* Python is a general-purpose, dynamic, high-level, and interpreted programming language.
* It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures.
* Python is a case-sensitive language, which means that uppercase and lowercase letters are treated differently.
* Python is an easy-to-learn yet powerful and versatile scripting language, which makes it attractive for Application Development.
* With its interpreted nature, Python's syntax and dynamic typing make it an ideal language for scripting and rapid application development.
* It is a multipurpose programming language because it can be used with web, enterprise, 3D CAD, etc.
* Python is an open-source, cost-free programming language. It is utilized in several sectors and disciplines as a result.
* In Python, code readability and maintainability are important. As a result, even if the code was developed by someone else, it is easy to understand and adapt by some other developer.
* Python has many third-party libraries that can be used to make its functionality easier. These libraries cover many domains, for example, web development, scientific computing, data analysis, and more.

**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:** It supports object-oriented programming, making writing reusable and modular code easy.
* **Open Source Language:** Python is open source and free to use, distribute and modify.
* **Extensible:** Python can be extended with modules written in C, C++, or other languages.
* **Learn Standard Library:** Python's standard library contains many modules and functions that can be used for various tasks, such as string manipulation, web programming, and more.
* **GUI Programming Support:** Python provides several GUI(Graphical User Interface) frameworks, such as Tkinter and PyQt, allowing developers to create desktop applications easily.
* **Integrated:** Python can easily integrate with other languages and technologies, such as C/C++, Java, and . NET.
* **Embeddable:** Python code can be embedded into other applications as a scripting language.
* **Dynamic Memory Allocation:** Python automatically manages memory allocation, making it easier for developers to write complex programs without worrying about memory management.
* **Wide Range of Libraries and Frameworks:** Python has a vast collection of libraries and frameworks, such as NumPy, Pandas, Django, and Flask, that can be used to solve a wide range of problems.
* **Versatility:** Python is a universal language in various domains such as web development, machine learning, data analysis, scientific computing, and more.

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.

Variables

* Variables are containers for storing data values.

**Creating Variables**

* Python has no command for declaring a variable.
* A variable is created the moment you first assign a value to it.
* Variables do not need to be declared with any particular type, and can even change type after they have been set.
* If you want to specify the data type of a variable, this can be done with casting.
* You can get the data type of a variable with the type() function
* String variables can be declared either by using single or double quotes
* Variable names are case-sensitive.

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).

**Identifiers in Python**

**Identifier**is a user-defined name given to a variable, function, class, module, etc. The identifier is a combination of character digits and an underscore. They are case-sensitive i.e., ‘num’ and ‘Num’ and ‘NUM’ are three different identifiers in python.

**Rules for Naming Python Identifiers**

* It cannot be a reserved python keyword.
* It should not contain white space.
* It can be a combination of A-Z, a-z, 0-9, or underscore.
* It should start with an alphabet character or an underscore ( \_ ).
* It should not contain any special character other than an underscore ( \_ ).

**Examples of Python Identifiers**

***Valid identifiers:***

* *var1*
* *\_var1*
* *\_1\_var*
* *var\_1*

***Invalid Identifiers***

* *!var1*
* *1var*
* *1\_var*
* *var#1*
* *var 1*

# **Python Keywords**

* Python has a set of keywords that are reserved words that cannot be used as variable names, function names, or any other identifiers:

|  |  |
| --- | --- |
| **Keyword** | **Description** |
| [and](https://www.w3schools.com/python/ref_keyword_and.asp) | A logical operator |
| [as](https://www.w3schools.com/python/ref_keyword_as.asp) | To create an alias |
| [assert](https://www.w3schools.com/python/ref_keyword_assert.asp) | For debugging |
| [break](https://www.w3schools.com/python/ref_keyword_break.asp) | To break out of a loop |
| [class](https://www.w3schools.com/python/ref_keyword_class.asp) | To define a class |
| [continue](https://www.w3schools.com/python/ref_keyword_continue.asp) | To continue to the next iteration of a loop |
| [def](https://www.w3schools.com/python/ref_keyword_def.asp) | To define a function |
| [del](https://www.w3schools.com/python/ref_keyword_del.asp) | To delete an object |
| [elif](https://www.w3schools.com/python/ref_keyword_elif.asp) | Used in conditional statements, same as else if |
| [else](https://www.w3schools.com/python/ref_keyword_else.asp) | Used in conditional statements |
| [except](https://www.w3schools.com/python/ref_keyword_except.asp) | Used with exceptions, what to do when an exception occurs |
| [False](https://www.w3schools.com/python/ref_keyword_false.asp) | Boolean value, result of comparison operations |
| [finally](https://www.w3schools.com/python/ref_keyword_finally.asp) | Used with exceptions, a block of code that will be executed no matter if there is an exception or not |
| [for](https://www.w3schools.com/python/ref_keyword_for.asp) | To create a for loop |
| [from](https://www.w3schools.com/python/ref_keyword_from.asp) | To import specific parts of a module |
| [global](https://www.w3schools.com/python/ref_keyword_global.asp) | To declare a global variable |
| [if](https://www.w3schools.com/python/ref_keyword_if.asp) | To make a conditional statement |
| [import](https://www.w3schools.com/python/ref_keyword_import.asp) | To import a module |
| [in](https://www.w3schools.com/python/ref_keyword_in.asp) | To check if a value is present in a list, tuple, etc. |
| [is](https://www.w3schools.com/python/ref_keyword_is.asp) | To test if two variables are equal |
| [lambda](https://www.w3schools.com/python/ref_keyword_lambda.asp) | To create an anonymous function |
| [None](https://www.w3schools.com/python/ref_keyword_none.asp) | Represents a null value |
| [nonlocal](https://www.w3schools.com/python/ref_keyword_nonlocal.asp) | To declare a non-local variable |
| [not](https://www.w3schools.com/python/ref_keyword_not.asp) | A logical operator |
| [or](https://www.w3schools.com/python/ref_keyword_or.asp) | A logical operator |
| [pass](https://www.w3schools.com/python/ref_keyword_pass.asp) | A null statement, a statement that will do nothing |
| [raise](https://www.w3schools.com/python/ref_keyword_raise.asp) | To raise an exception |
| [return](https://www.w3schools.com/python/ref_keyword_return.asp) | To exit a function and return a value |
| [True](https://www.w3schools.com/python/ref_keyword_true.asp) | Boolean value, result of comparison operations |
| [try](https://www.w3schools.com/python/ref_keyword_try.asp) | To make a try...except statement |
| [while](https://www.w3schools.com/python/ref_keyword_while.asp) | To create a while loop |
| with | Used to simplify exception handling |
| yield | To end a function, returns a generator |

Comments in python

* 1. single-line comment

A single-line comment of Python is the one that has a hashtag # at the beginning of it and continues until the finish of the line. If the comment continues to the next line, add a hashtag to the subsequent line and resume the conversation.

### 2. **Multi-Line Comments in Python**

Python does not provide the option for [multiline comments](https://www.geeksforgeeks.org/multiline-comments-in-python/).

We can multiple hashtags (#) to write multiline comments in Python. Each and every line will be considered as a single-line comment.