

Python Introduction

Python Introduction.....	1
What is Python?	2
What can Python Do?	2
Why Python?.....	2
Good to know.....	3
Table of contents	3
Check the Python in PC.....	3
Python Quickstart	4
Using Blank Lines.....	4
The Python Command Line.....	4
Python Indentation.....	5
Using Comments.....	6
Multiline comments.....	6
Variables	7
Using Variables.....	7
Variable Names	8
Many Values to Multiple Variables.....	8
One Value to Multiple Variables	8
Unpack a Collection	9
Output Variables.....	9
Global Variables	10
The global Keyword.....	11
Python Editors.....	12
User Input	12
Basic Python Course Topics	14

What is Python?

Python is a popular general purpose high level programming language. It was developed by **Guido van Rossum**, and released in **1991**.

What can Python Do?

- Web Development (server-side), there are sophisticated frameworks available ex: Django, Flask etc
- Machine Learning.
 - Recommendation systems. Ex: [Youtube](#), [Netflix](#), [Amazon](#) etc. based on the user browse it provides suggestions..
 - Face recognition, voice recognition apps.
- Software Development ex: Skype.
- Data Analysis/Visualization.
- System scripting, Game Development etc

Why Python?

- **Cross-platform** – Python works on different platforms ([Windows](#), [Mac](#), [Linux](#), [Raspberry Pi](#), etc).
- **Easy to learn** - Python has a simple syntax similar to the English language.
- **Easy to maintain** - Python has syntax is short and simple when compare with other programming languages.
- Python runs on an interpreter system (where code is interpreted line by line)
- **GUI Programming** – Python supports GUI applications.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform.
- **Scalable** – Python provides a better structure and support for large programs
- Python can be treated in a procedural way, an object-oriented way or a functional way.

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as **Thonny**, **Pycharm**, **Netbeans** or **Eclipse** which are particularly useful when managing larger collections of Python files.

Table of contents

- Python Quick start
- The Python Command Line
- Python Indentation
- Using Comment
- Using Variables
- Variable Names
- Many Values to Multiple Variables
- One Value to Multiple Variables
- Unpack a Collection
- Output Variables
- Global Variables
- The global Keyword
- Python Getting Started

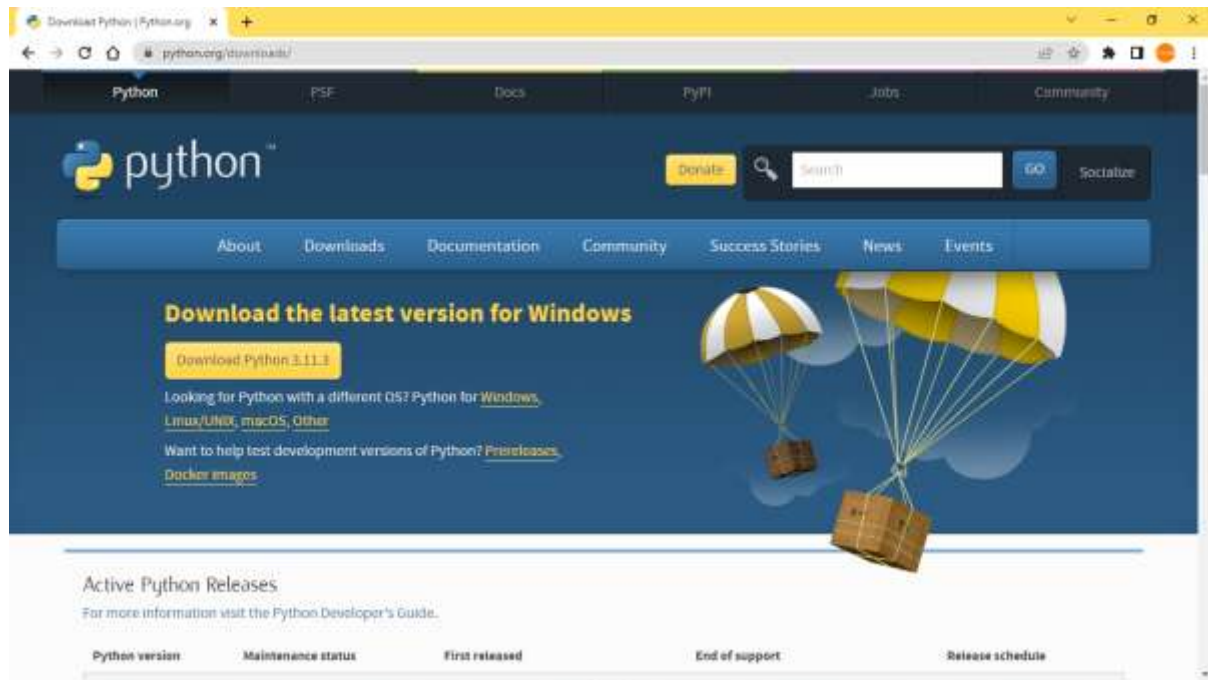
Check the Python in PC

Open command prompt and type the following command.

```
C:\>python -version
```

Download:

website: <https://www.python.org/downloads/>



Python Quickstart

Python is an interpreted programming language, you need to write a python code into a texteditor and save it as filename.py. and execute it.

