استفاده کنید.

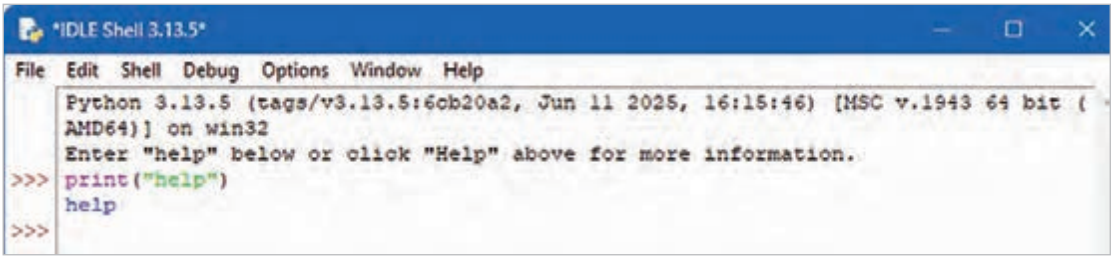
با استفاده از منوی Run دستور Run Module و یا با زدن کلید F5 اولین برنامه ایجاد شده را اجرا کنید.
در برخی از لپ‌تاپ‌ها برای اجرای برنامه لازم است کلید ترکیبی Fn+F5 را فشار دهید. (شکل ۴)



شکل ۴

همان‌طور که در شکل ۴ مشاهده می‌کنید، نتیجهٔ اجرای این کد در پنجره IDLE Shell نمایش داده می‌شود. کدهای ساده را می‌توانید در محیط IDLE Shell تایپ کرده و با زدن کلید Enter آن را اجرا نمایید (شکل۵). برای خروج از این محیط exit() و یا Ctrl+d را تایپ کنید و یا به سادگی پنجره را با زدن دکمه Close در بالای آن ببندید.

ساختار دستوری نوشتن کد (syntax) در پایتون:

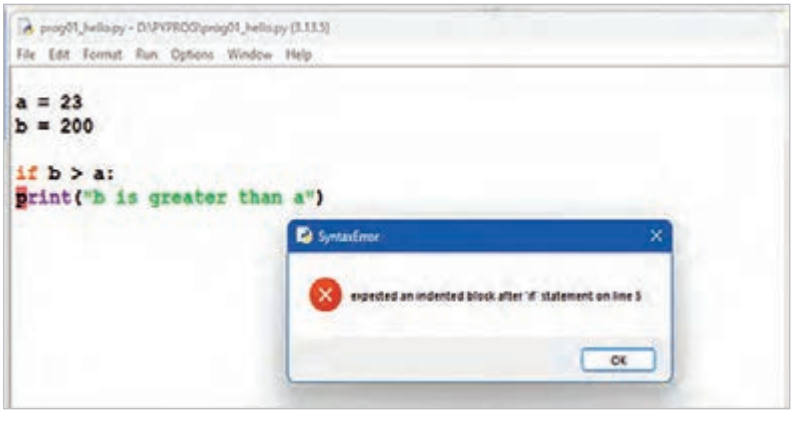


شکل ۵

پایتون با هدف خوانایی بیشتر طراحی شده است و شباهت‌هایی با زبان انگلیسی دارد. همچنین از مفاهیم ریاضی نیز تأثیر پذیرفته است. برخلاف بسیاری از زبان‌های برنامه‌نویسی که برای پایان هر دستور از نقطه‌ویرگول (;) و پرانتز استفاده می‌کنند، پایتون از خطوط جدید برای تکمیل و جداسازی دستورات بهره می‌برد.

تورفتگی(indent)در پایتون: تورفتگی به فاصله‌هایی که در ابتدای یک خط کد قرار می‌گیرند، اشاره دارد. درحالی‌که در سایر زبان‌های برنامه‌نویسی، تورفتگی در کد فقط برای خوانایی است، در پایتون تورفتگی **بسیار مهم** است. پایتون از تورفتگی برای نشان‌دادن یک بلوک کد استفاده می‌کند.

اگر از تورفتگی صرف‌نظر کنید، پایتون به شما خطا می‌دهد. تعداد فاصله‌ها به‌عنوان یک برنامه‌نویس به شما بستگی دارد، رایج‌ترین استفاده چهار فاصله است، اما حداقل باید یکی باشد. باید از تعداد مساوی فاصله در یک بلوک کد استفاده کنید، در غیر این صورت پایتون به شما خطا می‌دهد (شکل ۶). در خط۵ از کد، بلوک دستور if که بعداً خواهید خواند، باید به اندازه یک Tab یا همان ۴ فاصله به جلو برود. همان‌طور که مشاهده می‌شود پایتون اجرای برنامه را با یک خطا متوقف کرده است.



شکل ۶

در کد زیر روش صحیح فاصله گذاری را مشاهده می‌کنید (شکل۷).

```
a = 33
b = 200
if b > a:
    print("b is greater than a")
```

```
b is greater than a
```

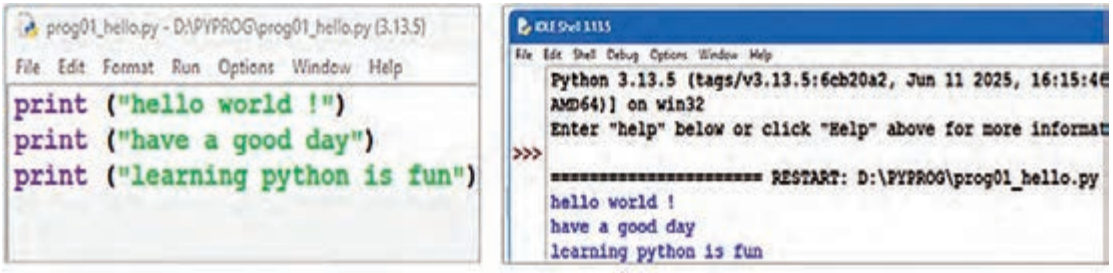
شکل ۷

دستورات (Statements) در پایتون:

هر برنامهٔ رایانه‌ای از مجموعه‌ای از دستورالعمل‌ها تشکیل شده است که باید توسط رایانه اجرا شوند. در زبان‌های برنامه‌نویسی، به این دستورالعمل‌ها، دستور یا Statement گفته می‌شود. به‌عنوان‌مثال، دستور زیر متن «learning python is fun » را روی صفحه‌نمایش چاپ می‌کند:

print(“learning python is fun! “)

وجود چندین دستور در برنامه: بیشتر برنامه‌های پایتون از چندین دستور پشت‌سرهم تشکیل می‌شوند. رایانه این دستورات را به همان ترتیب که نوشته‌اید یکی پس از دیگری اجرا می‌کند. در نتیجه، ترتیب نوشتن دستورات در برنامه اهمیت دارد (شکل ۸).



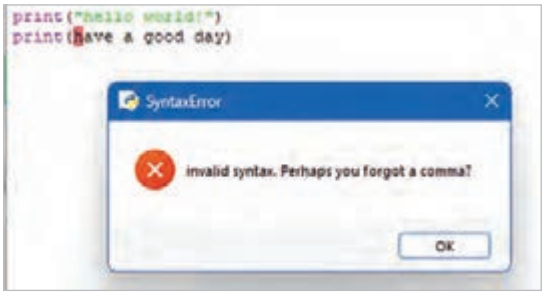
شکل ۸

چاپ رشته‌های متنی در پایتون: پیش‌تر مشاهده شد که برای نمایش متن در خروجی برنامه، از تابع print() استفاده می‌شود. در زبان پایتون، متن باید داخل گیومه قرار گیرد. برای این منظور می‌توان از کوتیشن تکی (' ') یا کوتیشن دوتایی (" ") استفاده کرد (شکل ۹).

```
print("hello world!")
print('have a good day')
```

```
print("""Welcome to our online store
Here you can buy computer products
Thank you for choosing us""")
```

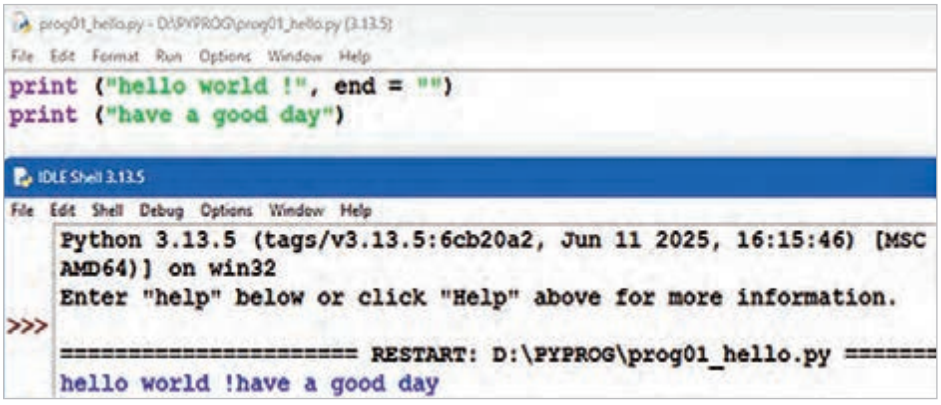
شکل ۹



شکل ۱۰

ـ در پایتون از سه کوتیشن Triple Quotes (' ' ') یا (" " ") برای چاپ متن‌های چندخطی یا متنی که شامل کوتیشن است، استفاده می‌شود.
ـ اگر فراموش کنید متن را داخل علامت نقل‌قول قرار دهید، پایتون خطایی صادر می‌کند (شکل ۱۰).

ـ اگر می‌خواهید چندین کلمه را در یک خط چاپ کنید، می‌توانید از پارامتر end استفاده کنید (شکل ۱۱).



شکل ۱۱

چاپ اعداد در پایتون:

ـ می‌توانید از تابع **print()** برای نمایش اعداد صحیح یا اعشاری استفاده کنید (شکل ۱۲).



شکل ۱۲

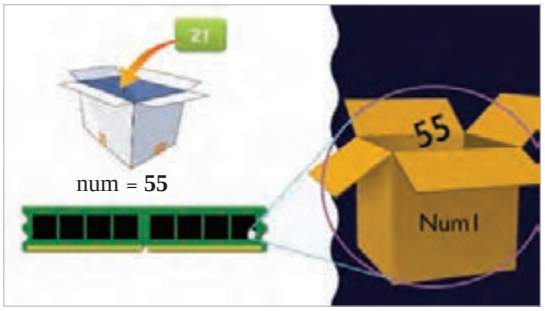
ـ می‌توانید عملیات ریاضی را درون آن انجام دهید.
ـ همچنین در دستور **print** می‌توانید اعداد و متن را با هم ترکیب کنید (شکل ۱۳).



شکل ۱۳

متغیر (variable) و انواع داده

متغیر، محلی از حافظه RAM است که برای ذخیره موقت داده استفاده می‌شود. در پایتون برای ایجاد متغیر نیازی نیست که ابتدا آن را تعریف (declare) و سپس مقداردهی (initialize) کنید؛ نوع داده به‌صورت خودکار تشخیص داده می‌شود (شکل۱۴). کافی است که در سمت چپ عملگر انتساب یعنی " = " نام متغیر و در سمت راست آن، مقدارش را بنویسید. name="baran" , num = 55

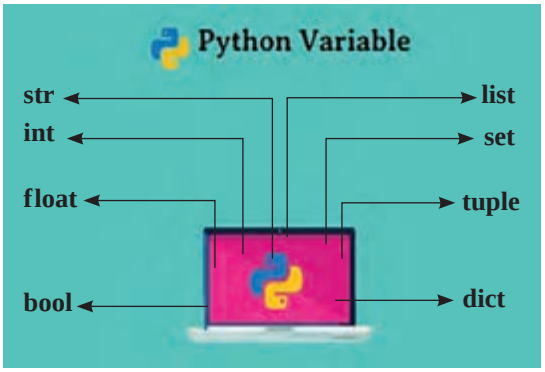


شکل ۱۴

پایتون از انواع مختلفی از متغیرها پشتیبانی می‌کند (شکل ۱۵).

پایتون از روی مقداری که به متغیر اختصاص داده می‌شود، نوع آن را تشخیص می‌دهد.

مثال: num1=15 در این مثال پایتون نوع متغیر را از نوع عدد صحیح تعیین می‌کند و نیازی به بیان نوع متغیر هم نیست.



شکل ۱۵

انواع داده پرکاربرد در پایتون

نوع داده رشته‌ای (str): هر چیزی که **متن باشد،** مثل اسم، پیام، ایمیل، رمز عبور، شماره تلفن، توضیح، نوع داده رشته‌ای (**str**) است.

نوع داده **str** که اختصار یافته کلمه **string** است را می‌توانید با استفاده از علامت نقل‌قول تکی یا دوتایی یا سه‌تایی تعریف کنید (شکل۱۶).

```
name = "sara"
city = 'Tehran'
password = '''123456'''
username = """teacher"""
```

شکل ۱۶

ـ نوع داده عدد صحیح (int): عددهایی که با آنها محاسبه انجام می‌دهید و اعشار ندارند همان عددهای صحیح (**int**) یا همان **integer**، هستند (شکل۱۷).

```
5   age = 18
6   score = 90
7   count = 5
```

شکل ۱۷

ـ نوع داده عدد اعشاری (float): این داده‌ها عددی هستند و در محاسبات ریاضی به کار گرفته می‌شوند و دارای مقدار اعشاری هستند. مثل:

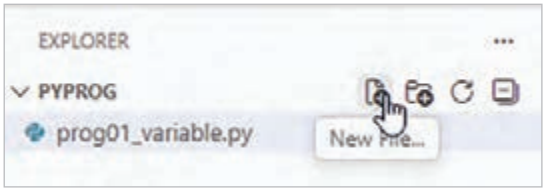
height = 1.75

ـ تابع type(): خروجی تابع **type()**, **نوع داده** هر متغیر می‌باشد (شکل۱۸).

```
name1 = "Sama"
num1 = 13.02
x = 5
print(type(x))
print(type(name1))
print(type(num1))
```

```
<class 'int'>
<class 'str'>
<class 'float'>
```

شکل ۱۸



شکل ۱۹

از این مرحله به بعد کدها را در محیط VS Code بنویسید. برای انجام این کار برنامه VS Code را اجرا کرده و سپس از منوی فایل دستور new folder یک‌پوشه جدید به نام PyProg ایجاد کنید و سپس در آن پوشه از منو فایل با انتخاب گزینه New File، کادر EXPLORER انتخاب دکمه New File و یا کلیدهای ترکیبی Ctrl+ N فایل جدیدی به نام prog01.py در این پوشه ایجاد کنید (شکل ۱۹). حالا صفحه ویرایشگر کد باز شده و می‌توانید دستورات خود را در آن بنویسید.

برای اجرای دستورات از دکمه ▷ در سمت راست‌بالای صفحه برنامه یا منوی Run استفاده کنید. نتیجه اجرای برنامه در پایین صفحه، نمایش داده خواهد شد (شکل ۲۰).



شکل ۲۰

شیوه‌نامه نگارش پایتون (PEP)

نحوه نگارش کدهای پایتون و قراردادهای کدنویسی را ارائه می‌دهد و باعث خوانایی بیشتر و تمیز (clean) شدن کدها می‌شود. برای مشاهده می‌توانید به pep8.ir مراجعه کنید.

کنجکاوی



پس از مشاهده اصول pep8 به سؤالات زیر پاسخ دهید:
۱ حداکثر در یک خط چند کاراکتر می‌توان نوشت؟
۲ تورفتگی‌ها چند کاراکتر باشد؟
۳ آیا می‌توان در ابتدای یک خط از کاراکتر فاصله استفاده کرد؟

در VSCode پس از کدنویسی با فشار کلیدهای Alt+Shift+F به طور خودکار بیشتر مشکلات فرمت کد را بر اساس pep8 اصلاح می‌کند. به‌شرط آنکه یکی از افزونه‌های autopep8، Black formatter ، Ruff یا yapf نصب شده باشد.

قواعد نام‌گذاری متغیرها در پایتون

- نام متغیر نمی‌تواند شامل کاراکتر فاصله باشد. به طور مثال نمی‌توان متغیری با نام is valid تعریف کرد.
- نام‌گذاری متغیرها در پایتون باید با یک حرف یا زیر خط آغاز شود. مثل (personl , num_)
- نام متغیر می‌تواند شامل حروف، اعداد و زیر خط باشد.
- نام متغیرها به حروف کوچک و بزرگ حساس است age، Age و AGE سه متغیر متفاوت هستند.
- نام متغیر نباید هیچ یک از کلمات کلیدی پایتون باشد. نام‌های import, var، const، if، for به‌عنوان نام متغیر اشتباه است. چون این کلمات از کلمات کلیدی پایتون هستند.

نکته



کلمه کلیدی (Keyword) در پایتون به واژه‌هایی گفته می‌شود که خودِ زبان پایتون برای دستورها و ساختارهایش رزرو کرده است. این کلمات معنای از پیش تعریف‌شده دارند و به همین دلیل نمی‌توان از آنها به‌عنوان نام متغیر، تابع یا کلاس استفاده کرد.

کنجکاوی

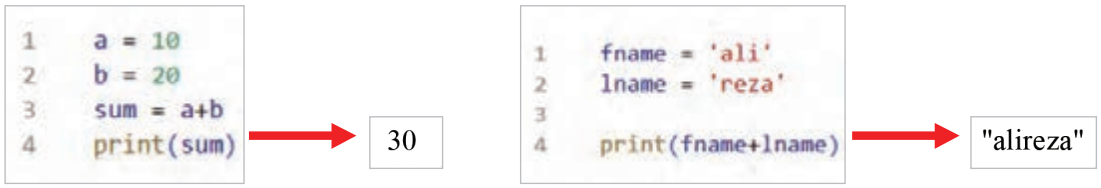


در پایتون روش‌های مختلفی برای تعیین نام متغیرهای چندکلمه‌ای وجود دارد. درباره آنها تحقیق کنید و در کلاس ارائه دهید.

- **عملگر (operator)**: نمادهای ویژه‌ای هستند که عملیات خاصی را روی مقادیر یا متغیرها انجام می‌دهند. در این عبارات a + b یا c * d یا ۴/۲ به +، * و / عملگر (operator) گفته می‌شود.

- **عملوند (operand)**: مقدار یا مقدارهایی هستند که عملگرها روی آن کار می‌کنند. مثلاً در عبارت a + b علامت "+" عملگر و a , b عملوند هستند.

عملوندهایِ عملگر " +" در پایتون می‌توانند، هر دو عددی یا هر دو رشته‌ای باشند که در این صورت در حالت اول، جمع ریاضی و در حالت دوم الحاق (**concatenation**) اتفاق می‌افتد (شکل ۲۱).



شکل ۲۱

کنجکاوی

درصورتی‌که در عملگر + یکی از عملوندها عددی و دیگری رشته‌ای باشد، چه اتفاقی می‌افتد؟

پروژه

پروژه آشپزباشی:

برنامه‌ای بنویسید که از درون لیست غذا، یکی را به‌تصادف انتخاب کند.

```
import random
foods= ["omlet", "koko", "kabab","ash"]
item= random.choice(foods)
print(item)
```

در خط اول کتابخانه **random** را به درون برنامه وارد (**import**) کنید تا بتواند در خط ۳ از متد **choice** برای انتخاب تصادفی استفاده کند.

در خط دوم شما می‌توانید لیست غذاهای محبوبتان را به لیست اضافه کنید و در منزل برای انتخاب غذا به مادر خود کمک کنید.

در خط سوم، از درون لیست **foods** یک غذا به تصادف انتخاب و درون متغیر **item** ذخیره می‌شود.

در خط چهارم، متغیر **item** که هم اکنون درون آن نام یک غذای تصادفی ذخیره است چاپ می‌شود.

ـ به نظر شما برای اینکه شانس یک غذا در انتخاب تصادفی بیشتر شود چه تغییری در کد می‌توان اعمال کرد؟

پاسخ: در کد زیر با تکرار نام آش، شانس انتخاب آش بیشتر شده است.

```
foods = ["omlet", "koko", "kabab", "ash", "ash", "ash", "ash"]
```

برخی از انواع عملگرها در پایتون:

جدول ۱ برخی از عملگرهای ریاضی را نشان می‌دهد که می‌توانید از آنها در انجام محاسبات ریاضی استفاده کنید.

جدول ۱

عملگر ریاضی	
عملگر	کاربرد
+	جمع
–	تفریق
*	ضرب
/	تقسیم
٪	باقی مانده تقسیم



شکل ۲۲

کنجکاوی

در مورد انواع دیگر عملگرها تحقیق کنید و در کلاس ارائه دهید.

دستورات ورودی و خروجی:

دستور print() : پیش‌تر گفته شد که دستور **print** برای نمایش خروجی استفاده می‌شود و با برخی از کاربردهای این دستور نیز آشنا شدید.

تابع **len()** : گاهی لازم است بدانید یک متن چند کاراکتر دارد. این تابع تعداد کاراکترهای یک متن را محاسبه می‌کند (شکل ۲۳).

```
text = "Hello World"
print("Length:", len(text))
```

```
Length: 11
```

شکل ۲۳

دستور input() : برای دریافت داده از کاربر و ذخیره در متغیر از دستور input استفاده می‌شود (شکل ۲۴).

```
my_age = input('enter your age:')
```

```
enter your age:
```

شکل ۲۴

این دستور مقدار وارد شده از طریق صفحه‌کلید را به‌صورت رشته‌ای (**string**) دریافت می‌کند و خروجی آن یک مقدار رشته‌ای است (حتی اگر "عدد" وارد کرده باشید.) (شکل ۲۵).

فعالیت

متغیرهای food، weighe و avg را به ترتیب برای غذای موردعلاقه‌تان، وزن و معدل خود تعریف کنید. از ورودی برای هریک مقداری را دریافت کرده و سپس مقادیر را نمایش دهید و نوع هر یک را تعیین نمایید.

```
age = input("Enter your age: ")
print(age + 1)
```

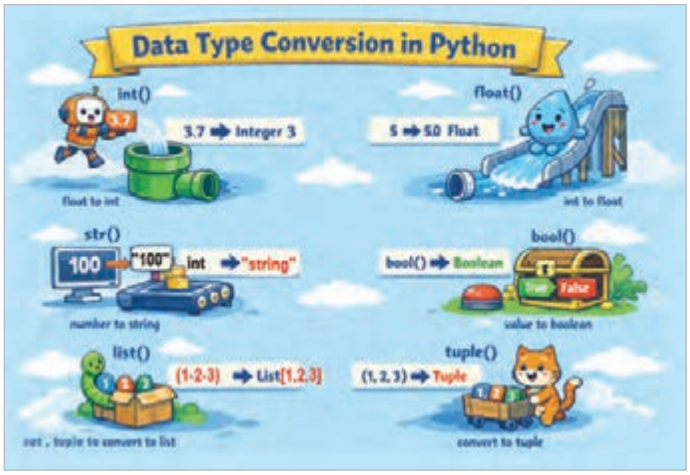
اجرای این کد با خطا مواجه می‌شود ، زیرا یک رشته را با عدد جمع می‌کنید و امکان ندارد!

```
TypeError: can only concatenate str (not "int") to str
```

شکل ۲۵

تبدیل نوع داده (Type Casting) در پایتون

در برنامه‌نویسی پایتون گاهی لازم است نوع یک داده را به نوع دیگری تبدیل کنیم. به این کار **تبدیل نوع داده** یا **Type Casting** گفته می‌شود.



شکل ۲۶

برای مثال، وقتی ورودی را با تابع input() دریافت می‌کنید، مقدار واردشده به‌صورت **رشته** (**str**) ذخیره می‌شود. اگر بخواهید با آن محاسبات ریاضی انجام دهید، لازم است آن را به عدد تبدیل کنید.
پایتون برای تبدیل نوع داده، توابع آماده‌ای در اختیار برنامه نویس قرار داده است، مانند:

- int() ← تبدیل به عدد صحیح
- float() ← تبدیل به عدد اعشاری
- str() ← تبدیل به رشته
- bool() ← تبدیل به مقدار منطقی (درست/ نادرست)

مطابق شکل۲۷ برای اینکه نوع داده را به نوع عددی تبدیل کنید می‌توانید به این صورت عمل کنید:
مثال: برنامه‌ای بنویسید که سن کاربر را دریافت و با یک پیغام مناسب نمایش دهد.

```
my_age = int(input("enter your age:"))
```

شکل ۲۷

مثال: به این مثال دقت کنید.
بعد از اجرا، خطا را مشاهده خواهید کرد. چون عدد مستقیماً به متن متصل شده است و این امکان‌پذیر نیست (شکل۲۸).

```
5   age = 20
6   print("your age: " + age)
```

```
TypeError: can only concatenate str (not "int") to str
```

شکل ۲۸

برای رفع این خطا لازم است، تبدیل نوع داده عددی به نوع داده رشته‌ای را به صورت زیر انجام دهید. پس از اجرا، کد خطایی مشاهده نخواهید کرد (شکل۲۹).

```
age = 20
print("Age: " + str(age))
```

شکل ۲۹

مثال: در خط۱ کد مقابل سال تولد کاربر دریافت می‌شود، تابع int آن را به نوع عددی صحیح تبدیل و در متغیر year ذخیره می‌کند. در خط۲ تفاضل عدد سال فعلی از سال تولد محاسبه و در متغیر age ذخیره می‌شود. در خط ۳ با دستور print سن کاربر با پیغام "your age:" نمایش داده می‌شود (شکل۳۰).

```
1   year = int(input("enter your birthday: "))
2   age = 1403 - year
3   print("your age:", age)
```

```
enter your birthday: 1381
your age: 22
```

شکل ۳۰

مثال: مقدار متغیر price عددی از نوع صحیح است. در خط ۲ برای تبدیل آن به یک عدد اعشاری از تابع float استفاده شده است (شکل ۳۱).

```
1   price = 150000
2   price_float = float(price)
3   print(price_float)
```

```
150000.0
```

شکل ۳۱

مثال: تابع input وزن را به‌صورت یک مقدار عددی دریافت می‌کند و خروجی آن یک مقدار متنی است که تابع float آن را به نوع عدد اعشاری تبدیل و در متغیر weight ذخیره می‌کند. در خط بعد دستور print دو رشته را با عملگر " + " به هم الحاق می‌کند (شکل ۳۲).

```
weight = float(input("Enter your weight: "))
print("Your weight is " + str(weight))
```

```
Enter your weight: 56
Your weight is 56.0
```

شکل ۳۲

مشاهده می‌کنید که در این خط تابع str بار دیگر متغیر weight را به نوع متنی تبدیل کرده است.

مثال: محاسبه قیمت کالا پس از اعمال ۹درصد مالیات بر ارزش افزوده (شکل ۳۳).

```
price = float(input("Enter product price: "))
tax = price * 0.09
total = price + tax
print("Total price: " + str(total))
```

```
Enter product price: 780000
Total price: 850200.0
```

شکل ۳۳

فعالیت

اندازه قد کاربر را به‌صورت یک عدد اعشاری گرفته و مشابه پیام زیر را نمایش دهید.

Your height: 1.75 meters.

فعالیت

برای راه‌اندازی یک فروشگاه هوشمند، مراحل زیر را انجام دهید.

دریافت نام مدیر فروشگاه از ورودی

دریافت سال شروع فعالیت فروشگاه

تبدیل نوع داده ورودی‌ها

نمایش پیام خوشامدگویی



شکل ۳۴

■ **لیست (list):**
لیست‌ها یکی از ۴ نوع داده در پایتون هستند که برای ذخیرهٔ مجموعه‌ای از داده‌ها در یک متغیر واحد استفاده می‌شوند، ۳ نوع دیگر Set ، Tuple و Dictionary هستند که هر کدام کاربردهای متفاوتی دارند.

■ **ایجاد لیست:** لیست‌ها با استفاده از کروشه‌ها (Square brackets)"[]" ایجاد می‌شوند.

در کد مقابل یک لیست شامل رشته‌ای از نام میوه‌ها ایجاد شده است. fruits = ["apple","banana","cherry"]
دستور مقابل یک لیست شامل اعداد 1 و 2 و 3 و 4 و 5 را ذخیره می‌کند. numbers = [1, 2, 3, 4, 5]

■ **ویژگی‌های لیست:** Mixed data لیست انواع مختلف داده را می‌تواند نگهداری کند. در کد زیر یک لیست که انواع داده متفاوت عددی، رشته‌ای، بولین و اعشاری و حتی لیست دیگر را در خود ذخیره می‌کند، ایجاد شده است (شکل۳۵).

```
mixed_list = ["Ali", 18, 17.5, True]
print(mixed_list)
```

شکل ۳۵

Indexed: عناصر لیست دارای ایندکس (Index) هستند؛ یعنی هر عنصر یک شماره دارد و اولین عنصر لیست دارای اندیس صفر است (شکل۳۶).

```
numbers = [10, 20, 30, 40]
print(numbers[0])  # 10 :خروجی
print(numbers[2])  # 30 :خروجی
```

شکل ۳۶

با ایندکس می‌توان به عناصر لیست دسترسی داشت (شکل۳۷).

```
fruits = ["apple", "banana", "cherry"]
print (fruits [0])
print(fruits[2])
```

```
apple
cherry
```

شکل ۳۷

Mutable: لیست قابلیت تغییر دارد؛ یعنی بعد از ایجاد لیست، می‌توان عناصر آن را ویرایش، حذف یا اضافه کرد. در این مثال عنصر دوم لیست که اندیس آن[۱] و مقدارش "banana" است به "oranges" تغییر داده شده است.

```
fruits [1] = "oranges"
print(fruits)
```

```
['apple ', 'oranges', ' cherry ']
```

شکل ۳۸

Ordered: لیست‌ها دارای ترتیب مشخص هستند: عناصر لیست به همان ترتیبی که وارد می‌شوند، ذخیره می‌گردند و این ترتیب تغییر نمی‌کند، مگر برنامه‌نویس آن را تغییر دهد.

عوامل مؤثر در ویژگی Ordered:

الف ـ حفظ ترتیب ورود عناصر: همان‌طور که در مثال مشاهده می‌کنید عناصر دقیقاً به همان ترتیبی که وارد شده‌اند، نمایش داده می‌شوند (شکل ۳۹).

```
fruits = ["apple", "banana", "orange"]
print(fruits)
```

```
['apple ', banana , 'oranges']
```

شکل ۳۹

ب ـ تأثیر ترتیب در نمایش عناصر لیست: اگر ترتیب تغییر کند، ایندکس‌ها هم تغییر می‌کنند (شکل ۴۰).

```
fruits = ["apple", "banana", "orange"]

print(fruits[0])  # apple
print(fruits[1])  # banana
print(fruits[2])  # orange
```

```
fruits = ["banana", "apple", "orange"]

print(fruits[0])  # banana
print(fruits[1])  # apple
print(fruits[2])  # orange
```

شکل ۴۰

ج ـ تغییر ترتیب عناصر: در این مثال فقط عنصر اول تغییر کرد، بقیه سر جای خود ماندند ← یعنی ترتیب مشخص است (شکل ۴۱).

```
fruits = ["apple", "banana", "orange"]

fruits[0] = "kiwi"
print(fruits)
```

```
['kiwi', 'banana', 'orange']
```

شکل ۴۱

د ـ مقایسه دو لیست با ترتیب متفاوت: در این مثال با اینکه عناصر یکی هستند، چون ترتیب متفاوت است، دو لیست برابر نیستند (شکل ۴۲).

```
list1 = [1, 2, 3]
list2 = [3, 2, 1]

print(list1 == list2)
```

```
False
```

شکل ۴۲

فعالیت



۱ لیستی با ۵ عنصر " book، phone، mouse، book، headset " ایجاد کنید و آن را نمایش دهید. سپس به‌جای عنصر شماره ۴ یک عنصر به نام " External HDD " را جایگزین کنید و نتیجه را نمایش دهید.

۲ لیستی با ۴ عنصر عددی دلخواه ایجاد کنید و سپس عنصر اول آن را نمایش دهید.

۳ لیستی ایجاد کنید که ۶ عنصر داشته باشد که نام ماشین‌های موردعلاقه و سال تولید آنها را ذخیره کند. سپس این لیست را نمایش دهید.

■ **افزودن عنصر جدید به لیست:** با استفاده از متد **append()** می‌توانید عنصر جدیدی را به انتهای لیست اضافه کنید (شکل ۴۳).

```
students = ["Ali", "Sara", "Reza"]
students.append("Nima")
print(students)
```

```
['Ali', 'Sara', 'Reza', 'Nima']
```

```
cities = ["Tehran", "Tabriz"]
cities.append("Shiraz")
print(cities)
```

شکل ۴۳

در پایتون برای افزودن یک عنصر بعد از یک عنصر مشخص در لیست، از متد **insert()** استفاده کنید.

list_name. insert (index, value)

نکته

برای شنیدن تلفظ صحیح این کلمه به آدرس زیر مراجعه کنید.

https://dictionary.cambridge.org/dictionary/english/insert

مثال: لیست وظایف زیر را در نظر بگیرید:

```
tasks = ["Study", "Exercise", "Sleep"]
```

وظیفه "Break" را بین "Exercise" و "Sleep" اضافه کنید. لیست را نمایش دهید (شکل ۴۴).

```
tasks = ["Study", "Exercise", "Sleep"]
tasks.insert(2, "Break")
print(tasks)
```

```
['Study', 'Exercise', 'Break', 'Sleep']
```

شکل ۴۴

فعالیت

برنامه‌های بنویسید که در لیست fruit که شامل ۲ نوع میوه است قیمت هر میوه را بعد از نام آن اضافه کرده و آن را نمایش دهد.



شکل ۴۵

حذف عنصر از لیست:

در پایتون برای حذف عناصر از یک لیست (list) روش‌های مختلفی وجود دارد. انتخاب روش مناسب به این بستگی دارد که بخواهیم بر اساس مقدار حذف کنیم یا بر اساس موقعیت (اندیس).

۱ **حذف عنصر با مقدار:** متد ()remove برای حذف مقدار یک عنصر از لیست است. اولین عنصری که همان مقدار را داشته باشد، حذف می‌شود.

list_name.remove(value)

مثال ۱: این برنامه با استفاده از متد ()remove مقدار ۱۲ را از لیست scores حذف کرده است (شکل۴۶).

```
scores = [10, 12, 15, 18, 20]
scores.remove(12)
print(scores)
```

```
[10, 15, 18, 20]
```

شکل ۴۶

مثال ۲: این برنامه با استفاده از متد ()remove مقدار اولین ۲۰ را از لیست numbers حذف کرده است (شکل۴۷).

```
1  numbers = [10, 20, 30, 40, 20, 90, 100]
2  numbers.remove(20)
3  print(numbers)
```

```
[10, 30, 40, 20, 90, 100]
```

شکل ۴۷

۲ **حذف عنصر با ایندکس (index):** متد ()pop برای حذف عنصر با ایندکس مشخص شده استفاده می‌شود.

list_name.pop(index)

مثال: در این برنامه ایندکس با شماره یک، یعنی دومین عنصر لیست حذف شده است (شکل۴۸).

نکته

دقت کنید اولین عضو لیست دارای اندیس صفر می‌باشد.

```
cities = ["Tehran", "Tabriz", "Shiraz", "Isfahan"]
cities.pop(1)
print(cities)
```

```
['Tehran', 'Shiraz', 'Isfahan']
```

شکل ۴۸

اگر داخل متد ()pop چیزی نوشته نشود، آخرین عنصر لیست حذف می‌شود. در کد زیر لیست food سه عنصر دارد (شکل۴۹).

```
food=["ash","koko","kabab"]
food.pop()
print(food)
```

```
['ash', 'koko']
```

شکل ۴۹

در خط ۲ متد ()pop چیزی نوشته نشده است؛ پس آخرین عنصر لیست یعنی "kabab" حذف می‌شود. اجرای برنامه این موضوع را تأیید می‌کند.

فعالیت

لیست کارها شامل ["Study", "Exercise", "Rest", "Sleep"] = tasks است. برنامه‌ای بنویسید که آخرین کار را از لیست حذف کند.

کنجکاوی

تحقیق کنید اگر درون متد pop عدد ۱- بیاید کدام عنصر حذف می‌شود؟ سایر اعداد منفی چطور؟ مثلاً:

food.pop(-۱)

کنجکاوی

متد ()clear و دستور del نیز برای حذف عناصر لیست کاربرد دارند. در مورد نحوه کاربرد آنها تحقیق کنید و در کلاس ارائه دهید.

ساختارهای شرطی:

ساختار شرطی به برنامه این امکان را می‌دهد که بر اساس برقرار بودن یا نبودن یک شرط، تصمیم‌گیری کند و مسیر اجرای دستورات را تغییر دهد. در این ساختار، ابتدا یک شرط بررسی می‌شود و سپس، در صورت درست بودن آن، مجموعه‌ای از دستورات اجرا می‌گردد.



شکل ۵۰

دستور if :

در زبان پایتون، دستور if ساده‌ترین و پرکاربردترین ابزار برای پیاده‌سازی ساختار شرطی است. این دستور یک عبارت منطقی را ارزیابی می‌کند که نتیجهٔ آن می‌تواند True یا False باشد. اگر نتیجهٔ شرط True باشد، بلوک کد مربوط به آن اجرا می‌شود و در غیر این صورت، این بلوک کد اجرا نخواهد شد.

نکته

دستورات مربوط به ساختار شرطی باید با یک tab (چند فاصلهٔ معادل) نسبت به خط دستور if جلوتر نوشته شوند. این تورفتگی مشخص می‌کند که کدام دستورات به‌شرط مربوط هستند.

نوع اول ساختارشرطی if :

```
if condition:
    statement
```

- **شرط (condition):** عبارتی منطقی است که نتیجهٔ آن True یا False است.
- **دستورات (statements):** دستوراتی هستند که فقط در صورت درست بودن شرط اجرا می‌شوند.
- **تورفتگی (Indentation):** این دستورات باید با یک tab نسبت به دستور if جلوتر نوشته شوند.

عملگرهای مقایسه‌ای (Comparison Operators): این عملگرها برای **مقایسه دو مقدار یا دو متغیر** استفاده می‌شوند و نتیجه آنها **همیشه یکی از دو مقدار True یا False** است (جدول ۲).

جدول ۲

عملگر مقایسه‌ای	
عملگر	کاربرد
==	مساوی است
!=	مساوی نیست
>	بزرگ‌تر از
<	کوچک‌تر از
>=	بزرگ‌تر یا مساوی
<=	کوچک‌تر یا مساوی

- **عملگر مساوی (==):** برای مقایسه دو مقدار یا دو تا متغیر استفاده می‌شود تا مشخص شود برابر هستند یا نه.

مثال: برنامه‌ای بنویسید که **نام کاربر** را دریافت کند. اگر نام واردشده برابر با مقدار admin باشد، پیام "welcome admin!" را نمایش دهد (شکل ۵۱).

```
1  username = input("enter username:")
2  if username == "admin" :
3      print("welcome admin!")
```

```
enter username:admin
welcome admin!
```

شکل ۵۱

عملگر (=!): مقایسه می‌کند آیا دو مقدار نامساوی هستند یا نه؟
در برنامه زیر در ابتدا price = 120 است و شرط درصورتی‌که مقدار آن **نامساوی** با ۱۰۰ باشد برقرار شده و پیام "price is not true" را چاپ می‌کند (شکل ۵۲).

```
price = 120

if price != 100:
    print("price is not true")
```

```
price is not true
```

شکل ۵۲

عملگرهای منطقی (Logical Operators)

در برنامه‌نویسی، برای بررسی هم‌زمان چند شرط و تصمیم‌گیری دقیق‌تر از **عملگرهای منطقی** استفاده می‌شود. این عملگرها امکان ترکیب چند عبارت شرطی را فراهم می‌کنند و نتیجهٔ آنها معمولاً یکی از دو مقدار **درست** (True) یا **نادرست** (False) است (جدول ۳).
عملگرهای منطقی معمولاً در کنار **دستورات شرطی** مانند if به کار می‌روند و به برنامه‌نویس کمک می‌کنند شرایط پیچیده‌تری را بررسی کند.

جدول ۲

عملگرهای منطقی	
عملگر	**کاربرد**
and	و
or	یا
not	نقیض

۱ـ عملگر and: این عملگر زمانی نتیجهٔ درست (True) می‌دهد که **تمام شرط‌ها درست باشند.**
مثال: در این مثال با استفاده از عملگر منطقی and هم زمان دو شرط **username** == "**admin**" و **password** == "1234" بررسی شده است و اگر کاربر هر دو را درست وارد کند پیام **login successful** نمایش داده می‌شود. برنامه را نوشته و اجرا کنید (شکل ۵۳).

```
username = input("Enter username: ")
password = input("Enter password: ")

if username == "admin" and password == "1234":
    print("Login successful")
```

```
Enter username: admin
Enter password: 1234
Login successful
```

شکل ۵۳

۲ـ عملگر or: این عملگر زمانی نتیجهٔ درست (**True**) می‌دهد که **حداقل یکی از شرط‌ها درست باشد**.
اگر هر دو شرط نادرست باشند، نتیجه نادرست خواهد بود.
مثال: در این برنامه شرط با عملگر **or** بررسی می‌شود؛ یعنی اگر کاربر نام روز **Friday** یا **Thursday** را وارد کند، پیام **Holiday** نمایش داده می‌شود (شکل۵۴).

```
day = input("Enter day name: ")
if day == "Friday" or day == "Thursday":
    print("Holiday")
```

```
Enter day name: Friday
Holiday
```

```
Enter day name: Thursday
Holiday
```

شکل ۵۴

فعالیت

با راهنمایی هنرآموز محترم خود کدهای زیر را توضیح دهید.

```
if role == "admin" or role == "teacher":
    print("Access allowed")
else:
    print("Access denied")
```

نوع دوم ساختار شرطی if – else :

```
if condition :
    statements_if
else:
    statements_else
```

ساختار if-else زمانی استفاده می‌شود که بخواهید در صورت درست بودن شرط، یک کار و در غیر این صورت کار دیگری انجام شود.

condition: عبارتی منطقی که نتیجهٔ آن True یا False است.
statements_if: دستوراتی که در صورت درست بودن شرط اجرا می‌شوند.
else: به معنی در غیر این صورت است. زمانی که شرط درست نباشد، برنامه وارد بخش **else** می‌شود و کدهای آن اجرا می‌گردد.
statements_else: دستوراتی که در صورت نادرست بودن شرط اجرا می‌شوند.

مثال: برنامه‌ای بنویسید که روزهای هفته را دریافت کرده و درصورتی‌که جمعه باشد، پیام " you are off" و در غیر این صورت "you have to go to school" نمایش داده شود.
به نظر شما اشکال کد شکل۵۵ چیست؟ اگر کاربر عبارتی غیر از روزهای هفته وارد کند چه پیغامی چاپ می‌شود؟

```
day = input("enter day name:")
if day == "friday" or day == "sunday":
    print("you are off!")
else:
    print("you have to go to school!")
```

```
enter day name:sunday
you are off!
```

```
enter day name:monday
you have to go to school!
```

شکل ۵۵

مثال: برنامه‌ای بنویسید که اعداد مابین ۰ و ۱۰ را دریافت کند، اگر عدد دریافت شده در این بازه بود، پیام "**The number is valid**" و در غیر این صورت پیام "**The number is not valid**" نمایش داده شود.
در این مثال دو شرط به طور هم‌زمان بررسی می‌شود (شکل۵۶).

```
number = int(input("enter number between 0-10:"))

if number > 0 and number < 10:
    print("The number is valid")
else:
    print("The number is not valid")
```

شکل ۵۶

فعالیت

برنامه‌ای بنویسید که **نمره درس ریاضی و زبان کاربر** را دریافت **و اگر هر دو بزرگ‌تر یا مساوی ۱۷** باشد، پیام " شما می‌توانید در رشته شبکه و نرم‌افزار ثبت‌نام کنید " را نمایش دهد و در غیر این صورت پیام" شما مجاز به انتخاب رشته نیستید " را نمایش دهد.

فعالیت

برنامه‌ای بنویسید که **سن کاربر** و **وضعیت عضویت** او را دریافت کند. اگر سن کاربر **بزرگ‌تر یا مساوی ۱۸ باشد و عضو سایت باشد**، پیام " Access allowed" را نمایش دهد: و در غیر این صورت پیام " Access denied" را نمایش دهد.

مثال: برنامه‌ای بنویسید که کاربر وضعیت اتصال اینترنت و سرعت اینترنت را وارد کند، اگر وضعیت اتصال "**yes**" و سرعت " بزرگ‌تر یا مساوی۳ " وارد شد، پیام " Internet is connected " چاپ شود و در غیر این صورت پیام "Internet is disconnect" چاپ شود (شکل۵۷).

```
internet = input("Is the internet connected? (yes/no): ")
speed = int(input("Speed: "))

if internet == "yes" and speed >= 3:
    print("Internet is connected")
else:
    print("Internet is disconnect")
```

```
Is the internet connected? (yes/no): yes
Speed: 3
Internet is connected
```

```
Is the internet connected? (yes/no): no
Speed: 4
Internet is disconnect
```

شکل ۵۷

مثال: برنامه‌ای بنویسید که اگر کاربر مهمان "guest" نبود، پیام "Welcome!" نمایش داده شود و در این صورت پیام "Access denied" نمایش داده شود.
not مقدار Boolean شرط را نقض می‌کند (شکل ۵۸).

```
name = input("Enter your name: ")

if not name == "guest":
    print("Welcome!")
else:
    print("Access denied")
```

شکل ۵۸

■ **ترکیب دو عملگر and و or**: در برنامه‌نویسی گاهی نیاز است بیش از یک شرط را با هم بررسی کرد. برای این کار می‌توان **عملگرهای منطقی and و or را با هم ترکیب کرد.**
مثال: برنامه‌ای بنویسید که نام کاربری و رمز عبور را دریافت کند و اگر نقش کاربر "teacher" **یا** "admin" بود **و** رمز واردشده برابر با "۱۲۳۴" باشد. پیام "Access allowed" و در غیر این صورت پیام "Access denied" چاپ شود (شکل ۵۹).

```
role = input("Enter role (student/teacher/admin): ")
password = input("Enter password: ")

if (role == "teacher" or role == "admin") and password == "1234":
    print("Access allowed")
else:
    print("Access denied")
```

شکل ۵۹

فعالیت

برنامه‌ای بنویسید که یک کاربر تنها در صورتی اجازه ورود داشته باشد که:
نام کاربری وارد شده باشد **و** (رمز عبور صحیح باشد یا کد تأیید فعال) باشد.

مثال: مطمئن شدن از نامعتبر بودن یک مقدار: در این مثال با عملگر مقایسه‌ای =! به معنای نابرابری، برنامه نوشته شده و اگر نام کاربری وارد شده نامساوی با "guest" باشد، از ورود او با پیغام مناسب جلوگیری می‌شود. برنامه را اجرا و نتیجه را با داده‌های مختلف بررسی کنید (شکل ۶۰).

```
banned_user = "guest"
username = input("Enter your username: ")

if username != banned_user:
    print("Welcome to the system")
else:
    print("Access denied for this user")
```

```
Enter your username: admin
Welcome to the system
```

```
Enter your username: guest
Access denied for this user
```

شکل ۶۰

مثال: در این برنامه عملگر > برای بررسی شرط بیشتر بودن نمره هنرجو در دروس پودمانی از عدد ۱۲ استفاده شده است. برنامه را اجرا و نتیجه را با داده‌های مختلف بررسی کنید (شکل ۶۱).

```
grade = int(input("Enter your grade: "))

if grade > 12:
    print("You passed!")
else:
    print("You failed!")
```

شکل ۶۱

مثال: در این برنامه عملگر > برای بررسی محدودیت سنی زیر ۱۸ سال برای ورود به سایت استفاده شده است. برنامه را اجرا و نتیجه را با داده‌های مختلف بررسی کنید (شکل ۶۲).

```
age = int(input("Enter your age: "))

if age < 18:
    print("You are underage")
else:
    print("You can enter")
```

شکل ۶۲

مثال: در این برنامه محدودیت تعداد سفارش کالا با عملگر =< مورد بررسی قرار گرفته است که تنها به تعداد ۵ یا کمتر از ۵ را شامل می‌شود. برنامه را اجرا و نتیجه را با داده‌های مختلف بررسی کنید (شکل۶۳).

```
quantity = int(input("Enter the number of items: "))

if quantity <= 5:
    print("You can purchase these items")
else:
    print("Maximum 5 items allowed")
```

شکل ۶۳

مثال: در این برنامه محدودیت پایه تحصیلی ۱۱ برای ورود به سیستم پیشرفته با عملگر = < برقرار شده است. برنامه را اجرا و نتیجه را با داده‌های مختلف بررسی کنید (شکل۶۴).

```
grade_level = int(input("Enter your grade level: "))

if grade_level >= 11:
    print("You can access advanced courses")
else:
    print("Access denied")
```

شکل ۶۴

مثال: در این برنامه شرط ورود به مرحله دوم مسابقه سن حداقل ۱۸ و امتیاز حداقل ۷۰ در نظر گرفته شده است. برنامه با عملگرهای مقایسه =< به معنی بزرگ‌تر مساوی و عملگر منطقی and به معنی " و " نوشته شده است. نتیجه عملگر and زمانی True است که هر دو عبارت `age >= 18` و `score >= 70` هر دو True باشند. برنامه را اجرا و نتیجه را با داده‌های مختلف بررسی کنید (شکل۶۵).

```
age = int(input("Enter your age: "))
score = int(input("Enter your score: "))

if age >= 18 and score >= 70:
    print("You are eligible for the competition")
else:
    print("You are not eligible")
```

شکل ۶۵

مثال: در این برنامه عملگر منطقی or بین دو عبارت قرار گرفته است و شرط را برای برقراری یکی از عبارات بررسی می‌کند. نتیجه آن زمانی true است که یکی از عبارات دو طرف or درست باشد. برنامه را اجرا و نتیجه را با داده‌های مختلف بررسی کنید (شکل۶۶).

```
age = int(input("Enter your age: "))

if age < 12 or age > 65:
    print("You can enter for free")
else:
    print("You need to pay")
```

شکل ۶۶

فعالیت

برنامه‌ای بنویسید که سن و نام کشور کاربر را دریافت کند و درصورتی‌که سن بزرگ‌تر یا مساوی ۱۸ باشد و نام کشور ایران بود، پیغام "اجازه ثبت‌نام دارید" نمایش دهد.

نوع سوم ساختار شرطی if – elif– else :

این ساختار زمانی استفاده می‌شود که **چند شرط مختلف** وجود داشته باشد و برنامه باید **یکی از آنها را انتخاب** کند. elif شرط جدید را در صورت برقرار نبودن شرط‌های قبلی بررسی می‌کند، یعنی اگر یکی از شرط‌های if یا elifها برقرار شود و دستور آن اجرا شود دیگر شرط سایر elifها بررسی نخواهد شد. elif مخفف else if است و برای بررسی شرط‌های بعدی استفاده می‌شود. تفاوت بین else و elif این است که بعد از else هیچ شرطی نوشته نمی‌شود.

ساختار کلی if – elif – else:

```
if condition1:
  statements1
elif condition2:
  statements2
else:
  statements_else
```

مثال: برنامه‌ای بنویسید که سطح بازی را از ورودی دریافت و سپس بر اساس آن پیامی مناسب نمایش دهد. برنامه را نوشته، اجرا و نتیجه را با داده‌های مختلف بررسی کنید (شکل۶۷).

```
level = input("Enter game level: ")

if level == "easy":
    print("Game is easy")
elif level == "medium":
    print("Game is normal")
elif level == "hard":
    print("Game id hard")
else:
    print("invalid level")
```

شکل ۶۷

مثال: برنامه‌ای بنویسید که پس از دریافت نوع غذا و نوشیدنی از کاربر در آن موارد زیر بررسی شود:
اگر پیتزا و نوشیدنی انتخاب شود ← قیمت ۱۴۰۰۰۰
اگر پیتزا بدون نوشیدنی← قیمت ۱۲۰۰۰۰
در غیر این صورت ← پیام«Other option» چاپ شود. در این مثال استفاده از عملگر منطقی and در شرط، مورداستفاده قرار گرفته است. برنامه را نوشته و اجرا کنید (شکل۶۸).

```
food = input("Choose a food (pizza / sandwich): ")
drink = input("Do you want a drink? (yes/no): ")

if food == "pizza" and drink == "yes":
    print("Price: 140000 Toman")
elif food == "pizza" and drink == "no":
    print("Price: 120000 Toman")
else:
    print("Other option")
```

شکل ۶۸

مثال: دستورات زیر وضعیت دستگاه را از کاربر دریافت می‌کند و بر اساس آن پیام مناسب را نمایش می‌دهد. برنامه زیر را اجرا و نتیجه را بررسی کنید (شکل۶۹).

```
status = input("Enter device status (on / off / error): ")

if status == "on":
    print("The device is working normally.")
elif status == "off":
    print("The device is turned off.")
elif status == "error":
    print("Device error! Please check the system.")
else:
    print("Invalid status.")
```

خروجی

```
Enter device status (on / off / error): on
The device is working normally.
```

شکل ۶۹

مزیت استفاده از ساختار if-elif-else نسبت به استفاده از چند if مجزا این است که اگر شرطی برقرار باشد، دیگر شرط‌های بعدی توسط برنامه بررسی نمی‌شوند. این ویژگی باعث افزایش سرعت اجرای برنامه و بهینه‌تر شدن عملکرد آن می‌شود.

فعالیت

فروشنده برای خریداران میوهٔ سیب تخفیف گذاشته است. برنامه‌ای بنویسید که نام میوه و تعداد خرید را از کاربر بگیرد: اگر میوه apple و تعداد حداقل ۵، پیام "Discount applied" نمایش داده شود.
در غیر این صورت پیام "No discount" نمایش داده شود.

فعالیت

برنامه‌ای بنویسید که کاربر سن خود را وارد کند. برنامه باید موارد زیر را بررسی کند:
اگر سن کاربر بین ۱۸ تا ۲۵ سال باشد پیام "Welcome, young adult" نمایش داده شود.
اگر سن کاربر بزرگ‌تر یا برابر ۲۶ باشد پیام "Welcome, adult" نمایش داده شود.
اگر سن کاربر کمتر از ۱۸ باشد پیام "You are underage" نمایش داده شود.

فعالیت

برنامه‌ای بنویسید که از دستور if , elif با دریافت وضعیت سفارش از فروشنده استفاده کند و بر اساس وضعیت سفارش (Registered /Sent /Delivered) در فروشگاه اینترنتی، نحوه ارسال کالا با پیام‌هایی مناسب اعلام شود:
Your order has been successfully delivered - Your order is on the way - Your order is being processed - Status unknown.

فعالیت

برنامه‌ای بنویسید که امتیاز دانش‌آموز را به‌صورت یک عدد صحیح مابین ۱ تا ۱۰۰ دریافت کند.
اگر امتیاز ≤ ۹۰← پیام "exelent"
اگر امتیاز بین ۶۹ تا ۹۰← پیام "good"
اگر امتیاز بین ۵۰ تا ۶۹← پیام "pass"
اگر امتیاز < ۵۰ ← پیام "fail" را نمایش دهد.

مثال: برنامه‌ای بنویسید که اگر مقدار choice یکی از گزینه‌های زیر بود، پیام مناسب را چاپ کند و در غیر این صورت پیام Food not available نمایش داده شود (شکل ۷۰).

الف) منوی غذا شامل دو گزینه زیر است:
Pizza -
burger -

```
choice = input("Choose your favorite food:")

if choice == "pizza":
    print("You ordered Pizza")
elif choice == "burger":
    print("You ordered Burger")
else:
    print("Food not available")
```

```
Choose your favorite food:burger
You ordered Burger
```

شکل ۷۰

ب) برنامه را گسترش دهید که اگر منوی غذا شامل سه گزینه زیر باشد: (شکل ۷۱)

- pizza
- burger
- pasta

```
choice = input("Choose your favorite food:")

if choice == "pizza":
    print("You ordered Pizza")
elif choice == "burger":
    print("You ordered Burger")
elif choice == "pasta":
    print("You ordered pasta")
else:
    print("Food not available")
```

```
Choose your favorite food:pasta
You ordered pasta
```

شکل ۷۱

در مثال بالا اگر منو ۲۰ غذا داشته باشد چه می‌شود؟ یا اگر فردا غذا جدید اضافه شود چه؟ اولین راه‌حلی که به نظر می‌رسد این‌است که از تعداد بیشتری if استفاده شود. ولی این کار، کد را طولانی‌تر و سخت‌تر می‌کند. اما راه حل آن استفاده درست از لیست‌ها و دستورات حلقه‌های تکرار است.

حلقه‌های تکرار (Loop):

در هنگام کدنویسی، اگر نیاز باشد یک دستور یا مجموعه‌ای از دستورات چندین بار اجرا شود، از **ساختارهای حلقه‌های تکرار** استفاده می‌شود. این ساختارها باعث می‌شوند کدها خواناتر، کوتاه‌تر و خواناتر و مدیریت آن آسان‌تر شود. به‌عنوان‌مثال، اگر لازم باشد برنامه‌ای نوشته شود که کلمهٔ python را ۱۰ بار چاپ کند، به‌جای آنکه دستور print('python') ده بار به صورت جداگانه نوشته شود، می‌توان این دستور را تنها یک‌بار داخل یک حلقه قرارداد تا اجرای آن به‌صورت خودکار ده بار تکرار شود.



شکل ۷۲

در زبان برنامه‌نویسی پایتون، برای پیاده‌سازی حلقه‌های تکرار دو ساختار اصلی وجود دارد:
۱ـ حلقهٔ for
۲ـ حلقهٔ while

■ **حلقهٔ for :** برای تکرار اجرای دستورات روی یک دنباله (مانند لیست، رشته، توپل، مجموعه و دیکشنری) استفاده می‌شود و معمولاً زمانی به کار می‌رود که تعداد تکرارها از قبل مشخص یا قابل‌تعیین باشد.
ساختار کلی حلقهٔ for در زبان پایتون به شکل زیر است:

```
for variable in sequence:
    statements
```

در این ساختار:
- variable متغیری است که در هر تکرار، مقدار یکی از عناصر دنباله را دریافت می‌کند.
- نام متغیر می‌تواند دلخواه باشد؛ مثلاً (i, count, num, student…)
- sequence یک دنباله مانند لیست، رشته، توپل، مجموعه یا دیکشنری است.
- statements مجموعه دستوراتی در بدنه حلقه هستند که در هر بار تکرار، اجرا می‌شوند. به این دستورات اصطلاحاً بلاک حلقه گفته می‌شود و بایستی به اندازه یک tab تورفتگی داشته باشند.

■ **پیمایش لیست‌ها**

مثال ۱: کد زیر عناصر یک لیست شامل سه‌رنگ (red, blue, green) را چاپ می‌کند (شکل۷۳).

```
colors = ["red",  "blue", "green"]
for item in colors:
    print("color: ", item)
```

```
color:  red
color:  blue
color:  green
```

شکل ۷۳

توضیح کد: در خط اول، یک لیست حاوی نام سه‌رنگ به‌صورت رشته‌ای تعریف شده است.
در خط دوم، دستور for عناصر لیست را به ترتیب پیمایش کرده و هر عنصر را در متغیر item قرار می‌دهد.
مثال ۲: نمایش سبد خرید، در این برنامه، ابتدا لیستی با نام shopping_cart ایجاد شده و کالاهای خریداری‌شده در آن ذخیره می‌شوند. سپس عناصر موجود در لیست با استفاده از حلقه for چاپ می‌شوند.
برنامه را اجرا و نتیجه را مشاهده کنید (شکل۷۴).

```
shopping_cart = ["Yogurt", "Milk", "Cheese", "Cream", "Honey", "Bread"]

print("Shopping Cart Items:")
for item in shopping_cart:
    print("Purchased : " + item)
```

```
Shopping Cart Items:
Purchased : Yogurt
Purchased : Milk
Purchased : Cheese
Purchased : Cream
Purchased : Honey
Purchased : Bread
```

شکل ۷۴

توضیح کد: در خط اول، یک لیست حاوی نام کالاهای خریداری شده به‌صورت رشته‌ای تعریف شده است.
دومین خط، یک پیغام مناسب را قبل از ورود به حلقه چاپ می‌کند.
در خط سوم، دستور for عناصر لیست را به ترتیب پیمایش کرده و هر عنصر را در متغیر item قرار می‌دهد.
در خط پایانی، یک پیغام حاوی نام تک‌تک عناصر لیست چاپ می‌شود.

پیمایش رشته‌ها

مثال: چاپ حروف کلمه python (شکل۷۵)

```
1   for ch in "Python":
2       print(ch)
```

```
P
y
t
h
o
n
```

شکل ۷۵

توضیح کد: در هر بار اجرای حلقه، یکی از حروف رشتهٔ python در متغیر ch قرار گرفته و همان مقدار در داخل حلقه چاپ می‌شود.

نمونه‌هایی از کاربرد حلقهٔ for

مثال ۱: برنامهٔ زیر برای شمارش تعداد عناصر یک لیست با استفاده از حلقهٔ for نوشته شده است (شکل۷۶).

```
fruits = ["apple", "banana", "apple", "orange", "apple"]

count = 0
for fruit in fruits:
    count += 1

print("count:", count)
```

```
count: 5
```

شکل ۷۶

توضیح کد: در خط اول، یک لیست حاوی نام چند میوه به‌صورت رشته‌ای تعریف شده است.
در خط دوم، متغیر count با مقدار اولیه صفر ایجاد می‌شود.
در خط سوم، حلقهٔ for عناصر لیست را به ترتیب پیمایش کرده و هر عنصر را در متغیر fruit قرار می‌دهد.
در خط چهارم، به‌ازای هر بار اجرای حلقه، یک واحد به متغیر count افزوده می‌شود. دو دستور زیر معادل هستند.

```
count += 1
count = count + 1
```

در خط پایانی، مقدار نهایی count چاپ می‌شود. توجه داشته باشید که این خط خارج از حلقه قرار دارد.

فعالیت

در پایتون یک تابع آماده برای انجام کد صفحه قبل وجود دارد؛ در مورد آن تحقیق کنید.

مثال ۲: اگر هدف، شمارش تعداد یک عنصر خاص (برای مثال، تعداد apple) درون حلقه باشد، می‌توانید با استفاده از دستور شرطی if برنامه را به‌صورت زیر تغییر دهید (شکل ۷۷).

```
fruits = ["apple", "banana", "apple", "orange", "apple"]

count = 0
for fruit in fruits:
    if fruit == "apple":
        count += 1

print("count:", count)
```

```
count: 3
```

شکل ۷۷

توضیح کد: در خط چهارم، دستور if بررسی می‌کند که آیا مقدار fruit با مقدار apple برابر است یا خیر؟ درصورتی‌که شرط فوق برقرار باشد، یک واحد به متغیر count افزوده می‌شود.

فعالیت

لیست‌ها دارای یک متد برای انجام کد بالا هستند؛ در مورد آن تحقیق کنید.

■ **استفاده از تابع range():** تا اینجا، از حلقهٔ for برای پیمایش عناصر موجود در لیست‌ها و رشته‌ها استفاده شده است.

در برخی موارد، لازم است یک دستور به تعداد دفعات مشخصی تکرار شود، بدون آنکه از قبل لیستی از مقادیر در اختیار باشد.

در چنین شرایطی، از تابع range استفاده می‌شود.

تابع range() یک دنباله از اعداد تولید می‌کند که از مقدار start آغاز شده و با گام‌هایی که در step مشخص می‌شود، تا قبل از مقدار stop ادامه می‌یابد.

Start: (اختیاری و پیش‌فرض)
Stop: (اجباری) است و خود آن انجام نمی‌شود.
Step: (اختیاری) ـ پیش‌فرض آن یک است.

```
for counter in range (start, stop, step):
    دستورات بدنه حلقه
```

شکل ۷۸

■ **اولین حالت:** در ساده‌ترین حالت، تابع () range تنها یک عدد دریافت می‌کند که به‌صورت range(stop) نوشته می‌شود. به مثال زیر توجه کنید:

مثال: حلقه‌ای برای چاپ اعداد ۰ تا ۴ (شکل ۷۹).

```
for i in range(5):
    print(i)
```

```
0
1
2
3
4
```

شکل ۷۹

توضیح کد: مقدار شروع (start) به طور پیش‌فرض صفر است و گام (step) نیز به طور پیش‌فرض مقدار یک را دارد. مقدار پایان (stop) عدد ۵ در نظر گرفته شده و خود این عدد در خروجی تولید نمی‌شود. تابع range(۵) دنباله‌ای از اعداد ۰ تا ۴ را تولید می‌کند. حلقهٔ for به تعداد ۵ بار اجرا می‌شود. متغیر i در هر تکرار مقدار یکی از این اعداد را دریافت می‌کند. در خط دوم، در هر بار اجرا مقدار متغیر i چاپ می‌شود.

■ **دومین حالت: استفاده از تابع به‌صورت range(start, stop) خواهد بود.**

مثال: حلقه‌ای برای چاپ اعداد از ۱ تا ۶ (شکل ۸۰).

```
for i in range(1, 7):
    print(i)
```

```
1
2
3
4
5
6
```

شکل ۸۰

توضیح کد: مقدار شروع (start) درون دستور range، یک است.

مقدار پایان حلقه (stop) عدد ۷ است؛ ولی خود عدد ۷ در نظر گرفته نخواهد شد. تابع range دنبالهٔ اعدادِ از ۱ تا ۶ را تولید می‌کند.

گام (step) نیز به طور پیش‌فرض برابر یک است.

■ **در حالت سوم،** تابع range سه عدد دریافت می‌کند که به‌صورت range(start, stop, step) نوشته می‌شود. به مثال زیر توجه کنید:

مثال: حلقه‌ای برای چاپ اعدادِ زوج کوچک‌تر از ۱۰ (شکل ۸۱).

```
for i in range(0, 10, 2):
    print(i)
```

```
0
2
4
6
8
```

شکل ۸۱

توضیح کد: مقدار شروع (start) در دستور range برابر با صفر است.
مقدار پایان (stop) عدد ۱۰ در نظر گرفته شده است، اما خود این عدد در خروجی تولید نمی‌شود. مقدار گام (step) برابر با ۲ است.
تابع range دنباله‌ای از اعدادِ زوج از ۰ تا ۸ را تولید می‌کند.

فعالیت

برنامهٔ قبل را به‌گونه‌ای تغییر دهید که دنبالهٔ اعداد زوج را از بزرگ به کوچک چاپ کند. برای این منظور، لازم است مقدار step منفی شود و مقادیر شروع (start) و پایان (stop) نیز متناسب با آن تغییر داده شوند.

فعالیت

برنامه‌ای بنویسید که مدت‌زمان انتظار برای دریافت غذا را به‌صورت یک عدد (بر حسب دقیقه) از کاربر دریافت کرده و شمارش معکوس را نمایش دهد.
در هر مرحله، پیام زیر چاپ شود:

Time remaining: X minutes

که در آن، به‌جای X مقدار زمان باقی‌مانده نمایش داده می‌شود.
پس از رسیدن مقدار شمارش به صفر، پیام زیر نمایش داده شود.

Your food is ready!

فعالیت

برنامه‌ای بنویسید که قد ۵ نفر را از ورودی دریافت کند و میانگین قد آنها را محاسبه و نمایش دهد.

استفاده از کاراکتر زیرخط (_) در نام متغیرها مجاز است؛ بنابراین می‌توان از آن به‌تنهایی نیز به‌عنوان نام یک متغیر استفاده کرد.
کد شکل ۸۲ را مشاهده کنید.

```
1   for i in range( 10 ):
2       print(i)
```

شکل ۸۲

در این مثال، از متغیر شمارندهٔ حلقه در بدنهٔ حلقه استفاده نشده است؛ بنابراین می‌توان به‌جای آن از کاراکتر زیرخط (_) استفاده کرد. استفاده از _ نشان می‌دهد که مقدار متغیر موردنظر در ادامهٔ برنامه استفاده نخواهد شد.

کنجکاوی

بررسی کنید که استفاده از _ به‌جای استفاده از نام متغیر چه مزایایی دارد؟

معرفی دستورات کنترلی (break و continue) در حلقه‌ها

در زبان پایتون، دستورات کنترلی break و continue برای کنترل جریان اجرای حلقه‌ها استفاده می‌شوند.

continue: برای رد شدن از یک بخش خاص از حلقه و رفتن به تکرار بعدی استفاده می‌شود.

break: اجرای حلقه را به طور کامل متوقف کرده و از آن **خارج** می‌شود.

مثال: برنامه زیر را نوشته و اجرا کنید. در این برنامه، نام تمام غذاهای موجود در یک لیست، هر کدام در یک خط چاپ می‌شوند. (شکل۸۳)

```
foods = ["koko", "kabab", "soup", "ash"]

for item in foods:
    print("Food:", item)
```

```
Food: koko
Food: kabab
Food: soup
Food: ash
```

شکل ۸۳

حال اگر بخواهید هنگام چاپ، نام یک غذای خاص (برای مثال، soup) نمایش داده نشود، می‌توان از دستور continue درون حلقه استفاده کرد. کد زیر را با دقت نوشته و اجرا کنید و هنگام نوشتن برنامه به تورفتگی‌ها (Indentation) توجه داشته باشید (شکل۸۴).

```
foods = ["koko", "kabab", "soup", "ash"]

for item in foods:
    if item=="soup":
        continue
    print("Food:", item)
```

```
Food: koko
Food: kabab
Food: ash
```

شکل ۸۴

توضیح کد: در خط "if item == "soup"، بررسی می‌شود که آیا مقدار متغیر item با مقدار soup برابر است یا خیر.

در صورت برقرار بودن شرط if، دستور continue اجرا شده می‌شود و باعث می‌شود حلقه بلافاصله به سراغ عنصر بعدی برود که "ash" است. به همین دلیل، در خروجی برنامه نام soup چاپ نمی‌شود.

به‌طورکلی، با استفاده از دستور continue، حلقه اجرای بخشی از کد درون حلقه را برای تکرارهای خاصی نادیده می‌گیرد و مستقیماً به ابتدای تکرار بعدی وارد می‌شود.

حال اگر بخواهید هنگام چاپ، با رسیدن به نام یک غذای خاص (برای مثال، soup) پیغام مشخصی چاپ شده و اجرای حلقه **خاتمه** یابد، می‌توان از دستور break درون حلقه استفاده کرد. کد زیر را بنویسید و اجرا کنید. هنگام نوشتن برنامه به تورفتگی‌ها (Indentation) توجه داشته باشید (شکل۸۵).

```
foods = ["koko", "kabab", "soup", "ash"]

for item in foods:
    if item=="soup":
        print ("Soup doesn't count as food!")
        break
    print("Food:", item)
```

```
Food: koko
Food: kabab
Soup doesn't count as food!
```

شکل ۸۵

توضیح کد: در خط سوم، بررسی می‌شود که آیا مقدار متغیر item با مقدار soup برابر است یا خیر.
در صورت برقرار بودن شرط if، ابتدا پیام "سوپ که غذا نیست!" چاپ می‌شود و سپس دستور break اجرا شده و کنترل اجرای برنامه به بعد از حلقه منتقل می‌شود. در این حالت، اجرای حلقه به طور کامل متوقف شده و تکرارهای بعدی انجام نخواهند شد (شکل۸۶).
حال دو برنامه زیر را نوشته و اجرا کنید.

```
1  for i in range(10):
2      if i==5:
3          break
4      print(i)
```

```
0
1
2
3
4
```

```
1  for i in range(10):
2      if i==5:
3          continue
4      print(i)
```

```
0
1
2
3
4
6
7
8
9
```

شکل ۸۶

کنجکاوی

با توجه به خروجی برنامه‌های بالا کاربرد کلمات break و continue را برای هر یک از برنامه‌ها بیان کنید.

حلقه while

این حلقه مجموعه‌ای از دستورات را تا زمانی که شرط حلقه True باشد، اجرا می‌کند و هر زمان که شرط حلقه False شود، دیگر حلقه اجرا نمی‌شود و اجرای حلقه پایان می‌یابد. از این نوع حلقه معمولاً زمانی استفاده می‌شود که **تعداد دفعات تکرار حلقه از قبل مشخص و معین نباشد.**



شکل ۸۷

while condition : یا while شرط حلقه :
statements → دستورات بدنه حلقه

مثال: شخصی در ابتدای یک تالار غذاخوری به میهمانان یک مراسم عروسی خوشامد می‌گوید. به او گفته شده که آنجا بایستد و بعد از پرسیدن نام آنها به میهمان وارد شده خوشامد بگوید و برای پایان کار او محدودیتی در نظر گرفته نشده است. برنامه زیر به کمک حلقه while این کار را انجام می‌دهد (شکل۸۸).

```
while True:
    name = input("Name: ")
    print("Welcome", name)
```

```
Name: dana
Welcome dana
Name: sara
Welcome sara
Name:
```

شکل ۸۸

توضیح کدهای برنامه: در جلوی دستور while به‌جای‌گذاشتن یک شرط خاص فعلاً از عبارت منطقی True استفاده شده است. با این کار حلقه بی‌وقفه تکرار می‌شود و پایانی برای توقف آن تعریف نشده است؛ چرا که شرط جلوی while همیشه True است و هیچ‌گاه False نمی‌شود.
در دو خط بعد نام میهمان پرسیده می‌شود و پیام خوشامدگویی چاپ خواهد شد.
باتوجه به اینکه پایانی برای اجرای برنامه وجود ندارد، برای توقف برنامه یا باید از کلیدی ترکیب Ctrl + C استفاده نمود و یا باید پنجره برنامهٔ در حال اجرا را بست.

مثال: آیا می‌توان برنامه فوق را طوری نوشت که بعد از ورود داماد، وظیفه خوشامدگویی پایان یابد؟

توضیح کد: همان‌طور که در خروجی کد شکل۸۹ مشاهده می‌شود، بعد از ورود نامِ داماد، برنامه پایان می یابد.

```
while True:
    name = input("Name: ")
    if name == "damad":
        break
    print("Welcome", name)
```

```
Name: hamid
Welcome hamid
Name: damad
```

شکل ۸۹

درون حلقه پس از دریافت نام میهمان به کمک شرط if بررسی می‌شود که آیا نام وارد شده برابر damad است یا خیر و اگر برابر باشد به کمک دستور break، روند اجرای حلقه خاتمه می‌یابد و اجرای برنامه به بعد از حلقه while منتقل می‌شود.

مثال: در این تالار به‌جز میهمانان، کارمندان (Employee) نیز در حال رفت و آمدند، حال اگر نخواهید به کارمندان خوشامد گفته شود می‌توان از یک دستور شرطی و دستور continue استفاده نمود.

توضیح کد: همان‌طور که در خروجی کد شکل۹۰ مشاهده می‌شود، بعد از ورود نام Employee، برنامه دوباره یک نام دیگر از کاربر دریافت می‌کند و به کارمند خوشامد گفته نخواهد شد. خروجی برنامه را به‌دقت ببینید.

```
while True:
    name = input("Name: ")
    if name == "damad":
        break
    if name == "Employee":
        continue
    print("Welcome", name)
```

```
Name: maryam
Welcome maryam
Name: Employee
Name: reza
Welcome reza
Name: damad
```

شکل ۹۰

برنامه با رسیدن به دستور continue اجرای مابقی دستورات حلقه را رها کرده و اجرای برنامه را به ابتدای حلقه منتقل می‌کند.

این برنامه گرچه به‌خوبی کار می‌کند؛ اما این ساختار، به خاطر وجود True در جلوی while برای برنامه‌نویسی خیلی مناسب نیست و می‌توان ساختار حلقه را به‌صورت مثال بعد تغییر داد.

مثال: به مسئول خوشامدگویی گفته شده که این کار را تا زمانی ادامه دهد که داماد نیامده است؛ کد به‌صورت زیر تغییر می‌کند.

دقت کنید که در این مثال شرط ادامه حلقه در جلوی دستور while آمده است و نیازی نیست که داخل حلقه چیزی را کنترل کنید.

توضیح کد: همان‌طور که در خروجی کد شکل ۹۱ مشاهده می‌شود، بعد از ورود نام داماد و خوشامدگویی به او برنامه پایان می‌یابد.

```
name = ""
while name != "damad":
    name = input("Name: ")
    print("Welcome", name)
```

```
Name: ali
Welcome ali
Name: damad
Welcome damad
```

شکل ۹۱

در اولین خط یک متغیر به نام name ساخته شده است و مقدار آن خالی است تا شرط حلقه در ابتدا برقرار باشد.

مفهوم خط دوم، یعنی تا زمانی که نام وارد شده یعنی مقدار متغیر (name) مخالف کلمه damad است، دستورات بدنه حلقه ادامه یابد.

وقتی کاربر کلمه damad را وارد کند بعد از چاپ پیام، برنامه به ابتدای حلقه بر می‌گردد؛ ولی این بار شرط حلقه False است و اجرای حلقه پایان می‌پذیرد. در اینجا نسبت به برنامه قبلی، کد بهتر و منطقی‌تر شده است.

مثال: به مسئول خوشامدگویی گفته شده است که آنجا بایستد و بعد از پرسیدن نام آنها به میهمان وارد شده خوش آمد بگوید و به‌محض ورود داماد وظیفه خوشامدگویی پایان می‌پذیرد؛ اما دیگر نیازی نیست به داماد خوش آمد گفته شود. کد بالا کمی تغییر خواهد کرد.

توضیح کد: در شکل ۹۲ در اولین خط به‌جای آنکه یک رشته خالی را درون متغیر name قرار دهیم، با کمک دستور input یک نام واقعی از کاربر گرفته شده است. بدین ترتیب اگر در اولین خط برنامه نام damad وارد شود، شرط حلقه False شده و دستورات بدنه حلقه دیگر اجرا نخواهند شد و برنامه پایان می‌یابد.

```
name = input("Name: ")
while name != "damad":
    print("Welcome", name)
    name = input("Name: ")
```

```
Name: sara
Welcome sara
Name: damad
```

شکل ۹۲

درون حلقه ابتدا خوشامدگویی چاپ شده است و در خط بعدی نامی دیگر دریافت خواهد شد تا در دور بعدی این نام درون شرط حلقه بررسی شود.

نکته

مثال‌های بالا همگی برای یادگیری نحوه استفاده از حلقه while در شرایط مختلف به برنامه‌نویس کمک می‌کند. با نوشتن مجدد کدها و فکرکردن به روند هر یک از برنامه‌های فوق، برنامه‌نویس مطالب آن را دقیق‌تر یاد می‌گیرد. برای یادگیری بهتر به مثال‌های صفحه بعد توجه کنید.

مثال: برنامه‌ای بنویسید که از کاربر یک عدد صحیح دریافت و مربع آن عدد را چاپ کند این کار تا زمانی ادامه یابد که کاربر عددی بزرگ‌تر از صفر وارد کند (شکل۹۳).

```
number = int(input("Enter number:"))
while number > 0:
    print("Squar of", number, "is:", number * number)
    number = int(input("Enter number: "))
```

```
Squar of 5 is: 25
Enter number: 4
Squar of 4 is: 16
Enter number: 0
```

شکل ۹۳

توضیح کد: در اولین خط یک عدد صحیح از کاربر دریافت شده و درون متغیر number قرار می‌گیرد.
مفهوم شرط while: یعنی تا زمانی که مقدار متغیر number بزرگ‌تر از صفر است دستورات حلقه، تکرار خواهد شد.
داخل حلقه ابتدا با یک پیغام مناسب، عدد گرفته شده و مربع (یعنی توان دو) آن چاپ خواهد شد و در ادامه یک عدد دیگر از کاربر گرفته می‌شود.
اگر در اولین خط برنامه مقداری کوچک‌تر یا مساوی صفر وارد شود، شرط حلقه False شده و دستورات بدنه حلقه دیگر اجرا نخواهند شد و برنامه پایان می‌یابد. دقت شود که در این مثال، مربع عددِ صفر و یا عددِ منفی چاپ نمی‌شود.

فعالیت

برنامه بالا را طوری تغییر دهید که علاوه بر مربع عدد، مکعب (Cube) عدد (یعنی توان ۳ عدد) نیز چاپ شود.

مثال: برنامه‌ای بنویسید که نام تعدادی غذا را از کاربر گرفته و به یک لیست خالی اضافه (append) کند. این کار تا زمانی ادامه می‌یابد که کاربر کلمه exit را وارد کند. پس از اتمام حلقه محتوای لیست به‌صورت یک‌جا چاپ خواهد شد (شکل۹۴).

```
food_list = []
food = input("Enter food name:")

while food != "exit":
    food_list.append(food)
    food = input("Enter food name:")

print("food list:", food_list)
```

```
Enter food name:koko
Enter food name:ash
Enter food name:kabab
Enter food name:exit
food list: ['koko', 'ash', 'kabab']
```

شکل ۹۴

توضیح کد: در اولین خط، یک لیست خالی ایجاد شده است.
همانند مثال‌های قبلی ابتدا نام یک غذا را بگیرید درون متغیر food ذخیره کنید.
مفهوم شرط while: یعنی تا زمانی که مقدار متغیر food مخالف رشته "exit" است دستورات حلقه، تکرار خواهد شد.
داخل حلقه ابتدا نام غذای گرفته می‌شود؛ به کمک متد append به انتهای لیست اضافه می‌شود و در ادامه نام یک غذای دیگر از کاربر گرفته می‌شود.
در آخرین خط و بیرون از بدنه حلقه، محتویات کامل لیست غذا چاپ می‌شود.
دقت کنید که آخرین خط نباید یک tab به جلو رود و بایستی دقیقاً زیر ستون while نوشته شود.
مثال: به کد زیر دقت کنید. این کد اعداد یک تا پنج را به کمک حلقه for زیر هم چاپ می‌کند (شکل۹۵).

```
for i in range(1, 6):
    print(i)
```

```
1
2
3
4
5
```

شکل ۹۵

حال اگر بخواهید همین کار را به کمک حلقه while انجام دهید، کد آن می‌تواند به‌صورت کد شکل۹۶ باشد.
توضیح کد: در اولین خط، متغیر i با مقدار پیش‌فرض یک، مقداردهی شده است.
همانند مثال‌های قبلی ابتدا نام یک غذا را گرفته درون متغیر food ذخیره کنید.
مفهوم شرط while: یعنی تا زمانی که مقدار متغیر i کوچک‌تر از ۶ است دستورات حلقه، تکرار خواهد شد. داخل حلقه ابتدا مقدار متغیر i چاپ شده است. در آخرین خط و درون حلقه به متغیر، یک واحد افزوده شده است.

```
i = 1
while i < 6:
    print(i)
    i = i +1
```

```
1
2
3
4
5
```

شکل ۹۶

همان‌طور که مشاهده می‌شود ساختار حلقه while برای حلقه‌های با تعداد تکرار معین پیچیده‌تر و نیازمند دقت بیشتر است. ازاین‌رو به برنامه‌نویسان توصیه می‌شود قبل از شروع به کدنویسی به این موضوع فکر کنند که طبق صورت‌مسئله موجود، بهتر است از کدام نوع حلقه استفاده کنند.

نکات مهمی که از حلقه while فرا می‌گیرید:

در این نوع حلقه‌ها از قبل نمی‌دانید چند بار تکرار لازم است؛ یعنی حلقه تا زمانی که شرط برقرار است، ادامه دارد.
تکرار وابسته به یک شرط است که شرط می‌تواند مقایسه، ورودی کاربر یا وضعیت برنامه باشد.
پایان کار را کاربر یا یک اتفاق مشخص، تعیین می‌کند. پس حلقه تا رسیدن به شرط پایان ادامه می‌یابد.
برای کنترل حلقه و پیشگیری از اجرای بی‌نهایت، لازم است قبل از شروع حلقه، متغیر حلقه، مقداردهی اولیه شود.

درون حلقه می‌توان از break برای خاتمه اجرای حلقه استفاده کرد.
درون حلقه می‌توان از continue برای بازگشت به ابتدای حلقه و اجرای دوباره آن استفاده کرد.
دستورات break و continue باید به همراه یک شرط if درون حلقه‌ها استفاده شوند.
برای پیاده‌سازی حلقه‌های با تعداد معین می‌توان از حلقه‌های while نیز استفاده نمود؛ اما این کار توصیه نمی‌شود؛ چرا که نوشتن این‌گونه برنامه‌ها با حلقه for ساده‌تر و منطقی‌تر است.

فعالیت

در مثال قبلی و در انتهای برنامهٔ بالا به کمک حلقهٔ for تک‌تک غذاهای لیست را زیر هم چاپ کنید.

فعالیت

برنامه‌ای بنویسید که کاربر بتواند به‌صورت مکرر سفارش غذا دهد، تا زمانی که "done" را وارد کند.
هر بار که غذای جدیدی از کاربر گرفته شد پیام "Your order for <food> has been added." نمایش داده شود و غذای جدید به لیست سفارش غذا اضافه (append) شود. وقتی کاربر "done" را وارد نمود، برنامه تمام سفارش‌ها را نمایش دهد.

ارزشیابی شایستگی مدل‌سازی داده

استاندارد عملکرد: هنرجو بتواند داده‌ها را در پایتون تعریف و مدیریت کند ، عملیات پایه را روی داده‌ها انجام دهد،ازساختارهای شرطی و حلقه‌ها استفاده کند و داده‌ها را به صورت سازمان یافته پردازش کند.

ردیف	مرا حل کار	شرایط انجام کار (ابزار،مواد، تجهیزات، زمان، مکان و ...)	ابزارهای ارزشیابی	استاندارد (شاخص‌ها/داوری/نمره دهی)	نتایج ممکن	نمره کسب شده
۱	مقدمات و شروع کار با پایتون	رایانه، vscode نصب‌شده، مرورگر، ۱۵ دقیقه، کارگاه	مشاهده عملی	ـ نصب پایتون و یک محیط توسعه (IDE) مناسب روی رایانه شخصی یا استفاده از محیط‌های آنلاین. ـ اجرای برنامه‌های ساده در محیط IDLE ـ نوشتن اولین دستور ()print. ـ درک تفاوت بین خط فرمان و اجرای فایل. ـ دریافت یک قطعه کد پایتون با خطاهای سینتکسی یا تورفتگی و اصلاح آن. ـ انجام کار با روش‌های دیگر ـ انجام کنجکاوی‌ها	۳	
				۱ـ اجرای برنامه‌های ساده در محیط IDLE. ۲ـ نوشتن اولین دستور ()print. ۳ـ درک تفاوت بین خط فرمان و اجرای فایل. ۴ـ دریافت یک قطعه کد پایتون با خطاهای سینتکسی یا تورفتگی و اصلاح آن.	۲	
				عدم انجام و یا انجام ناقص مرحله کاری	۱	
۲	متغیرها، انواع داده و تبدیل نوع	رایانه، vscode نصب‌شده، فایل تمرینی، ۱۵دقیقه	مشاهده عملی	نوشتن کدی که در آن چند متغیر با انواع داده مختلف (رشته، عدد صحیح، عدد اعشار) تعریف شده و مقادیر آنها در ادامه استفاده شود. ـ نوشتن کدی که یک نوع داده را به نوع دیگری تبدیل کند و از آن در محاسبات استفاده کند. ـ جواب دادن به سوالات تستی یا تشریحی در مورد نام‌گذاری صحیح متغیرها، تفاوت انواع داده، و زمان استفاده از تبدیل نوع۴. انجام کار با روش‌های دیگر ـ انجام کنجکاوی‌ها	۳	
				نوشتن کدی که در آن چند متغیر با انواع داده مختلف (رشته، عدد صحیح، عدد اعشار) تعریف شده و مقادیر آن‌ها در ادامه استفاده شود. ـ نوشتن کدی که یک نوع داده را به نوع دیگری تبدیل کند و از آن در محاسبات استفاده کند.ـ جواب دادن به سوالات تستی یا تشریحی در مورد نام‌گذاری صحیح متغیرها، تفاوت انواع داده، و زمان استفاده از تبدیل نوع.	۲	
				عدم انجام و یا انجام ناقص مرحله کاری	۱	
۳	عملگرها و ورودی/خروجی	رایانه، vscode نصب‌شده، ۱۵دقیقه	مشاهده عملی	ـ نوشتن برنامه‌ای که از عملگرهای ریاضی (جمع، تفریق، ضرب، تقسیم) و شاید مقایسه‌ای استفاده کند. ـ نوشتن برنامه‌ای که از کاربر اطلاعاتی (دریافت کند و سپس یک خروجی سفارشی نمایش دهد. ـ حل یک مسئله ساده که نیاز به ترکیب ورودی، پردازش با عملگرها و نمایش خروجی دارد (مثلاً محاسبه مساحت مستطیل با دریافت طول و عرض از کاربر). ـ انجام کارها با روش‌های دیگر ـ انجام کنجکاوی ها	۳	
				ـ نوشتن برنامه‌ای که از عملگرهای ریاضی (جمع، تفریق، ضرب، تقسیم) و شاید مقایسه‌ای استفاده کند. ـ نوشتن برنامه‌ای که از کاربر اطلاعاتی (دریافت کند و سپس یک خروجی سفارشی نمایش دهد. ـ حل یک مسئله سادهٔ که نیاز به ترکیب ورودی، پردازش با عملگرها و نمایش خروجی دارد (مثلاً محاسبه مساحت مستطیل با دریافت طول و عرض از کاربر).	۲	
				عدم انجام و یا انجام ناقص مرحله کاری	۱	

۴	ساختارهای داده پایه (لیست) و شرطی	رایانه، vscode نصب‌شده، ۲۰ دقیقه	مشاهده عملی	ـ ایجاد یک لیست، اضافه کردن آیتم جدید به آن، حذف یک آیتم، و نمایش محتوای لیست. ـ نوشتن یک کد با استفاده از if، elif، else یک تصمیم‌گیری انجام شود. ـ ترکیب شرط‌ها با استفاده از and, or, not برای پیاده‌سازی منطق‌های پیچیده‌تر. ـ ساخت یک برنامه ساده که لیست و شرط را ترکیب کند ـ انجام کنجکاوی‌ها ـ انجام کارها با روش‌های دیگر	۳	
				ـ ایجاد یک لیست، اضافه کردن آیتم جدید به آن، حذف یک آیتم و نمایش محتوای لیست ـ نوشتن یک کد با استفاده از if، elif، else یک تصمیم‌گیری انجام شود. ـ ترکیب شرط‌ها با استفاده از and, or, not برای پیاده‌سازی منطق‌های پیچیده‌تر. ـ ساخت یک برنامه ساده که لیست و شرط را ترکیب کند.	۲	
				عدم انجام و یا انجام ناقص مرحله کاری	۱	
۵	حلقه‌های تکرار: for	رایانه، vscode نصب‌شده، ۲۵ دقیقه	مشاهده عملی	ـ نوشتن کدی که روی عناصر یک لیست یا کاراکترهای یک رشته پیمایش کرده و هر کدام را چاپ کند. ـ استفاده از تابع ()range در حلقه for برای اجرای یک دستور به تعداد مشخص یا پیمایش اعداد در یک بازه. ـ نوشتن برنامه‌ای که با استفاده از حلقه for کاری تکراری را انجام دهد ـ انجام کنجکاوی‌ها ـ انجام کار با چند روش	۳	
				نوشتن کدی که روی عناصر یک لیست یا کاراکترهای یک رشته پیمایش کرده و هر کدام را چاپ کند. ـ استفاده از تابع ()range در حلقه for برای اجرای یک دستور به تعداد مشخص یا پیمایش اعداد در یک بازه. ـ نوشتن برنامه‌ای که با استفاده از حلقه for کاری تکراری را انجام دهد	۲	
				عدم انجام و یا انجام ناقص مرحله کاری	۱	
۶	حلقه‌های تکرار و دستورات کنترلی while	رایانه، vscode نصب‌شده، ۲۵ دقیقه	مشاهده عملی	ـ نوشتن برنامه‌ای که تا زمانی که یک شرط برقرار است، ادامه یابد ـ نوشتن کدی که در آن از break برای خروج از حلقه و از continue برای رد شدن از یک تکرار خاص استفاده شود. ـ حل یک مسئله که نیاز به ترکیب هوشمندانه حلقه‌ها (for یا while) و دستورات شرطی دارد. ـ انجام کنجکاوی‌ها ـ انجام کار با چند روش	۳	
				۱ـ نوشتن برنامه‌ای که تا زمانی که یک شرط برقرار است، ادامه یابد ـ نوشتن کدی که در آن از break برای خروج از حلقه و از continue برای رد شدن از یک تکرار خاص استفاده شود. ـ حل یک مسئله که نیاز به ترکیب هوشمندانه حلقه‌ها (for یا while) و دستورات شرطی دارد..	۲	
				عدم انجام و یا انجام ناقص مرحله کاری	۱	
شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش:				احراز همه شایستگی‌های غیرفنی	۲	
				عدم احراز هریک از شایستگی‌های غیرفنی	۱	

ارزشیابی کار (شایستگی انجام کار)	بلی □
	خیر □

توضیحات:

۱ـ معیار شایستگی انجام کار:

ـ کسب حداقل نمره ۲ از مراحل ۱، ۲ و ۳ (مراحل بحرانی)

ـ کسب حداقل نمره ۲ از بخش شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش

ـ کسب حداقل میانگین ۲ از مراحل کار

۲ـ تعریف سطوح شایستگی: صرف‌نظر از اینکه یک تکلیف کاری در چه سطحی از صلاحیت حرفه‌ای انجام می شود، در هر محیط کاری ممکن است انجام هر کار با کیفیت مشخصی مورد انتظار باشد. سطح شایستگی انجام کار، معیار اساسی ارزشیابی است.

مهارت (سطح سه): ماهر و قادر به آموزش و هدایت دیگران، توانایی برنامه‌ریزی و تحلیل، پاسخ‌گویی در برابر کارهای خود، سروکار داشتن با سطح وسیعی از کارها و فعالیت‌ها

تسلط (سطح چهار): خبرگی در انجام کار و آموزش دیگران، ایجاد نوآوری، سازگاری، عیب‌یابی، هدایت و راهنمایی دیگران.

واحد یادگیری ۲
کار با توابع و ماژول ها

آیا تابه‌حال اندیشیده‌اید

- اگر بخواهید یک مجموعه از اعداد بدون تکرار بسازید، یا ترتیب داده‌ها را حفظ کنید و تغییرناپذیر باشد، از چه نوع داده‌ای استفاده می‌کنید؟
- فرض کنید می‌خواهید اطلاعات یک دانش‌آموز مثل نام، سن و نمرات او را ذخیره کنید و سریع به هر مورد دسترسی داشته باشید، چه ساختار داده‌ای مناسب است؟
- اگر بخواهید یک عمل خاص را چند بار در برنامه استفاده کنید، بدون اینکه کد را تکرار کنید، چه راهی وجود دارد؟
- چگونه می‌توانید در یک تابع تعداد زیادی داده را پردازش کنید یا عملیات تکراری انجام دهید، بدون اینکه کد طولانی شود؟
- چگونه می‌توانید از کدهای آماده دیگران استفاده کنید؟ چه تفاوتی بین تابع و متد وجود دارد؟

در پایان از هنرجو انتظار می‌رود

۱ مجموعه‌ها و توپل‌ها را تعریف کند، عناصر آنها را اضافه یا حذف کند و کاربرد هریک را بداند.

۲ دیکشنری بسازد، کلید و مقدار اضافه کند، داده‌ها را با کلیدها بازیابی کند و کاربرد دیکشنری را در ذخیره‌سازی اطلاعات بداند.

۳ هنرجو قادر باشد توابع ساده بنویسد، پارامترها و آرگومان‌ها را استفاده کند و مقدار بازگشتی تابع را دریافت و به کار ببرد.

۴ هنرجو بتواند توابع پیچیده‌تری بنویسد، از حلقه‌ها در توابع استفاده کند و عملیات تکراری را در توابع مدیریت کند.

۵ ماژول‌ها را وارد کند، توابع و کلاس‌های آماده را در برنامه خود استفاده کند و ماژول‌های استاندارد پایتون را شناسایی کند.

۶ ماژول‌های محلی را ایجاد و استفاده کند.

استاندارد عملکرد

هنرجو بتواند توابع و ماژول‌ها را ایجاد و استفاده کند، داده‌ها و عملیات تکراری را با توابع مدیریت کند، از ماژول‌های محلی بهره ببرد و تفاوت بین تابع و متد را تشخیص دهد.

در برنامه‌نویسی، هرچه برنامه‌ها بزرگ‌تر و پیچیده‌تر می‌شوند، نیاز به سازماندهی و استفاده مجدد از کد بیشتر حس می‌شود. توابع و ماژول‌ها ابزارهایی هستند که به برنامه‌نویس کمک می‌کنند تا کدهای خود را مرتب، قابل فهم و قابل استفاده دوباره بسازد. در این واحد یادگیری، شما با ساخت توابع، استفاده از پارامترها و مقدار بازگشتی، نوشتن توابع پیشرفته با حلقه‌ها و همچنین استفاده از ماژول‌های آماده و ماژول‌های محلی آشنا می‌شوید. با یادگیری این مفاهیم، می‌توانید برنامه‌های سازمان‌یافته‌تر و حرفه‌ای‌تری ایجاد کنید.

مجموعه‌ها (set)



شکل ۹۷

مجموعه‌ها برای ذخیره چند مقدار در یک متغیر استفاده می‌شوند. مجموعه یکی از چهار نوع دادهٔ پایتون است که برای ذخیره‌سازی انواع داده‌ها مثل اعداد صحیح، اعداد اعشاری، رشته‌ها، بولین‌ها و توپل‌ها (tuple) به‌صورت مجموعه‌ای به‌کار می‌رود.

ساخت set :

استفاده از آکولاد {}:

این روش معمولاً برای ساختن مجموعه‌ای از عناصر مشخص و از پیش تعیین‌شده استفاده می‌شود.
پس از هر بار اجرای برنامه مشاهده خواهید کرد که ترتیب نمایش اعضای مجموعه فرق می‌کند (شکل۹۸).
یعنی set ترتیب نمایش عناصر را تضمین نمی‌کند و Unorder است.

```
fruitset = {"apple", "banana", "cherry"}
print(fruitset)
```

۱
```
{'cherry', 'apple', 'banana'}
```
۲
```
{'banana', 'apple', 'cherry'}
```
۳
```
{'cherry', 'banana', 'apple'}
```

شکل ۹۸

استفاده از متد (add) برای اضافه‌کردن عنصر جدید به set:

برنامه را اجرا کنید. مشاهده خواهید کرد که عناصر تکراری به مجموعه اضافه نشده‌اند (شکل۹۹).

```
fruits = {"apple", "banana", "orange"}
fruits.add("grape")
fruits.add("grape")
print(fruits)
```

```
{'apple', 'orange', 'grape', 'banana'}
```

شکل ۹۹

استفاده از متد (remove) برای حذف عنصر از set:

با استفاده از متد remove می‌توانید عناصر یک set را حذف کنید.

