

# Learn Python Programming Fundamentals: A Beginner's Guide [Updated 2020]

## Fundamental Concepts Of Python Programming Language



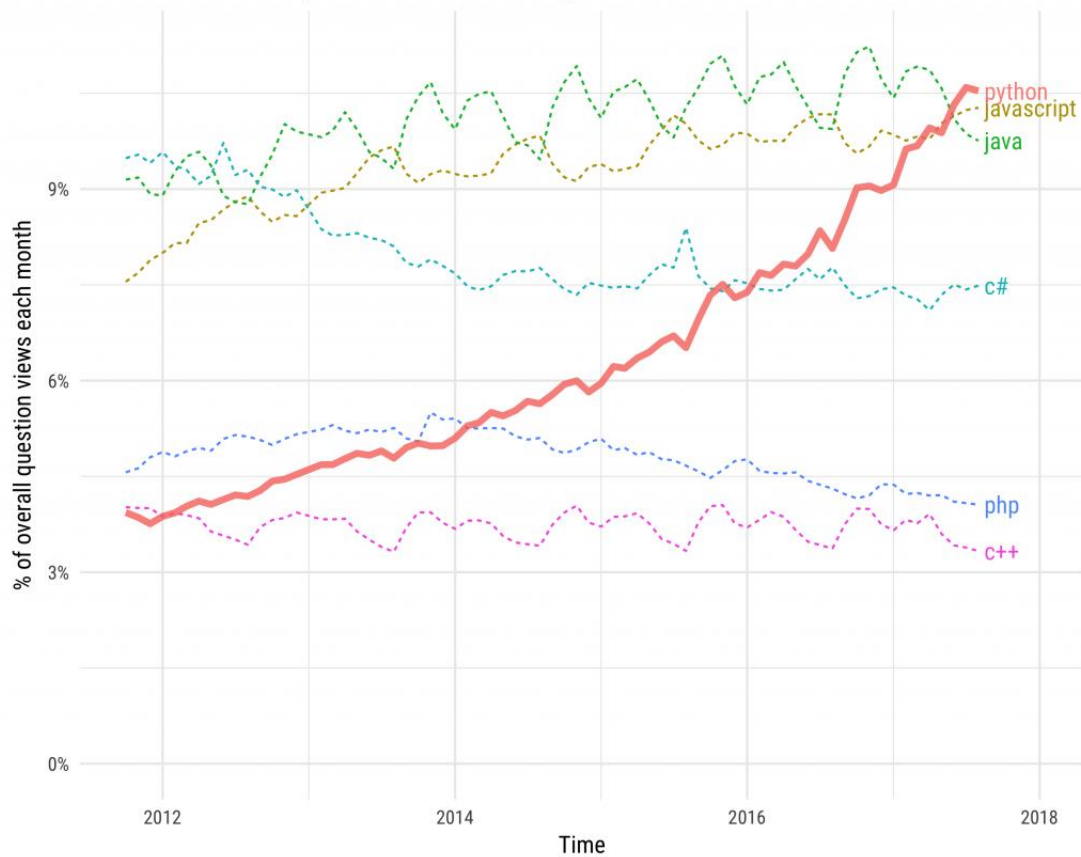
**simpliv**  
Learn. Teach. Earn

Python is one of the powerful, high-level, easy to learn programming language that provides a huge number of applications. Some of its features, such as being object-oriented and open source, having numerous IDE's, etc. make it one of the most in-demand programming languages of the present IT industry.

According to **TIOBE** index, as of January 2020, [Python](#) is one of the popular programming languages. By looking at the popularity of this programming language, many IT professionals, both beginners as well as experienced alike, are willing to build their career as a [Python developer](#).

## Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries



### Image source

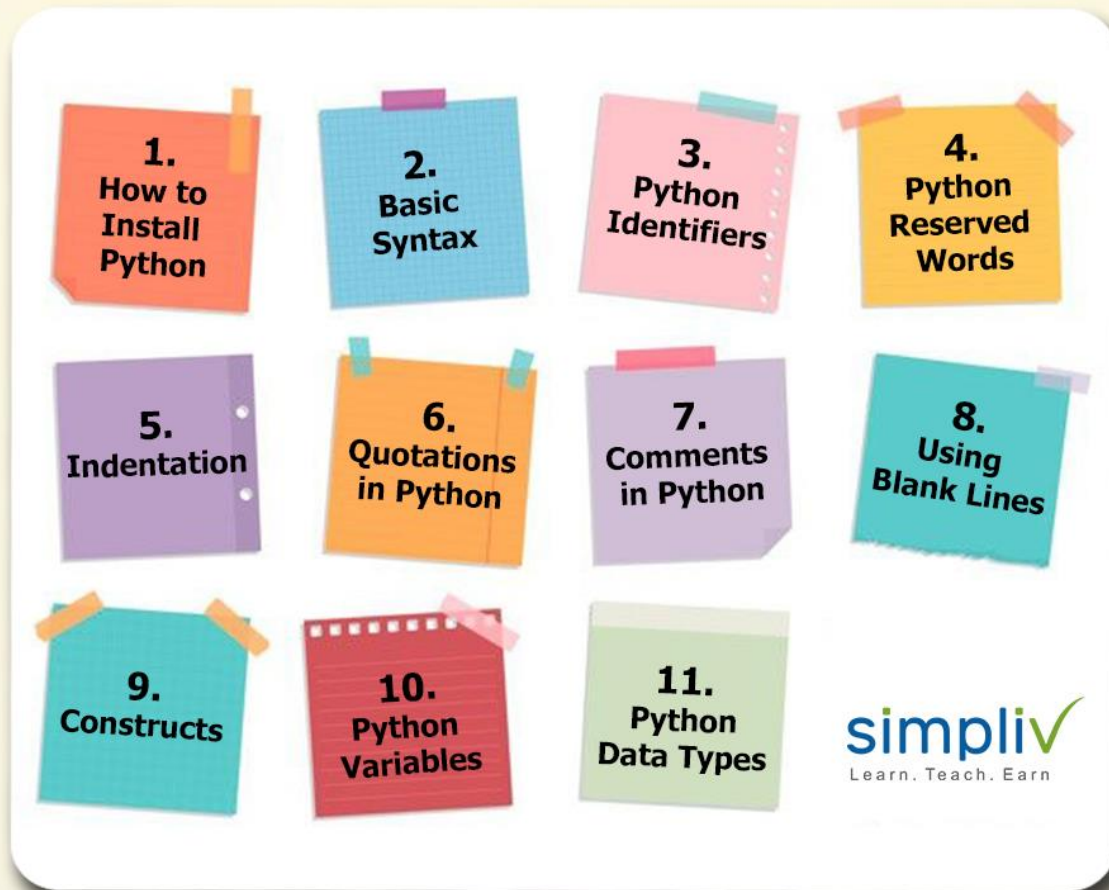
Many people have daunting questions like:

- **How one can start to learn Python?**
- What are the fundamental concepts you need to know to **learn Python?**

With an aim to help similar concerns, Simpliv is presenting this blog to discuss about the various fundamental concepts of **Python** programming and take you along to start writing Python programs on your own.

Before proceeding further, at this point, we would like to suggest that you read **blog** (first blog in this series) on introduction to Python programming language.

Without further ado, let us quickly look at the topics we will be covering in this blog:



- How to install Python
- Basic syntax
- Python identifiers
- Python reserved words
- Indentation
- Quotations in Python
- Comments in Python
- Using Blank lines
- Constructs
- Python Variables

- Python Data Types.

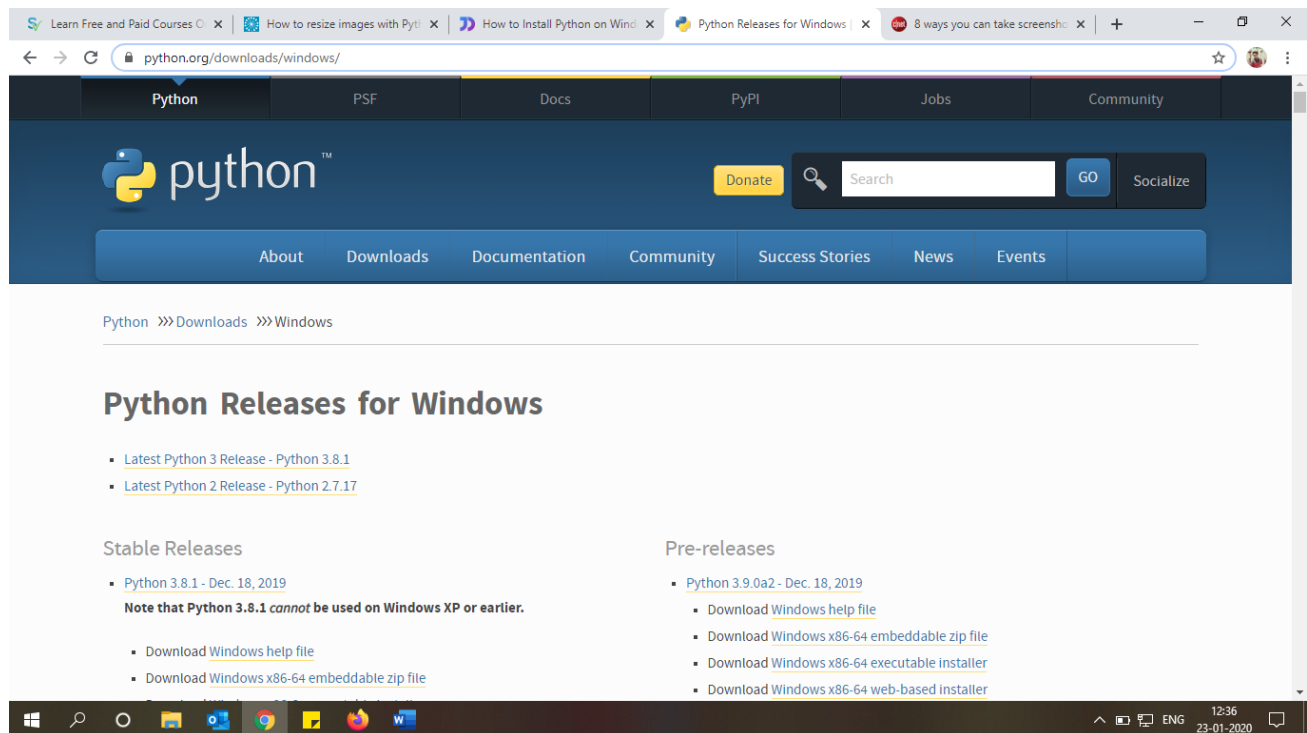
## Let us look at the 8 Steps to install Python

