

## 6

# Getting Started with Python

## In This Chapter

- 6.1 Introduction
- 6.2 Python – Pluses
- 6.3 Python – Some Minuses
- 6.4 Working in Python
- 6.5 Understanding First Program/Script

### 6.1 INTRODUCTION

The word *Python* – isn't it scary ? Does it bring the image of big reptile that we prefer to see either in jungles or zoo ? Well, it's time to change the image. Now on, you'll remember word *Python* for its playfulness and pleasant productivity. Confused ? Well, Don't be – because, now on you'll get introduced to a new programming language namely 'Python', which promises to make you a big programming fan :-).

Python programming language was developed by **Guido Van Rossum** in February 1991. Python is based on or influenced with *two* programming languages :

- ❖ *ABC language*, a teaching language created as a replacement of BASIC, and
- ❖ *Modula-3*

Python is an easy-to-learn yet powerful object oriented programming language. It is a very high level programming language yet as powerful as many other middle-level not so high-level languages like C, C++, Java etc.

In this chapter, we shall introduce you to playful world of *Piquant Python* [Word 'Piquant' means pleasantly stimulating or exciting to the mind]. So, are we ready ? And... here we go.

### 6.2 PYTHON – PLUSES

Though Python language came into being in early 1990's, yet it is competing with ever-popular languages such as C, C++, Java etc. in popularity index.

#### NOTE

Do you know Python, the programming language, was named after famous BBC comedy show namely *Monty Python's Flying Circus*.

Let's see what are these pluses of Python:

## 1. Easy to Use

Python is compact and very easy to use object oriented language with very simple syntax rules. It is a very high level language and thus very very programmer friendly.

## 2. Expressive Language

Python is an expressive language - fewer lines of code and simpler syntax. For example, consider following two sets of codes:

// In C++ : Swap Values	# In Python : Swap values
int a = 2, b = 3, tmp ;	a, b = 2, 3
tmp = a ;	a, b = b, a
a = b ;	
b = tmp ;	

which one is compact and easier to understand? Need I say more? :)

## 3. Interpreted Language

Python is an interpreted language, not a compiled language. It makes Python an easy-to-debug language and thus suitable for beginners to advanced users.

## 4. Its Completeness

For most types of required functionality is available through various modules of Python standard library<sup>1</sup>. For example, for diverse

functionalities such as socket, networking, databases, GUI development, scientific computing and many more, everything is available in Python standard library. That's why often called - Python forums "Swiss Army knife" of programming.

## 5. Cross-platform language

Python can run equally well on variety of platforms - Windows, Linux/Ubuntu, Macintosh, microcomputers, small phones etc. Isn't that amazing? And that makes Python a true cross-platform language. Or in other words, Python is a portable language.

## 6. Free and Open Source

Python language is freely available along with its source code.

## 7. Variety of Usage/Applications

Python has evolved into a powerful, complex and useful language over these years. These days Python is being used in many diverse fields/applications, some of which are:

- ❖ Scripting
- ❖ Rapid Prototyping
- ❖ Web Applications
- ❖ GUI Programs
- ❖ Game development
- ❖ Database Applications
- ❖ System Administrations

## 6.3 PYTHON – SOME MINUSES (SO HUMAN LIKE)

Although Python is very powerful yet simple language with so many advantages, it is not the perfect programming language. Let's see what these are:

### 1. Not the Fastest Language

Python is an interpreted language not a fully compiled one. Fully compiled languages are faster than their interpreted counterparts. So, Python offers faster development times but execution-times are not that fast compared to some compiled languages.

### 2. Lesser Libraries than C, Java, Perl

Python offers library support for almost all computing programs, but its library is still not competent with languages like C, Java, Perl as they have larger collections available. Sometimes in some cases, these languages offer better and multiple solutions than Python.

1. If you install Python through Anaconda Python Distribution it loads most libraries and packages with Python.
2. Python even has versions that run on different languages such as Java (Jython), .NET (IronPython) etc.

Not Strong on Type-binding

Python interpreter is not very strong on catching 'Type-mismatch' issues. For example, if you declare a variable as integer but later store a string value in it, Python won't complain or pin-point it.

So, now you are familiar with what all Python offers. As a free and open-source language, its users are growing by leaps and bounds.

As per Jan 2020 popularity index, Python was 2nd most popular programming language<sup>3</sup> after *JavaScript*. That is the reason, it's part of your syllabus. Together we'll make it playful Python :).

## 6.4 WORKING IN PYTHON

Before you start working in Python, you need to install Python on your computers. There are multiple Python distributions available today.

- ❖ Default installation available from [www.python.org](http://www.python.org) is called **CPython installation** and comes with *Python interpreter*, *Python IDLE (Python GUI)* and *Pip (package installer)*.
  - ❖ There are many other Python distributions available these days. **Anaconda Python distribution** is one such highly recommended distribution that comes preloaded with many packages and libraries (e.g., NumPy, SciPy, Panda libraries etc.).
  - ❖ Many popular IDEs are also available e.g., Spyder IDE, PyCharm IDE etc. Of these, Spyder IDE is already available as a part of Anaconda Python distribution.

To install any of these distributions, PLEASE REFER TO APPENDIX A. We shall learn to work with both these distribution types.