مثال: با استفاده از متد remove عنصر "banana" از مجموعه حذف شده است (شکل۱۰۰).

```
fruits = {"apple", "banana", "orange"}
fruits.remove("banana")
print(fruits)
```

```
{'apple', 'orange'}
```

شکل ۱۰۰

درصورتی‌که عضو وجود نداشته باشد در زمان اجرا پیغام خطا نشان داده می‌شود (شکل۱۰۱).

```
fruits = {"apple", "banana", "orange"}
fruits.remove("banana")
print(fruits)
fruits.remove("grape")
```

```
{'apple', 'orange'}
Traceback (most recent call last):
  File "c:\Users\          \Desktop\python\sets.py", line 4, in <module>
    fruits.remove("grape")
    ~~~~~~~~~~~~~~~~~~~~~^^^^^^^^^
KeyError: 'grape'
```

شکل ۱۰۱

کنجکاوی

در مورد سایر متدهای مربوط به set از سایت www.w3schools.com تحقیق کنید و در کلاس ارائه دهید.

استفاده از in برای بررسی وجود عنصر در set:



شکل ۱۰۲

مثال: در کد شکل ۱۰۳ وجود عنصر"**grape**" در مجموعه **fruits** مورد بررسی قرار گرفته است. خروجی کد نشان می دهد این عنصر در مجموعه وجود ندارد.

```
fruits = {"apple", "banana", "orange"}
if "grape" in fruits:
    print("grape is available")
else:
    print("grape is not available")
```

```
grape is not available
```

شکل ۱۰۳

فعالیت

برنامه‌ای بنویسید که یک **set** از اعداد داشته باشد و بررسی کند عدد ۱۰ در **set** وجود دارد یا خیر.

فعالیت

مجموعه‌ای از میوه‌ها وجود دارد:

fruits = {"apple", "banana", "orange", "grape"}

برنامه‌ای بنویسید که کاربر نام میوه‌ای وارد کند و اگر در **set** بود، پیغام دهد که آن میوه درون مجموعه موجود است و **set** نهایی را چاپ کنید.

دیکشنری (dictionary)

نوع داده dictionary شبیه دفترچه تلفن است که هر نام با یک شماره‌تلفن مرتبط است.

ـ دیکشنری‌ها از تعدادی جفت **کلید: مقدار** (key:value) تشکیل شده‌اند که به هر یک از آنها **item** گفته می‌شود. هر کلید به یک مقدار خاص اشاره دارد.

ـ برای ایجاد dictionary آیتم‌ها درون {} قرار گرفته و با **کاما** از یکدیگر جدا می‌شوند. همچنین هر کلید و مقدار آن با علامت **دونقطه** از هم جدا شده است.



شکل ۱۰۴

در مثال شکل۱۰۵ یک dictionary شامل سه آیتم (کلید و مقدار) به نام cardic تعریف شده است:

```
cardic = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964,
}
print(cardic)
```

```
{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
```

شکل ۱۰۵

ـ برای دسترسی به یک مقدار در dictionary بعد از نام dictionary، کلید، داخل براکت [] قرار داده می‌شود (شکل۱۰۶).

مثال: در کد شکل۱۰۶ یک دیکشنری به نام person با ۳ آیتم (کلید، مقدار) تعریف شده است. برای نمایش هر یک از عناصر داخل این دیکشنری از براکت [] استفاده شده است.

```
1  person = {
2      "name": "Arash",
3      "age": 20,
4      "city": "Tehran"
5  }
6
7  print(person["name"])
8
9  print(person["age"])
```

```
Arash
20
```

شکل ۱۰۶

نکته

اگر هر عنصر دارای عنوان مشخص باشد (مانند نام دانش‌آموز و نمره او)، استفاده از دیکشنری بهتر از لیست است؛ زیرا در دیکشنری می‌توان مستقیماً با نام کلید به مقدار معادل آن دسترسی داشت و نیازی به دانستن اندیس عنصر نیست.

فعالیت

یک دیکشنری به نام book بسازید که اطلاعات یک کتاب را نگه دارد:
"title": "python Basic"
author: "Fatemeh"
pages:250
نام نویسنده (author) و تعداد صفحات (pages) را چاپ کنید.

ـ کلیدهای متفاوت می‌تواند مقادیر یکسان داشته باشند.
مثال: دیکشنری status دارای دو آیتم است (شکل ۱۰۷).
آیتم اول: "user1":"active"
آیتم دوم: "user2":"active" کلیدها یکسان هستند ولی مقادیر هریک از آنها متفاوت است.
ـ در dictionary کلیدهای یکسان نمی‌توانند مقادیر یکسان داشته باشند. کلیدهای تکراری باعث بازنویسی مقدار قبلی می‌شوند، نه مقدارهای تکراری.

```
status = {
    "user1": "active",
    "user2": "active"
}
print(status)
```

```
{'user1': 'active', 'user2': 'active'}
```

شکل ۱۰۷

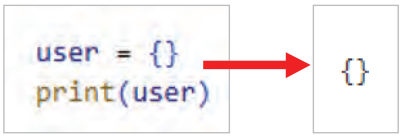
در مثال شکل ۱۰۸ دیکشنری car_dic دارای پنج آیتم است که دو تای آنها دارای کلیدهای با نام یکسان هستند. پس مقدار جدید به جای مقدار قبلی جایگزین می‌شود.

```
car_dic = {
    "brand": "ford",
    "model": "motagh",
    "year": 1964,
    "year": 1920 ,
    "age": 20
}
print(car_dic)
```

```
{'brand': 'Ford', 'model': 'Mustang', 'year': 1920}
```

شکل ۱۰۸

ـ برای ساخت یک dictionary خالی از {} استفاده می‌شود (شکل ۱۰۹).
مثال: یک دیکشنری خالی با نام user تعریف شده است.



شکل ۱۰۹

اضافه‌کردن item جدید به dictionary:

برای اضافه‌کردن یک عضو جدید به dictionary در پایتون، کافی است یک **کلید جدید** را مقداردهی کنید (شکل ۱۱۰).



شکل ۱۱۰

فعالیت

یک dictionary به نام product با کلیدهای name price ، بسازید. قیمت محصول درون dictionary را تغییر دهید و سپس dictionary را چاپ کنید.

مثال: یک dictionary خالی به نام student بسازید. سپس با استفاده از ورودی name و grade را به dictionary اضافه و در پایان دیکشنری را چاپ کنید (شکل ۱۱۱).

```
1   student = {}
2
3   student["name"] = input("enter name : ")
4   student["grade"] = int(input("enter grade : "))
5   print(student)
```

```
enter name : bahar
enter grade : 16
{'name': 'bahar', 'grade': 16}
```

شکل ۱۱۱

فعالیت

یک dictionary خالی به نام user بسازید. از ورودی مقادیری برای کلیدهای username و password و Age را بگیرید و داخل dictionary ذخیره و در پایان کل اطلاعات کاربر را چاپ کنید.



شکل ۱۱۲

حذف عنصر از dictionary:

روش‌های مختلفی برای حذف آیتم از dictionary وجود دارد ازآن‌جمله:

ـ در پایتون برای حذف یک آیتم از dictionary، از دستور del استفاده می‌شود. با این دستور، کلید و مقدار مربوط به آن به طور کامل از dictionary حذف می‌شود.

```
user = {
    "username": "ali",
    "age": 20,
    "city": "Tehran"
}
del user["age"]
print(user)
```

```
{'username': 'ali', 'city': 'Tehran'}
```

شکل ۱۱۳

مثال: در مثال کلید age و مقدار آن ۲۰، از dictionary حذف شده است. برنامه را بنویسید و نتیجه را مشاهده کنید (شکل۱۱۳). در برنامه مقابل برای حذف آیتم "age": 20 از دستور del استفاده شده است. مشاهده می‌کنید که با این دستور کلید و مقدار آن حذف شده‌اند.

```
user = {
    "username": "ali",
    "age": 20,
    "city": "Tehran"
}
if "age" in user:
    del user["age"]
print(user)
```

شکل ۱۱۴

ـ درصورتی‌که قصد حذف کلیدی را دارید که در dictionary وجود نداشته باشد، برنامه با خطا مواجه می‌شود؛ بنابراین قبل از حذف، باید وجود کلید بررسی شود. در برنامه مقابل برای جلوگیری از خطا پیش از حذف عنصر از لیست، وجود آن بررسی شده است. برنامه را وارد کرده و نتیجه را مشاهده کنید (شکل۱۱۴).

ـ برای **حذف عنصر** از dictionary می‌توانید از **متد** ()**pop** نیز استفاده کنید. متد ()pop، عنصری که نام کلید مشخص شده را دارد، حذف می‌کند. dictionary.pop(key)

مثال: در کد (شکل۱۱۵) عنصر "price" از دیکشنری product با استفاده از دستور product.pop("price") حذف شده است.

```
product = {
    "name": "Laptop",
    "price": 1500,
    "stock": 10
}

product.pop("price")
print(product)
```

```
{'name': 'Laptop', 'stock': 10}
```

شکل ۱۱۵

ـ متد ()pop علاوه بر حذف آیتم، می‌تواند مقدار آیتم حذف‌شده را برگرداند (شکل۱۱۶).

```
product = {
    "name": "Laptop",
    "price": 1500,
    "stock": 10
}

price_value = product.pop("price")
print("Price removed:", price_value)
print(product)
```

```
Price removed: 1500
{'name': 'Laptop', 'stock': 10}
```

شکل ۱۱۶

فعالیت

یک دیکشنری به نام inventory بسازید که شامل:

inventory = {"Bag": 5, "Ball": 8, "Laptop": 3,
"Mouse": 10, "Flash memory": 14}

وجود کالایی با نام "Bag" را بررسی و در صورت وجود، پیغام مناسبی چاپ شود که نشان‌دهندهٔ وجود کالا باشد و سپس آن آیتم را با متد ()pop حذف کنید.

مقدار حذف‌شده را چاپ کنید.

اگر کالا وجود نداشت، پیغامی مبنی بر عدم وجود آن نمایش داده شود.

سپس dictionary جدید را چاپ کنید.

پروژه بازی Quiz



شکل ۱۱۷

در این بازی ابتدا منویی که شامل انواع سؤالات است نمایش داده می‌شود و کاربر سؤال موردعلاقه‌اش را انتخاب می‌کند و سپس به سؤالات پاسخ می‌دهد، در صورت پاسخ صحیح یک امتیاز دریافت می‌کند و در غیر این صورت پیام پاسخ و در صورت تمام‌شدن سؤالات، امتیاز کاربر نمایش داده می‌شود.

مرحله ۱: در این مرحله باید منویی ایجاد کنید که شامل انواع سؤالات باشد. برای شروع دو نوع سؤال در منو قرار دهید. در دیکشنری menu به هر عدد یک نوع سؤال اختصاص‌یافته است. در صورت انتخاب عدد ۳، کاربر از بازی خارج می‌شود (شکل۱۱۸).

```
1  # منو و گزینه‌ها
2  menu = {
3      "1": "Math Quiz",
4      "2": "Geography Quiz",
5      "3": "Exit"
6  }
```

شکل ۱۱۸

مرحله ۲: سؤالات مربوط به هر نوع سؤال را بنویسید. دو دیکشنری math_quiz برای سؤالات ریاضی و geo_quiz برای سؤالات مربوط به مکان‌های جغرافیایی، هر کدام سه عنصر که شامل سؤال و پاسخ آن است (شکل۱۱۹).

```
math_quiz = {
    "2 + 2 = ?": "4",
    "5 * 3 = ?": "15",
    "10 - 7 = ?": "3"
}

geo_quiz = {
    "Capital of Iran?": "Tehran",
    "Capital of France?": "Paris",
    "Capital of Japan?": "Tokyo"
}
```

شکل ۱۱۹

مرحله ۳: در این مرحله دستوراتی بنویسید که منوها را نمایش دهد.
بعد از نمایش پیام خوشامدگویی و نمایش عنوان :menu برای هر یک از عناصر داخل دیکشنری، شماره menu و متن آن با یک علامت "-" از هم جدا شده و نمایش داده می‌شود تا کاربر بتواند از میان آنها یکی را انتخاب کند (شکل ۱۲۰).

```
print("Welcome to Quiz Game!")
print("Menu:")
for key in menu:
    print(key, "-", menu[key])

choice = input("Select an option: ")
score = 0
```

شکل ۱۲۰

مرحله ۴: در این مرحله دستورات مربوط به انتخاب منو را بنویسید (شکل ۱۲۱).

```
score = 0

if choice == "1":
    print("\nMath Quiz")
    for question in math_quiz:
        answer = input(question + " ")
        if answer == math_quiz[question]:
            print("Correct!")
            score += 1
        else:
            print("Wrong!")

elif choice == "2":
    print("\nGeography Quiz")
    for question in geo_quiz:
        answer = input(question + " ")
        if answer == geo_quiz[question]:
            print("Correct!")
            score += 1
        else:
            print("Wrong!")

elif choice == "3":
    print("Exiting the game...")

else:
    print("Invalid option")
```

شکل ۱۲۱

نکته

هر جایی که \n نوشته شود، خروجی از آن نقطه به خط بعدی می‌رود.

مرحله ۵: در این مرحله دستوراتی بنویسید که امتیازات کسب‌شده را نمایش دهد. امتیاز کاربر در صورت انتخاب گزینه "۱" و یا "۲" نمایش داده می‌شود (شکل ۱۲۲).

```
if choice in ["1", "2"]:
    print("\nYour final score is:", score)
```

شکل ۱۲۲

مرحله ۶: کدها را نوشته و برنامه را اجرا کنید!

فعالیت

بخش‌های جدید Quiz (مثلاً Sport ,Science) را اضافه کنید.
برای جواب غلط، یک پیام توضیحی بدهید.
امکان انتخاب چند بخش به‌صورت پشت‌سرهم را اضافه کنید.

کنجکاوی

با مراجعه به دیکشنری‌های مرجع تلفظ لغات را دقیق یاد بگیرید.
https://dictionary.cambridge.org/dictionary/english/tuple

توپل (Tuple)

توپَل در پایتون دنباله‌ای از داده‌های ترتیبی و غیر قابل تغییر است. به کمک توپل‌ها می‌توانید چندین مقدار از نوع‌های داده‌ای مختلف را در یک متغیر ذخیره و روی آنها عملیات انجام دهید. این ساختار داده را زمانی استفاده کنید که بخواهید داده‌ها در طول برنامه، تصادفاً توسط خود شما و یا سایر برنامه‌نویسان هم تیمی شما، تغییر نکنند.



شکل ۱۲۳

ایجاد Tuple:

tuple_name = (data1, data2, data3,...)

مثال: یک tuple با سه عضو برای نگهداری نام کاربری مدیران ارشد شرکت ایجاد و سپس آن را چاپ کند (شکل۱۲۴).

```
usernames=("super_admin","manager_admin","teacher_admin")
print(usernames)
```

```
('supper_admin', 'manager_admin', 'teacher-admin')
```

شکل ۱۲۴

مثال: در برنامهٔ شکل ۱۲۵ از یک توپِل برای نگهداری نام روزهای هفته استفاده شده است، زیرا نام روزهای هفته تغییرپذیر نیستند، پس این نوع ساختار داده مناسب نگهداری این مقادیر می‌باشد.

```
days = ("Monday", "Tuesday", "Wednesday",
        "Thursday", "Friday", "Saturday", "Sunday")
print(days)
```

```
('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday')
```

شکل ۱۲۵

دسترسی به عناصر Tuple:

ـ مانند لیست‌ها عناصر توپل هم دارای index هستند و دسترسی به عنصر با شمارهٔ ایندکس آن انجام می‌شود.

مثال: در این برنامه ابتدا یک توپل به نام name_port برای نگهداری نام درگاه‌های ورودی یک سیستم رایانه‌ای ایجاد و سپس عنصر سوم آن نمایش داده شده است. دقت کنید ایندکس عناصر از صفر است (شکل۱۲۶).

```
name_port = ("LPT1", "USB1", "USB2", "HDMI")
print(name_port[2])
```

```
USB2
```

شکل ۱۲۶

-عناصر یک توپل تغییرناپذیر (immutable) هستند. یعنی اگر بخواهید بعد از مقداردهی اولیه، آنها را تغییر دهید، برنامه پیغام خطا نمایش می‌دهد.

مثال: در مثال شکل۱۲۷ تلاش شده است که مقدار عنصر دوم توپل تغییرکند، ولی اجرای برنامه با خطا مواجه شده است.

```
daru=("Astaminofen","Asprin")
daru[1]= "adult cold"
print("daru")
```

```
Traceback (most recent call last):
  File "c:\Users\2948693748\Desktop\py_progs\py_files\daru.py", line 2, in <module>
    daru[1]= "adult cold"
    ~~~~^^^
TypeError: 'tuple' object does not support item assignment
```

شکل ۱۲۷

کنجکاوی

تحقیق کنید آیا می‌توانید تعداد عناصر یک توپل را تغییر دهید یا نه؟ برنامه‌ای بنویسید که نتیجه تحقیق شما را نشان دهد.
به نظر شما مزیت اصلی توپل‌ها چیست که برنامه‌نویس را ملزم می‌کند از آن استفاده کند؟

■ **حذف Tuple:** در پایتون امکان حذف عناصر یک توپل وجود ندارد. ولی می‌توانید کل یک توپل را با استفاده از دستور del حذف کنید.

مثال: در این برنامه با دستور del توپل days حذف شده است. بعد از اجرای آن هم پیغامی مبنی عدم تعریف توپل days نمایش داده می‌شود (شکل۱۲۸).

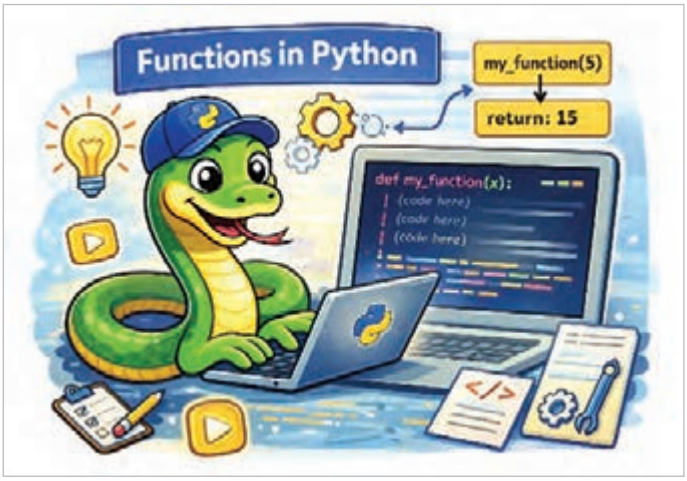
```
days = ("Monday", "Tuesday", "Wednesday",
        "Thursday", "Friday", "Saturday", "Sunday")
del days
print(days)
```

```
Traceback (most recent call last):
  File "c:\Users\2020101030\Downloads\tup.py", line 4, in <module>
    print(days)
          ^^^^
NameError: name 'days' is not defined
```

شکل ۱۲۸

تابع (Function)

تابع، بلاک یا قطعه کدی است که یک‌بار آن را تعریف کرده و بارها آن را در چندین مکان برنامه می‌توان فراخوانی کرد.



شکل ۱۲۹

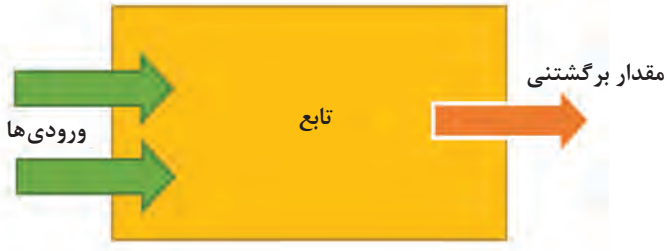
مزایای استفاده از تابع:

- برنامه‌نویس با تعریف یک تابع، تنها یک بار کد مورد نظر را می‌نویسد و سپس در هر نقطه‌ای از برنامه یا حتی در پروژه‌های دیگر که نیاز باشد، آن را فراخوانی و استفاده می‌کند.
- تقسیم کد به توابع کوچک و مشخص، درک و فهم کد را به طور چشمگیری افزایش می‌دهد.
- یافتن و رفع اشکالات و اعمال تغییرات در کد بسیار ساده‌تر می‌شود.
- سازماندهی کدها به‌صورت منطقی و مرتب و ساختار کلی برنامه آسان‌تر می‌شود.

تعریف تابع: یک تابع با استفاده از کلمه کلیدی def تعریف می‌شود و به دنبال آن نام تابع و پرانتز قرار می‌گیرد.

فرم کلی تعریف تابع با پارامتر ورودی به شکل زیر است:

def نام تابع (پارامترهای تابع):
دستور یا دستورات بدنه تابع
return مقدار یا مقادیر



شکل ۱۳۰

مثال: در این برنامه تابعی به نام show_store_name تعریف شده و یک دستور print ("Digital Online Store") در آن نوشته شده است (شکل ۱۳۱).

```
1   def show_store_name():
2       print("Digital Online Store")
```

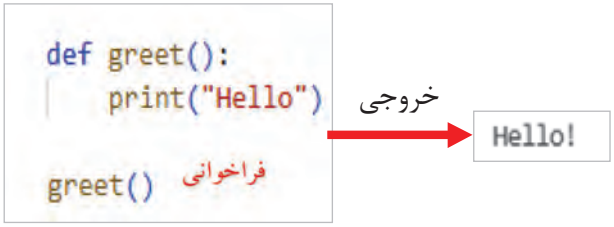
شکل ۱۳۱

مثال: در این برنامه نیز یک تابع به نام greet تعریف شده است و دستور print("Hello! ") در آن نوشته شده است (شکل ۱۳۲).

```
1   def greet():
2       print("Hello!")
```

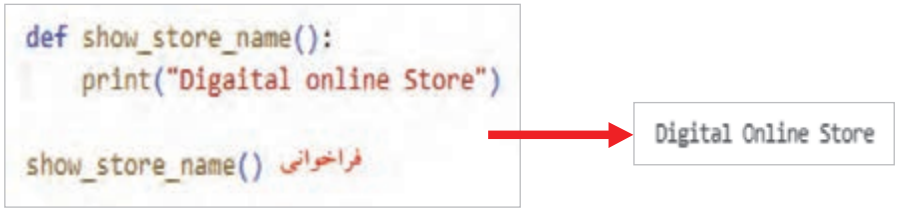
شکل ۱۳۲

مثال: در این برنامه تابع greet() بدون پارامتر تعریف شده است، پس در هنگام فراخوانی نیازی به ارسال آرگومان ندارد (شکل ۱۳۳).



شکل ۱۳۳

مثال: در این برنامه نیز تابع بدون پارامتر تعریف شده است. در هنگام فراخوانی نیازی به ارسال آرگومان ندارد (شکل ۱۳۴).



شکل ۱۳۴

پارامتر و آرگومان (Parameters & Arguments)

پارامتر چیست؟ تابع می‌تواند ورودی‌هایی را دریافت کند. به ورودی‌های تابع در زمان تعریف تابع، پارامتر (parameter) گفته می‌شود.

آرگومان چیست؟ ورودی‌های تابع که هنگام فراخوانی تابع به آن داده می‌شود.

ـ پس از تعریف، تابع باید فراخوانی شود تا اجرا شود. اگر تابع را فقط تعریف کنید ولی فراخوانی نشود هیچ اتفاقی نمی‌افتد. مثل این است که اصلاً تعریف نشده باشد.

ـ برای فراخوانی تابع نام آن را نوشته و به تعداد پارامترها، به آن آرگومان ارسال کنید (در اصطلاح پاس‌دادن گفته می‌شود) یک تابع را بارها می‌توانید فراخوانی کنید.

ـ نحوه فراخوانی تابع به شکل زیر است:

(آرگومان‌های تابع) :نام تابع

مثال: در مثال شکل۱۳۵ تابع show_product(name, price)

با دو پارامتر name , price تعریف شده است. در زمان فراخوانی هم باید دو مقدار به‌عنوان آرگومان ارسال شود.

```
1  def show_product(name, price):
2      print("Product:", name, "Price:", price)
3
4  show_product("Laptop", 50000000)
5  show_product("Mobile", 30000000)
```

فراخوانی

خروجی

```
Product: Laptop Price: 50000000
Product: Mobile Price: 30000000
```

شکل ۱۳۵

مثال: در مثال شکل۱۳۶ تابع print_age(age) دارای یک پارامتر به نام age است. در زمان فراخوانی هم یک آرگومان به آن ارسال شود. همان‌طور که مشاهده می‌کنید تابع print_age(age) دو بار و با دو آرگومان متفاوت فراخوانی شده است.

```
def print_age(age):
    print("you are" +str(age) + "years old.")

print_age(20)
print_age(25)
```

فراخوانی

خروجی

```
You are 20 years old.
You are 25 years old.
```

شکل ۱۳۶

نکته

ترتیب، تعداد و نوع پارامترها (در زمان تعریف تابع) با ترتیب، تعداد و نوع آرگومان‌ها (در زمان فراخوانی تابع) باید با هم یکسان باشد.

اگر ترتیب و نوع ارسال آرگومان‌ها رعایت نشود، خروجی اشتباه تولید می‌شود. در هنگام اجرا خطا نمی‌دهد؛ ولی منطق برنامه نادرست است.
راه‌حل، استفاده از **آرگومان کلیدی** (keyword arguments) یا همان آرگومان‌های با نام است، یعنی می‌توان در زمان فراخوانی تابع، نام پارامتر را به‌همراه عملگر انتساب نوشت.
مثال: به کدهای شکل ۱۳۷ دقت کنید. خروجی‌ها را بررسی کنید. فراخوانی بدون رعایت ترتیب آرگومان‌ها، باعث شده اول قیمت و سپس نام کالا چاپ شود و این منطق نادرستی است.

```
1  def show_product(name, price):
2      print("Product:", name, "Price:", price)
3
4  show_product(50000000 ,"Laptop")  فراخوانی بدون رعایت ترتیب
5  show_product("Mobile", 30000000)  فراخوانی با رعایت ترتیب
```

خروجی

```
Product: 50000000 Price: Laptop
Product: Mobile Price: 30000000
```

شکل ۱۳۷

مثال: در مثال شکل ۱۳۸ برای جلوگیری از خروجی اشتباه با ارسال آرگومان‌ها بدون ترتیب، از آرگومان‌های کلیدی استفاده شده است. برنامه را نوشته و نتیجه را با مثال قبل مقایسه کنید.

```
1  def show_product(name, price):
2      print("Product:", name, "Price:", price)
3
4  show_product(price=50000000, name="Laptop")
5  show_product(price=30000000, name=" Mobile Price")
```

شکل ۱۳۸

نکته

وقتی آرگومان‌ها **بدون نام پارامتر** به تابع داده شود، ترتیب ارسال آنها مهم است، چون هر آرگومان **به‌ترتیب به پارامترهای تابع نسبت داده می‌شود**. به این آرگومان‌ها **آرگومان‌های موقعیتی** (**positional arguments**) گفته می‌شود.

فعالیت

یک تابع به نام square بنویسید که یک عدد بگیرد و مربع آن را چاپ کند.

نکته

تابعی بنویسید که دو عدد را به‌عنوان پارامتر دریافت و حاصل‌جمع آنها را نشان دهد.

نکته

تابعی بنویسید که نام ۳ کاربر را از ورودی به‌عنوان پارامتر دریافت و آنها را چاپ کند.

مثال: تعداد آرگومان‌های ارسالی به تابع باید با تعداد پارامترهای آن در زمان تعریف تابع برابر باشد. در غیر این صورت در زمان اجرای پیام خطا نمایش داده می‌شود (شکل۱۳۹).

```
1   def show_product(name, price):
2       print("Product:", name, "Price:", price)
3
4   show_product("Laptop")
```

تعداد آرگومان ارسال شده کمتر از تعداد پارامترهای تابع است

```
Traceback (most recent call last):
  File "c:\Users\2948693748\Desktop\python\func2.py", line 4, in <module>
    show_product("Laptop")
    ~~~~~~~~~~~~^^^^^^^^^^
TypeError: show_product() missing 1 required positional argument: 'price'
```

پیام خطای نمایش داده شده پس از اجرای برنامه

شکل ۱۳۹

نکته

می‌توان برای پارامتر مقدار پیش‌فرض تعیین کرد. اگر آرگومان داده نشود، **مقدار پیش‌فرض** استفاده می‌شود.

مثال: در اولین خط کد شکل ۱۴۰، تابع با یک پارامتر تعریف شده است. در خط ۴ هنگام فراخوانی، آرگومان آن ارسال نشده است، بنابراین از مقدار پیش‌فرض آن "Tehran" استفاده می‌شود. در خط ۵ تابع با آرگومان فراخوانی شده است. برنامه را نوشته و نتیجه را بررسی کنید. مقدار پیش‌فرض را تغییر دهید و برنامه را اجرا کنید.

```
1   def delivery(city="Tehran"):
2       print("Delivery city:", city)
3
4   delivery()
5   delivery("Isfahan")
```

```
Delivery city: Tehran
Delivery city: Isfahan
```

شکل ۱۴۰

مثال: در این مثال تابع با دو پارامتر تعریف شده که مقدار پیش‌فرض برای آنها وجود دارد. در هر بار فراخوانی آرگومان‌ها با روش‌های مختلف ارسال شده‌اند. برنامه را نوشته، نتیجه را بررسی کنید. مقدار پیش‌فرض را تغییر دهید و دوباره برنامه را اجرا کنید (شکل ۱۴۱).

```
1  def make_tshirt(size="Large", text="I love Python"):
2      print("Size:",  size)
3      print("Text:", text)
4      print("--------------")
5
6  make_tshirt()
7  make_tshirt("Medium")
8  make_tshirt(text="coding is good")
9  make_tshirt("Small", "Coding is cool")
```

```
Size: Large
Text: I love Python
--------------
Size: Medium
Text: I love Python
--------------
Size: Large
Text: coding is good
--------------
Size: Small
Text: Coding is cool
--------------
```

شکل ۱۴۱

مقدار بازگشتی تابع (Return Value)

تابع می‌تواند **مقداری را محاسبه کند و به برنامه برگرداند** تا بعداً از آن استفاده شود که به آن مقدار بازگشتی (**Return Value**) گفته می‌شود. بدون return تابع کار خودش را انجام می‌دهد (مثلاً چاپ می‌کند) ولی هیچ مقداری به بیرون برنمی‌گرداند و اگر بخواهید نتیجه تابع را در متغیر ذخیره کنید مقدار آن None می‌شود (شکل ۱۴۲).

```
1  def add(a, b):
2      print(a + b)
3
4  add(3, 5)
```

شکل ۱۴۲

مثال: با اجرای برنامه شکل ۱۴۳ مقدار ۸ نمایش داده می‌شود. اما عدد ۸ **ذخیره نشده** و همچنین جای دیگری نمی‌توان از آن استفاده کرد.
حالا برنامه را با استفاده از return تغییر داده و نتیجه را مشاهده کنید.
این بار: تابع عدد **۸ را برگردانده** و مقدار بازگشتی در متغیر result ذخیره می‌شود و می‌توان دوباره از آن استفاده کرد.

```
1  def add(a, b):
2      return a + b
3
4  result = add(3, 5)
5  print(result)
```

شکل ۱۴۳

مثال: بدون استفاده از return فقط (price * count) محاسبه می‌شود و هیچ مقداری را به برنامه برنمی‌گرداند، پس با فراخوانی تابع total_ price (۲,۲۰۰۰۰۰) در متغیر x مقدار None ذخیره و با اجرای برنامه مقدار None هم چاپ می‌شود (شکل ۱۴۴).

```
def total_price(price, count):
    print(price * count)


x = total_price(200000, 2)
print(x)
```

```
400000
None
```

شکل ۱۴۴

مثال: با استفاده از return مقدار محاسبه‌شده به برنامه برگردانده می‌شود و می‌توان آن را در متغیر x ذخیره کرد و با چاپ متغیر x مقدار ذخیره‌شده، نمایش داده می‌شود (شکل ۱۴۵).

```
1  def total_price(price, count):
2      return  (price * count)
3
4  x=total_price(200000, 2)
5  print (x)
```

```
400000
```

شکل ۱۴۵

فعالیت

یک تابع به نام avg_numbers بسازید که سه عدد دریافت کرده و میانگین آنها را برگرداند.

مثال: تابعی بنویسید که قیمت کل خرید را حساب کند و اگر تعداد خرید بیشتر از ۵ بود، ۱۰٪ تخفیف بدهد.
داخل تابع از if استفاده شده تا شرطی بررسی شود.
اگر شرط (count > ۵) برقرار باشد مقدار با تخفیف برگردانده می‌شود.
اگر شرط برقرار نباشد، مقدار بدون تخفیف برگردانده می‌شود.
از return استفاده شده است تا نتیجه محاسبه شده به بیرون تابع برگردد و بتوان آن را چاپ یا ذخیره کرد (شکل ۱۴۶).

```
1   def total_price(price, count):
2       if count > 5:
3           return price * count * 0.9
4       else:
5           return price * count
6
7   x = total_price(200000, 7)
8   print(x)
9
10  y = total_price(200000, 3)
11  print(y)
```

شکل ۱۴۶

استفاده از حلقه در داخل تابع:

استفاده از حلقه در داخل تابع به این منظور است که بتوانید **دستورات مشخصی** را چندین بار و **به‌صورت خودکار**، تا **زمانی‌که شرط تعیین‌شده برقرار** است **یا تعداد دفعات مشخصی تکرار شوند**، اجرا کنید. این کار باعث **کاهش تکرار کد، افزایش خوانایی** و **بهینه‌سازی عملکرد تابع** می‌شود.
مثال: در این مثال، تابع تعریف شده اعداد طبیعی کوچک‌تر از ۶ را چاپ می‌کند. برنامه را اجرا و نتیجه را مشاهده کنید (شکل ۱۴۷).

```
def show_numbers():
    for i in range(1, 6):
        print("python")

show_numbers()
```

شکل ۱۴۷

مثال: ابتدا تابع تعریف شده یک پارامتر به نام names دارد که انتظار می‌رود آرگومان‌های دریافتی آن لیستی از اسامی باشند (شکل۱۴۸).
با حلقه for عناصر لیست را یکی‌یکی دریافت و به متغیر name اختصاص می‌دهد.
دستور print هم داخل حلقه اجرا شده و اسامی را جداگانه چاپ می‌کند.
قبل از فراخوانی تابع در اولین خط از برنامه، لیست students که شامل نام دانش‌آموزان است ایجاد می‌شود.
سپس تابع با آرگومان students فراخوانی می‌شود. حلقه داخل تابع یکی‌یکی اسامی داخل لیست را چاپ می‌کند.

```
def show_names(names):
    for name in names:
        print(name)

students = ["saman", "sohrab", "Reza"]
show_names(students)
```

```
saman
sohrab
Reza
```

شکل ۱۴۸

فعالیت

تابع نوشته‌شده حاصل جمع تمام اعداد یک لیست را محاسبه می‌کند. برنامه را بنویسید و اجرا کنید. با راهنمایی هنرآموز محترم خود خطوط کد نوشته‌شده در آن را توضیح دهید. تابع را به نحوی تغییر دهید که عناصر لیست را کاربر از صفحه‌کلید وارد کند.

```
def sum_list(numbers):
    total = 0
    for num in numbers:
        total += num
    return total

my_list = [10, 20, 30, 40]
result = sum_list(my_list)
print(result)
```

فعالیت

تابع نوشته‌شده که طول رشته ورودی کاربر را محاسبه و نمایش می‌دهد. برنامه را اجرا و آن را بررسی و همچنین با راهنمایی هنرآموز محترم خود کدهای برنامه مقابل را توضیح دهید.

```
def string_length(text):
    count = 0
    for char in text:
        count += 1
    return count
user_text=input("Enter your text")
length = string_length(user_text)
print("Length:", length)
```

فعالیت

کدهای مقابل بیشترین مقدار را در بین اعداد یک لیست محاسبه کرده و نمایش می‌دهد. کد را اجرا و خطوط آن را توضیح دهید.

```
def max_number(numbers):
    max_num = numbers[0]
    for num in numbers:
        if num > max_num:
            max_num = num
    return max_num

numbers = [10, 5, 20, 8, 15, 19, 18, 13, 14]
print("Max:", max_number(numbers))
```

فعالیت

تابعی بنویسید که تفاوت نمرهٔ دو دانش‌آموز را محاسبه و مقدار آن را برگرداند. با راهنمایی هنرآموز محترم خود کدهای برنامهٔ مقابل را توضیح دهید.

```
def num_diff(score1, score2):
    diff = score1 - score2
    if diff < 0:
        diff = - diff
    return diff

score1 = int(input("Enter score1 :"))
score2 = int(input("Enter score2 :"))
print(num_diff(score1, score2))
```

انواع توابع در پایتون

User-Defined Function (توابعی که برنامه‌نویس تعریف می‌کند): توابعی که تاکنون تعریف کردید در این دسته قرار دارند.

Built-in functions (یا توابع داخلی پایتون): توابعی که در پایتون از قبل تعریف شده‌اند و برنامه‌نویس آنها را در برنامه‌های خود استفاده می‌کند. کارهای رایج و تکراری را سریع، ساده و بدون نوشتن کد طولانی انجام می‌دهد. به کمک توابع داخلی، برنامه‌ها خواناتر، کوتاه‌تر و کم‌خطا می‌شوند.

در بخش‌های پیش با برخی از این توابع آشنا شدید، در ادامه نیز کاربرد برخی دیگر را یاد خواهید گرفت.

توابع ریاضی: این توابع فقط یا عمدتاً روی مقادیر عددی کار می‌کنند (جدول۴).

جدول۴

نام تابع	کاربرد	مثال	خروجی
abs()	محاسبه قدرمطلق عدد	print(abs(-12))	12
round()	گرد کردن عدد اعشاری	print(round(3.6)) print(round(3.1415, 2))	4 3.14
pow()	محاسبه توان	print(pow(2, 3))	8
int()	تبدیل به عدد صحیح	print(int(4.8))	4
float()	تبدیل به عدد اعشاری	print(float(5))	50
max()	بزرگ‌ترین عدد	Print(max(3, 9, 5))	9
min()	کوچک‌ترین عدد	Print(min(3, 9, 5))	3
sum()	جمع چند عدد	Print(sum([2, 4, 6]))	12

فعالیت

یک ماشین‌حساب با استفاده از توابع ریاضی ایجاد کنید. بخش‌هایی از کدهای مربوط به این برنامه نوشته شده است. با راهنمایی هنرآموز خود آنها را تکمیل کنید و برنامه را اجرا کنید.

```
print("Simple Calculator")
print("1- Addition")
print("2- Absolute Difference")
print("3- Power")
print("4- Division (Quotient & Remainder)")
print("5- Round a Number")
print("6- Max & Min")
print("7- (sum)")
print("8- Convert to Integer (int)")
print("9- Convert to Float (float)")

choice = input("Select an option (1-9): ")

if choice == "1":
    a = int(input("Enter first number: "))
    b = int(input("Enter second number: "))
    print("Result:", sum([a, b]))
```

```
elif choice == "3":
    a = int(input("Enter base: "))
    b = int(input("Enter exponent: "))
    print("Result:", pow(a, b))

elif choice == "4":
    a = int(input("Enter dividend: "))
    b = int(input("Enter divisor: "))
    q, r = divmod(a, b)
    print("Quotient:", q)
    print("Remainder:", r)
```

```
elif choice == "6":
    a = int(input("Enter first number: "))
    b = int(input("Enter second number: "))
    print("Max:", max(a, b))
    print("Min:", min(a, b))
```



```
elif choice == "8":
    num = float(input("Enter a number: "))
    print("Integer value =", int(num))
```



```
else:
    print("Invalid choice")
```

تا این بخش با توابعی مثل str, input , type, len, print آشنا شده‌اید. در ادامه با برخی از توابع داخلی غیرریاضی پایتون آشنا خواهید شد.

- **تابع sorted**: این تابع برای مرتب‌سازی رشته، مجموعه، توپل و دیکشنری مورد استفاده قرار می‌گیرد و نتیجه را در قالب یک لیست جدید برمی‌گرداند.

- **ساختار کلی تابع:**

sorted(iterable, reverse=false)

- **iterable**: داده‌ای که می‌خواهید مرتب کنید (لیست، رشته، توپل، مجموعه، دیکشنری و ...)
- **reverse**: اگر true باشد، ترتیب نزولی می‌شود. پیش‌فرض (false صعودی)

مثال: در این مثال تابع sorted رشته‌ای از کاراکترهای ذخیره‌شده در متغیر data را به‌صورت صعودی بر اساس کد اسکی کاراکترها مرتب می‌کند.

```
data = "hellu de823}Pl}"
print(sorted(data))
```

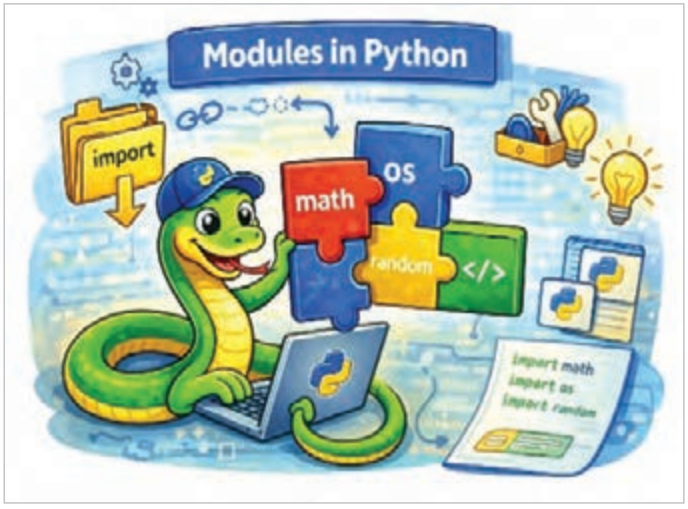
```
[' ', '2', '3', '8', 'P', 'd', 'e', 'e', 'h', 'l', 'l', 'l', 'u', '}', '}']
```

شکل ۱۴۹

فعالیت

نام شهرها را از کاربر دریافت و در یک لیست ذخیره کنید. سپس این لیست را به‌صورت نزولی مرتب کنید.

ماژول (module)



شکل ۱۵۰

ماژول یک فایل است که می‌تواند شامل چندین تابع، متغیر و کلاس باشد و برای سازمان‌دهی برنامه‌های بزرگ استفاده می‌شود. با استفاده از ماژول می‌توانید کدهای آماده را دوباره استفاده کنید و برنامهٔ خود را کوتاه‌تر و مرتب‌تر کنید.

انواع ماژول در پایتون:

ماژول استاندارد (standard)

ماژول‌هایی در پایتون وجود دارند که برنامه‌نویس فقط آنها را در برنامه خودش وارد می‌کند و نیازی به تعریف آنها نیست. مثل time، date، random، math و... . به دو روش می‌توان از ماژول‌های استاندارد استفاده کرد.

روش اول: وارد کردن کل ماژول (import mojvle):

این روش امن‌ترین راه است. چون از تداخل نام‌ها جلوگیری می‌کند. در این روش باید نام ماژول را قبل از تابع بیاورید.

مثال: Sqrt یک تابع آماده در ماژول math پایتون است، که برای محاسبه جذر اعداد استفاده می‌شود (شکل ۱۵۱).

```
import math
print(math.sqrt(16))
```

شکل ۱۵۱

مثال: لیستی از پیام‌های انگیزشی بسازید و با استفاده از ماژول random در هر اجرا یک پیام تصادفی چاپ کنید (شکل ۱۵۲).
برنامه را اجرا کنید و نتیجه را بررسی کنید.

```
import random

messages = ["Keep going", "You can do it", "Never give up"]
print(random.choice(messages))
```

شکل ۱۵۲

فعالیت

برنامه‌ای بنویسید که:
یک عدد تصادفی بین ۱ تا ۱۰ تولید کند. از کاربر بخواهد عدد را حدس بزند.اگر حدس درست بود، پیام شما موفق شدید چاپ کند؛ در غیر این صورت دوباره از کاربر ورودی بگیرد.ضمناً تعداد دفعات تلاش کاربر را هم چاپ کند.

فعالیت

برنامه‌ای بنویسید که شرایط زیر را داشته باشد:
- چند کلمه ازپیش‌تعیین‌شده در برنامه وجود داشته باشد.
- در هر بار اجرای برنامه، یکی از کلمات به‌صورت تصادفی انتخاب شود.
- تعداد دفعات مجاز برای حدس‌زدن، برابر با تعداد حروف کلمه انتخاب‌شده باشد.
- درصورتی‌که کاربر کلمه را به‌درستی حدس بزند: کلمه نمایش داده شود و پیام تبریک مناسب نمایش داده شود.
- در غیر این صورت: پیام پایان بازی نمایش داده شود.

روش دوم: وارد کردن بخش‌های خاص (from ... import):

from نام ماژول **import** نام تابع

در این روش، فقط توابع یا متغیرهای موردنیاز از یک ماژول وارد می‌شود. در این روش کد کوتاه‌تر و نوشتن ساده‌تر است.
مثال: فقط توابع sqrt و pow از ماژول math وارد برنامه می‌شوند (شکل ۱۵۳).

```
from math import sqrt, pow

print(sqrt(25))
print(pow(2, 3))
```

شکل ۱۵۳

بعد از این دستور؛ نیازی به نوشتن math.sqrt نیست. مستقیماً sqrt (25) نوشته می‌شود.

جدول۵- ماژول‌های پرکاربرد پایتون

ماژول	کاربرد
math	توابع ریاضی
random	تولید اعداد تصادفی
turtle	رسم شکل و گرافیک
datetime	تاریخ و زمان
os	کار با سیستم‌عامل

مثال: نمایش تاریخ و زمان فعلی: در این مثال فقط تابع now() مورد استفاده قرار گرفته است (شکل۱۵۴).

```
from datetime import datetime

now = datetime.now()
print("Current date and time:", now)
```

```
Current date and time: 2026-01-02 19:13:44.781451
```

شکل ۱۵۴

مثال: در این مثال، فقط نمایش ساعتِ زمان فعلی، نیاز است. بنابراین از روش دوم واردکردن ماژول استفاده شده است (شکل۱۵۵).

```
from datetime import datetime

now = datetime.now()
print("Hour:", now.hour)
```

شکل ۱۵۵

فعالیت

با استفاده از ماژول datetime برنامه‌ای بنویسید که **سال جاری** را نمایش دهد.

فعالیت

(بازی حدس کلمه)، ماژولی بنویسید که:

- یک کلمه مشخص داشته باشد. تعداد دفعات مجاز حدس‌زدن برابر با تعداد حروف کلمه باشد.
- اگر کاربر کلمه را درست حدس زد: کلمه چاپ شود و پیام تبریک نمایش داده شود.
- اگر کاربر در تعداد مجاز موفق نشد: پیام پایان بازی نمایش داده شود.
- این ماژول را در برنامه اصلی استفاده کنید.

ماژول محلی (local):

ماژول‌هایی هستندکه توسط برنامه‌نویس نوشته شده است و در برنامه import می‌شوند.

ابتدا یک فایل پایتون با پسوند py ایجاد کنید. سپس توابع و متغیرها را در آن قرار دهید. سپس می‌توانید آنها را با import در برنامه اصلی استفاده کنید.

مثال: فایل my_module.py را ایجاد کنید و در آن کدهای زیر را بنویسید (شکل۱۵۶).

my_module.py > ...

```
1   def greet(name):
2       return "Hello, " + name + "!"
```

شکل ۱۵۶

سپس در برنامه اصلی main.py این کدها را بنویسید (شکل۱۵۷).

main.py > ...

```
1   import my_module
2   message = my_module.greet("Ali")
3   print(message)
```

```
Hello, Ali!
```

شکل ۱۵۷

مثال: ابتدا یک فایل با نام math_module.py بسازید و توابع زیر را داخل آن بنویسید (شکل۱۵۸).

math_module.py > ...

```
1   def add(a, b):
2       return a + b
3
4   def multiply(a, b):
5       return a * b
```

شکل ۱۵۸

سپس در برنامه اصلی از آن استفاده کنید (شکل ۱۵۹).

```python
import math_module

print(math_module.add(5, 7))
print(math_module.multiply(3, 4))
```

```
12
12
```

شکل ۱۵۹

فعالیت

یک فایل با نام random_module.py بسازید و تابع زیر را داخل آن بنویسید:

```python
import random

def random_number():
    return random.randint(1, 100)
```

randint یک تابع آماده در ماژول **random** پایتون است که برای تولید عدد صحیح تصادفی در بازه‌ای خاص استفاده می‌شود.

سپس در برنامه اصلی ماژول ساخته شده را import نموده از آن استفاده کنید.

```python
import random_module
print("Random number:", random_module.random_number())
print("Random number:", random_module.random_number())
```

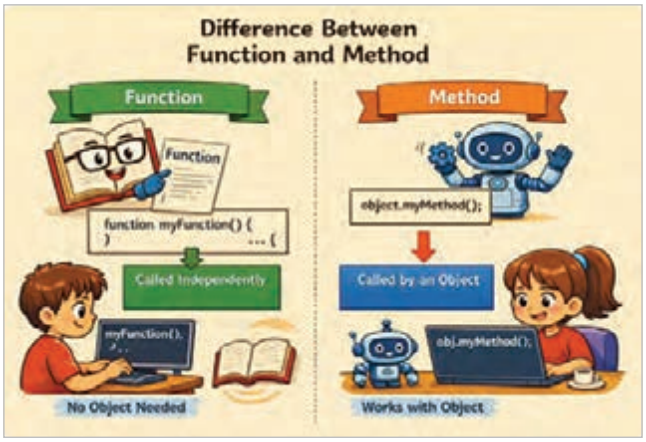
```
Random number: 22
Random number: 77
```

ـ برنامه را اجرا و خطوط کد را توضیح دهید.

■ **ماژول‌های شخص ثالث (3rd party)**: ماژول‌هایی که توسط دستور pip install نصب می‌شوند و در برنامه import می‌شوند. مثل Flask, PyGame، Django و...

تفاوت تابع و متد

در مطالب قبل، از دستوراتی مانند split()، append() و len() استفاده کرده‌اید، بدون آنکه تعریف رسمی آنها بیان شود. این موضوع کاملاً طبیعی است و در برنامه‌نویسی بسیار رایج است؛ زیرا بسیاری از مفاهیم ابتدا با استفاده عملی و سپس با تعریف دقیق معرفی می‌شوند.



شکل ۱۶۰

■ **تابع:(function)** دستوری مستقل است که برای انجام یک کار مشخص به کار می‌رود و به نوع داده خاصی وابسته نیست.

مثال: شمردن تعداد کاربران: تابع ()len تعداد کاراکترهای یک رشته را برمی‌گرداند (شکل ۱۶۱).

```
users = ["Arash", "Sara", "saba"]
print(len(users))
```

→ 3

شکل ۱۶۱

مثال: نمایش تعداد کاربران به‌صورت رشته‌ای (شکل ۱۶۲):

```
users = ["Nasim", "Sara", "Reza"]
print("User count: " + str(len(users)))
```

→ User count: 3

شکل ۱۶۲

■ **متد (Method):** نوعی تابع است که به یک کلاس یا شیء مشخص تعلق دارد و برای انجام عملیات روی داده‌های همان شیء استفاده می‌شود.

تفاوت اصلی تابع و متد در نحوهٔ **استفاده** و **وابستگی** آنهاست؛ تابع‌ها به‌صورت **مستقل فراخوانی** می‌شوند، اما متدها از طریق **یک شیء** یا **نوع داده** و **با استفاده از علامت نقطه (.)** اجرا می‌شوند و معمولاً **به داده‌های داخلی آن شیء دسترسی** دارند.

متد upper, lower: با استفاده از متد upper, lower می‌توانید همه متن را به حروف بزرگ یا کوچک انگلیسی تبدیل کنید.

فعالیت

با راهنمایی هنرآموز محترم خطوط کد مقابل را توضیح دهید.

```
your_text = input("enter you text")
print(your_text.upper())
print(your text.lower())
```

→

```
enter you text IT IS A TEST
 IT IS A TEST
 it is a test
```

متد split رشته را به لیستی از کلمات تجزیه می‌کند (شکل۱۶۳).

```
text = "Python Programming"
words = text.split()

numbers = [1, 2, 3]
numbers.append(4)

length = len(text)
```

شکل ۱۶۳

فعالیت

تابعی بنویسید که یک‌رشته را از ورودی دریافت کند و تعداد حروف بزرگ، کوچک و اعداد آن را نمایش دهد.

ارزشیابی شایستگی کار با توابع و ماژول‌ها

استاندارد عملکرد: هنرجو بتواند توابع و ماژول‌ها را ایجاد و استفاده کند، داده‌ها و عملیات تکراری را با توابع مدیریت کند، از ماژول‌های محلی بهره ببرد و تفاوت بین تابع و متد را تشخیص دهد.

ردیف	مرا حل کار	شرایط انجام کار (ابزار،مواد، تجهیزات، زمان، مکان و ...)	ابزارهای ارزشیابی	استاندارد (شاخص‌ها/داوری/نمره‌دهی)	نتایج ممکن	نمره کسب شده
۱	مجموعه‌ها (Set)	رایانه، vscode نصب‌شده، ۱۵ دقیقه	مشاهده عملی، چک لیست	ـ یک set با چند عنصر دلخواه بسازد ـ یک عنصر جدید به set اضافه کند. ـ یک عنصر از set حذف کند (هم عنصری که وجود دارد و هم عنصری که وجود ندارد تا خطای احتمالی را ببیند). ـ بررسی کند که آیا عنصر خاصی در set وجود دارد یا خیر. ـ یک dictionary خالی بسازد. ـ تفاوت اصلی set با list یا tuple را بداند (مثلا عدم وجود ترتیب و عناصر تکراری). کاربرد set در مواقعی که نیاز به حذف موارد تکراری داریم بداند ـ ویژگی کلید، مقدار (key ـ value) در dictionary را توضیح دهد ـ انجام کنجکاوی‌ها	۳	
				ـ یک set با چند عنصر دلخواه بسازد ـ یک عنصر جدید به set اضافه کند. ـ یک عنصر از set حذف کند (هم عنصری که وجود دارد و هم عنصری که وجود ندارد تا خطای احتمالی را ببیند). ـ بررسی کند که آیا عنصر خاصی در set وجود دارد یا خیر. ـ یک dictionary خالی بسازد. ـ تفاوت اصلی set با list یا tuple را بداند (مثلا عدم وجود ترتیب و عناصر تکراری).	۲	
				ـ آشنایی با تعاریف در حد توضیح مفاهیم ـ انجام ناقص مراحل	۱	
۲	ساختار داده Dictionary	رایانه، vscode نصب‌شده، فایل تمرینی، ۱۵دقیقه	مشاهده عملی، چک لیست	ـ یک dictionary با چند کلید- مقدار بسازد. ـ یک زوج کلید ـ مقدار جدید به dictionary اضافه ـ یک عنصر را بر اساس کلیدش از dictionary حذف کند. ـ سعی کند عنصری را با کلید ناموجود حذف کند تا با خطا مواجه شود و دلیل آن را توضیح دهد. ـ تفاوت بین کلید (key) و مقدار (value) در dictionary را بداند ـ اهمیت منحصربه‌فرد بودن کلیدها در dictionary را توضیح دهد. ـ روش‌های دسترسی به مقادیر در dictionary (با استفاده از کلید) را شرح دهد. انجام کنجکاوی‌ها	۳	
				ـ یک dictionary با چند کلید- مقدار بسازد. ـ یک زوج کلید ـ مقدار جدید به dictionary اضافه ـ یک عنصر را بر اساس کلیدش از dictionary حذف کند. ـ سعی کند عنصری را با کلید ناموجود حذف کند تا با خطا مواجه شود و دلیل آن را توضیح دهد.- تفاوت بین کلید (key و مقدار (value در dictionary را بداند	۲	
				ـ آشنایی با تعاریف و حد توضیح ـ انجام ناقص مراحل	۱	
۳	ساختار داده Tuple	رایانه، vscode نصب‌شده، ۲۰دقیقه	مشاهده عملی، پرسش شفاهی	یک tuple با چند عنصر (مثلاً مختصات یک نقطه یا اطلاعات یک شخص) بسازد ـ عناصر tuple با استفاده از اندیس دسترسی پیدا کند ـ تلاش کند یک عنصر را به tuple اضافه یا حذف کند و نتیجه (خطا) را مشاهده و دلیل آن را بیان کند ـ (اختیاری) تلاش کند یک tuple را حذف کند. ـ اصلی‌ترین ویژگی tuple که آن را از list متمایز می‌کند، بداند؟ اشاره به تغییرناپذیری یا Immutability در چه مواردی استفاده از tuple به list ارجحیت دارد؟ (مثلاً در مواردی که داده نباید تغییر کند، یا به عنوان کلید در dictionary) ـ انجام کنجکاوی‌ها	۳	
				ـ یک tuple با چند عنصر (مثلاً مختصات یک نقطه یا اطلاعات یک شخص) بسازد ـ عناصر tuple با استفاده از اندیس دسترسی پیدا کند ـ تلاش کند یک عنصر را به tuple اضافه یا حذف کند و نتیجه (خطا) را مشاهده و تلاش کند یک tuple را حذف کند. ـ اصلی‌ترین ویژگی tuple که آن را از list متمایز می‌کند، را بداند.	۲	
				ـ آشنایی با تعاریف در حد توضیح ـ انجام ناقص مراحل	۱	

۴	توابع (Functions) ـ مبانی و پارامترها	رایانه، vscode نصب‌شده، ۲۰ دقیقه	مشاهده عملی، تحویل فایل	یک تابع ساده بنویسد که فقط یک متن خوشامدگویی را چاپ کند. ـ تابعی بنویسد که دو عدد را به عنوان ورودی (پارامتر) بگیرد و مجموع آنها را چاپ کند. ـ این تابع را با مقادیر مختلف (آرگومان) فراخوانی کند. ـ بداند چرا از توابع در برنامه‌نویسی استفاده می‌شود ؟ ـ تفاوت بین «پارامتر» (تعریف تابع) و «آرگومان» (مقدار ارسالی هنگام فراخوانی) را توضیح دهد ـ انجام کنجکاوی‌ها	۳	
				یک تابع ساده بنویسد که فقط یک متن خوشامدگویی را چاپ کند.ـ تابعی بنویسد که دو عدد را به عنوان ورودی (پارامتر) بگیرد و مجموع آنها را چاپ کند.ـ این تابع را با مقادیر مختلف (آرگومان) فراخوانی کند. ـ بداند چرا از توابع در برنامه‌نویسی استفاده می شود؟	۲	
				آشنایی با تعاریف در حد توضیح ـ انجام ناقص مراحل	۱	
۵	توابع (Functions) ـ مقدار بازگشتی و کاربردها	رایانه، vscode نصب‌شده، ۱۰ دقیقه	مشاهده عملی، پرسش شفاهی	تابعی بنویسد که دو عدد را به عنوان ورودی بگیرد، حاصل ضرب آنها را محاسبه کرده و بازگرداند نتیجه بازگشتی تابع را در یک متغیر ذخیره کرده و سپس آن متغیر را چاپ کند ـ تابعی بنویسد که یک لیست از اعداد را به عنوان ورودی بگیرد و با استفاده از حلقه، مجموع عناصر لیست را محاسبه و بازگرداند ـ از تابع sort-ed برای مرتب‌سازی یک لیست استفاده کند و نتیجه را ببینند.ـ تفاوت بین تابعی که مقداری را print می‌کند و تابعی که مقداری را return می‌کند، توضیح دهد.ـ نقش دستور return در توابع توضیح دهد ـ انجام کنجکاوی‌ها	۳	
				تابعی بنویسد که دو عدد را به عنوان ورودی بگیرد، حاصل ضرب آن‌ها را محاسبه کرده و بازگرداند ـ نتیجه بازگشتی تابع را در یک متغیر ذخیره کرده و سپس آن متغیر را چاپ کند ـ تابعی بنویسد که یک لیست از اعداد را به عنوان ورودی بگیرد و با استفاده از حلقه، مجموع عناصر لیست را محاسبه و بازگرداند ـ از تابع sorted برای مرتب‌سازی یک لیست استفاده کند و نتیجه را ببینند	۲	
				آشنایی با تعاریف در حد توضیح ـ انجام ناقص مراحل	۱	
۶	ماژول‌ها (Modules) و تفاوت‌ها	رایانه، vscode نصب‌شده، ۲۰ دقیقه	مشاهده عملی، پرسش شفاهی	یک ماژول ساده (مثلاً یک فایل پایتون با چند تابع) در همان پوشه ایجاد کند ـ فایل اصلی را اجرا کند و تابع تعریف شده در ماژول محلی را با استفاده از import فراخوانی کند.ـ از یک ماژول استاندارد پایتون مانند math استفاده کند (مثلا import math و سپس math.sqrt() یا math.pi). (اختیاری) ـ نام یک ماژول شخص ثالث که قبلاً نصب شده را ذکر کند ـ هدف از استفاده از ماژول‌ها را توضیح دهد ـ تفاوت اصلی بین «تابع» و «متد» (method) را توضیح دهد ـ انجام کنجکاوی‌ها	۳	
				۱ـ یک ماژول ساده (مثلاً یک فایل پایتون با چند تابع) در همان پوشه ایجاد کند. ـ فایل اصلی را اجرا کند و تابع تعریف شده در ماژول محلی را با استفاده از import فراخوانی کند ـ از یک ماژول استاندارد پایتون مانند math استفاده کند.	۲	
				آشنایی با تعاریف در حد توضیح ـ انجام ناقص مراحل	۱	
شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش:				احراز همه شایستگی‌های غیرفنی	۲	
				عدم احراز هریک از شایستگی‌های غیرفنی	۱	
ارزشیابی کار (شایستگی انجام کار)					بلی □	
					خیر □	

توضیحات:

۱ـ معیار شایستگی انجام کار:

ـ کسب حداقل نمره ۲ از مراحل ۱، ۲، ۳، ۴، ۵ و ۶ (مراحل بحرانی)

ـ کسب حداقل نمره ۲ از بخش شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش

ـ کسب حداقل میانگین ۲ از مراحل کار

۲ـ تعریف سطوح شایستگی: صرف‌نظر از اینکه یک تکلیف کاری در چه سطحی از صلاحیت حرفه‌ای انجام می شود، در هر محیط کاری ممکن است انجام هر کار با کیفیت مشخصی مورد انتظار باشد. سطح شایستگی انجام کار، معیار اساسی ارزشیابی است.

مهارت (سطح سه): ماهر و قادر به آموزش و هدایت دیگران، توانایی برنامه‌ریزی و تحلیل، پاسخ‌گویی در برابر کارهای خود، سروکار داشتن با سطح وسیعی از کارها و فعالیت‌ها

تسلط (سطح چهار): خبرگی در انجام کار و آموزش دیگران، ایجاد نوآوری، سازگاری، عیب‌یابی، هدایت و راهنمایی دیگران.

پودمان دوم

طراحی صفحات وب ایستا



امروزه بخش عمده‌ای از خدمات، آموزش، تجارت، ارتباطات و سرگرمی‌ها از طریق فضای وب قابل دسترسی است. اگرچه وب در ابتدا صرفاً ابزاری برای نمایش متون و تصاویر ساده بود، اما با پیشرفت تکنولوژی به بستری تعاملی، هوشمند و کاربردی تبدیل شده است. این تحول، اهمیت طراحی بصری و تجربه کاربری را دوچندان کرده است، چرا که کاربران امروز به دنبال دریافت تجربه‌ای زیبا، ساده و کارآمد هستند. در همین راستا، طراحی رابط کاربری (UI) به عنوان یکی از ارکان کلیدی در توسعه وب‌سایت‌های استاندارد و جذاب شناخته می‌شود. تکامل وب را می‌توان در سه نسل اصلی دسته‌بندی کرد:

وب ۱.۰: حالت فقط خواندنی (Read-Only) و فاقد تعامل

وب ۲.۰: قابلیت تعامل کاربران و تولید محتوای مشارکتی

وب ۳.۰: هوشمند، شخصی‌سازی شده و مبتنی بر داده‌های معنایی

در ادامه این مسیر تکاملی، درک عمیق از اصول طراحی رابط کاربری (UI) و تجربه کاربری (UX) بدون تسلط بر زیرساخت‌های فنی وب امکان‌پذیر نیست؛ بنابراین در این پودمان، به بررسی دو ستون بنیادین توسعه وب یعنی HTML و CSS پرداخته می‌شود. در اینجا، نحوه پیاده‌سازی معنای محتوا با تگ‌های استاندارد HTML و کنترل گرافیکی و چیدمان آنها با CSS، به صورت فنی و ساختاریافته مورد تحلیل قرار می‌گیرد تا با ترکیب دقیق این دو تکنولوژی، ساختاری پایدار، واکنش‌گرا و ظاهری متناسب با استانداردهای مدرن وب قابل ارائه باشد.

واحد یادگیری ۱
طراحی ساختار صفحات وب با html

آیا تابه‌حال اندیشیده‌اید

- فرایند طراحی یک وب‌سایت دارای چه مراحلی است؟
- طرح اولیه وب‌سایت به چه شکلی ارائه می‌شود؟
- چگونه می‌توان عناصر مختلفی مانند متن، تصویر، ویدئو، منو و ... را در صفحه وب نمایش داد؟
- پیوند بین صفحات چگونه ایجاد می‌شود؟
- فرم‌های ورود اطلاعات چگونه طراحی می‌شوند؟
- CSS چه نقشی در طراحی و زیباسازی صفحات وب ایفا می‌کند؟
- طراحی واکنش‌گرای صفحه وب نسبت به پهنای دستگاه به چه شکلی انجام می‌شود؟

در پایان از هنرجو انتظار می‌رود

۱ صفحه اصلی سایت (index.html) را با ساختاری استاندارد ایجاد نماید.

۲ تگ‌های سطح بالای صفحه را به‌صورت صحیح استفاده نماید.

۳ منوی ناوبری صفحه را با استفاده از تگ <nav> و عنصر لیست، درون عنصر معنایی <header> ایجاد نماید.

۴ محتوای اصلی صفحه را درون عنصر معنایی <main> ایجاد نماید و توانایی به‌کارگیری تگ <section> را برای ایجاد بخش‌هایی از صفحه داشته باشد.

۵ برای نگهداری محتوای بخش‌هایی از صفحه از عنصر استفاده نماید.

۶ تگ کامنت (<-- --!>) را برای مستندسازی برخی کدها استفاده نماید.

استاندارد عملکرد

توانایی ایجاد یک ساختار استاندارد برای صفحه وب با به‌کارگیری صحیح تگ‌های HTML5، به‌گونه‌ای که ساختار و محتوای صفحه برای موتورهای جست‌وجو و ابزارهای صفحه‌خوان (Screen reader) قابل درک باشد، همچنین تنظیمات زبان صفحه و جهت نمایش محتوا در مرورگر به‌درستی انجام شود.

نحوه کار وب

وب (World Wide Web) در سال ۱۹۸۹ توسط تیم برنرزلی در سازمان پژوهش‌های هسته‌ای اروپا (CERN) ایجاد شد. هدف اولیه آن تسهیل به اشتراک‌گذاری اطلاعات بین محققان بود. اولین وب‌سایت در سال ۱۹۹۱ راه‌اندازی شد و از آن زمان به سرعت گسترش یافت و به بخشی اساسی از زندگی روزمره انسان‌ها تبدیل شد.

وب به‌وسیله پروتکل‌های خاصی مانند HTTP و HTTPS کار می‌کند. این پروتکل‌ها روش‌هایی را برای ارسال و دریافت اطلاعات بین مرورگرها و سرورها مشخص می‌کنند.

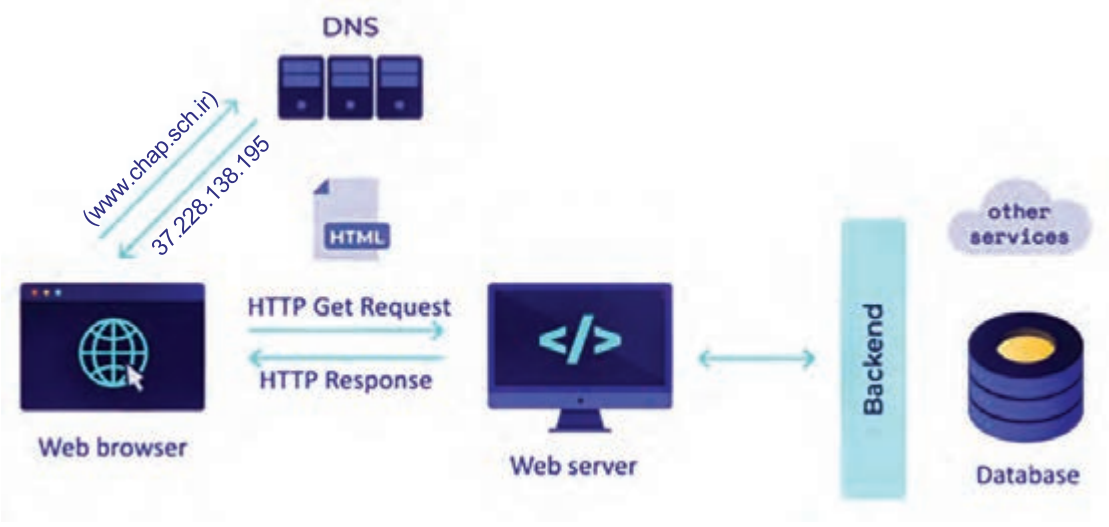
زمانی که کاربر یک آدرس وب را در مرورگر خود وارد می‌کند، مراحل زیر رخ می‌دهد:

۱ـ ترجمه نام دامنه: مرورگر به یک سیستم DNS[۱] متصل می‌شود تا نام دامنه (مثلاً www.chap.sch.ir) را به یک آدرس IP تبدیل کند. آدرس IP دریافتی متعلق به وب‌سرور[۲] مورد نظر خواهد بود.

۲ـ ارسال درخواست (HTTP Request): پس از دریافت آدرس IP، مرورگر یک درخواست HTTP به وب‌سرور مقصد ارسال می‌کند که شامل اطلاعاتی درباره نوع محتوا و درخواست کاربر است.

۳ـ پاسخ‌دهی سرور (HTTP Response): سرور درخواست را پردازش می‌کند، در صورت لزوم اطلاعاتی را از پایگاه داده (Database) دریافت می‌کند و سپس محتوا را به صورت یک پاسخ HTTP به مرورگر کاربر ارسال می‌کند. این محتوا می‌تواند شامل فایل‌های HTML، CSS، JavaScript و تصاویر باشد.

۴ـ نمایش محتوا: مرورگر محتوای دریافت شده را تفسیر می‌کند و آن را به‌صورت یک صفحه وب برای کاربر نمایش می‌دهد (شکل ۱).



شکل ۱

۱ـ DNS وظیفه تبدیل نام دامنه وب‌سایت به آدرس عددی IP را دارد. مرورگرها برای ارسال درخواست کاربر به وب‌سرور مقصد نیازمند آدرس IP آن سرور هستند. از آنجایی که به خاطر سپردن آدرس IP وب سایت‌ها برای کاربران دشوار است، بنابراین کاربران در هنگام ارتباط با مرورگر، از نام دامنه استفاده می‌کنند (مانند chap.sch.ir)، اما مرورگر آدرس دامنه را برای DNS ارسال می‌کند تا آدرس IP مربوط به وب سرور مقصد را به‌دست آورد.

۲ـ وب‌سرور یک سیستم سخت‌افزاری است که میزبان اسناد وب سایت (مانند فایل‌های HTML، تصاویر، ویدئوها، پایگاه داده‌ها و ...) است، این سخت افزار با استفاده از نرم‌افزارهای مشخصی وظیفه نگهداری و مدیریت اطلاعات وب‌سایت را به عهده دارد. وب‌سرورها درخواست کاربران را با واسطه از طریق مرورگر دریافت می‌کنند و سپس صفحات پاسخ را برای مرورگر کاربر ارسال می‌کنند.

برخی از مفاهیم اصلی طراحی وب

وب دارای مفاهیم کلیدی مختلفی است که آگاهی نسبت به آنها برای هر توسعه‌دهندهٔ وب (Web developer) ضروری است. در ادامه، به شرح این مفاهیم پرداخته می‌شود:

■ **طراح وب (Web designer)**: فرد یا تیمی که مسئول طراحی ظاهر نمایشی وب‌سایت است. این حرفه ترکیبی از طراحی گرافیکی، طراحی تجربه کاربری (UX) و طراحی رابط کاربری (UI)، ایجاد wire frame و نمونه اولیه است.

■ **تجربه کاربری (UX)**: تجربه‌ای که کاربر بعد از بازدید از وب‌سایت برای دسترسی به اطلاعات، محصولات و خدمات کسب می‌کند.

■ **رابط کاربری (UI)**: عناصر تعاملی صفحه مانند دکمه‌ها، فرم‌ها و منوها که کاربران به‌وسیله آنها با وب‌سایت تعامل دارند.

■ **Front-end**: بخشی از وب‌سایت که ارتباط مستقیمی با کاربر دارد. مباحث بخش فرانت‌اند شامل ظاهر صفحه وب، رابط کاربری (UI) و تجربه کاربری (UX) می‌شود.

■ **Back-end**: بخش پشت‌صحنهٔ وب‌سایت یا اپلیکیشن که شامل منطق اصلی برنامه، پردازش داده‌ها و ارتباط با پایگاه داده است و کاربر تعامل مستقیمی با آن ندارد.

■ **توسعه‌دهندهٔ وب (Web developer)**: فرد یا تیمی که طرح اولیهٔ وب‌سایت را با استفاده از زبان‌های برنامه‌نویسی، به صفحات وب قابل استفاده تبدیل می‌کند. توسعه‌دهندهٔ وب می‌تواند نقش طراح وب را نیز داشته باشد. وظایف اصلی توسعه دهندهٔ وب شامل موارد زیرمی‌باشد:

۱ـ توسعه‌دهندهٔ سمت کاربر (Front-end-developer): روی ظاهر سایت و عناصری تمرکز دارد که کاربر مستقیماً با آنها سروکار دارد.

۲ـ توسعه‌دهندهٔ سمت سرور (Back-end-developer): روی مباحث سمت سرور مانند مدیریت پایگاه داده و پردازش داده‌های کاربران تمرکز دارد.

■ **دامنه (Domain)**: نامی منحصربه‌فرد که برای دسترسی به یک وب‌سایت استفاده می‌شود، مانند google.com. کاربران با استفاده از نام دامنه وارد یک سایت می‌شوند. نام دامنه، جایگزینی قابل فهم برای آدرس‌های عددی IP است که به‌خاطر سپردن آدرس سایت را برای کاربران سهولت می‌بخشد.

■ **URL**: آدرس منحصربه‌فردی که برای پیدا کردن و شناسایی یک منبع خاص (مثل صفحه وب، عکس، ویدئو یا فایل) در اینترنت استفاده می‌شود. URL مخفف Uniform Resource Locator است (شکل ۲).

Host: فضایی برای ذخیره‌سازی اسناد وب‌سایت است، به‌طوری‌که امکان دسترسی به منابع سایت از طریق اینترنت فراهم گردد. تمام فایل‌های یک سایت (شامل فایل های HTML، CSS، اسکریپت‌ها، تصاویر و ...) باید روی یک رایانه وب‌سرور ذخیره گردند تا کاربران بتوانند با وارد کردن آدرس سایت به محتوای آن دسترسی پیدا کنند.



شکل ۲

Localhost (هاست محلی): یک سیستم رایانه‌ای مجهز به نرم‌افزار وب‌سرور است که برای اجرای کدهای سمت سرور (Back-End)، ایجاد و مدیریت پایگاه داده‌ها استفاده می‌شود. هاست محلی صرفاً برای تمرین، آزمایش و رفع اشکال صفحات وب‌سایت کاربرد دارد. نرم‌افزارهای وب‌سرور مختلفی وجود دارند که می‌توانند یک سیستم خانگی را به وب‌سرور تبدیل کنند، اما نکته مهم این است که تمام وب‌سرورهایی که در سطح اینترنت پاسخگوی کاربران هستند، دارای IP اینترنتی هستند؛ بنابراین سیستم‌های خانگی و کلاس درس که فاقد IP اینترنت هستند، امکان پاسخ‌دهی به کاربران راه دور را ندارند.

نکته

پس از طراحی و آزمایش صفحات سایت روی لوکال هاست، می‌توانید محتوای سایت خود را به یک وب هاست یا وب سرور انتقال دهید. دسترسی به یک وب سرور نیازمند پرداخت هزینه است، اما برخی شرکت‌ها امکان استفاده از هاست‌های رایگان را با امکانات و منابع محدود فراهم می‌کنند که برای یک پروژه واقعی مناسب نیستند. مناسب‌ترین‌هاست برای سایت‌هایی که منابع سخت افزاری قدرتمندی نیاز ندارند هاست‌های اشتراکی هستند.

هاست اشتراکی:

شرکت‌های هاستینگ طرح‌های مختلفی را برای ارائه خدمات خود دارند که هر طرح (Plan) دارای امکانات سخت‌افزاری و پهنای باند متفاوتی است. از جمله این طرح‌ها می‌توان به هاست اشتراکی، سرور مجازی، سرور اختصاصی و سرور ابری اشاره نمود. در طرح هاست اشتراکی امکانات سخت‌افزاری هاست مانند فضای دیسک، پهنای باند، CPU و حافظه RAM بین چندین وب سایت تقسیم می‌شود. به این ترتیب تمام توان سخت‌افزاری وب‌سرور صرف پاسخ‌دهی به درخواست‌های کاربران یک سایت نمی‌شود. به همین دلیل چنین هاست‌هایی برای سایت‌های کوچک با ترافیک محدود مناسب هستند.

■ **طراحی واکنش‌گرا (Responsive Design)**: یک روش طراحی صفحه که امکان تطبیق‌پذیری خودکار صفحه را بر اساس اندازه صفحه نمایش (دستگاه‌های مختلف مانند موبایل، تبلت و دسکتاپ) فراهم می‌کند.

■ **سئو (SEO)**: مجموعه‌ای از تکنیک‌ها برای بهبود رتبه وب‌سایت در نتایج موتورهای جست‌وجو است. سئو شامل بهینه‌سازی محتوا، ساختار URL و استفاده از متاتگ‌ها و ... است.

■ **دسترس‌پذیری (Accessibility)**: طراحی وب‌سایت به گونه‌ای که برای همه کاربران (از جمله افراد دارای ناتوانی‌های جسمی یا بینایی) قابل دسترسی باشد.

پایگاه‌داده: بسیاری از وب‌سایت‌ها برای ذخیره و مدیریت اطلاعات از پایگاه‌داده‌ها (Database) استفاده می‌کنند. پایگاه‌داده‌ها می‌تواند شامل اطلاعات کاربران، محصولات، مقالات و محتواهای دیگر باشند.

■ **کنترل نسخه:** سیستم‌های کنترل نسخه امکان مدیریت تغییرات کد را فراهم می‌کنند، به‌گونه‌ای که بازگشت به نسخه‌های قبلی به‌راحتی امکان‌پذیر باشد.

کنجکاوی

۱ در مورد پروتکل SSL تحقیق کنید و نتیجه آن را در کلاس ارائه دهید.
۲ درباره انواع وب‌سایت و کاربرد آنها تحقیق کنید و نتیجه آن را در کلاس ارائه دهید.

■ فرایند طراحی سایت

طراحی و پیاده‌سازی یک وب‌سایت، فرایندی مرحله‌به‌مرحله است که هر مرحله نقش مهمی در موفقیت نهایی پروژه دارد. اگر این مراحل به‌درستی و با ترتیب مناسب انجام شوند، نتیجهٔ کار سایتی کاربردی، زیبا و قابل گسترش خواهد بود.

■ مراحل کلی طراحی سایت

فرایند طراحی سایت به‌صورت خلاصه شامل مراحل زیر است:

- تعریف هدف و نیازمندی‌ها
- طراحی نقشه سایت (site map)
- ایجاد وایر فریم (wire frame)
- ایجاد ساختار صفحات با HTML
- طراحی ظاهر صفحات با CSS
- افزودن تعاملات کاربر با JavaScript
- طراحی پایگاه داده (مانند جدول کاربران، محصولات و سفارش‌ها)
- ایجاد ارتباط بین صفحات و پایگاه داده
- افزودن قابلیت ورود و ثبت‌نام کاربران
- پیاده‌سازی سبد خرید و پرداخت
- تست عملکرد سایت
- رفع خطاها و بهینه‌سازی
- انتشار سایت روی سرور

■ **تعریف هدف و نیازمندی‌ها:** پیش از شروع طراحی یا نوشتن کد صفحات سایت، باید مشخص شود که وب‌سایت به چه منظور و برای چه گروهی از کاربران طراحی می‌شود. شروع طراحی بدون شناخت نیازها و تعیین مأموریت وب‌سایت، معمولاً باعث بروز مشکلات اساسی در حین پیاده‌سازی پروژه و نیاز به طراحی مجدد می‌شود. به همین دلیل، اولین و مهم‌ترین گام در طراحی هر وب‌سایت، تعریف هدف و نیازمندی‌های آن است.

■ **تعیین هدف پروژه:** در این مرحله باید به پرسش‌های زیر پاسخ داده شود:

- هدف از ایجاد سایت چیست؟
- چه نوع کالا یا خدماتی در سایت ارائه می‌شود؟
- کاربران اصلی سایت چه کسانی هستند؟
- چه امکاناتی باید در سایت وجود داشته باشد؟

مثال

یک تیم طراح و توسعه‌دهندهٔ وب قصد دارد وب سایتی را برای فروش کتاب‌های درسی ایجاد نماید. اغلب کاربران این سایت هنرآموزان و هنرجویان هستند. امکانات مورد نیاز شامل ثبت نام کاربران، نمایش کالاها، سبد خرید و پرداخت آنلاین است.

■ **تحلیل نیازمندی‌ها:** در این مرحله، نیازهای سایت از دو جنبه بررسی می‌شود:

الف) نیازمندی‌های کاربر (User Requirements): این نیازمندی‌ها شامل ویژگی‌ها و امکاناتی است که کاربران در هنگام استفاده از سایت به آنها نیاز دارند (جدول ۱).

جدول ۱

نیاز کاربر	توضیح
ثبت‌نام و ورود	کاربر بتواند حساب کاربری ایجاد کرده و وارد سایت شود.
جست‌وجوی محصول	امکان جست‌وجو براساس نام یا دسته‌بندی کالا وجود داشته باشد.
سبد خرید	کاربر بتواند محصولات را به سبد خرید اضافه کند.
پرداخت اینترنتی	فرایند خرید از طریق درگاه پرداخت انجام شود.
مشاهده سفارش‌ها	کاربر بتواند سفارش‌های قبلی خود را مشاهده کند.

ب) نیازمندی‌های فنی (Technical Requirements): این نیازمندی‌ها شامل زبان‌های نشانه‌گذاری و برنامه‌نویسی، سیستم مدیریت پایگاه‌ها و توجه به امنیت و گسترش‌پذیری وب‌سایت است (جدول ۲).

جدول ۲

مورد	توضیح
زبان برنامه‌نویسی	Python (Django)
پایگاه داده	SQLite یا MySQL
طراحی ظاهر	HTML، CSS، Bootstrap
امنیت	رمزنگاری رمز عبور کاربران
قابلیت گسترش	امکان افزودن محصولات و امکانات جدید در آینده

■ **طراحی نقشهٔ سایت (Site Map):** پس از مشخص شدن هدف و نیازهای وب‌سایت، طراحی نقشهٔ سایت امری ضروری است. نقشهٔ سایت ساختار صفحات، ارتباطات میان آنها و مسیر حرکت کاربر را درون سایت مشخص می‌کند. طراحی صحیح نقشهٔ سایت موجب درک بهتر ساختار سایت و ساده شدن فرایند طراحی و پیاده‌سازی آن می‌گردد.

کار کارگاهی

مراحل طراحی نقشهٔ سایت

مرحلهٔ ۱: مشاهده و بررسی سایت‌های فروشگاهی

با راهنمایی هنرآموز خود، دو یا سه سایت فروشگاهی ساده را بررسی کنید.

به‌عنوان نمونه: فروشگاه موبایل، فروشگاه لباس، فروشگاه لوازم دیجیتال

مواردی که باید بررسی شود: چه صفحاتی در منوی سایت وجود دارد؟ کاربر از صفحهٔ اصلی به کدام صفحات هدایت می‌شود؟ فرایند ورود کاربر، انتخاب کالا و استفاده از سبد خرید چگونه انجام می‌شود؟ ظاهر سایت ساده است یا شلوغ؟

مرحلهٔ ۲: شناسایی صفحات اصلی سایت

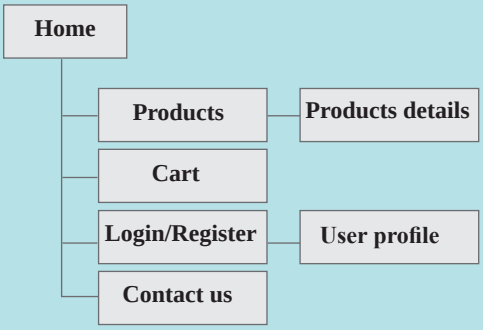
در این مرحله، لازم است صفحات مشترک میان سایت‌ها بررسی‌شده را شناسایی و یادداشت نمایید.

نمونه‌ای از صفحات رایج در یک فروشگاه اینترنتی عبارت‌اند از:

- Home
- Products
- Product Details
- Cart
- Login / Register
- User Profile
- About / Contact

مرحله ۳: ترسیم نقشهٔ سایت (Site Map)

در این مرحله نقشه سایت را به‌صورت نوشتاری یا گرافیکی طراحی کنید. یک روش ساده برای ترسیم نقشه این است که صفحهٔ اصلی در بالای نقشه و سایر صفحات به‌صورت زیر شاخه در زیر صفحهٔ اصلی قرار گیرند.



شکل روبه‌رو یک نقشهٔ نوشتاری محسوب می‌شود:
علاوه بر روش نوشتاری، نقشه سایت می‌تواند به‌صورت ترسیمی با استفاده از کادرها و فلش‌ها نمایش داده شود.

مرحله ۴: بیان مسیر حرکت کاربر

در این مرحله، مسیر حرکت کاربر در سایت توضیح داده می‌شود. برای توصیف مسیر به موارد زیر توجه کنید:

کاربر از چه صفحه‌ای وارد سایت می‌شود؟ چگونه محصول موردنظر را انتخاب می‌کند؟ از چه مسیری به سبد خرید می‌رسد؟

در چه مرحله‌ای وارد حساب کاربری می‌شود؟

مثال زیر یک نمونه از توصیف مسیر حرکت کاربر است:

کاربر ابتدا وارد صفحهٔ اصلی سایت می‌شود، سپس به صفحهٔ محصولات مراجعه کرده و یک محصول را انتخاب می‌کند. پس از مشاهدهٔ جزئیات محصول، آن را به سبد خرید اضافه کرده و در نهایت برای ثبت سفارش وارد حساب کاربری می‌شود.

طراحی ظاهر صفحات

پس از مشخص شدن هدف و نیازمندی‌ها، نوبت آن است که **ظاهر صفحات وب سایت** طراحی شود. هرچقدر ظاهر سایت زیباتر و منظم‌تر باشد، کاربران بیشتری جذب سایت خواهند شد. در یک سایت فروشگاهی، زیبایی سایت می‌تواند منجر به جذب مشتری و فروش بیشتر شود.

برای ایجاد عناصر درون صفحات وب از فرازبان HTML و برای سبک‌دهی به عناصر صفحه از زبان توصیفی CSS استفاده می‌شود (جدول ۳).

جدول ۳

ابزارهای طراحی ظاهر سایت (فرانت‌اند)	
کاربرد	**زبان**
ساختار و عناصر صفحه (مثل متن، تصویر، جدول، دکمه)	HTML
زیباسازی وکنترل ظاهر عناصر شامل تنظیم رنگ‌ها، فونت‌ها ، اندازه‌ها، چیدمان و ...	CSS

■ **طراحی سایت به‌صورت تیمی:** فرایند طراحی سایت شامل مراحل متعددی است که هر کدام نیازمند مهارت‌های تخصصی هستند. به همین دلیل، طراحی سایت معمولاً به‌صورت تیمی انجام می‌شود و هر عضو مسئولیت مشخصی دارد.

نقش‌های اصلی در تیم طراحی سایت

۱ـ طراح تجربه کاربری (UX Designer): تحلیل نیاز کاربران و بهینه‌سازی تجربه استفاده از سایت

۲ـ طراح رابط کاربری (UI Designer): طراحی عناصر بصری مانند رنگ‌ها، فونت‌ها، دکمه‌ها و فرم‌ها

۳ـ توسعه‌دهندهٔ فرانت‌اند (Front-End Developer): پیاده‌سازی بخش ظاهری و تعاملی سایت

۴ـ توسعه‌دهندهٔ بک‌اند (Back-End Developer): برنامه‌نویسی سمت سرور، مدیریت پایگاه داده و امنیت

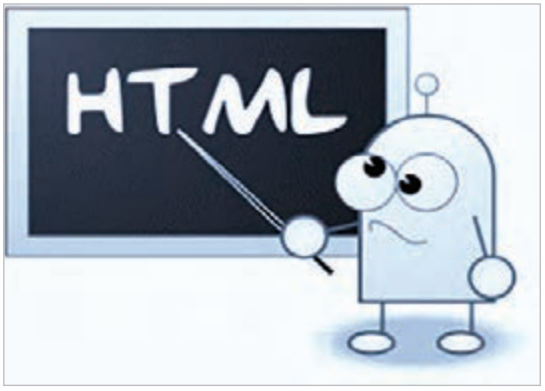
۵ـ متخصص محتوا (Content Specialist): تحقیق کلمات کلیدی و تولید محتوای متنی جذاب

۶ـ متخصص سئو (SEO Specialist): بهینه‌سازی سایت برای موتورهای جست‌وجو

بسته به حجم کار پروژه، بودجه و تعداد نفرات تیم طراح و توسعه‌دهنده وب‌سایت، هر نقش می‌تواند به یک یا چند نفر اختصاص داده شود. در پروژه‌های کوچک ممکن است یک فرد چندین نقش را بر عهده داشته باشد. در جدول ۴ ابزارهای متداول مورد استفاده توسط هر یک از نقش‌های تیم طراحی سایت معرفی شده است. این ابزارها به بهبود کیفیت پروژه و تسهیل همکاری بین اعضای تیم کمک می‌کنند.

جدول ۴ـ ابزارهای متداول در طراحی سایت

نقش	ابزارها
طراح UX و UI	Figma، Adobe XD، Sketch، InVision، Photoshop
توسعه‌دهندهٔ فرانت‌اند	HTML، CSS، JavaScript، Bootstrap
توسعه‌دهندهٔ بک‌اند	JavaScript (Node.js)، (Flask، Python، Django)
ویرایشگر کد	Visual Studio Code، Sublime Text، Atom
مرورگر و ابزار توسعه	Google Chrome، Mozilla Firefox، Microsoft Edge
پلتفرم‌های مدیریت محتوا و تولید محتوای تحت وب	WordPress، Google Docs، Grammarly
متخصص سئو	Google Analytics، SEMrush، Moz، Yoast SEO
سایر ابزار مورد استفاده تیم طراحی	FileZilla، Cyberduck
کنترل نسخه	Git، GitHub



شکل ۳

طراحی ساختار صفحه با HTML

در یک وب‌سایت، صفحات وب متعددی مانند موارد زیر وجود دارد:

- صفحهٔ اصلی (index.html)
- صفحهٔ محصولات (products.html)
- صفحهٔ سبد خرید (cart.html)
- صفحهٔ ورود/ثبت‌نام (login.html)

گام اول در طراحی این صفحات، ایجاد ساختار اولیه صفحات و قرار دادن عناصر صفحه درون آنهاست. این بخش به معرفی نحوه درج عناصر درون صفحات وب می‌پردازد.

نشانه‌گذاری عناصر در سند HTML: HTML مخفف Hypertext Markup Language به معنای زبان نشانه‌گذاری ابرمتن است. HTML یک زبان برنامه‌نویسی نیست، بلکه یک زبان نشانه‌گذاری برای ایجاد و ساختاردهی عناصر مختلف در صفحات وب به شمار می‌آید. جانمایی بخش‌های مختلف صفحه مانند سرصفحه، منو، ناحیه محتوای اصلی، ستون کناری و پاصفحه توسط HTML انجام می‌شود. علاوه‌براین عناصر مختلفی همچون متن، تصویر، ویدئو، صوت و ... توسط کدهای HTML درون بخش‌های مختلف صفحه درج می‌شود.



شکل ۴

معرفی تگ (Tag): یک صفحهٔ وب شامل عناصر مختلفی مانند تیترها، پاراگراف‌ها، پیوندها، تصاویر، لیست و غیره است. هر یک از عناصر موجود در صفحات وب توسط تگ‌ها یا تگ‌های تعریف‌شده در زبان نشانه‌گذاری HTML ایجاد می‌شود. تگ‌های HTML در موضوعات مختلف قابل استفاده هستند.



شکل ۵

شکل کلی یک تگ به صورت زیر است:

<TagName Attribute=""> Content </TagName>

اغلب تگ‌های HTML به‌صورت جفتی هستند. یعنی دارای تگ آغازین و پایانی هستند. تگ پایانی با علامت اسلش (/) شروع می‌شود. برخی تگ‌ها نیز به‌صورت خود بسته یا self-closing هستند و دارای تگ پایانی نیستند.

مثال

به‌طور مثال تگ p به‌صورت جفتی نوشته می‌شود و تگ img به‌صورت خودبسته:

```
<p> This is a paragraph </p>
<img src="./flower.jpg" />
```

عبارت Content به محتوای داخل تگ HTML اشاره دارد. مجموعه تگ‌های آغازین، پایانی و محتوای بین آنها، یک عنصر وب را تشکیل می‌دهد.
عبارت Attribute به ویژگی یک عنصر اشاره می‌کند. به طور مثال تگ <img> دارای ویژگی width، height و src برای تعیین پهنا، ارتفاع و آدرس تصویر است:

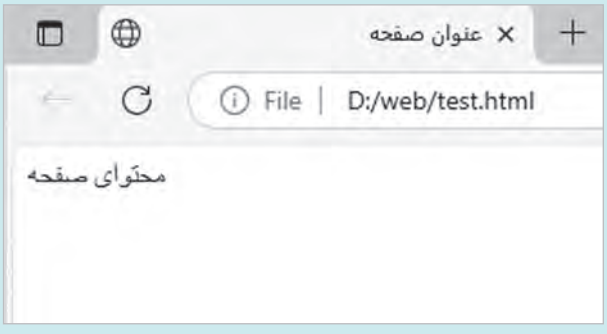
```
<img src="flower.jpg" width="200px" height="150px">
```

کار کارگاهی

ایجاد یک صفحه وب ساده

برای ایجاد یک صفحه وب، باید از یک ساختار استاندارد استفاده شود، در غیراین‌صورت مشکلات مختلفی در زمینه نمایش صفحه توسط مرورگرهای مختلف، دسترس‌پذیری محتوا برای افراد نابینا، سرعت بارگذاری و .... ایجاد می‌شود. ساختار اولیه یک صفحه وب شامل تگ‌ها (برچسب‌های) تصویر روبه‌رو است:

```
<!DOCTYPE html>
<html lang="fa">
<head>
  <meta charset="UTF-8">
  <title> عنوان صفحه </title>
</head>
<body>
    محتوای صفحه
</body>
</html>
```





شکل ۶

ـ **تگ <Doctype html!>**: در ابتدای یک صفحه وب از تگ <Doctype! html> استفاده می‌شود. این تگ به مرورگرها و موتورهای جست‌وجو اعلام می‌کند که این سند از نوع html5 است.

ـ **تگ <html>**: <html> تگ ریشه سند HTML است. تمام محتوای یک صفحه وب درون این تگ قرار می‌گیرد. این تگ دارای یک ویژگی مهم به‌نام lang است که زبان صفحه را برای مرورگر، ابزارهای صفحه‌خوان و موتورهای جست‌وجو مشخص می‌کند. تگ <html> دارای ویژگی دیگری با نام dir است که جهت نمایش محتوا را مشخص می‌کند و دارای دو مقدار rtl و ltr به معنای راست به چپ و چپ به راست است.



شکل ۷

ـ **تگ <head>**: این برچسب باید قبل از تگ <body> نوشته شود. در این بخش اطلاعات متا، عنوان صفحه، الحاق فایل‌های CSS و دیگر منابع غیرنمایشی قرار می‌گیرد. این اطلاعات برای مرورگر و موتورهای جست‌وجو مهم هستند.

ـ **تگ <title>**: این تگ عنوان صفحه را مشخص می‌کند که در نوار عنوان مرورگر و نتایج جست‌وجو نمایش داده می‌شود.

ـ **تگ‌های <meta>**: این تگ‌ها اطلاعاتی را درباره صفحه وب فراهم می‌کنند. برخی از متداول‌ترین تگ‌های متا عبارت‌انداز:

- **متاتگ Charset**: نوع رمزگذاری کاراکترهای صفحه وب را برای مرورگر مشخص می‌کند. این تگ به مرورگرها اعلام می‌کند که کاراکترهای صفحه را چگونه تفسیر کنند تا محتوای صفحه به درستی نمایش داده شود.
- **متاتگ viewport**: برای بهینه‌سازی نمایش صفحه وب بر روی دستگاه‌های موبایل و تبلت استفاده می‌شود:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

کنجکاوی

درباره این متاتگ تحقیق کنید و نتیجه را توسط چند تصویر درباره تأثیر به‌کارگیری این تگ بر چیدمان عناصر صفحه در دستگاه‌های مختلف نمایش دهید.



شکل ۸

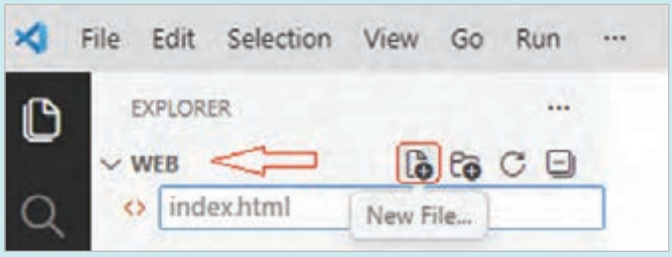
ـ **تگ <body>**: شامل محتوای قابل مشاهده برای کاربران است. تمام عناصر نمایشی مانند متن، تصاویر، ویدئوها و غیره در این بخش قرار می‌گیرند.

ایجاد یک صفحه وب ساده

کدهای HTML درون یک فایل متنی با پسوند html ذخیره می‌شوند. برای نوشتن این کدها می‌توان از ویرایشگرهای متنی ساده مانند Notepad یا ویرایشگرهای حرفه‌ای همچون VScode استفاده نمود. از قابلیت‌های برنامه VScode می‌توان به نمایش رنگی کدها، تکمیل خودکار کد، امکان اجرای سریع کد برنامه و اضافه کردن انواع افزونه‌ها اشاره کرد.

ایجاد فایل HTML

برای مجموعه تمرین‌ها یک پوشه با نام Web در مسیر Document یا یکی از درایوها ایجاد کنید تا فایل‌های HTML درون آن پوشه ذخیره گردد. در برنامه VScode از منوی file گزینه Open folder را انتخاب کنید و سپس پوشه Web را انتخاب کنید تا در مدت زمان کار روی پروژه محتوای پوشه در سمت چپ پنجره قابل مشاهده و در دسترس باشد.