Let us start by learning the steps to install Python. The following are the steps need to be followed while installing Python on Windows:

### Step 1:

Download python.exe or zip bundle from Python official website

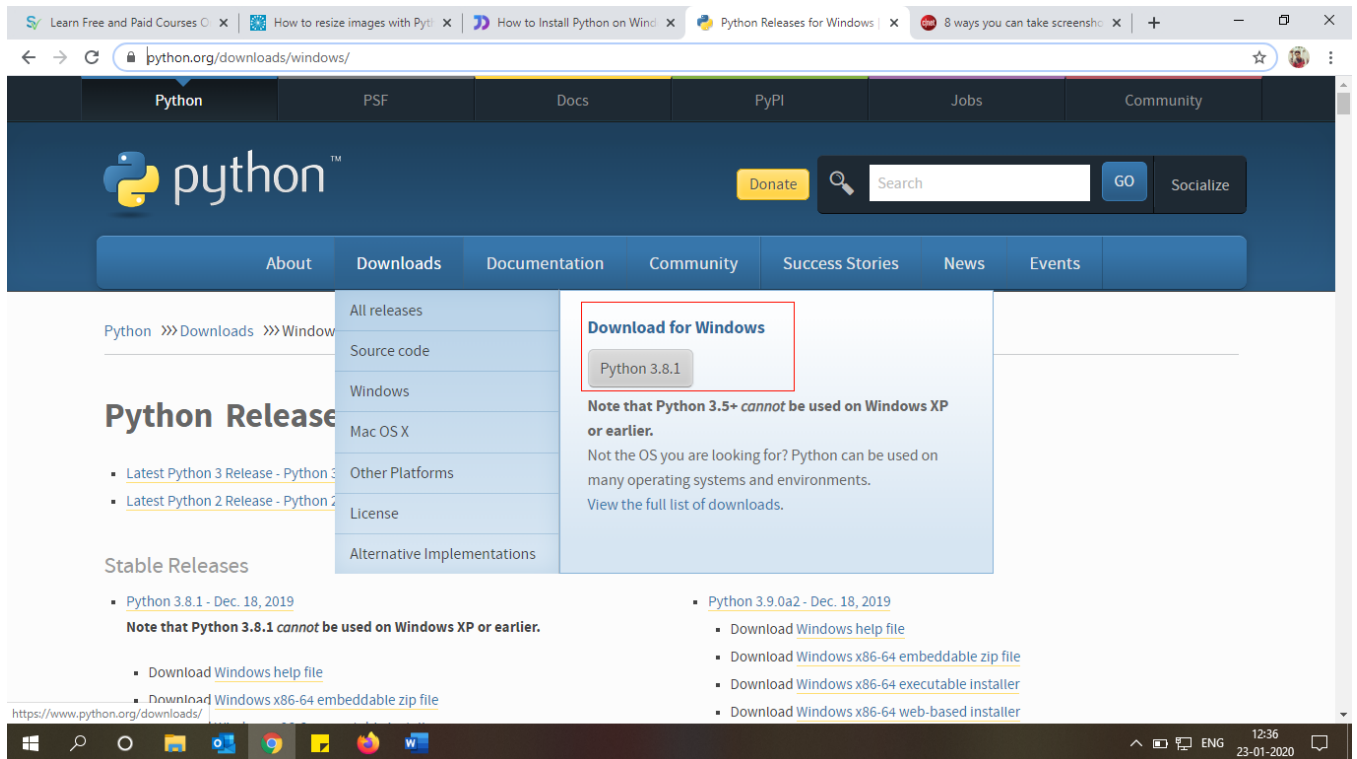
<https://www.python.org/downloads/windows/>.



The screenshot shows a web browser window displaying the Python.org website. The browser's address bar shows the URL [python.org/downloads/windows/](https://www.python.org/downloads/windows/). The website's navigation menu includes links for Python, PSF, Docs, PyPI, Jobs, and Community. The main content area is titled "Python Releases for Windows" and lists the latest Python 3 Release (Python 3.8.1) and the latest Python 2 Release (Python 2.7.17). Under the "Stable Releases" section, it lists Python 3.8.1 (Dec. 18, 2019) with a note that it cannot be used on Windows XP or earlier, and provides links to download the Windows help file and the Windows x86-64 embeddable zip file. Under the "Pre-releases" section, it lists Python 3.9.0a2 (Dec. 18, 2019) and provides links to download the Windows help file, the Windows x86-64 embeddable zip file, the Windows x86-64 executable installer, and the Windows x86-64 web-based installer. The Windows taskbar is visible at the bottom of the screenshot, showing the Start button, search icon, and several application icons.

### Step 2:

Select Downloads and download python.exe file for Windows.



