

حلقه‌های while

دستور while در پایتون یکی از راه‌های عمومی برای انجام تکرار است. یک دستور while به طور مکرر یک دستور واحد یا گروهی از دستورات را تا زمانی که شرط درست باشد اجرا می‌کند. دلیل اینکه آن را "حلقه" می‌نامند، این است که دستورات کد به صورت مکرر از طریق آن‌ها تکرار می‌شوند تا زمانی که شرط دیگر برآورده نشود.

قالب عمومی یک حلقه while به شرح زیر است:

```
while test:
    code statements
else:
    final code statements
```

بیایید به چند حلقه while ساده در عمل نگاه کنیم.

In [1]:

```
1 x = 0
2
3 while x < 10:
4     print(x)
5     x = x + 1 # x += 1
6
7 print('Outside of While loop')
```

```
0
1
2
3
4
5
6
7
8
9
Outside of While loop
```

توجه کنید که چند بار دستورات چاپ اتفاق افتاده و حلقه while ادامه داشت تا زمانی که شرط درست برآورده شد. مهم است که توجه داشته باشید که هنگامی که این اتفاق افتاد، کد متوقف شد. بیایید ببینیم چگونه می‌توانیم یک دستور else اضافه کنیم:

In [4]:

```
1 x = 10
2
3 while x < 10:
4     print(x)
5     x = x + 1 # x += 1
6 else:
7     print('All done as expected!')
8
9 print('Outside of While loop')
```

All done as expected!
Outside of While loop

break, continue, pass

ما می‌توانیم از دستورات `break`، `continue` و `pass` در حلقه‌های خود استفاده کنیم تا قابلیت‌های اضافی برای موارد مختلف اضافه کنیم. سه دستور تعریف شده توسط:

`break`: این دستور، اجرای نزدیکترین حلقه را متوقف می‌کند.

`continue`: این دستور اجرای دور کنونی حلقه را نیمه کاره رها کرده و به ابتدای دور بعدی خواهد رفت.

`pass`: هیچ کاری انجام نمی‌دهد.

درباره دستورات `break` و `continue`، قالب عمومی حلقه `while` به این شکل است:

```
while test:
    code statement
    if test:
        break
    if test2:
        continue
else:
```

دستورات `break` و `continue` می‌توانند در هر جایی از بدنه حلقه ظاهر شوند، اما معمولاً آن‌ها را در داخل یک دستور `if` قرار می‌دهیم تا بر اساس شرایط خاصی عملی را انجام دهیم.

بیا بیا پیش بروید و به چند مثال نگاه کنید!

In [12]:

```
1 x = 0
2
3 while x < 10:
4     if x % 2 == 0:
5         print('Continue')
6         x = x + 1
7         continue
8     print(x)
9     x = x + 1 # x += 1
10 else:
11     print('All done as expected!')
12
13 print('Outside of While loop')
```

Continue

1

Continue

3

Continue

5

Continue

7

Continue

9

All done as expected!

Outside of While loop

توجه کنید که چگونه ما یک دستور چاپی داریم وقتی $x==3$ ، و یک `continue` در حالی که از طریق حلقه `while` خارجی ادامه می‌دهیم، چاپ می‌شود. بیایید یک `break` را هنگامی که $x == 3$ قرار دهیم و ببینیم آیا نتیجه منطقی است:

In [10]:

```
1 x = 0
2
3 while x < 10:
4     print('x is currently: ',x)
5     print(' x is still less than 10, adding 1 to x')
6     x+=1
7     if x==3:
8         print('Breaking because x==3')
9         break
10    else:
11        print('continuing...')
12        continue
```

x is currently: 0

x is still less than 10, adding 1 to x

continuing...

x is currently: 1

x is still less than 10, adding 1 to x

continuing...

x is currently: 2

x is still less than 10, adding 1 to x

Breaking because x==3

نکته مهم درباره کاربرد همزمان دستورات break با وجود عبارت else برای حلقه

توجه داشته باشید که دستورات داخل else زمانی اجرا خواهند شد که حلقه بصورت طبیعی به اتمام برسد و در صورتیکه حلقه با دستور break متوقف شود، دیگر دستورات else اجرا نخواهند شد!

In [11]:

```
1 x = 0
2
3 while x < 10:
4     print('x is currently: ',x)
5     x+=1
6     if x==3:
7         print('Breaking because x==3')
8         break
9 else:
10    print('The while loop has been ended naturally')
11
12 print('outside loop')
```

```
x is currently: 0
x is currently: 1
x is currently: 2
Breaking because x==3
outside loop
```

توجه کنید که دستور else دیگری به آن نرسیده و continue هرگز چاپ نشده است!

پس از این مثال‌های ساده اما مختصر، شما باید در استفاده از دستورات while در کد خود راحت باشید.

اما یک کلمه احتیاط! امکان دارد که با دستورات while یک حلقه بی‌پایان در حال اجرا ایجاد کنید. به عنوان مثال:

In []:

```
1 # DO NOT RUN THIS CODE!!!!
2 while True:
3     print("I'm stuck in an infinite loop!")
```

یک یادداشت سریع: اگر شما سلول بالا را اجرا کردید، روی منوی Kernel بالا کلیک کنید تا kernel را مجدداً راه‌اندازی کنید!

یک مثال دیگر

در مثال‌هایی که تاکنون بیان کردیم، تعداد تکرارهای حلقه از پیش قابل تعیین بود، اما با حلقه while می‌توان حلقه‌هایی را نوشت که تعداد تکرارها از پیش مشخص نباشد.

برای مثال اگر منتظر ورودی خاصی از طرف کاربر باشیم و تا زمانی که کاربر ورودی مورد نظر را وارد نکرده است به او اجازه ندهیم به بخش دیگری مراجعه نماید.

In [16]:

```
1 x = 1
2
3 while x != '5':
4     x = input('Please enter number FIVE in the input:')
5 else:
6     print('you have entered number FIVE')
```

```
Please enter number FIVE in the input:0
Please enter number FIVE in the input:8
Please enter number FIVE in the input:4
Please enter number FIVE in the input:5
you have entered number FIVE
```