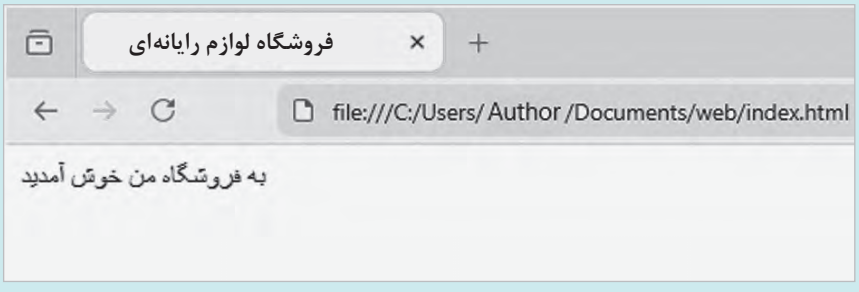


سپس روی دکمه New file کلیک کنید، نام فایل را مطابق تصویر بالا درون کادر متنی بنویسید و کلید Enter را بفشارید تا فایل ایجاد شود.

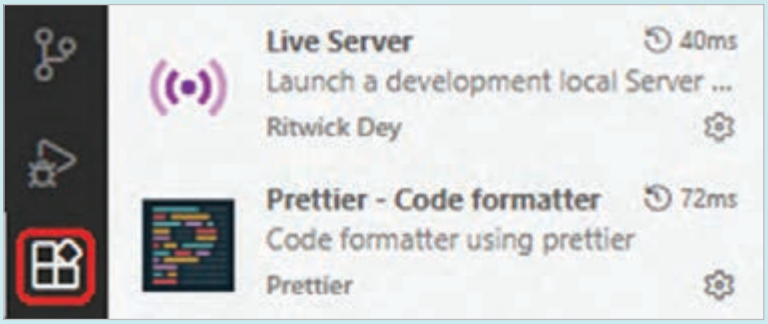
کدهای اولیه HTML را به همراه یک متن ساده، مطابق تصویر زیر بنویسید و با فشردن کلید Ctrl+S آن را ذخیره کنید.

نمایش صفحه وب در مرورگر:

ـ روش اول: برای مشاهده خروجی کدها این است که کلیدهای Win+E را بفشارید و پوشه Web را از مسیر مورد نظر خود باز کنید. سپس فایل index.html را با مرورگر دلخواه باز کنید. بعد از انجام هر تغییر در فایل Html لازم است که محتوای صفحه را در مرورگر تازه‌سازی (Refresh) کنید. این کار از طریق دکمه F5 یا کلیک روی دکمه در بالای صفحه امکان‌پذیر است.



ـ روش دوم: برای مشاهده خروجی استفاده از قابلیت Go Live در برنامه VScode است که در این حالت، مشاهده خروجی نیازی به تازه‌سازی صفحه ندارد. برای استفاده از قابلیت Go Live لازم است که افزونه Live Server را در برنامه Vscode نصب کنید.
بدین منظور روی دکمه Extensions در سمت چپ پنجره برنامه کلیک کنید و سپس افزونه live server را جست‌وجو و نصب کنید.



بعد از نصب افزونه، برای مشاهده خروجی برنامه کافیست رو دکمه Go live در پایین پنجره کلیک کنید.



به این ترتیب صفحه مرورگر به شکل زیر ظاهر می‌گردد:

```
<!DOCTYPE html>
<html lang="fa">
<head>
    <meta charset="UTF-8">
    <title> فروشگاه لوازم رایانه‌ای </title>
</head>
<body>

به فروشگاه من خوش آمدید

 </body>
</html>
```

خروجی



بعد از انجام هر تغییر جدید در کد صفحه، فایل را توسط کلید Ctrl+S ذخیره کنید، سپس توسط کلید ترکیبی Alt+Tab به صفحه مرورگر مراجعه کنید. خروجی جدید بدون نیاز به تازه‌سازی ظاهر می‌شود.

فعالیت

به محتوای نوار آدرس در دو روش نمایش خروجی توجه کنید. چه تفاوتی بین آنها وجود دارد؟ درباره آن در کلاس گفت‌وگو کنید.

عناصر متنی در صفحه وب

در یک صفحه وب انواع مختلفی از عناصر متنی وجود دارد که نشانه‌گذاری هر یک از آنها توسط تگ مشخصی صورت می‌پذیرد. این بخش به معرفی عناصر متنی صفحه وب می‌پردازد.

تیتربندی تگ‌های <h6>... <h1>:

صفحات وب دارای انواع متفاوتی از محتوا هستند. در یک صفحه وب که حاوی یک مقاله علمی یا ادبی باشد، برای سازمان‌دهی محتوا از تیترهای مختلف استفاده می‌شود. چنین تیترهایی در یک صفحه فروشگاهی کمتر به چشم می‌خورند، اما برای بخش‌های خاصی از صفحه کاربرد دارند.



شکل ۹

برای نشانه‌گذاری تیترهای صفحه وب از تگ <h1> تا <h6> استفاده می‌شود. تگ <h1> برای موضوع اصلی صفحه استفاده می‌شود. تگ <h2> برای تیترهای زیرمجموعه موضوع اصلی به کار می‌رود که این تیترها دسته‌بندی موضوعات مطرح شده در صفحه را مشخص می‌کنند؛ بنابراین دارای اهمیت بالایی هستند. تگ‌های <h3> تا <h6> برای تیترهایی با موضوعات جزئی‌تر استفاده می‌شوند.

شکل کلی تگ‌های تیترگذاری

<h1> تیتر اصلی صفحه </h1>
<h2> تیتر درجه ۲ </h2>
<h3> تیتر درجه ۳ </h3>

تیتر <h1> برای موتورهای جست‌وجو دارای اهمیت بالایی است چرا که موضوع صفحه را برای آنها مشخص می‌کند. این تیتر باید حاوی کلمات کلیدی موضوع صفحه باشد. توصیه می‌شود که این تیتر فقط یک‌بار در صفحه به کار برده شود.

به‌طور کلی تگ‌های <h1> تا <h6> سلسله مراتب موضوعات صفحه را مشخص می‌کنند. با استفاده درست این عناصر، کاربران می‌توانند محتوای صفحه را راحت‌تر مطالعه کنند و موضوعات مورد نظر خود را در میان محتواهای موجود پیدا کنند. علاوه بر این موتورهای جست‌وجو به راحتی متوجه موضوعات مطرح شده در صفحه می‌شوند و می‌توانند اطلاعات صفحه وب را به شکل بهتری در پایگاه داده خود ذخیره نمایند. تیترگذاری مناسب محتوا تأثیر قابل توجهی در رتبه صفحه وب در نتایج جست‌وجو دارد.

کار کارگاهی

به‌کارگیری تگ‌های تیترگذاری در صفحه test.html مطابق شکل زیر، تگ‌های تیتربندی را اعمال و نتیجه را در مرورگر مشاهده کنید.

```
<> test.html > ...
 1   <!DOCTYPE html>
 2   <html lang="fa">
 3   <head>
 4       <meta charset="UTF-8">
 5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
 6       <title>فروشگاه کتاب های درسی</title>
 7   </head>
 8   <body>
 9       <h1>کتاب‌های درسی رشته شبکه و نرم افزار</h1>
10       <h2>کتاب های پایه دهم</h2>
11       <h3>نگهداری سیستم های رایانه ای</h3>
12       <h3>ارایه دهنده خدمات رایانه ای</h3>
13       <h3>دانش فنی پایه</h3>
14       <h3>نقشه کشی فنی رایانه ای</h3>
15   </body>
16   </html>
```

کتاب‌های درسی رشته شبکه و نرم‌افزار
کتاب‌های پایه دهم
نگهداری سیستم‌های رایانه‌ای
ارائه‌دهنده خدمات رایانه‌ای
دانش فنی پایه
نقشه‌کشی فنی رایانه‌ای

■ **تگ **: این تگ برای نشانه‌گذاری پاراگراف‌ها مورد استفاده قرار می‌گیرد. محتوایی که بین و قرار می‌گیرد، از محتوای قبل و بعد خود جدا شده و در یک سطر جدید نمایش داده می‌شود. می‌توان درون از متن ساده، لینک <a>، برجسته‌سازی <b> یا و ... استفاده کرد.



شکل ۱۰

■ **تگ‌های و <b>**: برای نمایش یک کلمه به‌صورت ضخیم، می‌توان از دو تگ <b> و استفاده نمود. مرورگر وب محتوای این دو تگ را به‌صورت یکسان نمایش می‌دهد، اما موتورهای جست‌وجو در برخورد با آنها تفسیر متفاوتی دارند. اگر عبارتی با تگ نشانه‌گذاری شود، به معنای آن است که محتوای این عنصر دارای اهمیت بالایی است و با موضوع صفحه ارتباط دارد. ابزارهای صفحه‌خوان (Screen reader) نیز محتوای تگ را با صدای بلندتری می‌خوانند، درحالی‌که در برخورد با تگ <b> تغییری در لحن خود ایجاد نمی‌کنند، چرا که تگ <b> مفهوم خاصی را منتقل نمی‌کند.

■ **تگ‌های و <i>**: برای نمایش یک کلمه به صورت کج، از دو تگ و <i> استفاده می‌شود. هرچند که مرورگرهای وب محتوای این دو تگ را به صورت مشابه نمایش می‌دهند، اما معنای آنها در HTML و تأثیرشان بر روی موتورهای جست‌وجو و دسترسی به محتوا متفاوت است.

■ **تگ ** : مخفف عبارت Emphasis و به معنای تأکید است. استفاده از برچسب باعث نمایش متفاوت یک عبارت و افزایش خوانایی متن می‌گردد. این امر به بهبود تجربه کاربری کمک می‌کند و در نتیجه تأثیر مثبتی بر سئوی صفحه خواهد داشت. ابزارهای صفحه‌خوان نیز در برخورد با تگ با تغییر لحن صدای خود، اهمیت محتوای تگ را به کاربران نابینا اعلام می‌کنند.

■ **تگ <i>:** که مخفف عبارت italic است، بیشتر برای اهداف ظاهری و زیبایی‌شناسی به کار می‌رود و موتورهای جست‌وجو به آن توجه خاصی ندارند. همچنین دستگاه‌های صفحه‌خوان هنگام مواجه شدن با تگ <i> تغییری در لحن خود ایجاد نمی‌کنند.

■ **به‌کارگیری تگ‌های <i> و :**

test.html > ...

```
<!DOCTYPE html>
<html lang="fa" dir="rtl">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>کتاب‌های درسی رشته شبکه و نرم‌افزار رایانه</title>
</head>
<body>
    <h1>پایه دهم: رشته شبکه و نرم‌افزار رایانه</h1>
    <h2>معرفی کتاب‌های تخصصی پایه دهم</h2>
    <h3>سرفصل‌های اصلی دروس فنی</h3>
    <h4>لیست کتاب‌های درسی</h4>
    <h5>نگهداری سیستم‌های رایانه‌ای</h5>
    <p> در این درس، نحوه <strong>تشخیص عیب</strong>، <em>تعویض قطعات</em>
        سخت‌افزاری و نگهداری پیشگیرانه از کامپیوترها آموزش داده می‌شود.
    </p>
    <h5>ارائه‌دهنده خدمات رایانه‌ای</h5>
    <p>و <strong>خدمات پشتیبانی</strong> این کتاب به دانش‌آموزان می‌آموزد که چگونه
        <em>مشاوره فنی</em> را به کاربران ارائه دهند و با اصول کار در مراکز خدماتی آشنا شوند.
    </p>
</body>
</html>
```

شکل ۱۱

عبارت"trl"= dir در تگ <html> برای تعیین جهت نمایش محتوا از راست به چپ استفاده شده است. خروجی کد به‌صورت شکل ۱۲ خواهد بود:

پایه دهم: رشته شبکه و نرم‌افزار رایانه

معرفی کتاب‌های تخصصی پایه دهم

سرفصل‌های اصلی دروس فنی

لیست کتاب‌های درسی

۱. نگهداری سیستم‌های رایانه‌ای

در این درس، نحوه **تشخیص عیب**، *تعویض قطعات* سخت‌افزاری و نگهداری پیشگیرانه از کامپیوترها آموزش داده می‌شود.

۲. ارائه‌دهنده خدمات رایانه‌ای

این کتاب به دانش‌آموزان می‌آموزد که چگونه **خدمات پشتیبانی** و *مشاوره فنی* را به کاربران ارائه دهند و با اصول کار در مراکز خدماتی آشنا شوند.

توجه: مطالعه این درس برای ورود به پایه‌های بالاتر الزامی است.

شکل ۱۲



شکل ۱۳

■ **تگ‌های لیست:** تگ‌های لیست برای سازمان‌دهی و نمایش اطلاعات استفاده می‌شوند. این تگ‌ها امکان مشاهده محتوا را به‌صورت منظم و قابل فهم فراهم می‌کنند. استفاده از لیست‌ها برای سازماندهی محتوا باعث خوانایی و جذابیت محتوا برای کاربران و در نتیجه بهبود تجربه کاربری و رتبه بهتر در نتایج جست‌وجو می‌شود.

سه نوع لیست در HTML قابل نشانه‌گذاری است: لیست‌های مرتب، نامرتب و توضیحی.
این لیست‌ها توسط تگ‌های ، و <dl> ایجاد می‌شوند.

ـ **تگ <ul>:** مخفف عبارت Unordered list است. این تگ برای ایجاد یک لیست گلوله‌دار به‌کار می‌رود. برای علامت‌گذاری عناصر لیست از تگ <li> استفاده می‌شود.

کار کارگاهی

استفاده از تگ لیست:
فایل جدیدی به نام services.html ایجاد کنید. کدهای زیر را در بخش <body> وارد کنید و نتیجه را در مرورگر مشاهده کنید.
فایل products.html را ایجاد کنید تگ <ol> را جایگزین تگ <ul> نمایید و عبارات لیست شکل زیر را درون آن بنویسید. فایل را ذخیره کنید و نتیجه را در مرورگر مشاهده کنید.

```
services.html > ...
1  <!DOCTYPE html>
2  <html lang="fa" dir="rtl">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>فروشگاه کتاب های درسی</title>
7  </head>
8  <body>
9  <h2>خدمات ما</h2>
10 <ul>
11     <li>ارسال سریع سفارشات</li>
12     <li>به روزترین فروشگاه کتاب</li>
13     <li>ارسال به سراسر کشور</li>
14     <li>پشتیبانی 24 ساعته</li>
15 </ul>
16 </body>
17 </html>
```

خدمات ما
- ارسال سریع سفارشات
- به روزترین فروشگاه کتاب
- ارسال به سراسر کشور
- پشتیبانی ۲۴ ساعته

ـ **تگ <ol>:** مخفف عبارت Ordered list است. برای ایجاد لیست‌های شماره دار تگ <ol> استفاده می‌شود.

```
products.html > ...
1  <!DOCTYPE html>
2  <html lang="fa" dir="rtl">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>فروشگاه کتاب های درسی</title>
7  </head>
8  <body>
9  <h1>کتاب‌های رشتهٔ شبکه و نرم افزار رایانه</h1>
10 <p>تجربه ای تازه بخش از خرید اینترنتی برای شما تداعی می شود. <strong>بیشترین تخفیف</strong> با</p>
11 <ol>
12     <li><strong>دروس عمومی</strong></li>
13     <li><strong>دروس پایه</strong></li>
14     <li><strong>دروس شایستگی‌های غیر فنی</strong></li>
15     <li><strong>دروس شایستگی‌های فنی</strong></li>
16 </ol>
17 </body>
18 </html>
```

کتاب‌های رشتهٔ شبکه و نرم‌افزار رایانه

با بیشترین تخفیف تجربه‌ای لذت‌بخش از خرید اینترنتی برای شما تداعی می‌شود.

۱ـ دروس عمومی
۲ـ دروس پایه
۳ـ دروس شایستگی‌های غیرفنی
۴ـ دروس شایستگی‌های فنی

■ **پیوندها (links):** پیوندها همانند پل ارتباطی میان صفحات هستند که کاربران را به صفحات و منابع مورد نظرشان هدایت می‌کنند.

ـ برای ایجاد ارتباط بین صفحات از تگ <a> استفاده می‌شود. این تگ مخفف anchor و به معنای «لنگر» است. تگ <a> می‌تواند کاربر را به یک صفحه وب، تصویر، فایل pdf و موارد دیگر هدایت کند.



شکل ۱۴

ـ برای ایجاد پیوند از ساختار کلی زیر استفاده می‌شود:

 anchor Text

ـ در ویژگی href آدرس صفحه یا فایل مقصد نوشته می‌شود. عبارت href مخفف Hypertext reference است.

ـ عبارت anchor text به متن پیوند؛ همان چیزی که کاربر می‌بیند و روی آن کلیک می‌کند؛ اشاره دارد. چنانچه کاربر روی متن پیوند کلیک نماید به آدرس منبع مشخص شده در ویژگی href هدایت می‌شود.

ـ صفحه مقصد پیوندها به‌صورت پیش‌فرض در صفحه وب جاری باز می‌شود؛ در نتیجه محتوای قبلی صفحه ناپدید می‌گردد.

ـ برای باز شدن صفحه مقصد در برگه جدیدی از مرورگر، از ویژگی target با مقدار _blank استفاده می‌شود. مقدار پیش‌فرض ویژگی target برابر _self است؛ به این معنی که اگر ویژگی target در تگ <a> مشخص نشود، مقصد لینک در همان صفحه باز شود.

ـ ویژگی title حاوی توضیحی درباره لینک است. این ویژگی اختیاری است.

مثال

مثال‌های زیر نمونه‌هایی از پیوند به منابع مختلف است:

 تارنمای وزارت آموزش و پرورش

 دانلود کتاب طراحی صفحات وب

ـ مقصد یک لینک می‌تواند نقطه دیگری از صفحه جاری باشد. در این حالت شناسه (ID) عنصر مقصد در ویژگی href نوشته می‌شود.

کار کارگاهی

در صفحه test.html تغییرات زیر را ایجاد کنید و نتیجه را در مرورگر مشاهده نمایید.

```
<> test.html > ...
1   <!DOCTYPE html>
2   <html lang="fa" dir="rtl">
3   <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>مرجع نو</title>
7   </head>
8   <body>
9       <h1>مرجع نو</h1>
10      <p>ارائه دهنده آخرین نوبت چاپ کتاب‌های درسی</p>
11      <a href="./test.html">خانه</a>
12      <a href="./products.html">محصولات</a>
13      <a href="./services.html">خدمات ما</a>
14      <a href="#">سبد خرید</a>
15  </body>
16  </html>
```

مرجع نو

ارائه دهنده آخرین نوبت چاپ کتاب‌های درسی

خانه محصولات خدمات ما سبد خرید

افزودن تصاویر به صفحه وب

تصاویر یکی از عناصر مهم در محتوای یک صفحه هستند که می‌توانند تأثیرات عمیقی بر جذابیت، ارتباط پذیری و اثربخشی محتوای صفحه داشته باشند. استفاده از تصاویر در صفحات وب می‌تواند باعث جلب توجه کاربران شود به‌گونه‌ای که مدت زمان توقف آنها در صفحه بیشتر شود. به‌علاوه تصاویر نقش مهمی در توضیح و تبیین محتوای متنی دارند، چرا که می‌توانند اطلاعات را به صورت بصری منتقل کنند.



شکل ۱۵

ـ برای نمایش یک تصویر در صفحه وب از تگ <img> استفاده می‌شود. شکل کلی این تگ به‌صورت زیر است:

<**img src**="URL" **alt**="Description" **title**=" " **width**="value" **height**="value" />

■ **src**: این ویژگی آدرس تصویر را تعیین می‌کند.

■ **alt**: کلمه alt مخفف Alternative text است. مقدار این ویژگی، متن جایگزین تصویر را مشخص می‌کند. این متن در صورت عدم بارگذاری تصویر نمایش داده می‌شود. به‌علاوه مقدار Alt بیانگر مفهوم و محتوای تصویر برای کاربر و موتورهای جست‌وجو است.

■ **width و height**: این دو ویژگی پهنا و ارتفاع عناصر صفحه مانند تصویر را مشخص می‌کنند. مقدار این ویژگی‌ها می‌تواند برحسب پیکسل یا درصد باشد. مرورگر با مشاهده عبارت "width="100 پهنای عنصر را برای ۱۰۰ پیکسل قرار می‌دهد. بنابراین برای تعیین واحد بر حسب درصد باید از عبارتی مشابه "width="100 استفاده شود. بهتر است که برای تنظیم پهنا و ارتفاع بر حسب درصد از CSS استفاده شود.

■ **title**: یک متن توضیحی است که در صورت قرارگیری نشانگر ماوس روی تصویر ظاهر می‌شود.

کار کارگاهی

استفاده از تگ img

در فایل products.html برای هر یک از کتاب‌ها تصویری را نمایش دهید.

■ برای انجام این‌کار در پوشه web پوشه به نام images بسازید و تصاویر کتب را درون آن قرار دهید.

■ توسط نرم‌افزار paint، ابعاد تصاویر را در اندازه ۲۰۰×۲۸۰ تنظیم کنید. در صورت لزوم فضای خالی اطراف محصول را افزایش دهید.

■ سپس تصاویر را بعد از تیتر کتب نمایش دهید.

■ هر تصویر را تبدیل به پیوند کنید. با توجه به اینکه صفحه مقصد پیوندها موجود نیست، برای مقصد پیوند از علامت # استفاده کنید. سپس صفحه را با فشردن کلید (F5) تازه‌سازی کنید.

```
11  <ol>
12      <li><strong>دروس عمومی:</strong></li>
13      <li><strong>دروس پایه:</strong></li>
14      <li><strong>دروس شایستگی‌های غیر فنی:</strong></li>
15      <li><strong>دروس شایستگی‌های فنی:</strong></li>
16      <a href="#"><img src="./images/danesh_fanni_paye.jpg" alt="کتاب دانش فنی پایه"> </a>
17      <a href="#"><img src="./images/negahdarie_system.jpg" alt="نگهداری سیستم های رایانه ای"> </a>
18  </ol>
```

۱ـ دروس عمومی
۲ـ دروس پایه
۳ـ دروس شایستگی‌های غیرفنی
۴ـ دروس شایستگی‌های فنی



بیشتر بدانید

طراحی جدول (table)

جدول یکی از ابزارها در نمایش داده‌ها است؛ چرا که امکان نمایش منظم و قابل فهم اطلاعات را فراهم می‌کند. امروزه جداول برای نمایش داده‌های مختلف مانند لیست محصولات، قیمت‌ها، یا هر نوع اطلاعات نیازمند سازماندهی استفاده می‌شود.

برای ایجاد یک جدول در HTML، از تگ <table> و تگ‌های زیرمجموعه آن، به شرح زیر استفاده می‌شود:

- **<table>**: این تگ شروع و پایان جدول را مشخص می‌کند.
- **<caption>**: برای تعیین عنوان جدول به‌کار می‌رود. این تگ موضوع جدول را برای کاربران و موتورهای جست‌وجو مشخص می‌کند.
- **<tr>**: برای تعریف یک ردیف (Row) در جدول به‌کار می‌رود.
- **<th>**: برای ایجاد یک سلول عنوان (Header) در ردیف عنوان استفاده می‌شود. این تگ تجربه کاربری را بهبود می‌دهد.
- **<td>**: برای تعریف یک سلول داده (Data Cell) در جدول کاربرد دارد.

تگ <table> دارای ویژگی‌های مختلفی برای تعیین ظاهر جدول است که در ادامه به آنها اشاره می‌شود.

- **rowspan**: برای ادغام سلول‌ها در راستای عمود.
- **colspan**: برای ادغام سلول‌ها در راستای افقی.
- **tfoot، tbody، thead**: برای ساختاردهی معنایی.

کنجکاوی

درباره ویژگی rowspan، colspan و کاربرد آن در جدول تحقیق کنید و با یک پروژه نتیجه را در کلاس ارائه کنید.

نکته

در HTML5 استفاده از ویژگی border در برچسب <table> غیرمجاز است و تعیین سبک نمایش خطوط جدول باید توسط CSS صورت گیرد. با توجه به اینکه مبحث CSS در واحد کار دوم مطرح می‌شود، از ویژگی border صرفاً برای نمایش موقت خطوط استفاده می‌شود.

کار کارگاهی

به‌کارگیری تگ جدول

مطابق تصویر، تگ‌های مورد نیاز را برای ایجاد جدول بنویسید و نتیجه را در مرورگر مشاهده کنید.

```
<body>
<table border="1">
    <caption>محصولات موجود</caption>
    <thead>
        <tr>
            <th>کد محصول</th>
            <th>نام محصول</th>
            <th>قیمت</th>
            <th>موجودی</th>
            <th>تصویر محصول</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>210288</td>
            <td>دانش فنی پایه</td>
            <td>350000 تومان</td>
            <td>50</td>
            <td><img src="./images/danesh-fani.jpg" alt="دانش فنی پایه" width="150"></td>
        </tr>
    </tbody>
</table>
</body>
```

محصولات موجود

کد محصول	نام محصول	قیمت	موجودی	تصویر محصول
210288	دانش فنی پایه	350000 تومان	50	

طراحی فرم

فرم‌های ورود اطلاعات



شکل ۱۶

فرم‌ها ابزارهای قدرتمندی برای جمع‌آوری اطلاعات از کاربران هستند. یک فرم ورود داده، اطلاعات کاربران را دریافت می‌کند و سپس آنها را برای پردازش و ذخیره‌سازی به یک وب‌سرور ارسال می‌نماید. پردازش و ذخیره‌سازی در سمت سرور انجام می‌شود. یک فرم اطلاعاتی شامل تگ‌های مختلفی برای دریافت داده‌های کاربران است. این تگ‌ها به شرح زیر هستند:

<form>: برای تعریف یک فرم استفاده می‌شود.

شکل کلی این تگ به‌صورت زیر است:

```
<form action="URL" method="http_method" enctype="encoding_type">
```

ویژگی‌های عنصر فرم به شرح زیر است:

action: این ویژگی مشخص می‌کند که داده‌های فرم برای پردازش به چه آدرسی (URL) در سمت سرور ارسال شوند.

method: روش ارسال داده های فرم را به سرور مشخص می‌کند. این ویژگی دارای دو مقدار get و post است. در روش get داده‌های فرم به عنوان پارامترهای URL به سرور ارسال می‌شوند. این روش برای ارسال داده‌های حساس کاربر مناسب نیست. در روش post اطلاعات فرم به‌صورت پنهانی به سرور ارسال می‌شوند، این روش برای ارسال داده‌های حساس مناسب است. درجدول صفحهٔ بعد مقایسه دو روش ارسال اطلاعات به در فرم‌ها را مشاهده کنید (جدول ۵).

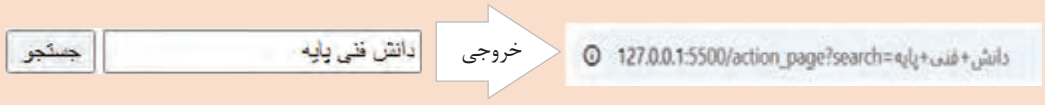
جدول ۵ــ مقایسه دو روش ارسال اطلاعات به در فرم‌ها

عملیات	روش GET	روش POST
نحوه ارسال داده‌ها	داده‌ها از طریق نوار آدرس مرورگر ارسال می‌شوند و در نتیجه توسط کاربر مشاهده می‌شوند.	داده‌ها در بدنه درخواست HTTP ارسال می‌شوند و در URL مشاهده نمی‌شوند.
محدودیت حجم داده	مرورگرها در هنگام ارسال درخواست توسط روش GET دارای محدودیت هستند. بسته به نوع مرورگر و تنظیمات سرور، طول رشته قابل ارسال از طریق نوار آدرس بین ۲۰۰۰ تا ۸۰۰۰ کاراکتر است.	محدودیت کمی دارد و می‌تواند برای ارسال فایل و متن‌های طولانی استفاده شود.
امنیت	امنیت پایینی دارد، علاوه بر نمایش داده‌های کاربران در نوار آدرس، امکان ذخیره شدن محتوای آدرس (کش شدن) در حافظه مرورگر موجود است.	امن‌تر از روش GET است اما امنیت کامل ندارد، مگر اینکه از پروتکل HTTPS استفاده شود.
کاربردها	برای دریافت اطلاعات از سرور (مانند جست‌وجو و فیلتر کردن). در این حالت کاربر قصد تغییر داده‌های سرور را ندارد.	برای ارسال اطلاعات به سرور جهت پردازش یا ذخیره‌سازی (مانند ثبت‌نام، ورود، آپلود فایل)

مثال

مثالی از به‌کارگیری متد GET

```
<form action="action_page" method="get">
    <input type="text" name="search" placeholder="جستجو">
    <input type="submit" value="جستجو">
</form>
```



مثال

مثالی از به‌کارگیری متد POST

```
<form action="action_page" method="post">
    <input type="email" name="email" placeholder="ایمیل">
    <input type="password" name="password" placeholder="رمز عبور">
    <input type="submit" value="ورود">
</form>
```



یک فرم حاوی مجموعه‌ای از عناصر برای دریافت اطلاعات مختلف از کاربران است که در جدول ۶ نمایش داده شده است:

جدول ۶ــ عناصر فرم (تگ هایی که در عنصر فرم مورد استفاده قرار می‌گیرند) و کاربرد آنها

عنصر فرم	کاربرد و شکل کلی عنصر
<label>	تعیین برچسب عناصر فرم.
	<label for=" "> </label>
<input>	دریافت اطلاعات مختلف از کاربر. نوع ورودی توسط ویژگی type مشخص می‌شود.
	<input type="" name="" id="" value="" placeholder="" required >
<fieldset>	گروه‌بندی عناصر فرم، این تگ باعث نمایش یک کادر دور گروه فیلدها می‌شود.
<legend>	<fieldset> <legend> </legend> </fieldset> برچسب legend عنوان گروه را مشخص می‌کند.
<textarea>	دریافت عبارات طولانی از کاربران، اندازه کادر متنی با ویژگی‌های rows و cols مشخص می‌شود.
	<textarea id="" name=" " rows="" cols="" required> </textarea>
<select>	ایجاد فهرست یا لیست کشویی. گزینه‌های لیست توسط تگ <option> مشخص می‌شود.
	<select id="" name="" required> <option value=""> </option> </select>

ویژگی type در عنصر <input> نوع داده ورودی کاربر را تعیین می‌کند. مقدار این ویژگی در جدول ۷ تشریح شده است:

جدول ۷ــ مقادیر ویژگی type در عنصر <input>

مقدار ویژگی type	کاربرد
text, password, email	دریافت مقادیر متنی کوتاه مانند نام کاربری، کلمه عبور و ایمیل.
tel, number	دریافت شماره تلفن و مقادیر عددی.
radio	ایجاد لیستی از گزینه‌ها که کاربر تنها مجاز به انتخاب یک گزینه باشد.
checkbox	ایجاد لیستی از گزینه‌ها به‌کار می‌رود که کاربر امکان انتخاب صفر، یک یا چند گزینه را دارد.
file	دریافت فایل از کاربران (آپلود).
submit	ایجاد دکمه ارسال اطلاعات
reset	محتوای فرم را پاک می‌کند.

برای ایجاد یک گروه دکمه رادیویی باید ویژگی name برای تمام دکمه‌های رادیویی گروه یکسان باشد، در غیراین‌صورت امکان انتخاب چند عنصر وجود دارد. دکمه‌های رادیویی معمولاً زمانی به کار می‌روند که انتخاب یک گزینه از میان چند گزینه مطرح باشد.

مثال

مثال:

```
<input type="radio" name="gender" value="male"> مرد
<input type="radio" name="gender" value="female"> زن
```

کار کارگاهی

ایجاد فرم ثبت سفارش

یک صفحه جدید با نام order_form.html ایجاد کنید و فرم ثبت سفارش را به صورت شکل زیر درون آن بنویسید. سپس نتیجه را در مرورگر بررسی کنید.

```
<form action="" method="post">
  <h2>ثبت سفارش</h2>
  <label for="fullname">نام و نام خانوادگی</label>
  <input type="text" id="fullname" required> <br><br>

  <label for="phone">تلفن همراه</label>
  <input type="tel" id="phone" placeholder="09xx-123-4567"> <br><br>

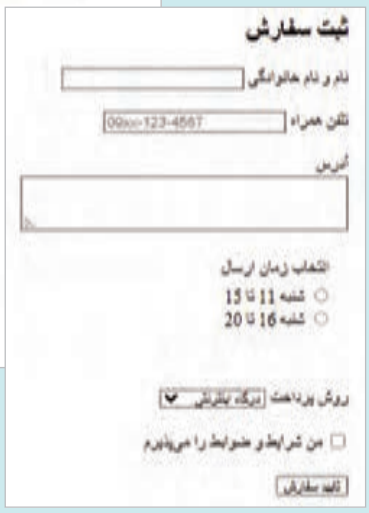
  <label for="address">آدرس</label><br>
  <textarea id="address" rows="3" cols="50" required></textarea> <br><br>

  <fieldset>
      <legend>انتخاب زمان ارسال</legend>
      <input type="radio" name="time" id="time1" value="saturday-11-15">
      <label for="time1">شنبه 11 تا 15</label>
      <br>
      <input type="radio" name="time" id="time2" value="saturday-16-20">
      <label for="time2">شنبه 16 تا 20</label>
  </fieldset> <br><br>

  <label for="payment-method">روش پرداخت</label>
  <select id="payment-method" required>
      <option value="online">درگاه اینترنتی</option>
      <option value="cash">پرداخت در محل</option>
  </select> <br><br>

  <input type="checkbox" id="terms">
  <label for="terms">من شرایط و ضوابط را می‌پذیرم</label>
  <br><br>
  <button type="submit">تأیید سفارش</button>
</form>
```

خروجی



- **name**: نام عنصر را تعیین می‌کند که در کدنویسی سمت سرور کاربرد دارد.
- **id**: شناسه عنصر فرم را تعیین می‌کند که باید مقداری منحصر به‌فرد باشد.
- **value**: مقدار پیش‌فرض عنصر فرم را تعیین می‌کند.
- **required**: استفاده از این ویژگی در عناصر فرم باعث می‌شود که در صورت خالی رها شدن یک فیلد پیغام خطا ظاهر شود.

فعالیت

یک فرم بازگشت کالا حاوی نام و نام خانوادگی، شماره سفارش، نام کالا، علت مرجوع کالا و تصویر کالا ایجاد کنید.

ساختاردهی به صفحه با عناصر معنایی (Semantic Elements)

عناصر معنایی در HTML به عناصری اطلاق می‌شود که محتوای درون خود را به وضوح توصیف می‌کنند. استفاده از این عناصر باعث می‌شود که ابزارهای صفحه‌خوان و موتورهای جست‌وجو درک بهتری از ساختار و محتوای صفحه داشته باشند. عناصر معنایی نه تنها سئوی صفحه را بهبود می‌دهند، بلکه سازمان‌دهی و نگهداری کد را ساده‌تر می‌کنند. این عناصر به شرح زیر هستند:

<main>: برای نمایش محتوای اصلی صفحه به‌کار می‌رود. هر صفحه باید تنها یک عنصر <main> داشته باشد.

<header>: برای تعریف سرصفحه یک سند استفاده می‌شود. درون این تگ امکان استفاده از تگ‌های مختلفی مانند <h1>، <p>، <nav>، ، <img> و <form> وجود دارد.

<nav>: در هنگام ایجاد منوهای ناوبری[1] استفاده می‌شود (شکل ۱۷).

```
<nav>
    <a href="./test.html">خانه</a>
    <a href="./products.html">محصولات</a>
    <a href="./services.html">خدمات ما</a>
    <a href="#">سبد خرید</a>
    <form action="action_page" method="post">
        <input type="text" name="search">
        <button type="submit">جستجو</button>
    </form>
</nav>
```

شکل ۱۷

۱ـ منوی ناوبری (Navigation Menu) شامل مجموعه‌ای از پیوندهاست که به کاربران اجازه می‌دهد در صفحات یا بخش‌های مختلف داخل یک وب‌سایت حرکت کنند. منوی ناوبری سایت معمولاً در بالا یا کنار یک صفحه وب قرار می‌گیرد.

■ **<section>**: برای تقسیم‌بندی محتوای صفحه به بخش‌های مختلف استفاده می‌شود (شکل۱۸).

```
<main>
    <section>
        <h2>درباره ما</h2>
        <p>ما یک فروشگاه آنلاین معتبر در زمینه عرضه کتاب‌های درسی هستیم.</p>
    </section>

    <section>
        <h2>خدمات ما</h2>
        <p>قیمت‌های مناسب، تخفیف‌های ویژه، ارسال سریع به سراسر کشور، تحویل رایگان درب منزل</p>
    </section>

    <section>
        <h2>تماس با ما</h2>
        <p>ایمیل: info@example.com</p>
    </section>
</main>
```

خروجی

درباره ما

ما یک فروشگاه آنلاین معتبر در زمینه عرضه کتاب‌های درسی هستیم.

خدمات ما

قیمت‌های مناسب، تخفیف‌های ویژه، ارسال سریع به سراسر کشور، تحویل رایگان درب منزل

تماس با ما

ایمیل: info@example.com

شکل۱۸

■ **<article>**: این برچسب برای نگهداری یک محتوای مستقل و کامل قابل استفاده است، مانند پست‌های وبلاگ (هر پست درون یک برچسب article) اخبار، (هر خبر درون یک برچسب article)، نظرات کاربران (هر نظر درون یک article مجزا) و کارت محصول (شامل تصویر، عنوان و قیمت).

```
<main>
    <section>
        <h2>درباره ما</h2>
        <p>ما یک فروشگاه آنلاین معتبر در زمینه عرضه کتاب‌های درسی هستیم.</p>
    </section>

    <section>
        <h2>خدمات ما</h2>
        <p>قیمت‌های مناسب، تخفیف‌های ویژه، ارسال سریع به سراسر کشور، تحویل رایگان درب منزل</p>
    </section>

    <section>
        <h2>تماس با ما</h2>
        <p>ایمیل: info@example.com</p>
    </section>
</main>
```

خروجی

درباره ما

ما یک فروشگاه آنلاین معتبر در زمینه عرضه کتاب‌های درسی هستیم.

خدمات ما

قیمت‌های مناسب، تخفیف‌های ویژه، ارسال سریع به سراسر کشور، تحویل رایگان درب منزل

تماس با ما

ایمیل: info@example.com

شکل۱۹

■ **<aside>**: برای نگهداری محتواهای جانبی که به محتوای اصلی صفحه ارتباط دارند، مانند پیوندها، نظرات و تبلیغات (شکل ۲۰).

```
<h2>کتاب‌های درسی</h2>
<p>
    در این بخش می‌توانید کتاب‌های درسی تمامی پایه‌ها و رشته‌ها را
    به‌صورت آنلاین مشاهده و خریداری کنید.
</p>

<aside>
    <h3>دسته‌بندی کتاب‌ها</h3>
    <ul>
        <li><a href="#">کتاب‌های ابتدایی</a></li>
        <li><a href="#">کتاب‌های متوسطه اول</a></li>
        <li><a href="#">کتاب‌های متوسطه دوم</a></li>
        <li><a href="#">کتاب‌های هنرستان</a></li>
    </ul>
</aside>
```

کتاب‌های درسی

در این بخش می‌توانید کتاب‌های درسی تمامی پایه‌ها و رشته‌ها را به صورت آنلاین مشاهده و خریداری کنید.

دسته‌بندی کتاب‌ها

- کتاب‌های ابتدایی
- کتاب‌های متوسطه اول
- کتاب‌های متوسطه دوم
- کتاب‌های هنرستان

شکل ۲۰

<footer>: برای تعریف بخش پایانی یک سند یا بخش استفاده می‌شود. برای تعیین ظاهر این عناصر از CSS استفاده می‌شود. در این قسمت اطلاعاتی مانند حق کپی‌رایت، نام نویسنده و طراح، لینک‌های تماس، سیاست‌های سایت و لینک شبکه‌های اجتماعی قرار می‌گیرد. (شکل ۲۱)

```
<footer>
    <p>کلیه حقوق مادی و معنوی این سایت متعلق به فروشگاه من است &copy</p>
</footer>
```

شکل ۲۱

■ عناصر غیرمعنایی

عناصر غیرمعنایی در HTML عناصری هستند که فقط برای گروه‌بندی و نگه‌داشتن محتوا استفاده می‌شوند و مفهوم خاصی دربارهٔ محتوای خود انتقال نمی‌دهند. دو نمونه مهم از این عناصر، توسط تگ‌های و ایجاد می‌شوند. این عناصر برای سازماندهی بخش‌های مختلف صفحه به‌کار می‌روند، اما چون معنای مشخصی ندارند، استفاده بی‌رویه از آنها می‌تواند درک صفحه را برای موتورهای جست‌وجو و ابزارهای دسترسی دشوار نماید.

ـ تگ یک عنصر block-level ایجاد می‌کند؛ به این معنی که محتوای این عنصر در یک خط جدید قرار می‌گیرد. این تگ برای ایجاد ساختارهای پیچیده در صفحات و تقسیم‌بندی محتوا به بخش‌های مختلف به کار می‌رود.

ـ تگ یک عنصر inline ایجاد می‌کند و محتوای آن در یک خط جدید قرار نمی‌گیرد. این تگ برای گروه‌بندی و استایل‌دهی بخش‌های کوچکی از متن در درون یک خط استفاده می‌شود.

ـ تگ‌های و جلوه بصری ندارند و برای تنظیمات ظاهری آنها از CSS استفاده می‌شود.

کار کارگاهی

به‌کارگیری عناصر معنایی و غیرمعنایی

فایل test.html را باز کنید و عناصر معنایی را مطابق تصویر زیر به‌کار ببرید:

```
<body>
    <header>
        <h1>مرجع نو</h1>
        <p>ارائه دهنده آخرین نوبت چاپ کتاب‌های درسی</p>
        <nav>
            <a href="./test.html">خانه</a>
            <a href="./products.html">محصولات</a>
            <a href="./services.html">خدمات ما</a>
            <a href="#">سبد خرید</a>
            <form action="action_page" method="post">
                <input type="text" name="search">
                <button type="submit">جستجو</button>
            </form>
        </nav>
    </header>
    <main>
        <section>
            <h2>محصولات پربازدید</h2>
            <div>
                <img src="./images/danesh_fanni_paye.jpg" alt="دانش فنی پایه">
                <img src="./images/negahdarie_system.jpg" alt="نگهداری سیستم های رایانه ای">
            </div>
        </section>
    </main>
    <footer>
        <p><b><i>&copy; کلیه حقوق مادی و معنوی این سایت متعلق به فروشگاه من است </i></b></p>
    </footer>
</body>
```

فایل را ذخیره کنید و نتیجه را در مرورگر مشاهده کنید.



توضیحات (Comments)

کامنت یا توضیحات در HTML ابزاری مفید برای مستندسازی و یادداشت‌برداری در کد HTML هستند. کامنت‌ها امکان اضافه کردن توضیحات، یادداشت‌ها یا اطلاعات اضافی را برای توسعه‌دهنده فراهم می‌کنند. توضیحات توسط مرورگر تفسیر نمی‌شوند؛ در نتیجه کاربران محتوای توضیحات را نمی‌بینند.

از کامنت‌ها برای خطایابی کد (Debugging) نیز استفاده می‌شود، به این صورت که با غیرفعال کردن موقت بخشی از کدها، می‌توان تأثیر آنها را بر روی اجرای برنامه بررسی نمود.

برای ایجاد کامنت در HTML از علامت <!-- --> استفاده می‌شود:

```
<!-- this is a comment -->
<p> This is a paragraph </p>
```

ارزشیابی شایستگی طراحی ساختار صفحات وب با html

استاندارد عملکرد کار: هنرجو بتواند با تحلیل نیازمندی‌ها، یک صفحه وب استاندارد **HTML**5 را با ساختار معنایی صحیح، سازمان‌دهی منطقی محتوا و بدون خطای ساختاری طراحی و پیاده‌سازی کند به‌گونه‌ای که در مرورگر به‌درستی نمایش داده شود.

ردیف	مراحل کار	شرایط انجام کار (ابزار،مواد، تجهیزات، زمان، مکان و ...)	ابزارهای ارزشیابی	استاندارد (شاخص‌ها/داوری/نمره‌دهی)	نتایج ممکن	نمره کسب شده
۱	تحلیل نیازمندی‌ها و طراحی اولیه	رایانه یا لپ‌تاپ مرورگر وب نرم‌افزار طراحی یا کاغذ و قلم ۲۵دقیقه فضای کارگاه یا کلاس مجهز به میز و نور مناسب	پرسش و پاسخ شفاهی درباره طراحی	ـ تفاوت بین وب جهان‌گستر (WWW) و اینترنت، و نقش مرورگرها را تشریح کند. ـ یک نقشه سایت ساده برای یک وب‌سایت فرضی (مثلاً وب‌سایت یک فروشگاه کوچک یا وبلاگ شخصی) طراحی کنند. ـ لیست نیازمندی‌های کلیدی (مثلا: نمایش محصولات، فرم تماس، درباره ما) برای همان وب‌سایت فرضی تهیه کنند. ـ توضیح مختصر در مورد اهمیت تعیین هدف پروژه در ابتدای طراحی. ـ انجام کنجکاوی‌ها	۳	
				تفاوت بین وب جهان‌گستر (WWW) و اینترنت، و نقش مرورگرها را بداند. ـ یک نقشه سایت ساده برای یک وب‌سایت فرضی (مثلا وب‌سایت یک فروشگاه کوچک یا وبلاگ شخصی) طراحی کنند. ـ لیست نیازمندی‌های کلیدی (مثلا: نمایش محصولات، فرم تماس، درباره ما) برای همان وب‌سایت فرضی تهیه کنند. ـ توضیح مختصر در مورد اهمیت تعیین هدف پروژه در ابتدای طراحی.	۲	
				ـ توضیح ناقص هریک از موارد شرح کار	۱	
۲	پیاده‌سازی ساختار پایه سند HTML ایجاد اولین صفحه	رایانه یا لپ‌تاپ ویرایشگر متن (VS Code, Notepad++) مرورگر وب برای تست صفحه فایل نمونه HTML ۱۵ دقیقه	آزمون کوتاه عملی برای ایجاد اسکلت HTML	ـ ایجاد یک فایل HTML پایه با استفاده از ویرایشگر متن (مانند Note-pad یا VS Code) که شامل تگ‌های اصلی `<!DOCTYPE html>`, `<html>`, `<head>` و `<body>` باشد. ـ اضافه کردن یک عنوان (`<title>`) به بخش `<head>` و یک متن ساده در بخش `<body>`. ـ نحوه اجرای فایل HTML در مرورگر را نشان دهند.ـ تشریح نقش هر یک از تگ‌های اصلی (`html`, `head`, `body`, `title`) در یک سند HTML.انجام کنجکاوی‌ها ـ انجام کار با روش‌های دیگر	۳	
				ـ ایجاد یک فایل HTML پایه با استفاده از ویرایشگر متن (مانند Notepad یا VS Code) که شامل تگ‌های اصلی `<!DOCTYPE html>`, `<html>`, `<head>` و `<body>` باشد. ـ اضافه کردن یک عنوان (`<title>`) به بخش `<head>` و یک متن ساده در بخش `<body>`. ـ نحوه اجرای فایل HTML در مرورگر را نشان دهند.	۲	
				انجام ناقص هریک از مراحل عملی ـ فقط توضیح نقش هر یک از تگ‌های اصلی (`html`, `head`, `body`, `title`) در یک سند HTML.	۱	
۳	عناصر اصلی محتوایی در HTML	همان ابزار مرحله قبل منابع نمونه برای محتوا و لینک‌ها مرورگر وب	پرسش و پاسخ عملی	ـ ایجاد یک صفحه HTML که شامل:ـ یک تیتر اصلی (`<h1>`) , تیتر فرعی `<h2>` تا `<h6>` و یک پاراگراف (`<p>`)ـ یک لیست نامرتب (`<ul>`) شامل ۳ـ ۴ موردـ یک لینک (`<a>`) که به یک وب‌سایت دیگر (مثلا oerp.ir) اشاره کند.ـ تفاوت بین تگ‌های `<strong>` و `<b>` و کاربرد هر کدام را توضیح دهند.ـ تفاوت بین لیست مرتب (`<ol>`) و نامرتب (`<ul>`) را شرح دهندـ انجام کنجکاوی‌ها ـ انجام کار با روش‌های دیگر	۳	
				ایجاد یک صفحه HTML که شامل:ـ ـ یک تیتر اصلی (`<h1>`) و یک پاراگراف (`<p>`)ـ یک لیست نامرتب (`<ul>`) شامل ۳ـ ۴ موردـ یک لینک (`<a>`) که به یک وب‌سایت دیگر (مثلا oerp.ir) اشاره کند. تفاوت بین تگ‌های `<strong>` و `<b>` و کاربرد هر کدام را توضیح دهند.ـ تفاوت بین لیست مرتب (`<ol>`) و نامرتب (`<ul>`) را شرح دهندـ	۲	
				ایجاد یک صفحه HTML که شامل: ـ یک تیتر اصلی (`<h1>`) و یک پاراگراف (`<p>`)ـ یک لیست نامرتب (`<ul>`) شامل ۳ـ۴ مورد	۱	

۴	استفاده از ساختارهای معنایی اصلی	رایانه با ویرایشگر و مرورگر منابع مستندات HTML5 ۲۰ دقیقه	بررسی فایل HTML با عناصر معنایی مشاهده عملکرد و سازمان‌دهی صحیح محتوا در مرورگر	۳	ـ عناصر معنایی (header, main, footer, section, article, nav) درست استفاده شده باشندـ صفحه استاندارد HTML5 داشته باشد. ـ انجام کنجکاوی‌ها ـ انجام کار با روش‌های دیگر	
				۲	عناصر معنایی (header, main, footer, section, article, nav) درست استفاده شده باشند۲صفحه استاندارد HTML5 داشته باشد.	
				۱	انجام ناقص هریک ازمراحل	
۵	ایجاد بخش‌بندی‌های تفصیلی و جانبی	همان ابزارهای مرحله قبل تصاویر و محتوای جانبی ۲۵ دقیقه	پرسش کتبی ـ آزمون عملی ـ مشاهده عملکرد هنرآموز	۳	ـ عناصر جانبی (aside-caption) به درستی اضافه شده باشندـ تصاویر همراه با توضیحات درست نمایش داده شوندـ ایجاد جدول با ۲ سطر و ۲ ستون ـ ایجاد فرم های ورود اطلاعات ـ انجام کنجکاوی‌ها ـ انجام کار با روش‌های دیگر	
				۲	عناصر جانبی (aside-caption) به‌درستی اضافه شده باشندـ تصاویر همراه با توضیحات درست نمایش داده شوند ـ ایجاد فرم‌های ورود اطلاعات	
				۱	انجام ناقص هریک از مراحل	
۶	تکمیل ساختار با عناصر غیرمعنایی و مستندسازی	همان ابزارهای قبلی منابع راهنما برای مستندسازی کد ۱۵ دقیقه	مشاهده مستندسازی و کامنت‌های نوشته‌شده اجرای صفحه و صحت نمایش نهایی	۳	ـ استفاده مناسب از div و span برای ساختاردهیـ ـ کامنت‌ها و مستندسازی کد کامل باشدـ مزایای استفاده از تگ‌های معنایی (مانند article, section) را برای سئو و دسترسی‌پذیری توضیح دهند. ـ انجام کنجکاوی‌ها ـ انجام کار با روش‌های دیگر	
				۲	ـ استفاده مناسب از div و span برای ساختاردهی ـ کامنت‌ها و مستندسازی کد کامل باشد.	
				۱	انجام ناقص هریک از مراحل ـ مزایای استفاده از تگ‌های معنایی (مانند article, section) را برای سئو و دسترسی‌پذیری توضیح دهند.	
شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش:				۲	احراز همه شایستگی‌های غیرفنی	
				۱	عدم احراز هریک از شایستگی‌های غیرفنی	
ارزشیابی کار (شایستگی انجام کار)						بلی ☐ خیر ☐

توضیحات:

۱ـ معیار شایستگی انجام کار:

ـ کسب حداقل نمره ۲ از مراحل ۱، ۲، ۳ و ۴ (مراحل بحرانی)

ـ کسب حداقل نمره ۲ از بخش شایستگی‌های غیرفنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش

ـ کسب حداقل میانگین ۲ از مراحل کار

۲ـ تعریف سطوح شایستگی: صرف‌نظر از اینکه یک تکلیف کاری در چه سطحی از صلاحیت حرفه‌ای انجام می‌شود، در هر محیط کاری ممکن است انجام هر کار با کیفیت مشخصی مورد انتظار باشد. سطح شایستگی انجام کار، معیار اساسی ارزشیابی است.

مهارت (سطح سه): ماهر و قادر به آموزش و هدایت دیگران، توانایی برنامه‌ریزی و تحلیل، پاسخ‌گویی در برابر کارهای خود، سروکار داشتن با سطح وسیعی از کارها و فعالیت‌ها

تسلط (سطح چهار): خبرگی در انجام کار و آموزش دیگران، ایجاد نوآوری، سازگاری، عیب‌یابی، هدایت و راهنمایی دیگران.

واحد یادگیری ۲
طراحی ظاهر صفحات وب با CSS

آیا تابه‌حال اندیشیده‌اید

- چگونه می‌توان رنگ متن و زمینه عناصر صفحه وب را مشخص نمود؟
- چگونه می‌توان چیدمان عناصر صفحه را با فواصل مناسب انجام داد؟
- آیا می‌توان بدون تغییر در ساختار HTML، ظاهر یک سایت را دگرگون نمود؟
- چگونه می‌توان چیدمان و استایل عناصر صفحه را در دستگاه‌های مختلف تغییر داد؟
- جذابیت بصری صفحه وب چه تأثیری روی تجربه کاربری دارد؟

در پایان از هنرجو انتظار می‌رود

۱ شیوه‌های مختلف تعریف CSS و اولویت اجرای آنها را بداند.

۲ توانایی تنظیم استایل عناصر متنی شامل نوع قلم، اندازه، رنگ، وزن و تراز آنها را داشته باشد.

۳ توانایی تنظیم ویژگی‌های ظاهری عناصر شامل رنگ، پس‌زمینه، شفافیت، خط دور و سایه را داشته باشد.

۴ انواع انتخاب‌گرهای ساده، ترکیبی، شبه‌کلاس و ویژگی را بشناسد و از آنها برای انتخاب دقیق عناصر به‌شکل بهینه استفاده نماید.

۵ با مفهوم مدل جعبه‌ای آشنا باشد و توانایی تنظیم فواصل داخلی (margin) و خارجی (padding) عناصر را داشته باشد.

۶ کاربرد ویژگی Display را بداند و از آن برای کنترل نحوه نمایش عناصر استفاده نماید.

۷ توانایی چیدمان واکنش‌گرای عناصر صفحه را توسط قابلیت‌های Flex و Grid داشته باشد.

۸ کدهای CSS را به شکل خوانا و با رعایت تورفتگی به‌کار ببرد و برای هر یک از استایل‌های صفحه از کامنت‌های مناسب و قابل درک استفاده نماید.

استاندارد عملکرد

با استفاده از قابلیت‌های CSS، ظاهر و چیدمان عناصر صفحه وب را مطابق با اصول و استانداردهای نوین طراحی وب، پیاده‌سازی و تنظیم نماید.

زیباسازی صفحه با CSS

در واحدکار قبل نحوه طراحی ساختار صفحات وب با استفاده از برچسب‌های HTML آموزش داده شد. صفحاتی که صرفاً توسط HTML ایجاد شده باشند، دارای متون ساده، پس زمینه سفید، فواصل و چیدمان پیش‌فرض می‌باشند و در نتیجه فاقد جذابیت بصری هستند. برای ایجاد ظاهری زیبا و کاربرپسند استفاده از CSS جهت استایل‌دهی به عناصر ضروری است. این ابزار به توسعه‌دهندگان وب امکان می‌دهد که ظاهر و چیدمان عناصر صفحه را به سادگی و با انعطاف بالا مدیریت کنند.

CSS مخفف عبارت Cascading Style Sheets و زبانی برای استایل‌دهی و کنترل ظاهر صفحات وب است. این زبان به توسعه‌دهندگان وب امکان می‌دهد که ظاهر و چیدمان عناصر را کاملاً جدا از ساختار اصلی صفحه مدیریت کنند. این موضوع باعث آسان‌تر شدن نگهداری و به‌روزرسانی سایت می‌شود. با استفاده از CSS می‌توان استایل‌های متنوعی را برای عناصر مختلف وب‌سایت تنظیم نمود؛ مانند تعیین رنگ متن و زمینه عناصر، ایجاد سایه، خط دور، تغییرات ظاهری در هنگام قرارگیری ماوس روی عناصر و ...



شکل ۲۲

ساختار CSS

CSS به‌شکل مجموعه‌ای از قوانین است که توسط انتخاب‌گرها (selectors) و ویژگی‌ها (properties) ایجاد می‌شوند. هر قانون معمولاً به صورت زیر نوشته می‌شود:

```
selector {
   property: value;
}
```

عبارت Selector به معنای انتخاب‌گر است و برای انتخاب عناصر صفحه استفاده می‌شود. به عنوان مثال برای تعیین استایل عناصر پاراگراف، از انتخاب‌گر p و برای انتخاب همه عناصر صفحه از علامت * استفاده می‌شود. عبارت Property به ویژگی‌های قابل مقداردهی در CSS گفته می‌شود، مانند color، font-size و border. عبارت Value به مقدار ویژگی عنصر اشاره دارد.

مثال

در مثال زیر رنگ زمینه عنصر h1 با مقدار orange تنظیم شده است.

```
h1 {
   background-color: orange;
}
```

نحوه به‌کارگیری CSS در صفحات

برای ایجاد و به‌کارگیری استایل‌های سفارشی روی عناصر صفحه از سه روش به شرح زیر استفاده می‌شود:

۱ CSS درون خطی(Inline): استایل عناصر درون تگ‌های HTML نوشته می‌شود.

مثال

در مثال زیر رنگ متن و پس زمینه عنصر h1 به‌صورت درون خطی تعیین شده است:

```
<h1 style="color:blue; background-color:orange">
```



۲ CSS داخلی (Internal):

- استایل عناصر درون تگ <style> در بخش <head> نوشته می‌شود.
- در سبک درون خطی در صورتی‌که تعداد ویژگی‌های اختصاص داده شده به عنصر زیاد باشد، کد عنصر شلوغ و ناخوانا می‌شود. بنابراین از CSS داخلی یا خارجی برای مدیریت بهتر استایل‌ها استفاده می‌شود.

مثال

مثال زیر نمونه‌ای از CSS داخلی برای تعیین ویژگی‌های عنصر h1 است:

```
<head>
   <style>
      h1 {
         color: blue;
         background-color: pink;
      }
   </style>
</head>
```

۳ CSS خارجی (External):

ـ در روش CSS خارجی، استایل عناصر، درون فایلی با پسوند CSS قرار می‌گیرد. محتوای این فایل نیازی به تگ <style> ندارد و صرفاً سبک عناصر درون آن تعیین می‌شود. فایل CSS باید توسط تگ <link> به صفحه وب الحاق شود تا قوانین این فایل روی عناصر صفحه وب اعمال شود. الحاق فایل CSS به فایل HTML در بخش <head> انجام می‌شود:

```
<head>
   <link rel="stylesheet" href="./styles.css">
</head>
```

■ استفاده از فایل‌های CSS خارجی، بهترین روش استایل‌دهی در پروژه‌های متوسط و بزرگ است. مهم‌ترین مزایای روش خارجی عبارت‌اند از:

۱ فایل CSS قابلیت استفاده مجدد دارد، یعنی می‌تواند به صفحات مختلف سایت الحاق شود.

۲ با جدا شدن کدهای CSS و HTML، نگهداری، به‌روزرسانی واشکال‌زدایی (Debug) استایل و کدهای HTML آسان‌تر می‌شود.

۳ مرورگرها می‌توانند فایل‌های CSS خارجی را در حافظه پنهان (cache) خود ذخیره کنند، این موضوع به بارگذاری سریع‌تر صفحات در دفعات بعدی ورود به سایت کمک می‌کند.

■ عیب این روش این است که فایل CSS باید به‌صورت مجزا بارگذاری شود و این کار ممکن است سرعت بارگذاری سبک عناصر صفحه وب را در اولین بارگذاری پایین بیاورد.

برای تمرین‌های کوچک در حد سبک‌دهی به یک عنصر می‌توان از CSS خطی استفاده نمود. سبک داخلی برای پروژه‌های کوچک و سبک خارجی برای پروژه‌های متوسط، بزرگ و پیچیده مناسب است.

اولویت اعمال انواع استایل‌ها

چنانچه سبک‌های مختلفی برای یک عنصر نوشته شده باشد، اولویت اعمال سبک‌ها به ترتیب درون خطی، داخلی و خارجی خواهد بود. اگر عنصری دارای هیچ یک از این سبک‌ها نباشد، از سبک عنصر والد خود استفاده می‌کند.

کار کارگاهی

بررسی اولویت اعمال سبک روی عناصر صفحه

ـ یک صفحه وب با نام test-style.html ایجاد کنید.

ـ یک فایل CSS با نام styles.css ایجاد کنید.

ـ محتوای فایل‌های HTML و CSS را مطابق تصویر مقابل تنظیم کنید.

ـ فایل‌ها را ذخیره و نتیجه را در مرورگر بررسی کنید.

■ در هر سه سبک رنگ متن مشخص شده است، اما رنگ سبک درون‌خطی (سبز) روی متن اعمال می‌شود.

■ رنگ زمینه فقط در سبک داخلی تعیین شده است، بنابراین رنگ زمینه زرد به عنصر h1 اعمال می‌شود.

■ پهنای عنصر فقط در سبک خارجی مشخص شده است، در نتیجه پهنای عنصر h1 برابر ۲۰۰ پیکسل خواهد بود.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="stylesheet" href="style.css">
    <style>
        h1{
            color: blue;
            background: yellow;
        }
    </style>
</head>
<body>
    <h1 style="color: green"> Heading text </h1>
</body>
</html>
```

```
style.css - Notepad
File  Edit  Format  View  Help
h1{
    color:red
    width:200px;
}
```

خروجی

Heading text

توضیحات (Comment) در CSS

ـ برای افزودن توضیحات در CSS به صورت تک‌خطی یا چندخطی از علائم /* */ از شکل ۲۳ استفاده می‌شود.

```
/* this is a comment */
```

شکل۲۳

نحوه قالب‌بندی متن در CSS

CSS دارای مجموعه‌ای از ویژگی‌ها برای قالب‌بندی متون است. این ویژگی‌ها امکان تعیین فونت، اندازه فونت، ارتفاع خطوط، تزئینات متن و ... را فراهم می‌کنند. ویژگی‌های عناصر متنی به شرح زیر است:

- **font-family و font-size**: تعیین نوع و اندازه فونت

- **text-align**: تعیین تراز متن شامل مقادیر left، right، center و justify

- **text-shadow**: ایجاد سایه با رنگ و اندازه دلخواه. ساختار کلی این ویژگی به شکل زیر است:

text-shadow: offset-x offset-y blur-radius color;

شکل۲۴

مقادیر این ویژگی به ترتیب مقدار جابه‌جایی افقی و عمودی سایه، میزان محو شدگی و رنگ سایه است.

- **text-transform**: تغییر شکل حروف به صورت بزرگ،کوچک و... شامل مقادیر lowercase، upper-cas، capitalize none،...

- **direction**: تعیین جهت نمایش متن، شامل مقادیر ltr و rtl برای چپ به راست و راست به چپ

- **line-height**: تعیین فاصلهٔ عمودی بین خطوط متنی. این ویژگی فاصلهٔ عمودی خط متن را تعیین می‌کند. مقدار این ویژگی شامل ارتفاع حروف به‌علاوه فضای خالی قبل و بعد آنهاست. شامل ارتفاع حروف (فونت) به‌اضافه فضای خالی قبل و بعد از آنهاست.

- **text-decoration**: تعیین تزیینات متن، شامل مقادیر none، underline، overline، line-through. ویژگی text-decoration با مقدار none موجب حذف زیرخط لینک می‌شود.

- **vertical-align**: تراز متن در راستای عمودی، شامل مقادیر top، bottom، baseline و...

فعالیت

ویژگی‌های قالب بندی متن را برای عناصر فایل test.html به کار بگیرید و نتیجه را در مرورگر مشاهده کنید.

طراحی ظاهر عناصر (رنگ ها، پس زمینه‌ها، شفافیت و خط دور)

CSS دارای ویژگی‌های مختلفی در زمینه تنظیم ظاهر عناصر است. این ویژگی‌ها برای تعیین رنگ متن، رنگ زمینه، تصویر پس‌زمینه،کنترل تصویر پس‌زمینه، شفافیت و خط دور عناصر به‌کار می‌روند. در ادامه به برخی از این ویژگی‌ها اشاره می‌شود.

- **color و background-color**: برای تعیین رنگ متن و زمینه عناصر به‌کار می‌رود. برای هر دو ویژگی فوق می‌توان از نام رنگ، کد هگزادسیمال رنگ یا کد RGB استفاده نمود.

مثال

در مثال روبه‌رو از هر سه روش تعیین مقدار رنگ استفاده شده است.

```
color: blue;
background-color: #0000FF;
background-color: rgb(0,0,255);
```

- **opacity**: شفافیت عناصر توسط این ویژگی تعیین می‌شود. مقدار opacity با عددی بین ۰ و ۱ تعیین می‌شود. مقادیر پایین این ویژگی باعث کاهش شفافیت عنصر و مقادیر نزدیک ۱ باعث افزایش شفافیت عنصر می‌شود.

- **background-image**: برای تعیین تصویر پس‌زمینه به‌کار می‌رود. برای مقداردهی به این ویژگی ابتدا عبارت url و سپس نام تصویر داخل پرانتز نوشته می‌شود.

مثال

مانند مثال روبه‌رو:

```
background-image:url(./images/background.jpg);
```

- **background-repeat**: تصاویری که پهنای صفحه را پوشش ندهند، تکرار می‌شوند. برای تنظیم نحوه تکرار یا عدم تکرار تصویر از این ویژگی استفاده می‌شود که دارای مقادیر مختلفی مانند no-repeat، repeat-x، repeat-y و ... است.

- **background-size**: اندازه تصویر پس زمینه را تعیین می‌کند. این ویژگی مشخص می‌کند که تصویر پس‌زمینه با چه ابعادی داخل عنصر نمایش داده شود. برای پوشش پهنای صفحه، بدون تکرار تصویر، از این ویژگی با مقدار cover استفاده می‌شود.

- **background-position**: برای تنظیم محل نمایش تصویر پس‌زمینه به‌کار می‌رود و دارای مقادیر مختلفی مانند center، left، right، left top، right bottom و غیره است.

■ **background-attachment**: اگر تصویر زمینه بزرگ‌تر از محدوده نمایش باشد، می‌توان با اسکرول کردن، بخش‌های پایین‌تر تصویر را به نمایش گذاشت. برای ثابت نگه داشتن تصویر در هنگام اسکرول صفحه، از این ویژگی با مقدار fixed استفاده می‌شود.

■ **border**: عناصر صفحه می‌توانند همراه با یک خط دور نمایش داده شوند. ایجاد خط دور توسط ویژگی border صورت می‌گیرد. مقادیر این ویژگی به ترتیب ضخامت خط دور، نوع خط دور و رنگ آن را تعیین می‌کنند.

border: 1px solid gray;

■ **border-radius**: برای گرد شدن گوشه‌ها از این ویژگی استفاده می‌شود و مقدار آن می‌تواند برحسب پیکسل یا درصد باشد:

border-radius:10px;

کار کارگاهی

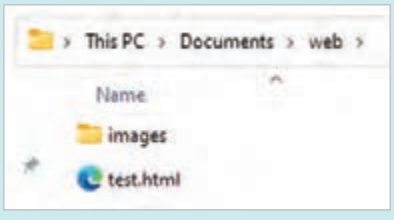
طراحی ظاهر عناصر (رنگ‌ها، پس‌زمینه‌ها، شفافیت و خط دور)

در فایل test.html کد HTML زیر را بنویسید و نتیجه را در مرورگر مشاهده کنید. سپس استایل‌ها را به‌صورت مرحله به مرحله اضافه کنید تا تأثیر هر یک را بر عناصر صفحه درک کنید.

```
<style>
    header{
        border:2px solid #b2b2b2;
    }
    h1{
        color: blue;
    }
    p{
        font-weight: bold;
    }
    nav{
        font-size:medium;
        font-family: Arial, Helvetica, sans-serif;
    }
    a{
        text-decoration: none;
        color: green;
    }
    div{
        background-image: url(./images/background.jpg);
    }
    footer{
        background: gray;
        font-size: 12px;
        color: whitesmoke;
        line-height:35px;
    }
    body{
        text-align:center;
    }
</style>
```

```
<body>
<header>
    <h1> مرجع تو </h1>
    <p> ارائه دهنده آخرین نوبت چاپ کتب درسی </p>
    <nav>
        <a href="#">خانه</a>
        <a href="#">محصولات</a>
        <a href="#"> خدمات ما </a>
        <a href="#"> سبد خرید </a>
        <form action="" method="post">
            <input type="text" name="search">
            <button type="submit">جستجو</button>
        </form>
    </nav>
</header>
<main>
  <section>
    <h2>محصولات پربازدید</h2>
    <div>
      <img src="./images/danesh-fani-paye.jpg">
      <img src="./images/negahdari-system.jpg">
    </div>
  </section>

</main>
<footer>
    &copy
    کلیه حقوق مادی و معنوی این وب سایت
    متعلق به فروشگاه من است
</footer>
</body>
```





در نرم‌افزار Paint یک فایل جدید با نام background.jpg ایجاد کنید و آن را درون پوشه images در مسیر فایل test.html ذخیره کنید. ابعاد آن را ۵۰×۵۰ پیکسل تنظیم کنید. درون آن دو دایره توخالی با رنگ ملایم ایجاد کنید (مانند تصویر روبه‌رو) و فایل را ذخیره کنید. این تصویر به پس زمینه عنصر div اعمال می‌گردد.

خروجی

انتخاب‌گرهای CSS

عنصر یا مجموعه عناصری که استایل CSS برای آنها نوشته می‌شود، انتخاب‌گر نامیده می‌شوند. CSS دارای انواع مختلفی از انتخاب‌گرها شامل انتخاب‌گرهای ساده، ترکیبی، شبه‌کلاس، شبه عنصر و ویژگی است که در این بخش به آنها پرداخته می‌شود.

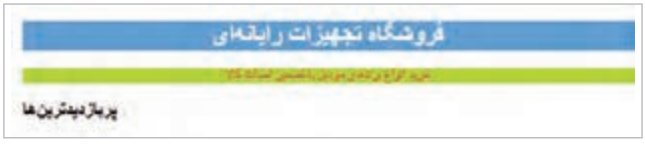
انتخاب‌گرهای ساده (Simple):

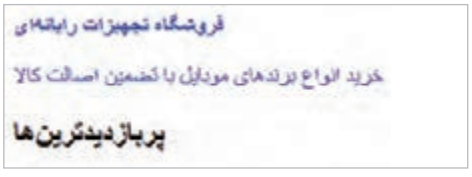
این انتخاب‌گرها از اسامی تگ‌ها، شناسه‌ها، کلاس‌ها و علامت * برای انتخاب عناصر استفاده می‌کنند. انواع انتخاب‌گرهای ساده در جدول ۸ آمده است:

جدول ۸ــ انواع انتخاب‌گرهای ساده

نوع انتخاب‌گر	عملکرد
عنصر (Element)	انتخاب عناصر صفحه بر اساس نام تگ، مانند h2، p، table، body و ...

```
<head>
<style>
    h1{
        background-color: rgb(95, 158, 209);
        color:#fff;
        text-align: center;
        }
    p{
        background-color: greenyellow;
        color: crimson;
        text-align: center;
    }
</style>
</head>
<body>
    <h1> فروشگاه تجهیزات رایانه ای </h1>
    <p>خرید انواع برندهای موبایل با تضمین اصالت کالا</p>
    <h2>پربازدیدترین ها</h2>
</body>
```



شناسه (Id)	**انتخاب عناصر صفحه براساس شناسه آنها:** انتخاب‌گرهای شناسه با علامت # شروع می‌شوند. این انتخاب‌گرها عناصر را با استفاده از شناسه (id) آنها انتخاب می‌کنند. به عنوان مثال برای انتخاب عنصری با شناسه test از عبارت #test استفاده می‌شود. مقدار شناسه یک عنصر درون ویژگی id مشخص می‌شود. هر شناسه فقط مربوط به یک عنصر است و نباید برای سایر عناصر صفحه تکرار گردد. در مثال زیر انتخاب‌گر test متعلق به عنصر <h2> است.  
کلاس (Class)	**انتخاب عناصر صفحه براساس نام کلاس:** انتخاب‌گرهای کلاس با علامت . شروع می‌شوند. این انتخاب‌گرها عناصر را با استفاده از نام کلاس (class) آنها انتخاب می‌کنند. ویژگی انتخاب‌گر کلاس این است که می‌تواند مجموعه‌ای از عناصر با نام کلاس یکسان را انتخاب نماید. به‌طور مثال انتخاب‌گر class1. تمام عناصر صفحه با ویژگی class="class1" را انتخاب می‌کند. در مثال زیر دو عنصر <h1> و <p> دارای نام کلاس یکسان هستند، پس رنگ متن هر دو آبی و اندازه فونت آنها ۱۸ پیکسل می‌گردد.  

گروهی(Group)	**انتخاب گروهی از عناصر**: انتخاب‌گرهای گروهی مجموعه‌ای از قوانین CSS را به چند عنصر اختصاص می‌دهند. در یک انتخابگر گروهی، نام عناصر به همراه کاما نوشته می‌شود. در مثال زیر عبارت h1, h2,p یک انتخاب‌گر گروهی محسوب می‌شود. `<head>` `<style>` `h1,h2,p{` `color: blue;` `text-align: center;` `}` `</style>` `</head>` `<body>` `<h1> فروشگاه تجهیزات رایانه ای </h1>` `<p>خرید انواع برندهای موبایل با تضمین اصالت کالا</p>` `<h2>پربازدیدترین ها</h2>` `</body>` قوانین بالا به تگ‌های <h2>، <h1> و <p> اعمال می‌شود. درنتیجه همهٔ آنها، آبی و وسط‌چین می‌شوند. فروشگاه تجهیزات رایانه‌ای خرید انواع برندهای موبایل با تضمین اصالت کالا پربازدیدترین‌ها
سراسری (universal)	**انتخاب تمام عناصر صفحه توسط علامت ***: در مثال زیر، مقدار حاشیه خارجی (margin) و فاصله داخلی (padding) در تمام عناصر صفحه برابر با صفر تنظیم شده است. `* {` `margin: 0;` `padding: 0;` `}`

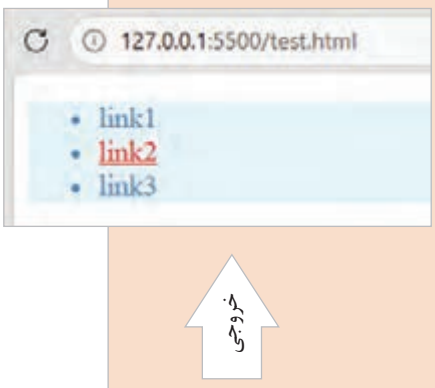
انتخاب‌گر ترکیبی(Combinator):

انتخاب‌گرهای ترکیبی در CSS برای تعریف رابطه ساختاری و موقعیتی میان دو یا چند انتخاب‌گر به‌کار می‌روند. مانند انتخاب‌گر فرزند با نماد > و انتخاب‌گر نوادگان با استفاده از فاصله، عناصری که به‌صورت تودرتو نوشته می‌شوند، رابطه والد و فرزندی پیدا می‌کنند. به‌عنوان مثال عنصر li فرزند عنصر ul است. درنتیجه برای انتخاب عناصر li عبارت ul li یا ul > li استفاده می‌شود. استفاده از علامت > باعث می‌شود که محدودهٔ تأثیرگذاری به فرزندان مستقیم یک عنصر کاهش یابد.

مثال

تصویر زیر مثالی را درباره به‌کارگیری انتخاب‌گر ترکیبی نمایش می‌دهد.

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
    <style>
        .top-menu ul{
            background: lightcyan;
        }
        ul li{
            color: steelblue;
        }
        ul li a{
            color: crimson;
        }
    </style>
</head>
<body>
<nav class="top-menu">
    <ul>
        <li> link1 </li>
        <li> <a href=""> link2 </a> </li>
        <li> link3 </li>
    </ul>
</nav>
</body>
</html>
```



انتخاب‌گرهای شبه‌کلاس (Pseudo-class Selectors)

انتخاب‌گرهای شبه‌کلاس امکان تعیین استایل عناصر را براساس وضعیت، موقعیت یا ویژگی‌های پویای آنها فراهم می‌کند. برخی از این انتخاب‌گرها وابسته به عملکرد کاربر فعال می‌شوند. شبه‌کلاس‌ها با علامت : بعد از نام عنصر نوشته می‌شوند. انتخاب‌گرهای شبه‌کلاس به شرح زیر هستند:

- **hover**: زمانی فعال می‌شود که کاربر اشاره‌گر ماوس خود را بر روی یک عنصر ببرد.
- **focus**: این رویداد زمانی اتفاق می‌افتد که یک عنصر (معمولاً عناصر فرم) فعال شود و کار امکان تعامل با آن را داشته باشد.
- **active**: زمانی فعال می‌شود که یک عنصر در حالت فعال قرار گیرد. مانند زمانی که کاربر روی یک دکمه کلیک می‌کند.
- **nth-child()**: امکان انتخاب فرزندان یک عنصر را بر اساس موقعیت آنها فراهم می‌کند.
- **first-child ()**: امکان انتخاب اولین فرزند یک عنصر را برای انتخاب اولین فرزند یک عنصر به‌کار می‌رود.
- **last-child()**: برای انتخاب آخرین فرزند یک عنصر به‌کار می‌رود.

انتخاب‌گر ویژگی (Attribute selector)

انتخاب‌گر ویژگی، انتخاب عناصر را بر اساس ویژگی‌های آنها انجام می‌دهد. به‌طور مثال برای انتخاب عناصر متنی فرم از انتخاب‌گر input[type="text"] و برای انتخاب عناصری که دارای ویژگی href باشند از انتخاب‌گر [href] استفاده می‌شود.

کار کارگاهی

کار با انواع انتخاب‌گرهای CSS (طراحی منو)

یک فایل با نام index.html ایجاد کنید و درون آن کدهای HTML زیر را بنویسید. فایل را ذخیره کنید و نتیجه را در مرورگر مشاهده کنید. سپس سبک عناصر را به‌صورت تدریجی به بخش head اضافه کنید و نتیجه هر سبک را بررسی کنید.

```
<body>
  <header>
    <nav class="top-menu">
      <ul>
        <li><a href="#">باشگاه مشتریان</a></li>
        <li><a href="#">وبلاگ</a></li>
        <li><a href="#">پیشنهاد ملایی</a></li>
        <li><a href="#">آموزش خرید</a></li>
        <li><a href="#">پیگیری سفارش</a></li>
      </ul>
    </nav>
  </header>
</body>
```

```
<style>
  .top-menu{
    background: rgb(235, 244, 248);
  }
  .top-menu ul{
    list-style: none;
  }
  .top-menu ul li a{
    text-decoration: none;
    line-height: 30px;
  }
  .top-menu a:hover{
    color: blue;
  }
  .top-menu ul li:nth-child(3) a{
    color: orange;
  }
  .top-menu ul li:first-child a:hover{
    color: coral;
  }
  [href]{
    color: purple;
  }
</style>
```

خروجی

باشگاه مشتریان
وبلاگ
پیشنهاد ملایی
آموزش خرید
پیگیری سفارش

فعالیت

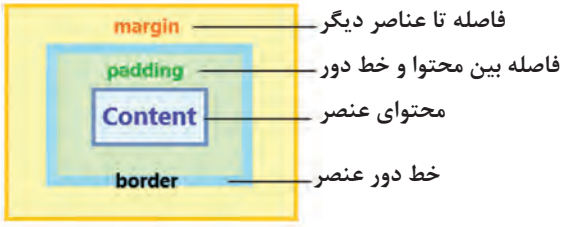
۱ در مقابل هر انتخاب‌گر، نوع آن را به‌صورت comment بنویسید. محتوای کامنت را درون علامت */ */ قرار دهید.

۲ ویژگی list-style با مقدار none چه نتیجه‌ای دارد؟

مدل جعبه‌ای (Box Model)

مدل جعبه‌ای (Box Model) در CSS برای طراحی و چیدمان عناصر در صفحات وب استفاده می‌شود. هر عنصر HTML به‌عنوان یک «جعبه» در نظر گرفته می‌شود که شامل چهار قسمت محتوا (content)، فضای اضافی داخل عنصر (padding)، خط دور (border) و فاصله عنصر تا عناصر مجاور (margin) است. در مثال زیر این تنظیمات قابل مشاهده و بررسی است: (شکل۲۵) .

ویژگی‌های width و height صرفاً پهنا و ارتفاع ناحیه محتوا (content) را مشخص می‌کنند. بنابراین مقداردهی به ویژگی‌های padding، margin و border باعث افزایش ابعاد نهایی جعبه می‌شود. این مقادیر باید در محاسبات ابعاد جعبه لحاظ شود. برای جلوگیری از تأثیر دو ویژگی border و padding روی پهنا و ارتفاع عنصر از ویژگی box-sizing با مقدار border-box استفاده می‌شود. با انجام این کار، مجموع ابعاد محتوا، padding و border در راستای افقی و عمودی برابر با مقادیر width و height خواهد بود.



شکل۲۴

کار کارگاهی

بررسی عملکرد مدل جعبه‌ای

فایلی با نام box-model.html ایجاد کنید و دو عنصر و <img> را به همراه سبک آنها درون فایل وارد کنید:

```
<head>
  <style>
    .product-item{
    width:max-content;
    height: max-content;
    padding:20px 10px;
    margin:20px auto;
    border:2px solid lightblue;
    background:rgb(246, 249, 251);
    }
    .product-item img{
      width: 150px;
      border:1px solid lightgray;
      margin:0 10px;
    }
  </style>
</head>
<body>
<div class="product-item">
  <img src="./images/physic.jpg" alt="کتاب فیزیک">
</div>
<div class="product-item">
  <img src="./images/chemistry.jpg" alt="کتاب شیمی">
</div>
</body>
```





در فایل index.html (کارگاه قبل) ویژگی‌های مارجین و پدینگ عنصر ul را برابر صفر قرار دهید (margin:0;padding:0) و برای عنصر header مقداری پدینگ تنظیم کنید (padding:10px20.px) سپس تغییرات منو را در مرورگر بررسی کنید.

مدل نمایش (Display Model)

مدل نمایش در CSS به نحوه نمایش و رفتار عناصر در صفحه وب اشاره دارد. این مدل تعیین می‌کند که عناصر چگونه در کنار یکدیگر قرار بگیرند و چگونه با هم تعامل کنند. مدل نمایش توسط ویژگی display با مقادیر مختلف تعیین می‌شود. در ادامه به شرح مقادیر این ویژگی پرداخته می‌شود:

block: عناصری با ویژگی display: block به صورت بلوکی نمایش داده می‌شوند و فضای کامل خط را اشغال می‌کنند. این عناصر به طور پیش‌فرض در یک خط جدید شروع می‌شوند و امکان تعیین width و height برای آنها فراهم است.

Inline: عناصر inline مانند کلمات رفتار می‌کنند، یعنی عناصر پشت سر هم قرار می‌گیرند و خط جدید ایجاد نمی‌کنند. این عناصر تنها به‌اندازهٔ محتوای خود فضا اشغال می‌کنند؛ در نتیجه تعیین width و height برای آنها بی اثر است. به‌علاوه اختصاص ویژگی‌های margin برای قسمت‌های بالا و پایین عنصر اثرگذار نخواهد بود.

Inline-block: این حالت ترکیبی از block و inline است. عناصر inline-block می‌توانند به صورت بلوکی رفتار کنند، اما درون خط جاری قرار می‌گیرند. این عناصر می‌توانند پهنا و ارتفاع داشته باشند و به اندازه محتوای خود فضا اشغال می‌کنند.

جدول ۹ـ مقایسه عناصر Block و Inline در CSS

ویژگی	عناصر Block	عناصر Inline
شروع در سطر جدید	همیشه در سطر جدید شروع می‌شوند.	در همان سطر ادامه پیدا می‌کنند.
پهنا (width)	حتی با تعیین پهنا، کل سطر را به خود اختصاص می‌دهد.	فقط به اندازهٔ محتوای عنصر است.
ارتفاع (height)	می‌توان تنظیم کرد.	معمولاً فقط به اندازهٔ محتوا است.
نمونه عناصر	, <p>, <h1>, <section>	, <a>, <b>, <i>
رفتار در کنار هم	هرکدام در خط جدا قرار می‌گیرند.	چند عنصر کنار هم قرار می‌گیرند.

کار کارگاهی

به‌کارگیری ویژگی display با مقادیر block ، Inline و Inline-block .

```
<!DOCTYPE html>
<html lang="fa" dir="rtl">
<head>
    <meta charset="UTF-8">
    <title>Diplay Examples</title>
    <style>
        body{
            font-family:tahoma;
        }
        .box{
            background: #77e9f8;
            padding:10px;
            margin:5px;
        }
        .block{
            display: block;
        }
        .inline{
            display: inline;
        }
        .inline-block{
            display: inline-block;
            width:120px;
            height:40px;
        }
    </style>
</head>
```

```
<body>
    <h2> Display: block </h2>
    <span class="box block"> باکس شماره ۱</span>
    <span class="box block"> باکس شماره ۲</span>
    <p>هر باکس در یک سطر مجزا قرار می‌گیرد و پهنای هر یک معادل پهنای کل صفحه است.</p>

    <h2> Display: inline </h2>
    <span class="box inline"> آیتم شماره ۱</span>
    <span class="box inline"> آیتم شماره ۲</span>
    <p>همانند کلمات در کنار یکدیگر قرار می‌گیرند، امکان تعیین پهنا و ارتفاع وجود ندارد.</p>

    <h2> Display: inline-block </h2>
    <span class="box inline-block"> باکس A </span>
    <span class="box inline-block"> باکس B </span>
    <p>باکس‌ها در کنار یکدیگر قرار می‌گیرند، امکان تعیین پهنا و ارتفاع موجود است.</p>
</body>
</html>
```

Display: block

باکس شماره 1

باکس شماره 2

هر باکس در یک سطر مجزا قرار می‌گیرد و پهنای هر یک معادل پهنای کل صفحه است.

Display: inline

ایتم شماره 1 ایتم شماره 2

همانند کلمات در کنار یکدیگر قرار می‌گیرند، امکان تعیین پهنا و ارتفاع وجود ندارد.

Display: inline-block

باکس A باکس B

باکس‌ها در کنار یکدیگر قرار می‌گیرند، امکان تعیین پهنا و ارتفاع موجود است.

کار کارگاهی

به‌کارگیری ویژگی display در طراحی فرم ورود به سایت

فایل login.html را در پوشه web برای ورود کاربر به سایت ایجاد کنید.

```
<!DOCTYPE html>
<html lang="fa" dir="rtl">
<head>
    <meta charset="UTF-8">
    <title> login form </title>
    <style>
        .login-box{
            max-width: 350px;
            padding: 25px;
            margin:50px auto;
            border-radius: 10px;
            box-shadow:0 0 10px #bbb;
        }
        .login-box img{
            display: block;
            margin:auto;
        }
        .login-box input{
            width: 100%;
            margin:10px 0 20px;
            padding: 8px;
            border: 1px solid #b5cad1;
            border-radius: 5px;
            box-sizing: border-box;
        }
        .links{
            font-size: 12px;
            text-align: center;
        }
        .links a{
            display: inline-block;
            width: 70px;
            color: #0087d7;
            text-decoration: none;
        }
    </style>
</head>
```

```
<body>
  <div class="login-box">
    <img src="./images/book-logo.jpg" alt="">
    <h3 style="text-align: center">ورود به سیستم </h3>

    <form method="post">
    <label for="username">نام کاربری </label>
    <input type="text" id="username">

    <label for="password">کلمه عبور</label>
    <input type="password" id="password">

    <input type="submit" value="ورود">

    <div class="links">
        <a href="#">فراموشی رمز</a>
        <span>حساب کاربری ندارید؟ </span>
        <a href="#">ثبت نام</a>
    </div>
    </form>
  </div>
</body>
```



چیدمان Flexbox

فلکس یک مدل طرح‌بندی در CSS است که برای ایجاد طرح‌های منعطف، واکنش‌گرا و یک بعدی استفاده می‌شود. فلکس توانایی ایجاد یک چیدمان افقی یا عمودی را برای عناصر صفحه دارد؛ به‌علاوه امکان تراز کردن عناصر و توزیع خودکار فضا را به توسعه‌دهنده می‌دهد. طرح‌هایی که توسط flexbox ایجاد می‌شوند، با اندازه‌های مختلف صفحه نمایش سازگاری خواهند داشت (جدول ۱۰).

تنظیم ویژگی‌های عنصر والد

برای چینش افقی یا عمودی عناصر فرزند در یک عنصر مشخص، از مدل فلکس استفاده می‌شود. بدین منظور ویژگی display با مقدار Flex برای عنصر والد به‌کار می‌رود. در حالت پیش‌فرض این کار باعث ایجاد چیدمان افقی برای فرزندان عنصر والد می‌گردد.

برای فعال‌کردن Flexbox کافی است ویژگی زیر را به عنصر والد اضافه کنید:

```
display: flex;
```

flex-direction: این ویژگی برای تعیین جهت چینش عناصر، به‌صورت افقی یا عمودی استفاده می‌شود. مقادیر این ویژگی در جدول ۱۰ مشخص شده است.

جدول ۱۰ـ مقادیر ویژگی flex-direction و تأثیر آنها روی نحوه چینش عناصر

مقدار	row	Row-reverse
توضیح	افقی (پیش فرض) flex-direction:row box1 box2 box3	افقی (راست به چپ) flex-direction:row-reverse box3 box2 box1
مقدار	column	Column-reverse
توضیح	عمودی از بالا به پایین flex-direction:column box1 box2 box3	عمودی از پایین به بالا flex-direction:column-reverse box3 box2 box1

flex-wrap: گاهی پهنای صفحه به اندازه‌ای نیست که همه عناصر در کنار یکدیگر قرار بگیرند. ویژگی flex-wrap مشخص می‌کند که عناصر اضافی به سطر بعد منتقل شوند یا همگی در همان سطر فشرده شوند (جدول ۱۱).

جدول ۱۱ــ مقادیر ویژگی flex-wrap

مقدار	wrap	nowrap
توضیح	انتقال به سطر بعد	عدم انتقال به سطر بعد

flex-flow: ویژگی flex-flow ترکیب flex-direction و flex-wrap است. به‌جای نوشتن دو خط کد می‌توان هر دو ویژگی را در یک خط تنظیم نمود. مانند زیر:

```
.container{
    flex-flow: row wrap;
}
```

justify-content: این ویژگی تراز عناصر و نحوه توزیع فضای خالی بین عناصر را در راستای محور اصلی کنترل می‌کند (جدول ۱۲).

جدول ۱۲ــ مقادیر ویژگی justify-content

مقدار	توضیحات	مقدار	توضیحات
Flex-start	چینش عناصر در سمت چپ	Space-around	ایجاد فضای خالی به میزان برابر در اطراف هر عنصر
Flex-end	چینش عناصر در سمت راست	Space-between	فضای خالی برابر بین عناصر (عنصر اول و آخر به لبه‌ها می‌چسبند)
Center	چینش عناصر در مرکز	Space-evenly	فضای خالی برابر در بین عناصر و همچنین بین عناصر و لبه‌ها

align-items: ویژگیalign-items در Flexbox برای تراز‌کردن عناصر در راستای محور متقاطع (عمود بر محور اصلی) استفاده می‌شود. به کمک این ویژگی می‌توان عناصر را در بالا، پایین و وسط یک سطر قرار داد (شکل ۲۶).

```
<!DOCTYPE html>
<html lang="en">
<head>
  <style>
    .container{
        width:300px;
        height:100px;
        display:flex;
        background:rgb(247, 247, 66);
        padding:0 20px;
        margin-bottom: 10px;
    }
    .container1{ align-items:stretch; }
    .container2{ align-items:flex-start; }
    .container3{ align-items:flex-end; }
    .container4{ align-items:center; }
    .box{
        width:70px;
        padding:5px 10px;
        margin:5px;
        text-align: center;
        font-size: 25px;
    }
    .box1{background:rgb(76, 193, 243);}
    .box2{
        background:rgb(255, 80, 138);
        padding:15px 10px}
    .box3{
        background:rgb(56, 245, 56);
        padding:25px 10px;}
  </style>
</head>
```

```
<body>
  <div class=" container container1">
    <div class="box box1">box1</div>
    <div class="box box2">box2</div>
    <div class="box box3">box3</div>
  </div>
  <div class="container container2">
    <div class="box box1">box1</div>
    <div class="box box2">box2</div>
    <div class="box box3">box3</div>
  </div>
  <div class="container container3">
    <div class="box box1">box1</div>
    <div class="box box2">box2</div>
    <div class="box box3">box3</div>
  </div>
  <div class="container container4">
    <div class="box box1">box1</div>
    <div class="box box2">box2</div>
    <div class="box box3">box3</div>
  </div>
</body>
</html>
```

box1 box2 box3

box1 box2 box3

box1 box2 box3

box1 box2 box3

شکل ۲۶

جدول ۱۳ـ مقادیر ویژگی align-items

مقدار	Stretch	flex-start	flex-end	center
توضیح	به‌صورت کشیده درون عنصر والد قرار می‌گیرد (پیش‌فرض)	در بالای عنصر والد قرار می‌گیرد.	در پایین عنصر والد قرار می‌گیرد.	در وسط عنصر والد قرار می‌گیرد.

طرح بندی صفحات با Flexbox

کار کارگاهی

تعیین چیدمان عناصر منوی ناوبری با استفاده از Flexbox

فایل index.html را باز کنید و تغییرات زیر را درون آن انجام دهید:

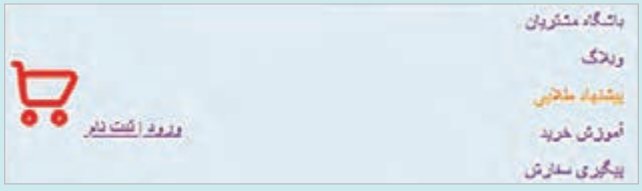
۱ عناصر زیر را درون برچسب <nav> بعد از برچسب اضافه کنید و درون آن آیکن سبد خرید و عبارت «ورود | ثبت نام» را قرار دهید:

```
<div class="user-actions">
  <a href="login.html">ورود | ثبت نام</a>
  <a href="#">
    <img src="./images/shopping-cart.svg">
  </a>
</div>
```

برای تهیه آیکن سبد خرید به سایت svgrepo.com مراجعه کنید و عبارت shopping-cart را جست‌وجو کنید. یک سبد خرید دلخواه را انتخاب کرده و سپس با کلیک روی گزینه Edit vector، تصویر را ویرایش نمایید. اندازه آیکن را روی مقدار حداقل تنظیم کنید. سپس روی دکمه Export کلیک کنید. یک فایل svg دانلود می‌شود. این فایل را با نام shopping-cart.svg در پوشه تصاویر ذخیره کنید.

۲ استایل نوار ناوبری (top-menu.) را به‌صورت شکل زیر تنظیم کنید:

```
.top-menu{
    background:rgb(224, 235, 243);
    display: flex;
    justify-content: space-between;
    align-items: center;
    max-width:1200px;
    margin:0 auto;
}
```



ـ ویژگی display: flex، نوار ناوبری را به یک نگهدارنده فلکس تبدیل می‌کند تا آیتم‌های درون آن (لیست منو در سمت راست و بخش user-action در سمت چپ) به صورت افقی چیده شوند.

ـ ویژگی justify-content : space-between;، لیست منو و بخش user-actions را به صورت افقی در دو انتهای نگهدارنده قرار می‌دهد.

ـ ویژگی align-items: center باعث می‌شود که عناصر درون نگهدارنده فلکس در راستای عمودی در مرکز قرار گیرند.

ـ ویژگی max-width: 1200px، حداکثر پهنای نوار ناوبری را برابر ۱۲۰۰ پیکسل قرار می‌دهد. این مقداردهی باعث می‌شود که نوار ناوبری در صفحه نمایش‌های بزرگ، بیش از حد کشیده نشود.

ـ ویژگی margin:0 auto برای قرار گرفتن محدوده نوار منو در وسط صفحه استفاده می‌شود.

۳ استایل عناصر درون برچسب <ul> را به شکل زیر تنظیم کنید:

ـ ویژگی display:flex برای چیدمان افقی عناصر درون برچسب <ul> استفاده شده است و ویژگی justify-content فضای بین عناصر را به شکل مناسب توزیع می‌نماید.

ـ ویژگی margin-left:25px فاصله هر عنصر منو را از عنصر سمت چپ مشخص می نماید.

```
.top-menu ul{
    list-style: none;
    padding: 0;
    margin: 0;
    display: flex;
    justify-content: space-around;
}
.top-menu ul li a{
    text-decoration: none;
    line-height: 30px;
    margin-left:25px;
}
```

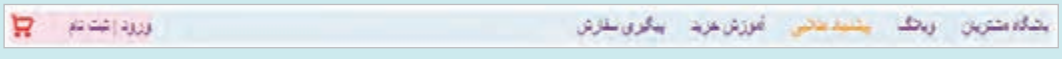


۴ استایل بخش user-actions را به شکل زیر تنظیم کنید:

ـ به‌کارگیری ویژگی display:flex موجب نمایش دو عنصر «ورود|ثبت نام» و سبد خرید به صورت افقی می‌شود و ویژگی align-items:Center این عناصر را در راستای محور عمودی در وسط ناحیه user-actions نمایش می‌دهد.

ـ پهنای تصویر سبد خرید باید به اندازه ۲۵ پیکسل و فاصله آن از سمت چپ (margin-left) باید با مقدار صفر تنظیم شود.

```
.user-actions{
    display: flex;
    align-items: center;
    gap:30px;
    background: pink;
}
.user-actions img{
    width: 25px;
}
a{
    text-decoration: none;
}
```



ـ رنگ‌های پس زمینه را از کلاس‌های top-menu، user-actions حذف کنید تا نتیجه زیر حاصل شود



ایجاد بخش اصلی صفحه

بعد از طراحی بخش هدر سایت، نوبت به طراحی محتوای اصلی صفحه، شامل بخش‌های «جدیدترین محصولات» و «خدمات ویژه» می‌رسد.