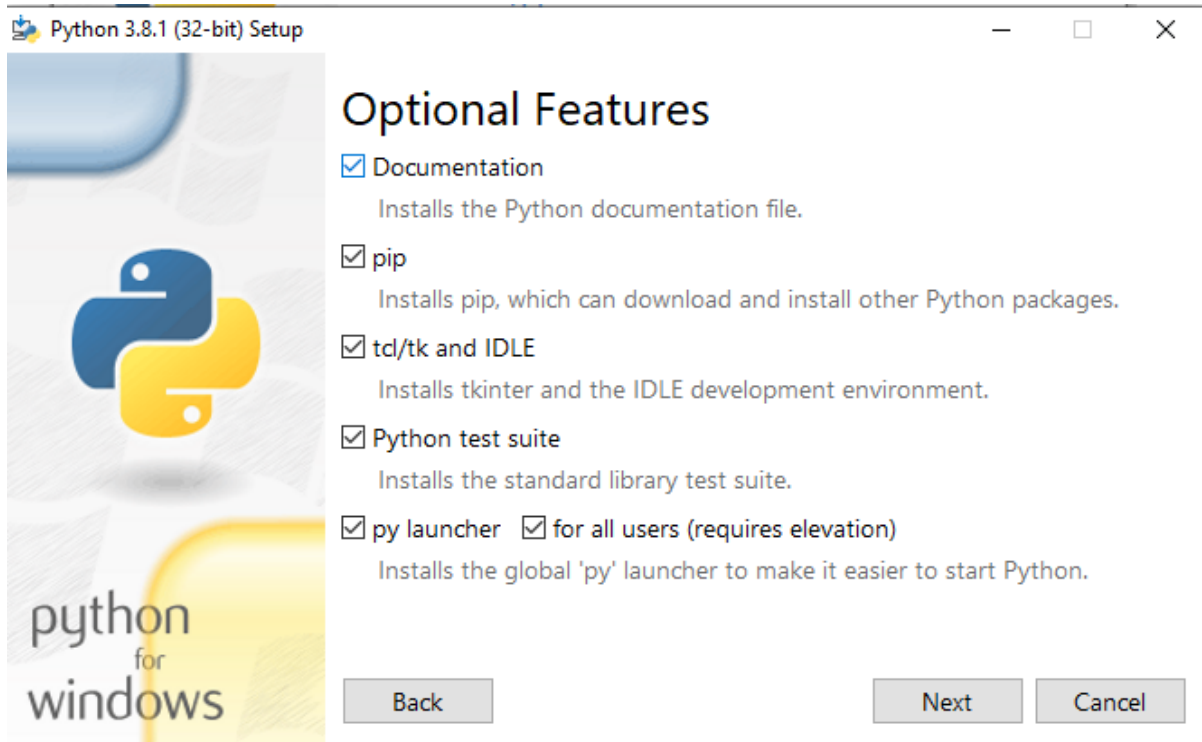
### Step 3:

Once the installer is downloaded, run the Python installer. Check on Install launcher for all users.



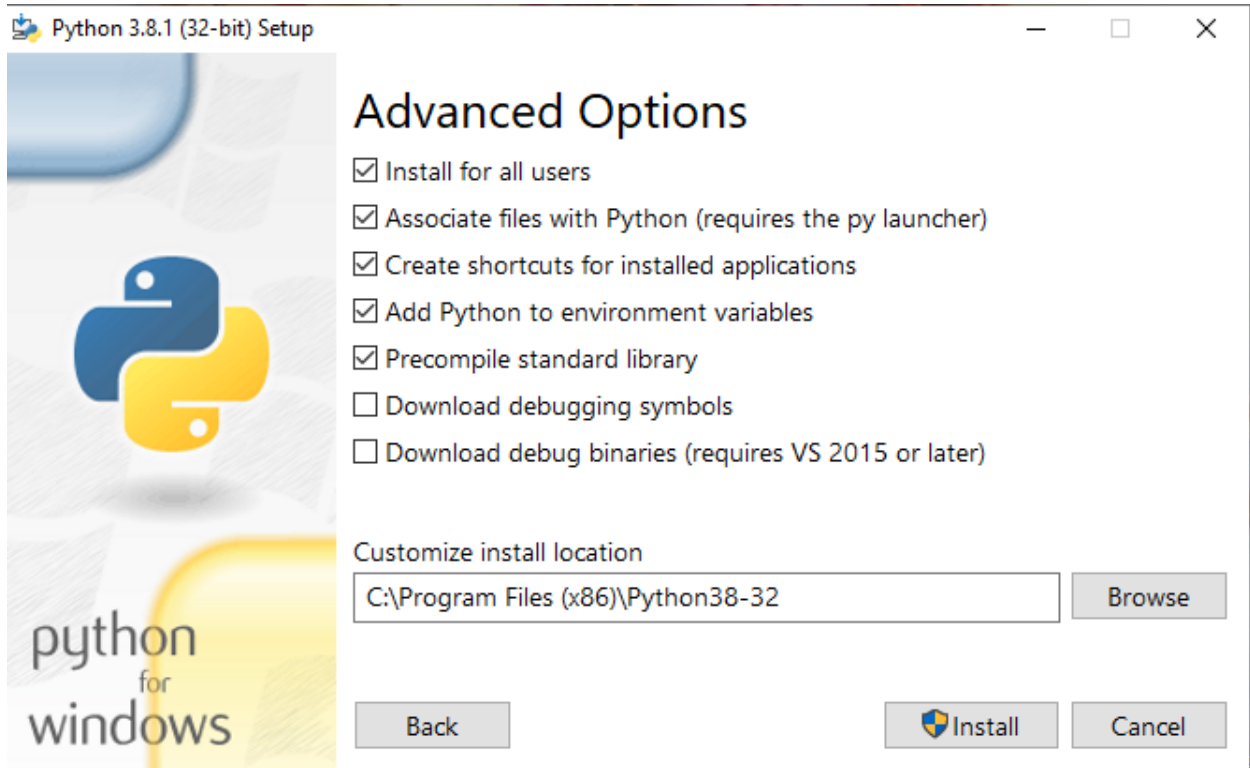
#### Step 4:

Select Customize installation. Check on all settings Document, pip, tcl/tk, python test suite, py launcher, for all users. Click on Next.



