# PYTHON

What is Python

Python is a general-purpose, dynamically typed, high-level, compiled and interpreted, garbage-collected, and purely object-oriented programming language that supports procedural, object-oriented, and functional programming.

History of Python

**Python was created by Guido van Rossum**. In the late 1980s, Guido van Rossum, a Dutch programmer, began working on Python while at the Centrum Wiskunde & Informatica (CWI) in the Netherlands. He wanted to create a successor to the **ABC programming language** that would be easy to read and efficient.

Why learn Python?

Python provides many useful features to the programmer. These features make it the most popular and widely used language. We have listed below few-essential features of Python.

* **Easy to use and Learn:** Python has a simple and easy-to-understand syntax, unlike traditional languages like C, C++, Java, etc., making it easy for beginners to learn.
* **Expressive Language:** It allows programmers to express complex concepts in just a few lines of code or reduces Developer's Time.
* **Interpreted Language:** Python does not require compilation, allowing rapid development and testing. It uses Interpreter instead of Compiler.
* [**Object-Oriented Language**](https://www.tpointtech.com/python-oops-concepts): It supports object-oriented programming, making writing reusable and modular code easy.
* **Learn Standard Library:** Python's standard library contains many modules and functions that can be used for various tasks, such as [string manipulation](https://www.tpointtech.com/string-manipulation-in-python), [web programming](https://www.tpointtech.com/django-tutorial), and more.
* **GUI Programming Support:** Python provides several GUI frameworks, such as [Tkinter](https://www.tpointtech.com/python-tkinter) and [PyQt](https://www.tpointtech.com/pyqt-library-in-python), allowing developers to create desktop applications easily.
* **Integrated:** Python can easily integrate with other languages and technologies, such as C/C++, Java, and . NET.
* **Big Data and Machine Learning:** Python has become the go-to language for big data and machine learning. Python has become popular among data scientists and machine learning engineers with libraries like [NumPy](https://www.tpointtech.com/numpy-tutorial), [Pandas](https://www.tpointtech.com/python-pandas), [Scikit-learn](https://www.tpointtech.com/what-is-sklearn-in-python), [TensorFlow](https://www.tpointtech.com/tensorflow), and more.

Where is Python used?

Python is a general-purpose, popular programming language, and it is used in almost every technical field. The various areas of Python use are given below.

* **Data Science:** Data Science is a vast field, and Python is an important language for this field because of its simplicity, ease of use, and availability of powerful data analysis and visualization libraries like [NumPy](https://www.tpointtech.com/numpy-tutorial), [Pandas](https://www.tpointtech.com/python-pandas), and [Matplotlib](https://www.tpointtech.com/matplotlib).

Features of Python.

* **Easy to use and Read -** Python's syntax is clear and easy to read, making it an ideal language for both beginners and experienced programmers. This simplicity can lead to faster development and reduce the chances of errors.
* **Dynamically Typed** - The data types of variables are determined during run-time. We do not need to specify the data type of a variable during writing codes.
* **High-level** - High-level language means human readable code.
* **Compiled and Interpreted** - Python code first gets compiled into bytecode, and then interpreted line by line. When we download the Python in our system form [org](https://www.python.org/) we download the default implement of Python known as CPython. CPython is considered to be Complied and Interpreted both.
* **Garbage Collected** - Memory allocation and de-allocation are automatically managed. Programmers do not specifically need to manage the memory.
* **Purely Object-Oriented** - It refers to everything as an object, including numbers and strings.
* **Cross-platform Compatibility** - Python can be easily installed on Windows, macOS, and various Linux distributions, allowing developers to create software that runs across different operating systems.
* **Rich Standard Library** - Python comes with several standard libraries that provide ready-to-use modules and functions for various tasks, ranging from **web development** and **data manipulation** to **machine learning** and **networking**.
* **Open Source** - Python is an open-source, cost-free programming language. It is utilized in several sectors and disciplines as a result.

Instead of Semicolon as used in other languages, Python ends its statements with a NewLine character.

