داده ساختارها و عملگرهای اصلی

کامنت‌ها در پایتون

کامنت‌های تک خطی در پایتون با علامت `#` و کامنت‌های چند خطی با `"""` مشخص می‌شوند.

```python
# Single line comments start with a number symbol.

""" Multiline strings can be written
    using three "s, and are often used
    as documentation.
"""
```

- **نکته**: کامنت‌های چند خطی در پایتون در واقع رشته‌های چند خطی هستند که چون در اطراف آنها عملگری وجود ندارد استفاده نمی‌شوند.

در پایتون سه نوع عدد وجود دارد؛ صحیح `int`، اعشاری `float` و مختلط `complex` که اعداد اعشاری حداقل دارای یک رقم اعشار هستند.

```
>>> type(3)
<class 'int'>
>>> type(3.0)
<class 'float'>
>>> type(1 + 1j)
<class 'complex'>
```

اعمال پایه‌ی ریاضی مانند بقیه‌ی زبان‌های برنامه‌نویسی است

- **نکته**: حاصل تقسیم در پایتون ۳، همواره عددی اعشاری است.

```
>>> 1 + 1
2
>>> 8 - 1
7
>>> 10 * 2
20
>>> 35 / 5
7.0
>>> 1 + 2j
(1+2j)
>>> type(1 + 2j)
<class 'complex'>
```

تقسیم صحیح که با علامت `//` نشان داده می شود، همیشه حاصل تقسیم را به سمت پایین گرد می‌کند. البته برای اعداد صحیح خروجی صحیح و برای اعداد اعشاری خروجی اعشاری خواهد بود.

```
>>> 5 // 3
1
>>> -5 // 3
-2
>>> 5.0 // 3.0   # works on floats too
1.0
>>> -5.0 // 3.0
```

علامت `%` عملگر باقیمانده و `**` عملگر توان است. همانطور که انتظار می‌رود پرانتز اولویت بالاتری نسبت به بقیه‌ی عملگرها دارد.

$$x^{**}y = x^y$$

```
>>> 7 % 3
1
>>> 2 ** 3
8
>>> (1 + 3) * 2
8
```

مثالی از اولویت عملگرها:

```
>>> a = 20
>>> b = 5
>>> c = 10
>>> d = 1
>>> (a + b) * c / d
250.0
>>> ((a + b) * c) / d
250.0
>>> (a + b) * (c / d)
250.0
>>> a + (b * c) / d
70.0
```

مقادیر بولی با `True` و `False` نشان داده می‌شوند و با عملگر بولی `not` نقیض می‌شوند. (دقت کنید که املای مقادیر بولی باید دقیقا به همین شکل باشند و حرف اول باید بزرگ و بقیه باید کوچک نوشته شوند)

```
>>> not True
False
>>> not False
True
```

`True` و `False` در واقع اسامی دیگری برای مقادیر ۱ و ۰ هستند.

```
>>> True + True
2
```

```
>>> True * 8
8
>>> False - 5
-5
```

عملگرهای مقایسه هم مقادیر عددی آنها را مورد نظر قرار می‌دهند.

```
>>> 0 == False
True
>>> 1 == True
True
>>> 2 == True
False
>>> -5 != False
True
```

عملگرهای منطقی `and` و `or` بین دو مقدار بولی می‌آیند و در صورتی که مقادیر اطراف آنها بولی نباشند آنها را با تابع `bool()` به مقادیر بولی تبدیل می‌کنند و نباید با عملگرهای بیتی `&` و `|` اشتباه شوند. مقدار بولی صفر برابر `False` و مقدار بولی هر عدد دیگری برابر `True` است.

```
>>> 0 and 2
0
>>> -5 or 0
-5
>>> bool(0)
False
>>> bool(4)
True
>>> bool(-6)
True
```

عملگرهای مقایسه، مشابه زبان‌های دیگر هستند.

```
>>> 1 == 1
True
>>> 2 == 1
False
>>> 1 != 1
False
>>> 2 != 1
True
>>> 1 < 10
True
>>> 1 > 10
False
>>> 2 <= 2
True

```

```
>>> 2 >= 2
True
```

پایتون اجازه می‌دهد چند مقایسه به صورت خلاصه‌تر نوشته شود.(*Chaining*)

$$x < y < z \equiv x < y \; and \; y < z$$

```
>>> 1 < 2 and 2 < 3
True
>>> 1 < 2 < 3
True
>>> 2 < 3 and 3 < 2
False
>>> 2 < 3 < 2
False
```

عملگر `==` نشان می‌دهد که آیا دو طرف آن مقادیر یکسانی دارند یا نه ولی عملگر `is` نشان می‌دهد که آیا دو طرف آن به یک جا اشاره می‌کنند یا نه. در مثال پایین ابتدا لیستی ایجاد شده و متغیر `a` به آن اشاره میکند سپس مقدار اشاره‌گر `a` داخل `b` ریخته می‌شود و عملگر `is` نشان می‌دهد که هر دو به یک جای حافظه اشاره می‌کنند. سپس لیستی جدید(با مقداری مشابه قبلی) ایجاد شده و `b` به آن اشاره میکند پس عملگر `is` نشان می‌دهد که دو شی `a` و `b` به یک جای حافظه اشاره نمی‌کنند ولی عملگر `==` نشان می‌دهد که این دو مقادیر یکسانی دارند.

```
>>> a = [1, 2, 3, 4]  # Point a at a new list, [1, 2, 3, 4]
>>> b = a             # Point b at what a is pointing to
>>> b is a            # if a and b refer to the same object
True
>>> b = [1, 2, 3, 4]  # Point b at a new list, [1, 2, 3, 4]
>>> b is a            # if a and b refer to the same object
False
>>> b == a            #if a's and b's objects are equal
True
```

`None` کلمه‌ای رزرو شده در پایتون است که تقریبا معادل `null` در زبان‌های دیگر است. برای چک کردن این که متغیری `None` است باید از عملگر `is` استفاده شود.

```
>>> a = "etc"
>>> a is None
False
>>> a = None
>>> a is None
True
```

تکه کد پایین نشان می‌دهد در هر ساختار داده‌ای کدام مقدار معادل مقدار بولی `False` است. این ساختار داده‌ها در ادامه توضیح داده خواهند شد.

```
>>> bool(0) #numeric
False
>>> bool("") #string
False
>>> bool([]) #list
False
>>> bool({}) #set
False
>>> bool(()) #tuple
False
```

در پایتون اگر عملگری قبل از مساوی بیاید، آن عملگر روی متغیر و عدد بعد از مساوی اعمال می‌شود و نتیجه را دوباره در متغیر می‌ریزد.

```
>>> a = 10
>>> a
10
>>> a *= 10 #-> a = a * 10
>>> a
100
>>> a += 5 #-> a = a + 5
>>> a
105
>>> a /= 5 #-> a = a / 5
>>> a
21.0
>>> a **= 2 #-> a = a ** 2
>>> a
441.0
```

در واقع مثال بالا معادل کد زیر است:

```
a = 10
a = a * 10
a = a + 5
a = a / 5
a = a ** 2
```