



مبانی برنامه‌سازی با زبان Python

مدرس: عیسی مرادنژاد

تنظیم: پردیس ساداتیان مقدم

سرفصل

- تکرار (Iteration)
- حلقه ی تکرار For در پایتون (دستور For)
- حلقه ی تکرار While در پایتون (دستور While)
- Pass Statement
- Continue Statement
- Break Statement



تکرار (Iteration)

- اجرای مکرر تعدادی دستور
- حلقه یا loop
- حلقه‌های تکرار پایتون (For, While)



دستور For

- یکی از پرکاربردترین حلقه‌ها در زبان پایتون
 - داخل حلقه ورودی:
۱. Counter (متغیر شمارنده)
 ۲. کلید واژه **in**
 ۳. Range – بازه ی تکرار

تابع Range

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

```
0  
1  
2  
3  
4
```

```
for i in range (6,10):  
    print (i)
```

```
6  
7  
8  
9
```

- یک مثال ساده


```
# Measure some strings:
```

```
words = ['cat', 'window', 'defenestrate']
```

```
for w in words:
```

```
    print(w, len(w))
```

```
cat 3
```

```
window 6
```

```
defenestrate 12
```

```
friends = ['hasan', 'sara', 'ali', 'pardis', 'negar']
```

```
count = 0
```

```
for name in friends:
```

```
    print ('salam', name)
```

```
    count = count + 1
```

```
print ('I said', count, 'hello' )
```

```
salam hasan
```


```
salam sara
```

```
salam ali
```

```
salam pardis
```

```
salam negar
```

```
I said 5 hello
```



```
a = ['Mary', 'had', 'a', 'little', 'lamb']
for i in range(len(a)):
    print(i, a[i])
```

```
0 Mary
1 had
2 a
3 little
4 lamb
```

```
for f in ['Sara', 'Mahsa', 'Ali', 'Mohammad']:
    print (f)
```

```
Sara
Mahsa
Ali
Mohammad
```

چند مفهوم

Score = 0

Score += 1

Score -= 1

```
for f in ['Sara', 'Mahsa', 'Ali', 'Mohammad']:  
    print (f)
```

