



آموزش نکته به نکته و
سوالات طبقه‌بندی شده

طراح سایت

ویژه **هنرستان**

✓ برنامه‌نویسی پایتون

✓ طراحی صفحات وب ایستا

✓ توسعه صفحات وب

✓ توسعه وب پایتون

✓ برنامه‌نویسی جنگو



کد: ۲۶۵۰۱



سرشناسه
عنوان و نام پدیدآور : دکتر مظاهر مقصودلو.
ویژه هنرستان / پدیدآورندگان: دکتر مظاهر مقصودلو;
ویراستار: سوده زارعی.
مشخصات نشر : تهران : چهارخونه
مشخصات ظاهری : ۴۰۰ ص.؛ مصور، جدول، نمودار؛ ۲۲×۲۹ س.م.
فروست : چهارخونه؛ ۲۶۵۰۱.
شابک : 978-600-305-236-9
وضعیت فهرست نویسی : فیپا
موضوع : طراحی سایت -- راهنمای آموزشی (متوسطه)
شناسه افزوده : انتشارات چهارخونه. دکتر مظاهر مقصودلو
رده بندی کنگره : QA۱۳۹
رده بندی دیویی : ۵۱۰/۷۶
شماره کتابشناسی ملی : ۱۰۵۹۲۴۷۹
اطلاعات رکورد کتابشناسی : فیپا

آموزش نکته به نکته و سوالات طبقه بندی شده طراح سایت

● ناشر: انتشارات چهارخونه

● پدیدآورندگان: دکتر مظاهر مقصودلو

● ویراستار: سوده زارعی

● صفحه آرایی: سمیه نجار

● لیتوگرافی: امیر گرافیک

● چاپ و صحافی: یگانه

● شمارگان: ۱۰۰۰ جلد

● قیمت و سال چاپ: جهت مشاهده قیمت بارکد را اسکن کنید.

● فروشگاه اینترنتی: www.4Khooneh.org



«کلیه حقوق برای ناشر محفوظ است و هرگونه نسخه برداری پیگرد قانونی دارد»

تلفن مرکز پخش: ۰۹۱۲-۶۲۰۰۰۲۶ - ۶۶۹۲۷۷۹۶ - ۶۶۹۲۸۱۷۱

جهت دریافت کتاب در تهران از طریق پیک و در شهرستانها از طریق پست با شماره تلفن ۶۶۹۲۸۰۲۹ (۰۲۱) تماس حاصل فرمایید یا از طریق سایت به صورت اینترنتی اقدام نمایید.

مقدمه کتاب

در دنیای امروز که فناوری اطلاعات و اینترنت بخش جدایی‌ناپذیر زندگی انسان‌ها شده‌اند، طراحی و توسعه وب به یکی از مهارت‌های ارزشمند، کاربردی و آینده‌دار تبدیل شده است. یادگیری این مهارت، علاوه بر آشنایی با دنیای برنامه‌نویسی، به پرورش خلاقیت و تقویت تفکر منطقی کمک می‌کند و زمینه را برای ورود به دنیای گسترده فناوری فراهم می‌کند. کتاب درسی «طراح سایت» با هدف آموزش مفاهیم و مهارت‌های موردنیاز این حوزه، هنرجویان را با مباحث گوناگون برنامه‌نویسی و طراحی وب آشنا می‌سازد اما برای یادگیری عمیق‌تر، مرور مؤثر مطالب و آمادگی کامل برای آزمون‌ها، برخورداری از یک منبع کمک‌آموزشی جامع و هدفمند می‌تواند بسیار راهگشا باشد.

کتاب کمک‌آموزشی «کلید بیست طراح سایت» با رویکردی آموزشی، منسجم و آزمون‌محور و به عنوان اولین کتاب کمک‌آموزشی جامع طراح سایت تدوین شده است تا مسیر یادگیری را برای هنرجویان ساده‌تر، روشن‌تر و اثربخش‌تر کند. تلاش کرده‌ایم مطالب کتاب درسی را مطابق با سرفصل‌ها و واحدهای یادگیری، به زبانی روان و به‌صورت تشریحی ارائه کنیم تا هنرجو بتواند مفاهیم را به‌درستی درک کند، ارتباط میان مباحث را بشناسد و آموخته‌های خود را با اطمینان بیشتری به کار گیرد.

