

حلقه‌ها

حلقه‌ی while

تا وقتی که شرط جلوی while برقرار باشد، حلقه اجرا می‌شود.

```
1 x = 0
2 while x < 4:
3     print(x)
4     x += 1
```

به طور مثال خروجی تکه کد بالا به صورت زیر است:

```
1 0
2 1
3 2
4 3
```

حلقه‌ی for

حلقه‌های for یک شی iterable را می‌پیمایند. این شی می‌تواند یک لیست باشد یا دنباله ای از اعداد.

```
1 >>> for animal in ["dog", "cat", "mouse"]:
2     ...     print(animal)
3 dog
4 cat
5 mouse
```

در مثال بالا animal نماینده یکی از اعضای لیست است. در این مثال روی لیست پیمایش انجام شده است.

تابع range()

برای استفاده‌ی معمول از دستور for (مشابه استفاده این دستور در زبان های دیگر) می‌توان از تابع range() برای تولید دنباله‌ای از اعداد استفاده کرد.

خروجی تابع range() دنباله‌ای از اعداد است. این تابع به سه روش قابل استفاده است:

روش ۱: range(n) که اعداد بازه صحیح [0, n) را خروجی می‌دهد.

روش ۲: range(l, r) که اعداد بازه صحیح [l, r) را خروجی می‌دهد.

روش ۳: range(l, r, step) که خروجی آن به صورت زیر می‌باشد.

$$l, l + 1 \times step, l + 2 \times step, \dots, l + \lfloor \frac{r - l - 1}{step} \rfloor \times step$$

برای درک چگونگی کارکرد این تابع به کد زیر توجه کنید :

```
1 for i in range(5):
2     print(i, end=' ')
3
4 for i in range(5, 10):
5     print(i, end=' ')
6
7 for i in range(0, 10, 2):
8     print(i, end=' ')
```

خروجی مثال بالا :

```
1 0 1 2 3 4
2 5 6 7 8 9
3 0 2 4 6 8
```

به توجه به اینکه `for` در پایتون به طور پیش فرض اندیس اعضای لیست را برمی گرداند و خود اعضا را برمی گرداند؛ می توانید برای داشتن اندیس هر عضو همراه با آن از تابع `enumerate` استفاده کنید.

```
1 >>> l = ["dog", "cat", "mouse"]
2 >>> enumerate(l)
3 <enumerate object at 0x00000228FB906A68>
4 >>> list(enumerate(l))
5 [(0, 'dog'), (1, 'cat'), (2, 'mouse')]
```

کلمه‌ی break

کلمه‌ی `break` باعث می شود برنامه از حلقه خارج شود. استفاده‌ی آن در جایی که حلقه های تودرتو داریم، برنامه را از درونی ترین حلقه خارج می کند.

```
1 >>> for i in range(4,8):
2     ...     print(i)
3     ...     if i== 6 :
4     ...         break
5     ...
6 4
7 5
8 6
```

- در مثال بالا می بینید که با رسیدن به کلمه‌ی `break` برنامه از حلقه خارج شده و عدد ۷ چاپ نشده است.

```
1 >>> for i in range(3):
2     ...     for j in range(3):
3     ...         print(j)
4     ...         break
5     ...
6 0
7 0
8 0
```

- در این مثال مشاهده می‌کنید که تنها از حلقه درونی خارج شده و حلقه بیرونی همچنان اجرا می‌شود.