



Day 1

Python Basic

What is Python?

- Python is a high-level interpreted language that employs an object-oriented approach.
- Python is more preferred because of its simplicity, powerful libraries, and readability.
- Python is portable and an open source.

Application of Python

1. **Web Development**(Django, Pyramid, Flask, and Bottle for developing web frameworks)
2. **Game Development**(PySoy, PyGame)
3. **Scientific and Numeric Applications**(SciPy, Pandas, IPython, Numeric Python, NLP)
4. **Artificial Intelligence and Machine Learning**(SciPy, Pandas, Seaborn, Keras, TensorFlow, and Scikit-learn for ML)
5. **Software Development**

6. Enterprise-level/Business Applications
7. Education programs and training courses
8. Operating Systems
9. Web Scraping Applications

Installation Of Python

1. Click on the link below

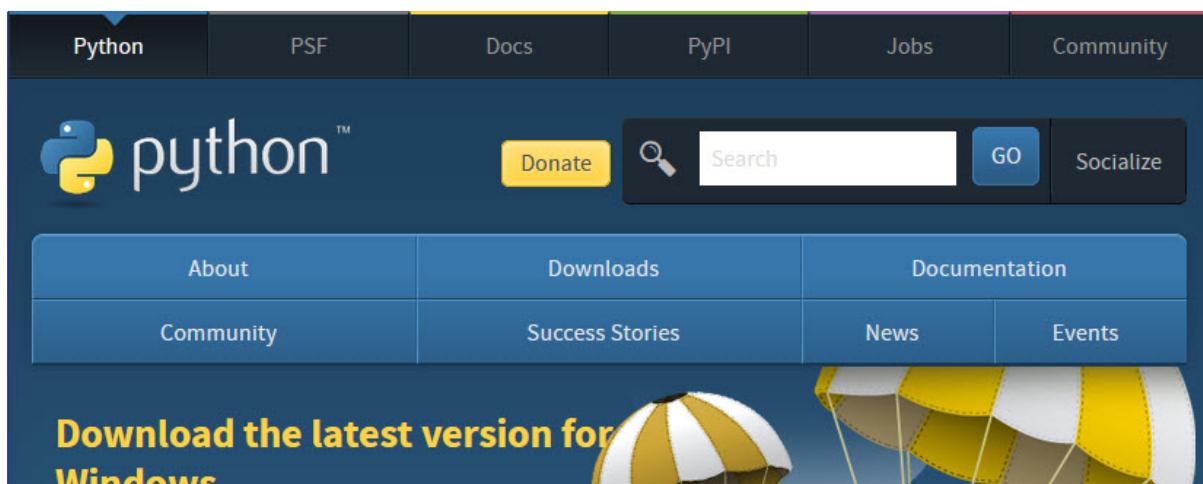
Download Python

Information about specific ports, and developer info Source and binary executables are signed by the release manager or binary

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



2. A window like below will appear on the screen



3. *Click on Downloads>>> Windows*

The following screen will appear. Click on the latest release.

Python Releases for Windows

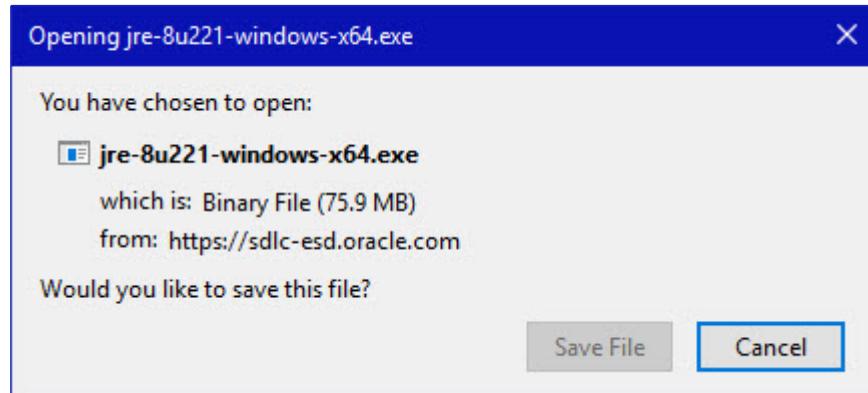
- [Latest Python 3 Release - Python 3.9.5](#)
- [Latest Python 2 Release - Python 2.7.18](#)

