Comprehensive Python Programming

From Fundamentals to Advanced

(21 Oct 2024 - 13 Nov 2024)

IHUB DivyaSampark IIT Roorkee

Ritvij Bharat Pvt. Ltd.

Functions

Led by : Shreyas Shukla

Functions allow us to create blocks of code that can be easily executed many times, without having to constantly rewrite the entire block of code.

Functions will be a huge leap forward in your capabilities as a Python programmer.

Problems you are able to solve can also be a lot harder!

Combine everything you've learned so far (control flow, loops, etc.) with functions to become an effective programmer.

- Discouraged or Frustrated?
  Do not worry, this is completely normal and very common!
- Be patient with yourself.

Let’s see how to create functions with Python!

def Keyword

Creating a function with **def** keyword, correct indentation, and proper structure.

```
def function_name():
```

Keyword telling Python this is a function.

```
def function_name():
```

You decide on the function name. Notice “snake casing”

```
def function_name():
```

Snake casing is all lowercase with underscores between words

```
def function_name():
```

Parenthesis at the end. Later on we can pass in arguments/parameters into the function.

```
def function_name():
```

A colon indicates an upcoming indented block. Everything indented is then “inside” the function

```
def function_name():
    """
    Docstring explaining function.
    """
```

Optional: Multi-line string to describe function.

```
def function_name():
    """
    Docstring explaining function.
    """
    print("Hello")
```

Code then goes inside the function.

```
def function_name():
    ”””
    Docstring explaining function.
    ”””
    print(“Hello”)
```

Function can then be executed/called to see the result.

```
>> function_name():
>> Hello
```

```
def function_name():
    ”””
    Docstring explaining function.
    ”””
    print(“Hello”)
```

```
>> function_name():
>> Hello
```

Resulting Output

```
def function_name():
    ”””
    Docstring explaining function.
    ”””
    print(“Hello ”+name)
```

```
>> function_name():
>> Hello
```

Functions can accept arguments to be passed by the user.

```
def function_name(name):
    ”””
    Docstring explaining function.
    ”””
    print(“Hello ”+name)
```

```
>> function_name(Shreyas):
>> Hello Shreyas
```

Functions can accept arguments to be passed by the user.

Typically we use the **return** keyword to send back the result of the function, instead of just printing it out.

return allows us to assign the output of the function to a new variable.

We will have a deeper discussion of the **return** keyword later on in the notebook.

```
def add_function(num1,num2):
    return num1+num2
```

```
>> result = add_function(1,2)
>>
>> print(result)
>> 3
```

Return allows to save the result to a variable.

```
def add_function(num1,num2)::
    return num1+num2
```

```
>> result = add_function(1,2)
>>
>> print(result)
>> 3
```

Most functions will use return. Rarely will a function only print()

Let’s start creating functions with Python.