یکی از ویژگی‌های مهم این اثر، توجه ویژه به نکات کلیدی، جزئیات مهم و ظرایف کنکوری است. نکاتی که گاهی در مطالعه سطحی از چشم دور می‌مانند، اما می‌توانند نقش مهمی در پاسخ‌گویی دقیق به پرسش‌های امتحانی و تستی داشته باشند. در این کتاب، علاوه بر آموزش مفاهیم، بر نکات مهم و قابل‌توجه هر مبحث تأکید شده است تا هنرجو بتواند با دیدی دقیق‌تر مطالعه کند و برای مواجهه با انواع سؤال‌ها آمادگی بیشتری به‌دست آورد.

برای آنکه یادگیری صرفاً به مطالعه مطالب محدود نشود، در پایان هر واحد یادگیری، مجموعه‌ای از سؤالات تشریحی همراه با پاسخ ارائه شده است تا هنرجو بتواند میزان درک خود را ارزیابی کند، نقاط ضعفش را بشناسد و آموخته‌های خود را تثبیت کند. همچنین، در پایان هر پودمان، سؤالات تستی با پاسخ‌های تشریحی و در ادامه، سه مرحله آزمون جامع همراه با پاسخ تشریحی در نظر گرفته شده است تا هنرجو با مرور مطالب، سنجش میزان تسلط و شناسایی مباحث نیازمند تمرین بیشتر، آمادگی خود را برای آزمون‌های پیش‌رو افزایش دهد. ساختار این کتاب به‌گونه‌ای طراحی شده است که هنرجو بتواند از آن در طول سال تحصیلی، هنگام مطالعه هر درس، در زمان مرور و جمع‌بندی و همچنین در دوران آمادگی برای آزمون استفاده کند. توضیحات تشریحی، نکات مهم، پرسش‌های خودآزمایی، تست‌های پودمانی و آزمون‌های جامع، مجموعه‌ای هماهنگ از ابزارهای یادگیری را در اختیار او قرار می‌دهند.

هدف ما تنها ارائه پاسخ سؤالات نیست بلکه کمک به درک بهتر مفاهیم، تقویت قدرت تحلیل و ایجاد آمادگی واقعی برای پاسخ‌گویی به پرسش‌هاست. باور داریم که یادگیری مؤثر زمانی شکل می‌گیرد که مطالعه، تمرین، ارزیابی و رفع اشکال در کنار یکدیگر قرار گیرند. به همین دلیل، این کتاب می‌کوشد فراتر از یک مجموعه سؤال و جواب، همراهی آموزشی برای هنرجویان باشد.

این اثر با هدف ارائه یک منبع کمک‌آموزشی جامع و منسجم برای درس طراح سایت تهیه شده است و امیدواریم بتواند به هنرجویان، هنرآموزان و همه علاقه‌مندان این حوزه در مسیر یادگیری بهتر و دستیابی به نتایج مطلوب کمک کند.

این کتاب را گامی در مسیر یادگیری عمیق‌تر، آمادگی بهتر و موفقیت شما در دنیای طراحی و توسعه وب می‌دانیم.

فهرست مطالب

پودمان اول: برنامه‌نویسی پایتون

واحد یادگیری ۱: مدل‌سازی داده	۶
سؤال امتحانی (واحد یادگیری اول)	۴۹
واحد یادگیری ۲: کار با توابع و ماژول‌ها	۵۱
سؤال امتحانی (واحد یادگیری دوم)	۷۸
تست‌های پودمان اول	۸۰
آزمون جامع ۱ پودمان اول	۹۱
آزمون جامع ۲ پودمان اول	۹۵
آزمون جامع ۳ پودمان اول	۹۹

پودمان دوم: طراحی صفحات وب ایستا

واحد یادگیری ۱: طراحی ساختار صفحات وب با HTML	۱۰۴
سؤال امتحانی (واحد یادگیری اول)	۱۲۹
واحد یادگیری ۲: طراحی ظاهر صفحات وب با CSS	۱۳۱
سؤال امتحانی (واحد یادگیری دوم)	۱۵۷
تست‌های پودمان دوم	۱۵۹
آزمون جامع ۱ پودمان دوم	۱۷۱
آزمون جامع ۲ پودمان دوم	۱۷۵
آزمون جامع ۳ پودمان دوم	۱۷۹