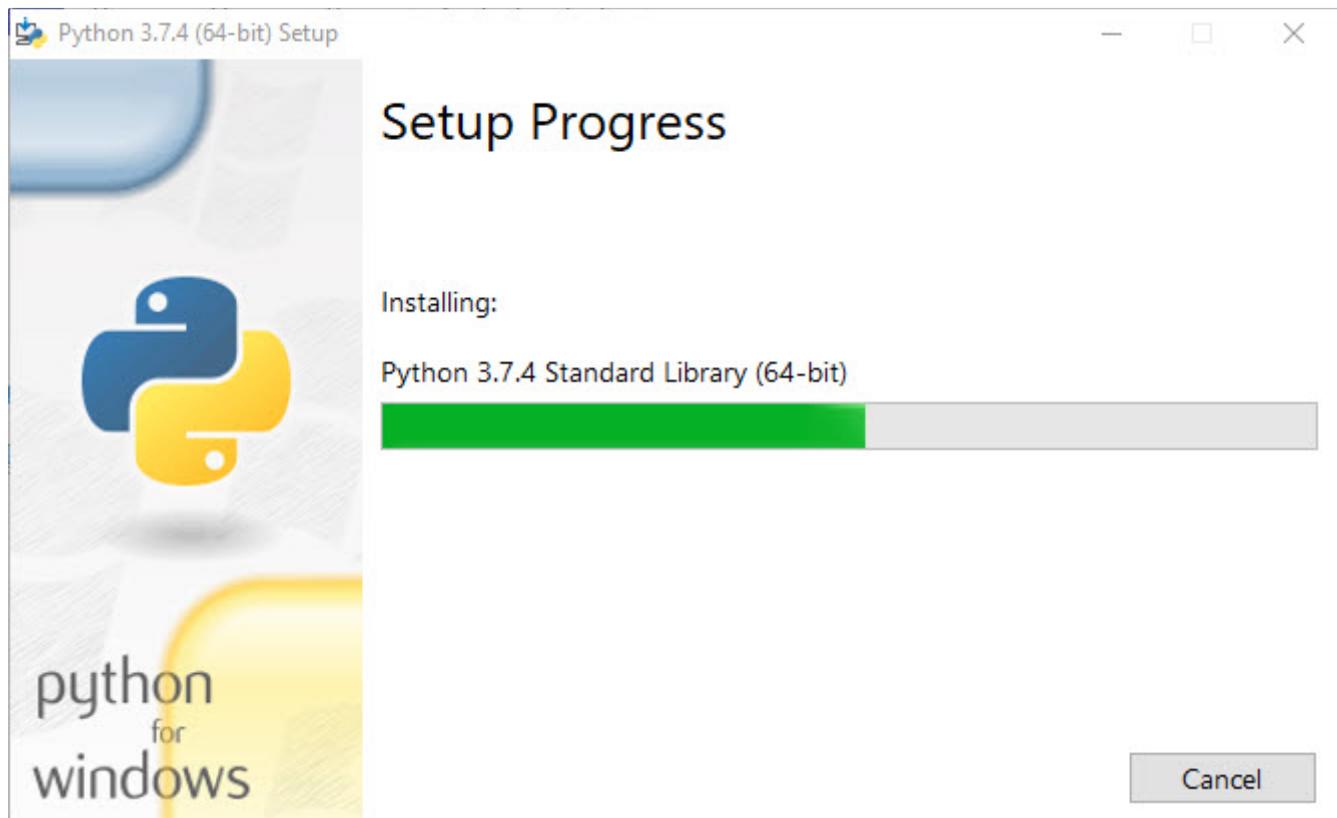
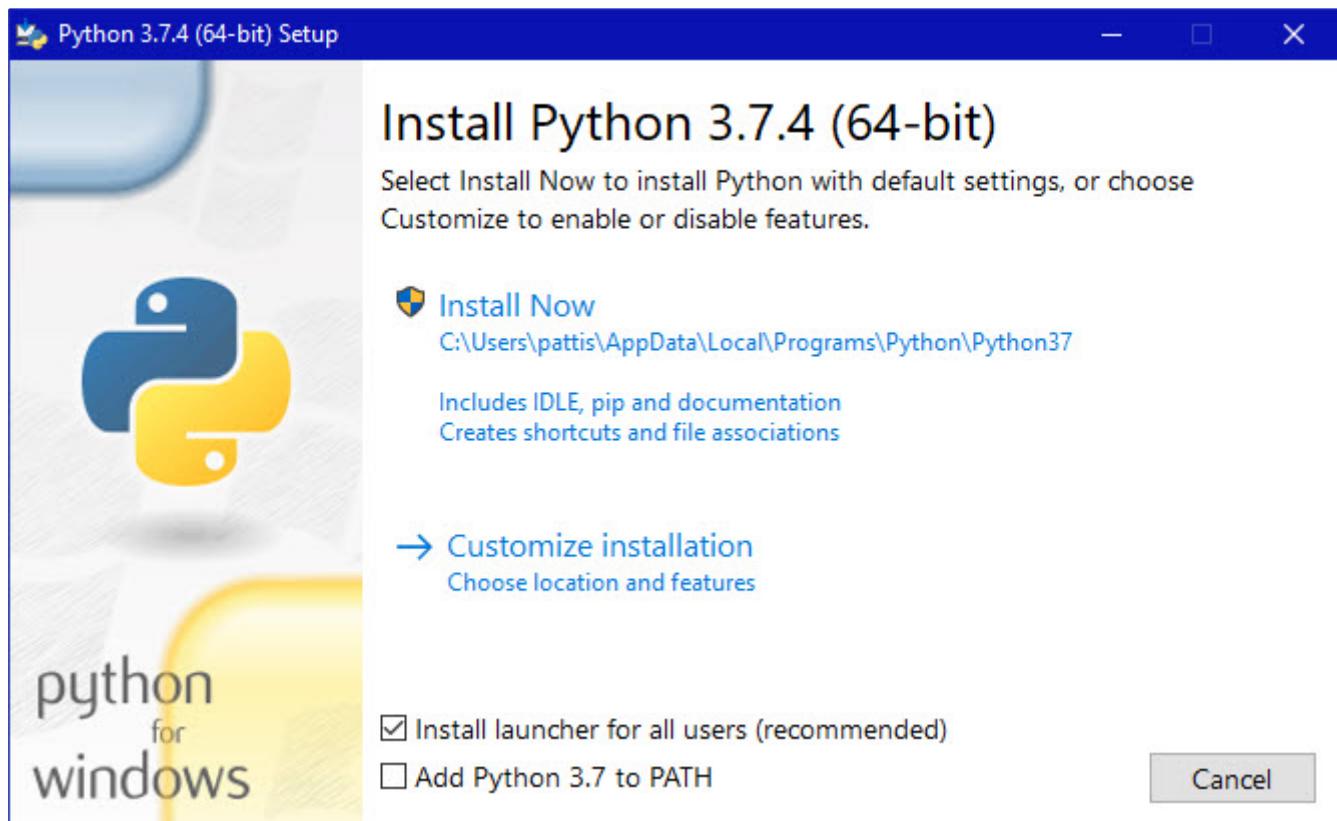
Stable Releases