Once you have Python installed on your computers, you are ready to work on it. You can work in Python in following different ways :

(i) in Interactive mode (also called Immediate Mode)      (ii) in Script mode

#### 6.4.1 Working in Default CPython Distribution

The default distribution, CPython, comes with **Python interpreter**, **Python IDLE** (GUI based) and **pip** (package installer). To work in *interactive* as well as *script* mode, you need to open **Python IDLE**.

#### 6.4.1A Working in Interactive Mode (Python IDLE)

Interactive mode of working means you type the command – one command at a time, and the Python executes the given command there and then and gives you output. In interactive mode, you type the command in front of Python command prompt `>>>`. For example, if you type `2 + 5` in front of Python prompt, it will give you result as 7 :

*Result returned  
by Python*

**>>> 2 + 5** ← command/expression given here

7

# Path Wala

142

To work in interactive mode, follow the process given below:

(i) Click Start button → All Programs → Python 3.6.2 → IDLE (Python 3.6.2)

Or

Click Start button → all Programs → Python 3.6.2 → Python (command line)

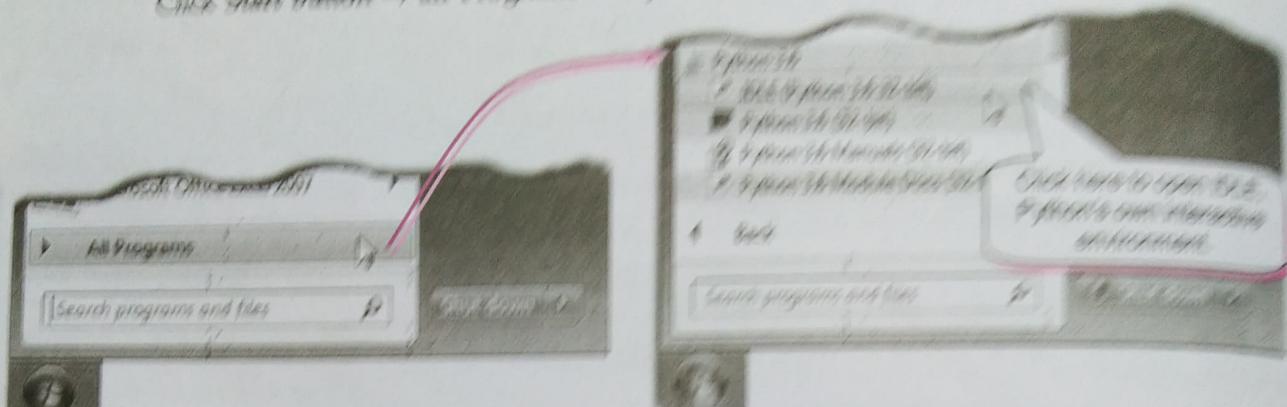


Figure 6.1 (a) Starting Python Shell.

- (ii) It will open Python Shell [see Fig. 6.1(b)] where you'll see the Python prompt (three '>' signs i.e., `>>>`).
- (iii) Type commands in front of this Python prompt and Python will immediately give you the result. [see Fig. 6.1(c)]

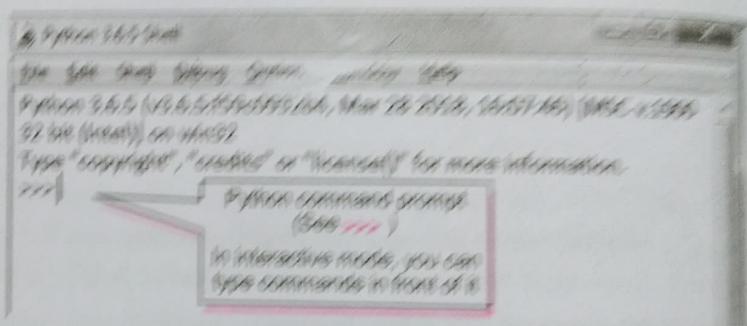


Figure 6.1 (b) Python's interactive interpreter - Python Shell.

## NOTE

The interactive *interpreter* of Python is also called **Python Shell**.

This way of giving name or expression in front of `>>>` is called displaying.

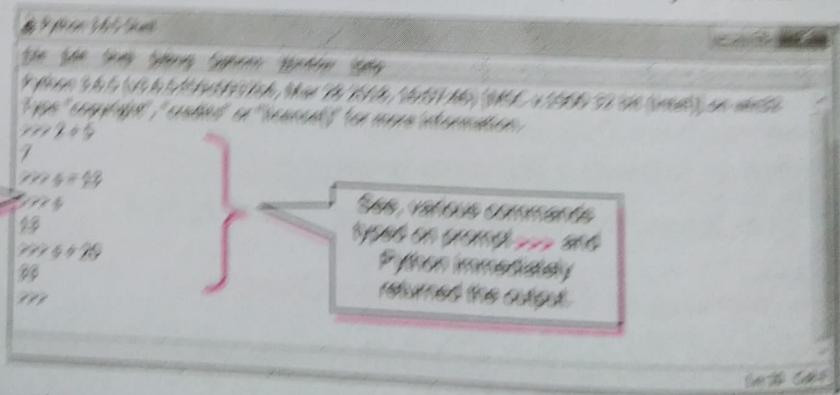


Figure 6.1 (c) Interactive commands and their output in Python Shell.

For example, to print string "Hello" on the screen, you need to type the following in front of Python prompt (`>>>`)

```
>>> print("Hello")
```

And Python interpreter will immediately display string Hello below the command. To display, you just need to mention name or expression [Fig. 6.1(c)] in front of the prompt.