پودمان سوم: توسعه صفحات وب

واحد یادگیری ۱: توسعه صفحات وب تعاملی با بوت‌استرپ	۱۸۴
سؤال امتحانی (واحد یادگیری اول)	۲۱۰
واحد یادگیری ۲: طراحی صفحات وب تعاملی با جاوا اسکریپت	۲۱۲
سؤال امتحانی (واحد یادگیری دوم)	۲۵۷
تست‌های پودمان سوم	۲۵۹

آزمون جامع ۱ پودمان سوم	۲۶۷
آزمون جامع ۲ پودمان سوم	۲۷۱
آزمون جامع ۳ پودمان سوم	۲۷۶

پودمان چهارم: توسعه وب پایتون

واحد یادگیری ۱: تولید برنامه‌های شیء‌گرا	۲۸۱
سؤال امتحانی (واحد یادگیری اول)	۳۰۲
واحد یادگیری ۲: توسعه برنامه شیء‌گرای	۳۰۴
سؤال امتحانی (واحد یادگیری دوم)	۳۱۷
تست‌های پودمان چهارم	۳۱۹
آزمون جامع ۱ پودمان چهارم	۳۳۰
آزمون جامع ۲ پودمان چهارم	۳۳۴
آزمون جامع ۳ پودمان چهارم	۳۳۸

پودمان پنجم: برنامه‌نویسی جنگو

واحد یادگیری ۱: ایجاد وب‌اپلیکیشن	۳۴۳
سؤال امتحانی (واحد یادگیری اول)	۳۵۷
واحد یادگیری ۲: توسعه وب‌اپلیکیشن	۳۵۹
سؤال امتحانی (واحد یادگیری دوم)	۳۸۲
تست‌های پودمان پنجم	۳۸۴
آزمون جامع ۱ پودمان پنجم	۳۸۹
آزمون جامع ۲ پودمان پنجم	۳۹۳
آزمون جامع ۳ پودمان پنجم	۳۹۷

پودمان اول | برنامه‌نویسی پایتون

این پودمان به چه درد ما می‌خوره؟

شاید با خودتون بگید: «برنامه‌نویسی پایتون و مدل‌سازی داده رو یاد بگیریم که چی؟!» 🤖

بیاید از دنیای واقعی شروع کنیم!

هر برنامه‌ای که روی کامپیوتر یا گوشی اجرا می‌شه، یه روز فقط یک ایده بوده؛ ایده‌ای که با کمک برنامه‌نویسی به یک نرم‌افزار واقعی تبدیل شده. اما برای تبدیل ایده‌هامون به برنامه‌های کاربردی، از کجا باید شروع کنیم؟ اصلاً چرا پایتون رو یاد بگیریم، نه یک زبان برنامه‌نویسی دیگه؟

پایتون چه چیزی داره که اون رو به انتخاب جذابی تبدیل می‌کنه؟

پایتون به خاطر ساختار خوانا و دستورهای نسبتاً ساده‌اش، برای شروع برنامه‌نویسی انتخاب مناسبیه. لازم نیست از همون اول درگیر قواعد پیچیده بشید؛ می‌تونید راحت‌تر روی منطق برنامه‌نویسی و حل مسئله تمرکز کنید. از طرفی، پایتون فقط برای شروع نیست! از توسعه وب و تحلیل داده گرفته تا هوش مصنوعی و خودکارسازی کارها، در زمینه‌های مختلفی کاربرد داره. همین گستردگی باعث می‌شه چیزهایی که یاد می‌گیرید، در مسیرهای مختلف برنامه‌نویسی به کارتون بیان.

توی این پودمان یاد می‌گیرید چطور داده‌ها رو سازمان‌دهی و مدیریت کنید، با استفاده از توابع کدهایی مرتب‌تر و قابل استفاده مجدد بنویسید و با کمک ماژول‌ها از قابلیت‌های آماده برای توسعه برنامه‌هاتون استفاده کنید.

اما این تازه شروع ماجراست! این مهارت‌ها پایه‌ای برای قدم‌های بعدی شما در دنیای برنامه‌نویسی هستن؛ از طراحی صفحات وب گرفته تا ساخت وب‌اپلیکیشن‌هایی که می‌تونن اطلاعات رو مدیریت کنن و به درخواست‌های کاربران پاسخ بدن.

پس آماده باشید! قراره با یکی از کاربردی‌ترین زبان‌های برنامه‌نویسی شروع کنید، منطق برنامه‌نویسی رو یاد بگیرید و قدم‌به‌قدم به کسی تبدیل بشید که می‌تونه ایده‌هاش رو به برنامه‌های واقعی تبدیل کنه.

