

# 2

## Getting Started

Let Us  
**Python**



*“On your mark, set, go...”*



### Contents

- Python Specification
- Python Installation under Windows
- Python Installation under Linux
- Python Resources
- Third-party Packages
- More Sophisticated Tools
- Working with Python
- Python Programming Modes
- Determining Python Version
- Exercises



## Python Specification

- Python is a specification for a language that can be implemented in different ways. There are many implementations of this specification written in different languages.
- Different popular Python implementations are:
  - CPython - is the reference implementation, written in C.
  - PyPy - Written in a subset of Python language called RPython.
  - Jython - Written in Java.
  - IronPython - Written in C#.
- All the implementations are compilers as well as interpreters. The compiler converts the Python program into intermediate bytecode. This bytecode is then interpreted by the interpreter.

## Python Installation under Windows

- Python has evolved over the years. At the time of writing of this edition the latest version for Windows and Linux environments was Python 3.8.2.
- Python is not shipped as part of Windows OS. So we need to install it separately. For this we need to download the Python installer from [www.python.org/downloads/](http://www.python.org/downloads/). (<http://www.python.org/downloads/>.)
- While downloading ensure that you choose the appropriate installer from the following, based on whether you wish to install it on a 32-bit machine or a 64-bit machine:
  - 64-bit machine: Download Windows x86-64 executable installer
  - 32-bit machine: Download Windows x86 executable installer
- Once you have chosen and downloaded an installer, execute it by double-clicking on the downloaded file. A dialog shown in Figure 2.1 will appear on the screen.
- In this dialog check the check box 'Add Python 3.8 to PATH' to ensure that the interpreter will be placed in your execution path.



Figure 2.1

- Click on 'Install Now' and the installation will happen in a few minutes. Python files will get installed in the directory:

C:\Users\Kanetkar\AppData\Local\Programs\Python\Python38-32

- In this path 'Kanetkar' will be substituted by your user name and 'Python38-32' by version number of Python installation that you have downloaded and installed.
- If you forget to check the check box, you can add the path mentioned above to PATH variable through Control Panel | System | Environment Variables | Edit. The PATH variable already contains many semicolon separated values. Append the above path to existing values.

## Python Installation under Linux

- Most Linux distributions already contain Python in them. However, the installed Python version may not be the latest one. You can check the version as shown below:

```
$ python3 --version
```

- If you find that the version is not the latest one, then you can install it using the command:

```
$ sudo apt-get install python3.8
```

## Python Resources

- Python source code, binaries and documentation is available at:
  - Python official website: [www.python.org](http://www.python.org) (<http://www.python.org>)
  - Documentation website: [www.python.org/doc](http://www.python.org/doc) (<http://www.python.org/doc>)
- Program development in Python can be done in 3 ways:
  - Using built-in IDLE.
  - Using third-party IDEs.
  - Using online Python shells.
- Third-party development tools and the links from where they can be downloaded are given below:
  - NetBeans IDE for Python:  
<https://download.netbeans.org/netbeans/6.5/python/ea/> (<https://download.netbeans.org/netbeans/6.5/python/ea/>)
  - PyCharm IDE for Python:  
<https://www.jetbrains.com/pycharm> (<https://www.jetbrains.com/pycharm>)
  - Visual Studio Code IDE:  
<https://code.visualstudio.com/download> (<https://code.visualstudio.com/download>)
- If you do not wish to install any Python development tool on your machine, then you can use any of the following online Python shells:
  - <https://www.python.org/shell/> (<https://www.python.org/shell/>)
  - <https://ideone.com/> (<https://ideone.com/>)
  - <https://repl.it/languages/python3> (<https://repl.it/languages/python3>)

## Third-party Packages

- Pythonistas in Python community create packages (libraries) and makes it available for use for other programmers. They use PyPI—Python Package Index ([www.pypi.org](http://www.pypi.org)) (<http://www.pypi.org>) to host their packages. PyPI maintains the list of such third-party Python packages available.
- There are third-party packages available for literally doing everything under the sun. Some packages that are popularly used for creating Data Science applications include:
  - NumPy: Advanced mathematical operations library with support for large multi-dimensional arrays and matrices.
  - SciPy: Scientific computing library for optimization, integration, interpolation, signal processing, image processing, etc.
  - Pandas: Library for manipulating numerical tables and time series.

- Matplotlib: 2D and 3D Data visualization library.
- OpenCV: Open source Computer vision library.
- You too can register at PyPI and upload your packages there. You should follow the guidelines given at [www.pypi.org](http://www.pypi.org) (<http://www.pypi.org>) to create the package, build it and upload it to the Python Package Index.
- pip is a commonly used tool for installing packages from PyPI. This tool gets installed when you install Python.