Figure 6.1(c) shows you some sample commands that we typed in Python shell and the output returned by Python interpreter.

## NOTE

Interactive mode proves very useful for testing code; you type the commands one by one and get the result or error one by one.

### 6.4.1B Working in Script Mode (Python IDLE)

What if you want to save all the commands in the form of program file and want to see all output lines together rather than sandwiched between successive commands? With interactive mode, you cannot do so, for :

- ⇒ Interactive mode does not save the commands entered by you in the form of a program<sup>4</sup>.
- ⇒ The output is sandwiched between the command lines [see Fig. 6.1(c)].

The solution to above problems is the **Script mode**. To work in a script mode, you need to do the following :

#### Step 1 : Create Module / Script / Program File

Firstly, you have to create and save a module / Script / Program file. To do so, follow these instructions :

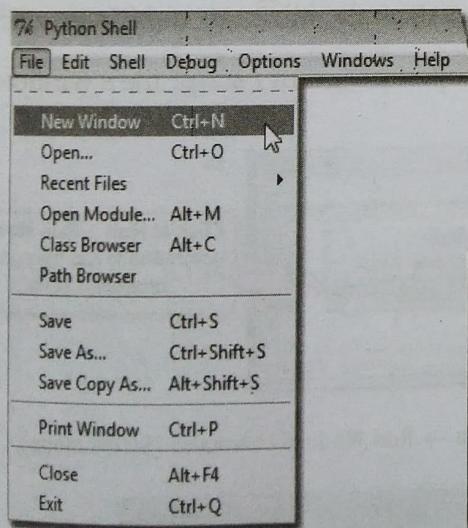
- (i) Click **Start button** → **All Programs** → **Python 3.6.x** → **IDLE**. [Fig. 6.2(a)]
- (ii) Click **File** → **New** in **IDLE Python Shell**. [Fig. 6.2(a)]
- (iii) In the New window that opens, type the commands you want to save in the form of a program (or script). [Fig. 6.2(b)]

For instance, for the simple **Hello World** program, you'll need to type following line :

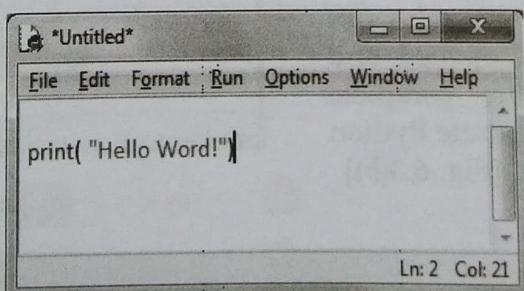
```
print ("Hello World ! ")
```

#### NOTE

You can display as well as print values in interactive mode, but for script mode, **print( )** command is preferably used to print results.



(a) **File** → **New** command in Python Shell



(b) Type commands in new blank file (script mode)

To see  
Working in Python IDLE  
in action



Scan  
QR Code

Figure 6.2

- (iv) Click **File** → **Save** and then save the file with an extension **.py**. The Python programs has **.py** extension [Fig. 6.2(c)]. For instance, we gave the name to our program as **Hello.py**.

4. Python GUI Shell IDLE lets you save the entire session (commands followed by their results – as it appears on screen) but that is not the Python program/script containing only the instructions.

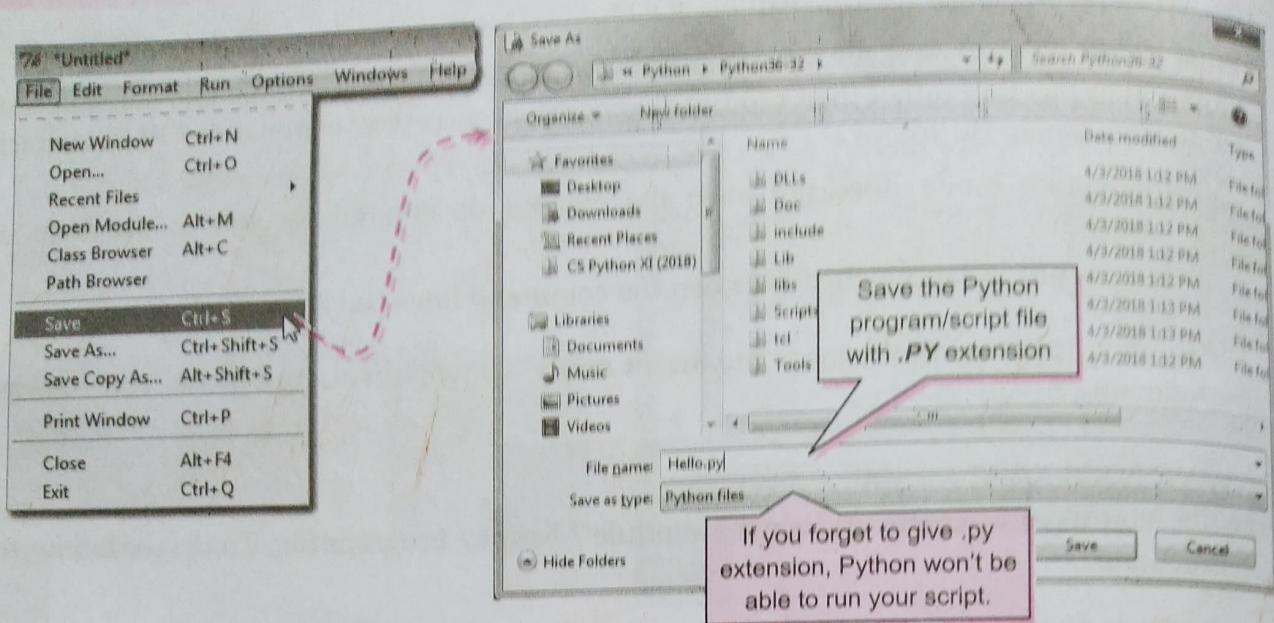


Figure 6.2 (c) Save file with .py extension with File → Save command (Script mode).