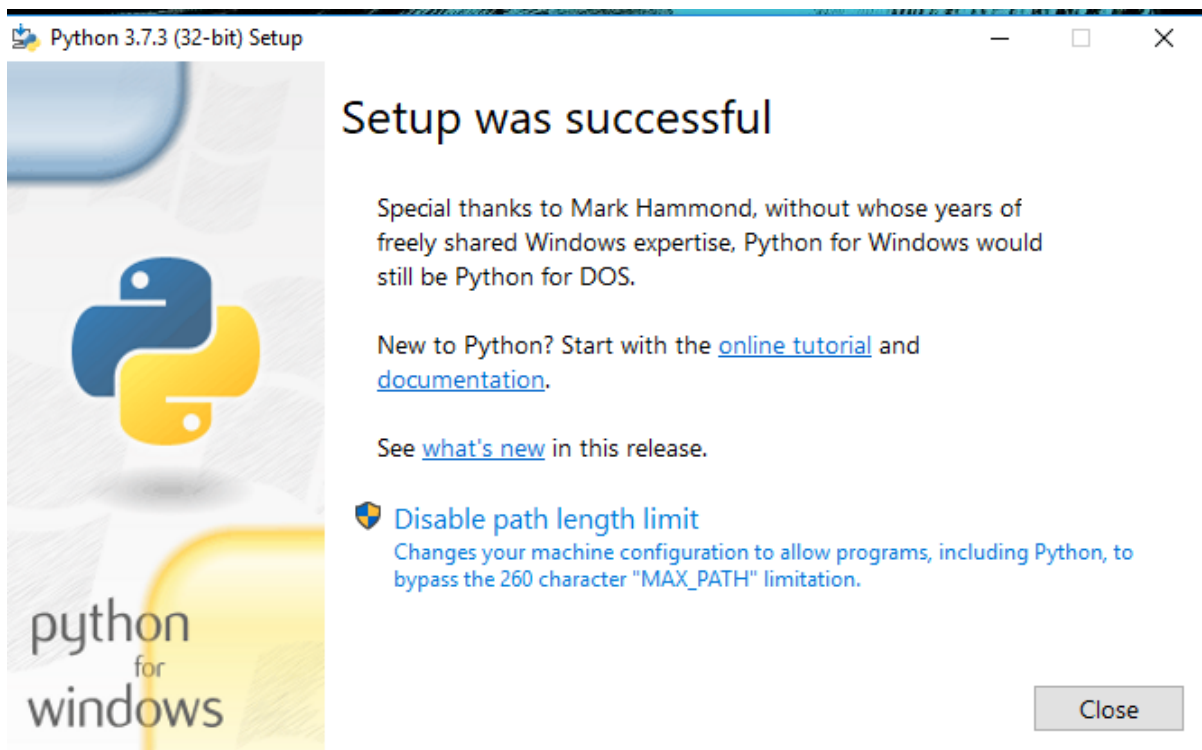
#### Step 5:

This takes you to **Advanced Options** available while installing Python. Here, select the **Install for all users** and **Add Python to environment variables** check boxes.



### Step 6:

Once installation is done, you can see that the installation is successful.