برای طراحی بخش اصلی صفحه از برچسب <main> استفاده می‌شود. این برچسب باید بعد از برچسب </header> و قبل از </body> قرار گیرد.

محتوای نواحی «جدیدترین محصولات» و «خدمات ویژه» باید درون عنصر <main> واقع شود.

کار کارگاهی

طراحی بخش «جدیدترین محصولات» با استفاده از Flexbox

۱ یک برچسب <main> بعد از برچسب </header> درج کنید و درون آن یک عنصر <section> قرار دهید. یک عنصر <section> با کلاس container درون عنصر <main> ایجاد کنید. این عنصر باید حاوی یک عنوان (h2) و یک عنصر نگهدارنده (div) برای نگهداری کارت محصولات باشد. عنوان «جدیدترین محصولات» را درون تگ <h2> بنویسید.

۲ یک کلاس با عنوان products به عنصر اختصاص دهید و درون عنصر products کارت محصولات را قرار دهید. کارت محصول را توسط یک عنصر با کلاس product-card ایجاد کنید. هر کارت حاوی تصویر، عنوان و قیمت محصول است.

```
<section  class="container">
  <h2>جدیدترین محصولات</h2>
  <div class="products">
     <div class="product-card"></div>
     <div class="product-card"></div>
     <div class="product-card"></div>
     <div class="product-card"></div>
  </div>
</section>
```

۳ برای آنکه مجموعه تصویر، عنوان و قیمت قابل کلیک باشد، کل محتوای کارت را درون برچسب <a> قرار دهید. تصمیم‌گیری در خصوص چیدمان ستونی تصویر، عنوان و قیمت محصول در تنظیمات عنصر <a> انجام می‌شود.

```
<article class="product-card">
    <a href="#">
        <img src="./images/Computer-Aided-Design.jpg" alt="کتاب نقشه کشی رایانه ای">
        <h3>نقشه کشی رایانه ای</h3>
        <p> 170.000 تومان </p>
    </a>
</article>
```

کارت محصول

```css
.container{
    max-width: 1200px;
    margin: 20px auto;
}
.container h2{
    text-align: center;
    margin-bottom: 30px;
    font-size: 32px;
}
.products{
    display: flex;
    justify-content: space-center;
    flex-wrap: wrap;
    gap:30px;
}
.product-card a{
    display: flex;
    flex-flow: column;
    align-items: center;
    padding:10px 30px 0;
    box-sizing: border-box;
    border:2px solid lightgray;
    border-radius: 5px;
    text-decoration: none;
}
.product-card h3{
    font-size: 16px;
    margin-bottom: 5px;
}
.product-card p{
    margin-top: 0;
}
.products img{
    width: 170px;
    border-radius: 5px;
}
```

محتوای سه کارت دیگر را نیز مشابه کد صفحه قبل بنویسید. ویژگی عناصر را به‌صورت تدریجی به صفحه اضافه کنید و نتیجه هر تغییر را در مرورگر بررسی کنید.

ویژگی max-width حداکثر پهنای ناحیه نگهدارنده را مشخص می‌کند.

ویژگی margin:20px auto عنصر نگهدارنده را در راستای افقی در وسط صفحه قرار می‌دهد.

ویژگی display:flex توزیع کارت‌های موجود در عنصر products را توسط تکنیک فلکس انجام می‌دهد.

ویژگی justify-content:center کارت‌ها را در وسط ناحیهٔ نگهدارنده توزیع می‌کند.

ویژگی flex-wrap:wrap باعث تغییر چیدمان عناصر در هنگام کوچک‌تر شدن پهنای دستگاه می‌گردد. به‌طوری‌که کارت محصولات به سطر بعدی منتقل شوند.

ویژگی gap فاصله بین کارت محصولات را مشخص می‌کند.

با توجه به اینکه مجموعه «تصویر، عنوان و قیمت محصول» درون عنصر <a> قرار گرفته است، پس تنظیم چیدمان ستونی این عناصر توسط انتخاب‌گر product-card a. انجام می‌شود.

ویژگی flex-direction:column چیدمان عناصر را به‌صورت ستونی انجام می‌دهد.

ویژگی align-items:center تراز عناصر را در راستای محور عمودی تنظیم می‌کند.



فعالیت

رنگ زمینه کارت‌ها را با مقدار rgb(251, 248, 249) تنظیم کنید و نتیجه را بررسی کنید.
چند کارت دیگر به مجموعه کارت‌ها اضافه کنید و نحوهٔ چیدمان کارت‌ها را مشاهده کنید.
پهنای مرورگر را تا حد امکان کم کنید و وضعیت چیدمان کارت‌ها را بررسی کنید. آیا با کم شدن فضا، کارت‌ها به سطر بعد منتقل می‌شوند؟ ویژگی flex-wrap:wrap را حذف کنید و مجدداً نتیجهٔ کوچک شدن پهنای صفحه را بررسی کنید.
مقدار ویژگی justify-content را با مقادیر space-around و space-evenly جایگزین کنید و نتیجهٔ چینش محصولات را بررسی کنید.

بیشتر بدانید

طرح‌بندی صفحه با چیدمان شبکه‌ای (Grid)

برخلاف مدل flex که تنها در یک بعد (سطر یا ستون) عمل می‌کند، مدل Grid امکان ایجاد سطرها و ستون‌های حاوی محتوا را به‌صورت هم‌زمان فراهم می‌آورد. با استفاده از سیستم grid، می‌توان کنترل دقیقی بر روی فاصله‌ها و اندازه‌ها داشت. این روش انعطاف‌پذیر است و برای ایجاد ساختارهای شبکه‌ای منظم و واکنش‌گرا بسیار مفید است. برای ایجاد یک چیدمان شبکه‌ای از ویژگی display با مقدار grid استفاده می‌شود. تعداد ستون‌های شبکه توسط ویژگی grid-template-columns و فاصلهٔ بین ستون‌ها توسط ویژگی gap تعیین می‌شود.

کار کارگاهی

طراحی بخش «خدمات ویژه» با استفاده از مدل Grid

در فایل index.html تغییرات زیر را به‌صورت تدریجی اعمال کنید.

۱ پنج تصویر لوگو، مطابق شکل زیر دانلود کنید و درون پوشه تصاویر سایت (images) ذخیره کنید.

۲ یک عنصر <section> با کلاس services-area برای بخش «خدمات ویژه» ایجاد کنید. این عنصر باید حاوی یک عنصر <h2> و یک نگهدارنده با کلاس services-grid باشد. عنصر آیتم‌های خدمات ویژه را درون خود جای می‌دهد.

۳ هر یک از آیتم‌های خدمات ویژه را توسط یک عنصر با کلاس service-item ایجاد کنید. این آیتم‌ها با استفاده از مدل گرید در چند ستون کنار هم قرار می‌گیرند.

```
<section class="services-area">
    <h2>خدمات ویژه</h2>
    <div class="services-grid">
        <div class="service-item"> </div>
        <div class="service-item"> </div>
        <div class="service-item"> </div>
        <div class="service-item"> </div>
        <div class="service-item"> </div>
    </div>
</section>
```

۴ هر آیتم خدمات ویژه شامل یک تصویر لوگو (img) و عنوان آن (p) است. کد HTML بخش خدمات ویژه را مشابه کد شکل زیر تکمیل کنید:

```
<section class="services-area">
    <h2>خدمات ویژه</h2>
    <div class="services-grid">
        <div class="service-item">
            <img src="./images/support.png" alt="هفت روز هفته- 24 ساعته">
            <p>هفت روز هفته- 24 ساعته</p>
        </div>
        <div class="service-item">
            <img src="./images/original-product.png" alt="ضمانت اصالت کالا">
            <p>ضمانت اصالت کالا</p>
        </div>
        <div class="service-item">
            <img src="./images/cash-on-delivery.png" alt="پرداخت در محل">
            <p>پرداخت در محل</p>
        </div>
        <div class="service-item">
            <img src="./images/express-delivery.png" alt="امکان تحویل اکسپرس">
            <p>امکان تحویل اکسپرس</p>
        </div>
        <div class="service-item">
            <img src="./images/days-return.png" alt="هفت روز ضمانت بازگشت کالا">
            <p>هفت روز ضمانت بازگشت کالا</p>
        </div>
    </div>
</section>
```

۵ استایل عناصر را به‌صورت تدریجی به صفحه اضافه کنید و نتیجه را در مرورگر مشاهده کنید.

ویژگی display: grid عنصر services-grid را به یک نگهدارنده گرید تبدیل می‌کند. در نتیجه تمام فرزندان مستقیم آن (service-item) به آیتم‌های گرید تبدیل شده و می‌توان آنها را در ردیف‌ها و ستون‌ها توزیع نمود.

ویژگی grid-template-columns با پنج مقدار auto پنج ستون ایجاد می‌کند که پهنای آنها وابسته به محتوای هر ستون خواهد بود. به کمک این ویژگی تصاویر لوگو در پنج ستون ظاهر می‌شوند.

ویژگی gap:20px فاصلهٔ میان ستون‌ها را برابر ۲۰ پیکسل قرار می‌دهد.

```
.services-area{
    max-width: 900px;
    margin:50px auto;
}
.services-area h2{
    text-align: center;
    margin-bottom: 20px;
    font-size: 32px;
}
.services-grid{
    display: grid;
    grid-template-columns:repeat(5,auto);
    gap:20px;
}
.service-item{
    display: flex;
    flex-flow: column;
    align-items: center;
    padding:10px;
}
.service-item:hover{
    background: rgb(242, 240, 251);
}
.service-item img{
    width:50px;
    margin-bottom:15px;
}
.service-item p{
    font-size: 14px;
    margin: 0;
    color:#555;
}
```



بیشتر بدانید

موقعیت‌دهی به عناصر صفحه توسط ویژگی position

ویژگی position یکی از مهم‌ترین ویژگی‌ها برای کنترل نحوه قرارگیری عناصر در صفحه است. این ویژگی برای تنظیم جایگاه یک عنصر نسبت به عناصر والد یا عناصر دیگر کاربرد دارد. ویژگی position دارای مقادیر static، relative، absolute، fixed و sticky به شرح زیر است:

static: تمام عناصر صفحه دارای مقدار پیش فرض static هستند.

relative: جایگاه یک عنصر را نسبت به موقعیت اولیه آن مشخص می‌کند. برای جابه‌جایی عنصر نسبت به موقعیت اولیه، از ویژگی‌های top، left، right و bottom استفاده می‌شود.

absolute: موقعیت عنصر را نسبت به عنصر والد مشخص می‌کند.

fixed: جایگاه یک عنصر را به‌صورت ثابت حفظ می‌کند.

تصویر یک دکمه chat-widget را در موقعیت (۳۰، ۶۰۰) به‌صورت ثابت نمایش دهید. برای این کار تصویر دکمه چت ویجت را در ابعاد ۴۵×۴۵ به صفحه اضافه کنید. برای تنظیم اندازه تصویر از برنامه paint استفاده کنید. تصویر را قبل از برچسب </main> قرار دهید:

```
<img class="chat-widget" src="./images/chat-widget.jpg" alt="chat-widget-logo">
```

```
.chat-widget{
  position:fixed;
  top:600px;
  right:20px;
  border-radius:50%;
  cursor: pointer;
}
```

استایل تصویر را مانند کدهای روبه‌رو تنظیم کنید.

تصویر این دکمه در گوشه پایین سمت راست صفحه نمایش داده خواهد شد.



مدیا کوئری (Media query)

مدیا کوئری یک تکنیک CSS است که امکان تعیین ویژگی‌های عناصر را بر اساس ویژگی‌های مختلف دستگاه‌ها (مانند اندازه صفحه‌نمایش، وضوح، جهت‌گیری و غیره) فراهم می‌کند. این قابلیت شرایطی را فراهم می‌کند که توسعه دهنده بتواند وب‌سایت‌ها و اپلیکیشن‌های وب خود را برای انواع دستگاه‌ها (مانند موبایل، تبلت و دسکتاپ) بهینه‌سازی نماید.

مدیا کوئری‌ها با استفاده از دستور @media در CSS نوشته می‌شوند و دارای ساختار زیر هستند:

```
@media media-type and (condition) {
    CSS rules
}
```

کار کارگاهی

استفاده از مدیاکوئری برای تغییر وضعیت عناصر در نمای موبایل

استایل‌های زیر را به انتهای سبک‌های فایل index.html اضافه کنید. فایل را ذخیره کنید و نتیجه را در مرورگر بررسی کنید.

`@media (max-width:600px){` `.top-menu ul {` `flex-direction: column;` `width: 100%;` `text-align: center;` `}`	تغییر چیدمان عناصر منو از حالت افقی به حالت عمودی اشغال پهنای کامل برای هر عنصر لیست وسط چین کردن متن گزینه‌ها
`.top-menu ul li {` `margin-bottom: 10px;` `}`	تنظیم فاصلهٔ عمودی بین گزینه‌های منو
`.user-actions {` `display: none;` `}`	مخفی کردن ناحیه user-action
`.services-grid {` `grid-template-columns: 1fr;` `gap: 30px;` `}`	ایجاد یک ستون واحد افزایش فاصله عمودی بین گزینه‌ها
`.service-item {` `max-width: 300px;` `margin: 0 auto;` `}`	قرار دادن آیتم‌های خدمات در وسط سطر
`.chat-widget{` `display: none;` `}` `}`	پنهان‌سازی لوگوی چت ویجت

فعالیت

برای بررسی عملکرد مدیا کوئری، لبهٔ سمت راست مرورگر را تا حد امکان به سمت چپ جابه‌جا کنید. با کوچک شدن پهنای پنجره، گزینه‌های منو به‌صورت عمودی چیده می‌شوند. ناحیهٔ user-actions ناپدید می‌گردد و تصاویر خدمت ویژه به‌صورت ستونی چیده می‌شوند.

ارزشیابی شایستگی طراحی ظاهر صفحات وب با CSS

استاندارد عملکرد کار: با استفاده از شیوه‌های CSS، ظاهر و چیدمان‌های پیچیده وب‌سایت‌ها را مطابق با استانداردها و طراحی‌های بصری مشخص پیاده‌سازی نماید.

ردیف	مرا حل کار	شرایط انجام کار (ابزار،مواد، تجهیزات، زمان، مکان و ...)	ابزارهای ارزشیابی	استاندارد (شاخص‌ها/داوری/نمره‌دهی)	نتایج ممکن	نمره کسب شده
۱	آشنایی با روش‌های افزودن CSS، مدیریت آنها	رایانه یا لپ‌تاپ سالم، ویرایشگر کد (VS Code یا مشابه)، مرورگر وب به‌روز، زمان ۲۵ دقیقه،	مشاهده نصب توسط هنرجو، تمرین عملی ـ مشاهده کدنویسی	استفاده صحیح از هر سه روش (Inline، Internal، External) ـ تشخیص و اصلاح تداخل استایل‌ها ـ رعایت ترتیب و اولویت استاندارد ـ اعمال صحیح تغییر ظاهری روی عنصر مورد نظر ـ اجرای کارها به چند روش ـ اجرای کنجکاوی‌ها	۳	
				استفاده صحیح از هر سه روش (Inline، Internal، External) ـ تشخیص و اصلاح تداخل استایل‌ها ـ رعایت ترتیب و اولویت استاندارد ـ اعمال صحیح تغییر ظاهری روی عنصر موردنظر	۲	
				انجام نادرست یا ناقص هر یک از مراحل کاری	۱	
۲	قالب‌دهی به متن	رایانه یا لپ‌تاپرایانه یا لپ‌تاپ سالم، ویرایشگر کد (VS Code یا مشابه)، مرورگر وب به‌روز، زمان ۲۵ دقیقه	مشاهده کد نوشته شده ـ کد نویسی	ـ با استفاده از ویژگی مربوط به قالب‌بندی متن را به درستی انجام دهد.ـ از ویژگی‌های مربوط به تعیین ظاهر عناصر برای تعیین رنگ متن، رنگ زمینه، تصویر پس‌زمینه، کنترل تصویر پس‌زمینه، شفافیت و خط دور عناصر به‌درستی استفاده کند ـ اجرای کارها با چند روش ـ اجرای کنجکاوی‌ها	۳	
				با استفاده از ویژگی مربوط به قالب‌بندی متن را به‌درستی انجام دهد.ـ از ویژگی‌های مربوط به تعیین ظاهر عناصر برای تعیین رنگ متن، رنگ زمینه، تصویر پس‌زمینه، کنترل تصویر پس‌زمینه، شفافیت و خط دور عناصر به‌درستی استفاده کند.	۲	
				اجرای نادرست و یا ناقص هر یک از مراحل کاری	۱	
۳	کار با انتخاب‌گرها در CSS	رایانه، مرورگر برای تست زمان ۱۵ دقیقه، اجرای کار در کلاس عملی یا آزمایشگاه رایانه.	مشاهده عملکرد قالب، تمرین گروهی	ـ تعیین استایل عناصر صفحه براساس رابطهٔ والد و فرزندی بین آنها با استفاده درست از انتخاب‌گرهای ساده ـ تعیین استایل عناصر براساس حالات و ویژگی‌های خاص آنها ـ با استفاده از انتخاب‌گرهای شبه کلاس ـ امکان انتخاب عناصر براساس ویژگی‌های آنها با استفاده از انتخاب‌گرهای ترکیبی ـ انجام کارها با چند روش ـ انجام کنجکاوی‌ها	۳	
				ـ تعیین استایل عناصر صفحه براساس رابطهٔ والد و فرزندی بین آنها با استفاده درست از انتخاب‌گرهای ساده ـ تعیین استایل عناصر براساس حالات و ویژگی‌های خاص آنها ـ با استفاده از انتخاب‌گرهای شبه کلاس ـ امکان انتخاب عناصر براساس ویژگی‌های آنها با استفاده از انتخاب‌گر ترکیبی	۲	
				به کارگیری انتخاب‌گرهای ساده	۱	
۴	کار با مدل های جعبه‌ای	رایانه، مرورگر برای تست زمان ۳۰ دقیقه، اجرای کار در کلاس عملی یا آزمایشگاه رایانه.	مشاهده عملکرد قالب	ـ عناصر معنایی (header, main, footer, section, article, nav) درست استفاده شده باشندـ صفحه استاندارد HTML5 داشته باشدـ انجام کنجکاوی‌ها ـ انجام کار با روش‌های دیگر	۳	
				ـ عناصر معنایی (header, main, footer, section, article, nav) درست استفاده شده باشندـ صفحه استاندارد HTML5 داشته باشد.	۲	
				انجام ناقص هریک ازمراحل	۱	
۵	مدل‌های نمایش عناصر در صفحه Flexbox-gride model - disply	رایانه ـ مرورگر برای تست زمان ۳۰ دقیقه، کار در کلاس عملی یا آزمایشگاه رایانه.	پرسش کتبی ـ آزمون عملی ـ مشاهده عملکرد هنرآموز	ـ عناصر جانبی (aside ـ caption) به‌درستی اضافه شده باشند ـ تصاویر همراه با توضیحات درست نمایش داده شوندـ ایجاد جدول با ۲ سطر و ۲ ستون ـ ایجاد فرم‌های ورود اطلاعات ـ انجام کنجکاوی‌ها ـ انجام کار با روش‌های دیگر	۳	
				ـ عناصر جانبی (aside ـ caption) به‌درستی اضافه شده باشندـ تصاویر همراه با توضیحات درست نمایش داده شوند ـ ایجاد فرم‌های ورود اطلاعات	۲	
				انجام ناقص هریک از مراحل	۱	

۶	تکمیل ساختار با عناصر غیرمعنایی و مستندسازی	رایانه مرورگر منابع راهنما برای مستندسازی کد ۱۵ دقیقه	مشاهده مستندسازی و توضیحات نوشته‌شده اجرای صفحه و صحت نمایش نهایی	ـ استفاده مناسب از div و span برای ساختاردهی ـ توضیحات و مستندسازی کد کامل باشد ـ مزایای استفاده از تگ‌های معنایی (مانند article, section) را برای سئو و دسترسی‌پذیری توضیح دهند ـ انجام کنجکاوی‌ها ـ انجام کار با روش‌های دیگر	۳	
				ـ استفاده مناسب از div و span برای ساختاردهی ـ نظرها و مستندسازی کد کامل باشد.	۲	
				انجام ناقص هریک از مراحل ـ مزایای استفاده از تگ‌های معنایی (مانند article, section) را برای سئو و دسترسی‌پذیری توضیح دهند.	۱	
شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش:				احراز همه شایستگی‌های غیرفنی	۲	
				عدم احراز هر یک از شایستگی‌های غیرفنی	۱	
ارزشیابی کار (شایستگی انجام کار)					بلی ☐	
					خیر ☐	

توضیحات:

۱ـ معیار شایستگی انجام کار:

ـ کسب حداقل نمره ۲ از مراحل ۱، ۲، ۳، ۴ و ۵ (مراحل بحرانی)

ـ کسب حداقل نمره ۲ از بخش شایستگی‌های غیرفنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش

ـ کسب حداقل میانگین ۲ از مراحل کار

۲ـ تعریف سطوح شایستگی: صرف‌نظر از اینکه یک تکلیف کاری در چه سطحی از صلاحیت حرفه‌ای انجام می‌شود، در هر محیط کاری ممکن است انجام هر کار با کیفیت مشخصی مورد انتظار باشد. سطح شایستگی انجام کار، معیار اساسی ارزشیابی است.

مهارت (سطح سه): ماهر و قادر به آموزش و هدایت دیگران، توانایی برنامه‌ریزی و تحلیل، پاسخ‌گویی در برابر کارهای خود، سروکار داشتن با سطح وسیعی از کارها و فعالیت‌ها

تسلط (سطح چهار): خبرگی در انجام کار و آموزش دیگران، ایجاد نوآوری، سازگاری، عیب‌یابی، هدایت و راهنمایی دیگران.

پودمان سوم

تولید صفحات وب تعاملی



تولید صفحات وب تعاملی به معنای ایجاد تجربه‌ای پویا و جذاب برای کاربران است که فراتر از نمایش ساده و بدون تغییر اطلاعات می‌باشد. امروزه وب‌سایت‌ها ابزارهای اصلی ارتباط و تبادل اطلاعات هستند، بنابراین باید بتوانند با کاربران خود تعامل داشته باشند. منظور از تعامل با کاربران این است که عناصر وب‌سایت امکان پاسخ‌دهی به فعالیت‌های کاربر را داشته باشند، مانند پاسخ‌گویی به کلیک‌ها و پر کردن اطلاعات فرم. استفاده از زبان‌های برنامه‌نویسی مانند جاوااسکریپت (JavaScript) امکان تعریف پاسخ برای فعالیت‌های کاربران را فراهم می‌کند. جاوااسکریپت با ایجاد انیمیشن‌ها و جلوه‌های بصری، جذابیت بیشتری به صفحات می‌بخشد و کاربران را به ماندن و کاوش بیشتر ترغیب می‌کند. علاوه بر قدرت جاوااسکریپت در ایجاد تعامل، انتخاب یک فریم‌ورک مناسب تأثیر زیادی بر زیبایی و سرعت ساخت سایت دارد. بوت‌استرپ (Bootstrap) به عنوان یک فریم‌ورک قدرتمند فرانت‌اند، با ارائه مجموعه‌ای از کلاس‌های CSS آماده و کامپوننت‌های جاوااسکریپتی، فرایند طراحی رابط‌های کاربری مدرن و واکنش‌گرا را تسهیل می‌کند. این ابزار به توسعه‌دهندگان امکان می‌دهد که بدون درگیر شدن با پیچیدگی‌های کدنویسی پایه، بر روی خلق تجربه‌های تعاملی و جذاب تمرکز کنند و صفحاتی بسازند که در تمام دستگاه‌ها به شکلی حرفه‌ای و یکسان نمایش داده شوند.

صفحات وب تعاملی و واکنش‌گرا علاوه بر بهبود تجربه کاربری، امکان شکل‌گیری ارتباطات مؤثر و ایجاد اعتماد بین کاربران و وب‌سایت‌ها را فراهم می‌کنند. در این پودمان، با یادگیری اصول برنامه‌نویسی جاوااسکریپت و کار با کلاس‌های بوت‌استرپ، مهارت طراحی صفحات وب تعاملی و واکنش‌گرا را فرا خواهید گرفت.

واحد یادگیری ۱
توسعه صفحات وب تعاملی با بوت استرپ

آیا تابه‌حال اندیشیده‌اید

- چگونه می‌توان طراحی وب‌سایت‌های واکنش‌گرا و سازگار با موبایل را بدون نوشتن کدهای CSS پیچیده انجام داد؟
- آیا فریم ورک‌هایی وجود دارند که کلاس‌های آماده CSS را برای توسعه‌دهندگان وب ارائه کنند؟
- بوت‌استرپ چه کلاس‌ها و اجزای آماده‌ای برای طراحی بخش‌های مختلف صفحات وب دارد؟
- چگونه می‌توان با بوت‌استرپ، فرم‌های ورود اطلاعات را با سرعت بالا و به شکل مدرن طراحی کرد؟
- چگونه می‌توان طراحی صفحات وب را با استفاده از بوت‌استرپ بهینه کرد تا در تمام دستگاه‌ها به بهترین نحو نمایش داده شوند؟

در پایان از هنرجو انتظار می‌رود

۱ طراحی کامل ساختار یک صفحه وب (شامل هدر، بدنه و فوتر) با استفاده از سیستم شبکه‌بندی (Grid System) بوت‌استرپ، به‌طوری‌که طرح‌بندی در اندازه‌های مختلف صفحه نمایش (موبایل، تبلت، دسکتاپ) به‌درستی واکنش نشان دهد.

۲ انتخاب و به‌کارگیری کلاس‌های کمکی (Utility Classes) مانند کلاس‌های رنگ، فاصله (Margin/Padding)، سایه‌ها و تایپوگرافی برای استایل‌دهی سریع و استاندارد به عناصر بدون نیاز به نوشتن CSS دستی برای موارد متداول.

۳ ساخت یک نوار ناوبری (Navbar) کاربردی و زیبا که شامل لینک‌ها، دکمه‌ها و قابلیت‌های واکنش‌گرا باشد و همچنین طراحی یک فوتر ساختاریافته.

۴ مهارت در پیاده‌سازی اجزای رابط کاربری مانند کروسل (تصاویر اسلایدی) و آکاردئون‌ها برای ارائه اطلاعات به شیوه‌ای جذاب و پویا.

۵ ساخت فرم‌های ثبت‌نام یا تماس با استایل‌دهی کامل بوت‌استرپ.

۶ تغییر و لغو استایل‌های پیش‌فرض بوت‌استرپ با استفاده از روش‌های سفارشی‌سازی برای هماهنگ کردن ظاهر وب‌سایت با برند یا نیازهای خاص پروژه.

استاندارد عملکرد

کلاس‌های آماده بوت‌استرپ را برای تعیین سبک عناصر مختلف صفحه مانند دکمه‌ها، فرم‌ها و منوها و... به کار ببرد و توانایی طراحی واکنش‌گرای صفحه را در دستگاه‌های مختلف داشته باشد.

تعریف فریم ورک

فریم‌ورک (Framework) یک چارچوب نرم‌افزاری از پیش طراحی‌شده است که به منظور سرعت بخشیدن به فرایند طراحی و توسعه وب‌سایت‌ها و اپلیکیشن‌های وب به‌کار می‌رود. یک فریم ورک با ارائهٔ مجموعه‌ای از ابزارها، کتابخانه‌ها و الگوهای آماده، به توسعه‌دهندگان امکان می‌دهد که تنها روی ویژگی‌ها و عملکردهای خاص پروژه تمرکز کنند و از کارهای تکراری و ابتدایی دوری کنند.

Bootstrap چیست؟

بوت‌استرپ یک فریم‌ورک متن باز برای طراحی ظاهر سایت است که شامل مجموعه‌ای از کدهای آماده نوشته‌شده با زبان‌های Html،Css و Java Script است. توسعه‌دهندگان به‌جای اینکه از صفر همه چیز را بنویسند از این کدهای آماده استفاده می‌کنند.

ویژگی‌های Bootstrap: بوت‌استرپ دارای ویژگی‌های متعددی در زمینه طراحی صفحات وب است که باعث محبوبیت آن در میان توسعه‌دهندگان و طراحان وب شده است. این ویژگی‌ها به شرح زیر است:

طراحی واکنش‌گرا (Responsive Design): بوت‌استرپ دارای یک سیستم گرید ۱۲ ستونی است که امکان طراحی واکنش‌گرا را برای دستگاه‌های مختلف فراهم می‌کند.

اجزای از پیش طراحی‌شده (Components): بوت‌استرپ شامل اجزای مختلفی مانند دکمه‌ها، فرم‌ها، منوی ناوبری، کاردها، مدال‌ها و نوار جست‌وجو است که به‌صورت آماده قابل استفاده هستند.

پشتیبانی از جاوااسکریپت: بوت‌استرپ حاوی کتابخانه‌های جاوااسکریپت برای ایجاد تعاملات و انیمیشن‌ها در بخش‌های مختلف صفحه است.

پشتیبانی از مرورگرها: بوت‌استرپ با اکثر مرورگرهای مدرن مانند Chrome، Firefox، Safari و Edge سازگار است.

مستندات جامع: سایت رسمی بوت‌استرپ (getbootstrap.com) دارای مستندات کامل برای آموزش نحوهٔ استفاده از اجزا و کلاس‌های بوت‌استرپ است.

کاربردهای Bootstrap

طراحی وب‌سایت‌های واکنش‌گرا: طراحی و توسعه وب‌سایت‌های شرکتی، فروشگاهی، وبلاگ‌ها و غیره.

ایجاد رابط‌های کاربری جذاب: طراحی رابط‌های کاربری زیبا و مدرن برای وب‌سایت‌ها و اپلیکیشن‌های وب.

پروژه‌های minimum viable product (MVP): ساخت نمونه‌های اولیه وب‌سایت‌ها (MVP) و ارزیابی ایده‌های طراحی.

طراحی فرم‌ها: طراحی فرم‌های زیبا و واکنش‌گرا توسط کلاس‌های آماده بوت‌استرپ.

به‌کارگیری بوت‌استرپ در صفحه وب

برای استفاده از امکانات بوت‌استرپ در یک صفحه وب، ابتدا باید فایل HTML پایه ایجاد شود و سپس فایل‌های CSS و جاوااسکریپت بوت‌استرپ به آن الحاق شود. فایل پایه شامل برچسب‌های <!DOCTYPE>، <html>، <head>، متاتگ‌های viewport، <title> و <body> است.

کار کارگاهی

استفاده از CDN[1] در صفحه وب

یک صفحه وب با نام test-bs.html ایجاد کنید و برچسب‌های اولیه آن را بنویسید. سپس سایت getbootstrap.com را باز کنید. صفحه را تا بخش Include via CDN پیمایش کنید تا کدهای آماده <Link> و <script> را مشاهده کنید. کد <link> را در بخش <head> و کد <script> را قبل از </body> کپی کنید. این دو کد فایل‌های CSS و جاوااسکریپت بوت استرپ را به صفحه وب الحاق می‌کنند.

کد لینک و اسکریپت بوت‌استرپ حاوی یک ویژگی به‌نام Integrity است که برای حفظ امنیت فایل دریافت شده از CDN استفاده می‌شود. با این حال امکان استفاده از کلاس‌های بوت‌استرپ بدون Integrity هم فراهم است. به همین دلیل در کد فوق ویژگی Integrity حذف شده است تا خوانایی کد افزایش یابد ولی استفاده از این ویژگی در پروژه‌های واقعی ضروری است.

```
<!doctype html>
<html lang="fa">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/css/bootstrap.min.css"
                                                              rel="stylesheet">
</head>
<body>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/js/
                                                    bootstrap.bundle.min.js">
</script>
</body>
</html>
```

کلاس‌ها و کامپوننت‌های بوت‌استرپ

کلاس‌های بوت‌استرپ مجموعه‌ای از استایل‌های از پیش تعریف شده هستند که امکان طراحی سریع و راحت اجزای رابط کاربری صفحات وب را فراهم می‌کنند. این کلاس‌ها که با استفاده از CSS ایجاد شده‌اند، به توسعه‌دهنده وب کمک می‌کنند که با نوشتن کدهای CSS کمتر، عناصری کاربرپسند و واکنش‌گرا را به صفحه وب اضافه کند.

۱ـCDN: Content Delivery Network (شبکه توزیع محتوا)

کامپوننت‌های بوت‌استرپ عناصر از پیش طراحی شده‌ای هستند که امکان طراحی سریع و آسان رابط کاربری وب‌سایت‌ها را فراهم می‌کنند. توسعه‌دهندهٔ وب می‌تواند به کمک این عناصر آماده، بدون نوشتن کدهای CSS و JavaScript، عناصر کاربرپسند و واکنش‌گرا را به صفحه وب اضافه کند. از جمله کامپوننت‌های بوت‌استرپ می‌توان به دکمه، نوار ناوبری[۱]، کروسل[۲]، آکاردئون[۳] و مُدال[۴] اشاره نمود. این بخش به معرفی کلاس‌ها و کامپوننت‌های بوت‌استرپ می‌پردازد.

کلاس‌های Utility : کلاس‌های Utility در بوت‌استرپ، به‌طور ویژه برای استایل‌دهی عناصر صفحه استفاده می‌شوند و به‌طور مستقیم با HTML و CSS کار می‌کنند. برای استفاده از این کلاس‌ها نیازی به جاوااسکریپت نیست. کلاس‌های آماده بوت‌استرپ برای تعیین رنگ، اندازه عناصر، فواصل، خط دور، سایه و غیره قابل استفاده هستند.

کلاس‌های رنگ: کلاس‌های رنگ بوت‌استرپ به دو دستهٔ رنگ‌های زمینه و رنگ متن تقسیم می‌شوند. هر یک از کلاس‌های رنگ دارای مفهوم و رنگ مشخصی هستند. کلاس‌های رنگ زمینه و متن به ترتیب دارای پیشوند bg و text هستند. برخی از این کلاس‌ها در جدول زیر گردآوری شده است:

جدول ۱ــ کلاس‌های رنگ Bootstrap

کلاس رنگ زمینه	کلاس رنگ متن	مفهوم
bg-primary	text-primary	رنگ اصلی
bg-secondary	text-secondary	رنگ ثانویه
bg-success	text-success	رنگ موفقیت
bg-danger	text-danger	رنگ خطر
bg-warning	text-warning	رنگ هشدار
bg-info	text-info	رنگ اطلاعات
bg-light	text-light	رنگ روشن
bg-dark	text-dark	رنگ تیره

۱- نوار ناوبری (Navbar): نوار بالاتری صفحه که معمولاً شامل لوگو، لینک‌ها و منوهاست.
۲- کروسل (Carousel): یک عنصر تعاملی که مجموعه‌ای از تصاویر یا محتوا را به صورت پشت سرهم در یک فضای مشخص نمایش می‌دهد و کاربر می‌تواند بین آنها جابه‌جا شود.
۳- آکاردئون (Accordion): یک بخش قابل باز و بسته شدن برای نمایش محتوا.
۴- مُدال (Modal): پنجره یا کادر محاوره‌ای که روی محتوای صفحه ظاهر می‌شود.

کنجکاوی

برای کسب اطلاعات بیشتر دربارهٔ سایر کلاس‌های رنگ، عبارت Bootstrap5.3 background و Bootstrap5.3 colors را در گوگل جست‌وجو کنید.

کار کارگاهی

استفاده از کلاس‌های رنگ بوت استرپ

فایل test-bs.html را باز کنید، سپس یک عنصر با مشخصات زیر بعد از برچسب body درج کنید.

```
<div class="text-light  bg-primary"> فروشگاه اینترنتی مرجع نو </div>
```

فایل را ذخیره کنید و نتیجه را در مرورگر بررسی کنید.
به انتهای کلاس رنگ، عبارت subtle را اضافه کنید (bg-primary-subtle) و نتیجه را در مرورگر مشاهده کنید.

کلاس‌های تعیین اندازه (Sizing)

بوت‌استرپ دارای دو مجموعه کلاس w-* و h-* برای تعیین پهنا و ارتفاع عناصر است. پارامتر * می‌تواند جایگزین اعداد ۲۵، ۵۰، ۷۵ و ۱۰۰ و عبارت auto شود. اختصاص کلاس w-75 برای یک عنصر باعث می‌شود که پهنای عنصر برابر ۷۵٪ پهنای عنصر والد باشد. کلاس‌های w-auto و h-auto پهنا و ارتفاع یک عنصر را به اندازهٔ محتوای آن تنظیم می‌نمایند.

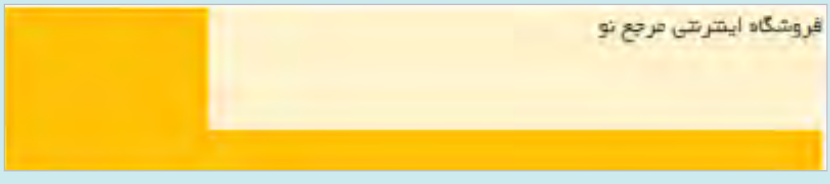
کار کارگاهی

استفاده از کلاس‌های Sizing

در فایل test-bs.html یک عنصر div دیگری به شکل زیر به صفحه وب اضافه کنید. این عنصر حاوی یک عنصر پاراگراف است که پهنا و ارتفاع پاراگراف، ۷۵٪ پهنا و ارتفاع عنصر div خواهد بود.

```
<div class="bg-warning" style="width:500px; height:100px">
    <p class="text-dark bg-warning-subtle w-75 h-75"> فروشگاه اینترنتی مرجع نو </p>
</div>
```

در کد فوق پهنا و ارتفاع عنصر پاراگراف توسط کلاس‌های w-75 و h-75 مشخص شده است. فایل را ذخیره کنید و نتیجه را در مرورگر بررسی کنید.



کلاس‌های margin و padding

بوت‌استرپ دارای دو مجموعه کلاس m{side}-size و p{side}-size برای تعیین فاصلهٔ اطراف و فضای اضافی داخل عناصر است که در CSS توسط ویژگی‌های margin و padding تعیین می‌شوند. پارامتر side تعیین‌کنندهٔ جهت درج حاشیه یا فضای خالی است و پارامتر size اندازه حاشیه یا فضای خالی را تعیین می‌کند. مجموعه کلاس‌های m و p و مفهوم آنها در جدول زیر آمده است:

جدول ۲- کلاس‌های margin و padding در بوت‌استرپ

کلاس‌های margin	کلاس‌های padding	مفهوم
m-size	p-size	در تمام جهات
m{t}-size	p{t}-size	بالا (top)
m{e}-size	p{e}-size	سمت راست در ltr (end)
m{b}-size	p{b}-size	پایین(bottom)
m{s}-size	p{s}-size	سمت چپ در ltr (start)
m{x}-size	p{x}-size	چپ و راست (محور x)
m{y}-size	p{y}-size	بالا و پایین (محور y)

مقدار size عددی بین صفر تا پنج است. هر یک از اعداد این محدوده بیانگر فاصله مشخصی بر حسب px است.

جدول ۳ـ مقادیر پارامتر size بر حسب واحد اندازه گیری px

Size	اندازه بر حسب px
۱	4px
۲	8px
۳	16px
۴	24px
۵	48px

پارامتر size می‌تواند دارای مقدار auto نیز باشد. عبارت auto به معنای تنظیم خودکار است. استفاده از mx-auto باعث تنظیم فاصله چپ و راست عنصر با مقادیر مساوی می‌گردد، به‌طوری‌که عنصر در وسط عنصر والد خود قرار گیرد.

کار کارگاهی

استفاده از کلاس‌های margin و padding

در ادامه کارگاه قبل یک عنصر div با مشخصات زیر به صفحه وب اضافه کنید:

```
<div class="text-light bg-primary w-75 mt-4 mx-auto p-3">فروشگاه اینترنتی فرجع نو</div>
```

در کد فوق مقدار فاصله عنصر از عنصر بالایی برابر 24px یا (mt-4) و مقدار فاصله چپ و راست آن به‌صورت خودکار (mx-auto) تنظیم شده است. همچنین فضای اضافی داخل عنصر از همه جهات برابر با 16px (p-3) تعیین شده است.

کلاس‌های border، rounded و shadow

بوت‌استرپ حاوی مجموعه‌ای از کلاس‌ها درباره خط دور، گردی گوشه‌ها و سایر عناصر است. کلاس‌های border، برای ایجاد خط دور، تعیین رنگ و ضخامت آن مورد استفاده قرار می‌گیرند. کلاس‌های rounded برای گردکردن گوشه‌ها و کلاس‌های shadow برای ایجاد سایر عناصر کاربرد دارند. جدول ۴ به معرفی انواع کلاس‌های border، rounded و shadow می‌پردازد.

جدول ۴ــ انواع کلاس‌های border، rounded و shadow

سایه و اندازه آن	گردی گوشه‌ها	رنگ خط دور	خط دور و ضخامت آن
shadow-none shadow-sm shadow shadow-lg	rounded rounded-circle rounded-pill rounded-top rounded-bottom rounded-start rounded-end	border-primary border-secondary border-success danger-danger border-info border-warning border-light border-dark	border border-1 border-2 border-3 border-4 border-5

جدول ۵ــ کلاس‌های بوت‌استرپ برای تعیین اندازه گردی گوشه‌ها و ضخامت خط دور

اندازه	مقدار گردی گوشه‌ها	ضخامت خط دور	خط دور
0px (no rounding)	rounded-1	1px	border-1
2px	rounded-2	2px	border-2
4px	rounded-3	3px	border-3
6px	rounded-4	4px	border-4
	rounded-5	5px	border-5

در مجموعه کلاس‌های rounded، کلاس rounded-circle برای گرد شدن کامل یک عنصر به‌کار می‌رود. شرط گردشدن کامل یک عنصر، برابر بودن پهنا و ارتفاع آن است. کلاس rounded-pill نیز برای گرد شدن گوشه‌ها همانند قرص یا کپسول به‌کار می‌رود. کلمات start و end به سمت چپ و راست یک عنصر اشاره دارند.

کار کارگاهی

استفاده از کلاس‌های border، shadow و rounded

در ادامه کارگاه قبل، دو عنصر با مشخصات زیر به صفحه test-bs.html اضافه کنید.

برای ایجاد کادر مستطیلی تصویر زیر، کد زیر را بنویسید:

```
<div class="text-primary bg-light w-25 my-4 mx-auto p-3 border border-primary border-1
rounded shadow">ارائه خدمات باکیفیت به دانش آموزان</div>
```

برای ایجاد تصویر مدور، تصویر یک محصول را با ابعاد یکسان به صفحه اضافه کنید و تنظیمات آن را به شکل زیر انجام دهید:



کلاس‌های typography

کلاس‌های متنی بوت‌استرپ برای تنظیم اندازه و وزن فونت، تراز، استایل لیست‌ها و موارد دیگر استفاده می‌شود. جدول ۶ به برخی از کلاس‌های متنی اشاره دارد.

جدول ۶ــ کلاس‌های متنی Bootstrap

اندازه متن	وزن	تراز متن	استایل لیست	آرایش متن
fs-1 … fs-6 h1 … h6 display-1 … display-4	fw-bold fw-light fw-lighter	text-start text-center text-end	list-unstyled list-inline	text-decoration-none text-capitalize text-uppercase

- کلاس‌های fs-* برای تنظیم اندازه فونت عناصر متنی از کوچک (fs-6) به بزرگ (fs-1) کاربرد دارد.
- کلاس‌های h1 تا h6 برای تنظیم اندازه برچسب‌های عنوان استفاده می‌شود.
- کلاس‌های display-* برای ایجاد عناوین بزرگ و جلب توجه کاربر استفاده می‌شوند و مفهوم خاصی را منتقل نمی‌کنند.

کلاس list-unstyled استایل پیش فرض لیست را حذف می‌کند؛ به این معنی که علائم یا شماره‌های کنار عنصر ناپدید شده و مقدار padding-left برابر صفر می‌گردد.
کلاس list-inline عناصر لیست را آماده چیدمان افقی می‌کند، به‌شرطی که عناصر لیست دارای کلاس list-inline-item باشند. این نوع لیست برای منوهای ساده و ناوبری‌های کوچک کاربرد دارد.

کلاس‌های flex

بوت‌استرپ دارای کلاس‌های مختلفی در زمینه طراحی عناصر صفحه با تکنیک Flexbox است. سیستم فلکس دارای دو کلاس پایه با اسامی d-flex و d-inline-flex است که برای عناصر نگهدارنده استفاده می‌شوند. به‌طور مثال بخش معرفی محصولات نیازمند یک عنصر نگهدارنده است که محصولات درون آن به‌صورت انعطاف‌پذیر کنار هم قرار گیرند. جدول زیر برخی از کلاس‌های فلکس را براساس موضوعات مختلف معرفی می‌کند.

جدول ۷ـ کلاس‌های Flexbox

تراز افقی عناصر	تراز عمودی عناصر	جهت چیدمان	انتقال به سطر بعد
justify-content-start justify-content-center justify-content-end justify-content-between justify-content-around justify-content-evenly	align-items-start align-items-center align-items-end align-items-baseline	flex-row flex-row-reverse flex-column flex-column-reverse	flex-wrap flex-nowrap flex-wrap-reverse

کلاس‌های *-justify-content برای تعیین تراز افقی محتوا در یک عنصر نگهدارنده استفاده می‌شوند. سه کلاس اول، محتوای یک عنصر را در سمت چپ (start)، وسط (center) و راست (end) تراز می‌کنند. سه کلاس بعدی نحوه توزیع عناصر را بر اساس فاصله بین آنها مشخص می‌کنند.
کلاس‌های *-align-items برای تعیین تراز عناصر در راستای عمودی استفاده می‌شوند.

کار کارگاهی

به‌کارگیری کلاس‌های مختلف در لایه تصاویر محصولات

در این کارگاه کد لایه «جدیدترین محصولات» را توسط کلاس‌های بوت‌استرپ به شکل زیر تغییر خواهید داد.

۱ فایل index.html مربوط به واحد کار ۱ را باز کنید. کد لینک و اسکریپت CDN بوت‌استرپ ۵ را در مکان مناسب اضافه کنید.

۲ در عنصر <main>، تغییرات زیر را اعمال کنید:

```
<main>
  <section class="product-section mx-auto my-4">
      <h2 class="py-3 fw-bold text-center">جدیدترین محصولات</h2>
      <div class="products d-flex flex-wrap justify-content-center gap-3 p-3">
          ...
      </div>
```

۳ استایل‌های مربوط به products-section h2، products و ویژگی margin در products-section را از فایل styles.css حذف کنید.



کار کارگاهی

طراحی بخش معرفی فروشگاه

در این کارگاه یک بخش با محتوای متنی در انتهای صفحه فروشگاه ایجاد خواهید کرد. برای طراحی این قسمت از کلاس‌های مختلف بوت‌استرپ استفاده کنید.

در فایل index.html برچسب <article> را به شکل زیر قبل از برچسب </main> اضافه نمایید:

```
<article class="about-us mx-auto p-3 bg-light text-secondary text-end rounded my-5 border">
</article>
```

تیترها و پاراگراف‌های زیر را درون برچسب <article> اضافه کنید:

```
<h1 class="text-primary">فروشگاه اینترنتی مرجع نو</h1>
<p class="text-justify">پاراگراف اول</p>
<h2 class="text-primary">مرجع نو، کتاب های درسی و کمک درسی</h2>
<h2 class="text-primary">پشتیبانی حرفه ای مرجع نو؛همراه شما در تمام مراحل</h2>
<p class="text-justify">پاراگراف سوم</p>
```

فایل را ذخیره کنید و نتیجه را در مرورگر مشاهده کنید.

بوت‌استرپ نسخه ۳ دارای کلاسی به نام text-justify بود که برای تراز کردن دوطرفه متن (justify) به‌کار می‌رفت. این کلاس در بوت‌استرپ نسخه ۵ موجود نیست. بنابراین، برای ایجاد تراز دوطرفه متن در پاراگراف‌ها، لازم است کلاس سفارشی text-justify را با استفاده از کد زیر به فایل styles.css اضافه نمایید:

```
.text-justify{
    text-align:justify;
}
```

محتوای عناصر پاراگراف را بر اساس تصویر زیر تکمیل نمایید. برای نمایش کلمات به‌صورت ضخیم (bold)، از تگ مناسب استفاده کنید فایل را ذخیره و نتیجه را در مرورگر بررسی کنید.



برای اصلاح نمای موبایل، کلاس w-۷۵ را از عنصر article حذف کنید و استایل مربوط به آن را به فایل styles.css اضافه کنید:

```
article{
    width:75%;
}
```

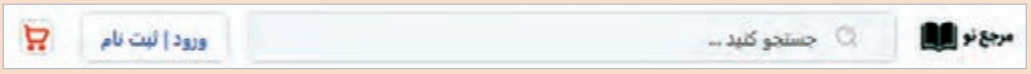
برای دستگاه‌هایی با پهنای کمتر از ۶۰۰ پیکسل قانون زیر را در فایل styles.css اضافه کنید:

```
@media (max-width:600px){
    article{
        width:90%;
    }
}
```

بیشتر بدانید

طراحی نوار ابزار حاوی لوگو، کادر جست‌وجو، گزینه‌های ورود | ثبت نام و سبد خرید

در این کارگاه ناحیه‌ای مشابه تصویر زیر را در بالای نوار منو ایجاد خواهید کرد:



یک لوگو با ارتفاع ۷۰ پیکسل ایجاد کنید و سپس فایل تصویر را با نام logo.jpg در پوشه images سایت ذخیره نمایید.

درون کادر جست‌وجو باید یک آیکن جست‌وجو ظاهر شود. برای دانلود آیکن جست‌وجو از سایت svgrepo.com استفاده کنید و عبارت search را در آن جست‌وجو کنید. آیکن دلخواه خود را توسط دکمه Edit vector در کوچک‌ترین اندازه و رنگ دلخواه تنظیم کرده و سپس روی دکمه Export کلیک کنید. فایل آیکن را با نام search.svg در پوشه تصاویر سایت (images) ذخیره نمایید.

تصویر صفحهٔ قبل حاوی لوگو، کادر جست‌وجو (شامل آیکن جست‌وجو و کادر متنی)، گزینه ورود| ثبت‌نام و سبد خرید است. این عناصر را به شکل زیر به صفحه index.html قبل از عنصر nav اضافه کنید:

```
<div class="toolbar ">
    <img src="./images/logo.png" alt="logo" class="logo ms-2 ">
    <div class="search-box border border-1 rounded px-4 bg-light d-flex flex-row align-items-center">
        <img src="./images/search.svg" alt="جستجو">
        <input type="text" placeholder="جستجو ..." class="bg-light border-0 w-100 p-2" style="outline: none" />
    </div>
    <div class="signin-signup border rounded py-2 px-3 mx-3 ">
        <a href="./signup-form.html" class="text-decoration-none"> ورود | ثبت نام </a>
    </div>
    <div class="shopping-cart border rounded me-2 ">
        <a href="./shopping-cart.html"><img src="./images/shopping-cart.svg"></a>
    </div>
</div>
```

استایل عناصر را در نمای دسکتاپ به شکل زیر تنظیم کنید:

```
.toolbar{
    max-width: 1200px;
}
.search-box{
    max-width: 700px;      /* حداکثر پهنای کادر جستجو */
    flex: 1;               /* اشغال شدن فضای اضافه درون سطر توسط کادر جستجو */
}
.search-box img{
    width: 35px;          /* پهنای آیکن جستجو */
}
.shopping-cart img{
    width: 25px;          /* پهنای تصویر سبد خرید */
}
@media (max-width:600px) {
    .logo, .signin-signup{     /* تنظیمات عناصر در نمای موبایل */
        display: none;         /* مخفی سازی لوگو و گزینه ورود|ثبت نام */
    }
    .search-box{
        width: 70%;       /* تغییر پهنای کادر جستجو در نمای موبایل */
        margin-left: 10px;    /* فاصله کادر جستجو تا آیکن سبد خرید */
    }
}
```

کلاس‌های زیر را به عنصر toolbar اضافه کنید. سپس فایل را ذخیره کرده و نتیجه را در مرورگر بررسی نمایید.

```
<div class="toolbar d-flex justify-content-around align-items-center py-2 px-3 mx-auto"></div>
```

با تنظیمات CSS فوق، گزینه «**ورود|ثبت‌نام**» و **لوگو** از نمای موبایل حذف می‌شود، بنابراین این گزینه‌ها باید در جای دیگری نمایش داده شود. بهتر است این گزینه‌ها به‌صورت یک لایهٔ ثابت در پایین صفحه نمایش داده شوند. بدین منظور کد زیر را قبل از </main> وارد کنید:

```
<div class="bottom-toolbar w-100 fixed-bottom bg-white justify-content-around align-items-center px-4">
    <a href="./index.html"> خانه </a>
    <img src="./images/marjaeNo-logo.jpg" alt="لوگوی مرجع نو">
    <a href="./login.html"> ورود | ثبت نام</a>
</div>
```

کلاس fixed-bottom لایه bottom-toolbar را در پایین صفحه ثابت نگه می‌دارد.
استایل زیر را برای عنصر bottom-toolbar بنویسید:

```
.bottom-toolbar {
  height: 50px;
  box-shadow: 0 -3px 10px lightgray;
  display: none;  /* مخفی کردن لایه در نمای تبلت و دسکتاپ */
}
```

استایل زیر را برای نمایش صحیح نوار ابزار در نمای موبایل اضافه کنید:

```
@media (max-width:600px) {
  .bottom-toolbar {
    display: flex;
  }
  .bottom-toolbar a {
    text-decoration: none;
  }
}
```

با این تنظیمات عنصر bottom-toolbar به‌صورت شکل زیر در پایین دستگاه موبایل ظاهر می‌شود.



برای فاصله گرفتن منو از toolbar، از کلاس mt-3 در برچسب <nav> استفاده کنید.
با توجه به اینکه گزینه‌های **ورود|ثبت نام** و **سبد خرید** به بخش toolbar اضافه شده است، بنابراین عنصر user-actions را از منوی ناوبری (عنصر top-menu) حذف کنید. سپس استایل‌های مربوط به آن را از فایل styles.css حذف کنید. با حذف شدن ناحیه user-action نیازی به ویژگی‌های فلکس در کلاس top-menu. نیست. پس ویژگی‌های فلکس را هم حذف کنید. در انتها آنچه درون کلاس top-menu باقی می‌ماند ویژگی max-width خواهد بود.

```
.top-menu {
  max-width: 1200px;
}
```

بهتر است که گزینه‌های منو در راستای افقی در وسط صفحه توزیع شوند، برای این کار از کلاس justify-content-center در عنصر ul استفاده کنید:

```
<nav class="top-menu mt-3 mx-auto">
    <ul class="d-flex justify-content-center p-0 m-0 list-unstyled">
```

توسط کد فوق تمام ویژگی‌های انتخاب‌گر top-menu ul. با کلاس‌های بوت‌استرپ معادل جایگزین شد. بنابراین ویژگی‌های درون انتخابگر top-menu ul. را از فایل styles.css حذف کنید.
انتظار می‌رود که با تنظیمات فوق نتیجه زیر حاصل شود:



سفارشی‌سازی کلاس‌های بوت استرپ

کلاس‌های بوت‌استرپ دارای ویژگی‌ها و مقادیر از پیش تعیین شده‌ای هستند که امکان طراحی سریع عناصر صفحه را فراهم می‌کنند. گاهی لازم است که عملکرد این کلاس‌ها توسط توسعه‌دهندهٔ صفحات تغییر نماید. این کار توسط CSS و SASS[1] امکان‌پذیر است. برای تغییر استایل یک کلاس توسط CSS لازم است که ویژگی کلاس مورد نظر با مقادیر جدید تنظیم شود. در حالت پیش‌فرض اعمال کلاس بوت‌استرپ دارای اولویت بالاتری است. بنابراین برای بالا بردن اهمیت کلاس سفارشی باید از کلمه important ! در انتهای مقدار جدید استفاده شود، در غیر این‌صورت مقدار جدید اعمال نمی‌شود.

کار کارگاهی

تغییر استایل کلاس رنگ بوت‌استرپ

در این کارگاه رنگ کلاس bg-light را در فایل index.html تغییرخواهید داد. بدین منظور استایل زیر را برای کلاس bg-light به شکل زیر بنویسید:

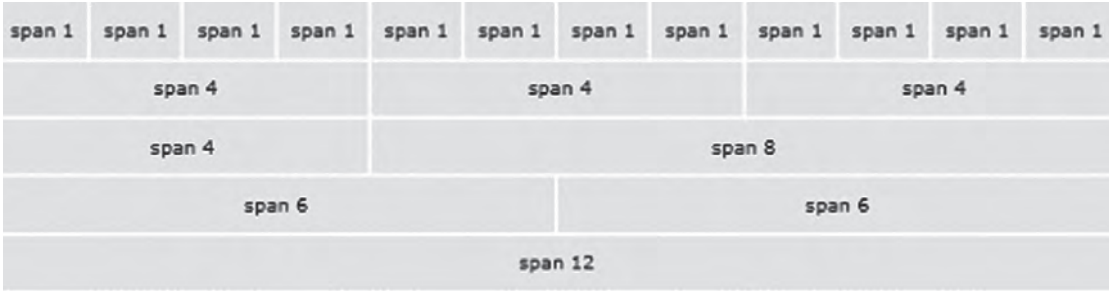
```
.bg-light {
  background-color: #f0f8f8 !important;
}
```

الحاق فایل styles.css باید بعد از الحاق فایل bootstrap.min.css انجام شود، در غیر این‌صورت اولویت اجرا با تنظیمات پیش فرض کلاس بوت استرپ خواهد بود. کلاس bg-light در دو عنصر کادر جست‌وجو و article مورد استفاده قرار گرفته است. بعد از ذخیره‌سازی کد فوق بررسی کنید که آیا تغییر رنگ در این دو عنصر اعمال شده است یا خیر.

۱ـ ابزاری است که نوشتن، سازمان‌دهی و نگهداری کدهای CSS را در پروژه‌های وب ساده‌تر و حرفه‌ای‌تر می‌کند.

شبکه‌بندی صفحه (Grid Systems)

بوت‌استرپ دارای یک سیستم شبکه‌بندی با نام Grid است که توسط Flexbox ایجاد شده است و امکان تقسیم پهنای صفحه را به ۱۲ستون فراهم می‌کند. هر سلول در این شبکه می‌تواند محتوای بخشی از صفحه را درون خود نگه دارد. سیستم Grid امکان گروه‌بندی ستون‌ها را برای ایجاد ستون‌های عریض‌تر فراهم می‌کند به‌طوری‌که فضای کافی برای قرار دادن محتوا درون یک ستون موجود باشد. سیستم Grid واکنش‌گرا است و بسته به اندازه صفحه، ستون‌ها را به‌طور خودکار مرتب می‌کند.



استفاده از کلاس‌های شبکه‌بندی (rows و columns)

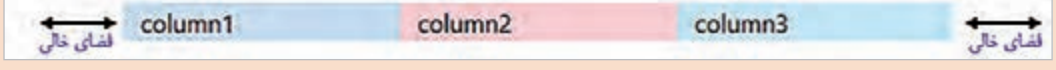
برای ایجاد سطرها و ستون‌های شبکه از کلاس‌های row و col استفاده می‌شود. کلاس row باعث می‌شود سلول‌های شبکه در یک خط افقی قرار بگیرند و فواصل مناسب بین آنها اعمال شود. پهنای ستون‌ها به‌صورت خودکار توسط پارامتر size در نام کلاس col مشخص می‌شود. شکل کلی کلاس col به صورت col-breakpoint-size است، پارامتر breakpoint به اندازه دستگاه (sm, md, lg,xl, xxl) و پارامتر سایز به اندازه ستون اشاره دارد. به‌عنوان مثال col-sm-3 یعنی اشغال سه ستون از ۱۲ ستون در تمام دستگاه‌های بزرگ‌تر یا مساوی ۵۷۶ پیکسل.

شبکه سطرها و ستون‌ها باید درون یک عنصر نگهدارنده قرارگیرد. بوت‌استرپ دارای دو کلاس با اسامی container و container-fluid است که برای ایجاد نگهدارنده استفاده می‌شوند. تفاوت این دو نگهدارنده در میزان حاشیه چپ و راست آنهاست. کلاس container یک فضای خالی با پهنای مساوی در چپ و راست عنصر نگهدارنده ایجاد می‌کند، در حالی‌که کلاس container-fluid تمام پهنای صفحه را اشغال می‌کند.

مثال

در مثال زیر یک سطر با سه ستون ایجاد می‌شود که سهم هر ستون برابر ۴ از ۱۲ خواهد بود:

```
<div class="container">
    <div class="row">
        <div class="col bg-primary-subtle"> column1 </div>
        <div class="col bg-danger-subtle"> column2 </div>
        <div class="col bg-info-subtle"> column3 </div>
    </div>
</div>
```



ستون‌های مثال صفحه قبل در تمام دستگاه‌ها در کنار هم نمایش داده می‌شوند. اگر به جای کلاس col از کلاس col-sm-4 استفاده شود، این ستون‌ها در تمام دستگاه‌های بزرگ‌تر یا مساوی ۵۷۶ پیکسل در کنار هم خواهند بود، اما در دستگاه‌های کوچک‌تر، به‌صورت پشته روی هم قرار می‌گیرند. پارامترهای breakpoint و size مشخص می‌کنند که نحوه نمایش ستون‌ها در دستگاه‌های مختلف (موبایل، تبلت، دسکتاپ) چگونه باشد.

سیستم grid دارای کلاس‌های مختلفی برای اندازه ستون‌ها در دستگاه‌های مختلف است که در جدول زیر به آنها اشاره شده است:

جدول ۸ـ پارامترهای کلاس col- برای دستگاه‌های مختلف

پارامتر	کاربرد	توضیح
بدون پیشوند	برای دستگاه‌های بسیار کوچک ـ پهنای صفحه کمتر از ۵۷۶ پیکسل	col-
sm-{n}	برای دستگاه‌های کوچک ـ پهنای صفحه بزرگ‌تر یا مساوی ۵۷۶ پیکسل	col-sm-
md-{n}	برای دستگاه‌های متوسط ـ پهنای صفحه بزرگ‌تر یا مساوی ۷۶۸ پیکسل	col-md-
lg-{n}	برای دستگاه‌های بزرگ ـ پهنای صفحه بزرگ‌تر یا مساوی ۹۹۲ پیکسل	col-lg-
xl-{n}	برای دستگاه‌های بزرگ ـ پهنای صفحه بزرگ‌تر یا مساوی ۱۲۰۰ پیکسل	col-xl-
xxl-{n}	برای دستگاه‌های بزرگ ـ پهنای صفحه بزرگ‌تر یا مساوی ۱۴۰۰ پیکسل	col-xxl-

{n} عددی از ۱ تا ۱۲ است. منظور از n تعداد ستون‌هایی است که یک عنصر از ۱۲ ستون grid اشغال می‌کند.

مثال

در مثال زیر یک ردیف با دو ستون در دو اندازه $\frac{۴}{۱۲}$ و $\frac{۸}{۱۲}$ ایجاد می‌شود:

```
<div class="row">
    <div class="col-sm-4"> .col-sm-4 </div>
    <div class="col-sm-8"> .col-sm-8 </div>
</div>
```

این دو ستون در دستگاه‌هایی با پهنای بزرگ‌تر یا مساوی ۵۷۶ پیکسل در کنار هم و با اندازه‌های $\frac{۴}{۱۲}$ و $\frac{۸}{۱۲}$ نمایش داده می‌شوند. اما در دستگاه‌های کوچک‌تر، محتوای ستون‌ها روی هم قرار می‌گیرند و در نتیجه هر ستون ۱۰۰٪ پهنای صفحه را اشغال خواهد کرد.



پهنای دستگاه: بزرگ‌تر یا مساوی ۵۷۶ پیکسل پهنای دستگاه: کمتر از ۵۷۶ پیکسل

مثال

برای ایجاد طرح‌بندی‌های پویا و انعطاف‌پذیر می‌توان کلاس‌های فوق را به‌صورت ترکیبی استفاده نمود. مانند مثال زیر:

```
<div class="row">
   <div class="col-sm-9  col-lg-4  bg-warning"> .col-sm-9  .col-lg-4 </div>
   <div class="col-sm-3  col-lg-8 bg-info"> .col-sm-3  .col-lg-8 </div>
</div>
```

این دو ستون در دستگاه‌های کوچک‌تر از ۵۷۶ پیکسل زیر هم نمایش داده می‌شوند؛ در دستگاه‌هایی با پهنای بزرگ‌تر از ۵۷۶ پیکسل و کوچک‌تر از ۹۹۲ پیکسل در اندازه $\frac{۹}{۱۲}$ و $\frac{۳}{۱۲}$ و در دستگاه‌های بزرگ‌تر با اندازه $\frac{۴}{۱۲}$ و $\frac{۸}{۱۲}$ نمایش داده می‌شوند.

کار کارگاهی

طرح‌بندی فوتر صفحه وب

در فایل index.html، ناحیه فوتر را به شکل زیر بعد از برچسب </main> ایجاد کنید:

```
<footer class="py-3">
   <div class="container mx-auto">
      <div class="row">
         <div class="footer-col col-sm-3 bg-danger-subtle"> column1 </div>
         <div class=" footer-col col-sm-3 bg-info-subtle"> column2 </div>
         <div class=" footer-col col-sm-3 bg-dark-subtle"> column3 </div>
         <div class=" footer-col col-sm-3 bg-primary-subtle"> column4 </div>
      </div>
   </div>
   <p> کلیه حقوق مادی و معنوی سایت متعلق به سایت مرجع تواست </p>
</footer>
```

برای محتوای ستون اول از کد زیر استفاده کنید:

```
<p class="fw-bold fs-5"> خدمات مشتریان  </p>
<p> <a href="#"> پرسش های متداول  </a> </p>
<p> <a href="#"> راهنمای خرید  </a> </p>
<p> <a href="#"> شرایط بازگشت محصول  </a> </p>
```

ستون دوم مربوط به آیکن شبکه‌های اجتماعی است. برای به‌دست آوردن کد آیکن‌ها وارد آدرس icons.getbootstrap.com شوید و در انتهای این صفحه، در بخش CDN لینک فایل CSS آیکن‌ها را کپی کنید و در بخش <head> بچسبانید. این لینک به شکل زیر است:

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/
                        bootstrap-icons@1.13.1/font/bootstrap-icons.min.css">
```

سپس به ابتدای صفحه آیکن‌ها برگردید و برای جست‌وجوی آیکن، در کادر filter نام شبکه اجتماعی مورد نظر را بنویسید. بعد از ظاهر شدن آیکن‌ها روی یک مورد به دلخواه کلیک کنید تا صفحه اختصاصی آیکن باز شود. در بخش icon font کد html را کپی کنید و آن را در ستون دوم فوتر بچسبانید. محتوای ستون دوم را به شکل زیر تنظیم کنید:

```
<p class="fw-bold fs-5"> شبکه های اجتماعی </p>
<div>
    <a href="#"> <i class="bi bi-instagram fs-4"> </i> </a>
    <a href="#"> <i class="bi bi-twitter fs-4 me-4"> </i> </a>
    <a href="#"> <i class="bi bi-telegram fs-4 me-4"> </i> </a>
    <a href="#"> <i class="bi bi-linkedin fs-4 me-4"> </i> </a>
</div>
```

در ستون سوم کد شکل زیر را وارد کنید:

```
<p class="fw-bold fs-5"> درباره ما </p>
<p> <a href="#"> مرجع نو در یک نگاه </a> </p>
<p> <a href="#"> اهداف و تعهدات </a> </p>
<p> <a href="#"> سوالات متداول </a> </p>
```

در ستون چهارم کد شکل زیر را بنویسید:

```
<p class="fw-bold fs-5"> ارتباط با ما </p>
<p> 021-12345678 :تلفن </p>
<p> ایمیل: info@marjeno.com </p>
<p> ساعت پاسخگویی: شنبه تا چهارشنبه 9 الی 17 </p>
```

برای حذف زیرخط پیوندها و تنظیم رنگ و اندازه متون فوتر، استایل شکل زیر را به فایل styles.css اضافه کنید:

```
footer a:link,a:visited{
    color: black;
}
footer a, footer p{
    font-size: 14px;
}
```

برای تنظیم استایل پاراگراف «کلیه حقوق مادی و معنوی ....» از کد شکل زیر استفاده کنید:

```
footer>p {
  text-align: center;
  border-top: 1px solid lightgray;
}
```

این استایل متن پاراگراف را در وسط صفحه به همراه یک خط در بالای آن نمایش می‌دهد. استایل فوق فقط روی پاراگراف آخر اعمال می‌شود، چرا که تنها این پاراگراف فرزند مستقیم عنصر footer است.

برای تغییر تراز عناصر در نمای موبایل و جابه‌جا شدن محل نمایش پاراگراف آخر، استایل شکل زیر را اضافه کنید:

```
@media (max-width:600px){
    .footer-col{
        text-align:center;    /* تنظیم عناصر فوتر در وسط صفحه موبایل */
        padding:10px 0;       /* ایجاد فضای خالی اضافی در بالا و پایین هر ستون فوتر */
    }
    body{
        padding-bottom: 50px;    /* بالا آوردن متن کپی رایت */
    }
    footer>p {
        padding-top: 20px;    /* تنظیم فاصله پاراگراف تا خط بالای آن */
    }
}
```

به‌دلیل حضور bottom-navbar در پایین صفحه، امکان مشاهده پاراگراف انتهایی فوتر در نمای موبایل فراهم نیست؛ به همین دلیل موقعیت پاراگراف توسط ویژگی position و bottom تنظیم شده است.

بعد از اینکه همه عناصر به‌درستی در جای خود قرار گرفتند، رنگ زمینهٔ ستون‌ها را حذف کنید.

واکنش‌گرایی و کلاس‌های مرتبط

واکنش‌گرایی به‌معنای توانایی یک صفحه وب برای تغییر و سازگاری با اندازه‌ها و ویژگی‌های مختلف دستگاه‌ها و صفحه نمایش‌ها است. در یک طراحی واکنش‌گرا عناصر وب‌سایت به‌طور خودکار اندازه و چیدمان خود را تغییر می‌دهند تا مناسب با اندازهٔ صفحه نمایش کاربر باشند. هدف از انجام یک طراحی واکنش‌گرا (Responsive) این است که تجربه کاربری بهتری برای بازدیدکنندگان فراهم شود. برای طراحی عناصر واکنش‌گرا، روش‌های مختلفی به شرح زیر موجود است:

۱ Media query: برای تنظیم ویژگی‌های عناصر در دستگاه‌های مختلف

۲ Flexbox: برای مدیریت نحوه چینش عناصر و فضای بین آنها

۳ CSS Grid: برای سازمان‌دهی عناصر در سطرها و ستون‌ها

۴ استفاده از max-width به جای width برای اینکه تصاویر در اندازه‌های مختلف به‌درستی نمایش داده شوند:

```
img {
  max-width: 100%;
  height: auto;
}
```

۵ استفاده از واحدهای نسبی ٪، rem و em به‌جای استفاده از واحدهای ثابت (مانند px)

۶ به‌کارگیری فریم‌ورک‌هایی مانند بوت‌استرپ که دارای کلاس‌های واکنش‌گرای آماده هستند.

کنجکاوی

در مورد واحدهای rem و em تحقیق کنید.

کلاس‌های واکنش‌گرای بوت‌استرپ دارای موضوعات مختلفی هستند که در جدول ۹ به آنها پرداخته شده است:

جدول۹ــ کلاس‌های واکنش‌گرای بوت‌استرپ

نوع کلاس	نام کلاس‌ها
Container	container و container-sm، container-md، container-lg و ...
Grid	row، row-cols، col، col-sm، col-md و col-lg و ...
Layout	d-flex، d-*-flex، d-*-inline، d-*-block start، d-*-none مانند زیر: d-sm-none، d-md-block، flex-md-row، justify-content-sm-center، align-items-md- و ...
Margin	m-*-size مانند mb-sm-3 و mx-sm-۱
Padding	p-*-size مانند py-sm-3، px-lg-3
Display Classes	d-none ، d-block، d-*-none، d-*-block
Image	img-fluid
Components	button، form، navbar، cards، carousel و ...

در کلاس‌های جدول ۱۰ علامت * معادل lg،sm و ... است.

جدول۱۰ــ Breakpoints

پهنای صفحه	کلاس	Breakpoint
کمتر از 576 px	xs	Extra small
576 px و بیشتر	sm	Small
768 px و بیشتر	md	Medium
992 pxو بیشتر	lg	Large
1200 pxو بیشتر	xl	Extra large
1400 pxو بیشتر	xxl	Extra extra large

کار کارگاهی

کار با کلاس‌های واکنش‌گرای Bootstrap

در این کارگاه با به‌کارگیری کلاس‌های بوت‌استرپ تغییراتی را در صفحه index.html ایجاد خواهید کرد.

بخش اول استفاده از کلاس‌های row-cols: در کارگاه قبل اندازه ستون ها توسط کلاس col-sm-3 مشخص شد. این کلاس باعث می‌شود که در دستگاه‌های بزرگ‌تر از ۵۷۶ پیکسل، ستون‌ها در کنار هم نمایش داده شوند و در دستگاه‌های کوچک‌تر، به‌صورت پشته (stack) روی هم قرار گیرند.

بوت استرپ دارای کلاس واکنش‌گرای دیگری با نام row-cols است که امکان تعیین تعداد ستون‌ها (در هر ردیف) را در دستگاه‌های مختلف فراهم می‌کند؛ به‌گونه‌ای که اندازه ستون‌ها در عنصر row مشخص می‌شود، نه در عناصر col. جدول زیر نحوهٔ استفاده از این دو روش را نمایش می‌دهد.

روش تعیین اندازه ستون	توضیحات
تعیین اندازهٔ ستون در عناصر col	`<div class="row">` `  <div class="col-6"> Column A</div>` `  <div class="col-6"> Column B</div>` `</div>` این کد یک سطر (row) با دو ستون (col-6) ایجاد می‌کند. هر ستون نیمی از پهنای سطر ($\frac{۶}{۱۲}$) را اشغال می‌نماید. A | B
تعیین اندازه ستون در عناصر row	`<div class="row row-cols-2">` `  <div class="col bg-info-subtle text-center"> A </div>` `  <div class="col bg-danger-subtle text-center"> B </div>` `  <div class="col bg-warning-subtle text-center"> C </div>` `  <div class="col bg-primary-subtle text-center"> D </div>` `</div>` A | B C | D کلاس row-cols-2 باعث ایجاد ۲ ستون در هر سطر می‌گردد. در کد فوق ۴ عنصر col وجود دارد؛ بنابراین دو سطر، هر یک حاوی دو ستون تولید می‌شود. در صورت افزودن تعداد بیشتری از col، ستون‌ها به ردیف بعدی منتقل می‌شوند. Bootstrap به‌طور خودکار عرض ستون‌ها را تنظیم می‌کند؛ بنابراین نیازی به تعیین اندازه دستی برای هر ستون ندارید.

۱ در فایل index.html کدهای زیر را بعد از جدیدترین محصولات اضافه کنید:

```
<!-- دسته‌بندی‌های پر بازدید -->
<section class="most-visited mt-2 mb-5 text-center ">
    <h2 class="text-primary mb-5 pt-5 fw-bold border-bottom border-primary d-inline-block">
      دسته‌بندی‌های پر بازدید
    </h2>
    <div class="row row-cols-1 row-cols-sm-2 row-cols-lg-4 mx-auto w-50">
        <div class="col">
            <img src="./images/lesson.jpg" > <h3> </h3>
        </div>
        <div class="col">
            <img src="./images/assistance_edu.jpg" > <h3> </h3>
        </div>
        <div class="col">
            <img src="./images/entrance_exam.jpg" > <h3> </h3>
        </div>
        <div class="col">
            <img src="./images/story.jpg" > <h3> </h3>
        </div>
    </div>
</section>
```

فایل را ذخیره کنید و نتیجه را در مرورگر بررسی کنید. سپس مراحل زیر را انجام دهید و بعد از هر مرحله، نتیجه را مشاهده کنید.

۲ برای تمام تصاویر از کلاس‌های border، border-4 و rounded-circle استفاده کنید تا تصاویر به شکل گرد و دارای خط دور باشند.

۳ برای هر کلاس col کلاس mb-3 را اضافه کنید تا تصاویر در نمای موبایل از تصویر زیرین خود فاصله بگیرند.

۴ برای تنظیم اندازه تصویر، استایل شکل زیر را به فایل styles.css اضافه کنید.

```
.most-visited img {
  width: 100px;
}
```

عنوان هر محصول را داخل برچسب<h3> قرار دهید:

```
<img src="./images/mobile.jpg" class="border border-4 rounded-circle"><h3>کتاب درسی  </h3>
```

۵ برای تنظیم اندازه متن h3 و فاصله آن از تصویر بالا، استایل شکل زیر را بنویسید:

```
.most-visited h3{
    font-size: 16px;
    margin-top:10px;
}
```

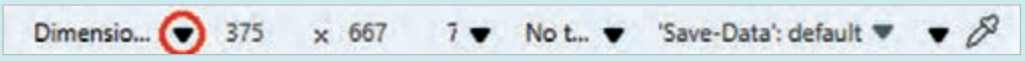
انتظار می‌رود که بعد از سرانجام مراحل قبل خروجی زیر در نمای دسکتاپ حاصل شود.



۶ برای مشاهده وضعیت تصاویر در دستگاه‌های مختلف، کلید f12 را بفشارید. در پاسخ به پنجره پیغام، دکمه Open Dev tools را انتخاب کنید. مطابق تصویر زیر از نوار ابزار بالای صفحه دکمه Toggle Device را انتخاب کنید:



۷ از کشوی Dimension نوع دستگاه را انتخاب کنید و سپس وضعیت تصاویر را مشاهده کنید.



۸ در صورت تمایل گزینه Responsive را از درون کشو انتخاب کنید و سپس با درگ کردن لبه سمت راست دستگاه، پهنای آن را به دلخواه تغییر دهید.

عملکرد کلاس‌های row-cols در کارگاه فوق به شرح زیر است:

۱-row-cols-sm: در دستگاه‌هایی با پهنای ۵۷۶ پیکسل و بیشتر، تصاویر محصولات در ۲ ستون نمایش داده می‌شود.

۲-row-cols-lg: در دستگاه‌هایی با پهنای ۹۹۲ پیکسل و بیشتر، تصاویر محصولات در ۴ ستون نمایش داده می‌شود.

بخش دوم استفاده از کلاس‌های container

کلاس‌های container واکنش‌گرا هستند و پهنای متفاوتی را در دستگاه‌های مختلف برای عنصر نگهدارنده ایجاد می‌کنند.

کلاس‌های container و پهنای آنها در دستگاه‌های مختلف

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-large ≥1200px	XX-Large ≥1400px
.Container	100%	540px	720px	960px	1140px	1320px
.Container-sm	100%	540px	720px	960px	1140px	1320px
.Container-md	100%	100%	720px	960px	1140px	1320px
.Container-lg	100%	100%	100%	960px	1140px	1320px
.Container-xl	100%	100%	100%	100%	1140px	1320px
.Container-xxl	100%	100%	100%	100%	100%	1320px
.Container-fluid	100%	100%	100%	100%	100%	100%

در کارگاه «**طراحی بخش معرفی فروشگاه**» پهنای عنصر article در دستگاه دسکتاپ و موبایل به ترتیب ۷۵٪ و ۹۰٪ تعیین شد. در اینجا قصد داریم به‌جای استفاده از کدهای CSS از کلاس‌های واکنش‌گرای container استفاده کنیم. بدین منظور مراحل زیر را دنبال کنید:

در عنصر article از کلاس container-sm استفاده کنید:

```
<article class="container-sm mx-auto p-3 … border">
```

هر دو استایل عنصر article را از فایل styles.css حذف کنید.

فایل‌های HTML و CSS را ذخیره کنید و نتیجه را در مرورگر بررسی کنید. با این تغییر، پهنای عنصر article برای دستگاه‌هایی با پهنای صفحه ۵۷۶ پیکسل و بزرگ‌تر، توسط کلاس container تعیین می‌شود و برای دستگاه‌های کوچک‌تر، پهنای عنصر برابر ۱۰۰٪ خواهد بود.

بخش سوم کار با کلاس‌های display

در کارگاه «**طراحی Toolbar**» یک نوار ابزار در بالای صفحه، طراحی شد. گزینه‌های «ثبت نام|ورود» و لوگوی آن در نمای موبایل مخفی شد و در پایین صفحه درون یک لایه جدید با نام bottom-toolbar قرار گرفت. هدف این کارگاه این است که کنترل نمایش/عدم نمایش عناصر لایه‌های toolbar و bottom-toolbar به کلاس‌های d-* سپرده شود و کدهای CSS مربوط به آنها حذف گردد.

ابتدا کلاس d-flex و d-lg-none را به مجموعه کلاس‌های عنصر bottom-toolbar اضافه کنید:

```
<div class="bottom-toolbar d-flex d-lg-none fixed-bottom bg-white...">
```

ویژگی display:none را از مجموعه ویژگی‌های عنصر bottom-toolbar. در فایل styles.css حذف کنید.
در قانون media، انتخاب‌گر bottom-toolbar. را حذف کنید. فایل را ذخیره و نتیجه را در مرورگر بررسی کنید.
برای مخفی کردن دو عنصر «ثبت نام|ورود» و لوگو در پهنای کمتر از ۹۹۲ پیکسل، کلاس‌های d-none و d-lg-block را به عناصر signin-signup و logo در عنصر toolbar اضافه کنید:

```
<img src="./images/logo.jpg" alt="logo" class="logo d-none d-lg-block ms-2 ">
<div class="signin-signup border rounded py-2 px-3 mx-3 d-none d-lg-block ">
```

بعد از اضافه کردن کلاس‌های d-*، استایل زیر را از قانون media حذف کنید.

```
.logo, .signin-signup {
  display: none;    /* مخفی کردن لوگو و گزینه‌های ورود|ثبت نام */
}
```

فایل را ذخیره و نتیجه را در مرورگر بررسی کنید. انتظار می‌رود که در دستگاه‌های ۹۹۲ پیکسل و بزرگ‌تر، چهار عنصر لوگو، کادر جست‌وجو، گزینه «ورود|ثبت نام» و سبد خرید حضور داشته باشند، اما در دستگاه‌های کوچک‌تر، دو عنصر لوگو و «ورود|ثبت نام» مخفی شوند.
برای کنترل مقدار فضای اضافی در اطراف عنصر Toolbar، کلاس p-3 را به عنصر toolbar اختصاص دهید و سپس ویژگی padding را در کلاس toolbar حذف کنید. از کشوی Dimension در مرورگر، گزینهٔ Responsive را انتخاب کنید و سپس نتیجهٔ کار را بررسی کنید.

```
<div  class="toolbar p-3">
```

بخش چهارم ایجاد تصویر واکنش گرا

هدف این بخش اضافه کردن یک بنر به بالای عنصر toolbar است، به‌طوری‌که تصویر بنر نسبت به پهنای صفحه واکنش‌گرا باشد. برای این کار تصویر top-banner.jpg را در ابتدای برچسب <header> درج کنید و کلاس img-fluid را برای آن استفاده کنید:

```
<img src="./images/top-banner.gif" class="img-fluid w-100">
```

کلاس img-fluid یکی از کلاس‌های مهم Bootstrap برای واکنش‌گرا کردن تصاویر است و باعث می‌شود که تصویر متناسب با اندازهٔ والدش تغییر اندازه دهد؛ از عرض container بزرگ‌تر نشود و همچنین در موبایل، تبلت و دسکتاپ درست نمایش داده شود.

البته ایجاد تصویر واکنش‌گرا به استفاده از کلاس img-fluid خلاصه نمی‌شود. بهتر است در خصوص تصاویری که تمام پهنا را اشغال می‌کنند، نسخهٔ دیگری از تصویر برای دستگاه‌های کوچک‌تر طراحی شود. در مثال زیر سه نسخه از تصویر بنر برای موبایل، تبلت و دسکتاپ استفاده شده است. مرورگر با توجه به پهنای صفحه، تصویر مناسب را برای نمایش انتخاب می‌کند:

```
<picture>
  <source media="(max-width: 768px)" srcset="./images/top-banner-small.gif">
  <source media="(min-width: 769px) and (max-width: 992px)" srcset="./images/top-banner-medium.gif">
  <source media="(min-width: 993px)" srcset="./images/top-banner.gif">
  <img src="top-banner.gif" alt="banner" class="img-fluid w-100">
</picture>
```

اکثر مرورگرهای مدرن امروزی از برچسب picture پشتیبانی می‌کنند؛ بنابراین در مرورگرهای مدرن، تصاویری که در برچسب‌های <source> تعیین شده است، نمایش داده می‌شوند. در مرورگرهای قدیمی مانند اینترنت اکسپلورر ۱۱ و قدیمی‌تر، تصویر پیش‌فرض ظاهر می‌شود. در کد شکل بالا، تصویر پیش‌فرض در برچسب <img> مشخص شده است.

فعالیت

سه تصویر بنر در سه اندازه مختلف، برای نمایش در موبایل، تبلت و دسکتاپ طراحی کنید و کدنویسی نمایش آنها را مطابق با کد فوق انجام دهید. انتظار می‌رود خروجی به صورت شکل زیر حاصل گردد.



توجه

چنانچه لبه‌های بنر مماس با لبه‌های مرورگر نیست، مقدار padding و margin را برای عنصر header برابر صفر قرار دهید.

اجزای رابط کاربری (User Interface Components)

کاربران صفحات وب توسط اجزای مختلفی با صفحه وب تعامل و ارتباط دارند. این اجزا شامل دکمه‌ها، منوها، نوار جست‌وجو، آیکن‌ها، فرم‌ها، تصاویر، رنگ‌ها، فونت‌ها و سایر عناصر بصری هستند که تجربه کاربری را شکل می‌دهند. بوت‌استرپ دارای مجموعه‌ای از کلاس‌های آماده برای اجزای رابط کاربری مختلف مانند نوارناوبری، فرم، دکمه، منوی آکاردئون و... است که در این بخش به آنها پرداخته می‌شود.

Navbar

کلاس navbar برای ایجاد نوار ناوبری (Navigation Bar) یا منوی بالای صفحه استفاده می‌شود. این کلاس به‌همراه کلاس‌های دیگری همچون navbar-expand-* برای تعیین رفتار و ظاهر نوار ناوبری در اندازه‌های مختلف صفحه نمایش به کار می‌رود. مجموعهٔ کلاس‌های Navbar به شرح زیر است:

navbar: کلاس اصلی برای ایجاد نوار ناوبری است.

navbar-expand-*: این کلاس‌ها تعیین می‌کنند که نوار ناوبری در چه پهنایی به صورت افقی ظاهر شود و در چه پهنایی جمع شود. کاراکتر * می‌تواند جایگزین عبارات sm، md، lg یا xl شود. به‌طور مثال استفاده از navbar-expand-sm باعث می‌شود که نوار ناوبری در دستگاه‌هایی با عرض ۵۷۶ پیکسل و بیشتر، به‌صورت افقی ظاهر شود و در دستگاه‌های کوچک‌تر جمع شده و به شکل دکمه همبرگری (☰) نمایش داده شود.

navbar-brand: برای نمایش لوگو یا نام برند در نوار ناوبری استفاده می‌شود.

navbar-nav: برای ایجاد لیست ناوبری (Navigation list) در نوار ناوبری استفاده می‌شود.

nav-link: برای لینک‌های موجود در لیست ناوبری استفاده می‌شود.

navbar-toggler: برای دکمهٔ کوچک منو استفاده می‌شود. این دکمه در نمای موبایل ظاهر می‌شود تا کاربر با کلیک روی آن، محتوای منو را مشاهده نماید و با کلیک مجدد منو را ببندد. (دکمه‌ای برای کنترل باز و بسته شدن منو ایجاد می‌کند)

navbar-toggler-icon: به منظور نمایش آیکن همبرگر (☰) برای دکمه منو در حالت جمع‌شده به‌کار می‌رود.

navbar-collapse: برای گروه‌بندی و مخفی کردن گزینه‌های منو در دستگاه‌های کوچک استفاده می‌شود.

dropdown: برای ایجاد منوهای کشویی (پایین افتادنی) در نوار ناوبری به‌کار می‌رود.

navbar-light و navbar-dark: برای تعیین رنگ محتوای نوار ناوبری استفاده می‌شود تا خوانایی عناصر درون آن حفظ شود. کلاس navbar-light موجب ایجاد یک نوار ناوبری با پس زمینه روشن و رنگ متن تیره می‌گردد و کلاسnavbar-dark یک نوار ناوبری با پس‌زمینهٔ تیره و متن‌های روشن ایجاد می‌کند. انتخاب صحیح این کلاس‌ها باعث افزایش خوانایی و بهبود تجربه کاربری می‌شود.

کار کارگاهی

طراحی نوار ناوبری

در این کارگاه، نوار ناوبری بوت‌استرپ را با قابلیت جمع شدن، به صفحه index.html اضافه خواهید کرد. قبل از انجام این کار، کد مربوط به top-menu را حذف کنید یا آن را به‌طور کامل درون علائم <!-- --> قرار دهید تا مرورگر محتوای آن را تفسیر نکند.

عملکرد صحیح نوار ناوبری نیازمند فایل bootstrap.bundle.min.js است. اگر این فایل در بخش <head> یا قبل از </body> موجود نیست، آن را مطابق توضیحات استفاده از CDN در صفحه وب به صفحه اضافه کنید و گرنه با کلیک روی دکمه همبرگری (☰)، محتوای آن نمایش داده نخواهد شد.

کدهای شکل زیر را جایگزین عنصر top-menu نمایید:

```
<nav class="navbar navbar-expand-lg navbar-dark bg-primary">
    <div class="container-fluid">
        <button type="button" class="navbar-toggler" data-bs-toggle="collapse"
                                                     data-bs-target="#navbarCollapse">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarCollapse"> <!--محل نوشتن لیست--> </div>
    </div>
</nav>
```

کد بالا شامل دو عنصر «دکمه» و «ناحیه قابل جمع شدن» است. برای ایجاد دکمه از برچسب <button> استفاده شده است. درون این برچسب یک عنصر با کلاس navbar-toggler-icon قرار دارد که موجب نمایش یک آیکن سه خطی می‌گردد.

محتوای لیست درون یک عنصر نگهدارنده با شناسه navbar Collapse قرار گرفته است. انتظار می‌رود که با کلیک روی دکمه منو، ناحیه حاوی لیست باز شود. بدین منظور از ویژگی data-bs-target با مقدار #navbarCollapse استفاده شده است.

data-bs-target="#navbarCollapse" شناسه (ID) منویی را مشخص می‌کند که باید باز یا بسته شود.

ویژگی data-bs-toggle="collapse" باعث می‌شود که با کلیک روی دکمه، عمل باز و بسته شدن منو انجام شود.

رنگ نوار ناوبری توسط bg-primary تعیین شده است در صورت تمایل می‌توانید از سایر کلاس‌های رنگ زمینه استفاده نمایید.

کلاس collapse باعث می‌شود که محتوای داخل div به‌صورت جمع‌شده (collapsed) باشد و با دکمه toggle باز/بسته شود. (این کلاس قابلیت باز و بسته شدن محتوا را فعال می‌کند.)

کلاس navbar-collapse باعث می‌شود که هنگام باز شدن منو، لینک‌ها به‌صورت عمودی (زیر هم) قرار گیرند و مقدار فضای داخلی (padding) منو به‌صورت استاندارد و هماهنگ با سایر اجزای ناوبری باشد. در صورتی‌که فقط از collapse استفاده شود لینک‌های منو ظاهر نامرتبی خواهند داشت.

حالا نوبت نوشتن لیست عناصر منو است. کد شکل زیر را در محل تعیین‌شده در کد قبل اضافه کنید:

```
<ul class="navbar-nav mx-auto mb-2 mb-lg-0">
    <li class="nav-item"> <a class="nav-link active" href="#">باشگاه مشتریان</a> </li>
    <li class="nav-item"> <a class="nav-link" href="#">وبلاگ</a> </li>
    <li class="nav-item"> <a class="nav-link" href="#">پیشنهاد طلایی</a> </li>
    <li class="nav-item"> <a class="nav-link" href="#">آموزش خرید</a> </li>
    <li class="nav-item"> <a class="nav-link" href="#">پیگیری سفارش</a> </li>
</ul>
```

در عنصر والد لیست (ul) از کلاس navbar-nav و در عناصر لیست از کلاس nav-item استفاده شده است. در حالت پیش‌فرض گزینه‌های منوی ناوبری در سمت راست صفحه ظاهر می‌شود. برای تنظیم حاشیه چپ و راست منو از کلاس mx-auto استفاده شده است.

ـ فایل را ذخیره کنید و نتیجه را در مرورگر بررسی کنید.

کلاس navbar-expand-lg باعث می‌شود که نوار منو در پهنای بالاتر از ۹۹۲ پیکسل به‌صورت افقی ظاهر گردد. پارامتر lg را به md تبدیل کنید و نتیجه تغییر را بررسی کنید.

در سایت‌های فروشگاهی، منوی ناوبری در نمای موبایل و تبلت پنهان می‌شود، کلاس‌های d-none و d-md-block را به عنصر nav اضافه کنید و نتیجه را در نمای موبایل و تبلت بررسی کنید.

کنجکاوی

کلاس active که فقط روی گزینه اول اعمال شده است چه تأثیری در ظاهر گزینه دارد؟

ـ گزینه اول اغلب سایت‌های فروشگاهی «دسته‌بندی محصولات» است که حاوی لیستی از طبقه‌بندی محصولات می‌باشد. برای ایجاد این گزینه، کد شکل زیر را قبل از اولین عنصر لیست فوق (باشگاه مشتریان) اضافه کنید:

```
<li class="nav-item dropdown">
   <a class="nav-link active dropdown-toggle" href="#" id="navbarDropdown" role="button"
                                              data-bs-toggle="dropdown">دسته‌بندی محصولات
   </a>
   <ul class="dropdown-menu">
       <li><a class="dropdown-item" href="#">کتاب درسی</a></li>
       <li><a class="dropdown-item" href="#">کتاب کمک درسی</a></li>
       <li><a class="dropdown-item" href="#">کتاب کنکور</a></li>
       <li><a class="dropdown-item" href="#">کتاب داستان</a></li>
   </ul>
</li>
```

nav-item: برای هر یک از گزینه‌های منو به‌کار می‌رود.

dropdown: این کلاس برای عنصر حاوی زیرمنو به‌کار می‌رود.
nav-link: برای استایل‌دهی به لینک‌های نوار ناوبری (Navbar) استفاده می‌شود و لینک را مطابق استایل Navbar نمایش می‌دهد.
active: این کلاس ظاهر یک گزینه را متفاوت (ضخیم‌تر) از سایر گزینه‌ها نمایش می‌دهد و به‌معنای فعال بودن یک گزینه است.
dropdown-toggle: این کلاس به جاوااسکریپت بوت‌استرپ اطلاع می‌دهد که این عنصر قابلیت باز و بسته شدن دارد، همچنین یک فلش کوچک ▼در کنار گزینه منو اضافه می‌کند که به‌معنای داشتن زیر منو است.
data-bs-toggle="dropdown": برای فعال‌سازی عملکرد دو حالته منو (باز و بسته) کاربرد دارد.
dropdown-menu: این کلاس معمولاً برای عنصر <ul> به‌کار می‌رود و باعث می‌شود که محتوای لیست به‌صورت یک جعبه پنهان در زیر گزینه منو قرار گیرد.
dropdown-item: برای تعیین استایل هر یک از گزینه‌های منوی کشویی کاربرد دارد.
کلاس active را از گزینه اول لیست مرحلهٔ قبل(باشگاه مشتریان)، حذف کنید. فایل را ذخیره کنید. انتظار می‌رود که نتیجهٔ زیر حاصل شود:



همان‌طورکه گفته شد، کلاس dropdown-toggle باعث نمایش فلش رو به پایین می‌شود. در صورت تمایل می‌توانید آن را حذف کنید، اما لازم است که با افزودن استایل زیر، نمایش زیرمنو را در حالت hover فعال کنید:

```
.bg-light {
  background-color: #f0f8f8 !important;
}
```

نکته

با توجه به اینکه رویداد hover در نمای موبایل اتفاق نمی‌افتد، پس این کد فقط در نمای دسکتاپ عمل می‌کند.

دکمه‌ها

دکمه یکی از عناصر تعاملی صفحه است که امکان انجام یک عملیات را با کلیک کاربر فراهم می‌کند. ایجاد دکمه در صفحه وب از طریق سه برچسب زیر امکان پذیر است:

۱ <button>
۲ <input>
۳ <a>

در بوت‌استرپ، کلاس‌های مختلفی برای استایل‌دهی به دکمه‌ها وجود دارد. این کلاس‌ها به شما امکان می‌دهند تا ظاهر دکمه‌ها را از نظر رنگ، اندازه، و سایر ویژگی‌ها سفارشی کنید. برای تعیین ظاهر دکمه توسط کلاس‌های بوت‌استرپ، استفاده از کلاس btn ضروری است. برای انتخاب رنگ و اندازه دکمه می‌توان کلاس‌های دیگر مربوط به دکمه را نیز به کلاس btn اضافه نمود.
این کلاس به‌صورت خلاصه در جدول ۱۱ آمده است:

جدول ۱۱ـ کلاس‌های دکمه در بوت استرپ

نوع کلاس	نام کلاس‌ها
رنگ	btn-primary، btn-secondary، btn-danger، btn-light، btn-dark، btn-link و...
خط دور دکمه‌ها	btn-outline-primary، btn-outline-secondary و....
اندازه دکمه‌ها	btn-sm، btn-lg
وضعیت غیرفعال	disabled

مثال

در مثال زیر شش دکمه با برچسب‌ها و کلاس‌های مختلف ایجاد شده است:

```
<input type="button" class="btn btn-success" value="Input button">

<a href="#" class="btn btn-danger" role="button"> Link button </a>

<button type="button" class="btn btn-secondary" disabled> Disabled button </button>
<button type="button" class="btn btn-outline-primary"> Outline button </button>
<button type="button" class="btn btn-info btn-sm"> Small button </button>
<button type="button" class="btn btn-primary btn-lg"> Large button </button>
```



بیشتر بدانید

طرح‌بندی صفحه اختصاصی محصول

در این کارگاه، صفحه نمایش اطلاعات محصول را در سه ستون حاوی تصویر محصول، ویژگی‌های محصول و قیمت محصول طراحی خواهید کرد. در این صفحه بخش header و footer مشابه فایل index.html است، فقط در بخش main دارای محتوای متفاوت است. جهت سرعت گرفتن کار طراحی، یک کپی از فایل index.html ایجاد کنید. برای این کار فایل index.html را در ویرایشگر متنی باز کنید. از منوی File گزینه Save As را انتخاب کنید و سپس عبارت product-details.html را به‌عنوان نام صفحه بنویسید.

محتوای برچسب <main> را به‌طور کامل پاک کنید. فایل را ذخیره کنید و صفحه product-details را در مرورگر به نمایش بگذارید تا از درستی عملکرد خود اطمینان حاصل کنید. در این مرحله صفحه product-details باید حاوی نوار بنر بالای صفحه، نوار لوگو، نوار ناوبری و فوتر باشد.
کد شکل زیر را برای طرح‌بندی صفحهٔ محصول درون برچسب <main> بنویسید:

```
<div class=" product-information container-sm px-1 mx-auto mt-5 ">
    <div class="row">
        <div class=" product-image col-md-4 col-lg-4"> تصویر محصول </div>
        <div class=" product-details col-md-4 col-lg-5"> اطلاعات محصول </div>
        <div class=" purchase-info col-md-4 col-lg-3"> اطلاعات فروش </div>
    </div>
</div>
```

در این طرح‌بندی از سیستم Grid استفاده شده است. سهم ستون‌ها در دستگاه‌های ۹۹۲ پیکسل و بزرگ‌تر برابر $\frac{۴}{۱۲}$، $\frac{۵}{۱۲}$ و $\frac{۳}{۱۲}$ است. در پهنای ۷۶۸ تا ۹۹۲ پیکسل سهم هر ستون برابر $\frac{۴}{۱۲}$ است و در دستگاه‌های کوچک‌تر محتوای ستون‌ها روی هم قرار می‌گیرند.
پهنای عنصر product-information را به شکل زیر در فایل styles.css تعریف کنید:

```
.bg-light {
  background-color: #f0f8f8 !important;
}
```

تصویر کتاب «دانش فنی پایه» را به شکل زیر به ستون اول اضافه کنید:

```
<img src="./images/Basic-Technical-Knowledge.jpg" class="img-fluid w-75">
```

برای اینکه تصویر محصول در تمام دستگاه‌ها در وسط ستون قرار گیرد، کلاس‌های d-flex، justify-content-center و align-items-center را به ستون اول، بعد از کلاس col-lg-4 اضافه کنید:

```
<div class=" product-image col-md-4 col-lg-4 d-flex align-items-center justify-content-center ">
```

برای اینکه اندازهٔ تصویر در دستگاه‌های کوچک‌تر از ۷۶۸ پیکسل کنترل شود، استایل زیر را به فایل styles.css اضافه کنید:

```
.product-image img{
  max-width:400px;
}
```

محتوای ستون دوم را به‌صورت شکل زیر بنویسید:
کلاس fs-3 اندازه متن و کلاس text-sm-end تراز متن را (در سمت راست ستون) تعیین می‌کند.

