



فیلش هم اینجاست

پایتون نوشتن کدها با

۲ Python Programming

فاطمه حیدری هانی (هنر آموز)
فاطمه نیک بخش (هنر جو)

با توجه به اینکه دنیا بیش از پیش به سوی فناوری‌های جدید پیش می‌رود، این نیاز احساس می‌شود که دانش‌آموزان و دانشجویان نیز بیش از گذشته با نحوه کارکردن این فناوری‌ها آشنا شوند. یکی از راه‌های این آشنایی، آموزش اصول برنامه‌نویسی است. در این مطلب در هشت شماره به آموزش مقدماتی زبان پایتون می‌پردازیم. در شماره اول با زبان پایتون و مزایا و معایب آن آشنا شدیم و نحوه نصب برنامه پایتون و اجرای برنامه‌های آن در رایانه را یاد گرفتیم. در این شماره با مفهوم داده، متغیر، مقداردهی آن و روش کار و کاربرد تابع‌های پرینت (چاپ) و تایپ (حروف نگاری) آشنا خواهیم شد.

نحوه نوشتن کدها با پایتون و چگونگی ذخیره‌سازی آن‌ها

دنباله‌ای از دستورات که انجام کارها روی یک دستگاه رایانه‌ای را بر عهده می‌گیرند، برنامه نام دارند. در ابتدا نحوه ذخیره و اجرای کدها و برنامه‌ها را در پایتون بررسی می‌کنیم. همان‌طور که پیش‌تر گفته شد، برای نوشتن کدها به زبان پایتون، از محیط IDLE خود پایتون استفاده می‌کنیم و دستورات را در آن تایپ می‌کنیم (تصویر ۱).

برای اجرای این دستورات، ابتدا باید برنامه را در قالب یک فایل ذخیره کنیم و یک نام برای فایل در نظر بگیریم (تصویرهای ۲ و ۳). پسوند‌های .py، .pyw یا .pyi ذخیره می‌شوند. برای اجرا، از فهرست run گزینه run module را انتخاب یا کلید F۵ را فشار دهید (تصویر ۴).

انواع داده‌ها

برای نوشتن هر برنامه معمولاً با مجموعه‌ای از داده‌ها (مقدارهایی از اعداد یا متن‌ها) در

انواع متفاوت سر و کار دارید؛ مانند داده‌های متنی، عدد صحیح، عدد اعشاری، تصویر، صدا و داده منطقی. پرکاربردترین آن‌ها در پایتون عبارت‌اند از: داده رشته‌ای یا متنی؛ داده اعشاری (.)؛ داده صحیح؛ داده منطقی.

داده رشته‌ای یا متنی: بخشی از متن را توصیف می‌کند که به آن رشته نیز گفته می‌شود. مانند نام، نام خانوادگی، آدرس و... متنی یا رشته‌ای هستند.

داده اعشاری: عدد را به صورت اعشاری توصیف می‌کند: مانند ۳۲/۵، ۰۰/۲۵.

داده صحیح: عدد را به صورت مثبت یا منفی بیان می‌کند: مانند ۱۰-، ۱۰۵۰+، ۵۰۶-.

داده منطقی:

درست یا نادرست بودن مقداری را توصیف می‌کند. مثلاً ۱ < ۲ است، پس مقدار درست است. یا ۱۰ = ۱۵، پس مقدار غلط است.

تعریف و مقداردهی متغیرها

متغیر به مکانی از حافظه می‌گویند که امکان ذخیره و بازیابی داده‌ها را برای برنامه‌نویسان فراهم می‌کند. هر متغیر یک نام دارد و برای دسترسی به محتوای آن معمولاً از نام استفاده می‌شود. کارکرد متغیر مثل ظرف آشپزخانه است که هر کدام نامی دارند و غذایی را در خود نگه می‌دارند؛ مثل لیوان شیر، قابلمه غذا و سبزی. برای مقداردهی متغیرها، از عملگر «=» که دستور انتساب نام دارد، به این صورت استفاده می‌شود:

در اکثر زبان‌های برنامه‌نویسی قبل از استفاده از متغیر باید نوع داده مشخص شود، ولی در زبان پایتون بعد از مقداردهی داده، نوع آن به‌طور خودکار مشخص می‌شود و در حین برنامه قابل تغییر است.

زمانی که بخواهیم یک پروژه را به‌صورت واقعی بنویسیم و اجرا کنیم، با تعداد زیادی متغیر روبه‌رو هستیم که باید برای هر کدام اسمی مشخص کنیم تا در حین نوشتن برنامه بتوانیم به‌راحتی با آن‌ها کار کنیم و مغزمان متغیرها را در برنامه جابه‌جا استفاده نکند. برای نام‌گذاری هر متغیر باید از قوانین نام‌گذاری آن در زبان برنامه‌نویسی پایتون تبعیت کنیم. این قوانین چنین هستند:

- نام متغیر نمی‌تواند با عدد شروع شود؛ مانند: score۱
- نام متغیر می‌تواند با _ شروع شود. مانند: _FirstName

نام متغیرها به حروف بزرگ و کوچک حساس است. مانند این نام‌ها که با هم متفاوت هستند:

```
firstname      Firstname
firstName     FirstName
```

نام متغیرها تنها می‌تواند شامل نویسه‌های (کاراکترهای) حرفی، عددی و زیرخط (_) باشد.

