In [ ]:

```
# Lambda
```

In [ ]:

```
def f(n):
    return n*2
print(f(3))

g = lambda n : n * 2
print(g(3))
```

In [ ]:

```
add = lambda x, y :x + y
print(add(2, 3))   # 5
```

In [ ]:

```
add = lambda x, y : (x + y , x - y)
print(add(2, 3))   # (5 , -1)
```

In [ ]:

```
m = lambda : print('hello')
m()   # hello
```

In [ ]:

```
d = {'a':3, 'b':7, 'c':5}
print(d[max(d.keys() , key = (lambda k: d[k]))])  # 7
```

In [ ]:

```
# map

lst = ['ali', 'reza']
u = []
for i in lst:
    x = i.upper()
    u.append(x)
print(u)            # ['ALI', 'REZA']

print(list(map(str.upper, lst)))     # ['ALI', 'REZA']
```

In [ ]:

```
name  = ['ali', 'reza']
score = [13,18]

print(list(zip(name,score)))   # [('ali', 13), ('reza', 18)]

print(list(map(lambda x, y : (x,y) , name , score))) # [('ali', 13), ('reza',
```

In [ ]:

```
lst = ['a' , 'A']

x = []
for i in lst:
    x.append(ord(i))
print(x)                      # [97, 65]

print(list(map(ord, lst)))   # [97, 65]
```

In [ ]:

```
scores = [12, 8 , 19, 15 , 7]
print(list(map(lambda x : x>9 , scores)))
# [True, False, True, True, False]
```

In [ ]:

```
a = [3, 2, 1]
b = [6, 4, 7]

print(list(map(lambda x, y :x+y, a, b)))   # [9, 6, 8]
```

In [ ]:

```
def f(x):
    return x + 5

def g(y):
    return y * 2

funcs = [f, g]

lst = [1, 2, 3]

for i in lst:
    print(list(map(lambda a : a(i) , funcs)))

'''
[6, 2]
[7, 4]
[8, 6]
'''
```

In [ ]:

```
# filter

scores = [12, 8 , 19, 15 , 7]
print(list(filter(lambda x : x>9 , scores)))   # [12, 19, 15]

lst = ['radar', 'ali', 'madam' , 'php']
palindrome = lambda x : (x == ''.join(reversed(x)))
print(list(filter(palindrome , lst)))  # ['radar', 'madam', 'php']
```

In [ ]:

```
lis = [2, 7, '', 9, {}, 8, [], 12]
print(list(filter(None,lis)))       # [2, 7, 9, 8, 12]
```

In [ ]:

```
# reduce
from functools import reduce

lis = [4, 8, 3, 5]

# add = lambda a,b : a+b

def add(x, y):
    return x + y

print(reduce(add, lis))  # 20   : (((4+8)+3)+5)

'''
def my_reduce(func, seq):
    r = seq[0]
    for i in seq[1:]:
        r = func(r,i)
    return r
print(my_reduce(add, lis))   # 20
'''
```

In [ ]:

```
print('----- sorted --------')

lst = [5, 2, 3, 1, 4]
print(sorted(lst))          # [1, 2, 3, 4, 5]
```

In [ ]:

```
 students = [
            {'id' : 1 ,'name' : 'taha' , 'score': 19},
            {'id' : 6 ,'name' : 'sara' , 'score': 8},
            {'id' : 4 ,'name' : 'omid' , 'score': 15},
            {'id' : 3 ,'name' : 'mahsa', 'score': 19}
          ]

print(sorted(students , key = lambda x : x['score']))
'''
[{'id': 6, 'name': 'sara', 'score': 8},
{'id': 4, 'name': 'omid' , 'score': 15},
{'id': 1, 'name': 'taha' , 'score': 19},
{'id': 3, 'name': 'mahsa', 'score': 19}]
'''
```

In [ ]:

```
student = [
            ( 1 , 'taha' , 19),
            ( 6 , 'sara' , 8),
            ( 4 , 'omid' , 15),
            ( 3 , 'mahsa', 19)
          ]

from operator import itemgetter

print(sorted(student , key = itemgetter(2)))
'''
[(6, 'sara', 8),
(4, 'omid', 15),
(1, 'taha', 19),
(3, 'mahsa', 19)]
'''
print(sorted(student , key = itemgetter(2) , reverse = True))
'''
[(1, 'taha', 19),
(3, 'mahsa', 19),
(4, 'omid', 15),
(6, 'sara', 8)]
'''
print(sorted(student , key = itemgetter(2, 1) ))
'''
[(6, 'sara', 8),
(4, 'omid', 15),
(3, 'mahsa', 19),
(1, 'taha', 19)]
'''
```

In [ ]:

```
d = {'ali':13, 'sara':17, 'taha':15}
print(sorted(d.items() , key = lambda x : x[1]))
# [('ali', 13), ('taha', 15), ('sara', 17)]
```

دانشگاه شهید مدنی آذربایجان
برنامه نویسی مقدماتی با پایتون
امین گلزاری اسکوئی
١٤٠٠-١٤٠١

Codes and Projects (click here) (https://github.com/Amin-Golzari-Oskouei/Python-Programming-Course-Basic-2021) slides and videos (click here) (https://drive.google.com/drive/folders/1ZsQjBJJ4UAAp9zrGxm3c4qrhnvGBUYHw)