Syntax to run a file at command line:

```
C:\Users\yourdir>python helloworld.py
```

helloworld.py

```
print("Hello, World!")
```

Using Blank Lines

```
print("Hello, World!")  
print("\n\n Welcome to Python Coding!")
```

The Python Command Line

Python can be run as a command line itself. To open python interpreter, at prompt type

```
C:\>python or c:\> py
```

Then you will see the python prompt(>>>) where you can type python commands as shown in the below picture:

```
>>> print("Hello, World!")
```

Output:

```
Hello, World!
```

```
C:\Users\Your Name>python
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, World!")
```

To quit the python command line interface:

```
exit()
```

Python Indentation

Python uses indentation to indicate a block of code.

```
if 5 > 2:
    print("yes")
```

Python will give you an error if you skip the indentation:

Example

Syntax Error:

```
if 5 > 2:
print("Yes!")
```

The number of spaces is up to you as a programmer, but it has to be at least one.

```
if 5 > 2:
    print("Yes!")
```

```
if 5 > 2:
    print("Yes!")
```

You have to use the same number of spaces in the same block of code, otherwise Python will give you an error:

```
if 5 > 2:
    print("Yes, Five is greater than two!")
    print("Yes it is!")
```

Using Comments

Python comment prefixed with (#)

```
#This is a comment
print("Hello, World!")
```

```
print("Hello, World!") #This is a comment
```

A comment does not have to be text that explains the code, it can also be used to prevent Python from executing code:

```
#print("Hello, World!")
print("Cheers, Mate!")
```

Multiline comments

```
...
line one
line two
line N..
...
```

Variables

Variables are containers for storing data values.

Using Variables

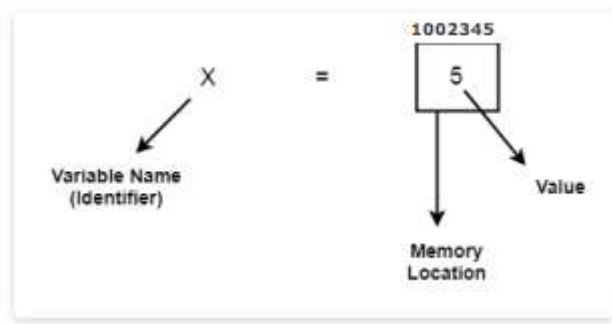
A variable is created the moment you first assign a value to it.

Example

```
x = 5
print(x)
```

1. 'x' is a variable name that references to a memory location in which 5 is stored.
2. Print function prints the value of x.

A variable in the memory:



```
y = "John"
print(y)
```

Variables do not need to be declared with a specific *type*, based on the assigned value, type has been set internally.

```
x = 4          # x is of type int
x = "Sally"    # x is now of type str
print(x)
```

More Examples:

```
a = "Hello World"
b = 100
```

```
c = 25.50
d = 5+6j

print ("Message: ", a)
print (b, c, b-c)
print(pow(100, 0.5), pow(c,2))
```

Variable Names

A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume). Rules for Python variables:

- A variable name must start with a **letter** or the **underscore** character
- A variable name cannot start with a **number**
- Contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are **case-sensitive** (age, Age and AGE are three different variables)

```
myvar = "John"
my_var = "John"
my_var = "John"
myVar = "John"
MYVAR = "John"
myvar2 = "John"
```

Not Allowed:

```
2myvar = "John"
my-var = "John"
my var = "John"
```

Many Values to Multiple Variables

```
x, y, z = "Orange", "Banana", "Cherry"
print(x)
print(y)
print(z)
```

One Value to Multiple Variables

```
x = y = z = "Orange"
print(x)
```

```
print(y)
print(z)
```

Unpack a Collection

If you have a collection of values in a list, tuple etc. Python allows you to extract the values into variables. This is called *unpacking*.

```
fruits = ["apple", "banana", "cherry"]
x, y, z = fruits
print(x)
print(y)
print(z)
```

Output Variables

The Python `print()` function is often used to output variables.

```
x = "Python is awesome"
print(x)
```

```
x = "Python"
y = "is"
z = "awesome"
print(x, y, z)
```

```
x = "Python "
y = "is "
z = "awesome"
print(x + y + z)
```

```
x = 5
y = 10
print(x + y)
```

```
x = 5
y = "John"
print(x + y)
```

printing multiple variables

```
x = 5
y = "John"
print(x, y)
```

printing multiple variables with specific separator

```
print("Hello", "how are you?", sep="---")
```

Ex2:

```
#!/usr/bin/python3.11
```

```
city="Hyderabad"
state="Telangana"
country="India"
print(city, state, country, sep=',')
```

To make these two lines appear in the same line, define end parameter in the first print() function and set it to a whitespace string " ".

```
city="Hyderabad"
state="Telangana"
country="India"

print("City:", city, end=" ")
print("State:", state)
```

e-Trainings.in

Global Variables

Variables that are created outside of a function (as in all of the examples above) are known as global variables.

