

دستور for در پایتون:

```
x=0
while x < 5:
    print(x)
    x=x+1
```

برای مثال اگر بخواهیم کد قبلی را با for بنویسیم، داریم:

```
for n in range(5):
    print(n)
```

• **range(5)** اعداد صحیح (از نوع int) از ۰ تا ۴ تولید می‌کند: ۰ و ۱ و ۲ و ۳ و ۴

ساختار حلقه‌های for به صورت زیر است:

```
for <متغیر> in range(<تعدادی_عدد>):
    <عبارت>
    <عبارت>
...

```

- در هر بار اجرای حلقه، <متغیر> یک مقدار می‌گیرد.
- اولین بار، <متغیر> کوچکترین مقدار ممکن را می‌گیرد.
- سری بعد، <متغیر> مقدار قبلی + ۱ را به خود می‌گیرد.
- ...

دستور range میتواند بیشتر از یک ورودی داشته باشد:

range (start, stop, step)

- مقادیر پیش فرض $start = 0$ و $step = 1$ که اختیاری (optional) هستند.
- تا زمانی که <متغیر> مقدار $stop-1$ به خود بگیرد حلقه اجرا می‌شود.

مثال با دو ورودی (شروع و پایان با مقدار پیش فرض $step=1$):

```
mysum= 0
for i in range(7, 10):
    mysum += i
print (mysum)
```

$$7+8+9=24$$

مثال با سه ورودی:

```
mysum = 0
for i in range(5, 11, 2):
    mysum += i
print(mysum)
```

$$5+7+9=21$$

مثال) اعداد زوج بزرگتر مساوی ۱۰ و کوچکتر از ۲۰ را چاپ کنید:
 ✓ در جدول زیر این مثال با ۴ روش مختلف حل شده است.

<pre>i=10 while i<20: if i % 2 == 0: print (i) i += 1</pre>	<pre>for i in range (10,20): if i % 2 ==0: print (i)</pre>
<pre>i=10 while i<20: print (i) i += 2</pre>	<pre>for i in range (10,20,2): print (i)</pre>

تمرین کلاسی (۱) با استفاده از حلقه‌ی for برنامه‌ای بنویسید که مقدار n را از ورودی بخواند، سپس مقدار n! را چاپ کند.
 میدانیم

$$n! = n \times (n - 1) \times (n - 2) \times \dots \times 2 \times 1$$

نوع داده ای رشته یا string در پایتون:

- این نوع داده ای می‌تواند شامل حروف، اعداد، و یا کاراکترهای ویژه مثل % * # @ یا ... باشد.
- مقدارهای رشته ای در بین دو quotation (کوتیشن ") یا دو single quotes (') نوشته می‌شوند:

```
hi = "hello"
salam = 'hello'
```

- رشته‌ها را میتوانیم با هم جمع کنیم، به این عمل به اصطلاح concatenate می‌گویند.

```
hi = "hello"
name = "Ali"
greet = hi + name
greeting = hi + " " + name
```

عملیات رشته ای:

۱. جمع دو رشته

```
'ab' + 'cd' #concatenation: 'abcd'
```

۲. n بار تکرار رشته

```
3 * 'SRTTU' #successive concatenation: 'SRTTUSRTTUSRTTU'
```

۳. طول رشته

```
len('SRTTU') #the length: 5
```

۴. کاراکتر موجود در n امین اندیس:

```
'SRTTU' [1] #indexing: 'R'
```

- اندیس از ۰ شروع می‌شود
- اندیس دهی بیشتر از ۱- طول رشته با ارور مواجه میشود. (حداکثر عدد طول رشته منهای یک)

S	R	T	T	U
0	1	2	3	4

'SRTTU' [0] ----- 'S'

'SRTTU' [1] ----- 'R'

...

'SRTTU' [4] ----- 'U'

'SRTTU' [5] ----- ارور

۵. برش یک زیر رشته:

'SRTTU' [1:3] #slicing: 'RT'

- زیر رشته ی از اندیس اول (قبل دو نقطه) تا یکی قبل از اندیس دوم (بعد از دو نقطه) را بر می‌گرداند.
- اگر قبل از : مقداری نبود، از ابتدای رشته (اندیس ۰) حساب می‌کند.
- اگر بعد از : مقداری نبود، تا انتهای جمله را حساب می‌کند.
- اگر هیچ کدام از این دو نبود؛ یک کپی ایجاد می‌کند.

چاپ (print) هم زمان رشته ها و اعداد:

```
x = 1
print(x)
x_str= str(x)
print("my fav num is", x, ".", "x =", x)
print("my fav num is "+ x_str+ ". "+ "x = "+ x_str)
```

مثال) خروجی کدهای زیر چیست؟

```
text = input("Type anything... ")
print(5*text)
```

?

```
num= int(input("Type a number... "))
print(5*num)
```

?

حلقه‌ی for بر روی رشته‌ها:

```
for <رشته> in <متغیر>:
    <عبارت>
    <عبارت>
...

```

- در هر بار اجرای حلقه، <متغیر> یک کاراکتر از رشته را به خود می‌گیرد.
- اولین بار، <متغیر> اولین کاراکتر موجود در رشته است.
- سری بعد، <متغیر> کاراکتر موجود در اندیس قبلی + ۱ است.
- ...

مثال) هر کدام از حروف موجود در رشته ی "salam" را جدا در یک خط مجزا چاپ کنید:

```
for letter in 'salam':
    print(letter)

```

مثال) خروجی برنامه‌ی زیر چیست؟

```
count = 0
for letter in 'Snow!':
    print('Letter # ' + str(count) + ' is ' + str(letter))
    count += 1
print(count)

```

مثال) خروجی برنامه‌ی زیر چیست؟

```
greeting = 'Hello!'
count = 0

for letter in greeting:
    count += 1
    if count % 2 == 0:
        print(letter)
    print(letter)

print('done')

```

اتمام حلقه با `break`:

- بلافاصله پس از اجرای دستور `break` از حلقه خارج می‌شود.
- کدهای بعد از `break` نادیده گرفته شده و اجرا نمی‌شود.
- اگر حلقه‌های تو در تو داشته باشیم، تنها از داخلی‌ترین حلقه خارج می‌شود.

```
while <condition_1>:  
    while <condition_2>:  
        <expression_a>  
        break  
        <expression_b>  
    <expression_c>
```

تمرین کلاسی ۲ خروجی برنامه‌ی زیر چیست؟

```
mysum = 0  
for i in range(5, 11, 2):  
    mysum += i  
    if mysum == 5:  
        break  
print(mysum)
```