## More Sophisticated Tools

- Many tools have come into existence to help Python programmers build and document their Data Science and Artificial Intelligence applications. These include:
  - Jupyter Notebook - It is a very flexible browser-based tool that lets us to interactively work with Python (and many other languages). It lets us put our Python code, output of the code and any kind of visualization or plot etc. in the same document called Notebook. It is a great tool doing modular program development.
  - Google Colab - This tool provides a free Jupyter notebook environment to execute code on Google's cloud servers. As a result, you can leverage the power of Google's hardware.
  - Spyder - This tool provides a **Scientific PYthon Development EnviRonment** with sophisticated testing and debugging features.
- Both Jupyter and Spyder are part of a very popular software distribution called Anaconda. So once you download and install Anaconda, you get Jupyter and Spyder ready-made.

## Working with Python

- Once Python is installed, program development can be done using the built-in Python Integrated Development and Learning Environment (IDLE).
- IDLE is a good development tool. It offers handy features like syntax highlighting, context-sensitive help and debugging.
- Syntax highlighting feature display keywords, functions, methods and strings in different colors making it easy to identify them.

- Context-sensitive help can be obtained by pressing Ctrl Space wherever you need help as you type the program. This is immensely useful since it is almost impossible to remember names of all functions and methods and their parameters.
- Debugger lets you locate any logical errors that you may have committed in your program by allowing you trace the flow of execution of the program. This tracing can be done a step at a time by setting up break points and by single stepping through the program. As you do so IDLE lets you watch the values of different variables as they change during execution.

## Python Programming Modes

- Python can be used in two modes:
  - Interactive mode - used for exploring Python syntax, seek help and debug short programs.
  - Script mode - used for writing full-fledged Python programs.
- Both modes are supported by IDLE (Python Integrated Development and Learning Environment).
- To use IDLE in Interactive mode:
  - Locate it in Windows by typing IDLE in Windows search bar and hit enter, or double click the IDLE icon.
  - It will open the Python shell window showing >>> Python shell prompt.
  - Execute the following Python code at this prompt.

```
>>> print('Keep calm and bubble on')
```

- It will display the message 'Keep calm and bubble on' followed by the >>> prompt.
- To use IDLE in Script mode:
  - Launch IDLE. In the IDLE shell window from the menu select File | New File. A new window will open. Type the following script in it:

```
print('Those who can't laugh at themselves...')  
print('leave the job to others.')
```

- Using File | Save and save the script under the name 'Test.py'.
- Execute the script from the Run menu or using F5. The two messages will get printed.
- Instead of IDLE if you decide to use NetBeans or Visual Studio Code for program development then follow the steps given below:
  - Create a new Python project 'Test'.
  - Type the script in Test.py.
  - Execute the script using F6 in NetBeans or Ctrl F5 in Visual Studio Code.
  - On execution it will print the two lines and then you are ready to create another project and another script in it.

## Determining Python Version

- Python has evolved over the years. You can determine the version installed on your machine through a simple Python script:

```
import sys
print(sys.version)
```

## Exercises

**[A]** Answer the following questions:

- (a) What do the prompts C:\>, \$ and >>> signify?
- (b) In which two modes can IDLE be used?
- (c) What is the purpose of the two programming modes offered by IDLE?
- (d) How can third party libraries be used in a Python program?

**[B]** Match the following pairs:

- |            |                                     |
|------------|-------------------------------------|
| a. pip     | 1. Advanced mathematical operations |
| b. Jupyter | 2. Scientific computing             |
| c. Spyder  | 3. Manipulate numerical tables      |
| d. PyPI    | 4. Visualization                    |

- |               |                                    |
|---------------|------------------------------------|
| e. NumPy      | 5. Computer vision                 |
| f. SciPy      | 6. Package installation tool       |
| g. Pandas     | 7. Build and document applications |
| h. Matplotlib | 8. Scientific library              |
| i. OpenCV     | 9. Python package index            |

**[C]** State whether the following statements are True or False:

- (a) Python is a specification that can be implemented through languages like Python, C#, Java, etc.
- (b) CPython is implementation of Python specification, written in C.
- (c) Python program is first compiled into byte code, which is then interpreted.
- (d) Most Linux distributions already contain Python.
- (e) Windows system doesn't contain Python and it needs to be separately installed.
- (f) Python programs can be built using IDLE, NetBeans, PyCharm and Visual Studio Code.
- (g) Third-party Python packages are distributed using PyPI.