فهرست مطالب

۱ درس‌نامه به صورت میکرو طبقه‌بندی شده

۲ نمونه سؤالات میکرو طبقه‌بندی شده

۳ پاسخ‌نامه تشریحی نمونه سؤالات

۴ سؤالات چهارگزینه‌ای پودمان اول

۵ آزمون‌های جامع پودمان اول

واحد یادگیری ۱: مدل سازی داده

آشنایی با پایتون و محیط برنامه نویسی

پایتون (Python) یک زبان برنامه نویسی سطح بالا، مفسری و همه منظوره است که در سال ۱۹۹۱ توسط «گیدو ون روسوم» طراحی و منتشر شد. نام این زبان از گروه کمدی بریتانیایی Monty Python گرفته شده است. هدف اصلی از طراحی این زبان، فراهم کردن ابزاری ساده، خوانا و در عین حال قدرتمند برای برنامه نویسی بود که حتی افراد غیرمتخصص بتوانند با آن ارتباط برقرار کنند. از مهم ترین کاربردهای پایتون می توان به موارد زیر اشاره کرد:

۱- علم داده (Data Science) با استفاده از کتابخانه هایی مانند NumPy و Pandas

۲- یادگیری ماشین با استفاده از کتابخانه هایی مانند TensorFlow و PyTorch

۳- توسعه وب با استفاده از چارچوب هایی مانند Django و Flask

۴- اتوماسیون و نوشتن اسکریپت برای کارهای تکراری

۵- توسعه بازی با استفاده از Pygame

۶- پردازش تصویر با استفاده از OpenCV

۷- توسعه اپلیکیشن های دسکتاپ

نکته ۱: پایتون یک زبان برنامه نویسی «مفسری» (Interpreted) است، یعنی کدهای زبان پایتون به صورت خط به خط، توسط مفسر (Interpreter) اجرا می شود، نه این که یک جا به کد ماشین تبدیل گردد.

نکته ۲: پسوند استاندارد فایل های پایتون py است. برنامه های پایتون را می توان با پسوند py در هر ویرایشگر متنی ایجاد کرد و سپس آن ها را برای اجرا به مفسر پایتون ارسال کرد.

محیط توسعه (IDE/IDLE)

محیط توسعه یکپارچه (IDE) نرم افزاری است که مجموعه ای از ابزارهای ضروری برای نوشتن، ویرایش، اجرا و اشکال زدایی کد را در یک رابط کاربری واحد فراهم می کند. یک IDE معمولاً شامل ویرایشگر متن با برجسته سازی نحوی (Syntax Highlighting)، تکمیل خودکار کد، اجرا کننده و دیباگر است.

IDLE (Integrated Development and Learning Environment) محیط رسمی پایتون است که به طور پیش فرض همراه با نصب پایتون عرضه می شود. این محیط از دو بخش اصلی تشکیل شده است:

۱- Shell Window: برای اجرای تعاملی دستورات

۲- Editor Window: برای نوشتن اسکریپت های کامل

در کنار IDLE، گزینه های دیگری نیز وجود دارند که عبارت اند از: Thonny که برای افراد مبتدی بسیار مناسب است. PyCharm یک IDE حرفه ای با قابلیت های پیشرفته است. Visual Studio Code (VSCode) یک ویرایشگر سبک وزن و رایگان است که با افزونه Python بسیار قدرتمند می شود. Jupyter Notebook برای تحلیل داده و آموزش بسیار رایج است.

برای شروع کار با IDLE کافی است پایتون را از python.org دانلود و نصب کنید. در Shell Window، علامت >>> نشان دهنده آمادگی برای دریافت دستور است.

اجرای برنامه‌های ساده در محیط IDLE

اجرای برنامه در IDLE دو مسیر اصلی دارد:

- ۱- حالت تعاملی (Interactive Mode): این حالت در Shell Window اتفاق می‌افتد، کاربر هر دستور را تایپ کرده و Enter می‌زند و مفسر بلافاصله نتیجه را نمایش می‌دهد. این حالت برای آزمایش سریع ایده‌ها مناسب است.
- ۲- حالت اسکریپت (Script Mode): در این حالت از منوی File گزینه New File انتخاب می‌شود، کد در ویرایشگر (Editor Window) نوشته می‌شود و سپس فایل با کلیدهای Ctrl+S ذخیره می‌شود. (پسوند .py) برای اجرای برنامه می‌توان از کلید F5 یا گزینه Run > Run Module استفاده کرد. این حالت برای برنامه‌های چندخطی ضروری است. هنگام بروز خطا، IDLE یک پیام خطا در Shell Window نمایش می‌دهد که شامل نوع خطا (مانند SyntaxError)، شماره خطی که خطا در آن رخ داده و توضیح مختصر درباره خطا است. IDLE همچنین خط مشکل‌دار را در ویرایشگر با رنگ قرمز مشخص می‌کند. قابلیت Color Coding در IDLE کمک بزرگی در یادگیری است: کلمات کلیدی (if, for, def) به رنگ نارنجی، رشته‌های متنی به رنگ سبز، کامنت‌ها به رنگ قرمز و کدهای معمولی به رنگ سیاه نمایش داده می‌شوند.



نکته ۱:

در زبان پایتون برای نام فایل‌ها و پوشه‌ها از حروف فارسی استفاده نکنید.

مثال ۱:

تکه برنامه زیر، اجرای یک برنامه پایتون در محیط Editor Window را نمایش می‌دهد و اجرای

آن توسط کلید F5 انجام می‌شود.



```
print("=== Welcome to Python IDLE ===")
print("This script runs from the Editor Window.")
print()
```

مثال ۲:

تکه برنامه زیر، اجرای یک برنامه پایتون در محیط Shell Window را نمایش می‌دهد که با

فشاردن کلید Enter اجرای خط به خط هر دستور در محیط کنسول نمایش داده می‌شود.



```
>>> 2 + 3
5
>>> "hello".upper()
'HELLO'
>>> type(42)
<class 'int'>
```



نکته ۲:

برای اجرای یک فایل کامل در IDLE Editor، از کلید میانبر Fn+F5 یا F5 یا منوی Run > Run

Module استفاده می‌شود؛ نتیجه اجرا در پنجره Shell نمایش داده می‌شود.

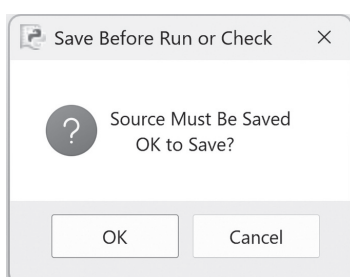


نکته ۳:

پیش از اجرای یک فایل جدید (که هنوز ذخیره نشده)،

IDLE به‌طور خودکار یک کادر محاوره‌ای برای ذخیره فایل به‌صورت زیر

نمایش می‌دهد؛ اجرای برنامه بدون ذخیره فایل ممکن نیست.



نکته ۴: برای خروج از محیط IDLE می‌توان از دستور `exit()` و یا کلیدهای ترکیبی `Ctrl+D` یا دکمه `Close` پنجره مربوطه استفاده کرد.

درک تفاوت بین خط فرمان و اجرای فایل

خط فرمان (Command Line یا Terminal) یک رابط متنی است که به کاربر اجازه می‌دهد دستورات را مستقیماً به سیستم‌عامل ارسال کند. برای اجرای پایتون از خط فرمان، ابتدا باید پایتون نصب باشد. با تایپ کردن `python` یا در برخی سیستم‌ها `python3` وارد حالت تعاملی می‌شوید. تفاوت اصلی بین اجرا از Terminal و IDLE در این است که Terminal یک محیط خالص و بدون ابزارهای اضافی است؛ هیچ رنگ آمیزی کد، تکمیل خودکار یا ویرایشگر یکپارچه‌ای ندارد. برای اجرای یک فایل، ابتدا با دستور `cd` به پوشه حاوی فایل رفته، سپس `python filename.py` را تایپ کنید. حالت تعاملی نتایج را ذخیره نمی‌کند. با هر بار بستن مفسر، همه کدها از بین می‌روند. به همین دلیل برای هر کار جدی باید کد را در فایل `.py` ذخیره کرده و آن را اجرا کرد.

نکته ۱: در حالت تعاملی (Shell/خط فرمان تعاملی)، مقدار هر عبارت به‌طور خودکار نمایش داده می‌شود، حتی بدون `print()`. اما درون یک فایل اجراشده، اگر دستوراتی را بدون `print()` بخواهیم اجرا کنیم آنگاه هیچ خروجی برای دستورات بدون `print()` نمایش داده نخواهد شد.