Now your program would be saved on the disk and the `save.py` file will have `.py` extension.

### Step 2 : Run Module / Script / Program File

After the program/script file is created, you can run it by following the given instructions :

- Open the desired program/script file that you created in previous Step 1 by using IDLE's File → Open command.  
If the program / script file is already open, you can directly move to next instruction.
- Click Run → Run Module command [Fig. 6.3(a)] in the open program / script file's window.  
You may also press F5 key.
- And it will execute all the commands stored in module / program / script that you had opened and show you the complete output in a separate Python Shell window. [Fig. 6.3(b)]

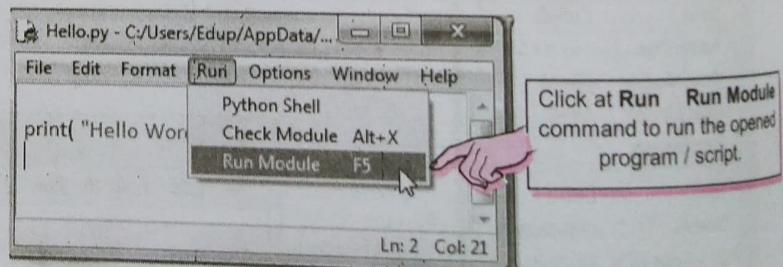


Figure 6.3 (a) Run → Run Module command (Script mode).

```

Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Edu/AppData/Local/Temp/Hello.py =
Hello World!
>>>

```

Figure 6.3 (b) Output of a module-run is shown in the shell window.

As you can see that with script mode, you can store all commands together in the form of a module / program / script and can get all output lines together. (No more command-output sandwiching :).

### 6.4.2 Working in Jupyter Notebook and Spyder IDE

Anaconda distribution comes with many preloaded packages and libraries. You can work in it in both interactive and script modes. Anaconda distribution provides the following tools that you can use to work in Python.

- ❖ **Jupyter notebook.** It is a web based, interactive computing environment.
- ❖ **Spyder.** It is a powerful Python IDE with many useful editing, interactive testing and debugging features.

Let us learn to work with both.

#### 6.4.2A Working in Jupyter Notebook

In order to work in *jupyter notebook*, you need to first launch it using **Anaconda Navigator**<sup>5</sup> as it has come preloaded with Anaconda distribution.

1. Launch Anaconda Navigator
2. From the Navigator window, click on Launch below **jupyter notebook** tile.



Figure 6.4 Launching applications from Anaconda Navigator.

5. Please note that *jupyter notebook* can also be installed separately, without Anaconda also.

3. Since **jupyter notebook** is a web based computing environment, it will be launched in a web browser.
- Your web browser will now show you **notebook dashboard** (see below)