```
<div class="details-container">
    <h1 class="fs-3 text-sm-end">دانش فنی پایه</h1>
    <ul class="list-unstyled p-0">
        <li> سال چاپ: 1404 </li>
        <li> انتشارات: مرجع نو </li>
        <li>سری: کتاب درسی</li>
        <li> رشته تحصیلی: شبکه و نرم افزار</li>
        <li> پایه تحصیلی: دهم</li>
    </ul>
</div>
```

استایل محتوای ستون دوم را مطابق شکل زیر تنظیم کنید:

```
.product-details ul li{
  line-height:40px;
  font-size: 14px;
}
```

در نمای موبایل، محتوای ستون دوم به لبهٔ سمت راست مرورگر می‌چسبد و جلوه خوشایندی ندارد. برای اینکه محتوا از سمت راست فاصله بگیرد، به‌طوری‌که در وسط ستون قرار گیرد، محتوای ستون دوم، درون عنصر div با کلاس details-container قرار گرفته است. ویژگی‌های این عنصر را مطابق شکل زیر تنظیم کنید:

```
@media (max-width:768px){
.details-container{
  max-width: max-content;          /* تنظیم پهنا به اندازه محتوا */
  margin:20px auto 0;              /* قرار گرفتن محتوا در وسط */
}
}
```

کد نوشته شده در شکل زیر را درون ستون سوم وارد کنید:

```
<div class="border rounded p-2">
    <ul class="list-unstyled">
        <li>فروشنده: فروشگاه مرجع نو </li>
        <li>ارزیابی علمکرد: عالی</li>
        <li> وضعیت کالا: آماده ارسال</li>
        <li>قیمت: 210.000 تومان</li>
    </ul>
    <button class="btn btn-primary w-100 mt-4">افزودن به سبد خرید </button>
</div>
```

استایل محتوای ستون سوم را به مطابق شکل زیر بنویسید:

```
.purchase-info ul {
  background: #f5f7fd;
  margin: 0;
  padding: 0;
  border-radius: 10px;
}
.purchase-info ul li {
  height: 60px;
  line-height: 60px;
  padding: 0 20px;
  border-bottom: 5px solid #fff;
  background-color: #ebf3f5;
}
```

انتظار می‌رود که نتیجهٔ کار در دستگاه‌های ۹۹۲ پیکسل و بزرگ‌تر به شکل زیر باشد:



فعالیت

در فایل index.html محصول «دانش فنی پایه» را به صفحهٔ product-details.html پیوند دهید.

فرم‌ها

فرم‌های ورود اطلاعات، ابزارهایی برای تعامل کاربر با صفحه و جمع‌آوری اطلاعات هستند. ظاهر فرم‌های ورود، نقشی حیاتی در تجربهٔ کاربری و موفقیت یک وب‌سایت دارد. فرم‌های جذاب و کاربرپسند، باعث افزایش تعامل کاربران و ایجاد حس اعتماد و رضایت می‌شوند. در مقابل، فرم‌های نامرتب و گیج‌کننده می‌توانند منجر به سردرگمی کاربران، افزایش نرخ خروج از صفحه و در نهایت کاهش اعتبار وب‌سایت شوند.

بوت‌استرپ دارای مجموعه‌ای از کلاس‌های آماده برای طرح‌بندی انواع فرم‌هاست. به کمک این کلاس‌ها می‌توان فرم‌های واکنش‌گرا و جذاب را با سرعت بالا و کمترین نیاز به کدنویسی ایجاد نمود. جدول ۱۲ حاوی برخی کلاس‌های فرم در بوت‌استرپ ۵ است.

جدول ۱۲ــ کلاس‌های فرم

نام کلاس	کاربرد
form-control	تعیین استایل فیلدهای ورودی مانند input, textarea و select
form-label	تعیین استایل برچسب‌های فرم
form-check	استایل‌دهی به کادرهای تأیید و دکمه‌های رادیویی
form-check-label	این کلاس برای متن کنار کادر تأیید یا دکمه رادیویی استفاده می‌شود.
form-check-input	برای استایل‌دهی به کادر تأیید و دکمه رادیویی
form-text	این کلاس برای نمایش توضیح کمکی زیر یک فیلد فرم استفاده می‌شود.

بیشتر بدانید

طراحی فرم ثبت نام

در این کارگاه یک فرم ثبت‌نام را به شکل روبه‌رو توسط کلاس‌های بوت‌استرپ طراحی خواهید کرد.



۱ یک صفحه وب با نام signup-form.html ایجاد کنید و کدهای اولیه سند HTML را درون آن بنویسید. سپس متاتگ‌های charset، viewport، title و لینک CDN بوت‌استرپ را در بخش <head> به صفحه اضافه کنید. همچنین جهت نمایش صفحه را به‌صورت rtl تنظیم کنید.

```
<!DOCTYPE html>
<html dir="rtl">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sign Up Form</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
                                                                rel="stylesheet">
</head>
<body>
</body>
</html>
```

۲ در صفحه signup-form.html تنها یک فرم نمایش داده می‌شود. این فرم باید در راستای افقی و عمودی در وسط صفحه ظاهر گردد. بنابراین ویژگی‌های برچسب body را به شکل زیر تنظیم کنید:

```
<body class="d-flex align-items-center" style="height:100vh">
```

ـ vh یک واحد اندازه‌گیری در CSS است که مخفف Viewport Height است. کلمه viewport به بخشی از صفحه اشاره دارد که در معرض دید کاربر است. 1vh برابر ۱٪ ارتفاع viewport است. بنابراین، height:100vh به معنای آن است که ارتفاع عنصر <body> باید برابر ۱۰۰٪ ارتفاع viewport باشد. به عبارت دیگر، این عنصر باید تمام ارتفاع صفحه را اشغال کند.

ـ استفاده از کلاس‌های d-flex و align-items-center باعث می‌شود که محتوای بخش body در وسط صفحه قرار گیرد.

۳ درون عنصر body یک عنصر نگهدارنده با محتوای شکل زیر ایجاد کنید:

```
<div class="container">
    <div class="row justify-content-center">
        <div class="col-10 col-md-6 col-lg-5 shadow border rounded p-4">
            <h2 class="text-center mb-4">ورود | ثبت نام</h2>
            <form>
                <!-- محتوای فرم -->
            </form>
        </div>
    </div>
</div>
```

۴ پهنای فرم باید در دستگاه‌های مختلف کنترل شود. برای این‌کار بهتر است از کلاس‌های grid استفاده شود تا بسته به اندازهٔ دستگاه، پهنای عنصر نگهدارنده فرم تغییر نماید. در کد فوق یک عنصر نگهدارنده با کلاس row ایجاد شده است و درون آن تنها یک ستون حاوی فرم و عنوان آن موجود است. با توجه به اینکه فرم درون ستون grid قرار دارد، بنابراین پهنای فرم وابسته به پهنای ستون تغییر خواهد کرد.

کلاس justify-content-center باعث قرار گرفتن محتوای عنصر نگهدارنده در وسط صفحه خواهد شد.

کلاس‌های col-10، col-md-6 و col-lg-5 پهنای ستون grid را در دستگاه‌های موبایل، تبلت و دسکتاپ تعیین می‌کنند.

۵ محتوای فرم را به مطابق کدهای شکل زیر در محل تعیین شده در کد فوق، به صفحه اضافه کنید:

```
<div class="row mb-2">
    <div class="col-md-6 mb-3 mb-md-0">
        <label for="firstName" class="form-label"> نام </label>
        <input type="text" class="form-control" id="firstName" required>
    </div>
    <div class="col-md-6">
        <label for="lastName" class="form-label"> نام خانوادگی </label>
        <input type="text" class="form-control" id="lastName" required>
    </div>
</div>
<div class="mb-2">
    <label for="phone" class="form-label"> شماره تلفن </label>
    <input type="tel " class="form-control" id="phone" required>
</div>
<div class="mb-3">
    <label for="Password" class="form-label"> کلمه عبور </label>
    <input type="password" class="form-control" id="password" required>
    <div class="form-text"> ورود حداقل 6 کاراکتر الزامی است </div>
</div>
<div class="mb-3 form-check d-flex justify-content-start">
    <input type="checkbox" class="form-check-input ms-2" id="termsCheck">
    <label class="form-check-label" for="termsCheck"> من با قوانین و مقررات موافقم </label>
</div>
<button type="submit" class="btn btn-primary w-100 py-2"> ثبت نام </button>
<div class="text-center mt-3">
    <p>حساب فعال دارید؟ <a href="signup-form.html"> ورود </a></p>
</div>
```

فایل را ذخیره کنید و نتیجه را در مرورگر بررسی کنید.

۶ رنگ متن عنوان فرم را به رنگ آبی و رنگ متن برچسب‌ها را به رنگ خاکستری تغییر دهید. این کار را توسط کلاس‌های بوت‌استرپ انجام دهید.

۷ زیرخط پیوند «ورود» در پایین صفحه را توسط کلاس بوت‌استرپ حذف کنید.

بیشتر بدانید

Modal

مُدال یک رابط کاربری است که به‌صورت یک کادر محاوره‌ای شناور در بالای صفحه، نمایش داده می‌شود و کاربر را وادار می‌کند که قبل از بازگشت به صفحه وب، با آن تعامل داشته باشد. این عنصر معمولاً برای نمایش اطلاعات اضافی، تأیید کاربر، فرم‌ها یا تعاملات دیگر با کاربر استفاده می‌شود. فشردن کلید Esc، کلیک روی دکمه close یا ناحیه‌ای خارج از پنجره مدال باعث بسته شدن کادر محاوره‌ای می‌شود.

پنجرهٔ مدال دارای سه بخش header، body و footer است. این بخش‌ها توسط کلاس‌های modal-header، modal-body و modal-footer ایجاد می‌شوند. مجموعه این سه بخش درون یک عنصر نگهدارنده با کلاس modal-content قرار می‌گیرند.

مدال یک عنصر وابسته به جاوااسکریپت است، پس اضافه کردن فایل جاوااسکریپت بوت‌استرپ قبل از </body> ضروری است.

برای نمایش کادر محاوره‌ای نیاز به یک عنصر فعال‌کننده مانند دکمه است.

ساختار کامل مدال در Bootstrap

```
.modal
└─ .modal-dialog
    └─ .modal-content
        ├─ .modal-header (اختیاری)
        ├─ .modal-body
        └─ .modal-footer (اختیاری)
```

کار کارگاهی

نمایش پیغام اضافه شدن محصول به سبد خرید توسط Modal

در کارگاه‌های قبلی، صفحه نمایش اطلاعات محصول (product-details.html) را طراحی کردید. در برخی سایت‌های فروشگاهی، زمانی که کاربر روی دکمه «افزودن به سبد خرید» کلیک می‌کند، یک پیغام با محتوای «محصول مورد نظر به سبد خرید اضافه شد» نمایش داده می‌شود. در این کارگاه قصد داریم صفحه product-details.html را به همین طریق تغییر دهیم.

فایل product-details.html را باز کنید و دکمه افزودن به سبد خرید را به مطابق کد شکل زیر تغییر دهید:

```
<button class="btn btn-primary w-100 mt-4" data-bs-toggle="modal"
                    data-bs-target="#addTocartMsg">افزودن به سبد خرید</button>
```

با کلیک روی این دکمه کادر محاوره‌ای (مُدال) با شناسه addTocartMsg ظاهر می‌شود.

btn: استایل یک دکمهٔ بوت‌استرپ را به عنصر <button> اختصاص می‌دهد.

"data-bs-toggle="modal: مشخص می‌کند این دکمه برای باز کردن مدال است.

"data-bs-target="#addToCartMsg: مشخص می‌کند کدام مدال باز شود.

کد مُدال را بعد از برچسب </button> به صفحه اضافه کنید:

```
<div class="modal fade" id="addTocartMsg" tabindex="-1"
                aria-labelledby="MyModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header d-flex justify-content-end">
                <button type="button" class="btn-close ms-0" data-bs-dismiss="modal"
                                                        aria-label="Close">
                </button>
            </div>
            <div class="modal-body">
                محصول مورد نظر با موفقیت به سبد خرید اضافه شد
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-primary" >مشاهده سبد خرید</button>
            </div>
        </div>
    </div>
</div>
```



tabindex ترتیب انتقال فوکوس (تمرکز) روی عناصر را هنگام استفاده از کلید Tab مشخص می‌کند. عبارت tabindex=-1 به معنای آن است که مُدال قابلیت دریافت فوکوس توسط کلید Tab را ندارد.
btn-close استایل دکمه بستن در پنجره مُدال را به‌صورت یک علامت ضربدر مشخص می‌کند.
data-bs-dismiss="modal": رفتار دکمه close را مشخص می‌کند؛ به‌صورتی که وقتی کاربر روی دکمه کلیک نماید، مُدال بسته شود.
دکمه close بالای پنجره مدال در حالت پیش‌فرض در سمت راست پنجره ظاهر می‌شود. به همین دلیل از کلاس‌های d-flex، justify-content-end و ms-0 برای تنظیم محل دکمه استفاده شده است.
بعد از کلیک روی دکمه «مشاهده سبد خرید»، باید صفحه shopping-cart.html باز شود. برای این کار از ویژگی onclick دکمه استفاده کنید و مقدار آن را برابر یک دستور جاوااسکریپت طبق شکل زیر قرار دهید:

```
<button type="button" class="btn btn-primary"
      onclick="window.location.href='shopping-cart.html' ">مشاهده سبد خرید</button>
```

کروسل (Carousel)

کروسل یک ابزار مبتنی بر جاوااسکریپت است که برای ایجاد یک مجموعه اسلاید یا گالری تصاویر افقی استفاده می‌شود. کروسل امکان نمایش چند محتوا یا تصویر را در یک فضای محدود فراهم می‌کند؛ به‌طوری‌که کاربران بتوانند با استفاده از دکمه‌های کنترلی یا با کشیدن (swipe)، محتواهای بعدی را مشاهده کنند.
در یک سایت فروشگاهی، کروسل‌ها یا اسلایدرها امکان نمایش مجموعه‌ای از محصولات یا محتوای تبلیغاتی را به‌صورت چرخشی فراهم می‌کنند. استفاده از کروسل تجربهٔ کاربری بهتری را برای کاربران ارائه می‌کند.
برای ایجاد یک کروسل بوت‌استرپ از کلاس carousel و کلاس‌های زیرمجموعه آن استفاده می‌شود. جدول۱۳ اسامی کلاس‌ها و کاربرد آنها را توصیف می‌کند:

جدول ۱۳ــ کلاس‌های کروسل

نام کلاس	کاربرد
Carousel	برای نگهدارنده اصلی کروسل استفاده می‌شود. نگهدارندهٔ اصلی، تمام اجزای کروسل را دربرمی‌گیرد.
Slide	انیمیشن اسلاید را در هنگام تغییر اسلایدها فعال می‌کند.
carousel-inner	برای نگهدارندهٔ عناصر اسلاید به‌کار می‌رود.
carousel-item	برای هر اسلاید در کروسل استفاده می‌شود. عنصری که دارای کلاس carousel-item است، محتوای اسلاید (تصاویر، متن و...) را در برمی‌گیرد.
active	مشخص می‌کند کدام اسلاید در حال حاضر نمایش داده می‌شود.
carousel-control-prev	برای دکمهٔ انتقال به اسلاید قبلی استفاده می‌شود.
carousel-control-next	برای دکمهٔ انتقال به اسلاید بعدی استفاده می‌شود.
carousel-indicators	برای ایجاد نشانگرهای روی اسلاید استفاده می‌شود؛ دایره‌های کوچکی که نشان می‌دهند چند اسلاید وجود دارد و در حال حاضر کدام اسلاید نمایش داده می‌شود.
carousel-caption	برای افزودن متن یا توضیحات به اسلایدهای کروسل استفاده می‌شود.

کار کارگاهی

طراحی کروسل بالای صفحه فروشگاه

در این کارگاه چند تصویر بنر تبلیغاتی را درون کروسل به نمایش خواهید گذاشت.

تصاویر بنرهای تبلیغاتی را با اسامی slide1، slide2 و slide3 درون پوشه تصاویر پروژه ذخیره نمایید.

کدهای شکل زیر را در فایل index.html در ابتدای برچسب <main> وارد کنید:

```
<div id="banners" class="carousel slide" data-bs-ride="carousel">
    <div class="carousel-inner">
        <div class="carousel-item active">
            <img src="./images/slide1.jpg" class="d-block w-100" alt="...">
        </div>

        <div class="carousel-item">
            <img src="./images/slide2.jpg" class="d-block w-100" alt="...">
        </div>

        <div class="carousel-item">
            <img src="./images/slide3.jpg" class="d-block w-100" alt="...">
        </div>

        <div class="carousel-item">
            <img src="./images/slide4.jpg" class="d-block w-100" alt="...">
        </div>
    </div>
    <!-- محل درج دکمه های قبلی و بعدی -->
</div>
```

فایل را ذخیره کنید و نتیجه را در مرورگر بررسی کنید.
در حال حاضر دکمه‌های کنترلی به کروسل اضافه نشده است، بنابراین برای مشاهده اسلاید بعدی ۵ثانیه صبر کنید.
ویژگی data-bs-ride="carousel" برای فعال‌سازی حرکت خودکار کروسل استفاده می‌شود. این ویژگی به عنصر کروسل اعلام می‌کند که اسلایدها را پس از یک مدت زمان مشخص تغییر دهد.
مدت زمان نمایش هر اسلاید در اسلایدر با ویژگی data-bs-interval قابل تنظیم است. برای این منظور، data-bs-interval="2000" را به تگ مربوطه اضافه کنید.

```
<div id="banners" class="carousel slide" data-bs-ride="carousel" data-bs-interval="2000">
```

کد دکمه‌های «بعدی» و «قبلی» را در محل مشخص‌شده در کد کروسل اضافه کنید:

```
<button class="carousel-control-prev" type="button" data-bs-target="#banners" data-bs-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Previous</span>
</button>
<button class="carousel-control-next" type="button" data-bs-target="#banners" data-bs-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Next</span>
</button>
```

data-bs-target="#banners": مشخص می‌کند این دکمه کدام Carousel را کنترل می‌کند.
data-bs-slide="prev": انتقال به اسلاید قبلی به کمک این ویژگی انجام می‌شود.
کلاس carousel-control-prev-icon: این کلاس، آیکن فلش چپ (←) را برای دکمهٔ «اسلاید قبلی» نمایش می‌دهد.
کلاس visually-hidden برای مخفی کردن عبارات Previous و Next از دید کاربران عادی استفاده شده است، اما ابزارهای صفحه‌خوان محتوای عنصر را برای کاربران نابینا می‌خوانند. در واقع عنصر با ویژگی visually-hidden برای کاربران نابینا نوشته می‌شود.
ویژگی aria-hidden="true" در دکمه‌های کنترلی به صفحه‌خوان‌ها اعلام می‌کند که این عنصر باید نادیده گرفته شود و برای کاربران نابینا در دسترس نیست.

آکاردئون

آکاردئون یکی از اجزای کاربردی بوت‌استرت است که به توسعه‌دهنده امکان می‌دهد محتوا را در بخش‌های قابل باز و بسته شدن نمایش دهد. آکاردئون به‌صورت پیش‌فرض بسته است. با کلیک روی عنوان هر بخش، محتوای مربوط به آن ظاهر می‌شود و با کلیک بعدی مخفی می‌گردد. این ویژگی برای سازمان‌دهی محتوای طولانی و نمایش اطلاعات به صورت فشرده بسیار مفید است.
آکاردئون می‌تواند در طراحی بخش‌های مختلفی کاربرد داشته باشد؛ مانند منوی ناوبری و کشویی، نمایش اطلاعات تکمیلی یک بخش، ایجاد بخش سوالات متداول و غیره.

در طراحی یک آکاردئون کلاس‌های مختلفی استفاده می‌شود. جدول ۱۴ اسامی کلاس‌ها و کاربرد هر یک را نشان می‌دهد.

جدول ۱۴ــ کلاس‌های آکاردئون

نام کلاس	کاربرد
Accordion	این کلاس برای ایجاد نگهدارنده اصلی آکاردئون استفاده می‌شود و تمام عناصر آکاردئون را در بر می‌گیرد.
Accordion-item	یک آکاردئون دارای چند بخش (Item) است که هر بخش مجزا از این کلاس استفاده می‌کند. هر بخش شامل یک header و یک محتوای قابل جمع شدن است.
Accordion-header	این کلاس برای ایجاد header در هر بخش استفاده می‌شود و معمولاً شامل یک دکمه یا پیوند است که با کلیک بر روی آن، محتوای مربوط به آن بخش آشکار یا مخفی می‌شود.
Accordion-button	استایل دکمه‌ای که روی آن کلیک می‌شود را تعیین می‌کند.
Accordion-collapse	این کلاس برای تعیین استایل عنصری که محتوای قابل جمع شدن دارد به‌کار می‌رود.
Accordion-body	این کلاس برای ایجاد محتوای قابل نمایش در هر بخش (Item) استفاده می‌شود.
Collapse	برای باز و بسته شدن محتوا به‌کار می‌رود.
show	این کلاس باعث می‌شود که محتوای یک عنصر به‌صورت باز نمایش داده شود.

عنصر با کلاس accordion-item باید به تعداد بخش‌های آکاردئون استفاده شود.

کار کارگاهی

طراحی بخش پاسخ به پرسش‌های متداول

در کارگاه طراحی فوتر، گزینه پرسش‌های متداول درون ستون اول قرار گرفت. در این کارگاه صفحه مربوط به این گزینه را طراحی خواهید کرد.

۱ـ فایل index.html را باز کنید و از منوی File گزینه Save As را انتخاب کنید. نام فایل جدید را faq.html قرار دهید. کلمه faq مخفف Frequently Asked Questions است.

۲ـ محتوای عنصر main را به‌طور کامل پاک کنید و کد آکاردئون را به طبق کدهای زیر درون آن بنویسید:

```
<div class="accordion w-75 mt-5 mx-auto mb-5" id="faq">
    <div class="accordion-item">
        <h2 class="accordion-header ">
            <button class="accordion-button" type="button" data-bs-toggle="collapse"
                    data-bs-target="#q1" aria-expanded="true" aria-controls="q1">
                سفارش‌ها چند روزه به دستم می‌رسه؟
            </button>
        </h2>
        <div id="q1" class="accordion-collapse collapse show" data-bs-parent="#faq">
            <div class="accordion-body">
                سفارش‌های شهر تهران: بین ۲ تا ۳ روز کاری
                <br/>سفارش‌های سایر شهرها: ۴ تا ۵ روز کاری بعد از ثبت سفارش
            </div>
        </div>
    </div>
    <div class="accordion-item">
        <h2 class="accordion-header">
            <button class="accordion-button" type="button" data-bs-toggle="collapse"
                    data-bs-target="#q2" aria-expanded="true" aria-controls="q2">
                چاپ کتاب‌ها برای چه مدتی است؟
            </button>
        </h2>
        <div id="q2" class="accordion-collapse collapse" data-bs-parent="#faq">
            <div class="accordion-body">
                این دوره معمولا داخل صفحه خود محصول.
                داخل سایت مرجع تو نوشته شده است. ما سعی می‌دهیم آخرین چاپ کتاب را ارسال کنیم
            </div>
        </div>
    </div>
</div>
```

این آکاردئون دارای دو بخش است. با توجه به اینکه کلاس show برای بخش اول استفاده شده است در نتیجه محتوای بخش اول در حال نمایش خواهد بود.

۳ـ در حالت پیش‌فرض فلش کنار هر دکمه به‌صورت رو به بالا و در کنار متن دکمه نمایش داده می‌شود. برای اصلاح این وضعیت از استایل زیر در فایل styles.css استفاده کنید:

```
.accordion-button::after {
  position:absolute;
  left:20px;
}
```

انتخاب‌گر accordion-button::after عنصر بعد از عنوان دکمه (آیکن فلش) را انتخاب می‌کند.

فعالیت

بخش سوم را با محتوای زیر به آکاردئون اضافه کنید:

درصورتی‌که کالا در فروش ویژه باشد، شامل محدودیت در تعداد خرید می‌شود. در غیر این صورت هیچ‌گونه محدودیتی وجود ندارد.

انتظار می‌رود که در پایان خروجی زیر را به‌دست آورید:



فعالیت

پرسش‌های متداول در بخش فوتر سایت را به فایل faq.html لینک کنید.

ارزشیابی شایستگی توسعه صفحات وب تعاملی با بوت‌استرپ

استاندارد عملکرد کار: هنرجو باید بتواند با استفاده از Bootstrap، قالب واکنش‌گرا و استاندارد طراحی کند، کامپوننت‌های آماده را سفارشی‌سازی نماید، ترکیب CSS و JS را به درستی به کار گیرد، اصول طراحی واکنش‌گرا را رعایت کند و در پایان یک صفحه وب تعاملی و کاربرپسند ارائه دهد.

ردیف	مراحل کار	شرایط انجام کار (ابزار،مواد، تجهیزات، زمان، مکان و ...)	ابزارهای ارزشیابی	استاندارد (شاخص‌ها/داوری/نمره‌دهی)	نتایج ممکن	نمره کسب شده
۱	آشنایی و شروع کار با Bootstrap	رایانه، مرورگر، فایل‌های CSS/JS، ۱۵ دقیقه	مشاهده عملی، تئوری	ـ توضیح صحیح فریم ورک ـ اضافه‌کردن صحیح Bootstrap از طریق CDN ـ نصب محلی Bootstrap در پروژه ـ لینک‌کردن فایل CSS و JS بدون خطا ـ انجام کنجکاوی‌ها ـ انجام مراحل با روش‌های دیگر	۳	
				ـ توضیح صحیح فریم‌ورک ـ اضافه‌کردن صحیح Bootstrap از طریق CDN ـ نصب محلی Bootstrap در پروژه ـ لینک‌کردن فایل CSS و JS بدون خطا	۲	
				ـ توضیح صحیح فریم‌ورک ـ انجام مراحل به‌صورت ناقص	۱	
۲	آشنایی با کلاس‌های آماده Bootstrap	رایانه ـ مرورگر،vs code، ۲۵ دقیقه	مشاهده عملی	انجام کنجکاوی‌ها ـ استفاده صحیح از کلاس‌های رنگ ـ اعمال spacing استاندارد ـ استفاده از Flex برای تراز کردن عناصر ـ سفارشی‌سازی یک کلاس رنگ ـ انجام کارها با روش‌های متفاوت	۳	
				ـ استفاده صحیح از کلاس‌های رنگ ـ اعمال spacing استاندارد ـ استفاده از Flex برای تراز کردن عناصر ـ سفارشی‌سازی یک کلاس رنگ	۲	
				انجام مراحل به‌صورت ناقص	۱	
۳	شبکه‌بندی و طراحی ساختار صفحه	رایانه ـ مرورگر،vs code، ۲۵ دقیقه	مشاهده عملی	ـ ایجاد ساختار چندستونه با rows و cols ـ استفاده صحیح از row و cols ـ طراحی فوتر با Grid ـ انجام کنجکاوی‌ها ـ انجام مراحل با چند روش	۳	
				ـ ایجاد ساختار چندستونه با rows و cols ـ استفاده صحیح از row و cols ـ طراحی فوتر با Grid	۲	
				انجام مراحل به صورت ناقص	۱	
۴	ـ ایجاد ساختار چندستونه با rows و cols ـ استفاده صحیح از row و cols ـ طراحی فوتر با Grid	رایانه ـ مرورگر،vs code، ۲۵ دقیقه	مشاهده عملی ـ آزمون تئوری	ـ تعریف Responsive Design ـ انجام کنجکاوی‌ها ـ استفاده ازکلاس‌های واکنش‌گرا ـ تغییر تعداد ستون‌ها در اندازه‌های مختلف ـ ایجاد تصویر واکنش‌گرا با img ـ fluid ـ انجام مراحل با چند روش	۳	
				ـ ایجاد منوی تعاملی ـ افزودن لینک و زیرمنوها	۲	
				انجام مراحل به‌صورت ناقص	۱	
۵	طراحی بخش‌های اصلی فروشگاه	رایانه ـ مرورگر،vs code، ۲۵ دقیقه	مشاهده عملی	طراحی نوار ابزار شامل: لوگو ـ جست‌وجو ـ ورود/ثبت‌نام ـ سبد خرید ـ طراحی کارت محصولات ـ طراحی صفحه اختصاصی محصول ـ انجام کنجکاوی‌ها ـ صفحه فروشگاه دارای: ساختار منظم ـ چیدمان استاندارد محصولات ـ سازگاری کامل با موبایل ـ رعایت فاصله‌گذاری و تراز مناسب انجام مراحل با روش‌های دیگر	۳	
				طراحی نوار ابزار شامل: لوگو ـ جست‌وجو ـ ورود/ثبت‌نام ـ سبد خرید ـ طراحی کارت محصولات ـ طراحی صفحه اختصاصی محصول	۲	
				طراحی نوار ابزار شامل: لوگو ـ جست‌وجو	۱	

۶	اجزای رابط کاربری (UI Components) و تعاملات	رایانه ـ مرورگر، vs code، ۳۰ دقیقه	مشاهده عملی	ـ طراحی Navbar واکنش‌گرا ـ ساخت فرم ثبت‌نام معتبر ـ ایجاد Modal فعال ـ پیاده‌سازی Carousel ـ عملکرد صحیح تمام کامپوننت‌ها ـ بدون خطای JS ـ هماهنگی ظاهری بین اجزا ـ تعامل کاربر بدون اختلال	۳	
				ـ طراحی Navbar واکنش‌گرا ـ ساخت فرم ثبت‌نام معتبر ـ پیاده‌سازی Carousel ـ عملکرد صحیح تمام کامپوننت‌ها	۲	
				ـ طراحی Navbar واکنش‌گرا ـ ساخت فرم ثبت‌نام	۱	
شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش:				احراز همه شایستگی‌های غیرفنی	۲	
				عدم احراز هر یک از شایستگی‌های غیرفنی	۱	
ارزشیابی کار (شایستگی انجام کار)						بلی ☐
						خیر ☐

توضیحات:

۱ـ معیار شایستگی انجام کار:

ـ کسب حداقل نمره ۲ از مراحل ۱، ۳ و ۴ (مراحل بحرانی)

ـ کسب حداقل نمره ۲ از بخش شایستگی‌های غیرفنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش

ـ کسب حداقل میانگین ۲ از مراحل کار

۲ـ تعریف سطوح شایستگی: صرف‌نظر از اینکه یک تکلیف کاری در چه سطحی از صلاحیت حرفه‌ای انجام می‌شود، در هر محیط کاری ممکن است انجام هر کار با کیفیت مشخصی مورد انتظار باشد. سطح شایستگی انجام کار، معیار اساسی ارزشیابی است.

مهارت (سطح سه): ماهر و قادر به آموزش و هدایت دیگران، توانایی برنامه‌ریزی و تحلیل، پاسخ‌گویی در برابر کارهای خود، سروکار داشتن با سطح وسیعی از کارها و فعالیت‌ها

تسلط (سطح چهار): خبرگی در انجام کار و آموزش دیگران، ایجاد نوآوری، سازگاری، عیب‌یابی، هدایت و راهنمایی دیگران.

واحد یادگیری ۲
طراحی صفحات وب تعاملی با جاوا اسکریپت

آیا تابه‌حال اندیشیده‌اید

- واکنش به کلیک، حرکت ماوس یا فشردن کلیدهای صفحه‌کلید توسط چه زبانی برنامه‌نویسی می‌شود؟
- بازی‌های ساده، اسلایدر تصاویر و منوهای بازشونده توسط چه زبانی ساخته می‌شوند؟
- چگونه می‌توان با کلیک روی یک دکمه، عملی را اجرا نمود؟
- آیا امکان دسترسی به عناصر صفحه و تغییر محتوا و ویژگی‌های آنها وجود دارد؟
- آیا می‌توان بدون بار گذاری دوباره صفحه، محتوای آن را تغییر داد؟
- چگونه می‌توان اعتبار اطلاعات وارد شده در فرم را پیش از ارسال بررسی نمود؟
- محاسبات عددی و پردازش رشته‌های ورودی کاربر چگونه انجام می‌شود؟

در پایان از هنرجو انتظار می‌رود

- با استفاده از متغیرها، عملگرها و ساختارهای کنترلی، منطق موردنیاز برنامه را پیاده‌سازی کند.
- نحوه تعریف وظایف تکراری را در قالب توابع بداند و از توابع داخلی جاوااسکریپت برای انجام سریع‌تر کارها استفاده نماید.
- با استفاده از انتخابگرهای DOM، محتوا و ویژگی‌های عناصر HTML را در پاسخ به رویدادهای کاربر تغییر دهد.
- ورودی‌های کاربر را در فرم‌ها اعتبارسنجی کرده و در صورت وجود خطا، پیام مناسب نمایش دهد.

استاندارد عملکرد

بتواند با استفاده از Java Script به‌صورت بهینه و امن، رفتارهای پویا و تعاملی را به صفحات وب اضافه کند و خروجی نهایی را در مرورگرهای مختلف اجرا کند.

تعریف جاوااسکریپت

در پودمان دوم، بخش‌هایی از صفحه اول یک فروشگاه اینترنتی با استفاده از HTML و CSS طراحی شد. این بخش‌ها ماهیتی ایستا دارند و فاقد قابلیت تعامل با کاربر هستند؛ به این معنا که نمی‌توانند در ازای ورودی‌های کاربر (مانند کلیک یا تایپ) واکنش مناسبی (مانند نمایش پیام یا تغییر محتوا) ارائه دهند. در واقع HTML ساختار و محتوای صفحه وب را تعریف می‌کند و CSS سبک نمایش عناصر صفحه را تعیین می‌کند.



شکل ۱

برای تبدیل ساختار و ظاهر ایستای صفحه به یک تجربه کاربری پویا، تعریف رفتار و منطق صفحه ضروری است. زبان برنامه‌نویسی جاوااسکریپت با فراهم کردن بستر لازم برای مدیریت رویدادها و پردازش داده‌ها، امکان تعامل مستقیم با کاربر و تغییر پویای محتوای صفحه را بدون نیاز به بارگذاری مجدد، میسر می‌سازد. به بیان ساده، جاوااسکریپت مسئولیت پاسخ‌گویی به اقدامات کاربر و پویا کردن رفتار عناصر صفحه را بر عهده دارد.

جاوااسکریپت یک زبان برنامه‌نویسی سمت کاربر (Front-End) است، به این معنی که برای اجرای اسکریپت‌های آن نیازی به وب‌سرور نیست و کدهای آن توسط مرورگر اجرا می‌شود. با این‌حال جاوااسکریپت قابلیت اجرا روی سرور را به کمک Node.js دارد. جاوااسکریپت یک زبان مفسری است، مفسر جاوااسکریپت یک نرم‌افزار است که کدهای نوشته شده به زبان جاوااسکریپت را تجزیه کرده، سپس آنها را به‌صورت خط به خط خوانده و اجرا می‌کند. مفسرهای جاوااسکریپت به‌عنوان بخشی از مرورگرهای وب عمل می‌کنند.

زبان جاوااسکریپت به‌عنوان زبان اصلی سمت کاربر، با امکان دستکاری مستقیم مدل اشیای سند (DOM)، امکان تغییر ساختار و محتوای عناصر صفحه را در پاسخ به اقدامات کاربر فراهم می‌کند. این زبان با پشتیبانی از الگوهای برنامه‌نویسی شیءگرا و رویدادمحور، به توسعه‌دهندگان امکان می‌دهد که منطق‌های پیچیده، اعتبارسنجی فرم‌ها و ارتباطات غیرهم‌زمان با سرور را پیاده‌سازی کنند. همچنین، گسترش دامنه کاربرد این زبان از طریق محیط‌های اجرا مانند Node.js، آن را به ابزاری چندمنظوره برای توسعه وب‌اپلیکیشن‌های تمام‌عیار تبدیل کرده است.

جاوااسکریپت توانایی اضافه کردن قابلیت‌های مختلفی را به صفحه وب دارد که از جمله آنها می‌توان به موارد زیر اشاره نمود:

- **سبد خرید پویا:** امکان افزودن، حذف یا تغییر تعداد محصولات در سبد خرید بدون بارگذاری مجدد صفحه و محاسبه خودکار مجموع قیمت.

- **اعتبارسنجی فرم‌ها:** بررسی صحت اطلاعات وارد شده در فرم ثبت‌نام یا خرید (مانند شماره تماس، کد پستی و ایمیل) قبل از ارسال نهایی.

■ **نمایش محصولات به‌صورت پویا:** فیلتر کردن محصولات بر اساس دسته‌بندی، قیمت یا برند و مرتب‌سازی لیست محصولات در لحظه.

■ **اسلایدر و گالری تصاویر:** ایجاد حرکت خودکار یا دستی تصاویر محصولات برای نمایش زوایای مختلف یک محصول.

■ **منوی واکنش‌گرا (Responsive):** باز و بسته شدن منوی موبایل با کلیک کاربر و نمایش مناسب در دستگاه‌های مختلف.

■ **ارسال پیام (Notification):** نمایش پیام‌های موفقیت (مانند «محصول به سبد اضافه شد») یا خطا به‌صورت پاپ‌آپ (pop up)

■ **تاریخ و ساعت:** نمایش تاریخ شمسی یا ساعت جاری برای اطلاع‌رسانی به کاربر.

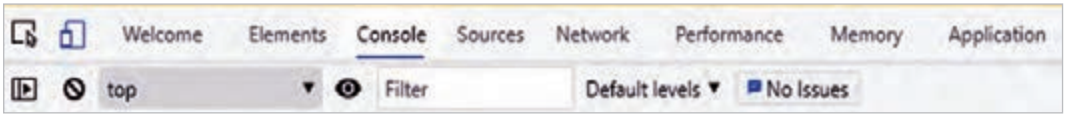
■ **ذخیره‌سازی موقت:** نگهداری اطلاعات ورود یا تنظیمات کاربر در مرورگر (کوکی‌ها) برای تجربه خرید آسان‌تر در بازدیدهای بعدی.

■ **محیط توسعه (Development Environment)**

برای شروع برنامه‌نویسی با جاوااسکریپت، نیاز به یک محیط توسعه مناسب است که امکان نوشتن کد، تست و اشکال‌زدایی آن را فراهم کند. مراحل آماده‌سازی محیط توسعه جاوااسکریپت به شرح زیر است:

■ **انتخاب و نصب ویرایشگر:** برای کدنویسی با جاوااسکریپت می‌توان از ویرایشگرهای مختلفی مانند VSCode، Sublime Text و Atom و Notepad++ استفاده نمود. این ویرایشگرها دارای امکاناتی نظیر رنگ‌آمیزی کد، تکمیل خودکار کد و ابزارهای خطایابی (Debug) هستند که به بهبود کارایی و تجربه برنامه‌نویسی کمک می‌کنند.

■ **انتخاب و نصب مرورگر وب:** تمامی مرورگرهای مدرن امروزی همچون Google Chrome، Mozilla Firefox، Microsoft Edge و Safari امکان اجرای کدهای جاوااسکریپت را به‌صورت پیش‌فرض دارند. به علاوه، این مرورگرها دارای ابزارهای توسعه‌دهنده (Developer tools) هستند که امکانات بیشتری را برای برنامه‌نویس وب فراهم می‌کنند. این ابزارها که توسط کلید F12 فعال می‌شوند، شامل کنسول، بخش Element (حاوی عناصر HTML و استایل عناصر)، Sources (منابع پروژه) و... هستند (شکل۲).



شکل ۲

افزودن جاوا اسکریپت به یک صفحه وب

برای درج کدهای جاوااسکریپت در صفحه وب، از تگ <script></script> استفاده می‌شود. این تگ می‌تواند در بخش <head>، <body> مورد استفاده قرار گیرد. محل نوشتن کد جاوااسکریپت وابسته به نوع کد و نیازهای خاص پروژه است. با این‌حال توصیه می‌شود که تا حد امکان این کدها **قبل از** تگ </body> نوشته شود. با انجام این کار ابتدا محتوای صفحه به‌طور کامل بارگذاری شده و در معرض دید کاربر قرار می‌گیرد، سپس اسکریپت‌ها اجرا می‌شوند، در نتیجه زمان بارگذاری صفحه کوتاه‌تر می‌شود. کدهای جاوااسکریپت می‌توانند در یک فایل خارجی با پسوند js نوشته شوند. در این صورت فایل js باید به فایل html الحاق شود. در بخش بعد توضیحات بیشتری درباره روش‌های فوق ارائه می‌شود.



شکل ۳

نوشتن کد جاوااسکریپت

یک کد جاوااسکریپت حاوی مجموعه‌ای از دستورالعمل‌هاست. هر دستورالعمل یک عملیات مجزا را تعریف می‌کند و با نقطه‌ویرگول (;) از دستور بعدی جدا می‌شود. یکی از ساده‌ترین و پرکاربردترین دستورات جاوااسکریپت، متد()console.log است که برای نمایش خروجی کد در کنسول مرورگر مورد استفاده قرار می‌گیرد. نحوه صحیح به‌کارگیری این دستور به شکل زیر است:

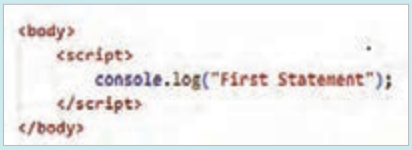
console.log (رشته ثابت یا متغیر);

نکته

توصیه می‌شود که در انتهای دستورات جاوااسکریپت از علامت سمی‌کالن (;) برای اعلام خاتمه دستور استفاده شود، اما استفاده از آن اجباری نیست.

کار کارگاهی

کدنویسی با زبان جاوااسکریپت





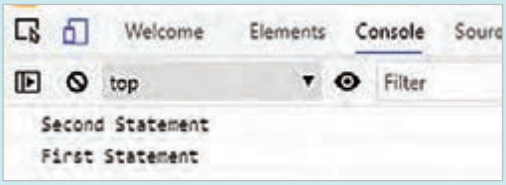
۱ برای شروع کار پوشه جدیدی با عنوان JavaScript ایجاد کنید و فایل کارگاه‌ها را درون آن ایجاد کنید.

۲ در پوشه JavaScript، یک فایل با نام workshop1.html ایجاد کنید و اسکریپتی مشابه تصویر روبه‌رو درون بخش <body> بنویسید.

۳ فایل workshop۱.html را اجرا کنید و برای مشاهده خروجی کد، کلید F12 را در مرورگر بفشارید تا ابزارهای توسعه‌دهنده (Developer Tools) ظاهر شوند. سپس برای مشاهده نتیجه اسکریپت روی دکمه Console کلیک کنید.

۴ اسکریپت مشابهی را به بخش <head> اضافه کنید و فایل را ذخیره کنید. انتظار می‌رود که نتیجه کار در کنسول مشابه تصویر روبه‌رو باشد:

```
<head>
    <title> test javascript </title>
    <script>
        console.log("Second Statement");
    </script>
</head>
```



مرورگر وب ابتدا اسکریپت بخش <head> و سپس اسکریپت بخش <body> را اجرا می‌کند. به همین دلیل عبارت Second Statement جلوتر نمایش داده شده است.

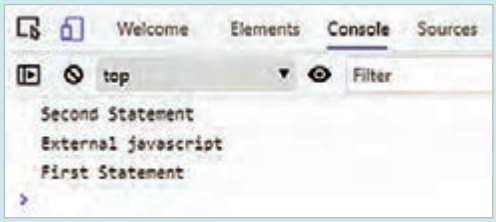
۵ یک فایل با نام script.js درون پوشه JavaScript ایجاد کنید و دستور زیر را درون آن بنویسید:

```
console.log("External javascript");
```

۶ سپس محتوای فایل را ذخیره کنید.

۷ در بخش <head> فایل workshop1.html، فایل script.js را به صفحه وب الحاق کنید. دستور الحاق را دقیقاً قبل از تگ </head> قرار دهید:

```
<head>
    <title> test javascript </title>
    <script>
        console.log("Second Statement");
    </script>
    <script src="./script.js"></script>
</head>
<body>
    <script>
        console.log("First Statement");
    </script>
</body>
```



۸ خروجی کد را در کنسول مرورگر مشاهده کنید و نتایج حاصل را به ترتیب نمایش عبارات در جای خالی بنویسید.

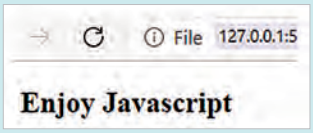
.....

۹ کد الحاق فایل جاوااسکریپت را به قبل از </body> منتقل کنید و نتیجه این تغییر را در جای خالی بنویسید.

.....

۱۰ برای نمایش یک عبارت در مرورگر (نه در بخش console) می‌توان از دستور document.Write استفاده کرد. این دستور عبارت داخل پرانتز را جایگزین محتوای فعلی صفحه می‌کند. دستور روبه‌رو را به فایل workshop1.html اضافه کنید:

```
<body>
    <script>
        console.log("First Statement");
        document.write("<h2>Enjoy Javascript</h2>");
    </script>
</body>
```



فایل HTML را ذخیره کنید و نتیجه را در صفحه اصلی مرورگر به نمایش بگذارید. عبارت Enjoy JavaScript در اندازه h2 درون صفحه مرورگر ظاهر می‌شود.

در صورتی که صفحه وب دارای محتوا باشد، دستور document.write() باعث ناپدید شدن محتوای جاری و جایگزین شدن آن با محتوای درون پرانتز می‌گردد. بر این اساس به‌کارگیری این دستور برای انجام برخی تمرین‌های کوچک اشکالی ندارد، اما برای استفاده از آن در پروژه باید با احتیاط رفتار شود.

کنجکاوی

درباره مزایای اسکریپت نویسی در فایل خارجی تحقیق کنید و نتایج را در کلاس ارائه کنید.

■ **توضیحات (comments)**: جاوااسکریپت امکان ایجاد توضیحات تک خطی و چند خطی را توسط علائم // و /* */ فراهم می‌کند. نوشتن توضیحات برای روشن کردن هدف، عملکرد و منطق کد ضروری است. از علائم کامنت برای لغو اجرای موقت برخی کدها به منظور اشکال‌زدایی برنامه نیز استفاده می‌شود.

```
console.log("Javascript");  // This statement outputs the string "Javascript" to the console
/* The code below
outputs the string "Javascript" to the console. */
console.log("Javascript");
```

شکل ۴

■ **متغیرها (Variables)**: جاوااسکریپت دارای دو کلمه کلیدی let و var برای تعریف متغیرها است. کلمه کلیدی var زمانی استفاده می‌شود که پشتیبانی از مرورگرهای قدیمی ضرورت داشته باشد. این کلمه تا سال ۲۰۱۵ برای تعریف متغیر استفاده می‌شد. از سال ۲۰۱۵ کلمه کلیدی let برای ایجاد متغیر معرفی شد.

مثال

مثال زیر نحوه تعریف متغیرهای x و y را با مقادیر مختلف نمایش می‌دهد:

```
let x = 10;
var y = 15.75;
```

تعریف متغیرها می‌تواند به‌صورت خودکار نیز انجام شود. تعریف خودکار متغیر زمانی انجام می‌شود که یک متغیر مقداردهی شود.

مثال

در مثال زیر دو متغیر x و z به‌صورت خودکار تعریف شده‌اند:

```
x = 10;
z = x + 10;
```

بیشتر بدانید

ثابت‌ها (constants): ثابت‌ها (const) دارای مقدار ثابتی در طول اجرای برنامه هستند. ثابت‌ها یک‌بار تعریف و مقداردهی می‌شوند و بعد از آن امکان مقداردهی مجدد ندارند.

مثال

در مثال زیر، ثابت PI با مقدار 3.141592 تعریف شده است. تغییر مقدار PI در خط دوم و سوم باعث رخداد خطا می‌گردد.

```
const PI = 3.141592;
PI = 3.14;
PI = PI + 10;
```

اصول نام‌گذاری متغیرها

- نام متغیر می‌تواند حاوی حروف الفبای انگلیسی، اعداد، علامت $ و زیرخط (_) باشد، اما در ابتدای نام متغیر نباید از اعداد استفاده شود.
- جاوااسکریپت نسبت به کوچکی و بزرگی حروف حساس است، بنابراین اسامی price و Price و PRICE سه متغیر متفاوت هستند.
- کلمات رزرو شدهٔ جاوااسکریپت (که معنای خاصی دارند)، نباید به‌عنوان نام متغیر استفاده شوند، مانند let و var، if، for و...

جاوااسکریپت امکان تعریف متغیرها را در یک خط نیز فراهم می‌کند، اما در پایان هر تعریف متغیر، باید از علامت کاما برای جداسازی استفاده شود.

```
let firstName = "Sara", lastName = "Jones", studentId = "۱۲۳۴۵۶۷";
```

مقداردهی به متغیرها توسط عملگر انتساب (=) انجام می‌شود. در صورتی‌که مقداردهی متغیر انجام نشود، مقدار آن تعریف نشده یا (Undefined) خواهد بود.

فعالیت

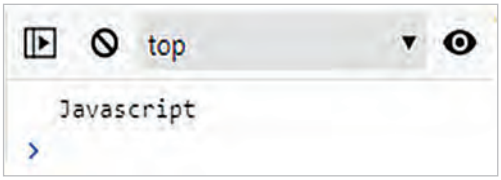
مشخص کنید که تعریف متغیرهای زیر به‌درستی صورت گرفته است یا خیر.

```
myVar = 2 ;          ……..……..
2x = 10;             ……..……..
user-name ="Safa";   ……..……..
var = 20;            ……..……..
```

نمایش مقدار متغیرها و رشته‌ها

برای نمایش مقدار یک متغیر در کنسول مرورگر از دستور console به شکل ۵ استفاده می‌شود:

```
<script>
    x = "Javascript";
    console.log(x);
</script>
```



شکل ۵

با توجه به حساسیت جاوااسکریپت نسبت به کوچکی و بزرگی حروف، نوشتن دستور console.log(X) باعث نمایش پیغام خطای زیر می‌شود:



شکل ۶

برای نمایش یک رشته متنی در کنسول یا صفحه وب، از علائم نقل قول در اطراف متن استفاده می‌شود، اما متغیرها نیازی به این علائم ندارند. در مثال زیر یک عبارت متنی به‌همراه مقدار متغیر x درون کنسول نمایش داده شده است.

```
<script>
    x = 100;
    console.log("The value of x is: " + x );
</script>
```



شکل ۷

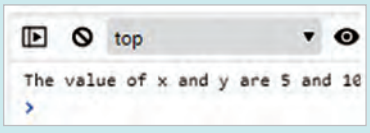
در اینجا عملگر + برای چسباندن دو عبارت به یکدیگر استفاده شده است. بسته به نوع داده‌ها، این عملگر می‌تواند به عنوان عملگر محاسباتی جمع یا الحاق رشته‌ها عمل کند.

کار کارگاهی

تعریف متغیر و نمایش آن در جاوااسکریپت

در ادامه اسکریپت فایل workshop1.html متغیرهای x و y را با مقادیر ۵ و ۱۰ تعریف کنید و سپس مقدار آنها را در کنسول مرورگر نمایش دهید:

```
<script>
  let x=5 , y=10;
  console.log("The value of x and y are " + x +" and "+ y);
</script>
```



یک متغیر به‌صورت زیر ایجاد کنید و سپس مقدار آن را در کنسول نمایش دهید.

let test;

با توجه به اینکه متغیر test مقدار دهی نشده است، درصورتی که از دستور console.log(test) استفاده شود، پیغام ..... ظاهر می‌شود.

انواع داده

در یک زبان برنامه‌نویسی، متغیرها می‌توانند انواع داده‌های مختلفی را درون خود ذخیره کنند. برای ذخیره هر نوع داده میزان حافظه متفاوتی مورد نیاز است؛ انواع داده‌ها به زبان‌های برنامه‌نویسی امکان می‌دهند که از حافظه به‌صورت مؤثرتری استفاده کنند. همچنین برنامه‌نویسان می‌توانند داده‌ها را به شکلی منطقی‌تر و منسجم‌تر سازماندهی کنند. جدول ۱۵ انواع داده‌های قابل استفاده در جاوااسکریپت را نمایش می‌دهد.

جدول ۱۵ــ انواع داده در جاوااسکریپت

ردیف	نوع داده	توضیح	مثال
۱	عددی (Number)	شامل مقادیر عددی صحیح و اعشاری	let x = 20 , y = 10.75;
۲	رشته‌ای (String)	عباراتی که داخل علامت نقل قول تکی یا دو تایی قرار می‌گیرند، رشته تلقی می‌شوند.	let family = "Smith"; let color = "Orange";
۳	منطقی (Boolean)	شامل مقادیر true و false	let a = true , b = false; let c = x>y , d = "amin" < "Amin";
۴	آرایه‌ها (Array)	متغیرهایی از نوع آرایه می‌توانند انواع داده مختلفی را درون خود نگهداری کنند.	let numbers = [10,20,30,40,50]; let names = ["sara", "mahdi", "amir"]; let data = [100, 5.75, "javascript", true];
۵	شیء (object)	اشیای جاوااسکریپت به‌صورت جفت‌های ویژگی: مقدار درون آکولاد مشخص می‌شوند.	let obj = {username: "Alfred" , password: "admin" };
۶	شیء تاریخ (Date)	شیء تاریخ با استفاده از کلاس Date ایجاد می‌شود.	const date = new Date("2022-03-25");

کار کارگاهی

کار با انواع داده‌ها

۱ یک سند HTML با نام workshop2.html ایجاد کنید و در بخش <body> برچسب <script> را بنویسید. متغیرهای x، Family، d و names را مطابق مثال‌های جدول فوق درون تگ <script> تعریف کنید و سپس مقادیر آنها را مانند دستور زیر در کنسول نمایش دهید:

```
console.log("The value of x is: "+ x);
```

۲ عملگر **typeof** در جاوااسکریپت نوع داده یک متغیر را مشخص می‌کند. این عملگر می‌تواند همراه با پرانتز یا بدون آن استفاده شود. در انتهای اسکریپت فایل workshop2.html این عملگر را مطابق مثال زیر برای بررسی نوع داده متغیرهای x، Family، d و names به‌کار ببرید و نتایج را در جای خالی بنویسید.

```
console.log("x: "+ typeof(x));
```

x: number Family: ..... d: ..... names: .....

۳ رفتار مرورگر را در تفسیر کدهای زیر بررسی کنید و جای خالی را بر اساس نتیجه‌گیری خود تکمیل کنید.

```
let a = 10 + 20 + "Blue";
let b = "Blue" + 10 + 20;
```

در خط اول، عملوندهای ۱۰ و ۲۰ در ابتدای عبارت محاسباتی قرار گرفته‌اند، پس جاوااسکریپت اعداد ۱۰ و ۲۰ را نوع داده number تلقی می‌کند، در نتیجه مقدار ..... درون متغیر a ذخیره می‌شود. در خط دوم به علت قرار گرفتن رشته Blue در ابتدای عبارت محاسباتی، عملوندهای بعدی، از نوع داده ..... تلقی می‌شوند و بنابراین مقدار ..... درون متغیرb ذخیره می‌گردد.

۴ نوع داده متغیر می‌تواند در طول برنامه تغییر نماید. در اسکریپت زیر متغیر x ابتدا به‌صورت تعریف‌نشده (Undefined) ایجاد شده است. نوع داده متغیر x در خط ۲ با دریافت مقدار ۱۰ به نوع ..... و در خط ۳ با دریافت مقدار sara1980 به نوع ..... تغییر می‌کند.

```
1  let x;
2  x = 10;
3  x = "sara1980";
```

تبدیل نوع داده‌ها

جاوااسکریپت امکان تبدیل نوع داده متغیرها را به دو روش امکان‌پذیر می‌سازد: تبدیل ضمنی (Implicit) و صریح (Explicit). وقتی تبدیل نوع داده متغیر به‌صورت خودکار انجام شود، تبدیل ضمنی صورت گرفته است.

مثال

مثال زیر نمونه‌ای از تبدیل نوع ضمنی را نمایش می‌دهد:

```
let s = 1;
console.log("type of s is: " + typeof s);
let w = 10;
s = w + "100";
console.log(s);
console.log("type of s is: "+typeof(s));
```

```
top
type of s is: number
10100
type of s is: string
>
```

در خط اول، متغیر s از نوع عددی تعریف شده است. در خط چهارم نتیجه جمع متغیر w و رشته ۱۰۰، عبارت ۱۰۱۰۰ است که درون متغیر s قرار می‌گیرد. در نتیجهٔ نوع داده متغیر s به رشته تبدیل می‌شود. به‌کارگیری عملگر typeof() در خط ۲ و ۶ باعث نمایش نوع داده متغیر s می‌شود.

تبدیل صریح نوع داده توسط توابع داخلی جاوااسکریپت انجام می‌پذیرد. توابع مربوط به تبدیل داده شامل parseInt()، parseFloat()، Number()، String() و Boolean() است که نوع داده آرگومان ورودی را به نوع داده صحیح، اعشاری، عددی، رشته‌ای و منطقی تبدیل می‌کنند.

کار کارگاهی

کار با توابع تبدیل نوع

متغیرهای زیر را در انتهای اسکریپت فایل workshop2.html ایجاد کنید.

```
x = "15.25" , y = 123 , z = 1;
1 let num1 = parseInt(x);          …15...    …number...
2 let num2 = parseFloat(y)         ……...     ……...
3 let num3 = Number(x);            ……...     ……...
4 let num4 = Number(true);         ……...     ……...
5 str = String(y);                 ……...     ……...
6 let bool = Boolean(z);           ……...     ……...
```

سپس مقدار و نوع داده متغیرهای ایجاد شده در خطوط ۲ تا ۷ را توسط تابع typeof() بررسی کنید و نتیجه را مانند مثال در جای خالی بنویسید. برای بررسی مقادیر و نوع متغیرها مانند الگوی زیر عمل کنید:

```
console.log(num1 +": "+ typeof(num1));
```

نکته

علت عدم استفاده از کلمه کلیدی let در متغیرهای x، y، z و str این است که این متغیرها در کارگاه‌های قبلی ایجاد شده‌اند. بنابراین استفاده مجدد از کلمه let باعث ایجاد خطا می‌شود.

عملگرها (Operators)

در زبان‌های برنامه‌نویسی عملگرها برای انجام عملیات روی متغیرها و مقادیر به‌کار می‌روند مقادیری که یک عملگر روی آنها عمل می‌کند، **عملوند** (Operand) نامیده می‌شوند. در عبارت z = x + y علائم + و = **عملگر** هستند و حروف x، y و z عملوند محسوب می‌شوند. در این بخش به شرح عملگرهای جاوااسکریپت پرداخته می‌شود.



شکل ۸

عملگرهای ریاضی (Arithmetic) : این عملگرها برای انجام انواع عملیات محاسباتی ریاضی مورد استفاده قرار می‌گیرند. شرح این عملگرها در جدول ۱۶ آمده است:

جدول ۱۶ــ عملگرهای ریاضی جاوااسکریپت

عملگر	توضیحات	مثال
+ , - , * , / , %	جمع، تفریق، ضرب و تقسیم و باقی‌مانده	x = y+z , a = a*2, n = n/10
++,--	افزایش و کاهش به میزان یک واحد	x++, y--, ++z, --w
**	توان	x = y**2

عملگرهای انتساب (Assignment): برای اختصاص یک مقدار به یک متغیر از این عملگرها استفاده می‌شود. برخی از این عملگرها ساده و برخی ترکیبی هستند؛ یعنی علاوه بر انتساب، عمل دیگری را نیز روی عملوندها انجام می‌دهند. جدول ۱۷ مجموعه این عملگرها را نمایش می‌دهد.

جدول ۱۷ــ عملگرهای انتساب در جاوااسکریپت

عملگر	توضیحات	مثال	عبارت معادل
=	مقدار عملوند سمت راست (y)، درون عملوند سمت چپ (x) قرار می‌گیرد.	x = y	x = y
+=	حاصل جمع عملوندهای x و y درون عملوند سمت چپ (x) قرار می‌گیرد.	x += y	x = x + y
-=	حاصل تفریق عملوندهای x و y درون عملوند سمت چپ (x) قرار می‌گیرد.	x -= y	x = x – y
*=	حاصل ضرب عملوندهای x و y درون عملوند سمت چپ (x) قرار می‌گیرد.	x *= y	x = x * y
/=	خارج قسمت تقسیم عملوند x بر عملوند y درون متغیر سمت چپ (x) قرار می‌گیرد.	x /= y	x = x / y
%=	باقیمانده تقسیم عملوند x بر عملوند y درون متغیر سمت چپ (x) قرار می‌گیرد.	x %= y	x = x % y
**=	عملوند x را به توان عملوند y می‌رساند و نتیجه را درون متغیر x قرار می‌دهد.	x **= y	x = x ** y

■ **عملگرهای مقایسه‌ای (Comparison):** این عملگرها برای مقایسه دو عملوند به‌کار می‌روند. مجموعه این عملگرها در جدول ۱۸ آمده است:

جدول ۱۸ــ عملگرهای مقایسه‌ای

عملگر	توضیحات
== و ===	== برای بررسی تساوی دو عملوند از لحاظ مقدار و === برای تساوی دو عملوند از لحاظ مقدار و نوع
=! و ==!	=! برای بررسی عدم تساوی مقدار دو عملوند و ==! برای بررسی عدم تساوی مقدار و نوع دو عملوند
> , <	کوچک‌تر یا بزرگ‌تر بودن عملوند اول نسبت به عملوند دوم
=< , >=	کوچک‌تر یا مساوی، بزرگ‌تر یا مساوی
?	عملگر شرطی سه‌تایی که برای انتخاب بین دو مقدار بر اساس یک شرط استفاده می‌شود.

■ **عملگرهای منطقی (Boolean):** از عملگرهای منطقی برای ترکیب یا معکوس کردن شرایط استفاده می‌شود. این عملگرها امکان ترکیب عبارات منطقی را در شرایط مختلف فراهم می‌کنند. جدول ۱۹ عملگرهای منطقی جاوااسکریپت را معرفی می‌کند:

جدول ۱۹ــ عملگرهای منطقی

عملگر	توضیحات
&&	عملگر AND بررسی می‌کند که آیا هر دو عملوند صحیح هستند یا خیر.
||	عملگر OR بررسی می‌کند که آیا حداقل یکی از شرط‌ها درست است یا خیر.
!	عملگر NOT وضعیت درست یا غلط بودن یک شرط را معکوس می‌کند.

کار کارگاهی



کار با انواع عملگرهای جاوااسکریپت

فایل جدیدی با نام workshop3.html ایجاد کنید و تگ <script> را در بخش body بنویسید. سپس اسکریپت زیر را درون آن وارد کنید و نتایج به‌دست آمده را در جاهای خالی بنویسید.

```
1 let x = 10, y = 2, z= 3, w=20, s="10", n = "sara", m="sina";
2 x++;                                    ……
3 y += x;                                 ……
4 --z;                                    ……
5 z *= y;                                 ……
6 x /= 2;                                 ……
7 w %= z;                                 ……
8 console.log("x:"+x+",y:"+y+",z:"+z,",w:"+w); ……
9 let a = (x < y);                        ……
10 Let b = (x > y) && (y == z);           ……
11 Let c = (x === s);                     ……
12 Let d = c || (y != z);                 ……
13 Let e = (n < m);                       ……
14 console.log("a: "+a + " , b:" + b + " , c:" + c , " , d:" + d + " , e: "+e);
15 result = (x === s)? "x is equal to s": "x is not equal to s";   ……
16 console.log(result);                   ……
```

نکته



در هنگام مقایسه دو عملوند توسط عملگرهای == و != نوع داده عملوندها اهمیتی ندارد و فقط مقادیر آنها بررسی می‌شود؛ درحالی‌که عملگر === و !== نوع داده عملوندها را نیز مد نظر دارد.

نکته



عبارت (x===s)? تساوی دو متغیر x و s را بررسی می‌کند. اگر نتیجه مقایسه true (درست) باشد، عبارت قبل از علامت **?** و اگر نتیجه false (غلط) باشد، عبارت بعد از علامت : در متغیر result ذخیره می‌گردد.

آرایه‌ها (Arrays)

متغیرهای آرایه امکان نگهداری چند عنصر با انواع داده‌های مختلف را دارند. در هنگام تعریف یک متغیر از نوع آرایه، مقادیر آرایه درون علامت کروشه نوشته می‌شوند. یک آرایه می‌تواند دارای عناصری با نوع داده یکسان یا متفاوت باشد.

مثال

در مثال زیر سه آرایه با مقادیر رشته‌ای، عددی و ترکیبی تعریف شده است:

```
let cars = ["Opel", "Saab", "Volvo", "BMW", "Hyundai"];
let numbers = [1, 20 , 5 , 25 , 1000 , 10 , 1 , 5];
let data = [123, "Opel", false , 12 , 35];
```

هر یک از عناصر آرایه دارای اندیس هستند. اندیس عناصر آرایه از صفر شروع می‌شود، بنابراین اگر یک آرایه دارای ۵ عنصر باشد، اندیس عنصر اول صفر و اندیس عنصر آخر ۴ خواهد بود. برای دسترسی به عناصر آرایه از اندیس عناصر استفاده می‌شود. شکل زیر نمایی از عناصر آرایه cars و مقادیر آنها را نمایش می‌دهد:

مقدار عناصر	Opel	Saab	Volvo	BMW	Hyundai
اندیس عناصر	۰	۱	۲	۳	۴

شکل ۹

برای دسترسی به عنصر اول آرایه از عبارت cars[۰] و برای دسترسی به عنصر چهارم از عبارت cars[۴] استفاده می‌شود.

مثال

در خط دوم مثال زیر مقدار عنصر سوم آرایه cars و در خط سوم کل آرایه درون کنسول نمایش داده می‌شود.

```
let cars = ["Opel","Saab","Volvo","BMW","Hyundai"];
console.log(cars[2]);
console.log(cars);
```



چنانچه اندیس عنصر آرایه مشخص نشود، کل عناصر آرایه روی صفحه چاپ می‌شود.

نکته

با تعریف آرایه توسط کلمه کلیدی const امکان تغییر عناصر آرایه موجود است؛ اما متغیر آرایه نمی‌تواند به آرایه دیگری نسبت داده شود. به‌عنوان مثال نوشتن عبارت زیر با خطا مواجه می‌شود:

```
cars = ["Tara", "Shahin", "Dena"," Lamari Eama"]
```

کار کارگاهی

ایجاد آرایه و چاپ مقادیر

۱ فایل جدیدی با نام workshop4.html ایجاد کنید و درون آن آرایه‌ای با نام colors با مقادیر yellow، blue، red، green و orange تعریف کنید. بعد از بررسی صحت برنامه، دستورات را در جاهای خالی بنویسید.

۲ دستوری بنویسید که عنصر blue را در کنسول مرورگر نمایش دهد.

.....

۳ دستوری بنویسید که تمام عناصر آرایه colors را به یک باره چاپ نماید.

.....

بیشتر بدانید

آرایه‌های چندبعدی

در بخش قبل نحوهٔ ایجاد یک آرایهٔ تک‌بعدی و دسترسی به عناصر آن معرفی شد. جاوااسکریپت امکان تعریف آرایه چندبعدی را نیز فراهم می‌کند. آرایه چند بعدی زمانی ایجاد می‌شود که هر یک عناصر آرایه محتوای آرایه دیگری باشند. رایج‌ترین نوع آرایه چندبعدی، آرایه دو بعدی است که می‌تواند به عنوان ماتریس استفاده شود.

در مثال زیر یک آرایهٔ دو بعدی با نام users تعریف شده است:

```
let users = [
            ["Arash", "Irani", 35, "Teacher"],
            ["Sara", "Rostami", 40, "Doctor"],
            ["Saba", "Farsi", 20, "Student"],
        ];
```

Arash [0][0]	Irani [0][1]	35 [0][2]	Teacher [0][3]
Sara [1][0]	Rostami [1][1]	40 [1][2]	Doctor [1][3]
Saba [2][0]	Farsi [2][1]	20 [2][2]	Student [2][3]

برای دسترسی به عنصر اول آرایهٔ اول (Arash) باید از عبارت users[0][0] استفاده شود و برای دسترسی به عنصر چهارم آرایهٔ دوم (Doctor) از عبارت users[1][3]. تصویر سمت راست، اندیس عناصر آرایهٔ دوبعدی users را نمایش می‌دهد.

کار با آرایه‌های دو بعدی

در فایل workshop4.html محتوای آرایهٔ users (مثال قبل) را بنویسید.

۱ دستورات زیر را بعد از معرفی آرایه وارد کنید و خروجی هریک از دستورات را در کنسول مرورگر بررسی کنید. نتیجه را مقابل هر دستور یادداشت کنید.

console.log(users); .....

console.log(users[1]); .....

console.log(users[2][3]); .....

۲ دستوری بنویسید که باعث نمایش عنصر Rose شود.

.....

اشیا در جاوااسکریپت

شیء (Object) یک ساختار داده است که برای ذخیرهٔ مجموعه‌ای از داده‌های مرتبط، به‌صورت جفت‌های کلید: مقدار استفاده می‌شود، در ساختار داده شیء، مجموعه‌ای از کلیدها (keys) به مقادیر (values) متناظر خود اشاره می‌کنند. اشیا برای ذخیره و سازماندهی داده‌ها و رفتارهای یک موجودیت استفاده می‌شوند. که ساختاری شبیه به دیکشنری در پاتیون دارد.

مثال

در مثال زیر شیء person دارای چهار ویژگی نام، نام خانوادگی، سن و رنگ چشم است که هر یک دارای مقدار مشخصی هستند:

```
const person = {firstName:"Arash", lastName:"Irani", age:35, eyeColor:"blue"};
```

در هنگام تعریف یک شیء استفاده از فواصل و ایجاد سطر جدید اهمیتی ندارد؛ بنابراین شیء person می‌تواند به‌صورت زیر نیز تعریف شود:

```
const person = {
    firstName: "Arash",
    lastName: "IRANI",
    age: 35,
    eyeColor: "blue"
};
```

دسترسی به مقدار ویژگی‌های یک شیء از طریق علامت نقطه یا کروشه امکان‌پذیر است. هر دو دستور زیر مقدار ویژگی firstName در شیء person را نمایش می‌دهد:

```
console.log(person.firstName);
console.log(person['firstName']);
```

بیشتر بدانید

یک شیء علاوه بر ویژگی‌ها دارای رفتارهایی است که با عنوان متد تعریف می‌شوند. تعریف متدها معمولاً در انتهای جفت‌های کلید-مقدار شیء انجام می‌شود.

روش جدید

```
methodName(){
 // دستوراتی که متد انجام می‌دهد
}
```

روش قدیم

```
methodName:function(){
 // دستوراتی که متد انجام می‌دهد
}
```

در خطوط ۱۳ تا ۱۵ کد صفحه بعد متد displayInfo() برای شیء person تعریف شده و در خط ۱۷ فراخوانی آن انجام شده است.

بیشتر بدانید

```
const person = {
    firstName: "Arash",
    lastName: "Irani",
    age: 35,
    eyeColor: "blue",
    displayInfo() {
        return `Name: ${this.firstName} ${this.lastName}, Age: ${this.age}, Eye Color: ${this.eyeColor}`;
    }
}
console.log(person.displayInfo());
```

در تعریف فوق، کلمه کلیدی **this** به شیء جاری (person) اشاره می‌کند و برای دسترسی به ویژگی‌های شیء جاری به‌کار می‌رود. این متد اطلاعات شیء person را به محل فراخوانی برمی‌گرداند.
برای فراخوانی متد شیء، ابتدا نام شیء (Person) و سپس نام متد(displayInfo) به همراه علائم () نوشته می‌شود.
رشته‌ای که توسط متد displayInfo() به محل فراخوانی برگردانده (return) می‌شود، باید درون علامت backtick قرار گیرد. این علامت در گوشه بالای صفحه کلید، زیر دکمه esc واقع شده است.

کار کارگاهی

ایجاد شیء در جاوااسکریپت

۱ فایل جدیدی با نام workshop5.html ایجاد کنید.

۲ یک شیء با نام book با ویژگی‌های title، author و year و مقادیر "JavaScript"، "Douglas Crockford" و ۲۰۰۸ ایجاد کنید.

.....

۳ دستوری بنویسید که مقدار ویژگی author در شیء book را چاپ نماید. .....

بیشتر بدانید

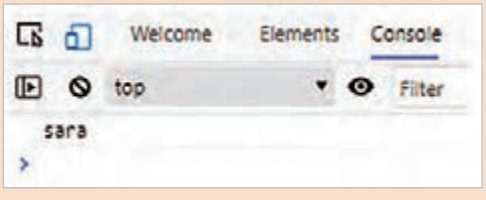
ترکیب اشیا و آرایه‌ها

جاوااسکریپت امکان ترکیب اشیا و آرایه‌ها را با هم فراهم می‌کند. برنامه‌نویس جاوااسکریپت می‌تواند آرایه‌ای از اشیا یا یک شیء شامل آرایه را ایجاد نماید.

مثال

مثال زیر آرایه‌ای از اشیا را ایجاد می‌کند:

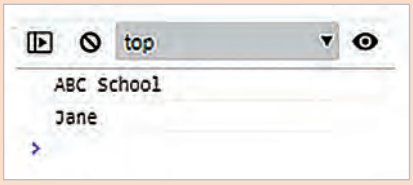
```
let students = [
    { name: 'sara', age: 20 },
    { name: 'mona', age: 22 }
];
console.log(students[0].name);
```



مثال

در مثال زیر شیء school حاوی آرایه students است:

```
14  let school = {
15    name: 'ABC School',
16    students: students
17  };
18  console.log(school.name);
19  console.log(school.students[1].name);
```



فعالیت

کار با اشیا و آرایه‌ها

در فایل workshop5.html آرایه students و school را مطابق مثال فوق تعریف کنید و نتایج را در کنسول مرورگر بررسی کنید

دستوری بنویسید که نام و سن خانم mona را در کنسول نمایش دهد.

.....

ساختارهای شرطی و حلقه در جاوااسکریپت

ساختارهای شرطی و حلقه از اجزای اساسی زبان‌های برنامه‌نویسی، از جمله جاوااسکریپت هستند. ساختارهای شرطی منطق برنامه را بر اساس شرایط خاص کنترل می‌کنند و ساختارهای حلقه عملیات خاصی را به‌صورت تکراری انجام می‌دهند. این بخش به معرفی این ساختارها می‌پردازد.

دستورات شرطی If

یک دستور شرطی امکان اجرای مجموعه‌ای از دستورات را به ازای برقراری یک شرط خاص فراهم می‌کند. بسته به نیاز برنامه، دستور if می‌تواند به سه صورت نوشته شود. جدول ۲۰ شکل کلی ساختار شرطی if را در سه حالت مختلف نمایش می‌دهد.

جدول ۲۰ـ انواع ساختارهای شرطی if

if ساده	if-else	else-if
if (condition){ statements **}**	**if (condition){** Statement1 **}else{** Statement2 **}**	if (**condition1**) { statement1 } else if (**condition2**) { statement2 } else { statement3 }

کلمه Condition به‌معنای شرط و کلمه statement به دستور یا دستوراتی اشاره دارد که به‌ازای برقرار شدن شرط اجرا می‌شوند. منظور از برقرار بودن شرط این است که مقدار عبارت Condition برابر True باشد. کلمه else به‌معنای «درغیراین‌صورت» است. محتوای else زمانی اجرا می‌شود که شرط if برقرار نباشد. یعنی مقدار عبارت condition برابر False باشد.

کار کارگاهی

کار با ساختارهای شرطی if

توسط ساختار شرطی **if**، زوج یا فرد بودن عدد n را بررسی کنید.

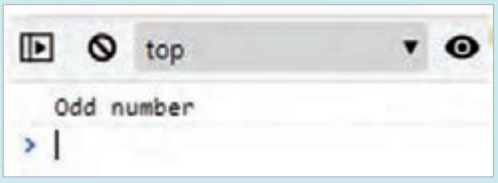
بدین منظور یک فایل جدید با نام workshop6.html ایجاد کنید و اسکریپت زیر را درون آن بنویسید:

فایل را ذخیره کنید و نتیجه را در کنسول مرورگر مشاهده نمایید. در این اسکریپت، باقی‌ماندهٔ تقسیم عدد n بر ۲ درون متغیر r ذخیره می‌شود. چنانچه مقدار باقی‌مانده صفر باشد (یعنی n بر ۲ بخش‌پذیر است)، عبارت "Even number" و در غیراین‌صورت عبارت "Odd number" روی صفحه ظاهر می‌شود. کلمه Even به‌معنای زوج و کلمه Odd به معنای فرد است.

```
let n = 17;
let r = n % 2;
if (r==0){
    console.log("Even number");
}
console.log("Odd number");
```

اسکریپت تشخیص زوج یا فرد بودن یک عدد می‌تواند توسط ساختار **if-else** نیز نوشته شود. اسکریپت قبل را طبق شکل زیر تغییر دهید و نتیجه را در کنسول مرورگر بررسی کنید.

```
if (r==0){
    console.log("Even number");
}else{
    console.log("Odd number");
}
```



ساختار if-else: امکان بررسی چند شرط را به‌صورت پی‌درپی فراهم می‌کند. در این مرحله رتبه یک دانش‌آموز بر اساس نمره کسب‌شده مشخص می‌شود. در انتهای اسکریپت قبل، کد روبه‌رو را بنویسید.

در این اسکریپت، متغیر score با مقدار ۷۰ برای نگهداری نمره دانش‌آموز تعریف شده است. ابتدا شرط score>=90 و سپس شرط score>=80 بررسی می‌شود. با توجه به اینکه

```
let score = 70;
if (score >= 90) {
    console.log("Grade: A");
} else if (score >= 80) {
    console.log("Grade: B");
} else if (score >= 70) {
    console.log("Grade: C");
} else if (score >= 60) {
    console.log("Grade: D");
} else {
    console.log("Grade: F");
}
```

شکل ۱۰

مقدار هر دو شرط نادرست است، بنابراین دستور console.log() در خطوط ۳ و ۵ اجرا نمی‌شود. سپس شرط score>=70 بررسی می‌شود. با توجه به برقرار بودن شرط، دستور شماره ۷ اجرا شده و عبارت Grade: C روی صفحه چاپ می‌شود. بعد از اجرای دستور ۷ شرط‌های بعدی بررسی نمی‌شود و کنترل برنامه به خط ۱۳ منتقل می‌شود.

بیشتر بدانید

ساختار Switch

ساختار switch برای انتخاب از بین چندین بلوک کد، بر اساس مقدار یک متغیر به‌کار می‌رود. درصورتی‌که شرایط مسئله فقط به مقایسه یک متغیر با مقادیر مشخص محدود شود، می‌توان از ساختار switch به‌جای else-ifهای متعدد استفاده نمود.

کار کارگاهی

استفاده از ساختار شرطی switch

اسکریپتی بنویسید که رتبهٔ یک دانش‌آموز را از متغیر grade دریافت کند و بر اساس مقدار رتبه، پیغام مناسبی را برای کاربر نمایش دهد.

بدین منظور اسکریپت موجود در فایل workshop6.html را درون علائم کامنت قرار دهید و در ادامه، کد روبه‌رو را برای بررسی رتبه بنویسید.

در این مثال ساختار شرطی switch مقدار متغیر grade را توسط caseهای مختلف بررسی می‌کند، به محض اینکه مقدار مقابل case با مقدار متغیر grade یکسان شود، دستور بعد از علامت: اجرا شده و سپس خروج از ساختار switch توسط دستور break انجام می‌شود. اگر مقدار متغیر grade با هیچ یک از مقادیر نوشته شده در مقابل caseها یکسان نباشد، دستور مربوط به حالت default اجرا می‌گردد.

```
let grade = "A";
let result;

switch (grade) {
  case "A":
    result = "Excellent";
    break;
  case "B":
    result = "Good";
    break;
  case "C":
    result = "Pass";
    break;
  default:
    result = "Unknown";
}
console.log(result);
```

نکته

اگر شرط یک case برقرار شود ولی دستور break در آن موجود نباشد، case بعدی هم توسط برنامه بررسی می‌شود.

فعالیت

در case A دستور break را حذف کنید و خروجی برنامه را در مرورگر بررسی کنید. سپس دستور break را از مقابل caseهای بعدی حذف کنید و نتیجهٔ آن را تحلیل نمایید.

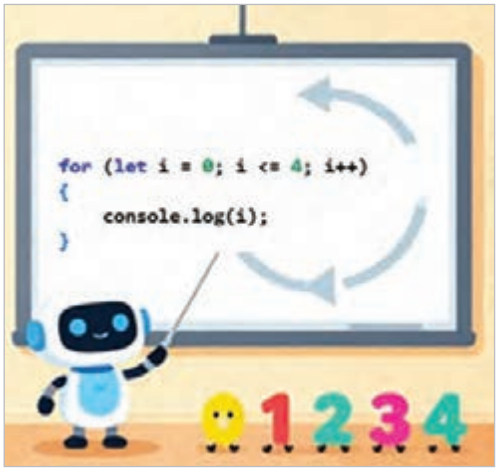
دستورات حلقه

در زبان‌های برنامه‌نویسی برای تکرار اجرای یک بخش از برنامه از دستورات حلقه استفاده می‌شود. حلقه‌های تکرار باعث کاهش حجم برنامه، خوانایی کد، بهینه‌سازی الگوریتم برنامه و افزایش کارایی برنامه می‌شوند. حلقه‌های تکرار انواع مختلفی دارند. در زبان جاوااسکریپت این حلقه‌ها شامل for، while، do while، for...of و for...in است.

حلقهٔ for

زمانی که تعداد دفعات تکرار یک کار مشخص باشد، از حلقهٔ for استفاده می‌شود. شکل کلی حلقهٔ for به‌صورت زیر است:

```
for (Initialization; Condition; Increment/Decrement) {
کدی که باید تکرار شود
}
```



شکل ۱۱

حلقهٔ for نیازمند یک شمارنده برای شمارش تعداد دفعات اجرای حلقه است. شمارنده حلقه یک متغیر است که دارای مقدار اولیه و پایانی است. عبارت Initiliazation به مقداردهی اولیه شمارنده حلقه اشاره دارد. در قسمت Condition شرط تکرار حلقه بررسی می‌شود. برای خروج از حلقه، شرط حلقه باید نقض شود. بدین منظور شمارنده حلقه باید در هر بار تکرار حلقه افزوده یا کاسته شود تا مقدار شمارنده از مقدار پایانی گذر کند. عبارات Increment و Decrement به‌معنای افزایش و کاهش مقدار شمارنده است. در صورت عدم استفاده از آکولاد برای تعیین محدوده بلوک حلقه، فقط اولین دستور جزء حلقه به شمار می‌آید.

کار کارگاهی

به‌کارگیری حلقه for

یک فایل جدید با نام workshop7.html ایجاد کنید و اسکریپت روبه‌رو را برای چاپ اعداد ۰ تا ۴ بنویسید.

```
for (let i = 0; i <= 4; i++) {
    console.log(i);
}
```

نکته

چنانچه شمارنده حلقه توسط کلمه کلیدی let تعریف شود، دسترسی به مقدار متغیر شمارنده فقط در داخل بلوک for امکان پذیر خواهد بود.

با توجه به این توضیحات درصورتی که بعد از علامت آکولاد بسته، دستور console.log(i) نوشته شود، پیغام خطا صادر می‌شود.

بیشتر بدانید

حلقه while و do while

این حلقه‌ها مجموعه‌ای از دستورات را تا زمان نقض شدن یک شرط، تکرار می‌کنند. شکل کلی این حلقه‌ها به‌صورت مقابل است:

جدول ۲۱

do while	while
do { کدی که باید تکرار اجرا شود } **while (condition);**	**while (condition)**{ کدی که باید تکرار شود }

در حلقه while، ابتدا شرط داخل پرانتز بررسی می‌شود، اگر شرط برقرار باشد دستورات داخل حلقه اجرا می‌شود، اگر شرط حلقه از ابتدا غلط باشد بنابراین محتوای حلقه اصلاً اجرا نمی‌شود.

محتوای حلقه do while یک بار اجرا می‌شود، سپس شرط حلقه بررسی می‌شود. در صورت برقرار نبودن شرط، خروج از حلقه صورت می‌پذیرد.

در صورتی‌که شرط حلقه نقض نشود، دستورات حلقه به‌طور مداوم تکرار می‌شوند. این وضعیت با عنوان حلقه بی‌نهایت (Infinite loop) شناخته می‌شود. حلقه بی‌نهایت می‌تواند منابع سیستم (مانند CPU و RAM) را به‌صورت بی‌رویه مصرف کند که همین مسئله موجب اختلال در عملکرد مرورگر خواهد شد.

به‌کارگیری حلقه while و do while

در فایل workshop7.html اسکریپت زیر را برای چاپ عناصر آرایه colors با استفاده از حلقه while بنویسید.

```
const colors = ["Yellow", "Blue", "Orange", "Red", "Purple"];
let index = 0;
while (index < 5) {
    console.log(colors[index]);
    index++;
}
```

```
top
Yellow
Blue
Orange
Red
Purple
```

فایل را ذخیره کنید و خروجی آن را در مرورگر به نمایش بگذارید.

حاصل جمع عناصر آرایه numbers را به کمک یک حلقه do while محاسبه کنید و نتیجه را در کنسول نمایش دهید.

```
const numbers = [1, 3, 5, 7, 9, 11, 13, 15, 17, 19];
let sum = 0, counter = 0;
do {
    sum += numbers[counter];  //sum = sum + numbers[counter]
    counter++;
} while (counter < 10);
console.log("The sum of values stored in the numbers array is: " + sum);
```

```
top  Filter  Default
The sum of values stored in the numbers array is: 100
>
```

■ **حلقه for...of**: این حلقه برای پیمایش عناصر یک مجموعه قابل تکرار مانند آرایه، رشته، مجموعه و... استفاده می‌شود. حلقه for...of امکان دسترسی به مقادیر عناصر آرایه را فراهم می‌کند.

■ **حلقه for ... in**: این حلقه برای پیمایش کلیدهای اشیا (Objects) و اندیس آرایه‌ها استفاده می‌شود.
شکل کلی ساختار for ... of و for ... in به‌صورت زیر است:

جدول ۲۲

for ... of	for ... in
for (const element **of** iterable) { ..... }	**for** (const key **in** object) { ..... }

متغیر element به عناصر آرایه، مجموعه یا کاراکترهای یک رشته و iterable به شیء قابل تکرار اشاره دارد.
متغیر key به کلید عناصر شیء و اندیس عناصر آرایه اشاره دارد.

کار کارگاهی

کار با ساختار حلقه for ... of و for ... in

۱ در فایل workshop7.html اسکریپت زیر را بنویسید و نتیجه را در کنسول مرورگر مشاهده کنید.

```
let firstNames = ['Sara','Raha','Mina','Sudabeh','Bahareh'];
for (const name of firstNames){
    console.log(name);
}
```

```
Sara
Raha
Mina
Sudabeh
Bahareh
```

متغیر name به عناصر آرایه firstName اشاره دارد؛ در هر بار اجرای حلقه، متغیر name به عنصر جدیدی از آرایه اشاره می‌کند.

کنجکاوی

در عبارت (**for** (const name **of** firstNames چرا از کلمه کلیدی const استفاده شده است؟

۲ برای بررسی عملکرد حلقه for ... in در پیمایش عناصر اشیا، اسکریپت زیر را بنویسید و نتیجه را در کنسول مرورگر بررسی کنید.

```
const languages = {en:"انگلیسی", fa:"فارسی", es:"اسپانیایی", fr:"فرانسوی"};
for(const lang in languages){
  console.log(lang + ":" + languages[lang]);
}
```

```
en:انگلیسی
fa:فارسی
es:اسپانیایی
fr:فرانسوی
```

در کد فوق متغیر languages یک شیء حاوی اسامی زبان‌هاست. در این شیء واژه‌های مخفف (مانند en، fa و...) به عنوان کلید عنصر و عبارت فارسی متناظر آن (مانند انگلیسی، فارسی و...) به عنوان مقدار عنصر مشخص شده است.
در این کد، حلقه for...in عناصر شیء languages را یک‌به‌یک پیمایش می‌کند و کلید هر عنصر را درون متغیر lang قرار می‌دهد. سپس دستور console.log() نام کلید و مقدار متناظر با آن را در کنسول نمایش می‌دهد.

دستور break

این دستور برای خروج از حلقه یا ساختار Switch به‌کار می‌رود. اگر دستور break درون حلقه استفاده شود، اجرای تمام دستورات بعد از آن متوقف شده و خروج از حلقه صورت می‌گیرد.

نکته

اگر این دستور درون ساختار switch استفاده شود، Caseهای بعدی بررسی نخواهند شد و کنترل برنامه به بعد از آکولاد بسته منتقل می‌شود.

کار کارگاهی

به‌کارگیری دستور Break

اسکریپتی بنویسید که اعداد دو رقمی آرایه زیر را در کنسول مرورگر نمایش دهد.

```
let data = [10,21,37,43,76,83,98,356,200,353,444,1000];
```

برای این‌کار در انتهای فایل workshop7.html، اسکریپت زیر را وارد کنید. سپس فایل را ذخیره کرده و خروجی آن را در کنسول به نمایش بگذارید.

```
let data = [10,21,37,43,76,83,98,356,200,353,444,1000];
let len = data.length;
for (let i = 0; i < len; i++){
    if (data [i] < 100){
        console.log(data [i]);
    }else
        break;
}
```

```
10
21
37
43
76
83
98
```

دستور continue

به‌کارگیری دستور continue در یک حلقه باعث می‌شود که اجرای فعلی حلقه متوقف و دور بعدی حلقه اجرا شود. به این معنی که بعد از اجرای دستور continue کنترل برنامه به ابتدای حلقه منتقل می‌شود و دور بعدی حلقه (در صورت امکان) اجرا می‌شود.

کار کارگاهی

به‌کارگیری دستور Continue

اعداد زوج بین ۲۰ تا ۳۵ را در کنار هم با جداکنندهٔ خط فاصله (-) نمایش دهید.

در انتهای اسکریپت فایل workshop7.html، اسکریپت زیر را بنویسید و نتیجه را در کنسول بررسی کنید.

```
let m=20;
let n=35;
let evenNumbers ="";
for (let num  =m; num <= n; num++){
   if (num %2 != 0)
      continue;
      evenNumbers = evenNumbers + num + " - ";
}
console.log(`Even Numbers between ${m} and ${n} are: ${evenNumbers}`);
```

```
Even Numbers between 20 and 35 are: 20 - 22 - 24 - 26 - 28 - 30 - 32 - 34 -
>
```

برای حذف خط فاصلهٔ آخر (-)، دستور زیر را قبل از دستور console.log بنویسید:

evenNumbers = evenNumbers.slice(0, -3);

بیشتر بدانید

حلقه‌های تو در تو

حلقه‌هایی که تاکنون معرفی شد، امکان پیمایش عناصر تک‌بعدی را فراهم می‌کنند. برای پیمایش عناصری همچون آرایه دوبعدی نیاز به استفاده از حلقه‌های تودرتو است. این نوع حلقه زمانی ایجاد می‌شود که درون یک حلقه، حلقه دیگری قرار گیرد. در یک حلقهٔ تودرتو، حلقهٔ داخلی به تعداد دفعات اجرای حلقه خارجی تکرار می‌شود.

بیشتر بدانید

به‌کارگیری حلقهٔ تودرتو

۱ـ جدول ضرب ۵*۵ را با استفاده از یک حلقه تودرتو نمایش دهید.

۲ـ اسکریپت زیر را در انتهای اسکریپت فایل workshop7.html بنویسید و نتیجه را کنسول بررسی کنید.

```
for (let i = 1; i <= 5; i++) {
    let row = "";
    for (let j = 1; j <= 5; j++) {
        row += i * j + "\t";
    }
    console.log(row);
}
```

```
1   2   3   4   5
2   4   6   8   10
3   6   9   12  15
4   8   12  16  20
5   10  15  20  25
>
```

علامت \t به معنای tab است و برای تنظیم فاصله بین مقادیر استفاده شده است.

فعالیت

به‌جای دستور console.log از دستور document.write برای نمایش مقادیر در صفحه مرورگر استفاده کنید.

کار کارگاهی

پیمایش آرایه دوبعدی با استفاده از حلقه for

فایل product.html را باز کنید و کدهای جاوااسکریپت زیر را به آن اضافه کنید.

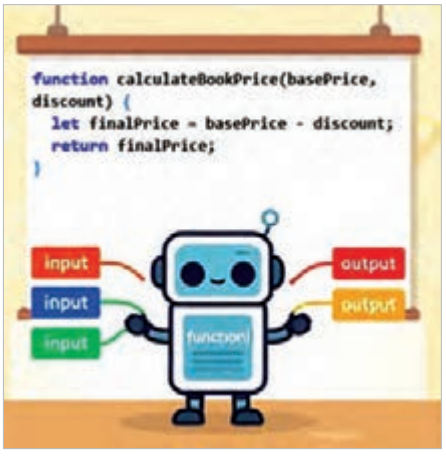
```
<script>
let books = [
    ["دهم","شبکه و نرم افزار","دانش فنی پایه"],
    ["دهم","شبکه و نرم افزار","نگهداری سیستم‌های رایانه‌ای"],
    ["دهم","شبکه و نرم افزار","ارائه دهنده خدمات رایانه‌ای"],
];
for (let i = 0; i <= books.length-1; i++) {
    for (let j = 0; j <= books[i].length-1; j++) {
        console.log(books[i][j]);
    }
    console.log(" _____ ");
}
```

```
دانش فنی پایه
شبکه و نرم افزار
دهم
 _____
نگهداری سیستم‌های رایانه‌ای
شبکه و نرم افزار
دهم
 _____
ارائه دهنده خدمات رایانه‌ای
شبکه و نرم افزار
دهم
 _____
```

توابع (Functions)

در زبان‌های برنامه‌نویسی به بلوک‌های کدی که یک کار مشخص را انجام می‌دهند، تابع گفته می‌شود. برنامه‌نویس می‌تواند مجموعه‌ای از دستورالعمل‌ها را در قالب یک تابع گروه‌بندی کند و سپس آن را به عنوان یک واحد مجزا مورد استفاده قرار دهد. این واحدهای مجزا می‌توانند بارها مورد استفاده قرار گیرند. استفاده از توابع در برنامه‌نویسی نه‌تنها سرعت توسعه را افزایش می‌دهد، بلکه خوانایی و قابلیت نگهداری کد را نیز افزایش می‌دهد. مجموعه دستورات نوشته‌شده برای ایجاد یک تابع، تحت یک نام مشخص قابل فراخوانی هستند. یک تابع می‌تواند مقادیری را به‌عنوان ورودی دریافت کند و بعد از اجرای دستورات داخل تابع، مقداری را به‌عنوان خروجی برگرداند.

در جاوااسکریپت برای ایجاد یک تابع از شکل کلی زیر استفاده می‌شود:




```
function functionName(parameters) {

}
```

شکل ۱۲

اصول تعریف تابع در جاوااسکریپت به شرح زیر است:

- در تعریف تابع ابتدا عبارت کلیدی function و سپس نام آن نوشته می‌شود.
- نام تابع باید با حرف کوچک شروع شود و هر کلمه جدید با حرف بزرگ آغاز شود.
- پارامترهای ورودی تابع درون پرانتز مشخص می‌شود.
- تعداد پارامترها می‌تواند صفر، یک یا بیشتر باشد. مجموعه دستورات تابع درون آکولاد نوشته می‌شوند.
- خروجی تابع معمولاً توسط تابع return به محل فراخوانی برگردانده می‌شود.
- دستورات درون تابع زمانی اجرا می‌شوند که تابع فراخوانی شود.
- برای فراخوانی تابع باید از نام تابع به همراه آرگومان‌های ورودی استفاده شود. شکل کلی فراخوانی تابع به‌صورت زیر است:

functionName(Arguments)

مثال

در مثال زیر تابعی به نام calculateCartTotal با سه پارامتر ورودی price3, price2, price1 ایجاد شده است. تابع calculateCartTotal مجموع مقادیر این سه پارامتر را محاسبه و درون متغیر total ذخیره می‌کند. دستور return مقدار متغیر total را به محل فراخوانی بازمی‌گرداند.