مثال ۱: اگر تکه برنامه زیر را در فایلی با پسوند `.py` ذخیره کرده باشیم و بخواهیم آن را از طریق خط فرمان اجرا کنیم آنگاه خروجی این تکه برنامه چه خواهد بود؟



```
print ("Python")
3*4
5-4*2+7
```

در تابع `print` اول، ابتدا مقدار `Python` در خروجی چاپ می‌شود و مکان‌نما به ابتدای سطر بعدی منتقل می‌شود و با توجه به اینکه دستورات سطر دوم و سوم در تابع `print()` قرار نگرفته‌اند و تکه کد مربوطه در حالت تعاملی قرار ندارد بلکه در فایل پایتون ذخیره و اجرا شده است بنابراین چیزی در خروجی برای این دوسطر چاپ نخواهد شد و خروجی نهایی به‌صورت زیر خواهد بود:



Python

قوانین اصلی نوشتاری (Syntax)

Syntax یا نحو، مجموعه قواعد دقیقی است که نحوه نوشتن صحیح کد را تعریف می‌کند. هر زبان برنامه‌نویسی دستور زبان خاص خود را دارد و کوچک‌ترین انحراف از قواعد زبان برنامه‌نویسی، منجر به خطا می‌شود. برخی از ویژگی‌های مهم Syntax پایتون عبارتند از:

- ۱- عدم نیاز به نقطه‌ویرگول (;) در پایان دستورات.
- ۲- استفاده از تورفتگی به‌جای آکولاد {}.
- ۳- حساس بودن به حروف بزرگ و کوچک (Case-Sensitive) یعنی `name` و `Name` دو متغیر متفاوت‌اند.

۴- امکان نوشتن کامنت یک سطر با علامت #.

۵- کامنت‌های چندخطی با رشته‌های سه گیومه‌ای نوشته می‌شوند.

۶- برای ادامه یک دستور طولانی در خط بعد، می‌توان از بک اسلش (\) استفاده کرد یا عبارت را داخل پرانتز قرار داد که به طور ضمنی ادامه خط را ممکن می‌کند.

تذکر: کامنت گذاری بخشی از Syntax زبان پایتون است که مفسر پایتون آن را نادیده می‌گیرد و در خروجی آن را اجرا نمی‌کند.

مثال ۱: اگر تکه برنامه زیر را در فایل با پسوند .py ذخیره کرده باشیم و بخواهیم آن را از طریق خط فرمان اجرا کنیم آنگاه خروجی این تکه برنامه چه خواهد بود؟



```
long_sum = 1 + 2 + 3 + \
           4 + 5 + 6
result = (100 +
          200 +
          300)
print(long_sum)
print(result)
```

در متغیر long_sum، با توجه به اینکه عبارت طولانی بوده، قسمتی از عبارت به سطر بعدی منتقل شده و در انتهای سطر اول از علامت \ استفاده شده است که تولید خطا نمی‌کند. همچنین برای متغیر result، نیز کل عبارت داخل پرانتز قرار گرفته و این عمل اجازه می‌دهد عبارت را در چندین سطر بنویسیم بنابراین این متغیرها تولید خطا نمی‌کنند و syntax زبان پایتون می‌باشد و در نهایت در توابع print() این متغیرها در خروجی چاپ شده‌اند و خروجی نهایی به صورت زیر خواهد بود:



```
21
600
```

نقش تورفتگی در بلاک‌های کد

تورفتگی (Indentation) در پایتون یک ویژگی اجباری است. منظور از تورفتگی، فاصله‌هایی است که در ابتدای یک خط کد قرار می‌گیرد. در زبان‌هایی مانند C یا Java، آکولاد {} بلاک‌های کد را مشخص می‌کند. اما در پایتون، تورفتگی خود تعریف کننده بلاک است. یک بلاک کد پس از دستوری که با دونقطه (:) تمام می‌شود آغاز می‌شود. دستوراتی مانند if، elif، else، for، while، def، class و with همگی با دونقطه پایان می‌یابند. همه دستورات در یک بلاک باید دقیقاً به میزان یکسان تورفته باشند.