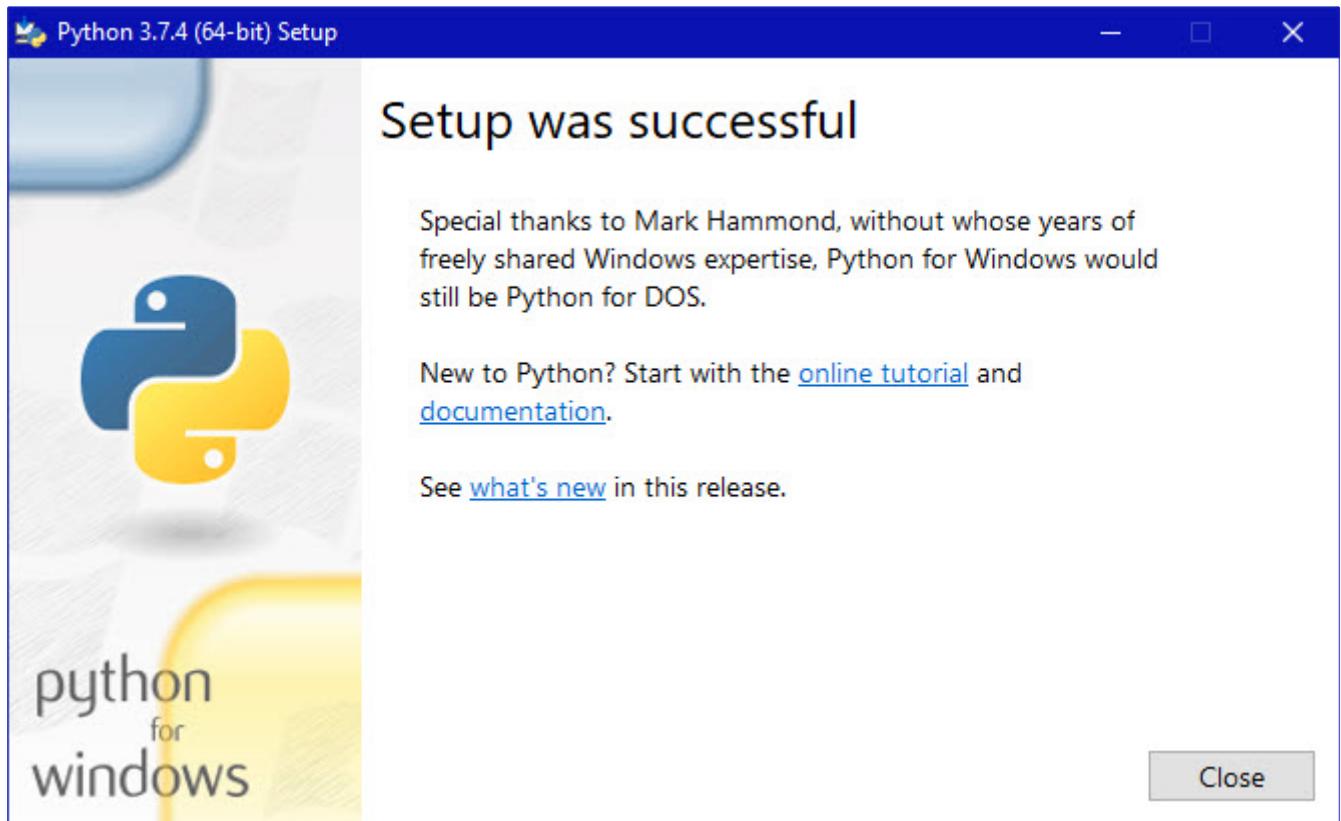
- Python 3.9.5 - May 3, 2021
 - Note that Python 3.9.5 cannot be used on Windows 7 or earlier.
 - Download [Windows embeddable package \(32-bit\)](#)
 - Download [Windows embeddable package \(64-bit\)](#)
 - Download Windows help file
 - Download Windows installer (32-bit)
 - Download Windows installer (64-bit)
- Python 3.8.10 - May 3, 2021
 - Note that Python 3.8.10 cannot be used on Windows XP or earlier.
 - Download [Windows embeddable package \(32-bit\)](#)