برای نام‌گذاری متغیرها، زمانی که نام آن‌ها بیشتر از ۱ کلمه است، روش‌های گوناگونی وجود دارند:

کوهان شتری: به‌جز کلمه اول نام، بقیه کلمات با حرف بزرگ شروع می‌شوند؛ مانند: myFirstName

ماری: هر کلمه نام با یک _ از دیگری جدا می‌شود؛ مانند: my_first_name

پاسکالی: هر کلمه با یک حرف بزرگ شروع می‌شود؛ مانند: MyFirstName

چند مثال از تعریف و مقداردهی متغیرها

در تصویر ۵ متغیری به نام score۱ تعریف شده است. مقدار ۱۵ داخل آن ذخیره شده و نوع داده آن عدد صحیح تشخیص داده شده است.

اگر شما مانند تصویر ۶ مقدار ۱۵/۵ را به متغیر score۲ انتساب دهید، نوع داده عدد اعشاری است.

در تصویر ۷ متغیر name۱ تعریف شده و مقدار fatemeh را به آن انتساب داده‌ایم. برای مقداردهی به‌صورت متنی، باید مقدار متنی موردنظر خود را بین "" یا '' قرار دهیم؛ مانند تصویرهای ۷ و ۸.

دستور خروجی

همه برنامه‌ها مقدارهایی را به‌عنوان ورودی دریافت می‌کنند و بعد از انجام محاسبات و پردازش‌های موردنیاز، آن‌ها را در خروجی نمایش می‌دهند. برای مثال می‌خواهیم برنامه‌ای بنویسیم که معدل دانش‌آموزان را حساب کند.

در پایتون برای نمایش یا چاپ خروجی برنامه برای کاربر، از تابع print استفاده می‌شود.

در تصویر ۹ ابتدا سه متغیر تعریف و مقداردهی شده‌اند و سپس با استفاده از تابع پرینت مقادیر آن‌ها نمایش داده شده است:

حالا اگر بخواهیم دو متغیر نام و نام خانوادگی در کنار هم نمایش داده شوند، مانند تصویر ۱۰ باید بنویسیم:

اگر شما مقداری را به‌طور مستقیم داخل

دستور (Print) نیز قرار دهید، مقدار را برای شما چاپ می‌کند (تصویر ۱۱).

تمرین: برنامه‌ای بنویسید که در آن سه متغیر نام، نام خانوادگی و سن تعریف و مقداردهی شوند و در نهایت پیامی به این صورت نمایش داده شود: «سلام، خوش آمدی، سن شما ... سال است.»

تشخیص نوع داده‌ها

یادتان هست گفتیم متغیرها مثل ظرف غذا هستند؟ حالا فرض کنید مادر شما می‌گوید قابلمه خورش را بیاور. روی گاز سه عدد قابلمه وجود دارد. حال چطور قابلمه خورش را تشخیص می‌دهید؟

بعضی وقت‌ها در حین نوشتن برنامه با خطاهایی روبه‌رو می‌شویم که مثلاً می‌گوید این دستور برای این نوع داده درست نیست. ما نیاز داریم نوع داده را تشخیص دهیم. برای تشخیص نوع داده از دستور (Type) استفاده می‌کنیم.

برای مثال، می‌خواهیم نوع داده‌های متغیرهای خود را چاپ کنیم. به این برنامه دقت کنید:

دستور (Type) فقط نوع داده را مشخص می‌کند، اما برای اینکه نوع داده چاپ شود، از دستور (Print) کمک می‌گیریم. در برنامه اجرا شده، نوع داده Str را برگردانده که منظور داده رشته‌ای (String) است.

بسی نوشتن‌ها

1. Float
2. Integer
3. Boolean

```
5.
a = 15
print(a)
# Output: 15
```

```
6.
a = 15.5
print(a)
# Output: 15.5
```

```
7.
a = "Fateme"
print(a)
# Output: Fateme
```

```
8.
a = "Fateme"
print(a)
# Output: Fateme
```

```
9.
a = 15
b = 15.5
print(a, b)
# Output: 15 15.5
```

```
10.
a = "Fateme"
b = "Hani"
print(a, b)
# Output: Fateme Hani
```

```
11.
a = 15
b = 15.5
c = "Fateme"
print(a, b, c)
# Output: 15 15.5 Fateme
```

```
12.
a = 15
b = 15.5
c = "Fateme"
print(a, b, c)
# Output: 15 15.5 Fateme
```

```
13.
a = 15
b = 15.5
c = "Fateme"
print(a, b, c)
# Output: 15 15.5 Fateme
```

```
14.
a = 15
b = 15.5
c = "Fateme"
print(a, b, c)
# Output: 15 15.5 Fateme
```