11.Functions

January 20, 2019

1 Functions

1.1 Why functions?

- Semantic enrichment of code
- Reusability

Without function

```
In [1]: t0 = 12
        t1 = -35
        t2 = 144

        f0 = t0 * 9 / 5 + 32
        f1 = t1 * 9 / 5 + 32
        f2 = t2 * 9 / 5 + 32
```

With function

```
In [2]: def celsium_to_fahrenheit(celsium_temperature):
            print(celsium_temperature * 9 / 5 + 32)


        t0 = 12
        t1 = -35
        t2 = 144

        f0 = celsium_to_fahrenheit(t0)
        f1 = celsium_to_fahrenheit(t1)
        f2 = celsium_to_fahrenheit(t2)
```

```
53.6
-31.0
291.2
```

1.2 Function morphology

Definition

```
def function_name(parameter1, parameter2):
    function_body
```

Usage

```
a = 3
function_name(a, 'Qua-qua')
```

For example

```
print("Hi, guys")
```

print - function name
"Hi, guys" - argument

```
In [1]: # Definition of function with 0 arguments
        def function_example():
            print('!')

In [4]: # Usage
        function_example()

!


In [5]: # Simple function
        def greet():
            print('Hi there')

In [6]: greet()

Hi there


In [7]: # Function with 1 argument called name
        def greet(name):
            print('Hi there', name)

In [8]: # Usage
        greet('Vasya')

Hi there Vasya
```

In the moment of line execution with that function usage passed argument binds with argument in function

In the previous example this takes place
name = 'Vasya'

After that function body operates with the passed value as a variable called name