

استفاده از متغیرها

بر خلاف زبان هایی مثل جاوا و سی شارپ، که هنگام تعریف متغیر باید نوع متغیر را هم مشخص می کردیم، در پایتون کافیست که فقط نام متغیر را نوشته

و به وسیله علامت مساوی یک مقدار به آن اختصاص دهیم :

```
variableName = Value
```

در مثال زیر نحوه تعریف و مقداردهی متغیرها نمایش داده شده است :

```
1 intVar      = 10
2 floatVar    = 12.5
3 boolVar     = True
4 StringVar   = "Hello World!"
5 listVar     = [1,5,8]
6 tupleVar    = ("Python","Programming","begginer")
7 dictionaryVar = {'Name': 'jack', 'family': 'Scalia', 'Age': 7}
8
9 print("intVar      = {0}" .format(intVar))
10 print("floatVar    = {0}" .format(floatVar))
11 print("boolVar     = {0}" .format(boolVar))
12 print("StringVar   = {0}" .format(StringVar))
13 print("listVar     = {0}" .format(listVar))
14 print("tupleVar    = {0}" .format(tupleVar))
15 print("dictionaryVar = {0}" .format(dictionaryVar))
```

```
intVar      = 10
floatVar    = 12.5
boolVar     = True
StringVar   = Hello World!
listVar     = [1, 5, 8]
tupleVar    = ('Python', 'Programming', 'begginer')
dictionaryVar = {'Name': 'jack', 'family': 'Scalia', 'Age': 7}
```

در خطوط 7-1، متغیر ها تعریف شده اند. اما نوع این متغیرها چیست؟ پایتون نوع متغیرها را بسته به مقداری که به آنها اختصاص داده می شود در نظر می گیرد. مثلا نوع متغیر StringVar در خط 4 از نوع رشته است، چون یک مقدار رشته ای به آن اختصاص داده شده است. به خطوط 5، 6 و 7 کد بالا توجه کنید. در خط 5 یک متغیر تعریف شده است و نوع داده ای که به آن اختصاص داده شده است از نوع list است. همانطور که در درس قبل اشاره شد،

برای تعریف list علامت [] به کار می رود و آیتم های داخل آن به وسیله کاما از هم جدا می شوند :

```
listVar = [1, 5, 8]
```

در خط 6 هم یک متغیر تعریف شده است و یک مقدار از نوع tuple به آن اختصاص داده شده است. در تعریف tuple به جای علامت [] از () استفاده می شود. تفاوت بین این دو را در درس های آینده بیشتر توضیح می دهیم. و اما در خط 7 یک نوع دیکشنری تعریف شده است. برای تعریف دیکشنری بین

کلید و مقدار علامت : و بین کلید/مقدارها هم علامت , قرار می گیرد :

```
dictionaryVar = {Key1:Value1, Key2:Value2, Key3:Value3}
```

مثلا در مثال بالا یک دیکشنری تعریف کرده ایم که سه آیتم یا کلید/مقدار دارد که بین آنها علامت کاما (,) قرار داده ایم. ولی بین یک کلید و مقدار مربوط به آن علامت : قرار گرفته است. برای اختصاص یک مقدار به چند متغیر می توان به صورت زیر عمل کرد :

```
identifier1 = identifier2 = ... indentifierN = Value
```

به مثال زیر توجه کنید :

```
num1 = num2 = num3 = num4 = num5 = 10
message1 = message2 = message3 = "Hello World!"

print(num1)
print(num4)
print(message1)
print(message3)
```

```
10
10
Hello World!
Hello World!
```

دقت کنید که برای متغیرهای تعریف شده در حالت بالا یک خانه حافظه تخصیص داده می شود، یعنی مقدار 10 در حافظه ذخیره شده و متغیرهای num1 و num2 و num3 و num4 و num5 به آن خانه از حافظه اشاره می کنند. همچنین می توان چند متغیر را تعریف کرد و برای هر یک از آن ها مقدار جداگانه ای مشخص نمود:

```
identifier1, identifier2, ... indentifierN = Value1, Value2, ... ValueN
```

به مثال زیر توجه کنید:

```
num1, num2, message1 = 10, 12.5, "Hello World!"

print(num1)
print(num2)
print(message1)
```

```
10
12.5
Hello World!
```

در کد بالا مقدار num1 برابر 10، num2 برابر 12.5 و message1 برابر Hello World! می باشد. در پایتون، متغیر ها هم باید تعریف و هم مقداردهی شوند. یعنی اگر متغیری را تعریف کرده و به آن مقداری را اختصاص ندهید و برنامه را اجرا کنید با خطا مواجه می شوید :

```
number

print(number)
```

```
Traceback (most recent call last):
  File "...", line 1, in
    number
NameError: name 'number' is not defined
```