نکته ۱: میزان استاندارد و توصیه شده تورفتگی در پایتون (طبق PEP8) چهار فاصله (۴ Space) است، نه یک تب؛ اگرچه از نظر فنی، هر مقدار ثابتی از تورفتگی پذیرفته می‌شود، اما به شرط یکنواختی در کل همان بلوک. رعایت صحیح تورفتگی یعنی تمام دستورات در یک بلاک دقیقاً با تعداد یکسانی از فاصله تورفتگی داشته باشند. خطاهای تورفتگی دو نوع اصلی دارند:

۱- IndentationError: این خطا زمانی رخ می‌دهد که پایتون انتظار تورفتگی دارد اما پیدا نمی‌کند یا بالعکس،

۲- TabError: این خطا از ترکیب Tab و Space ناشی می‌شود.

درک سطح تورفتگی برای نوشتن کد صحیح ضروری است. سطح ۰ برای کدهای اصلی خارج از همه بلاک‌ها است، سطح ۱ (۴ فاصله) برای بدنه اولین بلاک و سطح ۲ (۸ فاصله) برای بلاک‌های تودرتو. هر سطح با ۴ فاصله بیشتر از سطح قبل مشخص می‌شود. IDLE پس از هر دونقطه با Enter به‌طور خودکار ۴ فاصله ایجاد می‌کند.

نکته ۲: تمام خطوط هم‌سطح در یک بلوک، باید دقیقاً میزان تورفتگی یکسانی داشته باشند؛ ترکیب نامناسب تب (Tab) و فاصله (Space) در یک بلوک باعث خطای IndentationError می‌شود.

نکته ۳: شایع‌ترین خطاها در زبان پایتون عبارتند از:

۱- فراموش کردن تورفتگی پس از دونقطه (IndentationError: expected an indented block).

۲- تورفتگی اضافی روی دستوری که نباید تورفته باشد (IndentationError: unexpected indent).

۳- ناهماهنگی تورفتگی در داخل یک بلاک.

نوشتن چندین دستور متوالی

وقتی یک برنامه پایتون اجرا می‌شود، مفسر دستورات را به‌ترتیب از بالا به پایین اجرا می‌کند؛ این الگو را جریان ترتیبی (Sequential Execution) می‌نامند. نتیجه دستورات قبلی می‌تواند روی دستورات بعدی تأثیر بگذارد. هر دستور معمولاً در خط جداگانه‌ای نوشته می‌شود. ترتیب دستورات اهمیت حیاتی دارد. اگر قبل از تعریف متغیر از آن استفاده کنیم، پایتون خطای NameError می‌دهد. به همین دلیل همیشه باید قبل از استفاده از متغیر، آن را تعریف کرد.

نکته ۱: برای نوشتن چند دستور مستقل در یک سطر، می‌توان از سمی‌کالن (;) استفاده کرد؛ اگرچه این کار در پایتون معتبر است، اما طبق PEP۸ توصیه نمی‌شود.

مثال ۱: خروجی تکه برنامه زیر چیست؟



```
print("Python"); print(3.12)
```

در این تکه برنامه دو تابع print() در یک سطر استفاده شده‌اند که با علامت ; از یکدیگر جدا شده‌اند بنابراین خطا صادر نمی‌شود و با توجه به اینکه بعد از چاپ عبارت توسط تابع print() مکان‌نما به ابتدای سطر بعدی منتقل می‌شود بنابراین خروجی نهایی به‌صورت زیر خواهد بود:



```
Python
3.12
```

ساختار کلی یک اسکریپت ساده

یک اسکریپت پایتون حرفه‌ای از بخش‌های مشخصی تشکیل شده است. بخش اول Docstring ماژول است. یک رشته سه‌گیومه‌ای در ابتدای فایل که هدف، نویسنده و نسخه اسکریپت را توضیح می‌دهد. بخش دوم importها هستند که همه ماژول‌های موردنیاز در ابتدای فایل ذکر می‌شوند. بخش سوم تعریف ثابت‌های سطح ماژول است که با UPPER_CASE نوشته می‌شوند. بخش چهارم تعریف توابع و کلاس‌ها با Docstring کامل هر یک از توابع و کلاس‌ها می‌باشد. بخش پنجم، بلوک اجرایی اصلی است که داخل main قرار می‌گیرد. این بلوک فقط زمانی اجرا می‌شود که فایل مستقیماً اجرا شود نه import. ساختار منظم اسکریپت فواید عملی زیادی دارد که عبارت است از: درک سریع‌تر توسط کاربران، اشکال‌زدایی راحت‌تر، قابلیت استفاده مجدد بیشتر و نگهداری و توسعه آینده کد ساده‌تر.