Maktab Sharif

Python | Main course

Session 4

Review
Built-in functions
(Python Short-Hands)
Exercise

by Mohammad Amin H.B. Tehrani

www.maktabsharif.ir

Review

Local and Global variables

Maktab Sharif

Using keyword parameters is an alternative way to make function calls. The definition of the function doesn't change.
An example:

```
def square1(x):
    y = x * x
    return y

z = square1(10)
print(y)
```

```
def square2(x):
    y = x ** power
    return y

power = 2
result = square2(10)
print(result)
```

Local and Global variables

Maktab Sharif

Using keyword parameters is an alternative way to make function calls. The definition of the function doesn't change.
An example:

```
def func2():
    x *= 10

x = 10
func2()
```

```
def func1():
    print(x)

x = 10
func1()
```

What is your conclusion?

Example: Swap function

Maktab Sharif

Swap function

Write a function that, gets 2 values
Then swaps them.

Inputs:

```
def swap(...):
    ...

a = 10
b = 'Akbar'
swap()
print(a, b)
```

result:

```
Akbar 10
```

Example: Swap function

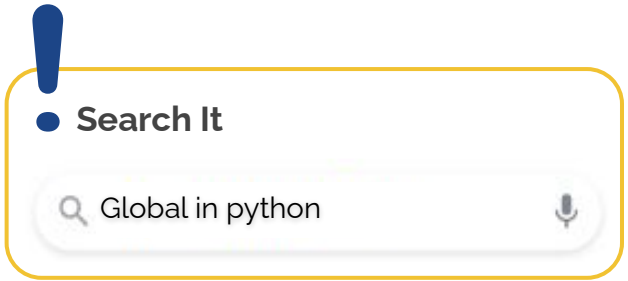
Maktab Sharif

Swap function

Write a function that, gets 2 values
Then swaps them.

Using **global**:

```
def swap():
    global a
    global b
    a, b = b, a
```



Example

Maktab Sharif

What's result of code below

```
def hi_all(students_list, teacher_list):
    teacher_list.extend(students_list)
    for x in teacher_list:
        print('> Hello', x, '!')


teachers = ['Shahin', 'Amirhossein', 'MohammadAmin']
students = ['Ali', 'Mohammad', 'Salar', 'Akbar', 'Nader']
print('Before function:', students, teachers, '', sep='\n\t')

hi_all(students, teachers)
print('\nAfter function:', students, teachers, sep='\n\t')
```

Example

Maktab Sharif

Why??

```
Before function:
     ['Ali', 'Mohammad', 'Salar', 'Akbar', 'Nader']
     ['Shahin', 'Amirhossein', 'MohammadAmin']

> Hello Shahin !
> Hello Amirhossein !
> Hello MohammadAmin !
> Hello Ali !
> Hello Mohammad !
> Hello Salar !
> Hello Akbar !
> Hello Nader !

After function:
     ['Ali', 'Mohammad', 'Salar', 'Akbar', 'Nader']
     ['Shahin', 'Amirhossein', 'MohammadAmin', 'Ali', 'Mohammad', 'Salar', 'Akbar', 'Nader']
```

Chapter 10

Some built-in function

Maktab Sharif

type(x)

Maktab Sharif

The **type()** function returns the type of the specified object

```
print(type(""))
print(type(124))
print(type(124.5))
print(type('124'))
print(type([]))
print(type(True))
print(type(range(5)))
print(type(str('123')))
```

len(...)

Maktab Sharif

The **len()** function returns the number of items in an object.

```
int_list = [1, 2, 3, 4, 5]
print(len(int_list))

float_tuple = (12.5, -2, 1.25, 0.5)
print(len(float_tuple))

print(len('Hello world!'))
string = "d e g a b l c f i h k j n"
print(len(string))

str_set = set('abcd')
print(str_set, len(str_set))
```

abs(num)

Maktab Sharif

The **abs()** function returns the absolute value of the specified number.

```
print(abs(-12))
print(abs(12))
print(abs(-0.5))
print(abs(0.5123))
print(abs(False))
print(abs('-12'))
```

round(float_num)

Maktab Sharif

The **round()** function returns a floating point number that is a rounded version of the specified number, with the specified number of decimals.

```
print(round(-12.8765, 2))
print(round(12.1245, 5))
print(round(-0.5342, 2))
print(round(0.5123, 2 ))
print(round(False, 1))
print(round('-12', 5))
```

sum(num_list)

Maktab Sharif

The **sum()** function returns a number, the sum of all items in an **list**.

```
int_list = [1,2,3,4,5]
print(sum(int_list))

float_list = [12.5, -2, 1.25, 0.5]
print(sum(float_list))

num_string = "12 5.5 -2.5 11"
num_str_list = num_string.split()
num_float_list = list(map(float, num_str_list))
print(sum(num_float_list))
```

max(num_list)

Maktab Sharif

The **max()** function returns the item with the highest value, or the item with the highest value in an iterable.

```
int_list = [1,2,3,4,5]
print(max(int_list))

float_list = [12.5, -2, 1.25, 0.5]
print(max(float_list))

num_string = "12 5.5 -2.5 11"
num_str_list = num_string.split()
num_float_list = list(map(float, num_str_list))
print(max(num_float_list))
```

min(num_list)