Pre-releases

- Python 3.10.0b3 - June 17, 2021
 - Download [Windows embeddable package \(32-bit\)](#)
 - Download [Windows embeddable package \(64-bit\)](#)
 - Download Windows help file
 - Download Windows installer (32-bit)
 - Download Windows installer (64-bit)
- Python 3.10.0b2 - May 31, 2021
 - Download [Windows embeddable package \(32-bit\)](#)
 - Download [Windows embeddable package \(64-bit\)](#)
 - Download Windows help file
 - Download Windows installer (32-bit)







5.To verify the installation Navigate to the directory then Double-click the icon/file python.exe

You will see this window where you will see the Python version that has been installed .

```
Select C:\Users\pattis\AppData\Local\Programs\Python\Python37\python.exe
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

A screenshot of a terminal window titled "Select C:\Users\pattis\AppData\Local\Programs\Python\Python37\python.exe". The window displays the Python 3.7.4 command line interface, showing the path, version, build date, and copyright information. It ends with a prompt ">>>".

Python IDE 🎨

- PyCharm
- Spyder
- Sublime
- Jupyter
- Atom
- Python IDLE
- VSCode
- Wing

- Thonny
- PyDev

Keywords

Keywords are special names that are already present in python.

Some of the examples are-

You can see the list of keywords by importing keyword package

```
import keyword
keyword.kwList
keyword.iskeyword('switch')
```

Comments

To make more convenient for others to read your code, and for describing your code

you can use comments in your code

```
#I am a single line comment #I am a multi #line comment '''this is a
docstring comment'''
```

Variables & Identifiers

Variables within Python follow these basic rules and difference from other languages :

- You do not need to pre-declare variables before they are used
- Python sets the object type based on the value it is assigned
- Variables are given simple alphanumeric names

Identifiers are user-defined names that we use to represent variables, classes, functions, modules etc.

Syntax:

```
name='Floxus' #variable name my_identifier=name #identifier
```

Variables are used to store some values with reserved memory location.

```
name='Floxus' #assigning variable
```

Blocks

Blocks start after a colon on the preceding line.

```
if(1): print('I am a block')
```

The block continues until a new block is created , or until the indentation for the block ends.

When indenting, use either tabs or spaces.