Python is a case-sensitive language, which means that uppercase and lowercase letters are treated differently. For example, 'name' and 'Name' are two different variables in Python.

In Python, comments can be added using the '#' symbol. Any text written after the '#' symbol is considered a comment and is ignored by the interpreter. This trick is useful for adding notes to the code or temporarily disabling a code block. It also helps in understanding the code better by some other developers.

['If'](https://www.tpointtech.com/python-if-else), 'otherwise', ['for'](https://www.tpointtech.com/python-for-loop), ['while'](https://www.tpointtech.com/python-while-loop), 'try', 'except', and 'finally' are a few reserved keywords in Python that cannot be used as variable names. These terms are used in the language for particular reasons and have fixed meanings. If you use these keywords, your code may include errors, or the interpreter may reject them as potential new Variables.

## Python Syntax

As we learned in the previous page, Python syntax can be executed by writing directly in the Command Line:

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

Or by creating a python file on the server, using the .py file extension, and running it in the Command Line:

C:\Users\Your Name>python myfile.py

### Python Indentation

Indentation refers to the spaces at the beginning of a code line.

Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important. Python uses indentation to indicate a block of code. Python will give you an error if you skip the indentation:

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

if 5 > 2:  
 print ("Five is greater than two!")   
if 5 > 2:  
        print ("Five is greater than two!")

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

Example

Syntax Error:

if 5 > 2:  
 print("Five is greater than two!")  
        print("Five is greater than two!")

### Python Variables

In Python, variables are created when you assign a value to it Variables are containers for storing data values. A variable is created the moment you first assign a value to it.

Variables in Python:

x = 5  
y = "Hello, World!"

Python has no command for declaring a variable.

Variables do not need to be declared with any particular type, and can even change type after they have been set.

Example

x = 4       # x is of type int  
x = "Sally" # x is now of type str

x = "daniel" # x is now of type str

print(x)

### Casting

**Casting in Python** refers to converting a variable from one data type to another. If you want to specify the data type of a variable, this can be done with casting.

Example

x = str (3)     # x will be '3'  
y = int (3)     # y will be 3  
z = float(3)   # z will be 3.0

### Get the Type

You can get the data type of a variable with the type () function.

Example

x = 5  
y = "John"  
print(type(x))  
print(type(y))

### Case-Sensitive

Variable names are case-sensitive.

Example

This will create two variables:

a = 4  
A = "Sally"  
#A will not overwrite a

### Variable Names

A variable can have a short name (like x and y) or a more descriptive name (age, car name, 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
* A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_)
* Variable names are case-sensitive (age, Age and AGE are three different variables)
* A variable name cannot be any of the [Python keywords](https://www.w3schools.com/python/python_ref_keywords.asp).

Example

Legal variable names:

myvar = "John"  
my\_var = "John"  
\_my\_var = "John"  
myVar = "John"  
MYVAR = "John"  
myvar2 = "John"

Example

Illegal variable names:

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

### Multi Words Variable Names

Variable names with more than one word can be difficult to read.

There are several techniques you can use to make them more readable

### Camel Case

Each word, except the first, starts with a capital letter:

myVariableName = "John"

### Pascal Case

Each word starts with a capital letter:

MyVariableName = "John"

### Snake Case

Each word is separated by an underscore character:

my\_variable\_name = "John"

### Many Values to Multiple Variables

Python allows you to assign values to multiple variables in one line:

Example

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

**Note:** Make sure the number of variables matches the number of values, or else you will get an error.

### 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*.

Example

Unpack a list:

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

### Output Variables

## Python Comments

Comments can be used to explain Python code. Comments can be used to make the code more readable. Comments can be used to prevent execution when testing code.

### Creating a Comment

Comments starts with a #, and Python will ignore them: Python does not really have a syntax for multiline comments.

To add a multiline comment, you could insert a # for each line:

Or, not quite as intended, you can use a multiline string.

Since Python will ignore string literals that are not assigned to a variable, you can add a multiline string (triple quotes) in your code, and place your comment inside it:

Example

"""  
This is a comment  
written in  
more than just one line  
"""  
print ("Hello, World!")