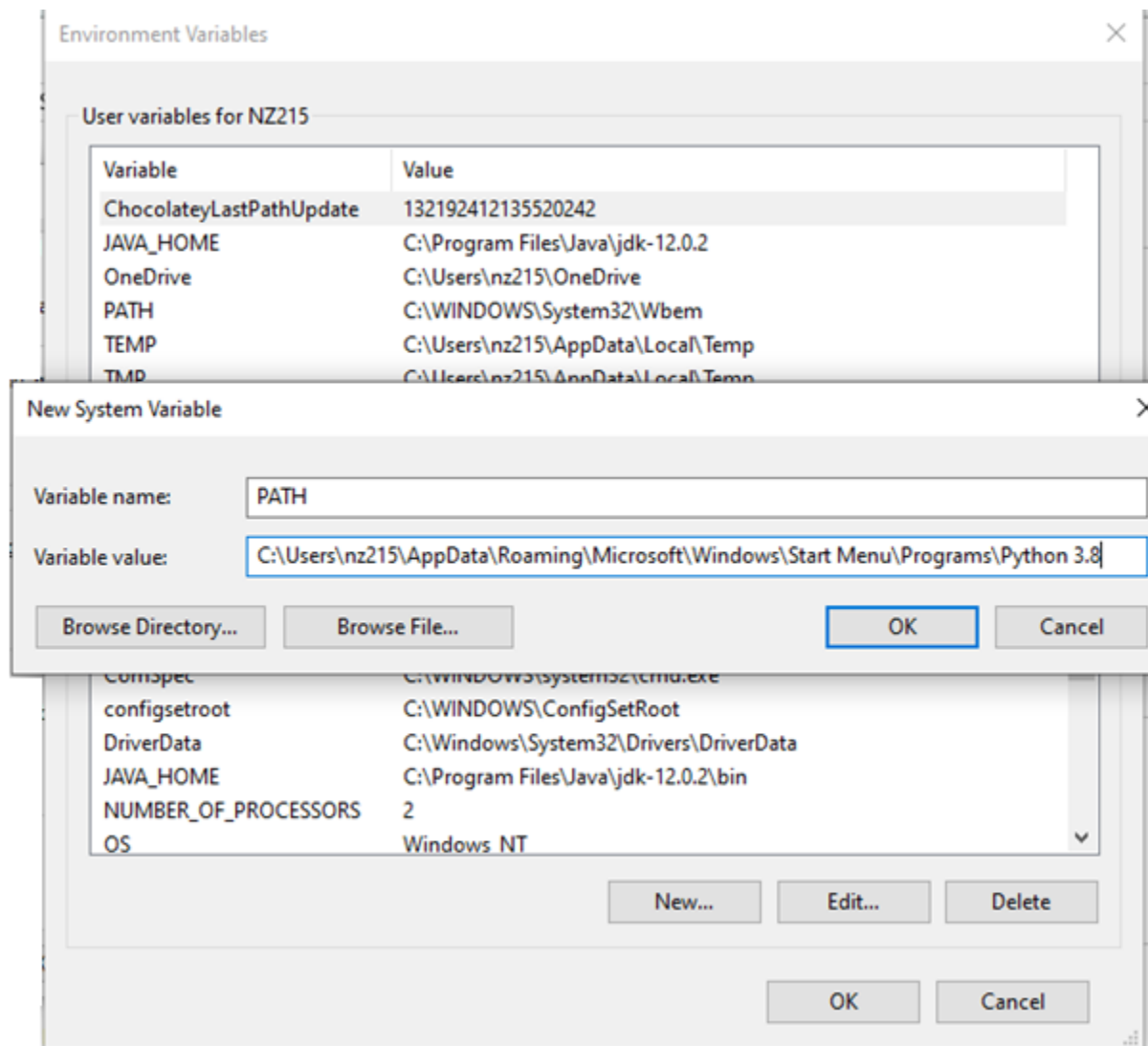
## Step 7:

Add Environment variables to run Python in Windows.

In the Start menu, search for “advanced system settings”. Select “View advanced system settings”. In the “System Properties” window, click on the “Advanced” tab and then click on the “Environment Variables” button.

Locate the Python installation directory on your system. If you followed the steps exactly as above, Python will be installed in below locations:

- C:\Program Files (x86)\Python37-32: for 32-bit installation
- C:\Program Files\Python37-32: for 64-bit installation.

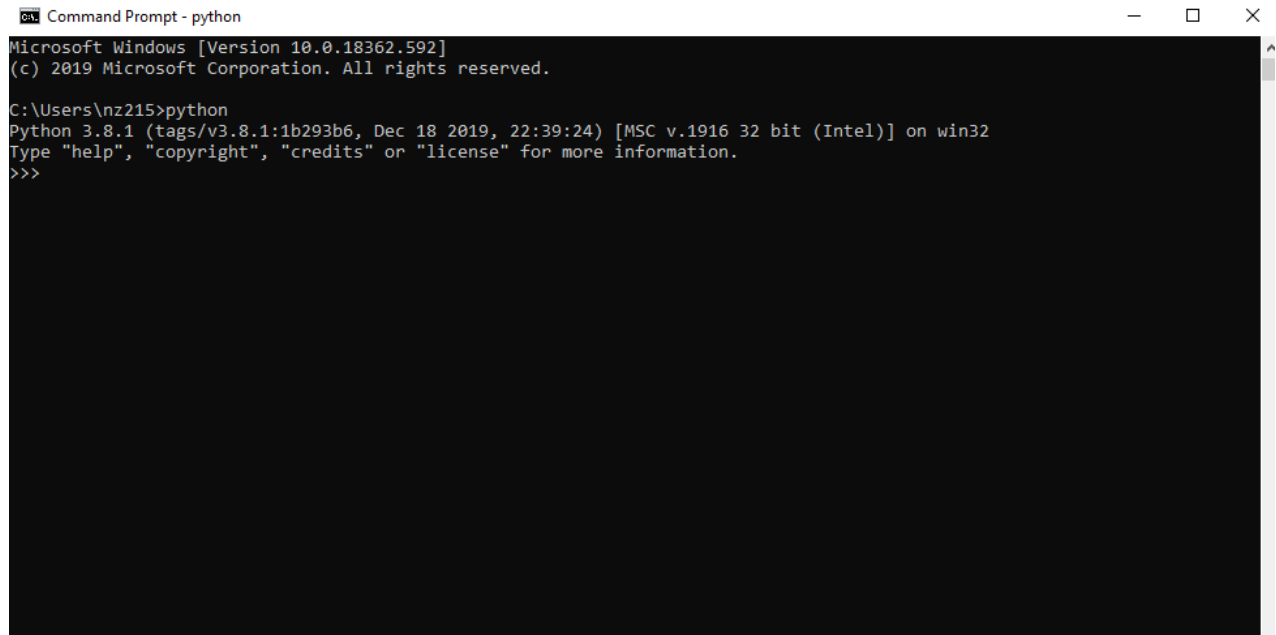




## Step 8: Verify Python Installation

You have now successfully installed Python 3.8.1 on Windows. You can verify if the Python installation is successful through the command line.