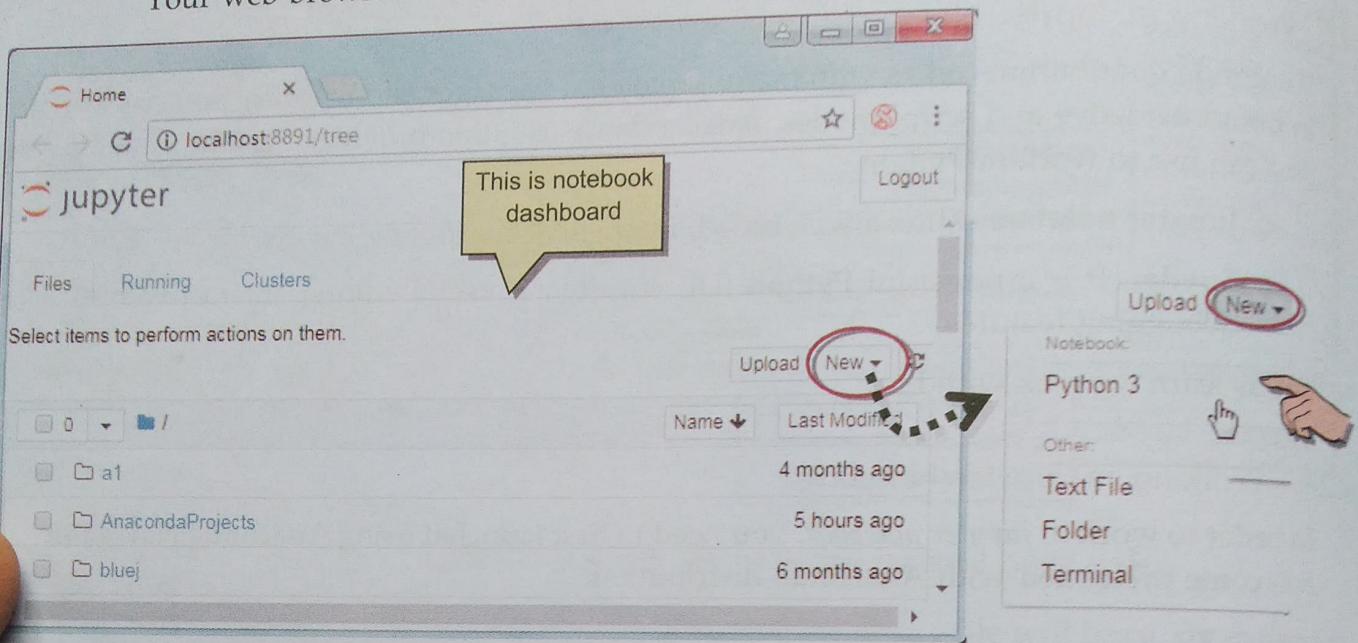


Figure 6.5

4. On the notebook dashboard, click at down-arrow next to **New** button and select **Python 3** to create a notebook for executing Python 3.x code. (see figure above).
5. In a new tab, it will open a new notebook where you can write and run your code.

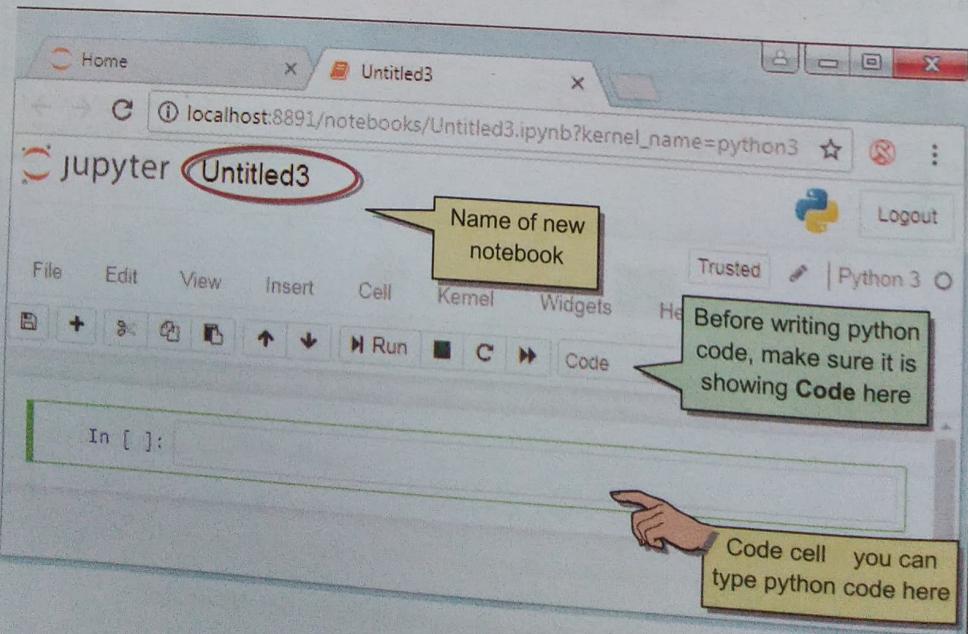


Figure 6.6

## Interactive Mode

- 6.** To run Python code in interactive mode, type code in code cell and click Run. (see below)

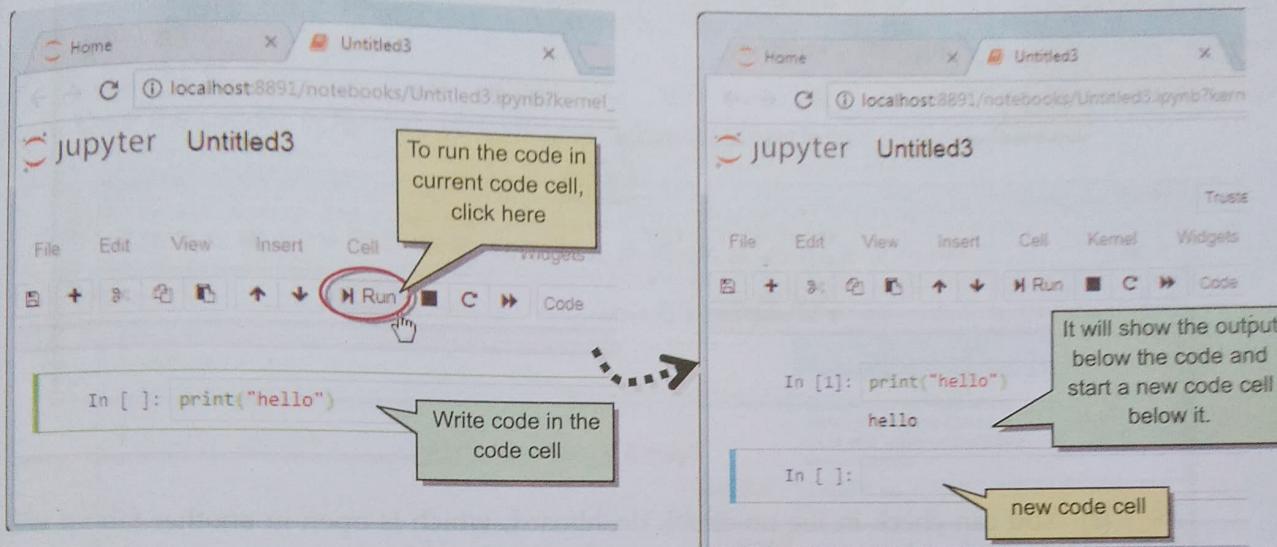


Figure 6.7

## Script Mode

- 7.** To run a script :

- (a) Write the code of a Python script – a group of Python commands – in a code cell.
- (b) Now click on its name to give it a new name (see below).