Initialization •

مقدار دهی اولیه

Increment •

افزایش به اندازه ی یک واحد

Decrement •

کاهش به اندازه ی یک واحد

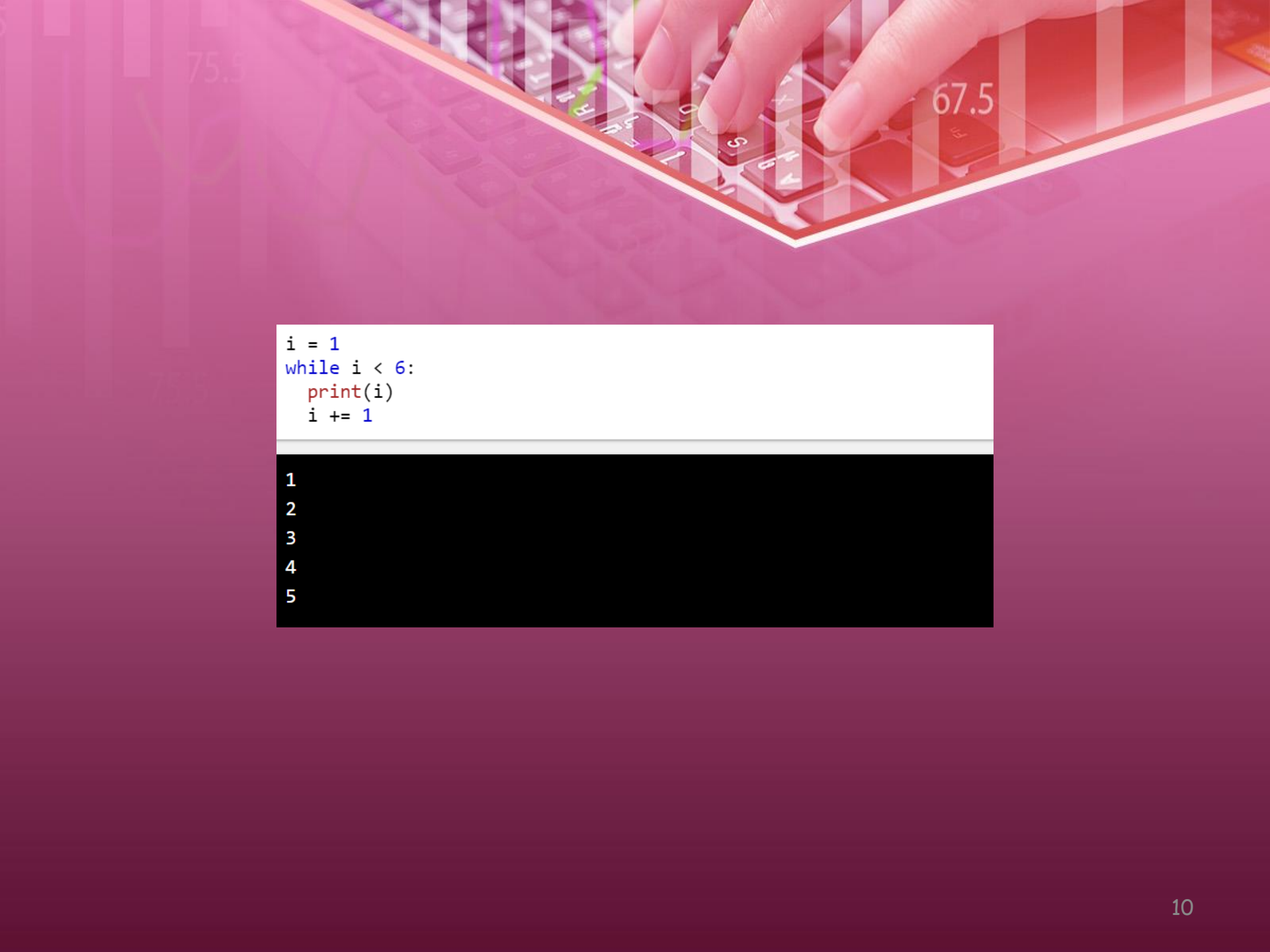
Traversal •

نوردیدن تمام عناصر یک لیست



دستور While

- این امکان را به برنامه می دهد که یک عمل را تا زمانی که شرطی برقرار باشد ادامه دهد.
- این شرط به عنوان ورودی به **While** داده می شود.
- تا زمانی که این شرط **True** باشد دستورات داخل حلقه تکرار می شود. هر زمان که شرط مورد نظر برقرار نباشد و مقدار **False** را برگرداند، برنامه از حلقه خارج شده و ادامه ی دستورات را اجرا می کند.



```
i = 1
while i < 6:
    print(i)
    i += 1
```

```
1
2
3
4
5
```

Pass Statement

```
for letter in 'Python':  
    if letter == 'h':  
        pass  
        print ('This is pass block')  
    print ('Current Letter :', letter)  
  
print ("Good bye!")
```

```
Current Letter : P  
Current Letter : y  
Current Letter : t  
This is pass block  
Current Letter : h  
Current Letter : o  
Current Letter : n  
Good bye!
```

Continue Statement

```
number = 0

for number in range(10):
    if number == 5:
        continue    # continue here

    print('Number is ' + str(number))

print('Out of loop')
```

```
Number is 0
Number is 1
Number is 2
Number is 3
Number is 4
Number is 6
Number is 7
Number is 8
Number is 9
Out of loop
```

Break Statement

```
number = 0

for number in range(10):
    if number == 5:
        break    # break here

    print('Number is ' + str(number))

print('Out of loop')
```

```
Number is 0
Number is 1
Number is 2
Number is 3
Number is 4
Out of loop
```


Nested For

```
list_of_lists = [['a', 'b', 'c'],[0, 1, 2],[9.9, 8.8, 7.7]]  
  
for list in list_of_lists:  
    for item in list:  
        print(item)
```

```
a  
b  
c  
0  
1  
2  
9.9  
8.8  
7.7
```

Nested While

```
i = 1
j = 5
while i < 4:
    while j < 8:
        print(i, ",", j)
        j = j + 1
    i = i + 1
```

```
1 , 5
2 , 6
3 , 7
```

For or While?

For

هر گاه حداکثر تعداد دفعات تکرار بدنه ی حلقه را بدانیم.
(definite iteration) در این حالت تکرار معلوم است.

While

هر گاه نیاز به تکرار بدنه حلقه تا زمان برقراری یک شرط وجود دارد و زمان برقراری شرط از پیش قابل پیشبینی نیست. (indefinite iteration)