```
1  function calculateCartTotal(price1,price2,price3){
2      let total = price1 + price2 + price3;
3      return total;
4  }
5  // فراخوانی تابع و ارسال قیمت سه محصول
6  const totalPrice = calculateCartTotal(250000,145000,99000);
7
8  console.log("مجموع مبالغ سبد خرید: ",totalPrice, " تومان");
```

```
top  Filter
مجموع مبالغ سبد خرید:  494000  تومان
```

نکته

پارامترها متغیرهایی هستند که در هنگام تعریف تابع استفاده می‌شوند. در مثال فوق متغیرهای price3, price2, price1 پارامتر ورودی تابع هستند. آرگومان به مقادیری گفته می‌شود که هنگام فراخوانی برای تابع ارسال می‌شوند. به عبارت دیگر آرگومان‌ها مقادیری هستند که به پارامترهای تابع نسبت داده می‌شوند.

کار کارگاهی

ایجاد تابع و فراخوانی آن

فایل جدیدی با نام workshop8.html ایجاد کنید و تابع calculateCartTotal را درون آن بنویسید. سپس تابع را با آرگومان‌های ورودی دلخواه فراخوانی کنید.

یک تابع لزوماً دارای مقدار بازگشتی نیست. یک تابع می‌تواند با هدف اجرای مجموعه‌ای از دستورالعمل‌ها نوشته شود، بدون اینکه مقداری را به محل فراخوانی بازگرداند. در این‌صورت نیازی به استفاده از دستور return نخواهد بود. در مثال زیر تابع showCartTotal سه مقدار را به‌عنوان پارامتر ورودی دریافت می‌کند، حاصل جمع آنها را محاسبه کرده و نتیجه را در کنسول مرورگر نمایش می‌دهد؛ اما هیچ مقداری را به محل فراخوانی برنمی‌گرداند.

```
1  function showCartTotal(price1,price2,price3){
2      let total = price1 + price2 + price3;
3      console.log("مجموع مبالغ سبد خرید: ",total, " تومان");
4  }
5  // فراخوانی تابع و ارسال قیمت سه محصول
6  showCartTotal(250000,100000,50000);
```

```
مجموع مبالغ سبد خرید:  400000  تومان
>
```

توابع آماده جاوااسکریپت (Built-in Functions)

جاوااسکریپت دارای مجموعه‌ای از توابع آماده است که برای انجام عملیات مختلف به‌کار می‌روند. توابع آماده جاوااسکریپت به برنامه‌نویسان امکان می‌دهند که به‌راحتی داده‌ها را مدیریت کرده و وظایف پیچیده را به سادگی پیاده‌سازی کنند. این توابع شامل توابع ریاضی، رشته‌ای، آرایه، اشیا، ورودی-خروجی و تاریخ و زمان است که در این بخش به معرفی آنها پرداخته می‌شود.

کار کارگاهی

به کارگیری توابع ریاضی جاوااسکریپت

توابع ریاضی جاوااسکریپت که درون شیء Math سازماندهی شده‌اند، مجموعه‌ای از ابزارهای قدرتمند و کارآمد برای انجام محاسبات پیچیده ریاضی هستند. دسترسی به توابع ریاضی به‌صورت نقطه‌ای (Math.functionName) امکان‌پذیر است و استفاده از آنها نه‌تنها کدنویسی را کوتاه‌تر و خواناتر می‌کند، بلکه از خطاهای احتمالی در پیاده‌سازی دستی الگوریتم‌ها نیز جلوگیری کرده و عملکرد بهینه‌تری در برنامه‌های کاربردی ایجاد می‌نماید. چند نمونه از توابع پر کاربرد ریاضی در جاوااسکریپت به شرح زیر است:

- Math.abs(): قدرمطلق مقدار آرگومان ورودی را برمی‌گرداند.
- Math.round(): مقدار آرگومان ورودی را گرد می‌کند.
- Math.random(): یک عدد تصادفی بین ۰ (شامل) تا ۱ (غیرشامل) برمی‌گرداند.
- Math.max(): بزرگ‌ترین عدد را از بین آرگومان‌های ورودی برمی‌گرداند.
- Math.min(): کوچک‌ترین عدد را از بین آرگومان‌های ورودی برمی‌گرداند.

در طراحی فروشگاه‌های اینترنتی، توابع ریاضی جاوااسکریپت نقشی حیاتی ایفا می‌کنند، زیرا مدیریت عملیات اصلی مانند محاسبه قیمت نهایی پس از اعمال تخفیف، جمع‌بندی اقلام سبد خرید و تعیین هزینه‌های حمل‌ونقل مستلزم استفاده دقیق از این توابع است. به‌منظور آشنایی با نحوهٔ فراخوانی و عملکرد این توابع، فایلی به‌نام shopping-cart.html ایجاد کنید و اسکریپت زیر را در آن بنویسید. سپس نتایج به‌دست‌آمده را در کلاس بررسی و تحلیل کنید.

```
const cart = [1250000, 850000, 499000];    //لیست قیمت محصولات
let total=0;                               // مقدار اولیه جمع مبالغ محصول
for(let price of cart){
    total+=price;
}
console.log("مجموع قیمت‌ها: ", total);

let itemCount = cart.length;               // تعداد کالاها
console.log("تعداد کالاها :", itemCount);

// بیشترین و کمترین قیمت برای اسلایدر قیمت
let maxPrice = Math.max(...cart);
let minPrice = Math.min(...cart);
console.log("گرانترین",maxPrice);
console.log("ارزانترین",minPrice);

// اعمال تخفیف 15 درصد
let discountPercent = 15;
let discountAmount = total * (discountPercent/100);
let totalAfterDiscount = total - discountAmount;
console.log("بعد از تخفیف: ", totalAfterDiscount);

// افزودن مالیات 9 درصد
let tax = totalAfterDiscount * 0.09;
let totalAfterTax = totalAfterDiscount +tax;
console.log("مجموع هزینه بعد از اعمال مالیات", totalAfterTax);

// هزینه ارسال- اگر کمتر از 3 میلیون باشد
let shipping =totalAfterTax<3000000? 50000: 0;
console.log("هزینه ارسال",shipping);

// قیمت نهایی
let finalPrice = totalAfterTax + shipping;

//گرد کردن قیمت نهایی
finalPrice = Math.round(finalPrice);
console.log("قیمت نهایی گرد شده", finalPrice);

//ساخت کد تخفیف تصادفی
let discountCode = Math.random().toString(36).substring(2,8).toUpperCase();
console.log("کد تخفیف: ", discountCode);
```

```
مجموع قیمت‌ها: 2599000
تعداد کالاها : 3
گرانترین 1250000
ارزانترین 499000
قیمت با تخفیف: 2209150
مجموع هزینه بعد از اعمال مالیات 2407973.5
هزینه ارسال 50000
قیمت نهایی گرد شده: 2457974
کد تخفیف: LH7LGP
```

به کارگیری توابع رشته ای در جاوااسکریپت

توابع و ویژگی‌های رشته‌ای جاوااسکریپت، ابزارهای ضروری برای پردازش و مدیریت متن هستند که امکان قالب‌بندی و تغییر رشته‌ها را با دقت و کارایی بالا فراهم می‌کنند. این متدها امکان اعتبارسنجی داده‌های ورودی کاربران، بهینه‌سازی متن برای نمایش یا ارسال به سرور را فراهم می‌کنند. برخی از توابع رشته‌ای جاوااسکریپت به شرح زیر است:

()toUpperCase: حروف رشته ورودی را به حالت بزرگ در می‌آورد.

()indexOf: اندیس موقعیت کاراکتر یا رشتهٔ داخل پرانتز را در رشته اصلی برمی‌گرداند.

()substring: یک زیررشته را از رشته اصلی برمی‌گرداند. ورودی‌های این تابع اندیس ابتدا و انتهای زیر رشته است.

()split: رشته را بر اساس جداکننده داخل پرانتز (مانند کاما) تقسیم می‌کند و آرایه‌ای از رشته‌ها را برمی‌گرداند.

()replace: دو رشته را به عنوان ورودی دریافت کرده و رشتهٔ دوم (/) را به‌جای رشتهٔ اول (-) در رشته اصلی قرار می‌دهد. این تابع در اولین موقعیت یافتن رشتهٔ مورد نظر(-)، جایگزینی رشته را انجام می‌دهد.

()startsWith: وجود یک حرف یا رشته را در ابتدای یک متغیر رشته‌ای بررسی می‌کند.

شیء string دارای یک ویژگی به‌نام **length** است که طول یک رشته را تعیین می‌کند. بر این اساس، دستور زیر مقدار ۱۱ را نمایش می‌دهد.

```
console.log(str1.length);
```

فایلی به نام user-info.html ایجاد کنید و اسکریپت زیر را برای آشنایی با عملکرد **توابع رشته‌ای** بنویسید:

```
// اطلاعات وارد شده توسط کاربر در صفحه ثبت نام
let userName = "Fatemeh_kh";
let phone = "0939-256-5511";
let birthDate = "2002-09-11";
let email = "user123@gmail.com";

// نمایش نام کاربر با حروف بزرگ (برای نمایش در فاکتور یا داشبورد)
console.log("نام کاربر: ",userName.toUpperCase());

//استخراج نام کاربری از ایمیل (برای سلام و خوش آمدگویی در داشبورد)
let index = email.indexOf("@");
let userId = email.substring(0,index);
console.log("شناسه کاربری :",userId);

// جدا کردن اجزای تاریخ تولد برای نمایش با قالبی زیباتر
let birthParts = birthDate.split("-");
console.log("جزئیات تاریخ تولد: ",birthParts);

// 11/09/2022 تبدیل فرمت تاریخ برای نمایش به صورت
let birthForDisplay = birthDate.replace(/-/g,"-");
console.log("تاریخ تولد برای نمایش: ", birthForDisplay);

// بررسی شروع شماره موبایل (برای اعتبارسنجی)
let isValidPhone = phone.startsWith("09");
console.log("شماره موبایل معتبر است؟", isValidPhone);
```

```
نام کاربر: FATEMEH_KH
شناسه کاربری : user123
جزئیات تاریخ : ▼(3) ["2002", "09", "11"]
                0: "2002"
                1: "09"
                2: "11"
                length: 3
              ▶__proto__: Array(0)
تاریخ تولد برای نمایش: 2002/09/11
شماره موبایل معتبر است؟ true
```

بیشتر بدانید



به‌کارگیری توابع آرایه در جاوااسکریپت

توابع آرایه‌ای جاوااسکریپت، ابزارهای مناسبی برای مدیریت، جست‌وجو، فیلتر و تبدیل داده‌های ساختاریافته هستند. به کمک این متدها می‌توان اعمال مختلفی همچون مرتب‌سازی (sort)، فیلتر کردن(filter) و استخراج اطلاعات، لیست‌ها را به سادگی و بدون نیاز به نوشتن حلقه‌های پیچیده، انجام داد. برخی از توابع آرایه به شرح زیر است:

()push: برای اضافه کردن یک یا چند عنصر به انتهای آرایه استفاده می‌شود.

()slice: برای استخراج بخشی از آرایه و ایجاد یک آرایه جدید به‌کار می‌رود.

()filter: آرایه جدیدی را با فیلتر کردن عناصر آرایه اصلی براساس یک شرط خاص ایجاد می‌کند.

()sort: مرتب‌سازی عناصر آرایه را انجام می‌دهد.

ویژگی length تعداد عناصر موجود در آرایه را برمی‌گرداند. به‌طور مثال عبارت names.length عدد ۶ را برمی‌گرداند.

فایلی به نام customer-list.html ایجاد کنید و اسکریپت زیر را برای آشنایی با عملکرد توابع آرایه بنویسید. نتایج اسکریپت را در کلاس بررسی و تحلیل کنید.

```
let customers = ['Ali','Milad','Parisa','Nader'];

//اضافه شدن مشتریان جدید به فروشگاه
customers.push("Amir","Neda");
console.log("لیست مشتریان پس از ثبت نام جدید",customers);

//(استخراج بخشی از لیست (مانند مشتریان VIP یا آخرین مشتریان ثبت نام کرده
let recentCustomers = customers.slice(4,6);
console.log("آخرین مشتریان ثبت نام شده: ",recentCustomers);

//مرتب سازی اسامی برای پخش مدیریت فروشگاه
customers.sort();
console.log("لیست مشتریان مرتب شده:", customers);

// پیدا کردن مشتریانی که نامشان با حرف A شروع می شود
// برای ارسال پیامک تبریک یا تخفیف ویژه
let aCustomers = customers.filter(name=>name.startsWith("A"));
console.log("مشتریانی که با A شروع می شوند", aCustomers);
```

```
لیست مشتریان پس از ثبت نام جدید
▶(6) ["Ali", "Milad", "Parisa", "Nader", "Amir", "Neda"]
آخرین مشتریان ثبت نام شده: ▶(2) ["Amir", "Neda"]
لیست مشتریان مرتب شده:
▶(6) ["Ali", "Amir", "Milad", "Nader", "Neda", "Parisa"]
مشتریانی که با A شروع می شوند ▶(2) ["Ali", "Amir"]
```

بیشتر بدانید

به کارگیری توابع و دستورات ورودی و خروجی جاوااسکریپت

توابع ورودی و خروجی (I/O) در جاوااسکریپت، پل ارتباطی بین برنامه و دنیای خارج از آن هستند و امکان ایجاد یک تجربه تعاملی و پویا را برای کاربران فراهم می‌کنند. در محیط مرورگر، این توابع برای دریافت ورودی از کاربر، نمایش جعبه پیغام، نمایش داده در کنسول و دریافت تأیید کاربر استفاده می‌شوند.

برای بررسی عملکرد توابع ورودی و خروجی، یک فایل جدید با نام user-interaction.html ایجاد کنید و کد زیر را در بخش body بنویسید.

```
<button onclick="alert('محصول مورد نظر به سبد خرید اضافه شد')">افزودن به سبد خرید</button>

<script>
    const telNumber = prompt("لطفا شماره تلفن خود را وارد کنید");
    document.write("<br> Your telephone number is: " + telNumber);
    const conf = confirm("مطمئنید که قصد حذف این محصول را از سبد خرید دارید؟");
    console.log(conf);
</script>
```

فایل را ذخیره کنید و نتیجه را در مرورگر مشاهده کنید.

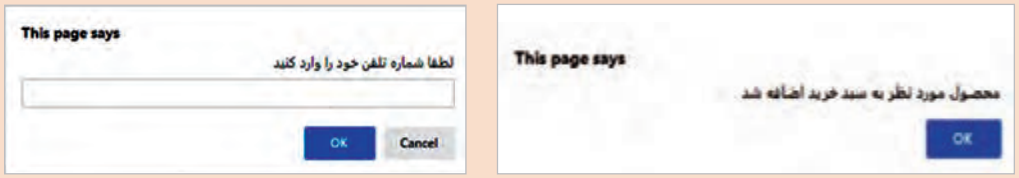


console.log(): نمایش مقادیر را در کنسول مرورگر به عهده دارد.

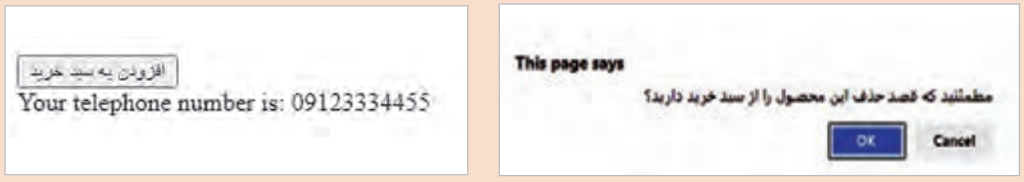
prompt(): نمایش یک کادر محاوره‌ای برای دریافت داده از کاربر. مقدار دریافتی به محل فراخوانی ارسال می‌شود.

alert(): نمایش پنجره هشدار با یک پیام و دکمه OK.

confirm(): نمایش یک پنجره تأیید شامل دکمه OK و Cancel. مقدار بازگشتی تابع True یا False است.



پنجره alert و prompt



پنجره confirm

document.write(): نوشتن مستقیم در صفحه وب (استفاده از این روش در برنامه‌نویسی مدرن توصیه نمی‌شود).

کنجکاوی

تحقیق کنید چرا استفاده از توابع ()prompt و ()confirm برای عملیات ورودی می‌تواند باعث مختل شدن تجربه کاربری شود؟

کلاس‌ها و اشیا

در زبان‌های برنامه‌نویسی، کلاس به معنای الگویی برای ایجاد اشیا است. یک کلاس حاوی مجموعه‌ای از ویژگی‌ها و متدهاست. شیئی که از روی یک کلاس تعریف می‌شود، دارای ویژگی‌ها و متدهای تعریف شده در آن کلاس خواهد بود.

بیشتر بدانید

در مثال زیر یک کلاس با عنوان Person ایجاد شده است که حاوی ویژگی‌های **name** و **age** است:

```
class Person {
constructor(name, age) {
    this.name = name;
    this.age = age;
}
greet() {
    console.log(`my name is ${this.name} and i am ${this.age} years old.`);
}
}
const person1 = new Person('amin', 47);
person1.greet();
```

```
top  Filter
my name is amin and i am 47 years old.
```

ـ متد **constructor** برای تعریف و مقداردهی ویژگی‌های یک شیء استفاده می‌شود. زمانی که یک شیء جدید از یک کلاس ایجاد می‌شود، متدconstructor به طور خودکار فراخوانی شده و مقادیر اولیه را به ویژگی‌های آن شیء اختصاص می‌دهد. در مثال فوق متد constructor دو پارامتر name و age را می‌پذیرد و آنها را به ویژگی‌های متناظر در شیء نسبت می‌دهد.

ـ کلاس person دارای یک متد بنام **greet** است که یک پیغام را در کنسول مرورگر نمایش می‌دهد.

ـ متغیر person1 یک شیء از نوع کلاس person است که مقدار ویژگی‌های آن برابر amin و ۴۷ است. این مثال نشان می‌دهد که برای ایجاد یک شیء از روی یک کلاس باید از کلمه کلیدی **new** استفاده شود. با توجه به اینکه کلاس person دارای متدی بنام greet است، شیء person1 نیز امکان استفاده از این متد را دارد. متد greet باعث نمایش عبارت "My name is amin and I am 47 years old" در کنسول خواهد شد.

ایجاد کلاس و شیء

فایل جدیدی با نام object.html ایجاد کنید و درون آن کلاس جدیدی با نام Car با ویژگی‌های auto_maker، model وyear ایجاد کنید.

یک متد با نام print_Details برای چاپ ویژگی‌های خودرو بنویسید. نحوه چاپ به‌صورت زیر باشد:

Auto maker: Toyota , Model: Camry , Year: 2020

ـ یک شیء با مقادیر Toyota، Camry، 2020 ایجاد کنید و متد print_details را برای شیء جدید مورد استفاده قرار دهید.

```
class Car {
// ویژگی‌های کلاس (Constructor)
constructor(auto_maker, model, year) {
    this.auto_maker = auto_maker;
    this.model = model;
    this.year = year;
}
// متد برای چاپ جزئیات خودرو
print_details() {
    console.log(`Auto maker: ${this.auto_maker}, Model: ${this.model}, Year: ${this.year}`);
}
}
// ایجاد یک شیء از کلاس Car
const myCar = new Car("Toyota", "Camry", 2020);
// فراخوانی متد برای چاپ اطلاعات
myCar.print_details();
```

```
top  Filter
Auto maker: Toyota, Model: Camry, Year: 2020
```

دسترسی به عناصر DOM

جاوااسکریپت برای انجام عملیات مختلف بر روی عناصر صفحه، به دسترسی به آن عناصر نیاز دارد. به همین دلیل جاوااسکریپت دارای متدهایی برای دسترسی به عناصر است. به‌عنوان مثال برای دسترسی به عنصری با شناسه demo از متد getElementById به‌شکل زیر استفاده می‌شود:

در این اسکریپت مقدار جدید عنصر para توسط ویژگی textContent مشخص شده است.

برای انتخاب و دسترسی به هر یک از عناصر صفحه روش‌های مختلفی وجود دارد که در این بخش به آنها اشاره می‌شود.

انتخاب براساس ID: عناصر صفحه براساس شناسه (id) آنها انتخاب می‌شوند.

انتخاب براساس کلاس: مجموعه‌ای از عناصر با نام کلاس مشخص انتخاب می‌شوند.

انتخاب براساس نام تگ: مجموعه‌ای از عناصر با نام تگ یکسان انتخاب می‌شوند.

```
<p id="demo"> This is a paragraph </p>

<script>
    const para = document.getElementById("demo");
    para.textContent = "Paragraph changed";
</script>
```

شکل ۱۲

```
document.getElementById("elmntId");

document.getElementsByClassName("products");

document.getElementsByTagName("p");
```

شکل ۱۳

کنجکاوی

برای مطالعه درباره سایر متدهای دسترسی به عناصر، عبارت «how to get elements by JavaScript» را در گوگل جست‌وجو کنید و از لیست نتایج سایت w3schools.com را باز کنید.

در ادامه تمرین‌هایی برای کار روی عناصر مختلف صفحه توسط جاوااسکریپت ارائه می‌گردد.

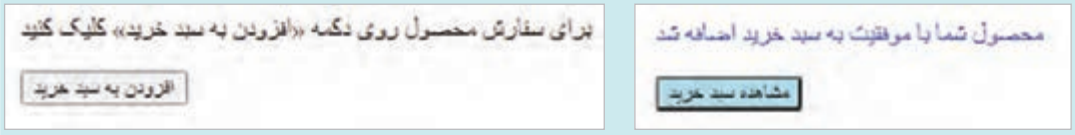
کار کارگاهی

مدیریت عناصر صفحه توسط جاوااسکریپت

دسترسی به محتوای عناصر، تغییر محتوا و سبک عناصر: اسکریپتی بنویسید که محتوا و سبک یک دکمه و عنصر متنی را توسط جاوااسکریپت تغییر دهد. این تغییرات باید در هنگام کلیک روی دکمه رخ دهد.

ـ یک فایل به نام product.html ایجاد کنید و کد زیر را در بخش <body> وارد کنید:

```
1  <p id="para">برای سفارش محصول روی دکمه «افزودن به سبد خرید» کلیک کنید</p>
2
3  <button id="addToCart" onclick="change()"> افزودن به سبد خرید  </button>
4
5  <script>
6  function change(){
7      const message = document.getElementById("para");
8      const btn = document.getElementById("addToCart");
9      message.innerHTML = "محصول شما با موفقیت به سبد خرید اضافه شد";
10     btn.textContent = "مشاهده سبد خرید";
11     message.style.color = "blue";
12     btn.style.background = "lightblue";
13 }
```



در خطوط ۶ تا ۱۳ اسکریپت فوق یک تابع با نام change() تعریف شده است که دسترسی به عناصر addToCart و para و سپس تغییر محتوا و سبک آنها را انجام می‌دهد. تابع change() هنگام کلیک روی دکمه فراخوانی می‌شود.

در خطوط ۷ و ۸ دسترسی به عناصر پاراگراف و دکمه انجام شده است. در خطوط ۹ و ۱۰ تغییر محتوای عناصر پاراگراف و دکمه، توسط ویژگی‌های innerHTML و textContent صورت گرفته است. در خطوط ۱۱ و ۱۲ تغییر سبک عناصر توسط ویژگی style انجام شده است. فایل را ذخیره کنید و نتیجه را در مرورگر بررسی کنید.

بیشتر بدانید

دسترسی به عناصر و تغییر ویژگی آنها: اسکریپتی بنویسید که ویژگی src یک تصویر را هنگام قرار گرفتن ماوس روی آن تغییر دهد.

ـ در فایل products.html تصویر یک محصول را به قسمت <body> اضافه کنید. سپس اسکریپت زیر را برای تغییر مقدار ویژگی src تصویر استفاده کنید.

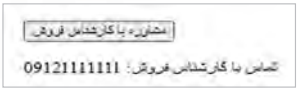
```
<script>
    function changeImage(){
        const image = document.getElementById("flower-img");
        image.setAttribute("src","./images/Basic-Technical-Knowledge-hover.jpg");
    }
    function resetImage(){
        const image = document.getElementById("flower-img");
        image.setAttribute("src","./images/Basic-Technical-Knowledge.jpg");
    }
</script>

<img id="flower-img" src="./images/Basic-Technical-Knowledge.jpg"
                onmouseover="changeImage()" onmouseleave="resetImage()">
```

فایل را ذخیره کنید و نتیجه را در مرورگر بررسی کنید.

با انتقال اشاره‌گر ماوس روی تصویر، رویداد onmouseover فعال شده و در نتیجه تابع changeImage() برای تغییر تصویر فراخوانی می‌شود. با کنار رفتن ماوس از روی تصویر، رویداد onmouseleave فعال شده و بنابراین تابع resetImage() فراخوانی می‌شود. این تابع تصویر اولیه را برمی‌گرداند.

```
<button id="consultButton" onclick="openPanel()">مشاوره با کارشناس فروش</button>
<script>
function openPanel(){
    const consultPanel = document.createElement("div");
    consultPanel.innerHTML = `<p> 09121111111 :تماس با کارشناس فروش</p>`;
    consultPanel.className = "consult-panel";
    document.body.appendChild(consultPanel);
    consultPanel.style.display = "block";     // نمایش پنل مشاوره
    setTimeout(function() {
        consultPanel.remove();
    }, 5000);
}
```



در هر دو تابع فوق از متد setAttribute() برای تغییر تصویر استفاده شده است.

```
#consultButton {
    position: fixed;
    bottom: 100px;
    right: 20px;
    padding: 10px 15px;
    cursor: pointer;
    border-radius: 5px;
    }
.consult-panel {
    position: fixed;
    bottom: 80px;
    right: 10px;
    border: 1px solid #ccc;
    background-color: #fff;
    padding: 15px;
    z-index: 1000;
    /* محل قرارگیری:روی همه عناصر */
    }
```

اضافه و حذف کردن عناصر توسط جاوااسکریپت:

اسکریپتی بنویسید که با کلیک روی یک دکمه با عنوان مشاوره، پانلی را برای راهنمایی کاربر به صفحه اضافه کند و بعد از چند ثانیه آن را ببندید.

ـ یک فایل جدید با نام open-panel.html ایجاد کنید و اسکریپت را طبق شکل روبه‌رو درون آن بنویسید.

استایل عناصر را به بخش <head> اضافه کنید. سپس فایل را ذخیره کنید و نتیجه را در مرورگر مشاهده کنید.

با کلیک روی دکمه «مشاوره با کارشناس فروش» تابع openPanel() فراخوانی می‌شود. این تابع در خط ۳ تا ۱۲ نوشته شده است.

ـ در خط ۴ یک عنصر div جدید توسط متد createElement ایجاد می‌شود.

ـ در خط ۵ محتوای عنصر توسط ویژگی InnerHTML تعیین می‌گردد.

ـ عنصر جدید نیازمند تعیین استایل است. این استایل توسط کلاس consultpanel مشخص می‌شود.

ـ در خط ۶ نام کلاس عنصر جدید تعیین می‌شود.

ـ در خط ۷ عنصر جدید توسط متد appendChild به صفحه اضافه می‌گردد.

ـ پانل مشاوره (عنصر div) به‌صورت پیش‌فرض مخفی است. در خط ۸ وضعیت نمایش پانل به block تغییر می‌کند.

ـ در خط ۹ متد setTimeout برای تعیین زمان حذف پانل مشاوره استفاده می‌شود.

ـ در خط ۹ پانل مشاوره بعد از ۵ ثانیه توسط متد remove حذف می‌شود.

دسترسی به اطلاعات فرم و اعتبارسنجی داده‌های کاربر: یک اسکریپت ساده برای اعتبارسنجی داده‌های ورودی کاربر بنویسید. در این اعتبارسنجی پر بودن فیلدهای متنی را بررسی کنید، به‌طوری‌که اگر یک فیلد متنی خالی باشد، پیغام هشداری را نمایش می‌دهد.

ـ یک فایل به‌نام sign-in.html ایجاد کنید و کد شکل زیر را درون آن وارد کنید:

```
<form id="loginForm" onsubmit="return validateForm()">
    <label for="username">نام کاربری</label><br>
    <input type="text" id="username" name="username"><br>
    <label for="password">کلمه عبور</label><br>
    <input type="password" id="password" name="password"><br>
    <label for="email">ایمیل</label><br>
    <input type="email" name="email" id="email"><br>
    <label for="remember">مرا به خاطر بسپار</label>
    <input type="checkbox" name="remember" id="remember"><br>
    <input type="submit" value="ورود" name="submit"><br>
</form>
```



این فرم حاوی سه فیلد برای دریافت نام کاربری، کلمه عبور و ایمیل است.

برای اعتبارسنجی فرم از اسکریپت شکل زیر استفاده کنید. این اسکریپت حاوی تابع validateForm() است که مقدار ورودی فیلدهای متنی را بررسی می‌کند. این تابع در رویداد onsubmit فرم صفحه قبل فراخوانی می‌شود.

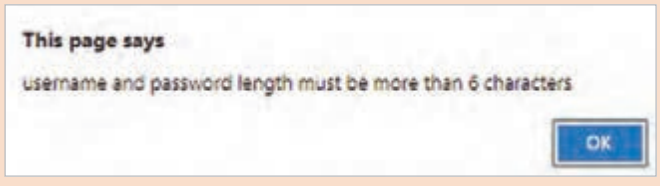
```
1  function validateForm() {
2  const username = document.getElementById('username').value;
3  const password = document.getElementById('password').value;
4  if (username === "") {
5      alert("Please enter your username");
6      return false;
7  }
8  if (password === "") {
9      alert("Please enter your password");
10     return false;
11 }
12 return true;
13 }
```



تابع validateForm() نام کاربری و کلمه عبور را در خط ۲ و ۳ دریافت می‌کند و مقدار آنها را درون دو متغیر username و password قرار می‌دهد. در خط ۴ تا ۷ خالی بودن username بررسی می‌شود، در صورت خالی بودن این متغیر، پیغام هشدار ظاهر می‌شود و دستور return false از ارسال محتوای فرم به سرور جلوگیری می‌کند. به‌طور مشابه در خط ۸ تا ۱۱ خالی بودن متغیر password بررسی می‌گردد. در صورتی‌که شرط ifهای خط ۴ و ۸ برقرار نباشد، دستور خط ۱۲ اجرا می‌شود و اطلاعات فرم به سرور ارسال می‌شود.

در ادامه طول نام کاربری و کلمه عبور را بررسی کرده؛ به‌طوری‌که اگر طول هر یک از این دو فیلد کمتر از ۶ کاراکتر باشد، پیغام هشدار نمایش داده شود. برای این‌کار شرط زیر را بعد از خط ۱۱ بنویسید.

```
if (username.length<6 || password.length<6){
    alert("username and password length must be more than 6 characters");
    return false;
}
```





نکته

DOM پایه تمام وب‌سایت‌های تعاملی است؛ تسلط بر آن یعنی قدم گذاشتن به دنیای حرفه‌ای طراحی وب.

ارزشیابی شایستگی طراحی صفحات وب تعاملی با جاوااسکریپت

استاندارد عملکرد کار: هنرجو باید بتواند با استفاده از JavaScript به‌صورت بهینه و امن، رفتارهای پویا و تعاملی را به صفحات وب اضافه کند و خروجی نهایی را در مرورگرهای مختلف بدون خطا اجرا نماید.

ردیف	مراحل کار	شرایط انجام کار (ابزار،مواد، تجهیزات، زمان، مکان و ...)	ابزارهای ارزشیابی	استاندارد (شاخص‌ها/داوری/نمره‌دهی)	نتایج ممکن	نمره کسب شده
۱	آشنایی، راه‌اندازی و شروع کدنویسی	رایانه، IDE، مرورگر، ۳۰دقیقه، کارگاه	مشاهده عملی - پرسش شفاهی	ـ تعریف صحیح جاوااسکریپت و کاربرد آن در وب ـ تشخیص تفاوت JavaScript با HTML و CSS ـ توضیح روش‌های افزودن (JS inline داخلی، خارجی) ـ ایجاد فایل js و اتصال صحیح آن به HTML ـ استفاده از console.log بدون خطا ـ اجرای اسکریپت در مرورگر ـ انجام کنجکاوی‌ها ـ انجام مراحل با روش‌های دیگر	۳	
				ـ ایجاد فایل js و اتصال صحیح آن به HTML ـ استفاده از console.log بدون خطا ـ اجرای اسکریپت در مرورگر	۲	
				ـ فقط تعریف جاوااسکریپت و کاربرد آن در وب ـ تشخیص تفاوت JavaScript با HTML و CSS ـ توضیح روش‌های افزودن (inline.Js داخلی، خارجی)	۱	
۲	تعریف و استفاده از متغیرها	رایانه، IDE، مرورگر، ۱۰ دقیقه	مشاهده عملی - پرسش شفاهی	ـ تعریف متغیر با let ـ تشخیص انواع داده ـ توضیح تفاوت == و === ـ شناخت عملگرهای منطقی و ریاضی ـ تبدیل نوع داده (Number, String) انجام محاسبات و مقایسه‌ها ـ انجام کنجکاوی‌ها ـ انجام مراحل با روش‌های دیگر	۳	
				ـ تعریف متغیر با let و const ـ تشخیص انواع داده ـ توضیح تفاوت == و === ـ شناخت عملگرهای منطقی و ریاضی ـ تبدیل نوع داده (Number, String) ـ انجام محاسبات و مقایسه‌ها	۲	
				ـ تعریف متغیر با let ـ تشخیص انواع داده ـ انجام ناقص مراحل	۱	
۳	آرایه‌ها و اشیا	رایانه، IDE، مرورگر، ۱۵ دقیقه	مشاهده عملی - پرسش شفاهی	تعریف آرایه و شیء ـ تشخیص تفاوت آرایه و شیء ـ شناخت نحوه دسترسی به عناصر ـ ایجاد آرایه یک‌بعدی و دوبعدی ـ ایجاد شیء با چند ویژگی ـ دسترسی و تغییر مقدار عناصر ـ ساخت آرایه محصولات و نمایش اطلاعات آن ـ ایجاد شیء دانش‌آموز با چند ویژگی و نمایش آن ـ پیمایش آرایه با حلقه ـ انجام کنجکاوی‌ها ـ انجام مراحل با روش‌های دیگر	۳	
				ـ ایجاد آرایه یک‌بعدی ـ ایجاد شیء با چند ویژگی ـ دسترسی و تغییر مقدار عناصر ـ ساخت آرایه محصولات و نمایش اطلاعات آن ـ شیء دانش‌آموز با چند ویژگی و نمایش آن ـ پیمایش با حلقه	۲	
				تعریف آرایه و شیء ـ تشخیص تفاوت آرایه و شیء ـ شناخت نحوه دسترسی به عناصر ـ ایجاد آرایه یک‌بعدی ـ انجام ناقص مراحل	۱	
۴	ساختارهای شرطی و حلقه‌ها	رایانه، IDE، مرورگر، ۱۵دقیقه	مشاهده عملی - پرسش شفاهی	توضیح کاربرد if ـ توضیح تفاوت حلقه for و while ـ شناخت break و continue ـ استفاده صحیح از if/else ـ استفاده از حلقه for، while ـ پیمایش آرایه با for...of ـ انجام کنجکاوی‌ها نجام مراحل با روش‌های دیگر	۳	
				توضیح کاربرد if ـ توضیح کاربرد for و while ـ شناخت break و continue ـ به‌کارگیری if، for، while ـ پیمایش آرایه با for...of	۲	
				انجام ناقص مراحل	۱	

۵	توابع و توابع آماده	رایانه، IDE، مرورگر، ۱۵ دقیقه	مشاهده عملی، پرسش شفاهی	ـ تعریف تابع ـ تفاوت پارامتر و آرگومان ـ شناخت توابع آماده Math و String و Array ـ ایجاد تابع با پارامتر فراخوانی تابع ـ استفاده از متدهایـ Math.random()ـ toUpperCase() ـ push()، pop ـ نوشتن تابع محاسبه میانگین ـ استفاده از تابع برای اعتبارسنجی ورودی ـ استفاده از متدهای آرایه در یک مثال کاربردیـ انجام کنجکاوی هایـ انجام مراحل با روش‌های دیگر	۳	
				تعریف تابع ـ به‌کارگیری توابع ریاضی و رشته‌ای	۲	
				انجام ناقص مراحل و یا دانش در حد تعریف	۱	
۶	کلاس‌ها و DOM تعامل با صفحه وب	رایانه، IDE، مرورگر، ۱۵ دقیقه	مشاهده عملی، پرسش شفاهی	ـ تعریف مفهوم DOM ـ توضیح ساختار document ـ تعریف کلاس و شیءـ ایجاد کلاس و نمونه‌سازی ـ انتخاب عناصر با: getElementById quer-ySelector ـ تغییر محتوا و استایل عناصرـ ایجاد دکمه‌ای که با کلیک متن صفحه تغییر کندـ ایجاد فرم ساده و دریافت مقدار ورودی ـ مدیریت رویداد (event handling) بدون خطا ـ انجام کنجکاوی‌ها ـ انجام مراحل با روش‌های دیگر	۳	
				ـ تعریف مفهوم DOM ـ توضیح ساختار document ـ تعریف کلاس و شیءـ ایجاد کلاس و نمونه‌سازی ـ انتخاب عناصر با: getElementById quer-ySelector	۲	
				تعریف مفهوم DOM ـ توضیح ساختار document ـ تعریف کلاس و شیءـ انجام ناقص مراحل	۱	
شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش:				احراز همه شایستگی‌های غیرفنی	۲	
				عدم احراز هر یک از شایستگی‌های غیرفنی	۱	
ارزشیابی کار (شایستگی انجام کار)					بلی ☐	
					خیر ☐	

توضیحات:

۱ـ معیار شایستگی انجام کار:

ـ کسب حداقل نمره ۲ از مراحل ۱، ۵ و ۶ (مراحل بحرانی)

ـ کسب حداقل نمره ۲ از بخش شایستگی‌های غیرفنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش

ـ کسب حداقل میانگین ۲ از مراحل کار

۲ـ تعریف سطوح شایستگی: صرف‌نظر از اینکه یک تکلیف کاری در چه سطحی از صلاحیت حرفه‌ای انجام می‌شود، در هر محیط کاری ممکن است انجام هر کار با کیفیت مشخصی مورد انتظار باشد. سطح شایستگی انجام کار، معیار اساسی ارزشیابی است.

مهارت (سطح سه): ماهر و قادر به آموزش و هدایت دیگران، توانایی برنامه‌ریزی و تحلیل، پاسخ‌گویی در برابر کارهای خود، سروکار داشتن با سطح وسیعی از کارها و فعالیت‌ها

تسلط (سطح چهار): خبرگی در انجام کار و آموزش دیگران، ایجاد نوآوری، سازگاری، عیب‌یابی، هدایت و راهنمایی دیگران.

پودمان چهارم

توسعه وب پایتون



در پودمان «توسعه وب پایتون»، شما با مفاهیم کلیدی برنامه‌نویسی پیشرفته در پایتون آشنا می‌شوید تا بتوانید از این زبان قدرتمند در طراحی و توسعه پروژه‌های واقعی وب استفاده کنید.

این پودمان شامل دو واحد اصلی است:

شیءگرایی در پایتون: هنرجو در این بخش ساختار کدنویسی حرفه‌ای را با استفاده از کلاس‌ها، اشیا، وراثت و چندریختی می‌آموزد تا برنامه‌های مقیاس‌پذیر و قابل نگهداری بنویسد.

توسعه شیءگرایی: تمرکز این بخش بر درک عمیق‌تر اصول شی‌گرایی و یادگیری نحوه کار با فایل‌ها برای ذخیره، خواندن و پردازش داده‌ها در محیط وب است.

واحد یادگیری ۱
تولید برنامه شیءگرا

آیا تابه‌حال اندیشیده‌اید

- در دنیای واقعی، اشیای مشابه (مانند خودروها) با وجود تفاوت در مدل و رنگ، ویژگی‌ها و رفتارهای مشترکی دارند. این موضوع به کدام مفهوم در برنامه‌نویسی مرتبط است؟
- چرا با افزایش حجم برنامه‌ها، نگهداری و اصلاح آنها دشوارتر می‌شود؟ چگونه می‌توان ساختار برنامه را به‌گونه‌ای طراحی کرد که تغییرات، کمترین تأثیر را بر سایر بخش‌ها داشته باشند؟
- اگر بخواهید در یک بازی رایانه‌ای، چند شخصیت با رفتارهای مشابه و ویژگی‌های متفاوت طراحی کنید، استفاده از چه روشی در برنامه‌نویسی مناسب‌تر است؟
- به چه دلیل در طراحی بسیاری از نرم‌افزارها و برنامه‌های بزرگ، از شیوه برنامه‌نویسی شیءگرا استفاده می‌شود؟
- «کلاس» در برنامه‌نویسی به چه چیزی شباهت بیشتری دارد: نقشه یک ساختمان یا خود ساختمان؟

در پایان از هنرجو انتظار می‌رود

۱ کلاس‌های پایتون را با ویژگی‌ها و متدهای مناسب، همراه با تعیین مقادیر پیش‌فرض و مدیریت دسترسی (public, protected, private) طراحی و پیاده‌سازی کند.

۲ از کلاس‌ها نمونه (Instance) بسازد و مقادیر و ویژگی‌های این نمونه‌ها را در طول اجرای برنامه به طور مؤثر تغییر دهد و مدیریت کند.

۳ با استفاده از ارث‌بری برای ایجاد سلسله مراتب کلاس‌ها، فراخوانی متدهای والد و بازنویسی آنها در کلاس‌های فرزند مطابق با نیازهای برنامه اقدام نماید.

۴ با استفاده از اصول OOP، کدی ماژولار، قابل فهم و با قابلیت استفاده مجدد (reusable) بنویسد که از تکرار کد جلوگیری کند.

۵ درک اولیه‌ای از نحوه عملکرد متاکلاس‌ها و نقش آنها در تعریف و ساختار کلاس‌ها پیدا کند.

۶ مسائل برنامه‌نویسی ساده تا متوسط را با استفاده از برنامه‌نویسی شیءگرا تحلیل کرده و راه‌حل‌های مبتنی بر شیء طراحی و پیاده‌سازی کند.

استاندارد عملکرد

با استفاده از مفاهیم برنامه‌نویسی شیءگرا در زبان پایتون، مسائل ساده و کاربردی را مدل‌سازی کند، کلاس مناسب طراحی نماید و با ایجاد و به‌کارگیری نمونه‌ها، برنامه‌ای خوانا، قابل اجرا و قابل توسعه تولید کند.

برنامه‌نویسی شیءگرا (Object–Oriented Programming)



شکل ۱

برنامه‌نویسی شیءگرا روشی از برنامه‌نویسی است که در آن، اجزای برنامه به‌صورت موجودیت‌هایی شبیه به دنیای واقعی مدل‌سازی می‌شوند. در این روش، به‌جای آن‌که فقط به دستورات و توابع فکر کنید، ابتدا به چیزهایی فکرمی‌کنید که در دنیای واقعی وجود دارند، ویژگی دارند و می‌توانند کاری انجام دهند.

به این چیزها در برنامه‌نویسی، شیء (Object) و به این شیوهٔ برنامه‌نویسی، شیءگرایی یا به اختصار (O.O.P.) گفته می‌شود.

در این رویکرد، هدف اصلی، بازآفرینی منطق و ساختار دنیای واقعی در محیط برنامه‌نویسی است. برای درک بهتر این مفهوم، می‌توان به بازی‌های استراتژیک رایج همچون Age of Empires یا Clash of Clans اشاره کرد. در این بازی‌ها، شخصیت‌ها و موجودیت‌ها به گروه‌های گوناگون تقسیم می‌شوند؛ از کارگران و سربازان گرفته تا وسایل نقلیه و ماشین‌آلات جنگی. برنامه‌نویسی شیءگرا با معرفی مفهومی به نام کلاس (Class)، به شما کمک می‌کند موجودیت‌های مشابه را در قالب یک گروه قرار دهید. به این ترتیب، می‌توانید هر گروه را به‌صورت یک کلاس در نظر بگیرید و ویژگی‌ها و کارهای مربوط به آن را در برنامه مشخص کنید.



شکل ۲

فعلاً از دنیای بازی‌ها فاصله گرفته و یک مثال کاملاً واقعی بررسی می‌شود. یک تلفن همراه قدیمی را در نظر بگیرید. این تلفن همراه صرفاً یک وسیلهٔ ساده نیست، بلکه دارای مجموعه‌ای از ویژگی‌هاست و قادر است وظایف و عملیات مختلفی را انجام دهد.



شکل ۳

برای مثال هر تلفن همراه این ویژگی‌ها را دارد:

- دارای رنگ است.
- دارای برند است.
- دارای مدل است.
- میزان شارژ باتری آن در هر لحظه مشخص است.

نکته

فعلاً با مثال‌هایی از گوشی‌های قدیمی شروع کرده و در ادامه درس به گوشی‌های هوشمند و گوشی‌های مخصوص بازی هم پرداخته خواهد شد.

فعالیت

بر اساس اطلاعات گوشی خودتان یا یک گوشی دلخواه، اطلاعات جدول زیر را تکمیل نمایید. (جدول ویژگی‌ها)

در جدول زیر، نمونه‌هایی از تلفن همراه و ویژگی‌های آنها را مشاهده می‌کنید:

نمونه‌ها / ویژگی‌ها	color	brand	model	battery
گوشی من	gray	nokia	۱۱۰۰	٪۳۲
گوشی شما				
گوشی دوستتان				

اگر دو گوشی هم‌مدل داشته باشید، ویژگی‌های آنها مشابه است، اما میزان باتری یا رنگ می‌تواند متفاوت باشد. هرکدام از این گوشی‌ها یک شیء جداگانه محسوب می‌شوند. به مشخصاتی که یک شیء دارد، ویژگی (Attribute) گفته می‌شود. این ویژگی‌ها مشخص می‌کنند که هر شیء چه چیزهایی دارد.

همچنین هر تلفن همراه کارهای مختلفی می‌تواند انجام دهد:

- روشن و خاموش شود.
- تماس بگیرد.
- پیام ارسال کند.

فعالیت

به‌جز کارهای فوق، با گوشی شخصی خودتان چه کارهای دیگری می‌توانید انجام دهید. جدول زیر را کامل کنید.

۱-	۳-
۲-	۴-

متد

کارهایی که یک شیء می‌تواند انجام دهد، متد (Method) نامیده می‌شود. در حقیقت متدها مانند دستورالعمل‌هایی هستند که به شیء می‌گویند چه کاری انجام دهد.

در دنیای واقعی، این تلفن همراه یک «شیء» می‌باشد و در برنامه‌نویسی شیءگرا نیز دقیقاً با همین نگاه، اشیا مدل‌سازی می‌شوند.

شیء

شیء (Object) به هر چیزی گفته می‌شود که:

- دارای ویژگی (Attribute) باشد.
- بتواند کار یا کارهایی (Method) انجام دهد.

نمونه (Instance)

در پایتون، شیءهایی که با استفاده از یک کلاس ساخته می‌شوند، نمونه (Instance) آن کلاس نامیده می‌شوند. بنابراین یک تلفن همراه مشخص (مثلاً موبایل شخصی شما)، یک نمونه (Instance) از کلاس تلفن همراه محسوب می‌شود.

نکته

در این کتاب از این به بعد به جای استفاده از عبارت شیء (Object) از اصطلاح نمونه (Instance) استفاده می‌شود.

تعریف کلاس (Class)

حال اگر بخواهید چندین تلفن همراه مختلف بسازید، نوشتن ویژگی‌ها و متدها برای هر کدام به‌صورت جداگانه کار درستی نیست.

برای این کار، به یک قالب کلی نیاز داریم که مشخص کند:

۱ هر تلفن همراه چه ویژگی‌هایی دارد.

۲ چه کارهایی می‌تواند انجام دهد.

به این قالب کلی، کلاس (Class) گفته می‌شود.

کلاس مانند نقشهٔ ساخت یک تلفن همراه است و هر تلفن همراهی که ساخته می‌شود، در واقع یک نمونه (Instance) از آن کلاس خواهد بود.

جدول ۱

اصطلاحات	معنی
کلاس (Class)	قالب کلی یا نقشه ساخت
نمونه (Instance)	موجود ساخته‌شده از روی یک کلاس
ویژگی (Attribute)	چه چیزی دارد؟
متد (Method)	چه کاری انجام می‌دهد؟

حالا نوبت آن است که در محیط برنامه‌نویسی، اولین کلاس (class) خود را ایجاد کنید و سپس با استفاده از آن کلاس، یک نمونه(Instance) بسازید. کد شکل ۴ را درون یک فایل جدید به دقت تایپ کنید.

```
1 class Mobile:
2     def __init__(self, brand, color):
3         self.brand = brand
4         self.color = color
5
6
7 my_mobile = Mobile("nokia", "gray")
8 print(my_mobile.brand)
9
```



شکل ۴ـ اولین کد برای تعریف کلاس و ساخت نمونه

خط ۱: یک کلاس به نام Mobile تعریف می‌شود.

- کلاس در واقع یک الگو یا نقشه برای ساختن موبایل‌هاست.
- دقت کنید کلاس، خود موبایل نیست بلکه الگویی برای ساخت موبایل‌هاست.
- بر اساس استانداردهای نام‌گذاری در پایتون (PEP8)، نام کلاس‌ها باید با استفاده از شیوهٔ PascalCase نوشته شود؛ به این معنا که حرف اول نام کلاس و حرف اول تمام کلمات تشکیل‌دهندهٔ آن بزرگ باشد.

خط ۲: این متد، نقش متدِ سازندهٔ کلاس را در پایتون بر عهده دارد. به این متد اصطلاحاً initializer هم می‌گویند.

- هر زمان که از کلاس Mobile یک نمونه ساخته شود، یعنی با اجرای خط ۷، این متد به‌طور خودکار اجرا می‌شود.
- وظیفه اصلی این متد مقداردهی اولیه به ویژگی‌های نمونه (Instance) است.
- دقت کنید که __init__ نامی قراردادی در پایتون است و نمی‌توان آن را تغییر داد. در سمت چپ و راست کلمه init دو بار کاراکتر زیرخط (underscore) تایپ شده است.

خطوط ۳ و ۴: ویژگی‌های نمونه را مشخص می‌کنند.

- هر موبایل یک برند و یک رنگ دارد.
- در کد ارائه شده، self.brand و self.color هر دو به عنوان ویژگی (attribute) شناخته می‌شوند.
- کلمهٔ self به همان نمونه‌ای اشاره می‌کند که در حال ساخته شدن است و brand و color نام ویژگی‌هایی هستند که به آن نمونه تعلق دارند.

خطوط ۵ و ۶: خطوط خالی

- بر اساس استانداردهای کدنویسی پایتون (PEP8)، لازم است پس از پایان تعریف کلاس، دو خطِ خالی قرار داده شود تا مرز بین کلاس و بخش کدهای سراسری (Global) به‌وضوح مشخص شود.

خط ۷: ساختن نمونه(Instance)

- در این خط، یک نمونه (instance) به نام my_mobile از روی کلاس Mobile ساخته‌اید.
- مقدار "nokia" برای ویژگیِ brand و مقدار "gray" برای ویژگیِ color در نظر گرفته شده‌اند.



شکل ۵

خط ۸: دسترسی به ویژگی

- با استفاده از نام نمونه و علامت نقطه (.)، به ویژگی‌های آن دسترسی پیدا می‌کنید.
- این دستور مقدار ویژگی brand را برای نمونه my_mobile چاپ می‌کند.

خط ۹: خط خالی

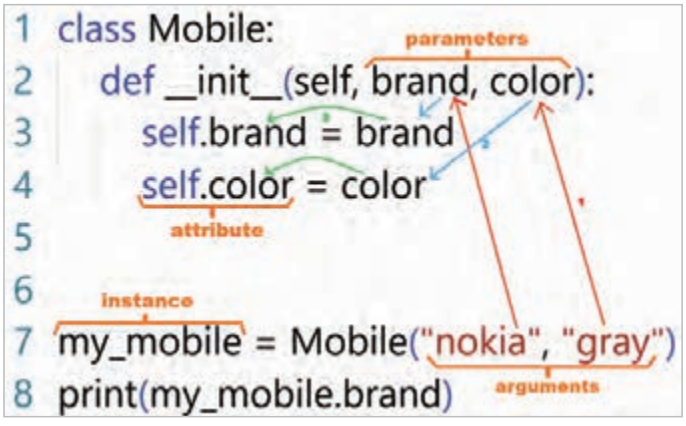
- طبق استانداردهای PEP8، پس از پایان کدها نیز یک خط خالی قرار داده می‌شود تا انتهای فایل از کدها به‌صورت خوانا و استاندارد مشخص شود.

توجه داشته باشید که اگر در این مرحله همهٔ جزئیات را به‌طور کامل متوجه نشدید، نگران نباشید؛ این یک مبحث جدید است و با ادامهٔ آموزش و فعالیت‌های بیشتر، مرور کدها و ساخت برنامه‌های بیشتر، به مرور، مفاهیم برایتان روشن می‌شود و جا می‌افتد.

حالا برنامه را اجرا کنید، می‌بینید که مقدار ویژگی brand از موبایلی که ساخته‌اید یعنی nokia چاپ می‌شود.

توجه داشته باشید در این مرحله، کلاس فقط شامل ویژگی‌ها (Attributes) است و فعلاً هیچ رفتار یا متدی (Method) به‌جز سازنده (__init__) برای نمونه‌ها تعریف نشده است.

حالا با یک تصویر روند اجرای برنامه را شرح داده شده است. شکل ۶ را به دقت مشاهده کنید.



شکل ۶ــ روند اجرای برنامه

همان‌طور که در شکل مشاهده می‌شود بعد از اجرای خط ۷ که برای ساختن نمونه my_mobile است مراحل زیر اتفاق می‌افتد:

۱ آرگومان‌های "nokia" و "gray" به متد __init__ فرستاده و درون پارامترهای brand و color قرار می‌گیرند.

۲ مقدار پارامتر brand درون self.brand و مقدار پارامتر color درون self.color ریخته می‌شوند.

۳ در این مرحله نمونه مورد نظر ساخته خواهد شد.

فعالیت

در انتهای برنامه دستوری اضافه کنید که رنگ موبایل را چاپ نماید.

فعالیت

داخل متد __init__ دستوری اضافه کنید که مدل گوشی شما را نیز به عنوان ویژگی در نظر بگیرد. فراموش نکنید که در خط ۷ نیز باید مدل گوشی خودتان را اضافه کنید. سعی کنید خودتان این فعالیت را انجام دهید ولی اگر هنوز ابهام داشتید در ادامه پاسخ این کد را می‌بینید و می‌توانید با کد خودتان مقایسه کنید.

در ادامه بر اساس کلاسی که تعریف کرده‌اید یک نمونه دیگر به عنوان موبایلی برای دوستتان بسازید. کد شکل ۷ را به دقت بنویسید.

```
1  class Mobile:
2      def __init__(self, brand, model, color):
3          self.brand = brand
4          self.model = model
5          self.color = color
6
7
8  my_mobile = Mobile("nokia", "1100", "gray")
9  friend_mobile = Mobile("samsung", "n100", "silver")
10
11 # Accessing my_mobile Attributes
12 print("brand:", my_mobile.brand)
13 print("model:", my_mobile.model)
14 print("color:", my_mobile.color)
15
16 # Accessing friend_mobile Attributes
17 print("brand:", friend_mobile.brand)
18 print("model:", friend_mobile.model)
19 print("color:", friend_mobile.color)
20
```

```
brand: nokia
model: 1100
color: gray
```

```
brand: samsung
model: n100
color: silver
```

شکل ۷ـ ساختن یک نمونه به عنوان گوشی برای دوستتان

خط ۴: یک ویژگی جدید برای مدل گوشی همراه تعریف و مقداردهی شده است.
خط ۹: یک نمونه جدید با نام friend_mobile با ویژگی‌هایی متفاوت از my_mobile ایجاد شده است.



شکل ۸

خطوط ۱۵ تا ۱۷: در این خطوط ویژگی‌های نمونه دوم چاپ شده است.

نکته

دقت کنید my_mobile و friend_mobile هر دو نمونه‌هایی مجزا هستند که از روی کلاس Mobile ساخته شده‌اند.

در ادامه یک متد به کلاس Mobile اضافه کنید تا مقدار تمام ویژگی‌های نمونه‌ای که ساخته‌اید را چاپ کند. کد قبلی را مانند شکل ۹ تغییر دهید.

```
1  class Mobile:
2      def __init__(self, brand, model, color): ...   دقت کنید این متد جمع شده است
6
7      def show_info(self):
8          print("brand:", self.brand)
9          print("model:", self.model)
10         print("color:", self.color)
11
12
13 my_mobile = Mobile("nokia", "1100", "gray")
14 my_mobile.show_info()
15
```

```
brand: nokia
model: 1100
color: gray
```

شکل ۹ـ افزودن متد show_info

خط ۷: این خط، تعریف یک متد (method) به نام show_info را در داخل کلاس Mobile انجام می‌دهد. متدها، توابعی هستند که به یک کلاس مرتبط هستند و می‌توانند به ویژگی‌های (Attributes) آن کلاس دسترسی داشته باشند.

- پارامتر self در تعریف متد، به خودِ نمونه‌ای که متد روی آن فراخوانی می‌شود، اشاره می‌کند. به عبارت دیگر، self به کلاس Mobile اجازه می‌دهد تا به ویژگی‌های (Attributes) خاص هر نمونه از نوع Mobile دسترسی پیدا کند.

خطوط ۸ تا ۱۰: این دستور، مقادیر ویژگی‌های برند، مدل و رنگ نمونه را با استفاده از برچسب‌های مشخص، نمایش می‌دهد.

خط ۱۴: در این خط متد show_info از نمونه my_mobile فراخوانی و اجرا می‌شود.

به همین ترتیب می‌توانید متدهای بیشتری برای این کلاس تعریف کنید. به عنوان مثال متد incoming_call برای نمایش پیام هنگام تماس دریافتی می‌باشد. کدهای این متد را در زیرِ متدِ show_info براساس شکل ۱۰ بنویسید.

```
def incoming_call(self, phone_number):
    print("Incomming Call")
    print("Number:", phone_number)
    print("This number is calling")

my_mobile = Mobile("nokia", "1100", "gray")
my_mobile.incoming_call("0912-345-6789")
```

```
Incomming Call
Number: 0912-345-6789
This number is calling
```

شکل ۱۰ـ افزودن متد incoming_call

- اولین خط، تعریف متدی به نام incoming_call را در داخل کلاس Mobile انجام می‌دهد. این متد یک پارامتر به نام phone_number را دریافت می‌کند که نشان‌دهنده شماره تلفن تماس‌گیرنده است.
- دستورات داخل متد، پیام "تماس ورودی" و شماره تلفن تماس‌گیرنده را در خروجی نمایش می‌دهند.
- دقت کنید این متد باید درون کلاس Mobile و هم‌سطح متد __init__ تعریف شود، بنابراین یک tab جلوتر از بدنه کلاس قرار می‌گیرد.
- در آخرین خط برنامه متد incoming_call را بر روی نمونه my_mobile فراخوانی می‌کند.
- آرگومان "۰۹۱۲-۳۴۵-۶۷۸۹" به عنوان پارامتر phone_number ارسال می‌شود که نشان‌دهنده شماره تلفن تماس‌گیرنده است. سپس متد اجرا می‌شود و پیام‌های مشخص شده و شماره تلفن ارائه شده را نمایش می‌دهد.

فعالیت

در کلاس Mobile متدی با نام make_call ایجاد کنید. این متد باید یک پارامتر به نام phone_number دریافت کند که شماره تلفنی که می‌خواهید با آن تماس بگیرید را در خود نگه دارد. متد باید با چاپ پیام‌هایی مناسب، شروع عملیات شماره‌گیری را اعلام کند. به عنوان مثال، می‌توانید پیام‌هایی مانند "Calling..." یا "Dialing..." را چاپ کنید.

تعیین مقدار پیش فرض برای ویژگی‌ها

برای جذابیت و کارایی بیشتر در ادامه، یک ویژگی تعریف خواهد شد که بعد از اجرای برنامه مقدار آنها قابل تغییر باشند. از این رو درون کلاس، ویژگی وضعیت روشن و خاموش بودن (status) و یک مقدار پیش فرض هم برای آن تعریف خواهد شد. کافیست در متد __init__ یک ویژگی جدید مانند کد زیر تعریف کنید.

```
self.status = "on"
```

توجه داشته باشید هنگام تعریف نمونه دیگر ضرورتی ندارد مقداری برای این ویژگی تعریف شود.

تغییر مقدار ویژگی: در این روش پس از ایجاد نمونه my_mobile به سادگی با نوشتن کد زیر، مقدار آن تغییر خواهد کرد.

```
my_mobile = Mobile("nokia", "1100", "gray")
my_mobile.status = "off "
```

در این دستور از نشانه dot یا همان نقطه برای دسترسی به ویژگی یک نمونه استفاده شده است. این خط از پایتون می‌خواهد برای نمونه my_mobile ویژگی status مرتبط با آن را پیدا کند و در ادامه مقدار "off " را درون آن قرار دهد.

ارث‌بری (Inheritance)

فرض کنید در یک شرکت تولیدکنندهٔ گوشی همراه، تاکنون تنها نسخه‌های پایهٔ تلفن همراه تولید می‌شده است؛ گوشی‌هایی که امکانات آنها به ویژگی‌هایی مانند برند، مدل، رنگ و وضعیت محدود بوده و فاقد سیستم‌عامل و دوربین هستند. مهم ترین کاری که با این گوشی‌های همراه می‌توان انجام داد، تماس تلفنی و ارسال پیامک است. در این شرکت، یک کلاس اصلی به نام Mobile وجود دارد که این ویژگی‌های پایه را مدل‌سازی می‌کند.

حالا شرکت تصمیم گرفته است نسل جدیدی از گوشی‌های همراه را طراحی کند؛ گوشی‌های هوشمند (Smartphone) که در کنار ویژگی‌های پایهٔ گوشی همراه، قابلیت‌هایی مانند سیستم‌عامل و دوربین را ارائه می‌دهند و همچنین مدل‌هایی تخصصی‌تر، مانند گوشی‌های مناسب بازی (Gaming phone)، که برای اجرای بهتر بازی‌ها با امکانات سخت‌افزاری قوی‌تر طراحی شده‌اند.

۱ـ Smartphone: تلفن هوشمند نوعی تلفن همراه است که با بهره‌گیری از سیستم‌عامل (Operating System)، امکان اجرای برنامه‌ها و ارائهٔ قابلیت‌های پیشرفته، فراتر از تماس و پیامک را فراهم می‌کند.

- دارای ویژگی دوربین پیشرفته
- دارای ویژگی سیستم عامل قابل نصب و ارتقا

۲ـ Gaming phone: نوعی گوشی هوشمند است که علاوه بر داشتن تمام قابلیت‌های یک Smartphone، به‌طور ویژه برای اجرای بازی‌ها طراحی شده و از امکانات سخت‌افزاری قوی‌تری برخوردار است.

- دارای ویژگی پردازنده گرافیکی قوی و مجزا
- دارای ویژگی حافظه گرافیکی بالا



شکل ۱۱ـ موبایل پایه، هوشمند و موبایل مخصوص بازی

در پی این تصمیم از برنامه‌نویسان شرکت خواسته شده است کلاس‌های لازمر برای موبایل‌های تخصصی را نوشته و تعریف کنند.

جدول ۲

	برند	مدل	رنگ	وضعیت	دوربین	سیستم عامل[۱]	حافظه	پردازنده گرافیکی
Mobile	✓	✓	✓	✓	×	×	×	×
Smartphone	✓	✓	✓	✓	✓	✓	×	×
GamingPhone	✓	✓	✓	✓	✓	✓	✓	✓

همان‌طور که در جدول مشاهده می‌شود، تمام گوشی‌ها ویژگی‌های مشترکی دارند: برند، مدل، رنگ و وضعیت. بنابراین، تعریف مجدد این ویژگی‌ها در هر یک از کلاس‌های گوشی‌های تخصصی، منجر به تکرار کد خواهد شد.

برای جلوگیری از تکرار کد و استفادهٔ بهتر از ویژگی‌های مشترک گوشی‌ها، می‌توان از مفهوم ارث‌بری (Inheritance) در برنامه‌نویسی شیءگرا استفاده کرد.

ارث‌بری به این معناست که یک کلاس جدید می‌تواند ویژگی‌ها (Attributes) و رفتار(Method) یک کلاس دیگر را به ارث ببرد و از آنها استفاده کند.

در این روش، ابتدا یک کلاس کلی‌تر تعریف می‌شود که شامل ویژگی‌های مشترک تمام گوشی‌ها مانند برند، مدل، رنگ و وضعیت است. به این نوع کلاس، کلاس والد (Parent class یا Super class) گفته می‌شود. در این مثال، کلاس Mobile به‌عنوان کلاس والد، شامل تمام ویژگی‌های مشترک همهٔ موبایل‌هاست.

۱- Operating System



شکل ۱۲

سپس کلاس‌های تخصصی‌تر براساس کلاس پایه ایجاد می‌شود. به این کلاس‌ها، کلاس فرزند (Child class) گفته می‌شود. برای نمونه، کلاس Smartphone که از کلاس Mobile ارث‌بری می‌کند، تمام ویژگی‌های کلاس Mobile را به‌صورت خودکار در اختیار دارد و می‌تواند علاوه بر آن، ویژگی‌ها و رفتارهای اختصاصی خود مانند سیستم‌عامل و دوربین را تعریف کند.

استفاده از این رویکرد نتایج زیر را به همراه دارد:

- کاهش حجم کدنویسی
- افزایش خوانایی کد
- بهبود قابلیت نگه‌داری برنامه

در ادامه یک کلاس جدید به نام SmartPhone تعریف شده است که از کلاس Mobile ارث‌بری می‌کند. کد شکل ۱۳ را ملاحظه بفرمایید.

```
1  class Mobile: ...
15
16 class SmartPhone(Mobile):
17     def __init__(self, brand, model, color, os, camera):
18         super().__init__(brand, model, color)
19         self.os = os
20         self.camera = camera
21
22
23 my_smartphone = SmartPhone("samsung", "s25", "gray", "android", "200MP")
24 print(my_smartphone.os)
25
```

ویژگی‌های عمومی ومشترک بین دو کلاس

ویژگی‌های اختصاصی همین کلاس

android

شکل ۱۳ـ تعریف کلاس SmartPhone

خط ۱۶: در این خط، کلاسی به نام SmartPhone تعریف می‌شود که از کلاس Mobile ارث‌بری می‌کند. این یعنی کلاس SmartPhone علاوه بر ویژگی‌ها و متدهای خودش، تمام ویژگی‌ها و متدهای کلاس Mobile را ارث‌بری کرده و به آنها دسترسی دارد.

- به کلاس Mobile در این حالت، کلاس والد (Parent class یا Super class) گفته می‌شود.
- به کلاس SmartPhone، کلاس فرزند (Child class) گفته می‌شود.
- در این خط، جلوی نام کلاس SmartPhone داخل پرانتز نام کلاس والد (Parent class) باید نوشته شود.

خط ۱۷: در این خط، متد سازندهٔ (__init__) کلاسِ SmartPhone تعریف می‌شود.

- این متد در زمان ساخته شدن یک نمونه جدید از کلاس SmartPhone به‌صورت خودکار اجرا می‌شود.

- پارامترهای این سازنده شامل ویژگی‌های عمومی گوشی (brand، model وcolor) و همچنین ویژگی‌های اختصاصی گوشی هوشمند (os و camera) هستند.

خط ۱۸: در این خط، سازنده کلاس والد (Mobile) با استفاده از تابع super فراخوانی می‌شود.

- از متد super برای دسترسی به متدها و ویژگی‌های کلاس والد در مبحث وراثت استفاده می‌شود.
- وقتی یک کلاس از کلاس دیگری ارث‌بری می‌کند، با استفاده از super می‌توان متدهای کلاس والد را فراخوانی کرد، بدون اینکه نام کلاس والد به‌صورت مستقیم نوشته شود.
- دقت کنید در این دستور تنها ویژگی‌های مشترک کلاس فرزند و کلاس والد به متد __init__ از کلاس والد ارسال شده است.
- با این کار، مقداردهی اولیه ویژگی‌هایی که در کلاسMobile تعریف شده‌اند انجام می‌گیرد.
- دقت کنید اگر این خط از کد نوشته نشود، ویژگی‌های کلاس والد در کلاس فرزند، مقداردهی نمی‌شوند.
- به این ترتیب، از تکرار کد جلوگیری شده و ویژگی‌های مشترک بین کلاس‌ها به شکل اصولی مدیریت می‌شوند.

خط ۱۹: در این خط، ویژگی os برای نمونه جاری ساخته می‌شود و مقدار آن برابر با مقدار ورودی پارامتر os قرار می‌گیرد.

این ویژگی نشان‌دهنده سیستم‌عامل گوشی هوشمند است.

خط ۲۰: در این خط، ویژگی camera برای نمونه جاری ساخته می‌شود و مقدار آن برابر با مقدار ورودی پارامتر camera قرار می‌گیرد.

- این ویژگی مشخصات دوربین گوشی هوشمند را نگهداری می‌کند.

خط ۲۳: در این خط، یک نمونه (Instance) از کلاس SmartPhone ساخته می‌شود.

- در زمان ساخت این نمونه، متد سازنده کلاس SmartPhone اجرا شده و مقادیر ارسال‌شده به پارامترها اختصاص داده می‌شوند.
- علاوه بر این، با استفاده از تابع super، سازنده کلاس والد (Mobile) اجرا می‌شود تا ویژگی‌های پایهٔ گوشی، که در کلاس والد تعریف شده‌اند، مقداردهی شوند.

خط ۲۴: در این خط، مقدار ویژگی os از نمونه my_ smartphone خوانده شده و در خروجی برنامه چاپ می‌شود.

- خروجی این دستور مقدار "android" خواهد بود.



شکل ۱۴

فعالیت

به برنامه فوق دستوراتی اضافه کنید تا سایر ویژگی‌های نمونه my_smartphone را چاپ کنید.

فراخوانی متدهای کلاس والد از طریق کلاس فرزند:

تا اینجا متوجه شدید که کلاس SmartPhone ویژگی‌های کلاس Mobile را به ارث برده است. اگر کد کلاس SmartPhone را مشاهده کنید، درون آن به‌جز متد __init__ متد دیگری وجود ندارد. برای بررسی اینکه آیا از کلاس والد (Parent) متدها را هم به ارث برده است یا خیر، دستور زیر را در انتهای برنامه نوشته و برنامه را اجرا می‌کنیم:

```
my_smartphon.show_info()  →  brand: samsung model: s25 color: gray
```

شکل ۱۵

با مشاهده خروجی متوجه می‌شوید نمونه my_smartphone متد show_info را از کلاس Mobile به ارث برده است.

فعالیت

به برنامه فوق دستوراتی اضافه کنید تا سایر متدهای نمونه my_smartphone را فراخوانی و اجرا کند.

ایجاد متد مستقل از والد درون کلاس فرزند:

در اینجا یک سؤال مطرح می‌شود که آیا کلاس فرزند (Child class) می‌تواند خودش به‌صورت مستقل متد داشته باشد؟

پاسخ مثبت است. برای این منظور می‌توانید متدهای جدید و مستقل درون کلاس فرزند تعریف و از آنها استفاده کنید.

در ادامه به کلاس SmartPhone متدهای take_picture برای گرفتن عکس و show_os_info برای نمایش ویژگی نام سیستم عامل را اضافه می‌کنید. کد شکل ۱۶ را به دقت نوشته و اجرا کنید.

```
16 class SmartPhone(Mobile):
17     def __init__(self, brand, model, color, os, camera): ...
21
22     def take_picture(self):
23         print("Picture taken successfully By Camera:", self.camera)
24
25     def show_os_info(self):
26         print("Operating System:", self.os)
27
28 my_smartphone = SmartPhone("samsung", "s25", "gray", "android", "200MP")
29 my_smartphone.take_picture()
30 my_smartphone.show_os_info()
31
```

وقت کنید کدهای این متد جمع

```
Picture taken successfully By Camera: 200MP
Operating System: android
```

شکل ۱۶ــ تعریف دو متد جدید درون کلاس SmartPhone

خطوط ۲۲ و ۲۳: در این خطوط، متد take_picture تعریف شده و درون آن یک پیغام مبنی بر موفقیت در گرفتن عکس چاپ شده است.
خطوط ۲۵ و ۲۶: در این خطوط، متد show_os_info تعریف شده و درون آن مشخصات ویژگی سیستم‌عامل چاپ شده است.

بازنویسی متد والد درون کلاس فرزند (Method Override)

می‌دانید که در کلاس Mobile متدی با نام show_info تعریف شده است که اطلاعات عمومی گوشی را نمایش می‌دهد. حال اگر کلاسی مانند SmartPhone از کلاس Mobile ارث‌بری کند، این متد نیز به کلاس فرزند منتقل شده و بدون تغییر قابل استفاده خواهد بود.
در گوشی‌های هوشمند، علاوه بر اطلاعات عمومی، نمایش اطلاعات بیشتری مانند نام سیستم‌عامل و مشخصات دوربین اهمیت دارد. استفادهٔ مستقیم از متد show_info کلاس والد برای کلاس SmartPhone کافی نیست و لازم است رفتار این متد متناسب با ویژگی‌های کلاس فرزند تغییر کند.
برای حل این مشکل، متد show_info در کلاس SmartPhone بازنویسی می‌شود. به این ترتیب، می‌توان ابتدا با دستور super() متد کلاس والد را اجرا کرد و سپس اطلاعات جدید مورد نیاز را نمایش داد. این کار باعث می‌شود در متد بازنویسی‌شدهٔ کلاس فرزند، ابتدا دستورات متد والد اجرا شده و سپس دستورات جدید نوشته و اجرا شوند.
اکنون می‌توانید متد show_info را مطابق کد زیر در کلاس SmartPhone اضافه نمایید و نتیجهٔ اجرای برنامه را مشاهده کنید.

```
def show_info(self):
    super().show_info()
    print("Operating System:", self.os)
    print("Camera:", self.camera)
```

شکل ۱۷

- در خط اول کد، متدی با نام show_info درون کلاس SmartPhone تعریف شده است. نام این متد دقیقاً با نام متد موجود در کلاس والد یکسان است؛ به همین دلیل این متد به‌عنوان بازنویسی (Override) متد کلاس والد شناخته می‌شود.
- در دومین خط، با استفاده از متد super، متد show_info از کلاس والد (Mobile) اجرا می‌شود. به این ترتیب، اطلاعات عمومی گوشی مانند برند، مدل و رنگ که در کلاس والد تعریف شده‌اند، در ابتدای همین متد نمایش داده می‌شوند.
- در سومین خط، مقدار ویژگی os که مربوط به کلاس SmartPhone است، چاپ می‌شود. این ویژگی نشان‌دهندهٔ نام سیستم‌عامل گوشی هوشمند می‌باشد و در کلاس والد وجود ندارد.
- در آخرین خط، مقدار ویژگی camera نمایش داده می‌شود. این ویژگی نیز مخصوص کلاس SmartPhone بوده و مشخصات دوربین گوشی هوشمند را نشان می‌دهد.

بازنویسی متد والد در کلاس فرزند، به گونه‌ای که نام متد یکسان باقی بماند اما نحوهٔ اجرای آن تغییر کند، بازنویسی متد یا Method Override نامیده می‌شود.

فعالیت

با توجه به مطالبی که تا کنون فرا گرفته اید، کلاس GamingPhone را مطابق توضیحات زیر پیاده‌سازی کنید:
کلاس GamingPhone از کلاس SmartPhone ارث‌بری می‌کند.
این کلاس، علاوه بر ویژگی‌های به‌ارث‌رسیده از کلاس پایه Mobile و کلاس SmartPhone، شامل دو ویژگی اختصاصی زیر است:
- gpu: نام پردازنده گرافیکی
- vram: میزان حافظه گرافیکی

این کلاس، علاوه بر متدهای به‌ارث‌رسیده از کلاس‌های Mobile و SmartPhone، شامل متد زیر نیز می‌باشد:
- متد gaming_mode: این متد هنگام فراخوانی، پیامی مناسب چاپ می‌کند که فعال بودن حالت بازی گوشی را نشان می‌دهد.

پیام چاپ‌شده می‌تواند شامل نام گوشی، نام پردازندهٔ گرافیکی و میزان حافظهٔ گرافیکی باشد.

تا اینجا با تعریف کلاس Mobile و ساخت نمونه‌هایی از آن آشنا شدید. همچنین دیدید که می‌توان به ویژگی‌های هر نمونه دسترسی داشت و مقدار آنها را تغییر داد. اما این پرسش مطرح می‌شود که آیا همهٔ ویژگی‌های یک کلاس باید به‌صورت آزاد قابل تغییر باشند؟

ویژگی‌های عمومی (Public)

در حالت عادی، ویژگی‌های یک کلاس، عمومی (Public) هستند. یعنی از هر بخش از برنامه می‌توان به آنها دسترسی داشت و مقدارشان را تغییر داد. برخی ویژگی‌ها حتی اگر به‌صورت عمومی تعریف شوند، مشکلی ایجاد نمی‌کنند؛ زیرا تغییر آنها باعث بروز خطای منطقی در برنامه نمی‌شود. در کلاس Mobile که قبلاً ساختیم، سه ویژگی brand، model و color از این نوع هستند زیرا:
- تغییر آنها خطرناک نیست.
- نیازی به کنترل خاص ندارند.

بنابراین می‌توانند به‌صورت Public باقی بمانند.
حال فرض کنید ویژگی باتری گوشی را نیز به‌صورت عمومی تعریف کنید (برای سادگی، فرض کنید مقدار اولیه باتری ۱۰۰٪ در نظر گرفته شده است):

```
self.battery = 100
```

در این حالت، می‌توان مقدار باتری را به شکل زیر تغییر داد:

```
my_mobile.battery = 150
my_mobile.battery = -20
```

این دستورات از نظر پایتون بدون خطا اجرا می‌شوند؛ اما از نظر منطقی مقادیر نامعتبری هستند. شارژ باتری یک تلفن همراه نمی‌تواند بیشتر از ۱۰۰ درصد یا مقدار منفی داشته باشد.
مشکل اینجاست که ویژگی‌های کلاس به‌صورت آزاد در دسترس هستند و هیچ کنترلی بر نحوهٔ تغییر آنها وجود ندارد. این موضوع می‌تواند باعث بروز خطاهای منطقی در برنامه شود و در برنامه‌های بزرگ، نگهداری و توسعهٔ آن را دشوار کند.
برای حل این مشکل، می‌توان برخی ویژگی‌ها را به‌گونه‌ای تعریف کرد که دسترسی مستقیم به آنها از بیرون کلاس محدود شود و تغییر آنها فقط از طریق متدهای مشخص انجام گیرد. به این نوع ویژگی‌ها، ویژگی خصوصی (Private) گفته می‌شود.

ویژگی‌های خصوصی (Private)

در پایتون، ویژگی‌های خصوصی با استفاده از دو زیرخط (_ _) در ابتدای نام آنها تعریف می‌شوند.
برای مثال، در کلاس Mobile می‌توان ویژگی باتری را به‌صورت زیر تعریف کرد:

```
self._ _battery = 100
```

با این کار:

- دسترسی مستقیم به ویژگی از بیرون کلاس محدود می‌شود.
- تغییر مقدار آن فقط از طریق متدهای کلاس امکان‌پذیر است.
- کنترل منطقی روی داده‌ها برقرار می‌شود.

ویژگی‌های محافظت‌شده (Protected)

در بخش‌های قبل دیدید که ویژگی‌های یک کلاس می‌توانند عمومی (Public) یا خصوصی (Private) باشند.
اما گاهی شرایطی وجود دارد که:

- نمی‌خواهید یک ویژگی از بیرون کلاس مستقیماً استفاده یا تغییر داده شود.
- ولی می‌خواهید کلاس‌های فرزند بتوانند به آن ویژگی دسترسی داشته باشند.

برای چنین حالتی از ویژگی‌های محافظت‌شده (Protected) استفاده می‌شود. در زبان پایتون، اگر نام یک ویژگی با یک زیرخط (_) شروع شود، آن ویژگی به‌عنوان Protected در نظر گرفته می‌شود.

```
self._status = "on"
```

جدول ۳

سطح دسترسی	نحوه نوشتن	توضیح
عمومی (Public)	battery	دسترسی آزاد از همه‌جا
خصوصی (Private)	_ _battery	دسترسی فقط داخل همان کلاس
محافظت‌شده (Protected)	_status	دسترسی از درون کلاس و کلاس‌های فرزند

این نوع نام‌گذاری نشان می‌دهد که ویژگی برای استفادهٔ داخلی کلاس و کلاس‌های فرزند طراحی شده است و نباید مستقیماً از بیرون کلاس تغییر داده شود.

نکته

Protected در پایتون یک قرارداد برنامه‌نویسی است، نه یک محدودیت اجباری. یعنی پایتون جلوی دسترسی مستقیم را نمی‌گیرد، اما به برنامه‌نویس هشدار می‌دهد که این ویژگی نباید مستقیماً از بیرون کلاس تغییر داده شود.

تا اینجا با سه سطح دسترسی به ویژگی‌های یک کلاس آشنا شدید:
سطوح دسترسی مختلف شامل ویژگی‌های عمومی، خصوصی و محافظت‌شده، به شما کمک می‌کنند تا دسترسی به داده‌های یک کلاس را کنترل کنید. به همین دلیل لازم است جزئیات داخلی کلاس‌ها پنهان شوند و دسترسی به داده‌ها به‌صورت کنترل‌شده انجام گیرد. این مفهوم در برنامه‌نویسی شیءگرا با عنوان کپسوله‌سازی (Encapsulation) شناخته می‌شود. کپسوله‌سازی به معنای:
پنهان‌سازی جزئیات داخلی یک کلاس و کنترل نحوهٔ دسترسی به داده‌ها و رفتارهای آن می‌باشد و یکی از مفاهیم اصلی برنامه‌نویسی شیءگرا محسوب می‌شود.

ارزشیابی شایستگی تولید برنامه شیء گرا

استاندارد عملکرد کار: هنرجو بتواند با به‌کارگیری مفاهیم شیءگرایی در پایتون، کلاس و شیء تعریف کند، وراثت و بازنویسی متدها را پیاده‌سازی نماید، سطوح دسترسی را مدیریت کند و رفتار کلاس‌ها را تا سطح مقدماتی متاکلاس تحلیل و اجرا کند به‌گونه‌ای که برنامه بدون خطا و با ساختار صحیح اجرا شود.

ردیف	مراحل کار	شرایط انجام کار (ابزار،مواد، تجهیزات، زمان، مکان و ...)	ابزارهای ارزشیابی	استاندارد (شاخص‌ها/داوری/نمره‌دهی)	نتایج ممکن	نمره کسب شده
۱	ایجاد و کار با کلاس پایه	رایانه یا لپ‌تاپ سالم، نصب Python (نسخه x.3)، ویرایشگر کد (مانند IDLE یا VS Code) زمان ۲۵دقیقه، اجرای فعالیت در کارگاه .	بررسی فایل کد نوشته‌شده، اجرای عملی برنامه و مشاهده خروجی، چک‌لیست عملکردی.	۱ـ تعریف صحیح کلاس و سازنده (__init__) ۲ـ ایجاد نمونه بدون خطای اجرایی ۳ـ تغییر صحیح مقدار ویژگی‌ها ۴ـ اجرای برنامه و نمایش خروجی درست ۵ـ انجام کنجکاوی‌ها ۶ـ انجام مراحل به روش‌های دیگر	۳	
				۱ـ تعریف صحیح کلاس و سازنده (__init__) ۲ـ ایجاد نمونه بدون خطای اجرایی ۳ـ تغییر صحیح مقدار ویژگی‌ها ۴ـ اجرای برنامه و نمایش خروجی درست	۲	
				انجام ناقص مراحل	۱	
۲	مدیریت ویژگی‌ها و سطوح دسترسی	رایانه مجهز به Python، شامل کلاس با ویژگی‌های مختلف، زمان ۱۵ دقیقه، محیط کارگاهی استاندارد.	مشاهده عملی، تحلیل کد هنرجو	۱ـ تعریف ویژگی با مقدار پیش‌فرض ۲ـ استفاده صحیح از _ و __ ۳ـ درک محدودیت دسترسی و تحلیل خطا ۴ـ رعایت قراردادهای نام‌گذاری ۵ـ انجام کنجکاوی‌ها ۶ـ انجام مراحل به روش‌های دیگر	۳	
				۱ـ تعریف ویژگی با مقدار پیش‌فرض ۲ـ استفاده صحیح از _ و __ ۳ـ درک محدودیت دسترسی و تحلیل خطا ۴ـ رعایت قراردادهای نام‌گذاری	۲	
				انجام ناقص مراحل	۱	
۳	پیاده‌سازی وراثت	رایانه با محیط اجرای Python، شامل کلاس والد، زمان ۲۵ دقیقه، کلاس رایانه.	مشاهده عملیات	۱ـ تعریف صحیح کلاس فرزند ۲ـ استفاده درست از super ۳ـ انتقال صحیح ویژگی‌های والد ۴ـ انجام کنجکاوی‌ها ۵ـ انجام مراحل به روش‌های دیگر	۳	
				۱ـ تعریف صحیح کلاس فرزند ۲ـ استفاده درست از super ۳ـ انتقال صحیح ویژگی‌های والد	۲	
				انجام ناقص مراحل	۱	
۴	توسعه و بازنویسی رفتار کلاس‌ها شرایط انجام کار	رایانه، زمان ۲۵ دقیقه، محیط کارگاهی.	مشاهده عملی	۱ـ تعریف متد جدید در کلاس فرزند ۲ـ بازنویسی صحیح متد والد (Override) ۳ـ انجام کنجکاوی‌ها ۴ـ انجام مراحل به روش‌های دیگر	۳	
				۱ـ تعریف متد جدید در کلاس فرزند ۲ـ بازنویسی صحیح متد والد (Override)	۲	
				انجام ناقص مراحل	۱	
۵	بررسی نوع و ساختار کلاس	رایانه مجهز به Python، زمان ۱۵ دقیقه، محیط کلاس عملی.	اجرای دستورات type() و تحلیل خروجی، پرسش شفاهی تحلیلی	۱ـ استفاده صحیح از type() ۲ـ تشخیص تفاوت کلاس و نمونه ۳ـ تحلیل خروجی به صورت مفهومی ۴ـ انجام کنجکاوی‌ها ۵ـ انجام مراحل به روش‌های دیگر	۳	
				۱ـ استفاده صحیح از type() ۲ـ تشخیص تفاوت کلاس و نمونه ۳ـ تحلیل خروجی به صورت مفهومی	۲	
				انجام ناقص هر مرحله	۱	

شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش:			احراز همه شایستگی‌های غیرفنی	۲	
			عدم احراز هر یک از شایستگی‌های غیرفنی	۱	
ارزشیابی کار (شایستگی انجام کار)				بلی ☐	
				خیر ☐	

توضیحات:

۱ـ معیار شایستگی انجام کار:

ـ کسب حداقل نمره ۲ از مراحل ۱، ۲، ۳ و ۴ (مراحل بحرانی)

ـ کسب حداقل نمره ۲ از بخش شایستگی‌های غیرفنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش

ـ کسب حداقل میانگین ۲ از مراحل کار

۲ـ تعریف سطوح شایستگی: صرف‌نظر از اینکه یک تکلیف کاری در چه سطحی از صلاحیت حرفه‌ای انجام می‌شود، در هر محیط کاری ممکن است انجام هر کار با کیفیت مشخصی مورد انتظار باشد. سطح شایستگی انجام کار، معیار اساسی ارزشیابی است.

مهارت (سطح سه): ماهر و قادر به آموزش و هدایت دیگران، توانایی برنامه‌ریزی و تحلیل، پاسخ‌گویی در برابر کارهای خود، سروکار داشتن با سطح وسیعی از کارها و فعالیت‌ها

تسلط (سطح چهار): خبرگی در انجام کار و آموزش دیگران، ایجاد نوآوری، سازگاری، عیب‌یابی، هدایت و راهنمایی دیگران.

واحد یادگیری ۲
توسعه برنامه شیءگرا

آیا تابه‌حال اندیشیده‌اید

- چرا نمی‌توان همیشه از متغیرها برای نگهداری اطلاعات استفاده کرد؟
- چه تفاوتی بین «ذخیره موقت اطلاعات در حافظه (RAM)» و «ذخیره در فایل» وجود دارد؟
- اگر چند بار برنامه را اجرا کنید و هر بار در فایل بنویسید، چه اتفاقی برای محتوای قبلی می‌افتد؟
- چه زمانی استفاده از فایل متنی ساده کافی است و چه زمانی بهتر است از JSON استفاده کنید؟
- اگر مسیر فایل اشتباه باشد، برنامه چگونه باید رفتار کند؟
- اگر قرار باشد یک برنامه فروشگاهی طراحی کنید، چه اطلاعاتی باید در فایل ذخیره شود و چرا؟

در پایان از هنرجو انتظار می‌رود

۱ از کلاس "Path" و ماژول "pathlib" برای مدیریت مسیرها (Path Objects) استفاده کند.

۲ محتوای فایل‌های متنی را بخواند و داده‌ها را در آنها بنویسد، اضافه کند و فایل‌ها را ایمن ببندد.

۳ داده‌ها را از چندین فایل بخواند، محتوای آنها را ترکیب کند و با استفاده از حلقه‌ها، مجموعه‌ای از فایل‌ها را پردازش نماید.

۴ خطاهای رایج مانند "FileNotFoundError" و "ZeroDivisionError" را تشخیص داده و با استفاده از "try/except" از توقف برنامه جلوگیری کند و پیام‌های خطای مناسب نمایش دهد.

۵ ساختار "JSON" را درک کرده و داده‌ها را با استفاده از ()json.loads بخواند و با ()json.dump در فایل بنویسد.

استاندارد عملکرد

هنرجو باید بتواند با استفاده از شیء Path به‌صورت ایمن و دقیق عملیات ایجاد، خواندن، نوشتن و مدیریت فایل‌های متنی و JSON را انجام دهد و خطاها و استثناهای احتمالی را به‌درستی کنترل کند.

آشنایی با فایل در پایتون

در این واحد یادگیری با کار با فایل‌ها و ذخیره‌سازی داده‌ها آشنا می‌شوید تا برنامه‌های شما کاربردی‌تر و پایدارتر شوند. می‌آموزید چگونه داده‌ها را ذخیره و بازیابی کنید و با استفاده از مدیریت استثناها از توقف برنامه در شرایط غیرمنتظره جلوگیری نمایید.

همچنین با ماژول json آشنا خواهید شد که امکان نگهداری اطلاعات کاربران را فراهم می‌کند.

فرض کنید برنامه‌ای برای ثبت مخاطبین یک تلفن همراه نوشته‌اید. اگر برنامه را اجرا کنید و چند مخاطب وارد کنید، تا زمانی که برنامه باز است اطلاعات در حافظه نگهداری می‌شوند. اما اگر برنامه بسته شود، تمام اطلاعات از بین می‌روند.



شکل ۱۸

برای اینکه اطلاعات به‌صورت دائمی ذخیره شوند و بعداً بتوانید دوباره به آنها دسترسی داشته باشید، لازم است از فایل‌ها استفاده کنید. فایل‌ها امکان ذخیره اطلاعات روی حافظه جانبی (مانند هارد دیسک) را فراهم می‌کنند.

کار با فایل

اطلاعات زیادی در فایل‌های متنی ذخیره می‌شوند؛ مثل نمرات دانش‌آموزان، لیست قیمت‌ها، داده‌های آب و هوا،ترافیک، یا متن یک داستان. برای اینکه یک برنامه بتواند از این اطلاعات استفاده کند، باید فایل را بخواند. اولین گام برای استفاده از یک فایل متنی، این است که فایل را وارد حافظه برنامه کنید، بعد از آن می‌توانیدکل فایل را یک‌جا بخوانید یا آن را خط‌به‌خط بررسی کنید. خواندن فایل به برنامه کمک می‌کند تا اطلاعات را تحلیل، نمایش یا تغییر دهد.



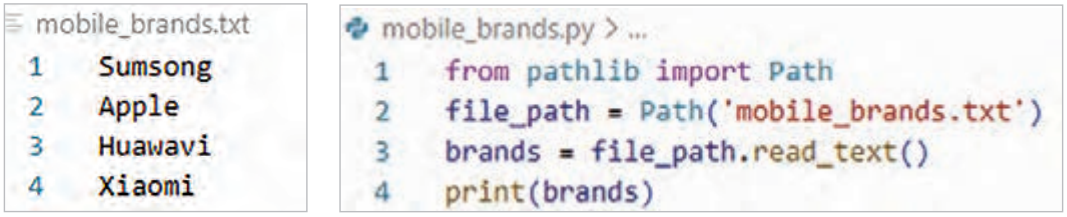
شکل ۱۹

ایجاد فایل متنی

ابتدا یک فایل متنی به نام mobile_bransd.txt بسازید و برندها را در آن بنویسید (هر برند دریک خط):

خواندن کل فایل در پایتون

برای خواندن کل محتویات فایل از کتابخانه pathlib استفاده کنید.



شکل ۲۰

توضیح کد:

from pathlib import Path

ـ این خط کلاس Path را از کتابخانهٔ pathlib وارد می‌کند. Path ابزاری برای کار با مسیر فایل‌ها و پوشه‌ها است و استفاده از آن ساده‌تر و امن‌تر از open معمولی است.

path = Path('mobile_brands.txt')

در اینجا path یک کلاس است که برای مدیریت مسیر فایل‌ها ساخته شده است. با نوشتن:

path = Path('mobile_brands.txt')

یک نمونه (object) از کلاس Path ساخته می‌شود. این نمونه می‌داند که فایل mobile_brands.txt کجاست، ولی هنوز خود فایل باز نشده یا خوانده نشده به عبارت ساده: path فقط نشانی فایل روی رایانه را نگه می‌دارد.
وقتی می‌نویسید path.read_text() آن وقت فایل باز و خوانده می‌شود، یک Path object ساخته شده است که به فایل mobile_brands.txt اشاره می‌کند. چون این فایل در همان پوشه‌ای است که فایل پایتون (.py) قرار دارد، فقط نام فایل کافی است. اگر فایل در پوشه‌ای دیگر بود، باید مسیر کامل یا نسبی آن را بدهید.

brnds = path.read_text()

ـ با متد read_text کل محتوای فایل خوانده می‌شود. نتیجه به‌صورت یک رشته (string) در متغیر contents ذخیره می‌شود. هر خط فایل در رشته با کاراکتر \n جدا شده است.

ـ محتویات brands درخروجی چاپ می‌شود.

```
print(brands)
```

خروجی مثال:

```
Sumsong
Apple
Huawavi
Xiaomi
```

شکل ۲۱

حذف خط خالی اضافی: گاهی read_text() یک خط خالی اضافی در انتهای فایل ایجاد می‌کند. برای حذف آن می‌توانید از متد rstrip استفاده کنید.

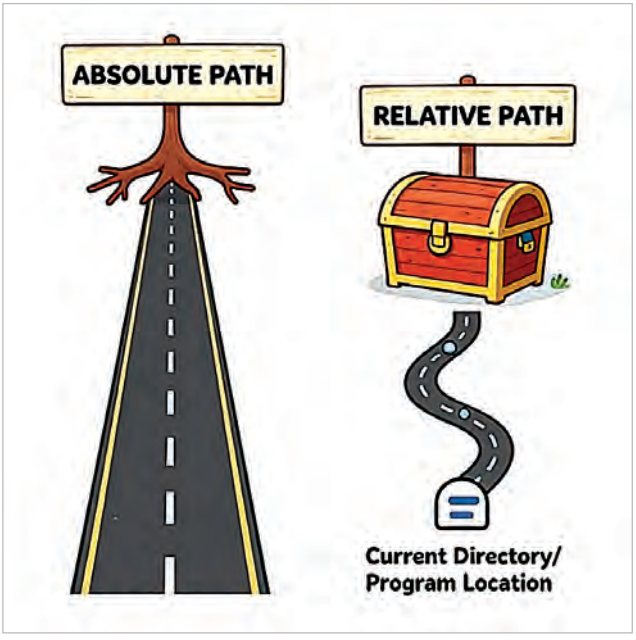
```
brands = brands.rstrip()
```

حالا خروجی تمیز و بدون خط خالی نمایش داده می‌شود.

مسیرهای نسبی و مطلق فایل‌ها

آدرس نسبی و مطلق مانند داشتن یک نقشه دقیق برای دسترسی به فایل‌های شماست. آدرس مطلق تضمین می‌کند که برنامه شما روی هر سیستمی، دقیقاً همان فایل را پیدا کند؛ اما آدرس نسبی به شما کمک می‌کند کدی بنویسید که بدون تغییر، در هر پوشه‌ای جابجا شود و باز هم فایل‌های مورد نیازش را بیابد.

وقتی نام فایلی مثل mobile_brands.txt را به Path می‌دهید، پایتون در پوشه‌ای که فایل در حال اجرا (یعنی برنامه py. شما) در آن قرار دارد، جستجو می‌کند.گاهی اوقات، بسته به نحوه سازماندهی کارهایتان، فایلی که می‌خواهید باز کنید ممکن است در همان پوشه برنامه شما نباشد. مثلاً، شاید برنامه‌هایتان را در پوشه‌ای به نام py_progs و داخل آن پوشه‌ای به نام text_files برای جدا کردن فایل‌های متنی از برنامه‌ها نگهداری کنید. با اینکه text_files داخل py_progs است، اگر فقط نام فایلی را در text_files به Path بدهید، کار نمی‌کند؛ چون پایتون فقط در py_progs جستجو می‌کند و متوقف می‌شود؛ به text_files نگاه نمی‌کند. برای اینکه پایتون بتواند فایل‌ها را از پوشه‌ای غیر از پوشه برنامه شما باز کند، نیاز به مسیر صحیح دارید.



شکل ۲۲

دو راه برای مشخص کردن مسیر فایل‌ها در برنامه‌نویسی وجود دارد:
۱ـ آدرس نسبی (Relative File Path): این آدرس به پایتون می‌گوید که مکان مورد نظر را نسبت به پوشه برنامه در حال اجرا پیدا کند. از آنجایی که text_files داخل py_progs است، باید مسیری بسازید که با پوشه text_files شروع شود و به نام فایل ختم شود. نحوه ساخت این مسیر این‌گونه است:

```
path = Path('text_files/filename.txt')
```

۲ـ آدرس مطلق (Absolute File Path): در این نوع آدرس دهی شما می‌توانید دقیقاً به پایتون بگویید فایل در کجای رایانه شما قرار دارد، صرف نظر از اینکه برنامه در حال اجرا کجاست. اگر آدرس نسبی کار نکرد، می‌توانید از آدرس مطلق استفاده کنید. مثلاً، اگر text_files را در پوشه‌ای غیر از py_progs قرار داده‌اید، دادن آدرس 'text_files/filename.txt' کافی نیست؛ چون پایتون آن را فقط داخل python_work جست‌وجو می‌کند. در این حالت، باید آدرس مطلق را بنویسید تا مشخص شود پایتون کجا را جست‌وجو کند. آدرس‌های مطلق معمولاً بلندتر هستند، چون از پوشه ریشه سیستم عامل شروع می‌شوند:

```
path = Path('/home/eric/py_progs/text_files/filename.txt')
```

با استفاده از آدرس‌های مطلق، می‌توانید فایل‌ها را از هر نقطه‌ای در سیستم خود بخوانید. برای شروع، ساده‌ترین کار این است که فایل‌ها را در همان پوشه برنامه خودتان یا در پوشه‌ای مانند text_files درون پوشه برنامه نگهداری کنید.

نکته

سیستم‌های ویندوز هنگام نمایش مسیر فایل‌ها از بک‌اسلش (\) به جای اسلش (/) استفاده می‌کنند، اما شما همیشه در کد خود از اسلش (/) استفاده کنید، حتی در ویندوز. کتابخانه pathlib به طور خودکار هنگام تعامل با سیستم عامل شما، نمایش صحیح مسیر را به کار می‌گیرد.

کار با محتویات فایل

بعد از خواندن فایل متنی، محتوای آن در برنامه به‌صورت یک رشته (string) در اختیارشما قرار می‌گیرد.

تبدیل محتوا به لیست: برای اینکه بتوانید روی هر برند تلفن همراه جداگانه کارکنید، لازم است متن فایل را به لیستی از خطوط تبدیل کنید.
حالا هر برند یک عضو از لیست است و می‌توانید با آن کارکنید.
-splitlines() رشته متنی را به لیستی از خطوط تقسیم می‌کند.