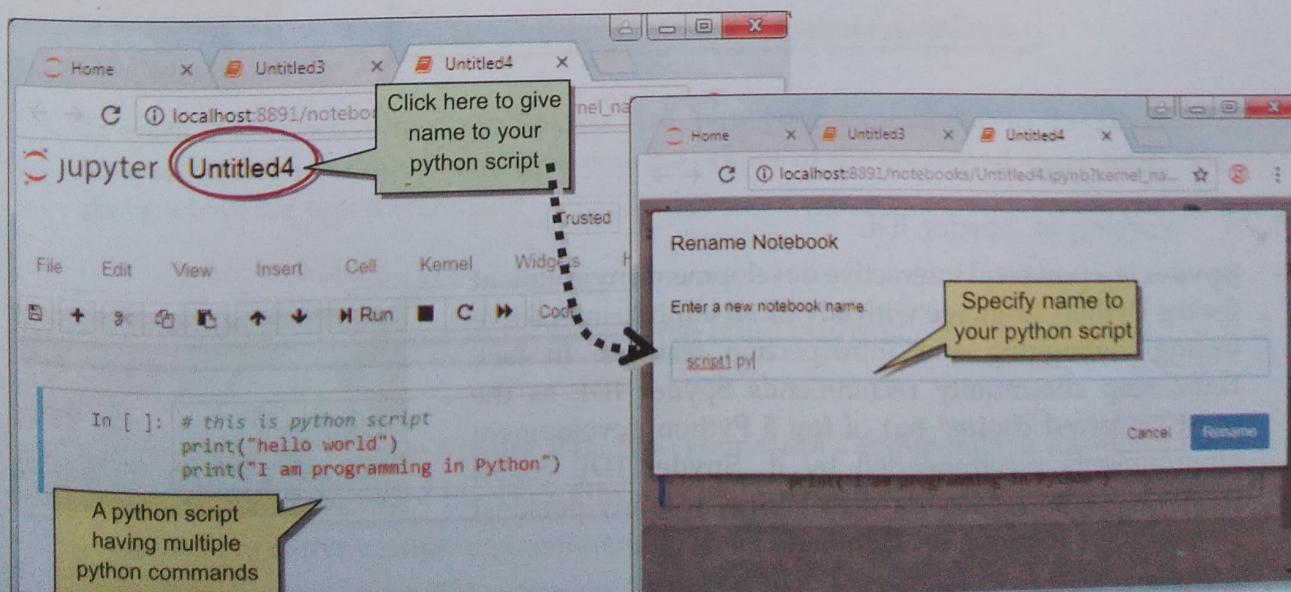


Figure 6.8

- (c) Click on save icon first to save it.
- (d) Click Run to run your script.

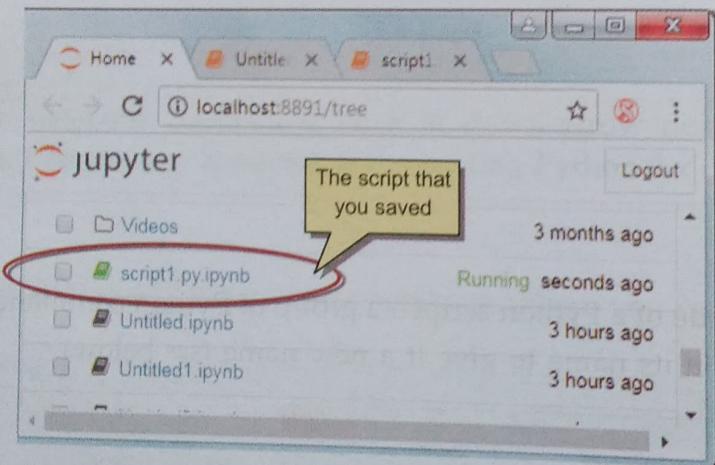
```

In [1]: # this is python script
       print("hello world")
       print("I am programming in Python")

       hello world
       I am programming in Python
    
```

Figure 6.9

- (e) You can check in the notebook dashboard, which is open in another tab ; it will show you the name of your currently saved script.



#### NOTE

To create a new notebook, use command  
File → New notebook.

Figure 6.10

#### 6.4.2B Working in Spyder IDE

**Spyder** is a powerful interactive development environment for the Python language with advanced editing, interactive testing, debugging and introspection features. In fact, DataCamp community recommends Spyder IDE as the most preferred choice<sup>6</sup> out of top 5 Python development environments recommended by it. Spyder IDE comes preloaded with Anaconda distribution. It is my personal favourite for working on Python.