همانطور که در درس قبل هم اشاره کردیم، یک رشته در اصل یک مجموعه از کاراکترهاست که در داخل علامت "" یا '' قرار دارند. هر کدام از این کاراکترها

دارای یک اندیس است که به وسیله آن اندیس قابل دسترسی هستند. اندیس کاراکترها در رشته از 0 شروع می شود. به رشته زیر توجه کنید :

```
message = "Hello World!"
```

در رشته بالا اندیس کاراکتر 0 برابر 4 است. برای درک بهتر به شکل زیر توجه کنید :

```
H e l l o   W o r l d !
0 1 2 3 4 5 6 7 8 9 10 11
```

حال برای چاپ یک کاراکتر (مثلا W) از این رشته کافیست که به صورت زیر عمل کنیم :

```
message = "Hello World! "

print(message[6])
```

```
W
```

همانطور که در کد بالا مشاهده می کنید کافیست که نام متغیر را نوشته، در جلوی آن یک جفت کروشه و در داخل کروشه ها اندیس آن کاراکتری را که می خواهیم چاپ شود را بنویسیم. چاپ مقدار با استفاده از اندیس در مورد List و Tuple هم صدق می کند :

```
listVar = [1, 5, 8]
tupleVar = ("Python", "Programming", "begginer")

print(listVar[2])
print(tupleVar[1])
```

```
8
Programming
```

و اما در مورد دیکشنری، شما باید نام کلید را بنویسید تا مقدار آن برای شما نمایش داده شود:

```
dictionaryVar = {'Name': 'jack', 'Family': 'Scalia', 'Age': 7}

print(dictionaryVar['Family'])
```

```
Scalia
```

نکته ای که بهتر است در همینجا به آن اشاره کنیم این است که کلید/مقدارها در دیکشنری می توانند از هر نوعی باشند و شما برای چاپ مقدار مربوط به یک کلید باید نام کلید را دقیق بنویسید. به مثال زیر توجه کنید :

```
dictionaryVar = {1:'Jack', '2':'Scalia', 3:7}

print(dictionaryVar['2'])
```

```
Scalia
```

در مثال بالا ما مقدار کلید '2' را چاپ کرده ایم. حال اگر به جای '2' عدد 2 را بنویسیم، یعنی علامت کوتیشن را نگذاریم با خطا مواجه می شویم :

```
dictionaryVar = {1:'Jack', '2':'Scalia', 3:7}

print(dictionaryVar[2])
```

```
Traceback (most recent call last):
  File "C:/MyFirstProgram.py", line 3, in
    print(dictionaryVar[2])
KeyError: 2
```

جانگهدار (Placeholders)

به تابع print() در خطوط (9-15) توجه کنید. این تابع به دو قسمت تقسیم شده است. قسمت اول یک رشته قالب بندی شده است و و قسمت دوم هم شامل تابعی به نام format() است که دارای مقدار یا مقادیری است که توسط رشته قالب بندی شده مورد استفاده قرار می گیرند. اگر به دقت نگاه کنید رشته قالب بندی شده دارای عدد صفری است که در داخل دو آکولاد محصور شده است. البته عدد داخل دو آکولاد می تواند از صفر تا n باشد. به این اعداد جانگهدار می گویند. این اعداد بوسیله مقدار یا مقادیری که در داخل تابع format() هستند جایگزین می شوند. به عنوان مثال جانگهدار {0} به این معناست

که اولین مقدار داخل تابع format() در آن قرار می گیرد. برای روشن شدن مطلب به شکل زیر توجه کنید :

```
print("The values are {0}, {1}, {2}, and {3}.".format(value1, value2, value3, value4))
```

```
print("The values are {0}, {1}, {2}, and {3}.".format(value1, value2, value3, value4));
```

جانگهدارها از صفر شروع می شوند. تعداد جانگهدارها باید با تعداد مقادیری که در داخل تابع format() آورده شده اند برابر باشد. برای مثال اگر شما چهار

جانگهدار مثل بالا داشته باشید باید چهار مقدار هم برای آنها بعد از رشته قالب بندی شده در نظر بگیرید. اولین جانگهدار با اولین مقدار و دومین جا نگهدار با

دومین مقدار جایگزین می شود. در ابتدا فهمیدن این مفهوم برای کسانی که تازه برنامه نویسی را شروع کرده اند سخت است اما در درسهای آینده مثالهای

یادی در این مورد مشاهده خواهید کرد.