

05.Python.Functions

3ikakke

Outline

- Objectives
- What are functions?
- In built functions
- User defined functions
- Arguments in functions
- Q&A
- Gist of the day

Objectives

- Understand what functions do
- Understand the anatomy of functions
- Recognize functions vs methods
- Know how to define your own functions

What are functions?

- Functions are mini programs
- They make carrying out some tasks easy
- They are reusable code
- There are in built function some of which we have seen already
 - `print()`
 - `type()`
 - `len()`
 - `enumerate()`
 - `range()`

- You can define your own functions
- Anatomy of a function

```
function(arg1, arg2, ... argx)
```

In built functions

- Type casting
 - int()
 - list()
 - dict()
 - set()
 - str()
- Mathematical
 - min()
 - max()
 - sum()
 - round()
- String
 - min()
 - max()

In built functions (contd)

- useful functions
 - dir()
 - help()
- Some functions we will consider in the future
 - reversed()
 - sorted()
 - filter()
 - map()

User-defined functions

- You can write your own functions

- All you have to do is understand the anatomy of functions
- Next step is literally to define a function

```
def my_function():
    'do something'
```

- Finally call the function

```
my_function()
```

Outputting explicitly or implicitly (returning)

```
def my_function():
    print("Hello world")

def another_function():
    return "hello world"

a = my_function()

b = another_function()

print(a)
print(b)
```

Arguments in functions

- You can have functions with argument(s)
- You can set defaults for the argument(s)
- Arguments can be of any type or types
- Arguments can be collections
- Arguments that are collections offer some level of flexibility
- Argument types can be combined along with collections

Function arguments

```
#No arguments
def first_function():
    return "Hello world"

#one argument
def second_function(arg):
    return arg

#two arguments
def third_function(arg1, arg2):
    return arg1 + arg2
```

```
#list or tuple as argument
def forth_function(*args):
    print(args)
    return len(args)

#dictionary as argument
def fifth_function(**args):
    print(args)
    return len(args)
```

Gist for the day

- Get the pdf version
- Get the gist
- The Jupyter Notebook will be shared

Q&A

Thanks for your contributions!