```
1  from pathlib import Path
2  file_path = Path('mobile_brands.txt')
3  brands = file_path.read_text()
4  brands = file_path.read_text().rstrip()
5  brands = brands.splitlines()
6  print(brands)
```

```
['Sumsong', 'Apple', 'Huawayi', 'Xiaomi']
```

شکل ۲۳

```
1  from pathlib import Path
2  file_path = Path('mobile_brands.txt')
3  brands = file_path.read_text()
4  brands = file_path.read_text().rstrip()
5  brands = brands.splitlines()
6  for brand in brands:
7      print("Brands:", brand)
```

```
Brand: Sumsong
Brand: Apple
Brand: Huawavi
Brand: Xiaomi
```

شکل ۲۴

شمارش داده‌ها: مثلاً تعداد برندهای موجود در فایل:

```
from pathlib import Path
file_path = Path('mobile_brands.txt')
brands = file_path.read_text()
brands = file_path.read_text().rstrip()
brands = brands.splitlines()
print("Number of Brands:", len(brands))
```

```
Number of Brands: 4
```

شکل ۲۵

فعالیت

برنامه را به نحوی تغییر دهید که بتوانید تعداد کاراکترهای هر برند را مشاهده کنید.

جست‌وجو در محتویات فایل

بررسی اینکه آیا یک برند خاص در فایل وجود دارد یا نه:

```
if  "Apple"in brands:
    print("Apple brands found!")
else:
    print("Apple brands not found!")
```

```
Apple brands found!
```

شکل ۲۶

پیمایش خطوط فایل با حلقه for

```
from pathlib import Path
file_path = Path('mobile_brands.txt')
brands = file_path.read_text()
brands = file_path.read_text().rstrip()
brands = brands.splitlines()
brand_text = ""
for brand in brands:
    brand_text += brand.rstrip()
    print(brand_text)
```

```
Sumsong
SumsongApple
SumsongAppleHuawavi
SumsongAppleHuawaviXiaomi
```

شکل ۲۷

فعالیت

کدهای برنامه را به نحوی تغییردهید که بعد از نام هر برند یک جداکننده قرار گیرد.

نوشتن در فایل: نوشتن یک خط

پس از تعیین مسیر، می‌توانید با استفاده از متد write_text() در یک فایل متنی بنویسید. برای دیدن نحوهٔ کار این تابع، یک پیام ساده نوشته و آن را به جای چاپ روی صفحه، در فایل متنی ذخیره کنید (شکل ۲۷):

ـ پس از نوشتن برنامه و اجرای آن، خروجی در ترمینال مشاهده نمی‌کنید. حالا فایل mobile_brands.txt را باز کنید. مشاهده می‌کنید متن شما در فایل نوشته و محتویات قبلی آن پاک شده است.

ـ اگر فایل از قبل وجود داشته باشد، write_text() محتوای فعلی فایل را پاک کرده و محتوای جدید را در فایل می‌نویسد.

ـ متد write_text() چند کار را در پشت صحنه انجام می‌دهد. اگر فایلی که مسیر به آن اشاره می‌کند وجود نداشته باشد، آن فایل را ایجاد می‌کند.

ـ همچنین، پس از نوشتن رشته در فایل، مطمئن می‌شود که فایل به درستی بسته شده است. فایل‌هایی که به درستی بسته نشده‌اند می‌توانند منجر به از دست رفتن یا خراب شدن داده‌ها شوند.

```
from pathlib import Path
file_path = Path('mobile_brands.txt')
brands = file_path.read_text()
brands = file_path.read_text().rstrip()
file_path.write_text("I love programming.")
```

شکل ۲۸

نکته

دو متد read_text و write_text به‌طور داخلی مدیریت بستن فایل را انجام می‌دهند. در این حالت‌ها، pathlib خودش تضمین می‌کند که فایل به‌طور ایمن باز و سپس بسته شود.

اضافه کردن برند جدید

برای نوشتن بیش از یک خط در یک فایل، باید رشته‌ای بسازید که شامل کل محتوای فایل باشد و سپس write_text () را با آن رشته فراخوانی کنید.

```
1  from pathlib import Path
2  file_path = Path('mobile_brands.txt')
3  brands = file_path.read_text()
4  brands = file_path.read_text().rstrip()
5  brands = "Nokia\n"
6  brands += "SONY\n"
7  brands += "Microsoft\n"
8  file_path.write_text(brands)
9  print("brand added success fully")
```

mobile_brands.txt

```
1  Sumsung
2  Apple
3  Huawavi
4  Xiaomi
5  Nokia
6  SONY
7  Microsoft
```

شکل ۲۹

فعالیت

دو برند دیگر "LG" و "Motorola" را به فایل "mobile_brands.txt" اضافه کنید.

فعالیت

برنامه‌ای بنویسیدکه در یک حلقه مدل تلفن‌های همراه را از کاربر بپرسد. وقتی پاسخ می‌دهد، آنها را در فایلی به نام "mobile_model.txt" بنویسد. هر مدل در یک خط از فایل ظاهر شود.

مدیریت استثناها (Exceptions)

در هنگام اجرای برنامه، ممکن است خطاهایی رخ دهند که باعث توقف برنامه شوند. پایتون برای برخورد با این خطاها از سازوکاری به نام استثنا (Exception) استفاده می‌کند.

هر وقت پایتون در اجرای برنامه با مشکلی روبه‌رو شود و نداند چه باید بکند، یک «شیء استثنا» ایجاد می‌کند. اگر برنامه‌نویس کدی برای مدیریت آن استثنا نوشته باشد، برنامه بدون توقف به کار خود ادامه می‌دهد. اما اگر استثنا مدیریت نشده باشد، اجرای برنامه متوقف و پیغامی شامل نوع خطا و محل آن نمایش داده می‌شود که به آن ردیابی (Traceback) می‌گویند.



شکل ۳۰

■ **مدیریت خطای اشتباه درنوع داده:** گاهی برنامه‌نویس اشتباهاً داده‌ای از نوع نادرست وارد می‌کند. مثلاً وقتی می‌خواهد رشته‌ای (متن) را به عدد تبدیل کند اما مقدارش فقط متن باشد، خطای ValueError پیش می‌آید:

```
1   age = int("Hello")
2   print(age)
```

```
  File "c:\Users\homepc021\Desktop\progrs\exep.py", line 2, in <module>
    age = int("Hello")
ValueError: invalid literal for int() with base 10: 'Hello'
```

شکل ۳۱



شکل ۳۲

■ **مدیریت خطای تقسیم بر صفر:** یکی از خطاهای رایج در برنامه‌نویسی، تقسیم عدد بر صفر است. این کار از نظر ریاضی ممکن نیست و پایتون سعی می‌کند برنامه را ادامه دهد ولی این کار ممکن نیست، پس برنامه متوقف و پایتون چیزی به اسم ردیابی برگشتی یا همان Traceback نشان می‌دهد.

یعنی وقتی پایتون با چنین دستوری روبه‌رو می‌شود، استثنایی به نام ZeroDivisionError ایجاد می‌کند. یعنی: خطا دقیقاً در کدام خط پیش آمده است. چه نوع خطایی بوده است (مثلاً ZeroDivisionError یا ValueError). و تابع یا بخشی از کد که باعث خطا شده است را نشان می‌دهد.

```
Traceback (most recent call last):
  File "c:\Users\homepc021\Desktop\progrs\exep2.py", line 1, in <module>
    print(5/0)
          ~~
ZeroDivisionError: division by zero
```

شکل ۳۳

از این اطلاعات می‌توانید برای اصلاح برنامه استفاده کنید. به پایتون بگویید که هنگام وقوع این نوع استثنا چه کاری انجام دهد. به این ترتیب، اگر دوباره اتفاق بیفتد، آماده خواهید بود.

■ **مدیریت خطا با بلوک‌های try-except**: برای مدیریت استثناها در پایتون از بلوک‌های try و except استفاده می‌شود.

ـ بخش try شامل دستوری است که ممکن است خطا ایجاد کند.

ـ بخش except مشخص می‌کند اگر خطایی رخ داد، برنامه چه واکنشی نشان دهد.

با استفاده از این روش، می‌توانید برنامه‌هایی بنویسید که به جای نمایش پیام‌های پیچیده و گیج‌کننده، پیام‌های قابل فهم به کاربر نشان دهد.

مدیریت خطای اشتباه درنوع داده: در این‌جا پایتون نمی‌تواند کلمه "Hello" را به عدد تبدیل کند، پس استثنا تولید می‌شود، اما برنامه با استفاده از بلوک try-except کنترل خطا را در دست می‌گیرد.

```
try:
    age = int("Hello")
except ValueError:
    print("please enter correct value")
```

شکل ۳۴

در اینجا پایتون نمی‌تواند عدد ۵ را بر عدد ۰ تقسیم کند، پس استثنا تولید می‌شود، اما برنامه با استفاده از بلوک try-except کنترل خطا را در دست می‌گیرد.

exep2.py

```
try:
  print(5/0)
except ZeroDivisionError:
  print("You can't divide by zero!")
```

شکل ۳۵

استفاده از استثنائات برای جلوگیری از خطا: مدیریت صحیح استثناها (خطاها) زمانی اهمیت حیاتی پیدا می‌کند که برنامه پس از وقوع خطا، همچنان کارهای دیگری برای انجام دادن داشته باشد. این موقعیت به‌ویژه در برنامه‌هایی که به طور مداوم از کاربر ورودی دریافت می‌کنند، رایج است. اگر برنامه بتواند به ورودی نامعتبر به شیوه‌ای مناسب پاسخ دهد، به‌جای متوقف شدن و از کار افتادن، می‌تواند به‌طور هوشمندانه ورودی‌های معتبرتری را از کاربر درخواست کند و به اجرای خود ادامه دهد.

برای یادگیری این موضوع توضیحاتی که در ادامه آمده است را با دقت بخوانید و انجام دهید.

ـ این برنامه از کاربر می‌خواهد که عدد اول را وارد کند و اگر کاربر برای خروج q را وارد نکند، عدد دوم را وارد می‌کند.

exep3.py > ...

```
print("Give me two numbers, and I'll divide them.")
print("Enter 'q' to quit.")
while True:
    first_number = input("\nFirst number: ")
    if first_number == 'q':
        break
    second_number = input("Second number: ")
    if second_number == 'q':
        break
    answer = int(first_number) / int(second_number)
    print(answer)
```

شکل ۳۶

ـ سپس این دو عدد را بر هم تقسیم می‌کند تا به جواب برسد.
ـ این برنامه هیچ کاری برای مدیریت خطاها انجام نمی‌دهد، اگر کاربر عدد دوم را صفر واردکند، درخواست تقسیم بر صفر باعث از کار افتادن آن می‌شود:

```
Give me two numbers, and I'll divide them.
Enter 'q' to quit.

First number: 8
Second number: 0
Traceback (most recent call last):
  File "c:\Users\homepc021:\Desktop\progrs\exep3.py", line 10, in <module>
    answer = int(first_number) / int(second_number)
             ~~~~~~~~~~~~~~~~~^~~~~~~~~~~~~~~~~~~~
ZeroDivisionError: division by zero
```

شکل ۳۷

نکته

اگرچه توقف ناگهانی برنامه (crash) قطعاً بد است، اما نمایش جزئیات فنی خطا (Traceback) نیز ایده خوبی نیست. این جزئیات برای کاربران معمولی سردرگم‌کننده است و در محیط‌های مخرب، اطلاعات ارزشمندی را در اختیار مهاجمان قرار می‌دهد. مهاجم با دیدن Traceback، نه تنها نام برنامه شما را می‌فهمد، بلکه می‌تواند بخشی از منطق کد را که دچار نقص شده‌است، مشاهده‌کند. یک مهاجم حرفه‌ای می‌تواند از این اطلاعات برای فهمیدن اینکه دقیقاً چه نوع حملاتی را باید علیه کد شما به کار ببرد، استفاده کند.



شکل ۳۸

■ **کاربرد The else Block:** وقتی در پایتون از دستورات try و except استفاده کنید، در واقع به مفسر (رایانه) می‌گویید: "یک کار را امتحان کن، و اگر خراب شد، این کار جایگزین را انجام بده."
حالا، بلوک else دقیقاً به چه دردی می‌خورد؟
else یعنی: "فقط و فقط در صورتی این کار را انجام بده که بلوک try کاملاً، بدون هیچ خطایی، موفقیت‌آمیز بود."

اگر خطایی در بلوک try (مثلاً تقسیم عدد بر صفر) رخ دهد، پایتون مستقیم به بلوک except می‌رود و بلوک else کاملاً نادیده گرفته می‌شود. این قابلیت کمک می‌کند، کدی را که فقط پس از موفقیت کامل اجرا شود (مثلاً نمایش نتیجه نهایی یا ذخیره در فایل)، از کدی که فقط برای تست خطا نوشته شده جدا کنید. این کار برنامه را هم تمیزتر و هم منطقی‌تر می‌کند.

exep3.py > ...

```
print("Give me two numbers, and I'll divide them.")
print("Enter 'q' to quit.")
while True:
    first_number = input("\nFirst number: ")
    if first_number == 'q':
        break
    second_number = input("Second number: ")
    if second_number == 'q':
        break
    try:
        answer = int(first_number) / int(second_number)
    except ZeroDivisionError:
        print("You can't divide by 0!")
    else:
        print(answer)
```

```
Give me two numbers, and I'll divide them.
Enter 'q' to quit.

First number: 9
Second number: 0
You can't divide by 0!

First number: 6
Second number: 1
6.0

First number: 8
Second number: 6
1.3333333333333333

First number: q
```

شکل ۳۹



شکل ۴۰

مدیریت خطای File Not Found Error: این خطا یکی از رایج‌ترین استثناهایی است که برنامه‌نویسان هنگام کار با فایل‌ها با آن روبه‌رو می‌شوند. دلایل متعددی برای این خطا وجود دارد: مسیر اشتباه، نام فایل غلط، یا حذف شدن فایل.

بلوک try-except اجازه می‌دهد که کد را اجرا کنید و در صورت بروز این مشکل خاص، به جای متوقف شدن برنامه، یک واکنش مناسب نشان دهید.

در مثال برنامه تلاش می‌کند فایلی را که اصلاً وجود ندارد بخواند.

```
from pathlib import Path
path = Path('alice.txt')
contents = path.read_text(encoding='utf-8')
```

شکل ۴۱



شکل ۴۲

توجه داشته باشید که از متد read_text() کمی متفاوت از آنچه قبلاً دیدید استفاده شده است. آرگومان encoding زمانی ضروری است که کدگذاری پیش‌فرض سیستم شما با کدگذاری فایلی که در حال خواندن آن هستید مطابقت نداشته باشد. این وضعیت به احتمال زیاد هنگام خواندن فایلی رخ می‌دهد که روی سیستم شما ایجاد نشده باشد.

پایتون نمی‌تواند از یک فایلی که وجود ندارد بخواند، بنابراین یک استثنا (Exception) ایجاد می‌کند (به عبارتی، خطا صادر می‌کند).

بنابراین برای مدیریت خطایی که ایجاد می‌شود، بلوک try با خطی که در traceback به عنوان مشکل‌ساز شناسایی شده است، آغاز می‌شود.



شکل ۴۳

در مثال کد خط ۴ که شامل read_text() است باعث بروز خطا شده است.

در این مثال، کد موجود در بلوک try خطای FileNotFoundError تولید می‌کند، بنابراین یک بلوک except که با خطا مطابقت دارد، باید نوشته شود. سپس پایتون کد موجود در آن بلوک را زمانی که فایل پیدا نمی‌شود اجرا می‌کند و نتیجه به جای یک traceback، یک پیام خطای کاربرپسند است:

```
Sorry, the file alice.txt does not exist.
```

شکل ۴۴

حالا اگر فایل alice.txt را ایجاد کنید و آن را در پوشه مناسب قرار دهید این بار بلوک try اجرا خواهد شد.
ـ برنامه فایل را پیدا می‌کند و وارد بلوک else می‌شود.
ـ متن آلیس در سرزمین عجایب را واردکنید، پایتون متن فایل را به‌عنوان یک رشته طولانی درنظر می‌گیرد.
ـ برای تولید لیستی از تمام کلمات موجود در کتاب، از متد رشته‌ای استفاده کنیدکه به طور پیش‌فرض یک رشته را در هر جایی که فضای خالی پیدا کند، تقسیم می‌کند.
ـ برای شمارش تعداد این کلمات از len() استفاده کنید.



```
exept5.py > ...
1   from pathlib import Path
2   path = Path('alice.txt')
3   try:
4       contents = path.read_text(encoding='utf-8')
5   except FileNotFoundError:
6       print(f"Sorry, the file {path} does not exist.")
7   else:
8       words = contents.split()
9       num_words = len(words)
10      print(f"The file {path} has about {num_words} words.")
```

```
The file alice.txt has about 190 words.
```

شکل ۴۵

کار با چند فایل

کار با چند فایل یکی از مهارت‌های مهم است، چون تقریباً هیچ برنامه واقعی فقط با یک فایل کار نمی‌کند.
کار با چند فایل یعنی برنامه بتواند بدون تکرار کد:
چندین فایل را پشت هم بخواند و روی همه آنها یک عملیات مشابه انجام دهد. پس لازم است یک تابع تعریف کنید و همچنین خطاها هم مدیریت و سپس نتایج ذخیره شوند.
برای انجام این کارکتاب‌های بیشتری را برای تحلیل اضافه کنید.
برای اینکه اجرای تحلیل روی چندین کتاب مختلف ساده‌تر و منظم‌تر شود بهتر است، بخش اصلی برنامه را به یک تابع به نام count_word منتقل کنید. اگر کدها را با برنامه قبل مقایسه کنید، متوجه می‌شوید تغییرات شامل اصلاح تو رفتگی و انتقال به بدنه تابع است.
خط ۲: تابع def count_words(path) یک ورودی به نام path می‌گیرد که مسیر یک فایل متنی است.
خط ۱۱: این خط از کد نام فایل‌ها به‌صورت رشته‌های متنی (String) در یک لیست به نامfilenames ذخیره می‌کند.

```
filenames = ['alice.txt','moby_dick.txt','little_women.txt']
```

خط ۱۲: در این خط ازیک حلقه for برای بررسی تک‌تک فایل‌های داخل لیست و به دست آوردن تعداد کلمات متن فایل استفاده شده است.
خط ۱۳: نام فایل به یک شیء از نوع Path تبدیل می‌شود.
خط ۱۴: تابع count_words برای آن فایل اجرا می‌شود.
و اگر فایلی وجود نداشته باشد، برنامه خطا نمی‌دهد و پیام مناسب نمایش می‌دهد.

multi_files.py > ...

```
1  from pathlib import Path
2  def count_words(path):
3      try:
4          contents = path.read_text(encoding='utf-8')
5      except FileNotFoundError:
6          print(f"Sorry, the file {path} does not exist.")
7      else:
8          words = contents.split()
9          num_words = len(words)
10         print(f"The file {path} has about {num_words} words.")
11 filenames = ['alice.txt','moby_dick.txt','little_women.txt','Mom_guest']
12 for filename in filenames:
13     path = Path(filename)
14     count_words(path)
```

```
The file alice.txt has about 190 words.
The file moby_dick.txt has about 99 words.
The file little_women.txt has about 244 words
Sorry, the file Mom guest does not exist.
```

شکل ۴۶



شکل ۴۷

نادیده گرفتن خطا (Failing Silently)

در مثال قبلی، به کاربر اطلاع داده می‌شد که یکی از فایل‌ها در دسترس نیست. با این وجود، در همه برنامه‌ها نیازی نیست تمام خطاهای رخ‌داده به کاربر نمایش داده شود.

در این حالت، خطا رخ می‌دهد، اما هیچ‌گونه پیامی به کاربر نمایش داده نمی‌شود و روند اجرای برنامه بدون وقفه ادامه می‌یابد.

برای پیاده‌سازی این حالت، از ساختار try-except استفاده می‌شود. با این تفاوت که در بخش except به‌صورت صریح به مفسر پایتون دستور داده می‌شود هیچ عملی انجام ندهد. پس دستورprint بعد از except را حذف کنید تاپیام خطا نمایش داده نشود و به جای آن دستور pass را اضافه کنید.

```
5        except FileNotFoundError:
6            pass
```

```
The file alice.txt has about 190 words.
The file moby_dick.txt has about 99 words.
The file little_women.txt has about 244 words.
```

شکل ۴۸

در زبان برنامه‌نویسی پایتون، دستور pass برای خالی نگه‌داشتن یک بلوک کد و انجام ندادن هیچ‌گونه عملی به کار می‌رود.

اگر فایل وجود نداشته باشد، خطا رخ می‌دهد، اما هیچ پیامی نمایش داده نمی‌شود، برنامه متوقف نمی‌شود و اجرای برنامه ادامه پیدا می‌کند.

فعالیت

همان‌طور که در برنامه تقسیم عدد بر صفر دیدید با یک پیام خطا کاربر از خطای تقسیم بر صفر آگاه می‌شد. این برنامه را به نحوی تغییر دهید که هرگاه مخرج یک تقسیم صفر بود بدون توقف برنامه، پیامی هم نمایش داده نشود.

ذخیره‌سازی داده‌ها

در بیشتربرنامه‌ها، مثل تنظیمات دلخواه کاربری برای یک بازی یا داده‌هایی برای نمایش نمودار، اطلاعاتی از کاربران دریافت می‌شود. صرف نظر از نوع آنها، اطلاعات وارد شده در ساختارهای داده‌ای مانند لیست‌ها و دیکشنری‌ها ذخیره می‌شوند. اما وقتی برنامه بسته می‌شود، این داده‌ها از بین می‌روند. برای حفظ اطلاعات برای دفعات بعد اجرای برنامه‌ها می‌توانید آنها را در فایل ذخیره کنید. یکی از ساده‌ترین روش‌ها برای این‌کار، استفاده از فایل‌های JSON است که کمک می‌کنند داده‌های پایتون به‌صورت متنی ذخیره و در اجرای بعدی برنامه دوباره خوانده‌شوند. این قالب مخصوص پایتون نیست و در زبان‌های برنامه‌نویسی دیگر نیز کاربرد دارد. در پایتون برای کار با JSON از ماژولjson استفاده می‌شود.



شکل ۴۹

json.dump() داده‌ها را به قالب JSON تبدیل می‌کند و json.loads() داده‌های ذخیره شده را دوباره به ساختارهای پایتون برمی‌گرداند.
فایل‌های JSON به دلیل سادگی، خوانایی و کاربردگسترده یکی از روش‌های مهم ذخیره‌سازی داده‌ها در برنامه‌نویسی هستند.

ذخیره یک لیست در فایل JSON

توضیح کد:

خط ۲ـ ماژول json وارد برنامه شده‌است.

خط ۳ـ لیستی از برندهای تلفن همراه برای ذخیره‌سازی ایجاد شده است.

خط ۵ـ با استفاده از تابع dump، داده‌های مورد نظر به قالب JSON تبدیل شده‌است. خروجی این تابع یک رشته متنی است که نمایانگر داده‌ها به‌صورت json است

writer_brand.py > ...

```
1  from pathlib import Path
2  import json
3  mobiles_brand = ["sumsung", "Apple", "SONY", "Huawei", "Nokia"]
4  path = Path('mobiles_brand.json')
5  contents = json.dumps(mobiles_brand)
6  path.write text(contents)
```

شکل ۵۰

خط ۶ـ این رشته متنی با استفاده از متد write-text داخل فایل ذخیره می‌شود.

این برنامه خروجی در صفحه نمایش ندارد، اما اگر فایل mobiles_brand.json را باز کنید خواهید دید که داده‌ها به شکلی ذخیره شده‌اند که بسیار شبیه به ساختار داده‌ها در پایتون است.

{} mobiles_brand.json > ...

```
1  ["sumsung", "Apple", "SONY", "Huawei", "Nokia"]
```

شکل ۵۱

خواندن داده‌ها از فایل JSON

برای برگرداندن اطلاعات ذخیره شده در فایل JSON به حافظه، لازم است برنامه‌ای دیگر نوشت.

reader_brand.py > ...

```
1  from pathlib import Path
2  import json
3  path = Path('mobiles_brand.json')
4  contents = path.read_text()
5  mobiles_brand = json.loads(contents)
6  print(mobiles brand)
```

```
['sumsung', 'Apple', 'SONY', 'Huawei', 'Nokia']
```

شکل ۵۲

توضیح کد:

خط ۲ــ ابتدا ماژول json وارد برنامه شده است.

خط ۳ــ سپس محتوای فایل JSON را خوانده و به‌صورت یک رشته متنی ذخیره شده است.

خط ۴ــ از آنجا که فایل JSON در اصل یک فایل متنی با قالب مشخص است، با استفاده از متد read_text() خوانده شده است.

خط ۵ــ در مرحله بعد با استفاده از تابع loads، این رشته متنی به ساختار داده‌ای مناسب در پایتون تبدیل شده است. در این برنامه، داده‌ها دوباره به‌صورت یک لیست از اعداد در حافظه ذخیره می‌شوند.

پس از این مرحله، می‌توانید از داده‌ها مانند سایر لیست‌های پایتون استفاده کنید. به این ترتیب اطلاعاتی که قبلاً در فایل JSON ذخیره شده بودند، در اجرای بعدی برنامه دوباره قابل استفاده خواهند بود.

مثال: برنامه‌ای بنویسید که مشخصات فنی یک گوشی هوشمند (شامل برند، مدل، مقدار رم و وضعیت گارانتی) را در قالب یک دیکشنری تعریف کند. سپس این برنامه باید:

با استفاده از کتابخانه json و کلاس Path از کتابخانه pathlib این دیکشنری را در فایلی به‌نام mobile_specs.json ذخیره کند.

در مرحله بعد، فایل را باز کرده، محتوا را خوانده و فقط «مدل» و «مقدار رم» گوشی را در خروجی چاپ کند.

فعالیت

خطوط کد برنامه را با کمک هنرآموز محترم خود توضیح دهید.

mobile_info.py > ...

```
1   import json
2   from pathlib import Path
3
4   mobile_info = {
5       "brand": "Samsung",
6       "model": "Galaxy S23",
7       "ram_gb": 12,
8       "has_warranty": True
9   }
10
11  path = Path('mobile_specs.json')
12
13  json_string = json.dumps(mobile_info, indent=4)
14  path.write_text(json_string)
15  print("Mobile specifications saved to JSON file.")
16
17  if path.exists():
18      content = path.read_text()
19      data = json.loads(content)
20
21      print(f"\nDevice Model: {data['model']}")
22      print(f"Memory Capacity: {data['ram_gb']}GB")
```

```
Mobile specifications saved to JSON file.

Device Model: Galaxy S23
Memory Capacity: 12GB
```

فعالیت

در کد نوشته شده قرار است اطلاعات آب‌وهوای یک شهر را در فایل ذخیره و سپس آن را بازخوانی کند. بخش‌هایی از کد که با علامت _____ مشخص شده است را با دستورات صحیح پر کنید.

```
import json
from pathlib import Path

weather_info = {
    "city": "Isfahan",
    "temp": 24,
    "status": "Sunny"
}

file_path = _____('weather.json')
# راهنمایی: ابتدا دیکشنری را به رشته تبدیل کنید و سپس در فایل بنویسید
json_data = json._____(weather_info, indent=4)
file_path._____(json_data)
print("Data saved successfully.")
if file_path._____():
    raw_text = file_path._____()
    # تبدیل متن به دیکشنری پایتون
    data = json._____(raw_text)
    print(f"City: {data['city']}")
    print(f"Current Temperature: {data['temp']}°C")
else:
    print("File not found!")
```

ارزشیابی شایستگی توسعه برنامه شیء گرا

استاندارد عملکرد کار: هنرجو باید بتواند با استفاده از شیء Path به‌صورت ایمن و دقیق عملیات ایجاد، خواندن، نوشتن و مدیریت فایل‌های متنی و JSON را انجام دهد و خطاها و استثناهای احتمالی را به‌درستی کنترل کند.

ردیف	مراحل کار	شرایط انجام کار (ابزار،مواد، تجهیزات، زمان، مکان و ...)	ابزارهای ارزشیابی	استاندارد (شاخص‌ها/داوری/نمره‌دهی)	نتایج ممکن	نمره کسب شده
۱	ایجاد و خواندن فایل با Path	رایانه با Python 3.x، ویرایشگر کد (VS Code/ ۱۵ دقیقه	مشاهده نصب توسط هنرجو، تمرین عملی بررسی کد و اجرای برنامه، مشاهده خروجی، پرسش کوتاه	ـ درک مفهوم فایل و شیء Path ـ ایجاد فایل و نوشتن محتوا ـ انجام کنجکاوی‌ها ـ انجام مراحل با روش‌های دیگر	۳	
				ـ درک مفهوم فایل و شیء Path ـ ایجاد فایل و نوشتن محتوا	۲	
				انجام ناقص مراحل	۱	
۲	پردازش و پیمایش خطوط فایل	همانند مرحله ۱	اجرای برنامه و بررسی حذف خطوط خالی و پیمایش صحیح	ـ خواندن خطوط فایل با splitlines ـ حذف خطوط خالی ـ انجام کنجکاوی‌ها ـ انجام مراحل با روش‌های دیگر	۳	
				ـ خواندن فایل بدون خطا ـ خواندن خطوط فایل با splitlines() ـ حذف خطوط خالی	۲	
				انجام ناقص مراحل	۱	
۳	استخراج و افزودن محتوا	همانند مرحله ۱	اجرای برنامه و بررسی استخراج و افزودن محتوا	ـ استخراج کلمات از خطوط ـ افزودن محتوا به فایل ـ انجام کنجکاوی‌ها ـ انجام مراحل با روش‌های دیگر	۳	
				ـ استخراج کلمات از خطوط ـ افزودن محتوا به فایل	۲	
				انجام ناقص مراحل	۱	
۴	جستجو و فیلتر محتوا	همانند مرحله ۱	اجرای برنامه و نمایش خطوط مطابق شرط	ـ جستجو در فایل ـ نوشتن تابع پردازش شرطی ـ انجام کنجکاوی‌ها ـ انجام مراحل با روش‌های دیگر	۳	
				ـ جستجو در فایل ـ نوشتن تابع پردازش شرطی	۲	
				انجام ناقص مراحل	۱	
۵	مدیریت استثنا و امنیت عملیات فایل	همانند مرحله ۱	اجرای برنامه، بررسی کنترل خطا و پیام‌های خطا	ـ درک try ـ except و else ـ مدیریت FileNotFoundError و Fail ـ Safe ـ انجام کنجکاوی ها ـ انجام مراحل با روش‌های دیگر	۳	
				ـ درک try ـ except و else ـ مدیریت FileNotFoundError و Fail ـ Safe	۲	
				انجام ناقص مراحل	۱	

۶	کار با JSON و Path	همانند مرحله ۱	اجرای برنامه، بررسی صحت ذخیره و بازیابی JSON	ـ تبدیل داده‌ها به JSON و ذخیره در فایل ـ خواندن فایل JSON و تبدیل مجدد به لیست ـ انجام کنجکاوی‌ها ـ انجام مراحل با روش‌های دیگر	۳	
				ـ تبدیل داده‌ها به JSON و ذخیره در فایل ـ خواندن فایل JSON و تبدیل مجدد به لیست	۲	
				انجام ناقص مراحل	۱	
شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش:				احراز همه شایستگی‌های غیرفنی	۲	
				عدم احراز هر یک از شایستگی‌های غیرفنی	۱	
ارزشیابی کار (شایستگی انجام کار)					بلی ☐	
					خیر ☐	

توضیحات:

۱ـ معیار شایستگی انجام کار:

ـ کسب حداقل نمره ۲ از مراحل ۱، ۲، ۳، ۵ و ۶ (مراحل بحرانی)

ـ کسب حداقل نمره ۲ از بخش شایستگی‌های غیرفنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش

ـ کسب حداقل میانگین ۲ از مراحل کار

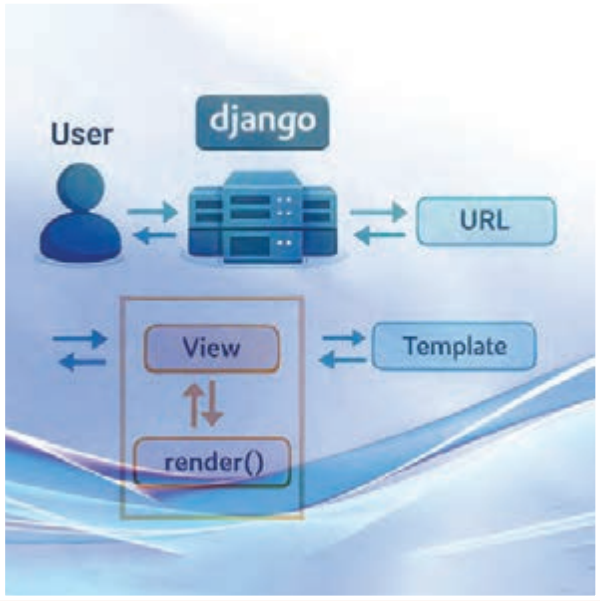
۲ـ تعریف سطوح شایستگی: صرف‌نظر از اینکه یک تکلیف کاری در چه سطحی از صلاحیت حرفه‌ای انجام می‌شود، در هر محیط کاری ممکن است انجام هر کار با کیفیت مشخصی مورد انتظار باشد. سطح شایستگی انجام کار، معیار اساسی ارزشیابی است.

مهارت (سطح سه): ماهر و قادر به آموزش و هدایت دیگران، توانایی برنامه‌ریزی و تحلیل، پاسخ‌گویی در برابر کارهای خود، سروکار داشتن با سطح وسیعی از کارها و فعالیت‌ها

تسلط (سطح چهار): خبرگی در انجام کار و آموزش دیگران، ایجاد نوآوری، سازگاری، عیب‌یابی، هدایت و راهنمایی دیگران.

پودمان پنجم

برنامه‌نویسی جنگو



تا حالا فکر کرده‌اید سایت‌هایی مثل اینستاگرام، دیوار یا کافه‌بازار چطور ساخته شده‌اند؟
وقتی وارد یک سایت می‌شوید و اطلاعاتی مثل وضعیت آب‌وهوا، پست‌ها، پیام‌ها یا قیمت‌ها را می‌بینید، پشت‌صحنهٔ این سایت‌ها چه اتفاقی می‌افتد؟
در پودمان‌های قبل، شما با مفهوم برنامه‌نویسی آشنا شده‌اید و کدهایی نوشته‌اید که روی رایانه خودتان اجرا می‌شوند.
اما در دنیای واقعی، برنامه‌نویسی فقط به اجرای برنامه روی یک سیستم محدود نمی‌شود.
وب‌سایت‌ها، فروشگاه‌های اینترنتی، شبکه‌های اجتماعی، سامانه‌های آموزشی و حتی بسیاری از اپلیکیشن‌های تلفن همراه، همگی به یک بخش بسیار مهم به‌نام backend متصل هستند. منظور از backend برنامه‌نویسی سمت سرور است جایی که منطق اصلی برنامه، پردازش داده‌ها، ارتباط با پایگاه‌داده و مدیریت کاربران انجام می‌شود.
در این پودمان با ابزاری قدرتمند با نام جنگو (Django) آشنا می‌شوید.

واحد یادگیری ۱
ایجاد وب اپلیکیشن

آیا تابه‌حال اندیشیده‌اید

- وب سایت با وب اپلیکیشن چه تفاوت‌هایی دارد؟
- چگونه می‌توانید یک وب اپلیکیشن بسازید؟
- یک وب اپلیکیشن چطور با کاربر تعامل می‌کند؟

در پایان از هنرجو انتظار می‌رود

۱ با مزایای جنگو آشنا شود.
۲ ساختار یک پروژه جنگو را درک کند.
۳ روند کار جنگو، جهت تعامل با کاربر را درک کرده و درباره آن استدلال نماید.
۴ مفهوم view، url و template در جنگو را درک کند.
۵ هنرجو بتواند به‌صورت عملی یک وب اپلیکیشن ساده را پیاده‌سازی کند.

استاندارد عملکرد

هنرجو باید با درک صحیحی که از ساختار جنگو کسب کرده است، بتواند یک وب اپلیکیشن ساده را در رایانه خود پیاده‌سازی کند.

جنگو چیست؟

جنگو (Django) یک framework منبع‌باز (open source) است که با زبان برنامه‌نویسی Python نوشته و برای ساخت و پیاده‌سازی بخش backend وب‌سایت‌ها (Website) و وب اپلیکیشن‌ها (Web Application) استفاده می‌شود.

جدول ۱ــ مقایسه وب‌سایت و وب اپلیکیشن

معیار	وب‌سایت (Website)	وب اپلیکیشن (Web Application)
هدف اصلی	اطلاع‌رسانی و نمایش محتوا	انجام عملیات و پردازش داده
تعامل کاربر	کم یا تقریباً یک‌طرفه	زیاد و دوطرفه
نیاز به لاگین	معمولاً ندارد	معمولاً دارد
پردازش سمت سرور	کم	زیاد
وابستگی به backend	کم یا متوسط	بسیار زیاد
تغییر محتوا توسط کاربر	معمولاً ندارد	دارد
مثال‌ها	سایت خبری، سایت شرکتی، بلاگ	پنل کاربری، فروشگاه آنلاین، سامانه آموزشی

مزایای جنگو

جنگو یک framework امن و سریع است که قابلیت‌های آمادهٔ زیادی را در اختیار برنامه‌نویس قرار می‌دهد. به همین دلیل، سرعت توسعهٔ نرم‌افزار در جنگو بسیار بالاست.

در پروژه‌های جنگو، با افزایش تعداد کاربران نیز می‌توان تنها با اعمال تغییراتی اندک، همچنان یک وب‌سایت سریع و امن داشت.

یکی دیگر از مزایای جنگو، وجود مستندات (راهنمای) جامع و همچنین یک جامعهٔ بزرگ و فعال از برنامه‌نویسان است. این یعنی اگر هنگام کار با جنگو با مشکلی مواجه شوید، با یک جست‌وجوی ساده در اینترنت می‌توانید راه‌حل‌ها و توضیحات زیادی از سوی برنامه‌نویسان دیگر پیدا کنید.

از دیگر مزیت‌های مهم جنگو، وجود فرصت‌های شغلی فراوان در حوزهٔ پایتون و framework جنگو است که آن را به گزینه‌ای مناسب برای یادگیری و ورود به بازار کار تبدیل می‌کند.

شرکت‌های بزرگی در دنیا و ایران از جنگو استفاده کرده‌اند.



شکل ۱ــ شرکت‌هایی که از جنگو استفاده کرده‌اند

کنجکاوی

در مورد شرکت‌های ذکر شده تحقیق کنید.

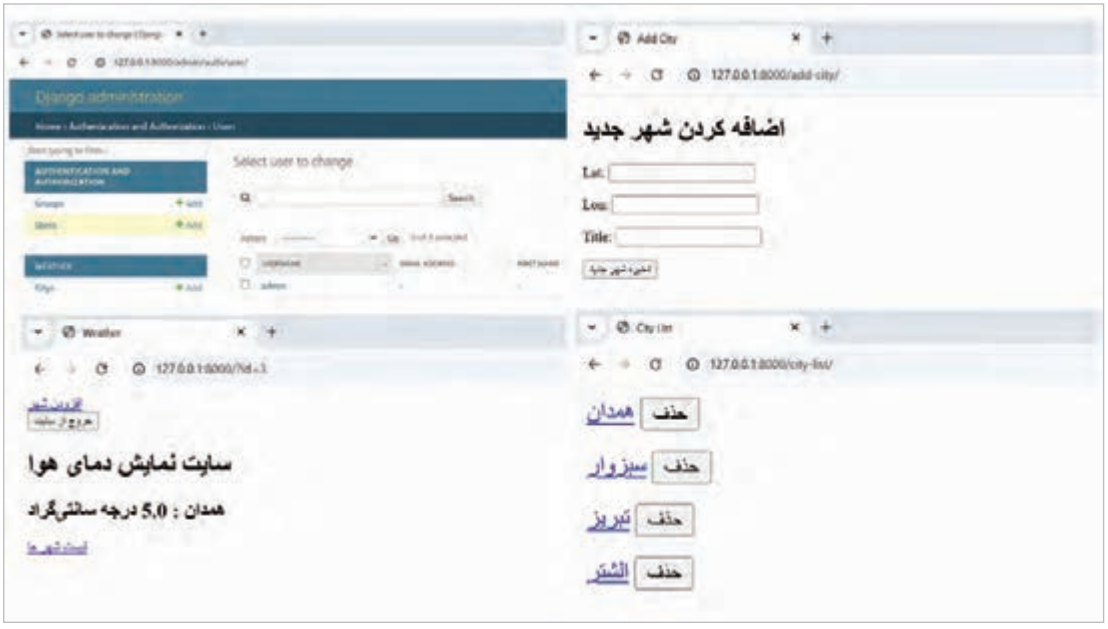
پروژه نهایی این کتاب

در پایان این کتاب، با استفاده از جنگو یک وب اپلیکیشن واقعی نمایش وضعیت آب‌وهوا طراحی خواهد شد (شکل ۲).

در این پروژه کاربر می‌تواند نام یک شهر را جهت نمایش وضعیت آب‌وهوا انتخاب و مشاهده نماید.

این وب اپلیکیشن اطلاعات آب‌وهوا را به‌صورت آنلاین دریافت کرده و نتیجه را به‌صورت لحظه‌ای نمایش می‌دهد.

پروژه به شما کمک می‌کند یاد بگیرید جنگو در عمل چگونه کار می‌کند و یک وب‌سایت چگونه ساخته می‌شود.

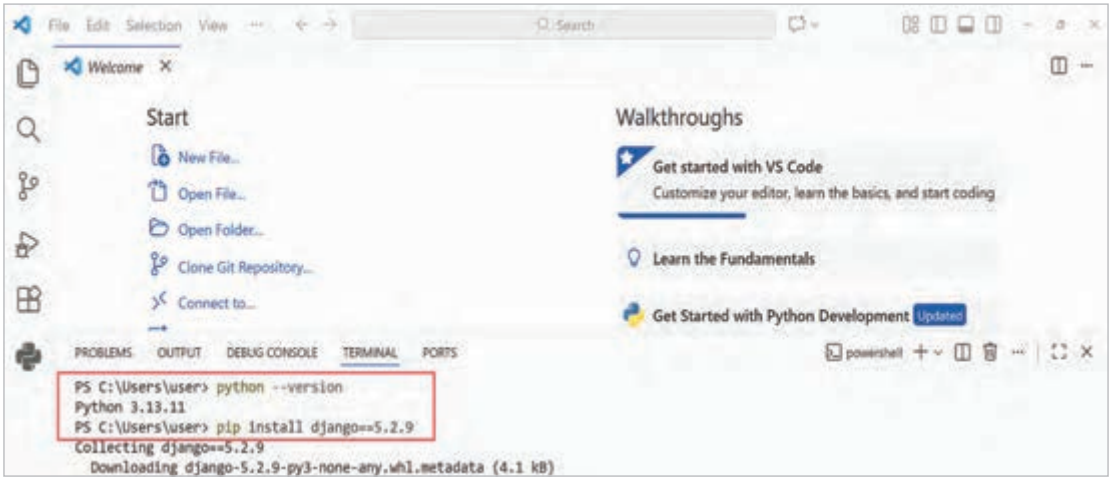


شکل ۲ـ بخش‌هایی از پروژه نهایی

نصب جنگو

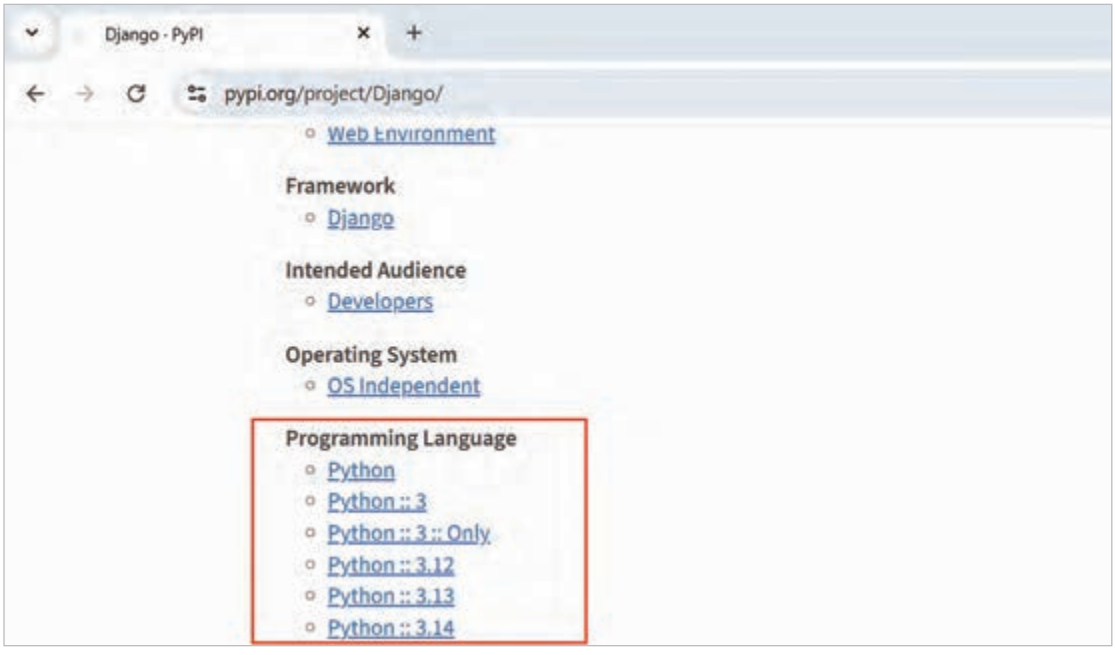
ابتدا مطمئن شوید آخرین نسخه پایتون روی رایانه شما نصب باشد و رایانه به اینترنت متصل باشد. برای این کار وارد برنامه Vscode شده اگر از قبل فایل یا پوشه‌ای باز بود با دستور Close Folder از منوی File آن را ببندید سپس از منوی terminal، یک ترمینال جدید باز کنید.

در ادامه همانند شکل ۳ درون قسمت ترمینال دستورات را به‌ترتیب اجرا نمایید.



شکل ۳ ـ نصب جنگو نسخه 5.2.9

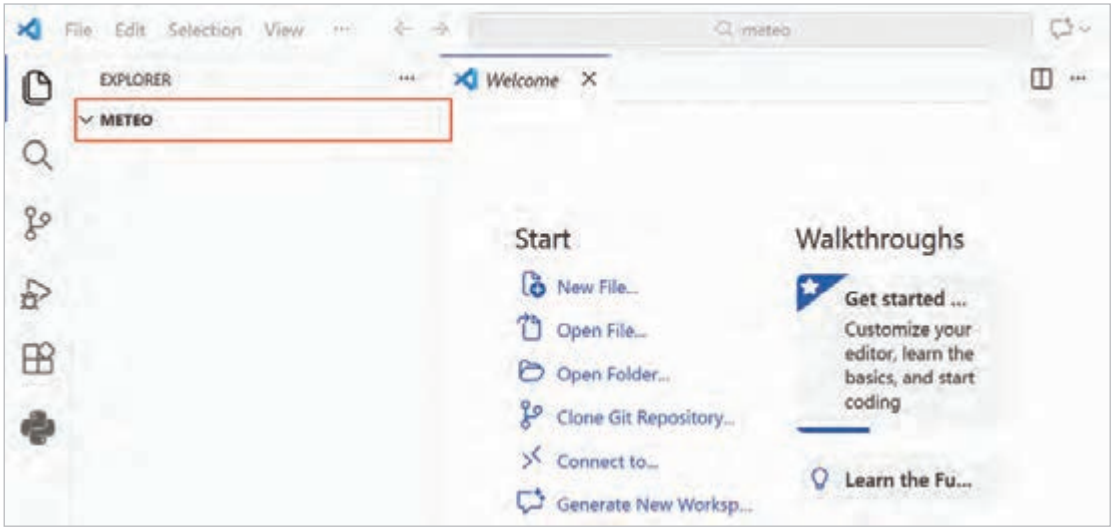
نمایش پیام Successfully installed Django-5.2.9 نشان‌دهنده نصب موفقیت‌آمیز جنگو است.
درصورتی‌که پیغام نصب موفقیت‌آمیز جنگو را مشاهده نکردید، از هنرآموز خود راهنمایی بگیرید.
می‌توانید آخرین نسخه منتشر شده جنگو را هم نصب کنید، فقط باید در سایت pypi.org و صفحه جنگو، نسخه پایتون سازگار را به‌دست آورید.
وارد آدرس https://pypi.org/project/Django شوید و در بخش Programming Language نسخه‌های پایتون پشتیبانی شده جنگو را مشاهده کنید (شکل ۴).



شکل ۴ـ نسخه‌های پایتون مناسب برای جنگو

ایجاد پروژه آب‌وهوا

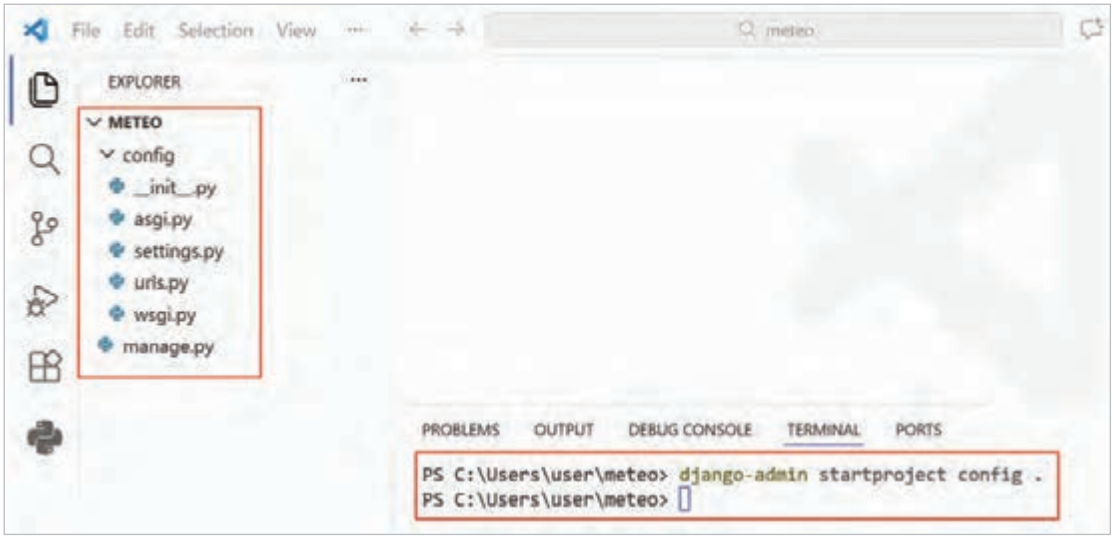
در Vscode از منو File گزینه Open Folder را کلیک کنید. سپس یک‌پوشه جدید با نام meteo در مسیر دلخواه ایجاد کرده و پوشه meteo ساخته شده را باز کنید (شکل ۵):



شکل ۵ــ باز کردن پوشه meteo

دوباره ترمینال را باز و دستور زیر را اجرا کنید (شکل ۶):

```
django-admin startproject config.
```



شکل ۶ــ ایجاد پروژه

دستور قبل، پوشه پروژه را با نام Config در مسیر جاری ایجاد می‌کند که حاوی فایل تنظیمات settings.py و فایل‌های اصلی پروژه است. حرف نقطه در انتهای دستور فوق به معنی مسیر جاری است.

آشنایی با برخی فایل‌های اصلی پروژه

فایل settings.py: تنظیمات اصلی پروژه در این فایل قرار می‌گیرند.

فایل urls.py: هرگاه کاربر وارد یک آدرس (URL) شود، این فایل تعیین می‌کند که سایت کدام صفحه یا بخش را نمایش دهد.

کنجکاوی

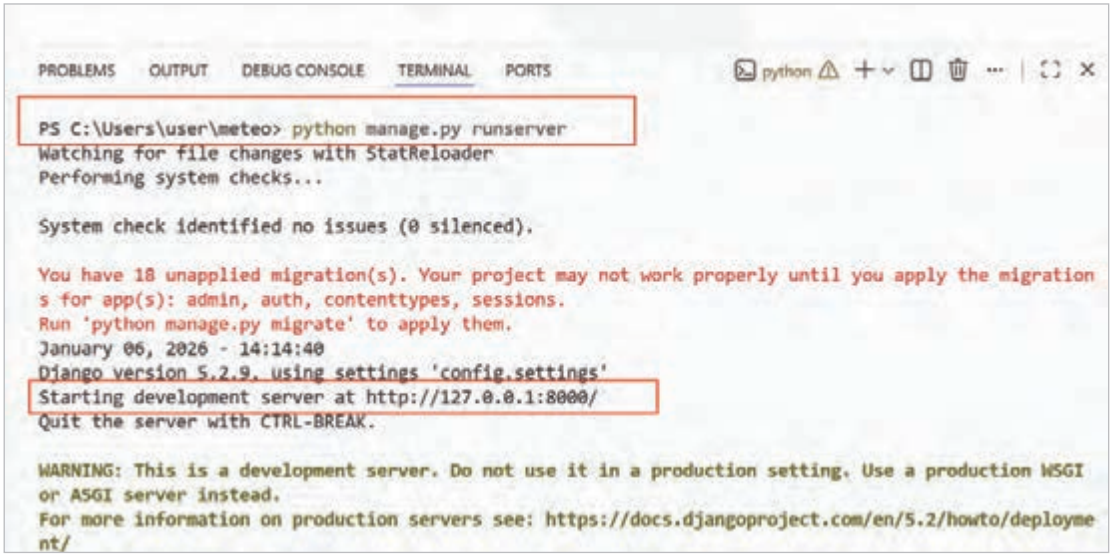
در مورد کاربرد فایل‌های دیگر موجود در پروژه تحقیق کنید.

اجرای پروژه جنگو

دستور زیر را در ترمینال اجرا نمایید.

```
python manage.py runserver
```

دستور runserver یک وب سرور (web server) مجازی کوچک برای پروژه جنگو راه‌اندازی می‌کند، این وب سرور فقط مخصوص برنامه‌نویس جنگو است تا بتواند پروژه جنگو را در رایانه خودش اجرا کند (شکل ۷).



شکل ۷ـ راه‌اندازی موفق وب سرور جنگو

بعد از اجرای دستور، یک فایل با نام db.sqlite3 کنار فایل manage.py ساخته می‌شود. این فایل، پایگاه‌داده پیش‌فرض جنگو است. جنگو از این فایل برای مدیریت پایگاه‌داده و جداول استفاده می‌کند.
همان‌طوری که در شکل ۷ مشخص شده است، می‌توانید سایت خود را در آدرس http://127.0.0.1:8000 مشاهده کنید.
با تایپ آدرس فوق درون نوار آدرس مرورگر، صفحه اصلی سایت را مشابه شکل ۸ مشاهده می‌کنید. ممکن است به‌دلیل نسخه‌های متفاوت جنگو تصویر متفاوتی مشاهده کنید.



شکل ۸ــ اجرای موفقیت‌آمیز سایت

آدرس ip که جنگو روی آن اجرا می‌شود، 127.0.0.1 است. این آدرس به آدرس local معروف است. این ip به رایانه جاری اشاره می‌کند، یعنی همین رایانه‌ای که سرور جنگو را روی آن اجرا کرده‌اید. اگر رایانه شما در یک شبکه قرار داشته باشد، رایانه‌های دیگر نمی‌توانند با این آدرس، به رایانه شما متصل شوند.

کنجکاوی

بررسی کنید، برای پروژه‌هایی که در بستر اینترنت در حال استفاده هستند، از چه وب‌سرورها و چه پایگاه داده‌هایی استفاده می‌شود؟

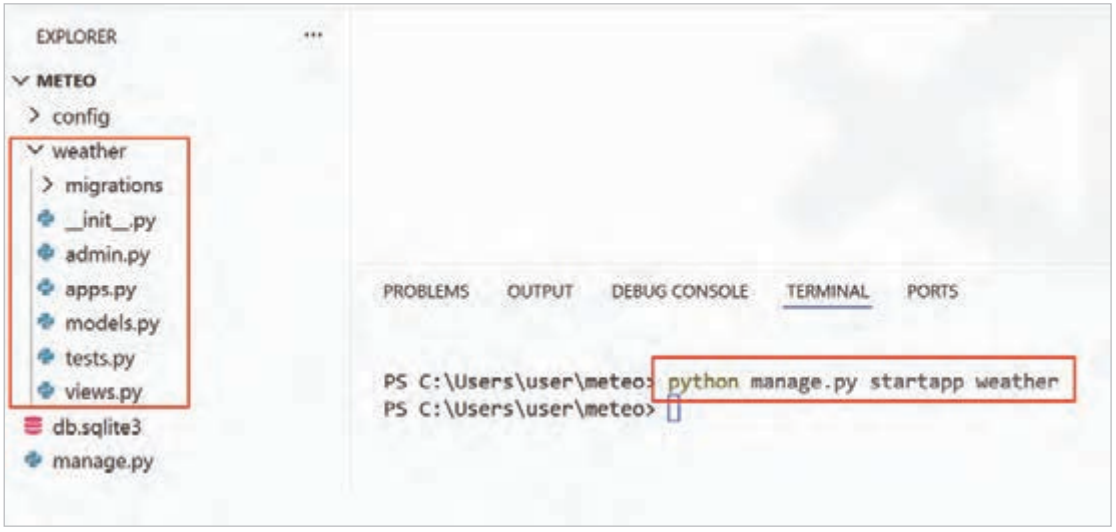
ساخت اولین app در جنگو

یک پروژه جنگو می‌تواند شامل یک یا چندین app (اپلیکیشن) باشد. پروژه‌ای که با دستور startproject ایجاد کردید، هنوز app ندارد. در جنگو می‌توانید برای وظایف مستقل یک app ایجاد کنید.
فرض کنید پروژه شما یک کارخانه در دنیای واقعی است، واحدهای تولید کارخانه هرکدام می‌توانند مانند یک app باشند. به‌عنوان‌مثال، واحدی که وظیفه رنگ‌کردن خودرو را برعهده دارد، مستقل از واحدهای دیگر می‌تواند عمل کند. کافی است یک خودرو به این واحد تحویل شود و رنگ درخواستی مشخص شود. این واحد کار خود را انجام داده و خودرو را رنگ شده تحویل می‌دهد.
هر app وظیفه خاصی انجام می‌دهد و باید به‌صورتی ایجاد گردد که قابلیت استفاده مجدد داشته باشد، یعنی یک app را بتوان در پروژه دیگری استفاده کرد و همان وظیفه را به‌خوبی انجام دهد.

برای ساخت app از دستور startapp استفاده می‌شود، ابتدا سرور جنگو را متوقف کرده (با ctrl + c) و سپس دستور زیر را اجرا کنید.

```
python manage.py startapp weather
```

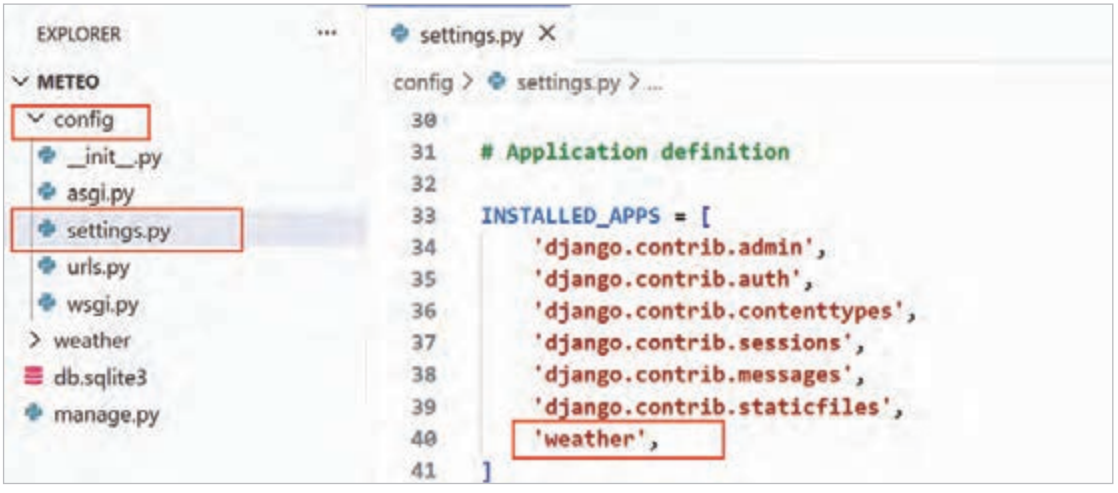
یک پوشه با نام weather ساخته می‌شود. پوشه weather در حقیقت یک app جنگویی است که وظیفه نمایش وضعیت آب‌وهوای شهر موردنظر را برعهده دارد.
پس از ساخت weather app ساختار پروژه همانند شکل ۹ خواهد بود.



شکل ۹ــ محتوای weather app

معرفی app ایجاد شده به جنگو در settings.py

در این مرحله، لازم است app ساخته شده به جنگو معرفی شود. برای این کار فایل settings.py را باز کنید و weather را مطابق شکل ۱۰ به انتهای لیست INSTALLED_APPS اضافه کنید.



شکل ۱۰ــ اضافه‌کردن weather به installed_apps

ساخت اولین view

هر view که در فایل views.py است، برحسب وظیفه‌اش، با کاربر تعامل می‌کند، پس برای نمایش دمای یک شهر، یک view باید ساخته شود.

فایل‌های View در جنگو مسئول پاسخ‌گویی به درخواست کاربران هستند، یعنی Request کاربر را بعد از مسیریابی می‌گیرند و به آن درخواست یک پاسخ درست می‌دهند.

مرحله اول: فایل views.py را باز کنید. یک تابع با نام show_temp_view همانند شکل ۱۱ اضافه کنید:



شکل ۱۱ـ ایجاد view با نام show_temp_view

توضیحات کدها:

خط ۱: کلاس HttpResponse از ماژول http وارد (import) شده است.

خط ۳: یک تابع با نام show_temp_view ایجاد شده است. هر view در جنگو، تابعی است که باید پارامتر request را داشته باشد.

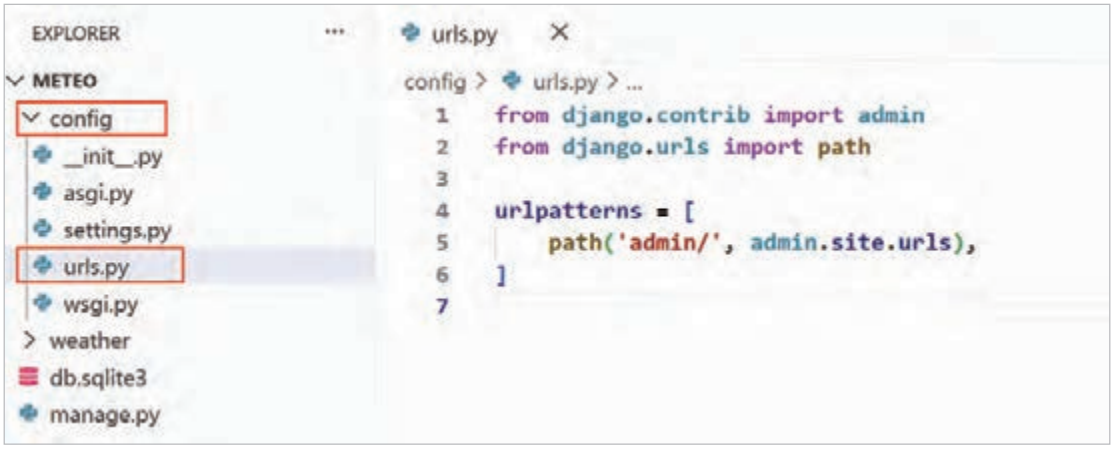
خط ۴: یک object از نوع HttpResponse ایجاد شده است. HttpResponse محتوای نهاییِ ارسال‌شده به مرورگر را در خود نگهداری می‌کند و مشخص می‌کند کاربر چه چیزی از سرور دریافت کند.

تابع show_temp_view در فایل views.py فقط یک متن ساده، نمایش خواهد داد.

فایل را ذخیره نمایید و سرور جنگو را با python manage.py runserver راه‌اندازی کنید.

آدرس صفحه اصلی را وارد کنید، باز هم صفحه خوشامدگویی را می‌بینید. پس لازم است تغییراتی اعمال کنید تا در صفحه اصلی سایت، به‌جای خوشامدگویی متن «همدان ۲۷ درجه سانتی‌گراد» نمایش داده شود.

فایل urls.py در پوشه config را باز کنید (شکل ۱۲).



شکل ۱۲ــ فایل urls.py

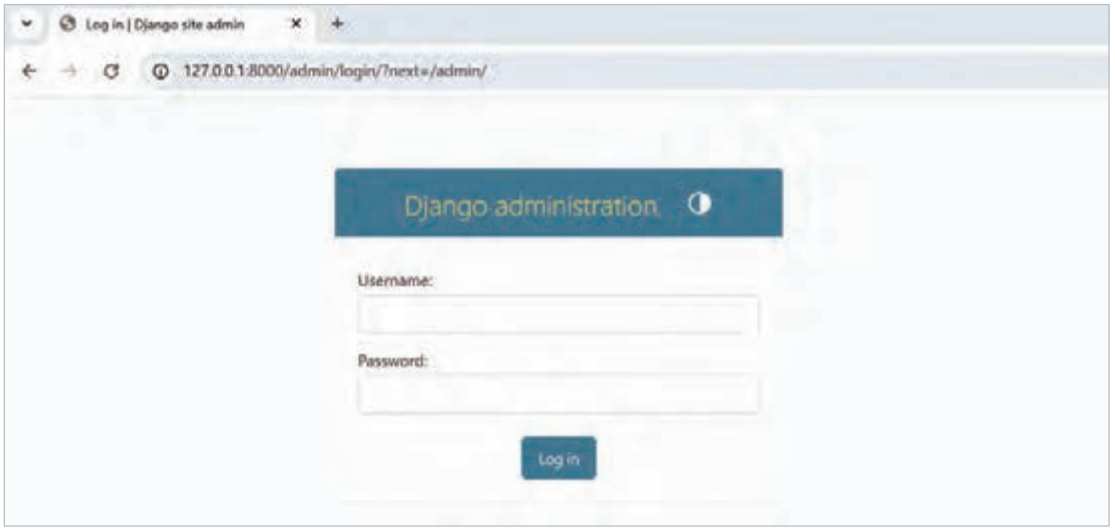
بعد از ایجاد پروژه، فایل urls.py نیز کنار settings.py ایجاد می‌شود. هرگاه کاربر وارد یک آدرس (URL) شود، این فایل تعیین می‌کند که سایت کدام صفحه یا بخش را نمایش دهد.

توضیحات کدها:

خط ۴: جنگو از یک لیست با نام urlpatterns برای مدیریت آدرس صفحات سایت استفاده می‌کند.

خط ۵: هر عضو از urlpattrns یک path است که معادل آدرس یک صفحه در سایت است. به‌صورت پیش‌فرض یک path برای صفحه پنل ادمین جنگو با مقدار admin/ وجود دارد که معادل آدرس http://127.0.0.1:8000/admin در مرورگر است.

آدرس http://127.0.0.1:8000/admin را در مرورگر باز کنید. صفحه ورود به پنل ادمین همانند شکل ۱۳ از شما نام کاربری و رمز عبور را درخواست می‌کند. در ادامه روش ساخت نام کاربری و رمز عبور را خواهید آموخت.



شکل ۱۳ــ صفحه ورود به پنل ادمین

مرحله دوم: دوباره وارد فایل urls.py شده و تغییراتی همانند شکل ۱۴ در فایل اعمال کنید.



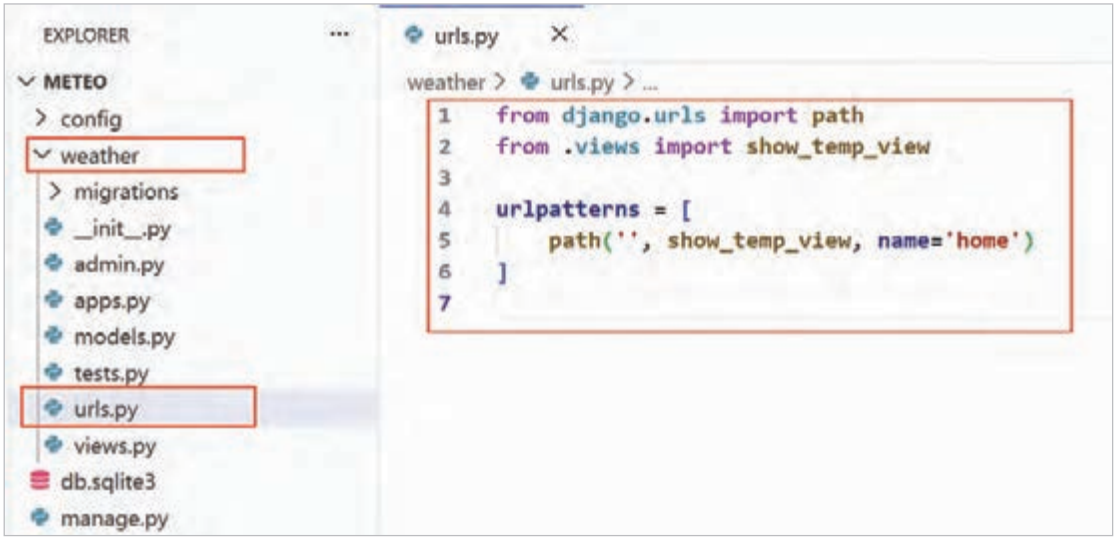
شکل ۱۴ــ اضافه‌کردن path جدید

توضیحات کدها:

خط ۲: ماژول include وارد (import) شده است.

خط ۶: یک path جدید اضافه شده است. اولین آرگومان تابع path در این دستور فقط یک‌رشته خالی است و به این معناست که اگر در آدرس (URL) رسیده به سرور جنگو، جلوی آدرس سایت هیچ عبارتی درج نشده بود، با استفاده از ماژول include به سراغ فایل urls.py در weather app بروید و کار را از آنجا ادامه دهید. در واقع این خط، درخواست‌های صفحهٔ اصلی را به weather app ارسال می‌کند.

مرحله سوم: وارد پوشه weather شده و برای weather app یک urls.py همانند شکل ۱۵ ایجاد کنید. هر app می‌تواند یک urls.py اختصاصی داشته باشد.



شکل ۱۵ــ ایجاد فایل urls.py برای app

توضیحات کدها:

خط ۱: تابع path وارد (import) شده است.

خط ۲: تابع show_temp_view از views.py وارد (import) شده است.

خط ۵: یک path اضافه شده است. این path به این معنی است که اگر کاربر صفحه اصلی سایت را باز کرد، show_temp_view از درون فایل views.py اجرا شود. به این path پارامتر name اضافه شده است.

در جنگو هر آدرس (URL) می‌تواند یک نام داشته باشد تا در هر جایی دیگر، راحت‌تر به آن ارجاع داده شود. این اسم هیچ تأثیری روی خود آدرس URL ندارد و فقط برای ارجاع‌دادن استفاده می‌شود.

پارامتر name در مسیرهای جنگو مانند یک اسم مستعار برای آن آدرس است که به آن URL name هم می‌گویند.

فایل را ذخیره نمایید و با runserver سرور جنگو را راه‌اندازی کنید.

صفحه اصلی سایت را در مرورگر باز کنید.

تبریک، شما اولین view جنگو خود را با موفقیت ایجاد کردید (شکل ۱۶).



شکل ۱۶ــ خروجی اولین view جنگو

فعالیت

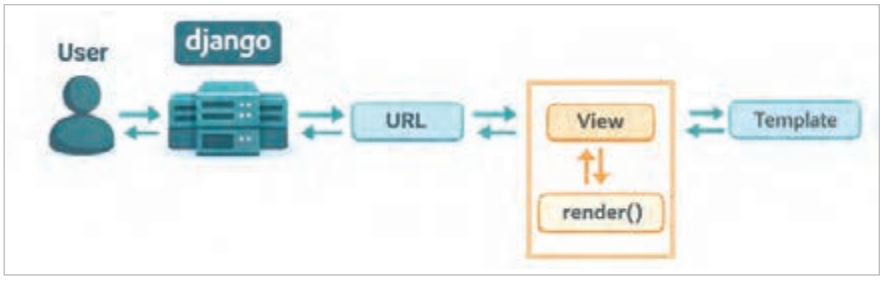
با تکمیل view و فایل urls.py اطلاعات دمای تهران را نمایش دهید.

آشنایی با تابع render

در جنگو برای نمایش بهتر view از فایل‌های html استفاده می‌شود. هر app در جنگو، می‌تواند فایل‌های html مخصوص به خود را به‌صورت جداگانه داشته باشد، برای انجام این کار یک‌پوشه به اسم templates در app ایجاد کنید و فایل‌های html را در آن ذخیره کنید. به این فایل‌های html در جنگو template گفته می‌شود.

شکل ۱۷ را مشاهده کنید، مسیر طی شدهٔ request (درخواست وب) در جنگو قابل‌مشاهده است.

طبق تصویر، درخواست ارسالی کاربر، ابتدا به سرور جنگو منتقل شده، سپس با استفاده از url به view مناسب ارسال می‌شود، هر view با کمک فایل html یا همان template اطلاعات قابل نمایش را آماده کرده و برای کاربر ارسال می‌کند.

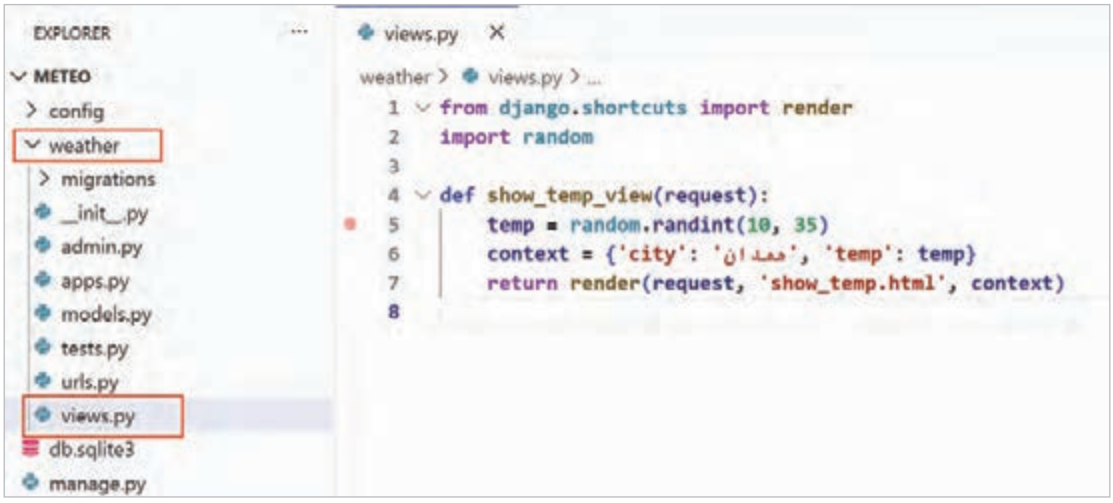


شکل ۱۷ــ مسیر طی شده یک request در جنگو

همان‌طورکه در شکل ۱۷ اشاره شده است، تابع render در view می‌تواند اطلاعات را به فایل html ارسال کند تا به کمک قابلیت‌های جنگو، اطلاعات ارسالی به فایلِ html با آن ترکیب شده و به کاربر نمایش داده شود.

در ادامه، در دو مرحله می‌توانید یک عدد تصادفی را به‌عنوان دمای هوای شهر همدان، با کمک تابع render به یک فایل html ارسال کرده و به کاربر نمایش دهید.

مرحله اول: فایل views.py را باز کنید و show_temp_view را همانند شکل ۱۸ تغییر دهید.



شکل ۱۸ـ استفاده از تابع render

توضیحات کدها:

خط ۱: تابع render از ماژول shortcuts وارد (import) شده است.

خط ۲: ماژول random وارد (import) شده است.

خط ۵: یک عدد تصادفی به‌عنوان دمای هوا برای شهر همدان انتخاب می‌شود، این اطلاعات در صفحه اصلی سایت نمایش داده خواهد شد.

خط ۶: یک دیکشنری شامل نام شهر (city) و دمای شهر (temp) ایجاد شده و در متغیری با نام context ذخیره شده است.

خط ۷: از تابع render استفاده شده است. برای این تابع سه آرگومان ارسال شده است. آرگومان اول request است که از ورودی view دریافت شده است. آرگومان دوم نام فایل html موجود در پوشه templates و آرگومان سوم یک دیکشنری است که برای html ارسال شده است.

مرحله دوم: یک‌پوشه با نام templates در weather app بسازید و یک فایل html با نام show_temp.html درون آن ایجاد کنید.

فایل show_temp.html و هر فایل html دیگری که در این پوشه قرار بگیرد، به‌اصطلاح یک template است.

فایل show_temp.html و محتوای آن را همانند شکل ۱۹ ایجاد کنید.



شکل ۱۹ــ محتویات show_temp.html

توضیحات کدها:

خط ۱۰: از {{city}} و {{temp}} جهت‌نمایش نام شهر و دمای شهر استفاده شده است. علامت {{}} جهت‌نمایش متغیرها در template به‌کار می‌رود.

این متغیرها، کلیدهای دیکشنری context هستند که تابع render آن را برای show_temp.html ارسال کرده است.

همان‌طورکه مشخص است، می‌توانید در template از این متغیرها، درون تگ‌های html استفاده کنید.

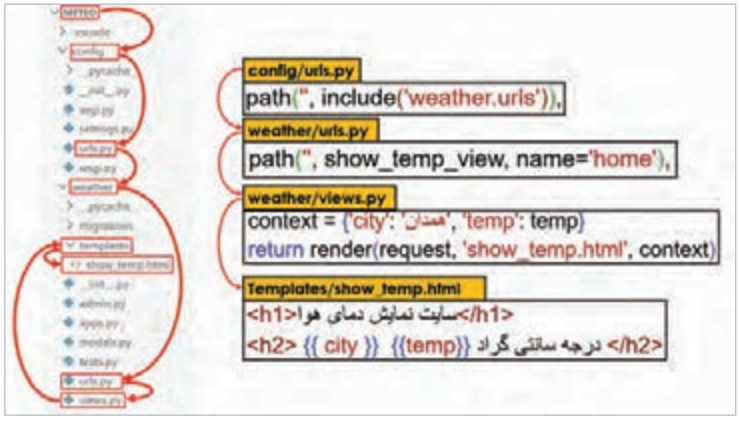
فایل‌ها را ذخیره کنید. سپس وارد terminal شده و با کلیدهای ترکیبی ctrl + c وب سرور جنگو را متوقف کنید و دوباره با دستور runserver آن را راه‌اندازی کنید.

اکنون صفحه اصلی سایت را refresh کنید. تغییرات جدید را مشاهده خواهید کرد (شکل ۲۰).



شکل ۲۰ــ مشاهده تغییرات جدید

روند طی شده نمایش دمای هوا، از لحظه ورود درخواست کاربر به سرور جنگو تا نمایش اطلاعات با template، در شکل ۲۱ شبیه‌سازی شده است.



شکل ۲۱ــ روند نمایش دمای هوا

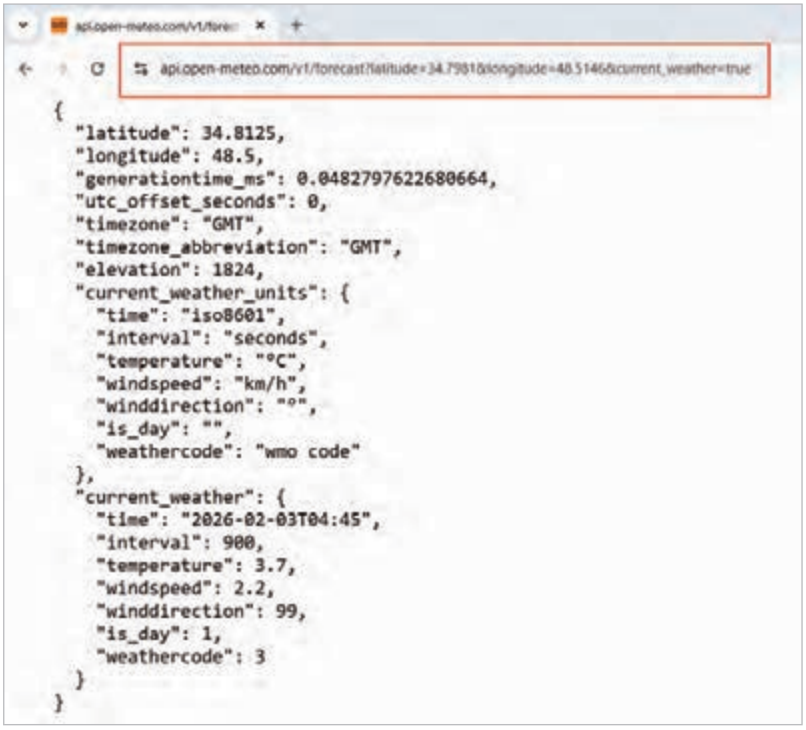
فعالیت

با استفاده از تابع render و ایجاد یک template مناسب برای view، دمای هوای شهر خودتان را با اعداد تصادفی نمایش دهید.

نمایش دمای هوای واقعی

جهت‌نمایش دمای هوای لحظه‌ای شهرها می‌توانید از سایت‌هایی مثل https://open-meteo.com استفاده کنید. این سایت ابزاری در اختیار برنامه‌نویس قرار می‌دهد تا بتواند دمای هوای شهرهای مختلف را نمایش دهد. سایت open-meteo یک API رایگان دارد که با دریافت طول و عرض جغرافیایی، اطلاعات آب‌وهوای منطقه موردنظر را در قالب JSON ارائه می‌دهد.

به‌عنوان‌مثال، اگر آدرس زیر را برای دریافت وضعیت آب‌وهوای شهر همدان در مرورگر باز کنید، نتیجه با قالب JSON نمایش داده خواهد شد (شکل ۲۲).



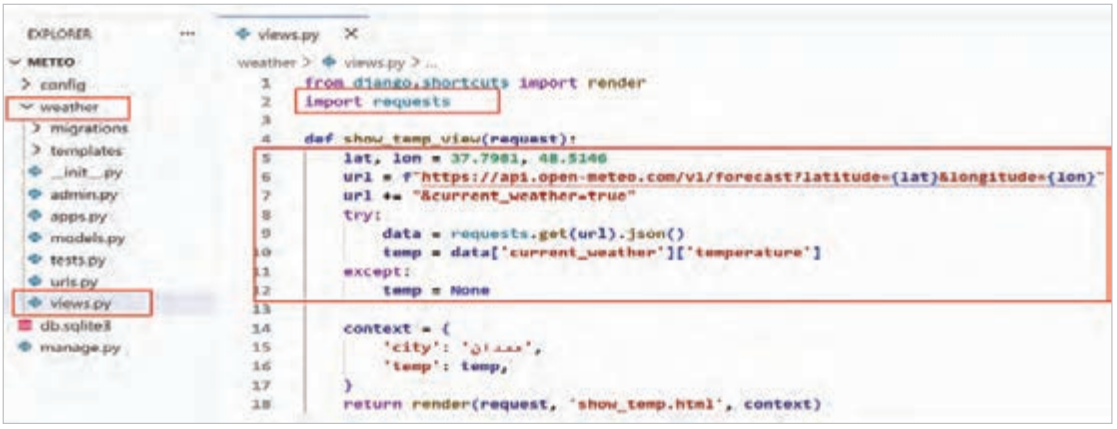
شکل ۲۲ــ دریافت دمای شهر همدان از API آنلاین

https://api.open-meteo.com/v1/forecast?latitude=34.7981&longitude=48.5146&current-weather=true

API مانند یک پل بین دو برنامه است. یعنی زمانی که یک برنامه (مثلاً پروژه جنگو) به اینترنت وصل می‌شود، از طریق API می‌تواند از یک سایت دیگر داده بگیرد.

در این پروژه، با استفاده از API سرویس open-meteo اطلاعات آب و هوایی دریافت می‌شود. open-meteo این اطلاعات را به‌صورت داده‌های منظم در قالب (Json) برمی‌گرداند، سپس پروژه جنگو آنها را نمایش می‌دهد.

ارسال مقدار latitude و longitude برای دریافت اطلاعات آب‌وهوا اجباری است.
اکنون جهت‌نمایش دمای فعلی شهر همدان تابع show_temp_view را از داخل فایل views.py همانند شکل ۲۳ تغییر دهید.



شکل ۲۳ـ تغییر در view جهت دریافت اطلاعات از API

توضیحات کدها:

خط ۲: کتابخانه requests وارد (import) شده است. این کتابخانه باید از قبل نصب شده باشد. با دستور (pip install requests)

خط ۵: مقادیر latitude و longitude همان طول و عرض جغرافیایی هستند که برای شهر همدان مشخص شده و در متغیرهای lat و lon ذخیره شده‌اند.

خط ۶ و ۷: متن درخواست به‌صورت get و با استفاده از lat و lon آماده شده است.

خط ۸: با استفاده از try خطای احتمالی مدیریت شده است، به‌عنوان‌مثال اگر اینترنت وصل نباشد، except در خط ۱۱ مقدار temp را برابر None خواهد کرد و سایت با مشکل مواجه نمی‌شود.

خط ۹: درخواست به سرور open-meteo.com ارسال شده و نتیجه با قالب json در دیکشنری data ذخیره شده است.

خط ۱۰: مقدار temperature از دیکشنری data استخراج شده و در متغیر temp ذخیره شده است.

در فایل show_temp.html تغییراتی برای مدیریت خطا همانند شکل ۲۴ اعمال کنید.



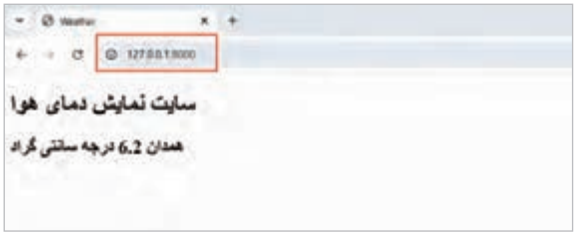
شکل ۲۴ـ تغییرات در template جهت مدیریت خطای احتمالی

توضیحات کدها:

خط ۱۰: در template از دستور شرطی if استفاده شده است. درصورتی‌که temp مقدار داشته باشد و None نباشد، دمای شهر نمایش داده می‌شود.

خط ۱۳: در صورت بروز خطا، پیغامی مناسب به کاربر نمایش داده می‌شود.

اکنون فایل‌ها را ذخیره کرده و پس از اطمینان از اتصال به اینترنت، صفحه اصلی سایت را refresh کنید. خروجی صفحه اصلی سایت، همانند شکل ۲۵ خواهد بود (واضح است که دمای هوا برای شما ممکن است متفاوت نمایش داده شود و البته سرعت باز شدن صفحه اصلی سایت، به‌سرعت اینترنت شما ارتباط دارد، چون در کدها به open-meteo.com متصل شده‌اید.)



شکل ۲۵ـ نمایش دمای همدان به‌صورت آنلاین

درصورتی‌که ارتباط با سایت برقرار نشود، خروجی همانند شکل ۲۶ خواهد بود.



شکل ۲۶ـ نمایش خطای مناسب به کاربر

فعالیت

با استفاده از سایت open-meteo.com دمای هوای شهر خود را در صفحه اول سایت نمایش دهید.

ارزشیابی شایستگی ایجاد وب اپلیکیشن

استاندارد عملکرد کار: هنرجو باید با درک صحیحی که از ساختار جنگو کسب کرده است، بتواند یک وب اپلیکیشن ساده را در رایانه خود پیاده‌سازی کند.

ردیف	مراحل کار	شرایط انجام کار (ابزار،مواد، تجهیزات، زمان، مکان و ...)	ابزارهای ارزشیابی	استاندارد (شاخص‌ها/داوری/نمره‌دهی)	نتایج ممکن	نمره کسب شده
۱	آشنایی با جنگو و پروژه نهایی	ابزار: رایانه زمان: ۱۰ دقیقه؛ مکان: کارگاه	پرسش شفاهی، آزمون کتبی کوتاه، اجرای عملی	۱ـ توضیح مفهوم و مزایای جنگو ۲ـ تحلیل پروژه آب‌وهوا و اجزای آن ۳ـ مقایسه جنگو با پایتون خالص ـ انجام کنجکاوی	۳	
				۱ـ توضیح مفهوم و مزایای جنگو ۲ـ تحلیل پروژه آب‌وهوا و اجزای آن۳ـ مقایسه جنگو با پایتون خالص	۲	
				انجام ناقص هر یک از مراحل	۱	
۲	نصب جنگو و راه‌اندازی محیط توسعه	ابزار: رایانه سیستم‌عامل نصب‌شده، Python،Django اینترنت (یا بسته آفلاین)، دسترسی نصب؛ زمان: ۱۵ دقیقه؛	آزمون کتبی کوتاه، اجرای عملی	۱ـ اجرای دستور pip install django بدون خطا ۲ـ بررسی نسخه جنگو ۳ـ آماده‌سازی پوشه پروژه ۴ـ انجام کارها با روشی دیگر ۵ـ انجام کنجکاوی‌ها	۳	
				۱ـ اجرای دستور pip install django بدون خطا ۲ـ بررسی نسخه جنگو ۳ـ آماده‌سازی پوشه پروژه	۲	
				انجام ناقص هر یک از مراحل	۱	
۳	ایجاد و اجرای اولین پروژه جنگو	ابزار: رایانه، Django نصب‌شده، ترمینال، مرورگر وب، زمان: ۱۵ دقیقه؛ مکان: کارگاه	آزمون کتبی کوتاه، اجرای عملی	۱ـ استفاده از دستور startproject و بررسی ساختار ۲ـ مشاهده صفحه پیش‌فرض جنگو ۳ـ تشریح فایل‌های اصلی پروژه ۴ـ انجام کارها با روشی دیگر ۵ـ انجام کنجکاوی‌ها	۳	
				۱ـ استفاده از دستور startproject و بررسی ساختار۲ـ مشاهده صفحه پیش‌فرض جنگو۳ـ تشریح فایل‌های اصلی پروژه	۲	
				انجام ناقص هر یک از مراحل	۱	
۴	ساخت و معرفی اولین app در پروژه	ابزار: رایانه، ترمینال، Django، زمان: ۱۵ دقیقه؛ مکان: کارگاه	آزمون کتبی کوتاه، اجرای عملی	۱ـ اجرای دستور startapp weather2 ۲ـ تغییر در settings.py و آزمایش اجرا ۳ـ بیان ارتباط پروژه با appها ۴ـ انجام کارها با روشی دیگر۵ـ انجام کنجکاوی‌ها	۳	
				۱ـ اجرای دستور startapp weather2 ۲ـ تغییر در settings.py آزمایش اجرا ۳ـ بیان ارتباط پروژه با appها	۲	
				انجام ناقص هر یک از مراحل	۱	
۵	ساخت اولین view و آشنایی با تابع render	ابزار: ویرایشگر کد (VS Code) ترمینال، مرورگر؛ زمان: ۲۵ دقیقه؛	آزمون کتبی کوتاه، اجرای عملی	۱ـ ایجاد تابع در views.py2 ۲ـ اتصال view به قالب با render ۳ـ ویرایش فایل urls.py4 ۴ـ انجام کارها با روشی دیگر۵ـ انجام کنجکاوی‌ها	۳	
				۱ـ ایجاد تابع در views.py2 ۲ـ اتصال view به قالب با render ۳ـ ویرایش فایل urls.py	۲	
				انجام ناقص هر یک از مراحل	۱	
۶	نمایش دمای واقعی در وب‌اپلیکیشن	ابزار: ویرایشگر کد(VS Code) و ترمینال، مرورگر؛ زمان: ۳۵ دقیقه	آزمون کتبی کوتاه، اجرای عملی	۱ـ اجرای view با context داده‌های دما ۲ـ مشاهده و تحلیل خروجی در مرورگر۳ـ بررسی عملکرد کلی وب‌اپلیکیشن۴ـ انجام کارها با روشی دیگر ۵ـ انجام کنجکاوی ها	۳	
				۱ـ اجرای view با context داده‌های دما۲ـ مشاهده و تحلیل خروجی در مرورگر۳ـ بررسی عملکرد کلی وب‌اپلیکیشن	۲	
				انجام ناقص هر یک از مراحل	۱	

شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش:			احراز همه شایستگی‌های غیرفنی	۲	
			عدم احراز هر یک از شایستگی‌های غیرفنی	۱	
ارزشیابی کار (شایستگی انجام کار)				بلی ☐	
				خیر ☐	

توضیحات:

۱ـ معیار شایستگی انجام کار:

ـ کسب حداقل نمره ۲ از مراحل ۱، ۲، ۳، ۴، ۵ و ۶ (مراحل بحرانی)

ـ کسب حداقل نمره ۲ از بخش شایستگی‌های غیرفنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش

ـ کسب حداقل میانگین ۲ از مراحل کار

۲ـ تعریف سطوح شایستگی: صرف‌نظر از اینکه یک تکلیف کاری در چه سطحی از صلاحیت حرفه‌ای انجام می‌شود، در هر محیط کاری ممکن است انجام هر کار با کیفیت مشخصی مورد انتظار باشد. سطح شایستگی انجام کار، معیار اساسی ارزشیابی است.

مهارت (سطح سه): ماهر و قادر به آموزش و هدایت دیگران، توانایی برنامه‌ریزی و تحلیل، پاسخ‌گویی در برابر کارهای خود، سروکار داشتن با سطح وسیعی از کارها و فعالیت‌ها

تسلط (سطح چهار): خبرگی در انجام کار و آموزش دیگران، ایجاد نوآوری، سازگاری، عیب‌یابی، هدایت و راهنمایی دیگران.

واحد یادگیری ۲
توسعه وب اپلیکیشن

آیا تابه‌حال اندیشیده‌اید

- یک سایت چطور با پایگاه داده ارتباط برقرار می‌کند؟
- یک وب اپلیکیشن چطور کاربر مهمان را تشخیص می‌دهد؟
- اطلاعاتی که کاربران در فرم‌ها وارد می‌کنند، چطور در پایگاه داده ذخیره می‌شود؟
- برنامه‌نویس چگونه می‌تواند از بروز خطاها در سایت جلوگیری کند؟

در پایان از هنرجو انتظار می‌رود

۱ بتواند با استفاده از جنگو در پایگاه داده تغییرات اعمال کند.
۲ مفهوم فیلد و جدول در جنگو را درک کرده و درباره آن استدلال نماید.
۳ با استفاده از جنگو، بتواند اطلاعات فرم‌ها را در پایگاه داده ذخیره نمایند.
۴ با مفهوم ORM در جنگو آشنا شود و بتواند پرس‌وجوهای ساده بنویسد.
۵ مشکلات و باگ‌های مرسوم در وب اپلیکیشن‌ها را بشناسد و بتواند به‌صورت عملی مشکلات را برطرف نماید.

استاندارد عملکرد

هنرجو بتواند با ساخت Model در جنگو و استفاده از فرم‌های HTML ارتباط با پایگاه داده را پیاده‌سازی کند و اطلاعاتی که کاربران وارد می‌کنند را در جداول ذخیره و بازیابی نماید.

ساخت جدول شهر در پایگاه داده

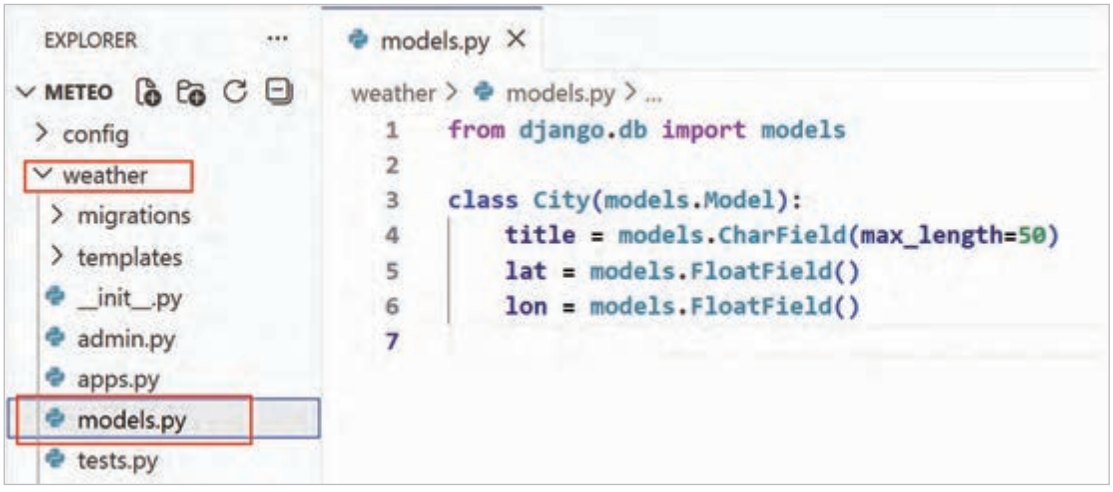
تا این مرحله سایت شما فقط دمای شهرهایی را نمایش می‌دهد که در view مشخص کرده‌اید، یعنی در show_temp_view اسم شهر ثابت است و همیشه دمای یک شهر، نمایش داده خواهد شد. روش بهتر این است که کاربر مهمان، قابلیت انتخاب شهر موردنظرش را داشته باشد.

جهت پیاده‌سازی این قابلیت، نیاز است، اطلاعات شهرها را به همراه طول و عرض جغرافیایی هر شهر، در پایگاه داده ذخیره کنید، سپس وقتی کاربر وارد سایت شد، لیستی از شهرها را مشاهده کرده و با کلیک روی نام شهر، دمای شهر موردنظر خود را ببینند.

برای پیاده‌سازی این قابلیت باید یک جدول (table) در پایگاه‌داده ایجاد کنید.

هر app یک فایل با نام models.py دارد. جنگو برای ساخت جداول از مدل‌ها (model) استفاده می‌کند. هر مدل در جنگو معادل یک جدول، با همان نام، در پایگاه‌داده است. در اولین قدم باید درون فایل models.py کلاس لازم برای تعریف جدول و ستون‌های جدول ایجاد شود.

فایل models.py در weather app را باز کنید. سپس محتوای این فایل را همانند شکل ۲۷ تغییر دهید.



شکل ۲۷ـ ساخت مدل City

توضیحات کدها:

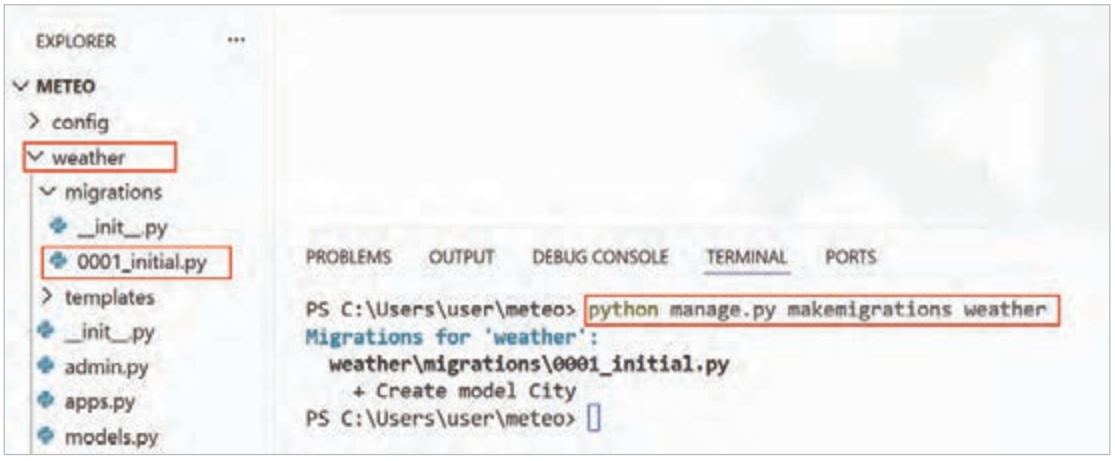
خط ۱: ماژول models از ماژول django.db، وارد (import) شده است. در واقع با این خط، ابزارها و کلاس‌های لازم برای تعریف مدل‌های پایگاه‌داده (Database Models) وارد (import) می‌شوند.

خط ۳: کلاس City جهت ساخت جدول شهرها با ارث‌بری از Model ایجاد شده است. Model کلاس پایه جنگو برای ساخت مدل‌ها (جدول‌های پایگاه‌داده) است و امکانات لازم برای ارتباط با پایگاه‌داده را فراهم می‌کند.

خط ۴: این دستور یک فیلد برای نگهداری نام شهر به‌صورت متن، در جدول می‌سازد که حداکثر ۵۰ کاراکتر طول دارد. برای CharField تعیین کردن max_length اجباری است.

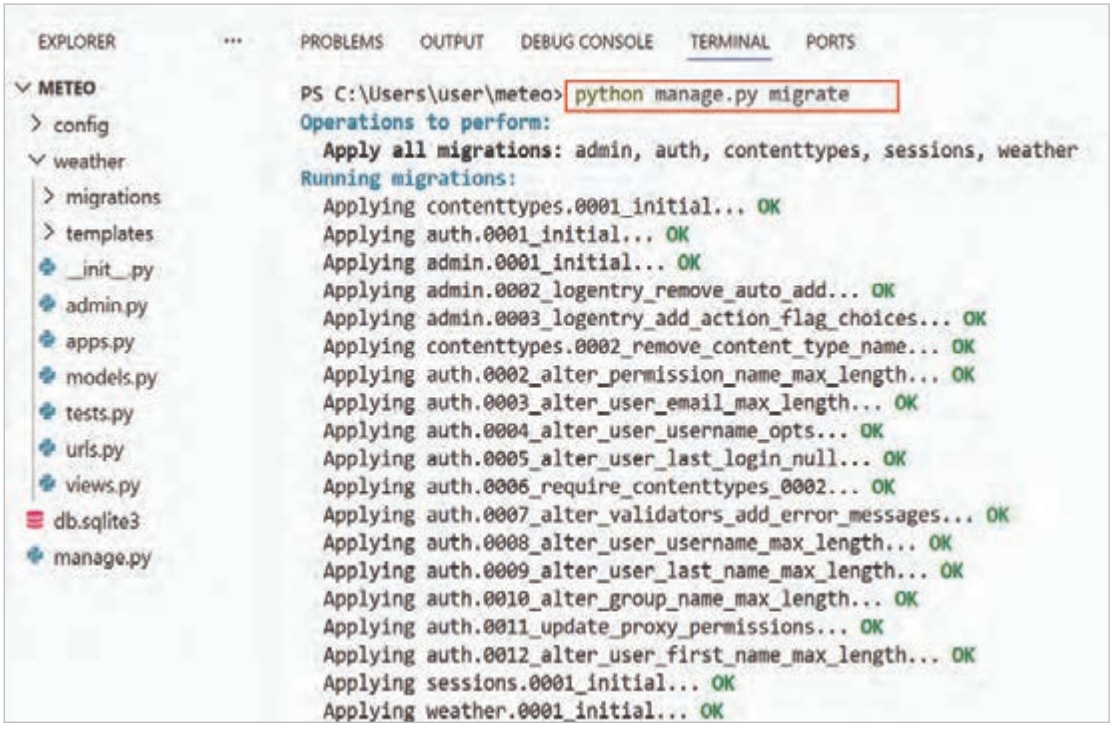
خط ۵: از FloatField جهت نگهداری اعداد اعشاری استفاده شده است. عرض جغرافیایی (Latitude) و طول جغرافیایی (Longitude) هر دو عدد اعشاری هستند.

برای استفاده از مدل City، نیاز است که جدول معادلش در پایگاه‌داده ساخته شود.
در ترمینال دستور makemigrations را همانند شکل ۲۸ اجرا کنید.



شکل ۲۸ـ اجرای دستور makemigrations

وقتی makemigrations را اجرا می‌کنید، درون weather app، فایلی با نام 0001_initial.py در پوشه migrations ایجاد می‌شود. این فایل کدهای لازم جهت تبدیل مدل به جدول واقعی را آماده‌سازی می‌کند.
در مرحله بعد دستور migrate را همانند شکل ۲۹ در ترمینال اجرا کنید.

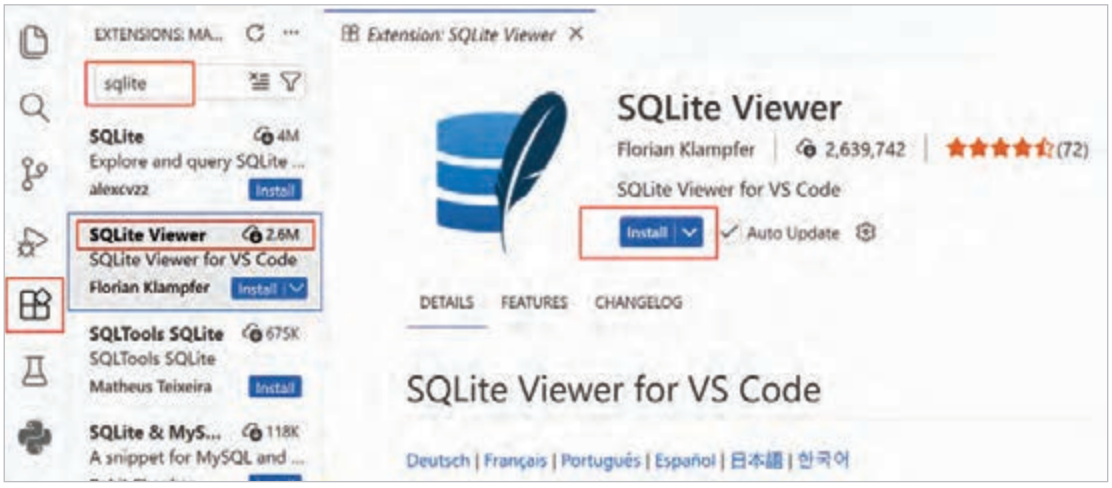


شکل ۲۹ـ اجرای دستور migrate

پس از اجرای migrate، جدول‌های پیش‌فرض جنگو و جدول city در پایگاه‌داده db.sqlite3 ساخته می‌شوند.

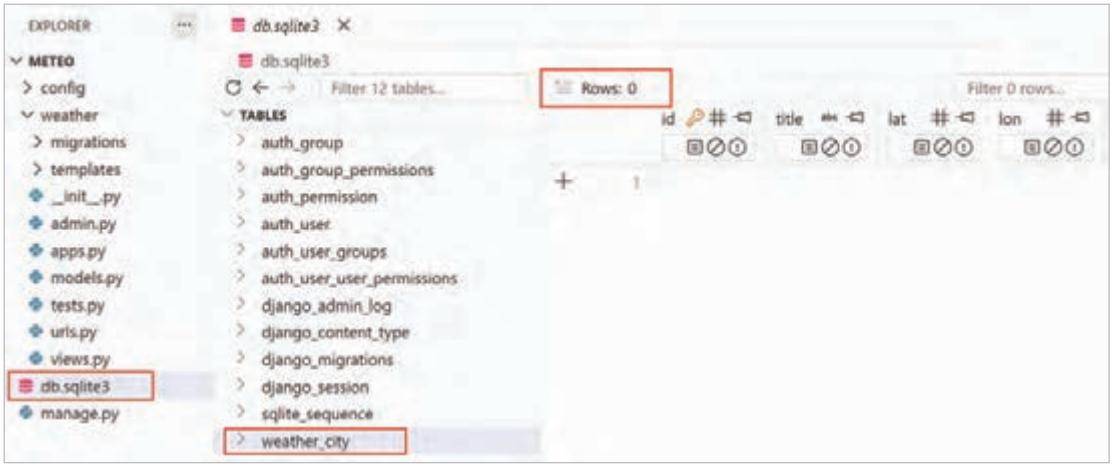
مشاهده جداول پایگاه‌داده

برای مشاهده ساختار و محتوای جداول درون پایگاه‌داده، می‌توانید افزونه SQLite Viewer را به‌عنوان یک ابزار کمکی نصب کنید. برای نصب این افزونه مطمئن شوید که اینترنت متصل باشد (شکل ۳۰).



شکل ۳۰ـ جست‌وجوی افزونه SQLite Viewer

پس از نصب افزونه ذکر شده، با کلیک بر روی فایل db.sqlite3 جدول‌های پیش‌فرض جنگو و جدول city را مشاهده خواهید کرد (شکل ۳۱).



شکل ۳۱ـ جداول پایگاه‌داده

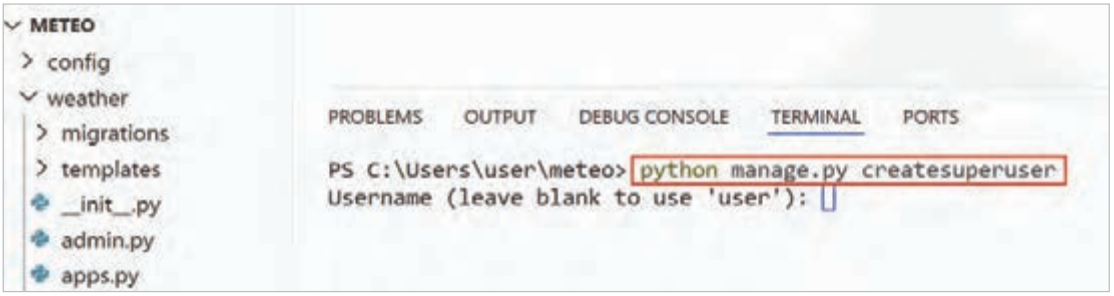
همان‌طورکه در شکل ۳۱ مشخص است، جدول city با پیشوند weather ساخته شده است. با کلیک روی جدول weather_city، لیست فیلدها در سمت راست نمایش داده می‌شود. در حال حاضر هیچ رکوردی در جدول city وجود ندارد.
با افزونه SQLite Viewer فقط می‌توانید محتوا و ساختار جداول را مشاهده کنید و امکان ذخیره اطلاعات وجود ندارد.
برای تغییر محتوای جدول city و یا افزودن نام شهرهایی جدید به جدول لازم است وارد پنل ادمین جنگو شوید. البته هنوز نمی‌توانید از طریق پنل admin داده‌های مربوط به شهرها را وارد دیتابیس کنید، چرا که هنوز کاربر admin تعریف نشده است؛ بنابراین در مرحله بعد نوبت تعریف کاربر admin است.

پنل مدیریت جنگو

برای ورود به پنل ادمین، نیاز به نام کاربری و رمز عبور است. دستور زیر را در ترمینال وارد کنید:

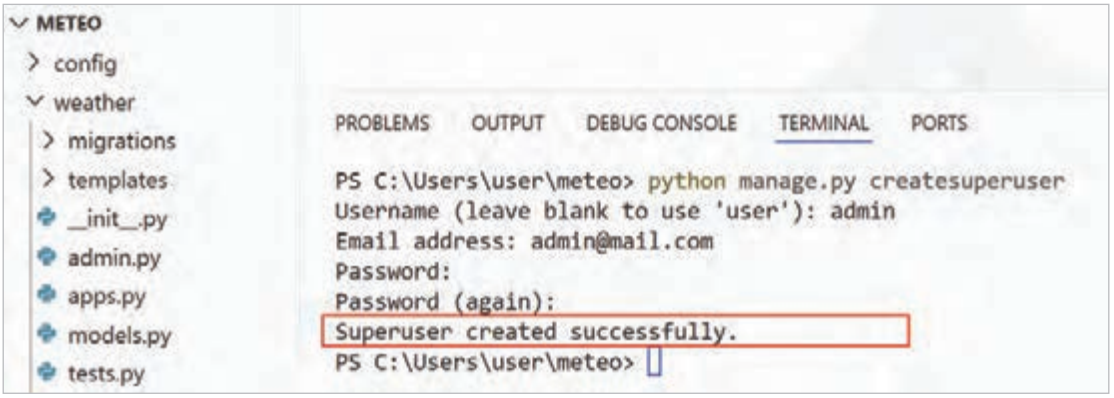
```
python manage.py createsuperuser
```

جنگو از شما نام کاربری درخواست می‌کند (شکل ۳۲).



شکل ۳۲ـ اجرای دستور createsuperuser

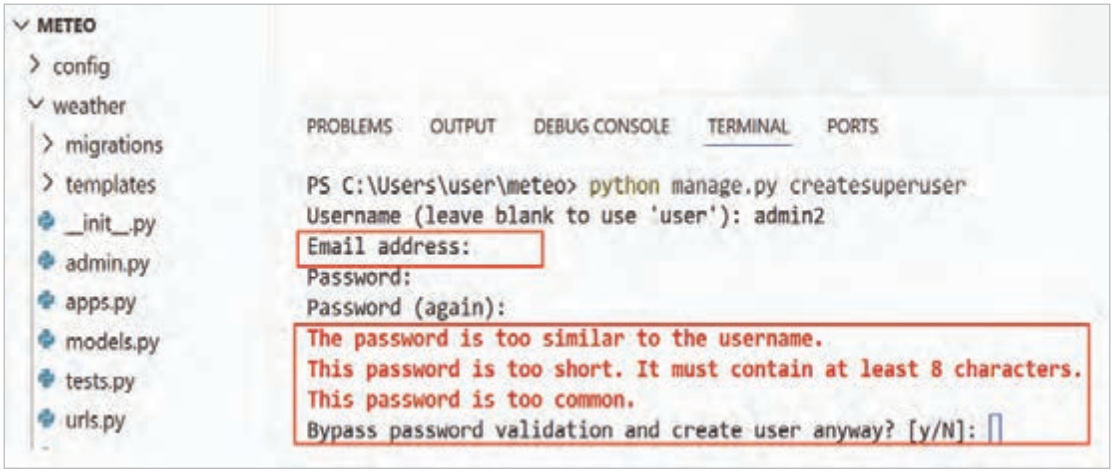
همان‌گونه که مشخص است، نام کاربر جاری ویندوز را به‌عنوان نام کاربری پیشنهاد داده است.
می‌توانید نام کاربری admin و یا هر نام کاربری دیگری را به‌دلخواه وارد کنید، سپس به‌ترتیب، ایمیل، کلمه عبور و تکرار کلمه عبور را وارد کنید (شکل ۳۳).



شکل ۳۳ـ ایجاد موفق کاربر ادمین

نکته

می‌توانید ایمیل را خالی وارد کنید، همچنین اگر رمز عبور ساده‌ای وارد کنید، همانند شکل ۳۴ به شما اخطار خواهد داد.



شکل ۳۴ـ اخطار ساده‌بودن رمز ادمین

پس از ایجاد کاربر ادمین، با افزونه SQLite Viewer می‌توانید، کاربر ایجاد شده را در جدول auth_user همانند شکل ۳۵ مشاهده کنید.



شکل ۳۵ـ نمایش کاربر admin ایجاد شده در دیتابیس

اکنون جهت ورود به پنل مدیریت ادمین جنگو، سرور جنگو را روشن کرده و صفحه ادمین را با تایپ آدرس زیر در نوار آدرس مرورگر، باز کنید:

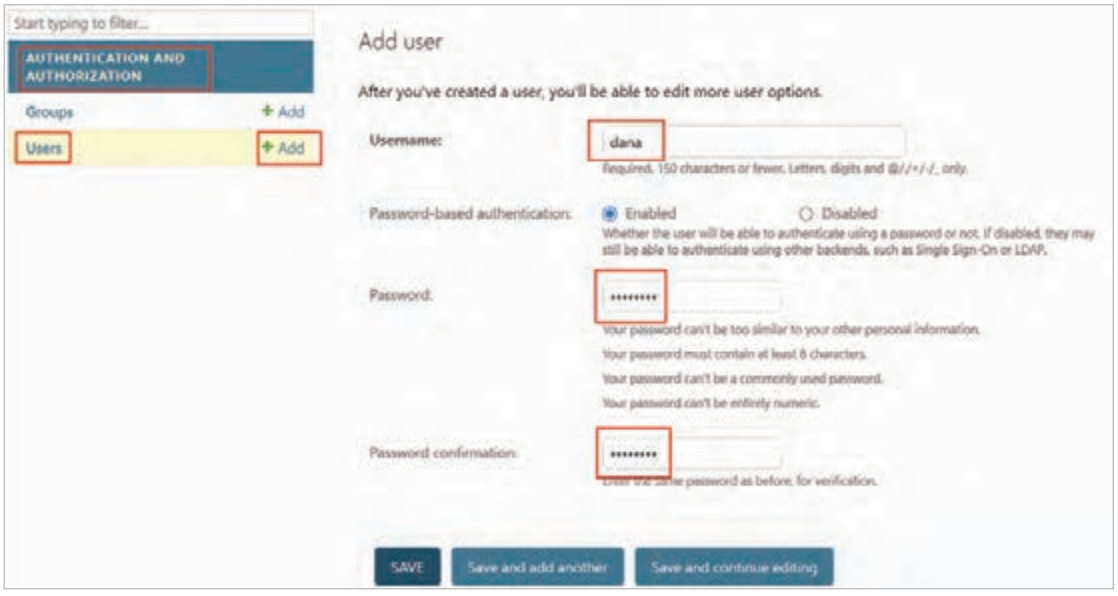
http://127.0.0.1:8000/admin/

سپس با نام کاربری و رمز عبوری که ساخته‌اید، وارد شوید.
پنل ادمین همانند شکل ۳۶ نمایش داده خواهد شد.



شکل ۳۶ــ ورود موفق به پنل ادمین جنگو

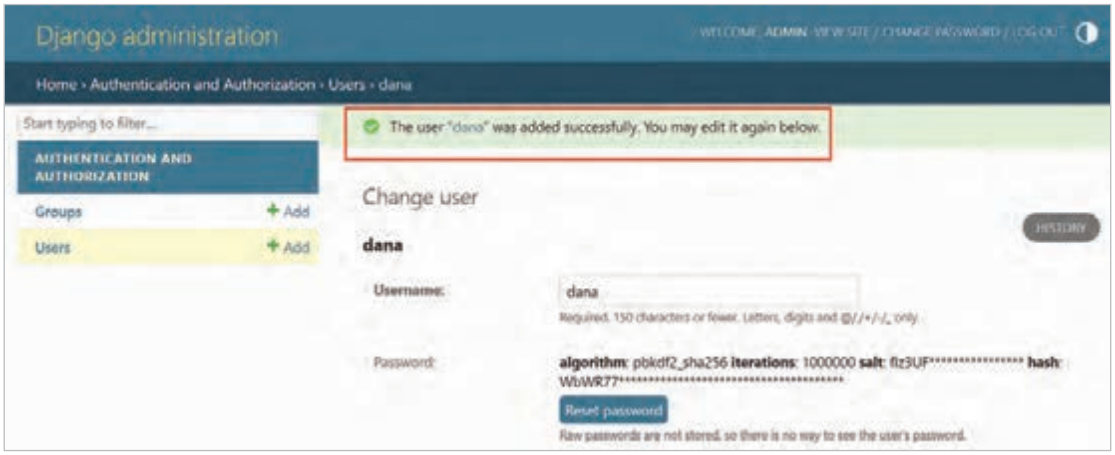
در قسمت Authentication and Authorization[1] با استفاده از Users می‌توان کاربران سایت را مشاهده و مدیریت کرد.
با استفاده از دکمه add می‌توان کاربر جدید ایجاد کرد. برای ایجاد کاربر جدید، لازم است فیلدهای نام کاربری، رمز عبور و تکرار رمز عبور را وارد کرده و دکمه save را بزنید (شکل ۳۷).



شکل ۳۷ــ فرم ساخت کاربر جدید

۱- احراز هویت و مجوز

پیغام ذخیره موفقیت‌آمیز همانند شکل ۳۸ مشاهده خواهید کرد. در این مرحله می‌توانید اطلاعات تکمیلی، مثل نام، نام خانوادگی، ایمیل و... را برای کاربر ایجاد شده ثبت کنید.



شکل ۳۸ـ ذخیره موفق کاربر

برای مشاهده لیست کاربران ایجاد شده، روی Users کلیک کنید (شکل ۳۹).



شکل ۳۹ـ مشاهده لیست کاربران

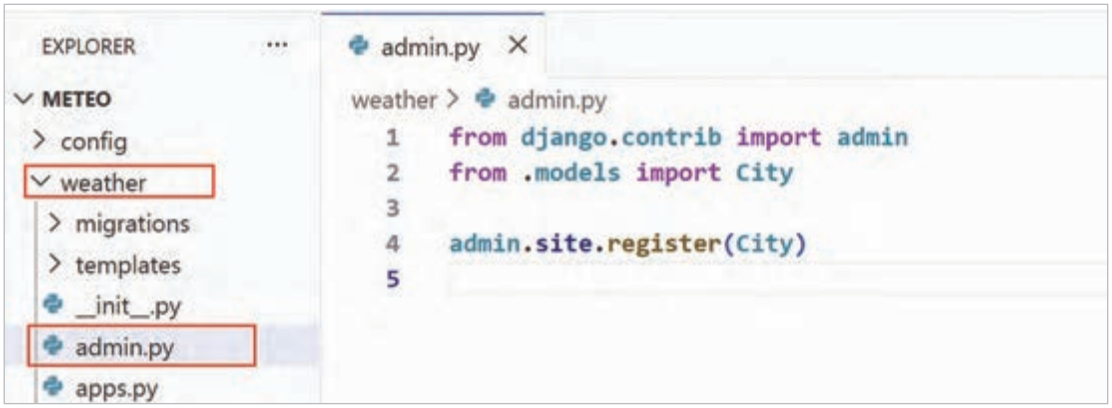
فعالیت

با کمک هنرآموز خود، کاربر دیگری ایجاد کنید. سپس با کاربرهای مختلف در پنل ادمین login کرده و تفاوت کاربرهای مختلف را بررسی کنید.

افزودن جدول شهر به پنل ادمین

اگر دقت کرده باشد در شکل ۳۶، در پنل ادمین جدول شهرها مشاهده نشد، با انجام مراحلی که در ادامه آمده است، می‌توانید جدول City را به پنل ادمین اضافه کنید.

هر app در جنگو یک فایل admin.py دارد. فایل admin.py در weather app را باز کنید و همانند شکل ۴۰، مدل City را به ادمین جنگو معرفی کنید.



شکل ۴۰ـ معرفی مدل City به پنل ادمین جنگو

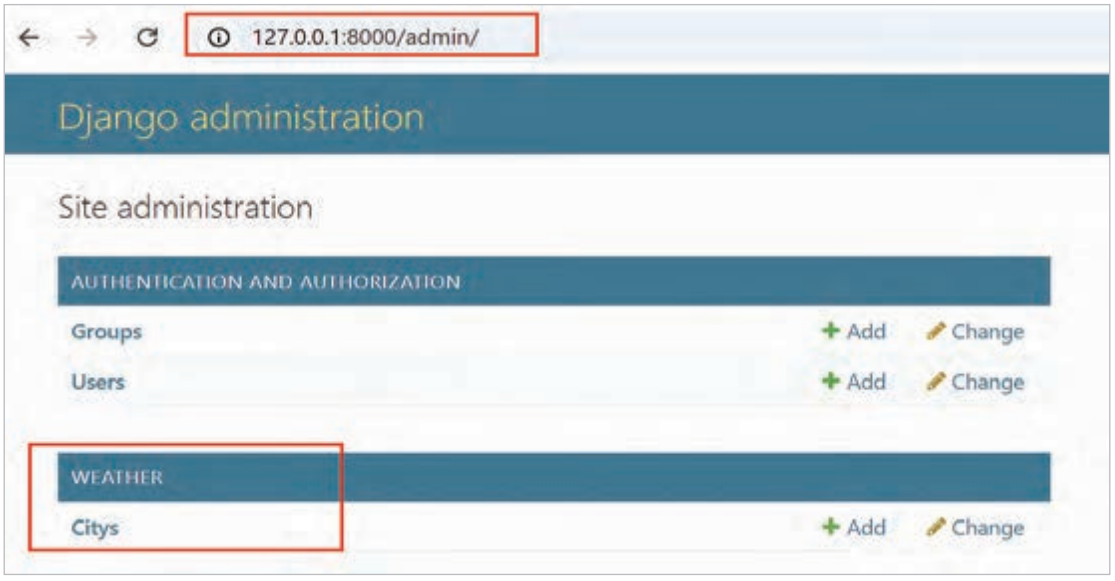
توضیحات کدها:

خط ۱: ماژول admin جهت مدیریت پنل ادمین از contrib وارد (import) شده است.

خط ۲: مدل City از فایل models.py وارد (import) شده است.

خط ۴: مدل City با دستور register به پنل ادمین معرفی شده است.

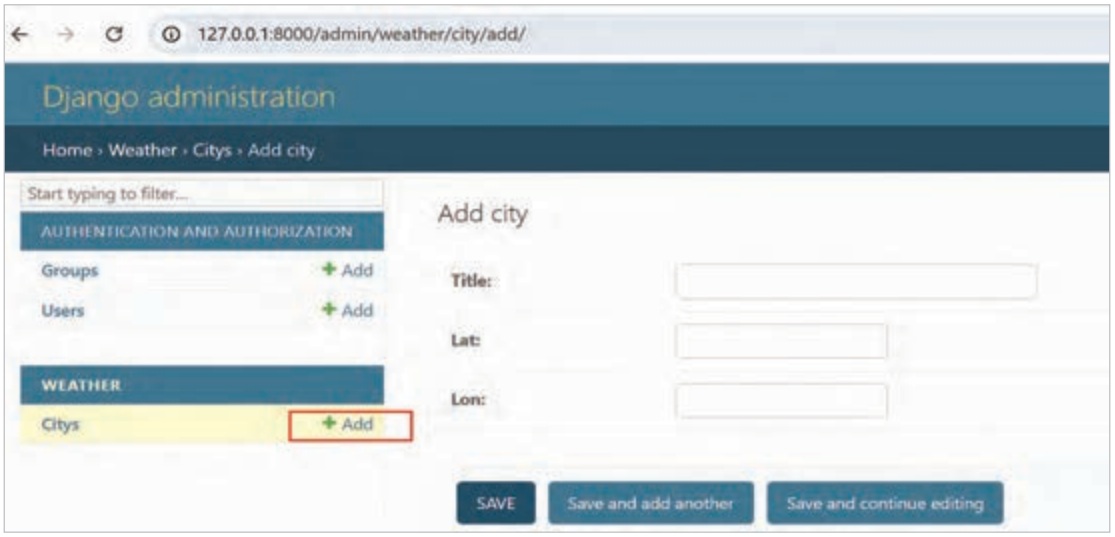
اکنون فایل را ذخیره کنید، سپس مرورگر را refresh کنید، جدول city به ادمین جنگو اضافه شده است و حرف s به انتهای نام مدل اضافه شده است (شکل ۴۱).



شکل ۴۱ـ اضافه شدن City به پنل ادمین

وارد کردن اطلاعات شهرها

در پنل ادمین، روی دکمه Add (مقابل Citys) کلیک کنید، فرم ثبت شهر (Add city) را مشاهده خواهید کرد (شکل ۴۲) .



شکل ۴۲ـ فرم Add city

فعالیت

با استفاده از اطلاعات جدول ۲، چند شهر را با فرم Add city در پایگاه داده ذخیره کنید.

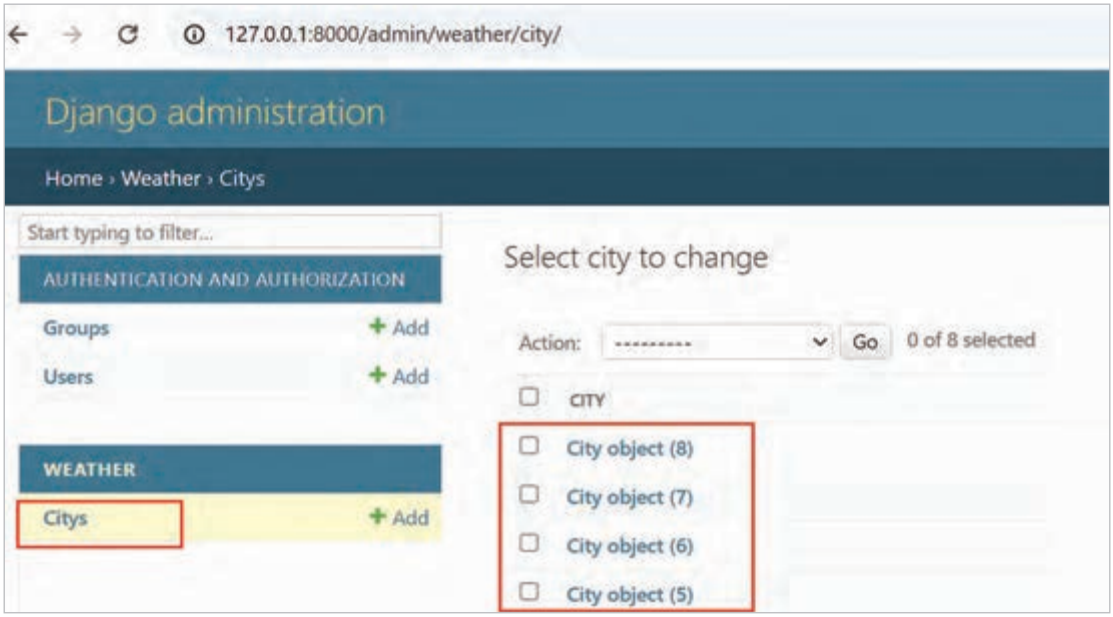
جدول ۲ـ طول و عرض جغرافیایی چند شهر

نام شهر	عرض جغرافیایی (lat)	طول جغرافیایی (lon)
میناب	27.1478	57.0738
همدان	37.7981	48.5146
تهران	35.7219	51.3347
الشتر	33.8632	48.2617
تفرش	34.6936	50.0154
بابل	36.5387	52.6765
جهرم	28.5177	53.5744
تبریز	38.0792	46.2887
سبزوار	36.2152	57.6678

فعالیت

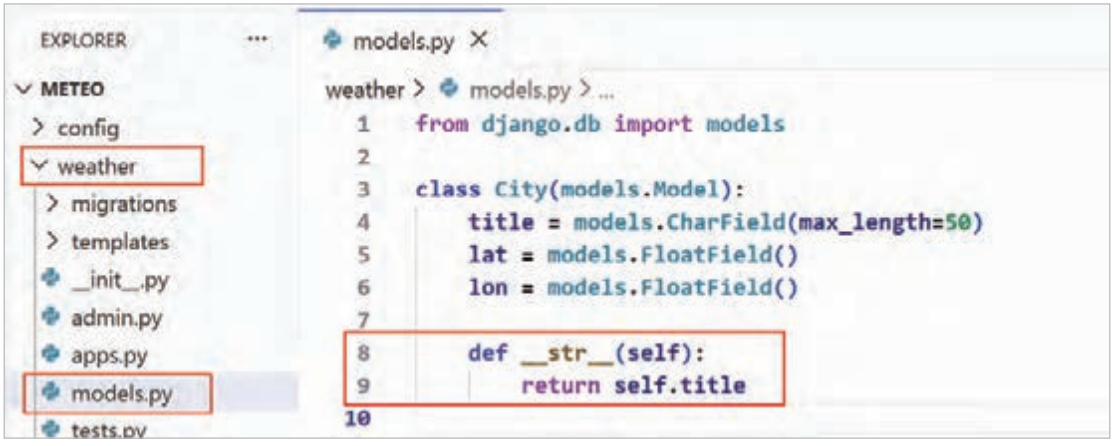
با کمک هنرآموز خود، مقدار طول و عرض جغرافیایی چند شهر دیگر را یافته و در پایگاه داده ذخیره کنید.

پس از ثبت شهرها، با کلیک روی Citys، لیست شهرهای ثبت شده نمایش داده می‌شود (شکل ۴۳).



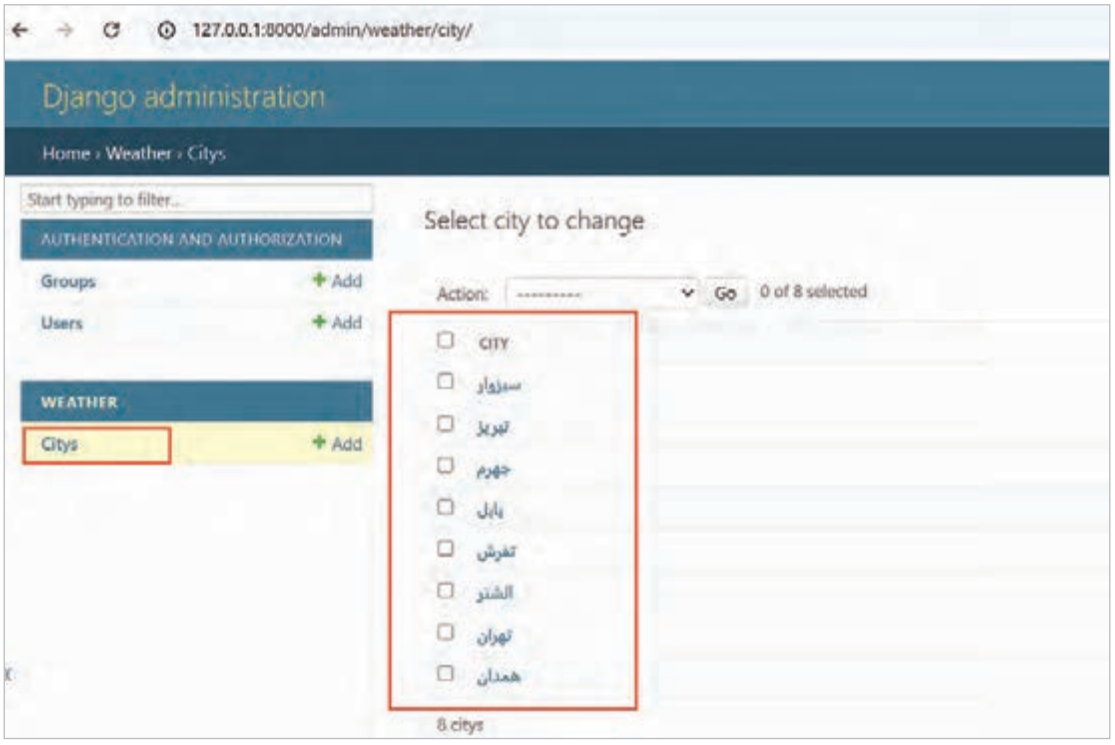
شکل ۴۳ـ لیست شهرهای ثبت شده

در شکل ۴۳ از City object برای نمایش عنوان هر شهر استفاده شده است. جهت نمایش مناسب عنوان شهرها، فایل models.py را باز کنید و تغییراتی همانند شکل ۴۴ برای مدل City اعمال کنید و فایل را ذخیره کنید.



شکل ۴۴ـ افزودن متد str به مدل شهر

اگر پنل ادمین را refresh کنید، نتیجه به‌صورت شکل ۴۵ خواهد بود.



شکل ۴۵ـ نمایش مناسب عنوان شهرها در پنل ادمین

کنجکاوی

در زبان انگلیسی جمع City برابر با Cities است، اما در پنل ادمین جنگو عنوان جدول به‌صورت اشتباه Citys نمایش داده می‌شود. بررسی کنید چطور می‌توان این عنوان را در بخش ادمین جنگو تغییر داد.

آشنایی با ابزار ORM جنگو

اکنون که چند رکورد حاوی نام چند شهر و مختصاتشان را به جدول شهرها اضافه کردید، می‌توانید از داخل فایل views.py یکی از رکوردها (مثلاً اولین رکورد) را بخوانید، وضعیت آب و هوای آن شهر را به‌دست آورید و جهت نمایش به فایل index.html ارسال کنید.

برای این‌کار لازم است ابتدا ابزار ORM جنگو را بشناسید.

برای کار با پایگاه داده از زبان SQL استفاده می‌شود، مثلاً دستور SELECT یکی از دستورات اصلی زبان SQL است.

با دستور SELECT به‌صورت زیر می‌توانید، همه شهرهای جدول City را از پایگاه داده دریافت کنید:

```
SELECT id, title, lat, lon FROM weather_city
```

اما در جنگو یک ابزار مفید با نام ORM وجود دارد که کار با پایگاه داده را راحت‌تر می‌کند. ORM مثل یک مترجم است. عملیات مربوط به ارتباط با پایگاه داده را با زبان پایتون دریافت و در پشت صحنه، به دستورات SQL ترجمه می‌کند.
به‌عنوان مثال دستور معادل پرس و جوی قبلی در ORM جنگو به‌صورت زیر است:

```
City.objects.all()
```

در این کد پایتونی، City نام مدل است.
اکنون کد show_temp_view را با استفاده از ORM جنگو همانند شکل ۴۶ تغییر دهید.



شکل ۴۶ـ استفاده از ORM جنگو

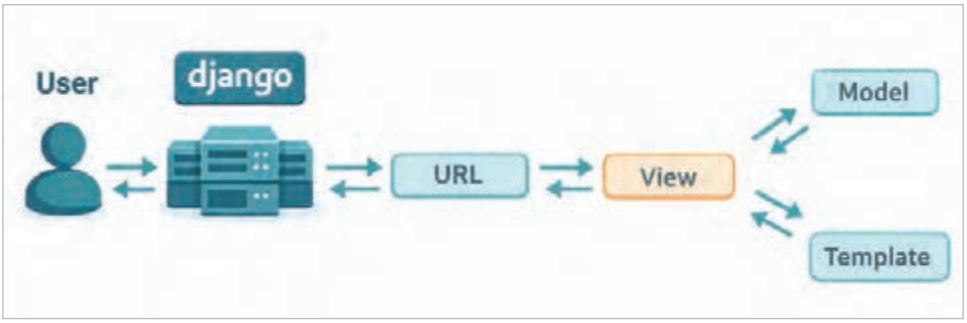
توضیحات کدها:
خط ۳: مدل City از models.py در این فایل import شده است.
خط ۶: با استفاده از ORM جنگو، اولین شهر ذخیره شده در جدول شهر، در متغیر city ذخیره شده است.
خط ۷: مقدار lat و lon از شهرِ واکشی شده از جدول، دریافت و در متغیرهای lat و lon ذخیره شده است.
خط ۱۷: نام شهر نیز از پایگاه داده دریافت شده و درون دیکشنری context ذخیره شده است.
فایل را ذخیره و صفحه اصلی سایت را باز کنید. دمای شهر همدان نمایش داده می‌شود.

نمایش لیست شهرها

تا اینجا توانستید چند شهر را ذخیره‌سازی کنید. قرار است که این شهرها را از پایگاه داده بخوانید و درون سایت به‌صورت یک لیست نمایش دهید تا کاربر بتواند با کلیک بر روی شهر دلخواهش، وضعیت دمای آن شهر را مشاهده کند.

این‌کار به کمک view و template صورت خواهد گرفت.

در جنگو، view می‌تواند اطلاعات را از دیتابیس دریافت کرده و سند html را جهت نمایش به کاربر آماده کند. شکل ۴۷ را مشاهده کنید، request از url به view ارسال شده و view با استفاده از model و template، اطلاعات را جهت نمایش به کاربر آماده می‌کند.



شکل ۴۷ـ دریافت اطلاعات از مدل توسط View

جهت نمایش لیست شهرها در صفحه اصلی سایت می‌توانید چهار مرحله را به‌صورت زیر انجام دهید:

۱ـ ساختن یک تابع با نام city_list_view در Views.py

۲ـ ساختن یک template با نام city_list.html

۳ـ تغییر و تنظیم فایل urls.py

۴ـ قرار دادن لینک «لیست شهرها» در صفحه اصلی

مرحله اول: یک view با نام city_list_view همانند شکل ۴۸ ایجاد کنید.



شکل ۴۸ـ افزودن city_list_view

توضیحات کدها:

خط ۶: با استفاده از ORM جنگو یک Query اجرا شده و لیست همه شهرها از جدول City واکشی شده است. ORM جنگو دستورات مختص به خود را دارد و این دستورات به کدهای SQL تبدیل شده و توسط پایگاه داده اجرا می‌شوند. سپس نتیجه دستور در متغیری با نام cities ذخیره شده است.

خط ۸: دیکشنری context برای city_list.html ارسال شده است، مقدار value این دیکشنری یعنی cities در خط ۶ مقداردهی شده است.

مرحله دوم: اکنون یک فایل html با نام city_list.html درون پوشه templates ایجاد کنید، کد این فایل را همانند شکل ۴۹ وارد کنید.



شکل ۴۹ـ افزودن city_list.html

توضیحات کدها:

خط ۱۰: یک حلقه روی cities اجرا شده است تا لیست شهرها را به‌صورت تگ p در مرورگر نمایش دهد. cities قبلاً در فایل views.py درون یک دیکشنری تعریف شده است.

خط ۱۱: با استفاده از {{ }} می‌توانید به فیلدهای هر item دسترسی داشته باشید. title فیلدی است که برای عنوان شهر، در مدل City تعریف کردید.

خط ۱۲: حتماً باید حلقه for را با endfor خاتمه دهید، در غیر این‌صورت در مرورگر با خطا مواجه خواهید شد.

مرحله سوم: فایل urls.py در weather app را همانند شکل ۵۰ تغییر دهید.



شکل ۵۰ــ افزودن path لیست شهرها به urls.py

توضیحات کدها:

خط ۶: یک path جدید جهت استفاده از city_list_view اضافه شده و به این معنی است که لیست شهرها در آدرس http://127.0.0.1:8000/city-list/ نمایش داده می‌شود.

مرحله چهارم: کد show_temp.html را جهت اضافه کردن لینک به صفحه لیست شهرها، همانند شکل ۵۱ تغییر دهید.



شکل ۵۱ــ اضافه کردن لینک به لیست شهرها

توضیحات کدها:

خط ۱۵: در این خط تگ a اضافه شده و مقدار href آن برابر عبارت {% url 'city_list' %} است.

در این عبارت {% url 'city_list' %} یک Template Tag است. از تگ url استفاده شده است و city_list آرگومان آن است. این خط با استفاده از نام مسیر city_list، آدرس لینک لیست شهرها را به صورت خودکار تولید می‌کند (شکل ۵۲).



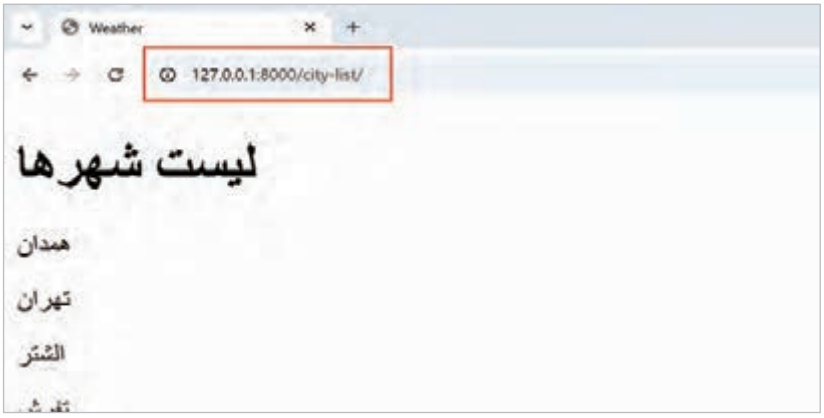
شکل ۵۲ــ روند کار تگ url در جنگو

اکنون فایل را ذخیره کرده و صفحه اصلی سایت را refresh کنید، لینک اضافه شده را همانند شکل ۵۳ مشاهده خواهید کرد.



شکل ۵۳ــ نمایش لینک لیست شهرها

اکنون روی لینکِ اضافه شده (لیست شهرها) کلیک کنید، همانند شکل ۵۴، لیست شهرها را مشاهده خواهید کرد.



شکل ۵۴ـ نمایش لیست شهرها

نمایش دمای شهر انتخاب شده توسط کاربر

شهرهایی که در لیست شهرها نمایش داده می‌شوند، قابلیت کلیک ندارند، لازم است تغییراتی اعمال کنید تا کاربر امکان کلیک کردن روی شهر مورد نظر خود را داشته باشد، تا دمای شهر انتخاب شده، نمایش داده شود.

ابتدا فایل city_list.html را باز کرده و همانند شکل ۵۵، تغییرات مورد نظر را اعمال کنید.



شکل ۵۵ـ افزودن لینک به هر شهر

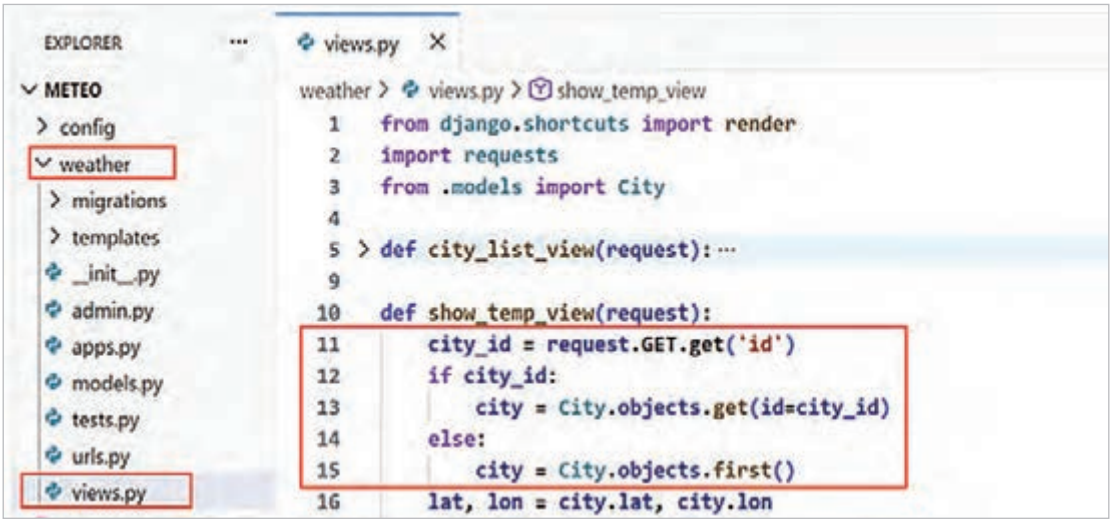
توضیحات کدها:

خط ۱۰: برای اینکه نام شهرها قابلیت کلیک شدن داشته باشند، یک تگ a درون تگ p ایجاد شده و جهت نمایش عنوان لینک، از متغیر {{ item.title }} استفاده شده است. آدرس لینک (href) به صفحه اصلی اشاره کرده و متغیر {{ item.id }} به پارامتر id مقدار داده است.

این کد id را به درخواست GET اضافه و به صفحه اصلی ارسال می‌کند. به عنوان مثال اگر آیدی شهر انتخاب شده ۱۱ باشد، کاربر به آدرس زیر منتقل می‌شود:

http://127.0.0.1:8000/?id=11

در گام بعد، views.py را باز کنید و show_temp_view را همانند شکل ۵۶ تغییر دهید.



شکل ۵۶ــ استفاده از id جهت انتخاب شهر

توضیحات کدها:

خط ۱۱: مقدار پارامتر id که از طریق متد GET در نوار آدرس ارسال شده است، در متغیر city_id ذخیره می‌شود.

درصورتی‌که پارامتر id در نوار آدرس وجود نداشته باشد، تابع get('id') که بعد از GET استفاده شده است، مقدار city_id را برابر None ذخیره خواهد کرد.

خط ۱۳: اگر city_id مقدار داشته باشد، با استفاده از ORM جنگو، شهر انتخاب شده توسط کاربر، به‌صورت یک شیء (object)، در متغیر city ذخیره می‌شود.

خط ۱۵: اگر city_id برابر None باشد (id در نوار آدرس وجود نداشته باشد)، اولین شهر از پایگاه داده واکشی می‌شود.

اکنون تمام فایل‌ها را ذخیره نموده، صفحه لیست شهرها را refresh کنید. همه شهرها لینک دارند. روی هر شهری که کلیک کنید، در صفحه جدید دمای هوای آن شهر مشاهده می‌شود (شکل ۵۷).



شکل ۵۷ــ نمایش دمای یک شهر کلیک شده

ورود به سایت (login)

در حال حاضر، صفحه اصلی و صفحه لیست شهرها توسط کاربر مهمان و کاربر ادمین قابل مشاهده هستند. اما هر سایت می‌تواند صفحه‌هایی داشته باشد که برای دیده شدن نیاز به login (ورود به سایت) داشته باشند. به‌عنوان مثال برای دیدن پنل ادمین جنگو، باید با نام کاربری و رمز عبور ادمین login کنید.
شما می‌توانید، امکان دسترسی به قسمت افزودن نام شهرها را فقط به کاربرانی اعطا کنید که با نام کاربری و رمز عبور خودشان وارد سایت می‌شوند.
جنگو خودش به‌طور پیش‌فرض امکاناتی برای مدیریت کاربران دارد تا بتوانید عملیات login و logout را درون سایت اجرا نمایید. فقط کافیست فایل template لازم یعنی login.html را در پوشه مناسب ساخته و در فایل urls اصلی پروژه (موجود در پوشه config) یک path جدید اضافه کنید.
مرحله اول: فایل urls.py اصلی پروژه را همانند شکل ۵۸ تغییر دهید.



شکل ۵۸ــ افزودن path جدید جهت login و logout

توضیحات کدها:

خط ۷: یک path جدید اضافه شده است و در ادامه منجر به این خواهد شد که App مدیریت کاربران جنگو (auth) به کمک urls و views پیش‌فرض، فایل login.html را نمایش دهد. برای مدیریت احراز هویت کاربر در پروژه جنگو، از ماژول django.contrib.auth استفاده شده است.

با اضافه‌کردن این خط کد به فایل urls.py، امکان ورود، خروج و تغییر رمز عبور به سایت اضافه می‌شود. همه این قابلیت‌ها توسط ماژول django.contrib.auth مدیریت می‌شوند و نیازی به تعریف دستی path دیگری ندارید.

مرحله دوم: در پوشه templates از weather app، یک‌پوشه جدید با نام registration ایجاد کنید، سپس درون آن یک فایل با نام login.html همانند شکل ۵۹ بسازید.



شکل ۵۹ــ افزودن فایل login.html

توضیحات کدها:

خط ۱۰: یک فرم با متدی از نوع post ایجاد شده است.

خط ۱۱: فیلد csrf_token به فرم اضافه شده است. csrf_token در جنگو یک مقدار امنیتی است که داخل فرم‌ها قرار می‌گیرد تا مطمئن شود درخواست ارسال‌شده واقعاً از همان سایت و توسط کاربر مجاز آمده است. این توکن از حملهٔ Cross-Site Request Forgery جلوگیری می‌کند.

خط ۱۲: فیلدهای فرم را به‌صورت تگ‌های<p> در HTML نمایش می‌دهد.

خط ۱۳: یک دکمه از نوع submit برای ارسال اطلاعات به سرور جنگو اضافه شده است.

کنجکاوی

در مورد csrf_token و حمله Cross-Site Request Forgery تحقیق و بررسی کنید.

مرحله سوم: در فایل settings.py، تنظیمات لازم را برای بعد از اتمام عملیات login و logout همانند شکل ۶۰ انجام دهید.



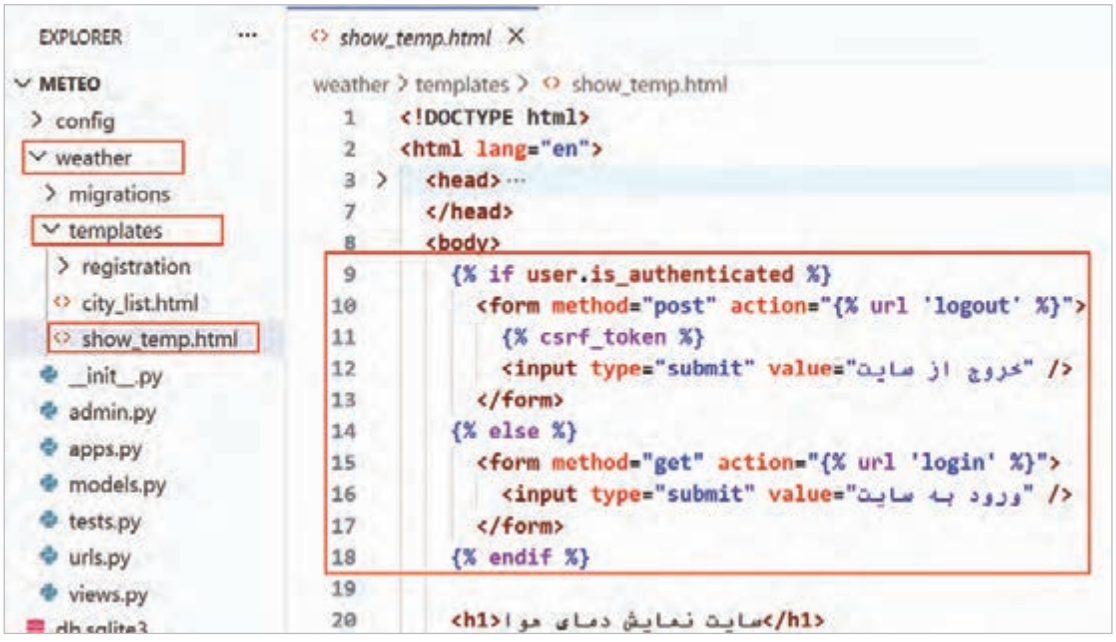
شکل ۶۰ــ تنظیمات login و logout در settings.py

توضیحات کدها

خط ۱۲۵: کاربر بعد از login موفق به صفحه home منتقل خواهد شد. home همان نامی‌است که برای آدرس صفحه اصلی در urls.py تعریف شده است.

خط ۱۲۶: کاربر بعد از logout موفق به صفحه home منتقل خواهد شد.

مرحله چهارم: در فایل show_temp.html، دکمه‌های ورود و خروج را همانند شکل ۶۱ اضافه کنید.



شکل ۶۱ــ افزودن دکمه ورود و خروج به صفحه اصلی

توضیحات کدها:

خط ۹: این دستور بررسی می‌کند که آیا کاربر وارد سیستم شده است یا خیر. اگر user.is_authenticated مقدار True داشته باشد؛ یعنی کاربر احراز هویت شده است و دستورات داخل این شرط اجرا خواهند شد.

خط ۱۰: در این خط یک فرم HTML، تعریف شده است که درخواست را با متد POST به آدرس خروج از حساب کاربری ارسال می‌کند.

خط ۱۴: از else استفاده شده است و به این معنی است، اگر کاربر login نباشد، چه چیزی به او نمایش داده شود.

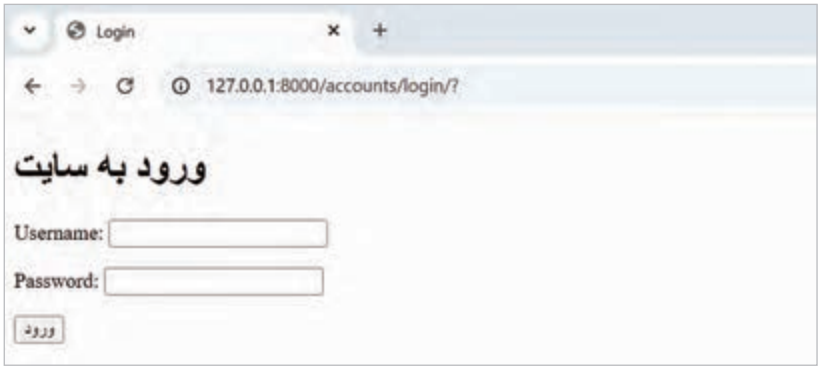
خط ۱۵: در این خط یک فرم با متد GET تعریف شده است که کاربر را به صفحه ورود به سایت یعنی صفحه login هدایت می‌کند.

اکنون فایل‌ها را ذخیره کرده و سرور جنگو را متوقف و دوباره راه‌اندازی کنید. سپس صفحه اصلی سایت را refresh کنید. اگر کاربر login نکرده باشد، دکمه ورود به سایت را مشاهده می‌کند (شکل ۶۲).



شکل ۶۲ـ نمایش دکمه ورود برای کاربر مهمان

با کلیک روی دکمه ورود به سایت، کاربر به صفحه login منتقل می‌شود (شکل ۶۳).



شکل ۶۳ـ فرم ورود به سایت

پس از ورود موفق به سایت، کاربر به صفحه اصلی سایت منتقل شده و این بار دکمه خروج از سایت برای کاربر نمایش داده می‌شود.

درصورتی‌که کاربر روی دکمه خروج از سایت کلیک کند، از سایت خارج شده و بلافاصله به صفحه اصلی منتقل خواهد شد.

ثبت شهر جدید با کاربر عضو سایت

تاکنون فقط در پنل ادمین و با کاربر admin امکان ثبت شهرها را داشتید. اما این امکان وجود دارد که کاربرهای عضو سایت هم بتوانند بدون مراجعه به پنل ادمین، شهر جدید را ثبت کنند.

برای این کار می‌توانید با استفاده از فرم‌های جنگو و انجام چهار مرحلهٔ زیر، این قابلیت را به سایت اضافه کنید.

۱ـ ایجاد forms.py برای ساخت فرم ثبت شهر جدید

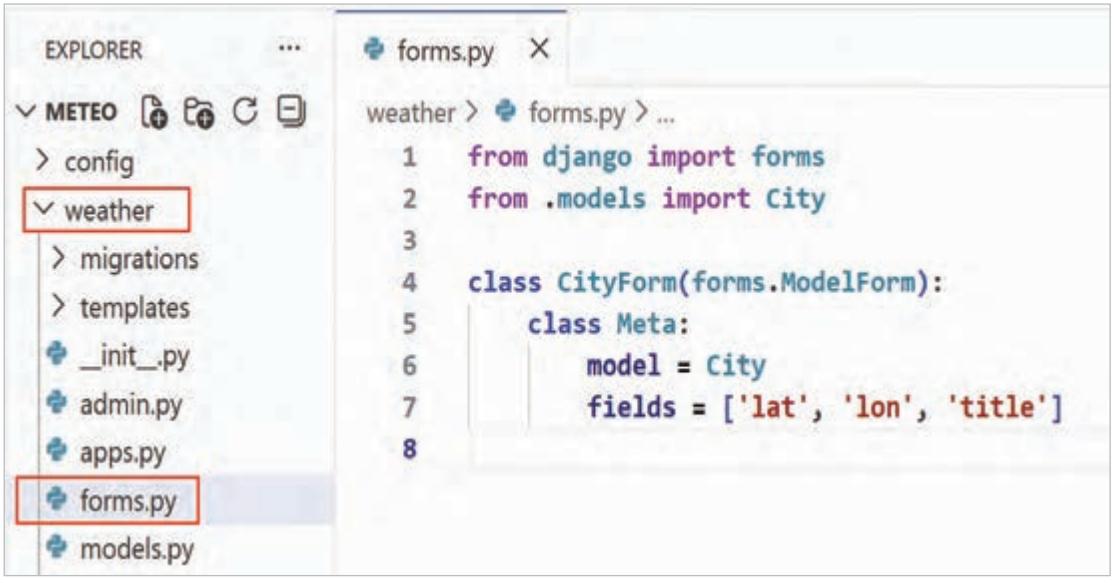
۲ـ ساختن template ساده برای فرم

۳ـ ساختن تابع add_city_view برای نمایش فرم

۴ـ اضافه‌کردن path مخصوص فرم در فایل urls.py

۵ـ قراردادن لینک «افزودن شهر» در صفحه اصلی

مرحله اول: یک فایل جدید در پوشه weather با نام forms.py ایجاد کنید و همانند شکل ۶۴ آن را تکمیل کنید.



شکل ۶۴ـ ایجاد CityForm

توضیحات کدها:

خط ۱: ماژول forms درون این فایل import شده است. این ماژول همه کارهای مربوط به فرم‌ها را پشتیبانی می‌کند.

خط ۲: کلاس City از فایل model درون این فایل import شده است. City همان مدل شهرها است که قبلاً تعریف کرده‌ایم.

خط ۴: یک کلاس با نام CityForm ساخته شده است که از ModelForm ارث‌بری می‌کند.

خط ۵: برای تعیین مدل و فیلدهایی که می‌خواهید نمایش داده شود باید از یک کلاس Meta در کلاس اصلی استفاده شود و در آن مدل و فیلدها مشخص گردد.

خط ۶: کلاس City به‌عنوان model تعیین شده است.

خط ۷: لیست فیلدهایی که می‌خواهید نمایش داده شود، تعیین شده است.

مرحله دوم: فایل add_city.html را همانند شکل ۶۵ ایجاد کنید.

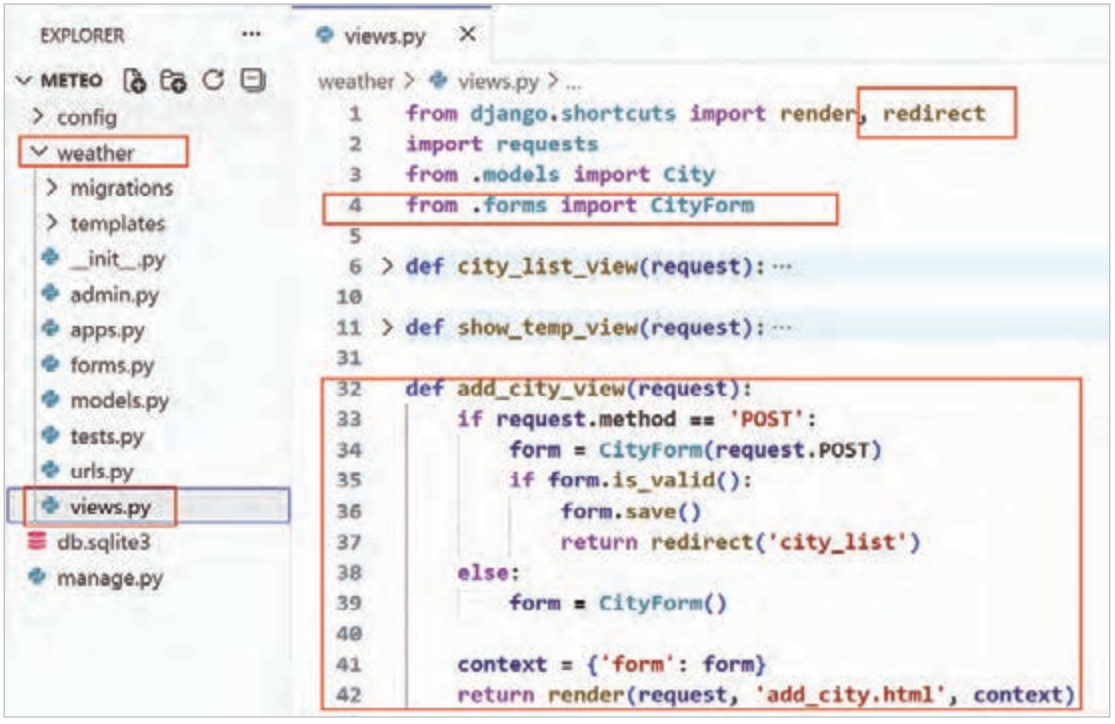


شکل ۶۵ــ افزودن add_city.html

توضیحات کدها:

یک فرم با متد از نوع post ایجاد شده است. این فرم طول و عرض جغرافیایی و همچنین نام شهر را از کاربر گرفته و برای سرور جنگو ارسال می‌کند تا در دیتابیس ذخیره شود.

مرحله سوم: در فایل views.py تابع add_city_view را همانند شکل ۶۶ اضافه کنید.



شکل ۶۶ـ ایجاد view جهت افزودن شهر

توضیحات کدها:

خط ۱: متد redirect درون این فایل import شده است.

خط ۴: کلاس CityForm از فایل forms.py درون این فایل import شده است.

خط ۳۳: اگر کاربر در صفحه «اضافه‌کردن شهر»، اطلاعات فرم را وارد کرده و دکمه «ذخیره شهر جدید» را کلیک کند، یک request از نوع post برای add_city_view ارسال می‌شود، در این خط نوع request با if بررسی شده است.

خط ۳۴: یک فرم از CityForm براساس نوع درخواست ارسال‌شدهٔ کاربر ایجاد و در متغیر form ذخیره شده است تا به کمک آن اطلاعاتی که کاربر به‌وسیله آن فرم ارسال کرده است، ذخیره شود.

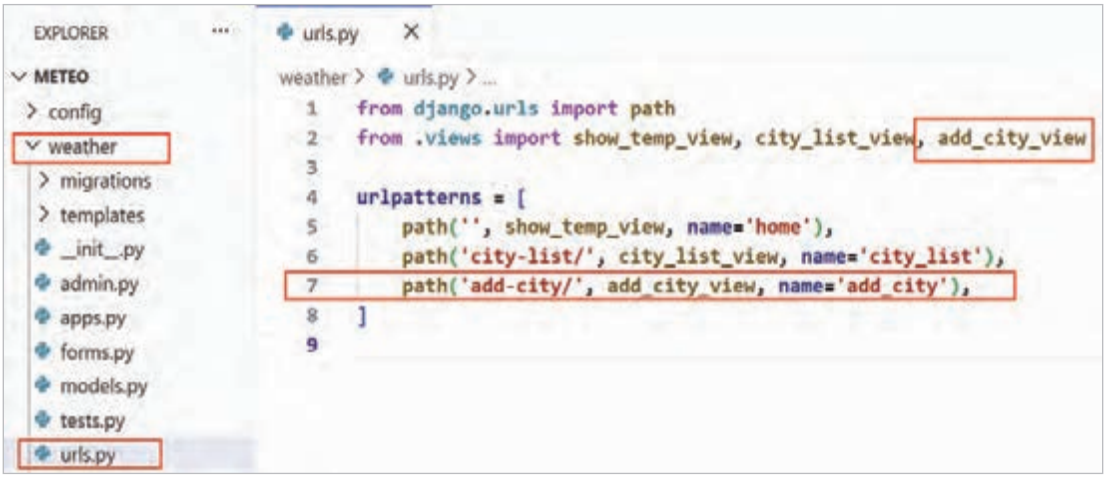
خط ۳۵: تابع is_valid فیلدهای وارد شده کاربر را بررسی می‌کند. در request ارسال کاربر، باید فیلدهای اجباری مقدار گرفته باشند و همچنین نوع فیلدها صحیح باشد.

خط ۳۶: در صورت معتبر بودن درخواست، اطلاعات در جدول city ذخیره خواهد شد.

خط ۳۷: پس از ذخیره موفق اطلاعات شهر، کاربر به صفحه لیست شهرها منتقل خواهد شد.

خط ۳۹: وقتی کاربر آدرس این صفحه را در مرورگر وارد می‌کند، یک درخواست از نوع GET اتفاق افتاده است، در این حالت، نوع request چیزی غیر از POST است، پس یک فرم جدید از CityForm ایجاد و در متغیر form ذخیره شده است تا فرم افزودن شهر به کاربر نمایش داده شود.

خط ۴۱: فرم ساخته شده، در دیکشنری context قرار گرفته و در خط بعد به add_city.html ارسال شده است.
مرحله چهارم: در فایل urls.py مربوط به weather app یک path جدید همانند شکل ۶۷ اضافه کنید.



شکل ۶۷ــ اضافه‌کردن path جدید برای افزودن شهر

توضیحات کدها:
یک path جدید جهت افزودن شهر اضافه شده است.
مرحله پنجم: در فایل show_temp.html یک لینک برای افزودن شهر، در خط ۱۰ همانند شکل ۶۸ اضافه کنید.



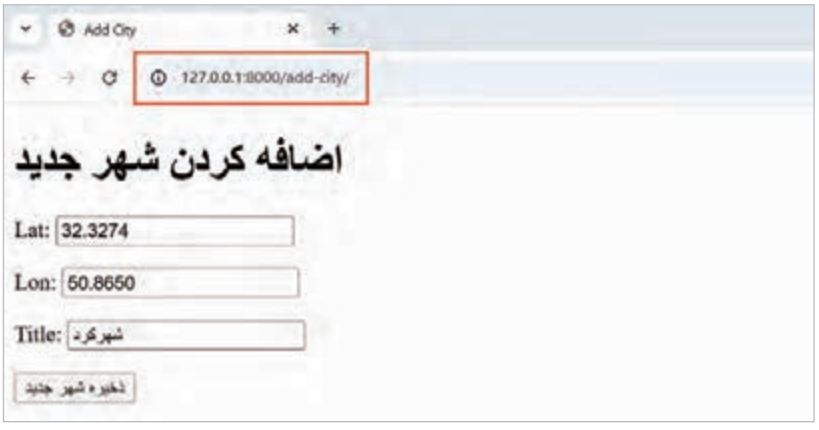
شکل ۶۸ــ ایجاد لینک به صفحه افزودن شهر

اکنون همه فایل‌ها را ذخیره و صفحه اصلی سایت را refresh کنید.
با دکمه [ورود به سایت] login کنید. لینک افزودن شهر برای شما نمایش داده خواهد شد (شکل ۶۹).



شکل ۶۹ــ نمایش لینک پس از login کاربر

بر روی لینک «افزودن شهر» کلیک کنید، سپس همانند شکل ۷۰، اطلاعات شهر موردنظر خود را وارد و دکمه ذخیره شهر جدید را کلیک کنید. در صورت ذخیره موفقِ این فرم، کاربر به صفحه لیست شهرها منتقل و شهر اضافه شده به انتهای لیست افزوده خواهد شد.

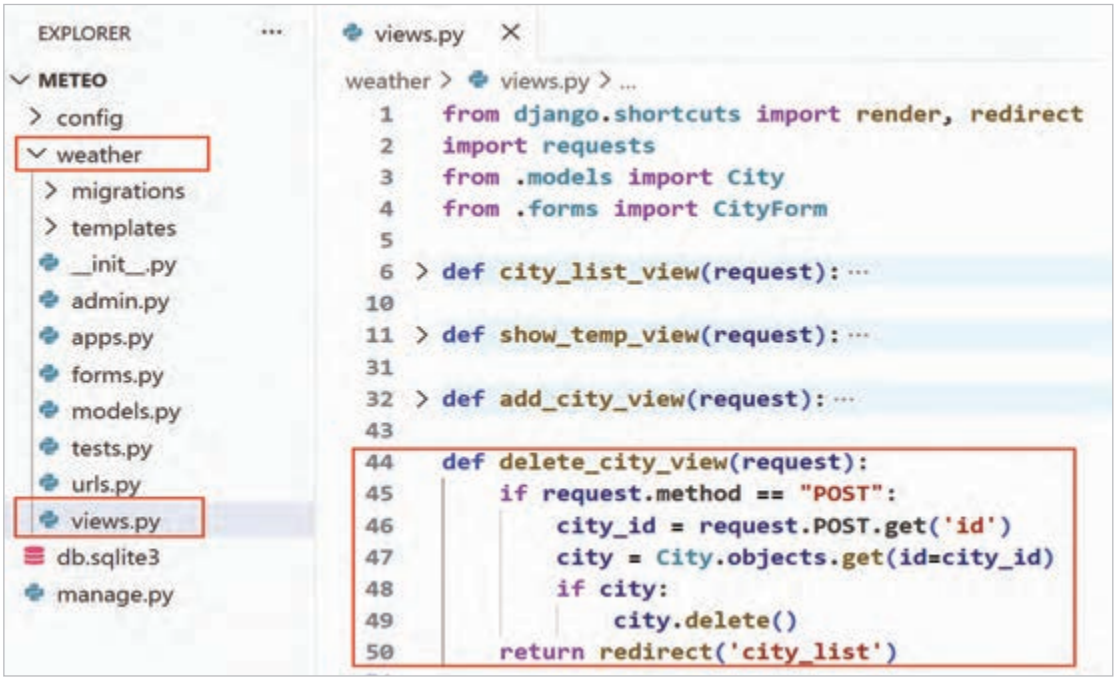


شکل ۷۰ــ فرم اضافه‌کردن شهر جدید

حذف شهر با کاربر عضو سایت

در این پروژه می‌توانید امکان حذف شهرها را به کاربر عضو سایت بدهید، این کار را در سه مرحله می‌توانید انجام دهید.

مرحله اول: یک view جدید با نام delete_city_view در views.py همانند شکل ۷۱ ایجاد کنید.



شکل ۷۱ـ ساخت view جهت حذف شهرها

توضیحات کدها:

خط ۴۹: دستور delete از ORM جنگو استفاده می‌کند و شهر انتخاب شده را از پایگاه‌داده حذف می‌کند.

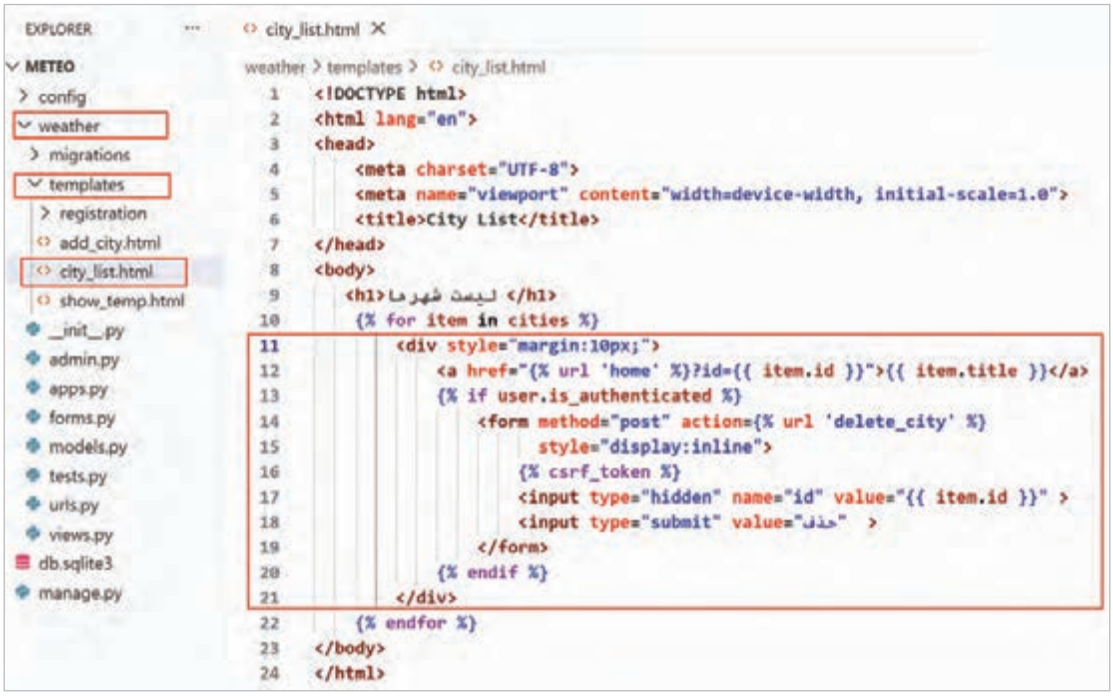
خط ۵۰: بعد از حذف‌کردن شهر، کاربر را به لیست شهرها منتقل می‌کند.

مرحله دوم: یک path جدید برای delete_city_view در urls.py همانند شکل ۷۲ اضافه کنید.



شکل ۷۲ـ افزودن path جدید برای حذف شهرها

مرحله سوم: فایل city_list.html را همانند شکل ۷۳ تغییر دهید.



شکل ۷۳ــ اضافه‌کردن دکمه حذف به city_list.html

توضیحات کدها:

خط ۱۳: این دستور باعث می‌شود دکمه حذف شهر فقط برای کاربر login شده نمایش داده شود.

خط ۱۴: یک فرم با متد از نوع POST ایجاد شده است. در action این فرم، به _delete city اشاره شده است، یعنی درخواست حذف به view حذف شهر ارسال خواهد شد.

خط ۱۷: آیدی شهر به‌صورت یک فیلد مخفی (hidden) اضافه شده است. آیدی شهر در خروجی نمایش داده نخواهد شد. فایل‌ها را ذخیره کنید، صفحه لیست شهرها را باز کنید. خروجی همانند شکل ۷۴ خواهد بود.



شکل ۷۴ــ دکمه حذف در لیست شهرها

با کلیک روی لینک حذف مقابل هر شهر، شهر موردنظر حذف شده و لیست شهرها به‌روزرسانی می‌شود. درصورتی‌که کاربر مهمان باشد، لینک حذف را مشاهده نخواهد کرد.

فعالیت

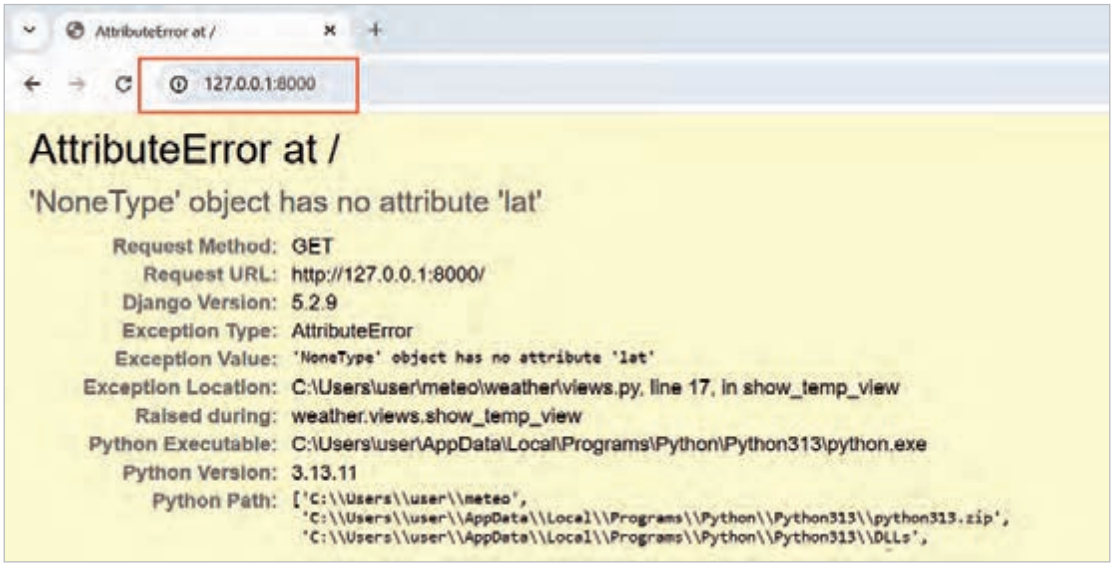
با دانشی که از پودمان‌های قبل کسب کرده‌اید، ظاهر سایت را با html و css و bootstrap زیبا کنید.

بهینه‌سازی و امن کردن پروژه

در این مرحله، برای بهینه‌سازی پروژه، ایرادها و باگ‌های امنیتی پروژه بررسی می‌شوند.

باگ کدنویسی شماره یک:

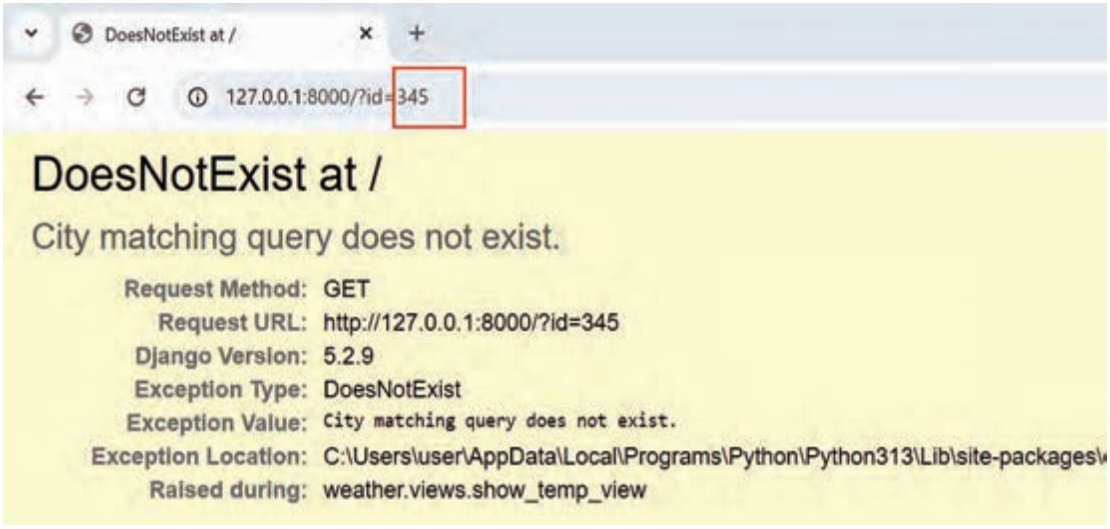
اگر هیچ شهری در جدول City وجود نداشته باشد یا همه شهرها حذف شده باشند، در صفحه اصلی سایت خطایی همانند شکل ۷۵ مشاهده خواهد شد.



شکل ۷۵ــ خطا در صورت خالی بودن جدول شهرها

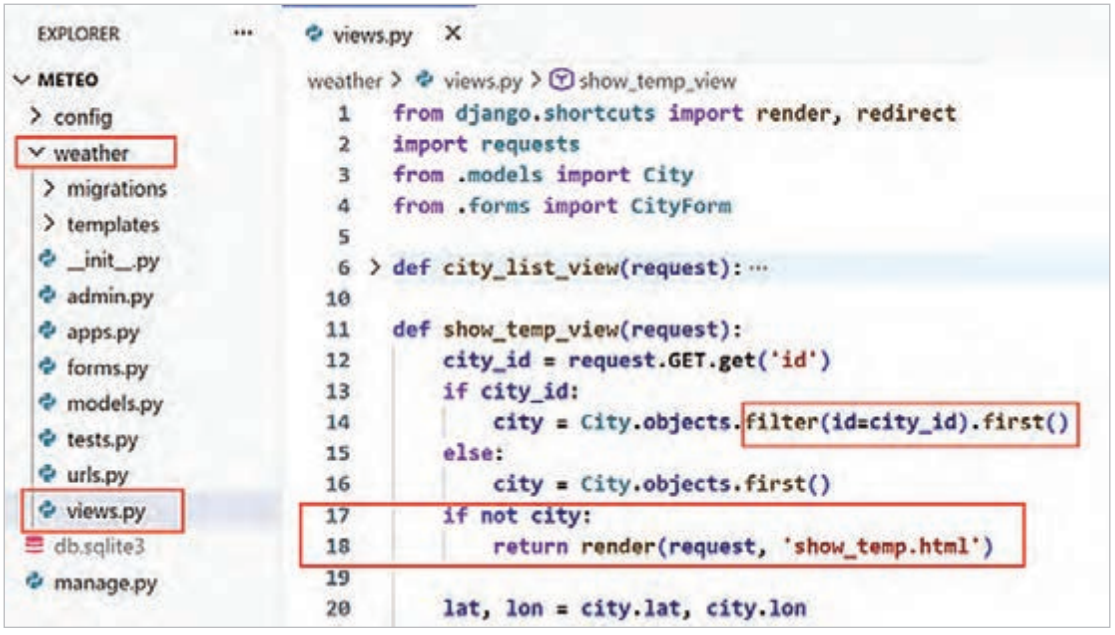
باگ کدنویسی دوم:

اگر کاربر id شهر را در نوار آدرس مرورگر، دستی تغییر دهد و id وارد شده در پایگاه‌داده وجود نداشته باشد، در صفحه اصلی سایت خطایی همانند شکل ۷۶ نمایش داده می‌شود.



شکل ۷۶ــ خطا در صورت اشتباه بودن آیدی شهر

راه‌حل باگ اول و دوم: برای رفع این ایراد، کد show_temp_view را همانند شکل ۷۷ تغییر دهید.



شکل ۷۷ــ رفع ایراد کد

توضیحات کدها:

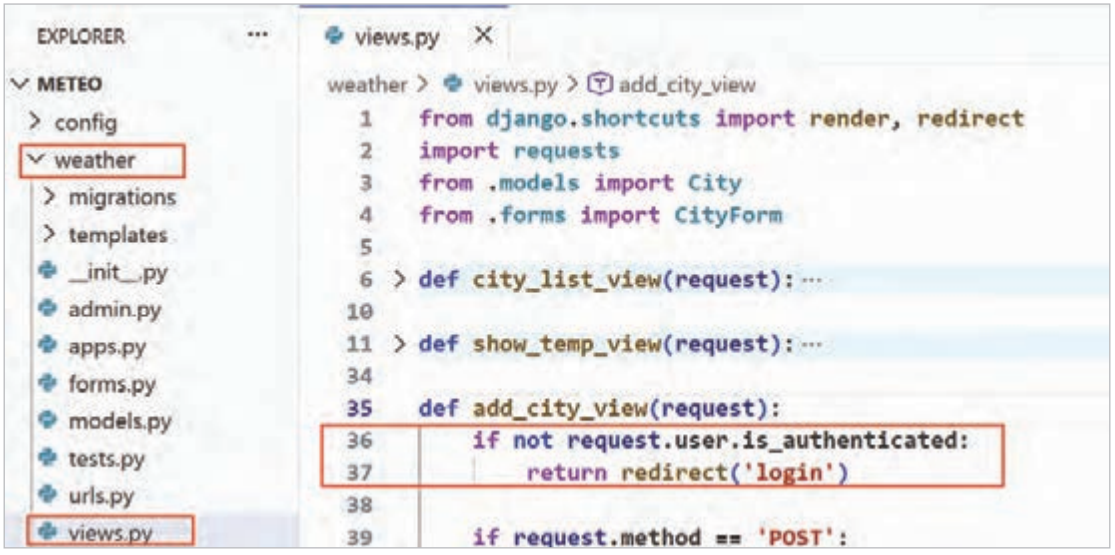
خط ۱۴: به‌جای استفاده از تابع get از تابع filter و first استفاده شده است. تابع get در صورتی خطا می‌دهد که مقدار موردنظر در دیتابیس وجود نداشته باشد. اما تابع filter این رفتار را ندارد و باعث بروز خطا نخواهد شد. پس **باگ دوم** را پوشش می‌دهد.

خط ۱۷: درصورتی‌که city مقدار نداشته باشد، درخواست به API آب‌وهوا ارسال نمی‌شود و مستقیماً show_temp.html نمایش داده می‌شود. این خط نیز باگ اول را پوشش می‌دهد، قبل از اصلاح کدها اگر هیچ شهری در دیتابیس وجود نداشت، مقدار city برابر None می‌شد. پس هنگام دریافت lat کد به خطا می‌خورد. با تغییراتی که اعمال شد، این باگ برطرف شده است.

باگ امنیتی:

اگر کاربر عضو سایت نباشد، اما آدرس صفحه افزودن شهر و حذف شهرها را داشته باشد، بدون login در سایت می‌تواند، شهر اضافه کرده و یا شهر حذف نماید.

راه‌حل باگ امنیتی: برای رفع این ایراد، کد add_city_view را همانند شکل ۷۸ تغییر دهید.



شکل ۷۸ــ رفع باگ امنیتی کد

توضیحات کدها:

خط ۳۶: درصورتی‌که کاربر login نکرده باشد، در خط بعدی به صفحه login منتقل خواهد شد. همین کد را می‌توانید برای delete_city_view هم استفاده کنید.

با این تغییرات، اگر کاربر login نباشد؛ ولی آدرس افزودن شهر و حذف شهر را در مرورگر وارد کند، به صفحه login منتقل خواهد شد.

کنجکاوی

بررسی کنید چه موارد دیگری می‌تواند این پروژه را بهبود دهد.

استاندارد عملکرد کار: هنرجو بتواند با ساخت Model در جنگو و استفاده از فرم های HTML ارتباط با پایگاه داده را پیاده‌سازی کند و اطلاعاتی که کاربران وارد می‌کنند را در جداول ذخیره و بازیابی نماید.

ردیف	مراحل کار	شرایط انجام کار (ابزار،مواد، تجهیزات، زمان، مکان و ...)	ابزارهای ارزشیابی	استاندارد (شاخص‌ها/داوری/نمره‌دهی)	نتایج ممکن	نمره کسب شده
۱	پایگاه‌داده، ORM و پنل مدیریت	PostgreSQL -DE: VS Code- فریم‌ورک جنگو ـ رایانه شخصی یا لپ‌تاپ با سیستم عامل ـ زمان ۲۵ دقیقه ـ مکان سایت رایانه	مشاهده عملی ـ پرسش کتبی ـ آزمون شفاهی	ـ توضیح تفاوت بین ساختار پایگاه‌داده و جداول آن ـ شرح وظیفه و اهمیت ORM در جنگو. ـ شناخت اجزای اصلی پنل مدیریت جنگو و کاربرد آن ـ موفقیت در تعریف صحیح مدل City با فیلدهای لازم (مانند نام شهر، دما و ...) ـ کار با ORM: ـ نوشتن کوئری‌های ساده با ORM برای خواندن (مثلاً City.objects.all()) داده‌ها. ـ موفقیت در ثبت City در پنل ادمین و امکان مشاهده، افزودن و ویرایش داده‌ها از طریق آن ـ وارد کردن چند رکورد نمونه (چند شهر) به جدول از طریق پنل ادمین. ـ انجام کنجکاوی‌ها ـ انجام کار با چند روش	۳	
				ـ موفقیت در تعریف صحیح مدل City با فیلدهای لازم (مانند نام شهر، دما و ...) ـ کار با ORM: ـ نوشتن کوئری‌های ساده با ORM برای خواندن (مثلاً City.objects.all()) داده‌ها. ـ موفقیت در ثبت City در پنل ادمین و امکان مشاهده، افزودن و ویرایش داده‌ها از طریق آن ـ وارد کردن چند رکورد نمونه (چند شهر) به جدول از طریق پنل ادمین.	۲	
				ـ توضیح تفاوت بین ساختار پایگاه‌داده و جداول آن ـ شرح وظیفه و اهمیت ORM در جنگو. ـ شناخت اجزای اصلی پنل مدیریت جنگو و کاربرد آن	۱	
۲	کار با داده‌ها و نمایش اطلاعات	PostgreSQL ـDE: VS Code ـ فریم‌ورک جنگو ـ رایانه شخصی یا لپ‌تاپ با سیستم عامل ـ زمان ۲۵ دقیقه ـ مکان سایت رایانه	مشاهده عملی ـ پرسش کتبی ـ آزمون شفاهی	ـ شناخت نقش views در پردازش درخواست‌ها و ارسال پاسخ ـ درک نحوه ارسال داده از view به template (قالب HTML) ـ آشنایی با مفاهیم اولیه تعامل کاربر (مانند کلیک روی لینک) ـ ایجاد یک view و template برای نمایش لیستی از نام شهرها که از پایگاه‌داده خوانده شده‌اند ـ پیاده‌سازی مکانیزم انتخاب شهر (مثلاً با لینک یا فرم) ـ ایجاد یک view که شهر انتخاب شده را دریافت کند ـ نمایش دمای مربوط به شهر انتخاب شده در صفحه ـ تعریف URLهای مناسب برای دسترسی به این قابلیت‌ها.	۳	
				ـ ایجاد یک view و template برای ـ نمایش لیستی از نام شهرها که از پایگاه‌داده خوانده شده‌اند ـ پیاده‌سازی مکانیزم انتخاب شهر (مثلاً با لینک یا فرم). ـ ایجاد یک view که شهر انتخاب شده را دریافت کند. ـ نمایش دمای مربوط به شهر انتخاب شده در صفحه ـ تعریف URLهای مناسب برای دسترسی به این قابلیت‌ها.	۲	
				ـ توضیح نقش views در پردازش درخواست‌ها و ارسال پاسخ ـ توضیح نحوه ارسال داده از view به template (قالب HTML) ـ توضیح مفاهیم اولیه تعامل کاربر (مانند کلیک روی لینک).	۱	
۳	احراز هویت و مدیریت دسترسی‌ها	PostgreSQL -DE: VS Code ـ فریم‌ورک جنگو ـ رایانه شخصی یا لپ‌تاپ با سیستم عامل ـ زمان ۲۵ دقیقه ـ مکان سایت رایانه	مشاهده عملی ـ پرسش کتبی ـ آزمون شفاهی	ـ توضیح با فرایند احراز هویت (Login/Logout) در جنگو ـ توضیح مفهوم سطوح دسترسی (Permissions) و چگونگی اعمال آن‌ها ـ شناخت مخاطرات امنیتی ـ دسترسی‌های غیرمجاز ـ راه‌اندازی صحیح سیستم ورود کاربران ـ اطمینان از اینکه صفحات افزودن و حذف شهر فقط برای کاربران وارد شده (عضو سایت) قابل دسترسی باشند ـ موفقیت در پیاده‌سازی قابلیت افزودن شهر جدید و حذف شهر موجود، فقط توسط کاربران وارد شده ـ انجام کنجکاوی ها	۳	
				راه‌اندازی صحیح سیستم ورود کاربران ـ اطمینان از اینکه صفحات افزودن و حذف شهر فقط برای کاربران وارد شده (عضو سایت) قابل دسترسی باشند ـ موفقیت در پیاده‌سازی قابلیت افزودن شهر جدید و حذف شهر موجود، فقط توسط کاربران وارد شده.	۲	
				ـ توضیح با فرایند احراز هویت ـ (Login/Logout) در جنگو. ـ توضیح مفهوم سطوح دسترسی (Permissions) و چگونگی اعمال آنها. ـ توضیح مخاطرات امنیتی و دسترسی‌های غیرمجاز ـ راه‌اندازی صحیح سیستم ورود کاربران.	۱	

۴	رفع اشکالات و بهینه‌سازی امنیتی	PostgreSQL -DE: VS Code- فریم‌ورک جنگو- رایانه شخصی یا لپ‌تاپ با سیستم عامل - زمان ۳۰دقیقه - مکان سایت رایانه	مشاهده عملی - پرسش کتبی - آزمون شفاهی	۳	- شناسایی انواع باگ‌های کدنویسی (منطقی، اجرایی) - درک اصول اولیه امنیت وب (مانند جلوگیری از تزریق کد، حملات CSRF و ...). آشنایی با مفاهیم اولیه بهینه‌سازی کد و عملکرد. - رفع باگ ۱ (جدول خالی): پیاده‌سازی منطق صحیح در view یا template برای مدیریت حالتی که جدول City خالی است، بدون ایجاد خطا. - رفع باگ ۲ (تغییر ID در URL): - پیاده‌سازی بررسی صحت id شهر در view قبل از تلاش برای نمایش اطلاعات آن، جهت جلوگیری از خطا یا نمایش اطلاعات نادرست. - رفع باگ امنیتی (دسترسی غیرمجاز): اطمینان کامل از اینکه هیچ کاربری بدون Login نمی‌تواند به صفحات افزودن/حذف شهر دسترسی پیدا کند و عملیات را انجام دهد. - ارائه کد تمیزتر و کارآمدتر (در صورت امکان، بسته به سطح دو	
				۲	رفع باگ ۱ (جدول خالی): پیاده‌سازی منطق صحیح در view یا template برای مدیریت حالتی که جدول City خالی است، بدون ایجاد خطا - رفع باگ ۲ (تغییر ID در URL): - پیاده‌سازی بررسی صحت id شهر در view قبل از تلاش برای نمایش اطلاعات آن، جهت جلوگیری از خطا یا نمایش اطلاعات نادرست. - رفع باگ امنیتی (دسترسی غیرمجاز): اطمینان کامل از اینکه هیچ کاربری بدون Login نمی‌تواند به صفحات افزودن/حذف شهر دسترسی پیدا کند و عملیات را انجام دهد.	
				۱	- شناسایی انواع باگ‌های کدنویسی (منطقی، اجرایی) - درک اصول اولیه امنیت وب (مانند جلوگیری از تزریق کد، حملات CSRF و ...). آشنایی با مفاهیم اولیه بهینه‌سازی کد و عملکرد.	
شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش:				۲	احراز همه شایستگی‌های غیرفنی	
				۱	عدم احراز هر یک از شایستگی‌های غیرفنی	
ارزشیابی کار (شایستگی انجام کار)						بلی ☐
						خیر ☐

توضیحات:

۱ـ معیار شایستگی انجام کار:

- کسب حداقل نمره ۲ از مراحل ۲ و ۳ (مراحل بحرانی)
- کسب حداقل نمره ۲ از بخش شایستگی‌های غیرفنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش
- کسب حداقل میانگین ۲ از مراحل کار

۲ـ تعریف سطوح شایستگی: صرف‌نظر از اینکه یک تکلیف کاری در چه سطحی از صلاحیت حرفه‌ای انجام می‌شود، در هر محیط کاری ممکن است انجام هر کار با کیفیت مشخصی مورد انتظار باشد. سطح شایستگی انجام کار، معیار اساسی ارزشیابی است.

مهارت (سطح سه): ماهر و قادر به آموزش و هدایت دیگران، توانایی برنامه‌ریزی و تحلیل، پاسخ‌گویی در برابر کارهای خود، سروکار داشتن با سطح وسیعی از کارها و فعالیت‌ها

تسلط (سطح چهار): خبرگی در انجام کار و آموزش دیگران، ایجاد نوآوری، سازگاری، عیب‌یابی، هدایت و راهنمایی دیگران.

منابع

۱ سایت مستندات رسمی پایتون نسخه ۳ https://docs.python.org/3/

۲ منبع جنگو: سایت مستندات رسمی جنگو https://docs.djangoproject.com/

۳ https://getbootstrap.com/

۴ https://developer.mozilla.org

۵ https://www.w3schools.com/