Maktab Sharif

The **min()** function returns the item with the lowest value, or the item with the lowest value in an iterable.

```
int_list = [1,2,3,4,5]
print(min(int_list))

float_list = [12.5, -2, 1.25, 0.5]
print(min(float_list))

num_string = "12 5.5 -2.5 11"
num_str_list = num_string.split()
num_float_list = list(map(float, num_str_list))
print(min(num_float_list))
```

sorted(list)

Maktab Sharif

The **sorted()** function returns a sorted list of the specified iterable object.

```
int_list = [1,2,3,4,5]
sorted_i_list = sorted(int_list)
print(sorted_i_list)

float_list = [12.5, -2, 1.25, 0.5]
sorted_f_list = sorted(float_list)
print(sorted_f_list)

string = "d e g a b l c f i h k j n"
str_list = string.split()
sorted_s_list = sorted(str_list)
print(sorted_s_list)
```

map(x, list)

Maktab Sharif

The **map()** function executes a specified function for each item in an iterable. The item is sent to the function as a parameter.
We can use it to convert all of list items to specific type.

```
str_list = ['-1','13.5', '-2.1', '11']
print(str_list)

float_list = list(map(float, str_list))
print(float_list)

final_list = list(map(lambda x: int(x)**2, float_list))
print(final_list)
```

More built-in functions

Maktab Sharif

`filer()`	`enumerate()`	`round()`	`bin()`	`sorted()`	`zip()`
`eval()`	`chr()`	`any()`	`hex()`	`reversed()`	`divmod()`
`exec()`	`ord()`	`all()`	`oct()`	`format()`	

Chapter 11

(Short-Hands)

Maktab Sharif

Conditional expression (ternary operator)

Maktab Sharif

Ternary operators also known as conditional expressions are operators that evaluate something based on a condition being true or false.

```
variable = first_value if condition else second_value
```

```
a, b = 10, 20

minimum = a if a < b else b

print(minimum)
```

```
a, b = 10, 20

if a < b:
    minimum = a
else:
    minimum = b

print(minimum)
```

List Comprehension (inline for)

Maktab Sharif

List comprehension offers a shorter syntax when you want to create a new list based on the values of an existing list.

```
newlist = [expression for item in iterable]
newlist = [expression for item in iterable if condition == True]
```

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

newlist = [x for x in fruits if "a" in x]

print(newlist)
```

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]
newlist = []

for x in fruits:
  if "a" in x:
    newlist.append(x)

print(newlist)
```

List Comprehension Examples

Maktab Sharif

```
l = [x**2 for x in range(1, 21)]
print(l)
```

```
s = 'Akbar neshan'
l = [x.upper() for x in s if x.lower() in 'aoieu']
print(l)
```

```
n = 10

for i in range(1, n+1):
    print(*[i*j for j in range(1, n+1)], sep='\t')
```

Lambda (inline function)

Maktab Sharif

A lambda function is a small anonymous function.
A lambda function can take any number of arguments, but can only have one expression.

`lambda` *`arguments`* : *`expression`*

```
f = lambda x: 2*x + 10
print(f(2))
```

```
def f(x):
    return 2*x + 10

print(f(2))
```

Lambda Examples

Maktab Sharif

```
f = lambda a, b, c : a + b + c
print(f(5, 6, 2))
```

```
l = list(map(lambda x: x ** 2, range(1, 21)))
print(l)
```

```
n = int(input('Enter a number: '))

x = lambda n: not [i for i in range(2, n) if not (n % i)]
y = lambda N: [i for i in range(2, N+1) if x(i)]

print(y(n))
```

Exercises

Exercise: Single digit

Maktab Sharif

Use functions!

تک‌رقمی

مهدی که از کدزدن خسته شده‌است، دیگر حوصله اعدادی که بیشتر از یک رقم دارند را ندارد. به همین خاطر به هر عدد چند رقمی که بر بخورد آن را به شیوه خاص خودش تبدیل به یک عدد تک رقمی می‌کند. به این شکل که عدد مورد نظر را با عدد حاصل از مجموع ارقام آن جایگزین می‌کند و به یک عدد جدید می‌رسد. سپس همین کار را با عدد جدید انجام می‌دهد و تا جایی که به یک عدد تک‌رقمی برسد به این کار ادامه می‌دهد. بعد از مدتی مهدی متوجه شد که با این کار نه تنها راحت تر نشده است، بلکه بیشتر درگیر اعداد شده است. در نتیجه از شما خواسته است در یک رقمی کردن عددها به او کمک کنید.

ورودی نمونه ۱

14

خروجی نمونه ۱

5

ورودی نمونه ۲

123456

خروجی نمونه ۲

3

Exercise: Fibonacci

Maktab Sharif

> Fibonacci
> Write a program that asks the user how many Fibonacci numbers to generate and then generates them into a list. (using functions)

input:

```
>> 10
```

output:

```
>> 1, 1, 2, 3, 5, 8, 13, 21, 34, 55
```

Exercise: GCD and LCM

Maktab Sharif

> GCD and LCM
> Write two functions, GCD (BMM) and LCM (KMM).

input:

```
>> 24 36
```

output:

```
>> GCD: 12
>> LCM: 72
```

Pre-reading

Search about:

1. * List copy in python
2. * Tuple in python (Tuple vs. list)
3. * Set in python (Set vs. list)
4. * Dictionary in python
5. args in python (*args)
6. kwargs in python (**kwargs)

Maktab Sharif