6. As per URL : <https://www.datacamp.com/community/tutorials/data-science-python-ide>.

## To launch Spyder IDE :

- Launch Anaconda Navigator
- Click on Spyder tile in Anaconda navigator (refer to Fig. 6.4)  
[Alternatively, you can directly click at Start button → Programs → Anaconda (folder name) → Spyder

## Spyder Interface

Once the spyder is loaded, you will see following interface :

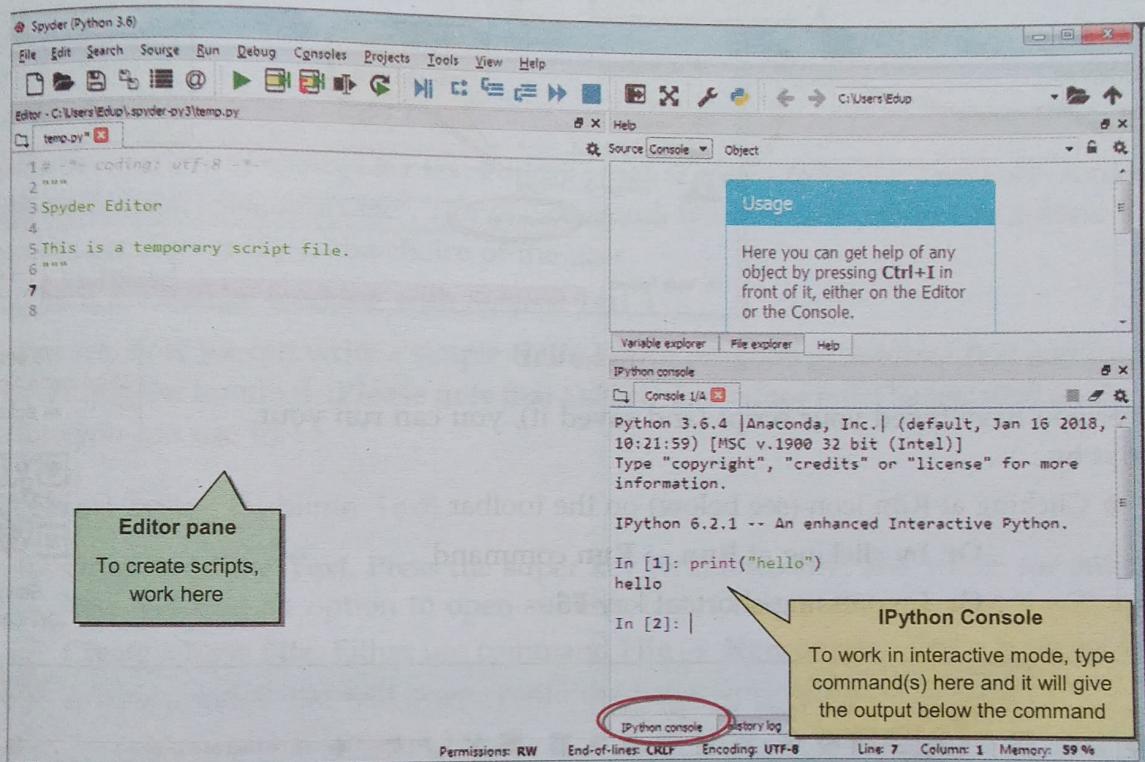


Figure 6.11

## Interactive Mode

To work in interactive mode in **Spyder IDE**, type your command(s) in the **IPython console** pane of spyder window. It will give you the output of the command there itself. See figure above where we typed command :

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

in the IPython console pane and it gave us the result below it.

## Script Mode

To work in script mode in **Spyder IDE**, type your script's commands in the editor pane.

- ⇒ To start a new script file, click **File → New File...**
- ⇒ To save current script, use command **File → Save** or **File → SaveAs**.
- ⇒ Python scripts have File extension as **.py**.
- ⇒ Please make sure to select file type as **Python Files**.

## NOTE

The IPython console takes commands in front of **In[ ]** prompt and often shows output with **Out[ ]** lines.