Global variables can be used by everyone, both inside of functions and outside.

```
x = "awesome"

def myfunc():
    print("Python is " + x)
```

```
myfunc()
```

If you create a variable with the same name inside a function, this variable will be local, and can only be used inside the function. The global variable with the same name will remain as it was, global and with the original value. Global variable can be accessed by multiple functions.

The global Keyword

Normally, when you create a variable inside a function, that variable is local, and can only be used inside that function.

To create a global variable inside a function, you can use the `global` keyword.

If you use the `global` keyword, the variable belongs to the global scope:

```
def myfunc():  
    global x  
    x = "fantastic"
```

```
myfunc()
```

```
print("Python is " + x)
```

Example

To change the value of a global variable inside a function, refer to the variable by using the `global` keyword:

```
x = "awesome"
```

```
def myfunc():  
    global x  
    x = "fantastic"
```

```
myfunc()
```

```
print("Python is " + x)
```

Python Editors

PyCharm

<https://www.jetbrains.com/pycharm/download>

VSCode

<https://code.visualstudio.com/download>

Install code runner extension.

Install Microsoft python language support extension.

User Input

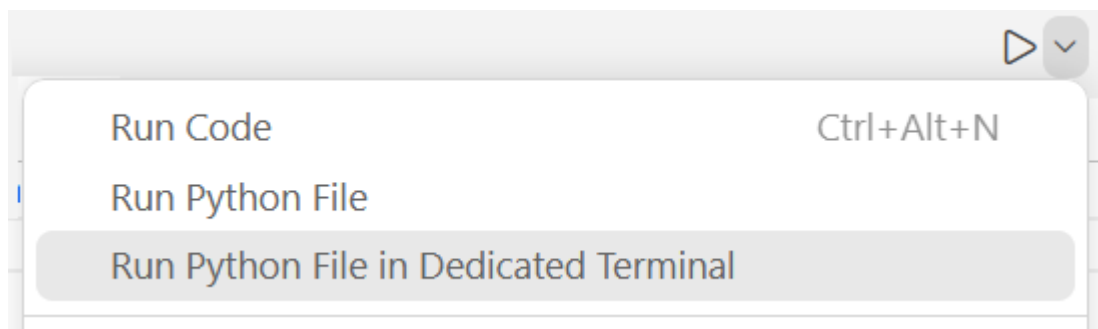
Python allows for user input.

That means we are able to ask the user for input.

Python 3.6 uses the `input()` method.

The following example asks for the username, and when you entered the username, it gets printed on the screen:

How to run?: Run in python program terminal window or choose 'Run Python File in Dedicated Terminal' option Run(playbutton) drop down menu.



```
username = input("Enter username:")  
print("Username is: " + username)
```

```
#!/usr/bin/python3.11

name = input("Enter your name - ")
city = input("Enter city name - ")
print ("Hello My name is", name)
print ("I am from ", city)
```

```
numeric input
width = input("Enter width : ")
height = input("Enter height : ")
area = width*height
print ("Area of rectangle = ", area)
```

#string to int

```
width = int(input("Enter width : "))
height = int(input("Enter height : "))
```

```
area = width*height
print ("Area of rectangle = ", area)
```

#!/usr/bin/python3.11
#string to float

```
amount = float(input("Enter Amount : ")) rate = float(input("Enter rate  
of interest : "))
```

```
interest = amount*rate/100
print ("Amount: ", amount, "Interest: ", interest)
```

Practice:

1. Print Person Details: (Name, City, Email, Phone)
2. Input and print Car Details (Company, Color, Mileage, year)
3. print Pet Details: [Penname, Pet type (Dog/Cat/Bird/Fish), Color, Age, Price]
4. print Book Details (Book title, Author, price, year)
5. Print product details Name, price, qty
6. print bill amount ,Tax including
7. Print Student Details (Name, RollNo, (Marks1, Marks2, Marks3), Calculate total, avg.
8. Print fare of distance travelled.
9. Find simple interest (si) formula: $SI = (PNR)/100$;

10. Calculate area of Square($SQA = side * side$)
11. Rectangle area = length * breadth;
12. Triangle area = $1/2bh$
13. Circle Area = $(PI * r * r)$

Basic Python Course Topics

-  Chapter1-Python-Introduction
-  Chapter2-Python-Datatypes
-  Chapter3-Python-Operators
-  Chapter4-Python-Conditions
-  Chapter5-Python-Strings
-  Chapter6-Python-Booleans
-  Chapter7-Python-range
-  Chapter8-Python-Loops
-  Chapter9-Python Lists
-  Chapter10-Python-Tuples
-  Chapter11-Python-Sets
-  Chapter12-Python-Dictionaries
-  Chapter13-Python-Functions
-  Chapter14-Python-Lambda
-  Chapter15-Python-Arrays

Trainer:



Anita S.

e-trainings Online Training Center

<https://e-trainings.in>

[Python Roadmap](#)

Python Roadmap

