

Python Functions

A function is a block of code which only runs when it is called.

You can pass data, known as parameters, into a function.

A function can return data as a result.

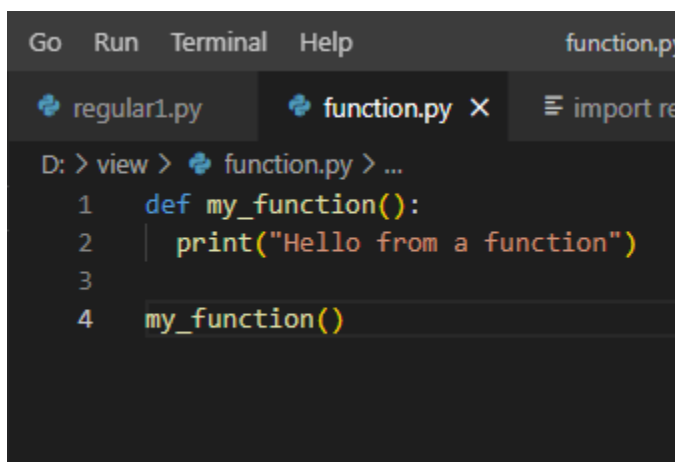
Creating a Function

In Python a function is defined using the `def` keyword

Calling a Function

To call a function, use the function name as shown below

Example (save file name `function.py` as shown below)



```
Go Run Terminal Help function.py
regular1.py function.py X import re
D: > view > function.py > ...
1 def my_function():
2     print("Hello from a function")
3
4 my_function()
```

Run `function.py` file in terminal as shown below to see output:-

```
PS D:\view> python function.py
Hello from a function
PS D:\view> |
```

Arguments

Information can be passed into functions as arguments.

Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.

The following example has a function with one argument (`fname`). When the function is called, we pass along a first name, which is used inside the function to print the full name:

Example:-

```
def my_function(fname):
    print(fname + " Refsnes")
```

```
my_function("Emil")
my_function("Tobias")
my_function("Linus")
```

output:-

```
Emil Refsnes
Tobias Refsnes
Linus Refsnes
```

Number of Arguments

By default, a function must be called with the correct number of arguments. Meaning that if your function expects 2 arguments, you have to call the function with 2 arguments, not more, and not less.

Example

This function expects 2 arguments, and gets 2 arguments:

```
def my_function(fname, lname):  
    print(fname + " " + lname)
```

```
my_function("Emil", "Refsnes")
```

output:-

```
Emil Refsnes
```

Arbitrary Arguments, *args

If you do not know how many arguments that will be passed into your function, add a `*` before the parameter name in the function definition.

This way the function will receive a *tuple* of arguments, and can access the items accordingly:

Example

If the number of arguments is unknown, add a `*` before the parameter name:

```
def my_function(*kids):  
    print("The youngest child is " + kids[2])
```

```
my_function("Emil", "Tobias", "Linus")
```

output :-

```
The youngest child is Linus
```