Search for the command prompt and type “python”. You can see that Python 3.8.1 is successfully installed.

A screenshot of a Windows Command Prompt window titled "Command Prompt - python". The window shows the following text: "Microsoft Windows [Version 10.0.18362.592] (c) 2019 Microsoft Corporation. All rights reserved. C:\Users\nz215>python Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>>". The text is displayed in a black background with white text, and the window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
Command Prompt - python
Microsoft Windows [Version 10.0.18362.592]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\nz215>python
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

## Basic syntax to write a Python program

Once you have installed Python software successful into your system, you can start writing Python program in the command prompt.

Python has many similarities and differences with other programming languages such as **Perl**, **C**, and **Java**, etc. Let us discuss the basic syntax to write a Python programming language.

Python provides two ways to run a program, and they are:

## Two Ways To Write a Python Program

Using Interactive Interpreter Prompt

Using a Script File

simpliv  
Learn. Teach. Earn

- Using Interactive interpreter prompt
- Using a script file.

We will discuss both of these separately in detail. Here in the below discussion, we will write a program to get the output as **Hello world**

- **Using interactive interpreter prompt:**

Using interactive prompt, users can execute the Python statement one by one. This mode of execution is preferred where the users are concerned about the output of each line of the Python program.

In Python3 `print()` function is used to print the message in the console. It is one of the standard functions used to print the output to the console.

Invoking the interpreter without passing a script file as a parameter brings up the following prompt –

```
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18, 2019, 22:39:24) [MSC v.1916 32 bit (Intel)]  
on win32
```

```
Type "help", "copyright", "credits", or "license" for more information.
```

```
>>>
```

Type the following text at the Python prompt and press Enter –

```
print("Hello world")
```

Now you will get the output as:

```
Hello world
```

In this program, we are using built-in `print()` function to print the string `Hello world` as the output. String is a sequence of characters and in Python they are enclosed inside **single quotes (")**, **double quotes(" ")** or in triple quotes(`""" """`).

If you want to print a list of five colors, you can write the program in the Python prompt as follows:

```
print ("Red")           # prints Red as output
print ("Blue")          # prints Blue as the output
print ("Yellow")        # prints Yellow as the output
print ("Orange")        # prints Orange as the output
print ("Saffron")       # prints Saffron as the output
```

Now you will get output as:

```
Red
Blue
Yellow
Orange
Saffron
```

- **Using a script file:**

The program can be written in a script. Python files have extension `.py`. You can write the code in a file and that can be executed later. Users can write the code using the editor and then save it with `.py` extension. For example, you can save the file with the extension `myfirstprogram.py`

To print **Hello world** we can write the code `Print ("Hello world");`

In order to run this `myfirstprogram.py` file, we need to run the command `$ python3 first.py` on the terminal.

Now the output message displays as **Hello world** on the console.

## Python Constructs

Now let us see some useful Python Constructs that provide a better idea of structure of Python code.



- **Functions**
- **Classes**
- **Modules**
- **Packages**
- **Lists.**

### 1. Functions:

In Python programming language a **function** is referred to as a collection of statements grouped under a name. You can use it whenever you there is a need to execute all those statements at a time. You can call the function as many times as you want it in your program. Function may return a value.

## 2. Classes:

Python is an Objected Oriented programming language. This language supports classes and objects. A class is an abstract data type. An object is a real world entity and an instance of a class.

## 3. Modules:

A collection of related classes and functions can be called as a module. These are some of the different types of modules:

- mathematical calculations
- string calculations
- web programming, etc.

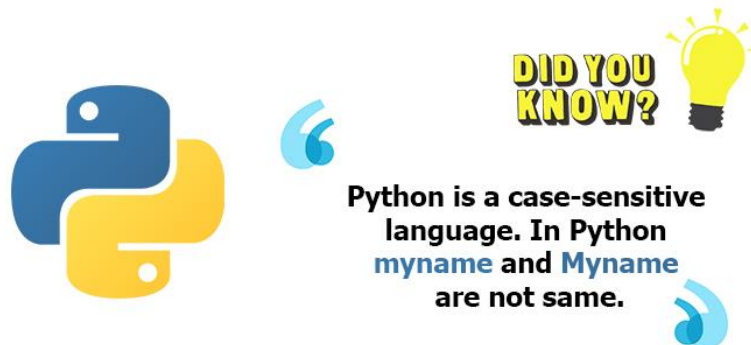
**4. Packages:** Python package is a collection of related modules. Developers can either import a package or can develop their own.

**5. Lists:** Lists can be a collection of values. They are declared in the CSV (Comma Separated Values) format and enclosed in square brackets.

## Python identifiers:

Python identifier is a name that is used to identify a variable, function, class, module or other object. It helps to differentiate one entity from another.

