

Python Programming

A brief of the content!





Functions

A function is a block of organized, reusable code that performs a specific task. It takes in input, performs some operations, and produces an output. Functions allow us to break down complex programs into smaller, manageable pieces. This promotes code reusability and enhances the overall structure of our code.





Defining a Function

To define a function in Python, we use the `def` keyword, followed by the function name, parentheses, and a colon. Let's take a look at the basic syntax:

```
python                                Copy code

def function_name(parameters):
    # Code block
    # Perform operations
    # Return a value (optional)
```

Calling a Function]

Once we have defined a function, we can call it from other parts of our code. To call a function, we simply write the function name followed by parentheses. Let's see an example:

```
python                                Copy code

# Function definition
def greet():
    print("Hello, world!")

# Function call
greet()
```



Arguments

Functions can also accept parameters, which are input values passed to the function. Parameters allow us to make our functions more flexible and versatile. Here's an example:

```
python
Copy code

# Function definition with parameters
def greet(name):
    print("Hello, " + name + "!")

# Function call with arguments
greet("Alice")
greet("Bob")
```

```
Hello, Alice!
Hello, Bob!
```

Return Statement

Sometimes, we want our functions to produce an output or a result. We can achieve this using the return statement. The return statement allows us to specify the value that the function should return back to the caller. Let's see an example:

```
python
Copy code

# Function definition with return statement
def add_numbers(a, b):
    return a + b

# Function call and storing the result
result = add_numbers(5, 3)
print("The sum is:", result)
```

```
The sum is: 8
```



Modules

In Python, a module is a file containing Python definitions and statements that can be imported into other Python scripts or modules. This allows you to reuse code and organize your code into logical units. Python has a large standard library of modules that provide a wide range of functionality. You can also create your own modules to encapsulate related functionality and make it easier to reuse in other projects. To use a module in your Python code, you can import it using the `import` statement and access the functions and variables defined in the module using the dot notation.





How to use Modules

1

```
1 import random
2
3 random_number = random.randint(1, 10)
4 print("Random number: ", random_number)
5
```

Import the entire library and call the functions you need using a dot notation

2

```
1 import random as rnd
2
3 random_number = rnd.randint(1, 10)
4 print("Random number: ", random_number)
5
```

Import the entire library with a short nickname!

3

```
1 from random import randint
2
3 random_number = randint(1, 10)
4 print("Random number: ", random_number)
5
```

Import only the one function you need.



To install libraries you don't have already...

First of all, don't forget to search for installing the library in hand. This will usually lead you to install that package very easily. However, this is a general guide to install new libraries:

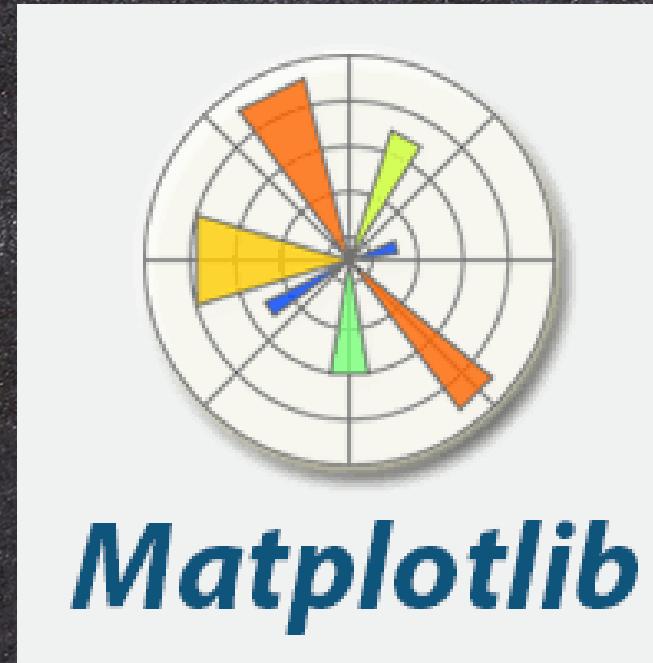
You can use a package manager such as `pip`. Here are the steps to install a library using `pip`:

1. Open a command prompt or terminal window.
2. Type `pip install <library_name>` and press Enter. Replace `<library_name>` with the name of the library you want to install.
3. Wait for `pip` to download and install the library and its dependencies. This may take a few minutes depending on the size of the library and your internet connection speed.
4. Once the installation is complete, you can import the library in your Python code using the `import` statement.





Two well-known libraries:



To visualize almost anything



Numerical Python!



String things

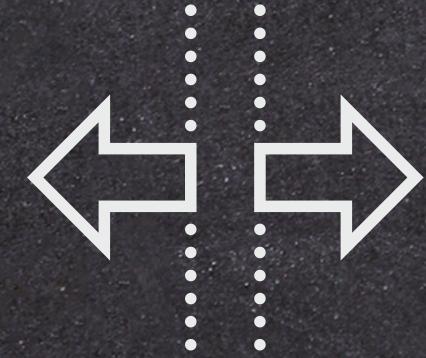




String methods



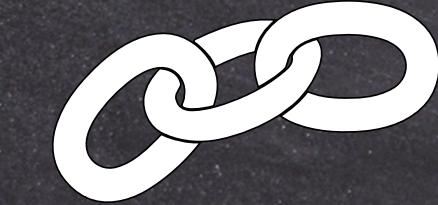
.strip()



.split("something")

Aa

.upper() and .lower()



something.join(a list)



.replace()

And a lot more more methods, waiting you to discover them!



Escape Characters!

Code	Description
\'	Single quotation
\\"	Backslash
\n	New Line
\r	Carriage return
\t	Tab
\b	Backspace
\f	Form feed
\ooo	Octal equivalent
\hhh	Hexadecimal equivalent

And of course, more to discover.



M. Amin Hosseiniya | Ali Rashedi

Presented by



Ali Rashedi



alirashedi2014@gmail.com



[ali-rashedi](https://www.linkedin.com/in/ali-rashedi)



M. Amin Hosseiniya



m.amin.hosseinniya@gmail.com



[mohammadamin-hosseinniya](https://www.linkedin.com/in/mohammadamin-hosseinniya)