**Some of the rules for writing identifiers are as follows:**



***Did You Know?: "Python is a case-sensitive language. In Python myname and Myname are not the same."***

01

The first character of the variable should be an alphabet or underscore ( \_ )

02

Identifiers can be a combination of letters.

03

Identifiers must not contain any white space or special characters ( ! , @ , # , % , ^ , & , \* )

04

Keywords cannot be used as identifiers

05

Identifiers can be of any length.

**simpliv**  
Learn. Teach. Earn

- The first character of the variable should be an alphabet or underscore ( \_ )
- Identifiers can be a combination of letters. They can be written in **lower case (a to z)** or **upper case (A to Z)** or in **digits (0 to 9)** or an **underscore( \_ )**. For example: Names like **myFirstprogram**, **var\_1**, **display\_this\_on\_screen** etc.
- Identifiers must not contain any white space or special characters ( ! , @ , # , % , ^ , & , \* )
- Keywords cannot be used as identifiers
- Identifiers can be of any length.

## Python Reserved words (Python keywords)

Python keywords are the reserved words. These keywords are used to convey a special meaning to the compiler/interpreter. Keywords cannot be used as a variable name, function name or any other identifier.



**Did You Know?:** "Python 3.7 has 33 keywords and the number can vary slightly in coming days."

All the keywords in Python are lowercase except **True**, **False** and **None**.

The following table gives the list of keywords used in Python.

True	False	None	and	As
assert	break	Continue	for	Lambda
try	from	Nonlocal	while	Del
global	not	With	elif	If
or	yield	Assert	else	Import
pass	break	Except	in	Raise
finally	is	Return		

The above table provides a list of keywords. However, the list may get vary according to different Python versions. You can type the following in the prompt to get the list of all the available keywords in the current version of Python being used.

```
import keyword
print (keyword.kwlist)
```

Now you will get the output of all the available keywords in your Python version.

## Indentation

Some of the programming languages such as Java, C, etc. use braces `{ }` to define the block of code. But Python uses indentation.

The number of spaces is variable, but it should be noted that all the statements within the block must be indented the same amount. The usage of indentation makes the Python code look neat and clean.

Generally, four whitespaces are used for indentation and that is preferred over using tabs. The following code is an example of indentation:

**Program:**

```
if True;
    print "True";
else;
    print "False"
```

Incorrect indentation will result in **IndentationError**

### Quotation in Python

To denote string literals **Python** accepts **single (')**, **double (")**, **triple (""')** quotes as long as the same type of quote starts and ends the string.

The below example shows how to use quotations in Python.

single_quotation	'word'
double_quotation	"Python is a powerful programming language"
triple_quotation	"""Python is a powerful programming language. It is open-source, and object oriented programming language"""

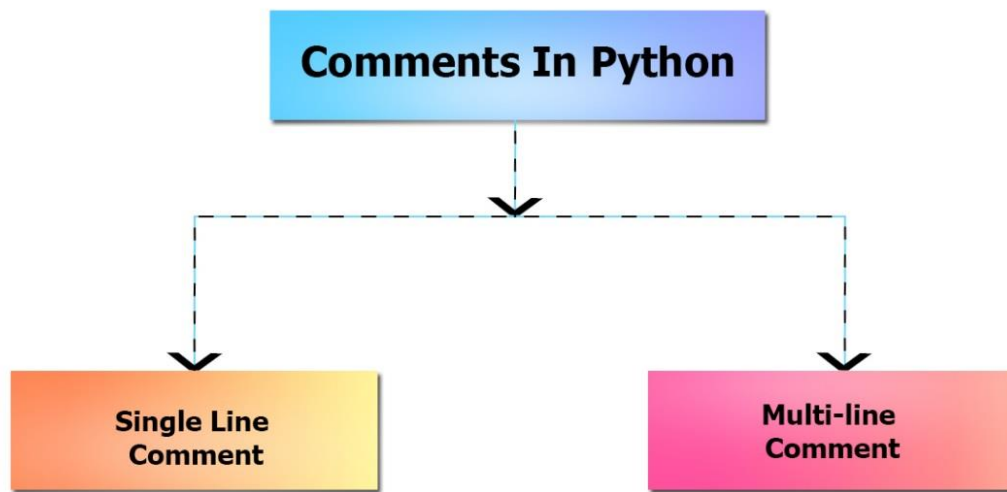
### Comments in Python

Comments play a very important role while writing a program. It describes the program to the user so that they can easily understand the program.

In Python **hash (#)** is used to start writing a comment. The interpreter does not interpret the comment.

Python supports two types of comments. They are:





1. Single Line Comment.
2. Multi-Line Comment.

### 1. Single Line Comment:

Single Line Comment starts with #?

The following program has single line comment.

```
# This program has single line comment  
print ("Hello world");
```

Once the above program is executed, we get the following output:

```
Hello world
```

Here you can note that only the output Hello world is displayed on the console as the output and the above comment written with # tag is not displayed.

### 2. Multi-Line Comment:

Developers can write multiple lines of comments to their code. It can be done in two ways. They are:

- By using **# tag** before start of each line of comment
- By using **triple comment** at the start and at the end of the comment.

We will see a program example for these types below.

## Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- HTML (Free /Available to everyone)
- PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)
- Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below