# Path Wala

COMPUTER SCIENCE WITH PYTHON - XI

150

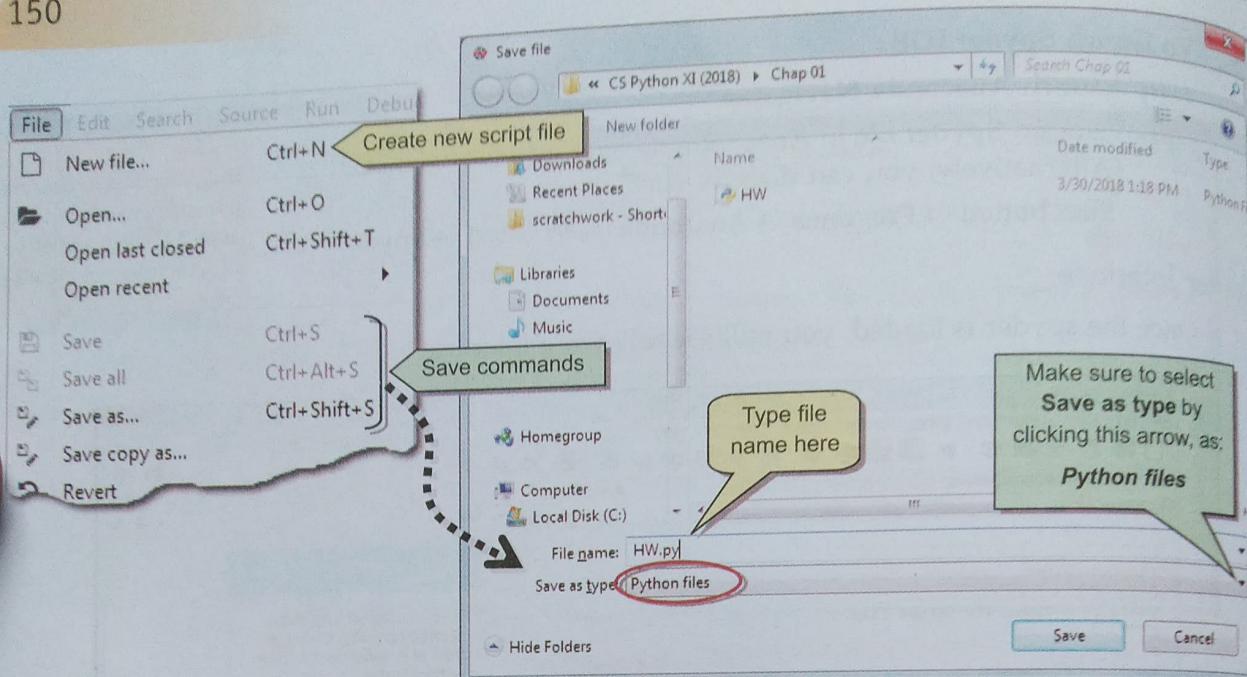


Figure 6.12

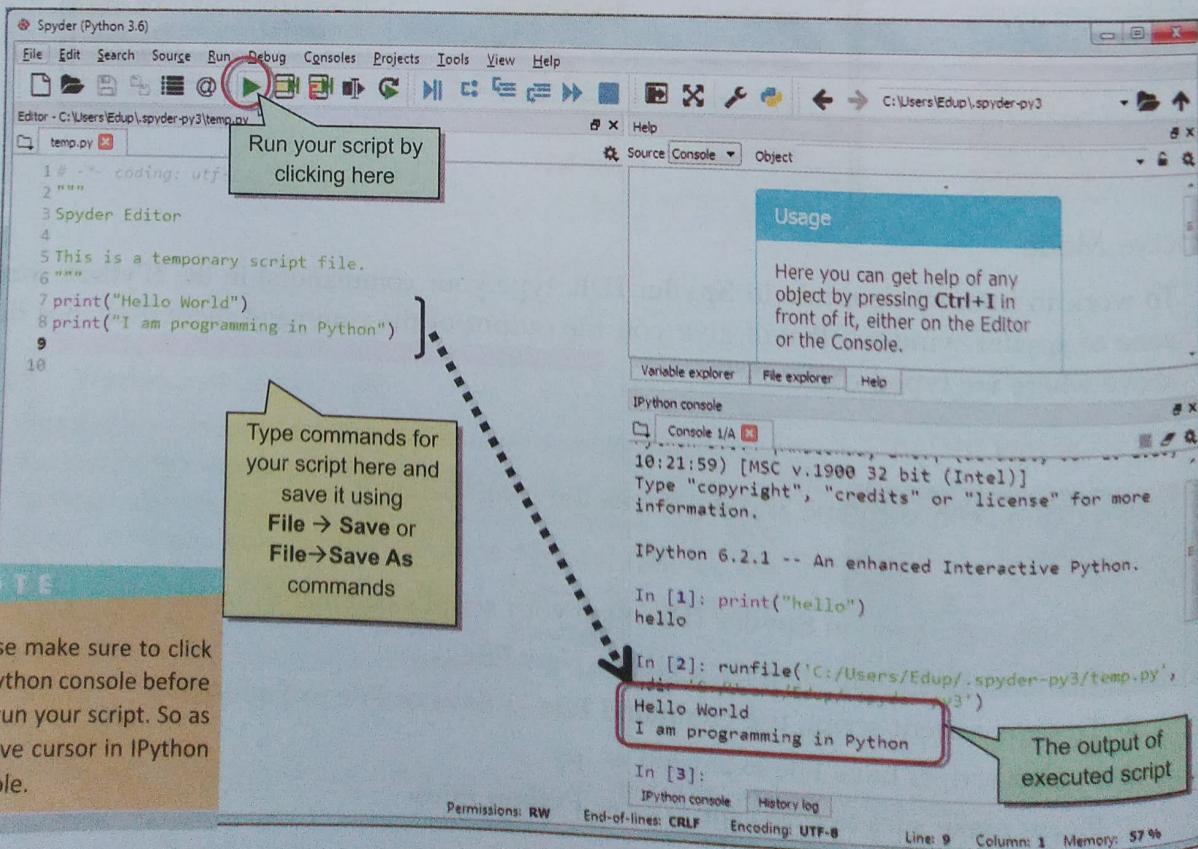
After you have typed your script (and saved it), you can run your script by :

- ❖ Clicking at Run icon (see below) on the toolbar
- Or by clicking at Run → Run command
- Or by pressing shortcut key F5.

To see  
Working in Spyder IDE  
in action



Scan  
QR Code



Now that you are familiar with different ways of working in Python, we can proceed with our discussion of Python. You can work on any of these.

### 6.4.3 Writing and Compiling Python Program with Command Line in Linux

To create and run a Python program on Linux platform, you need to do this :

- Write your program code in a text editor.
- Save your program with .py extension.
- Compile and run on the command prompt by giving appropriate command.

Let us see how.

#### Typing Python program in a Text Editor

There is a variety of choices for text editors when it comes to Linux. There are command line editors like *nano*, *vim*, *emacs* etc<sup>7</sup>. ; Or you can work with the GUI editors like *Sublime Text*, *atom*, *brackets* etc. This really is the choice of the user.

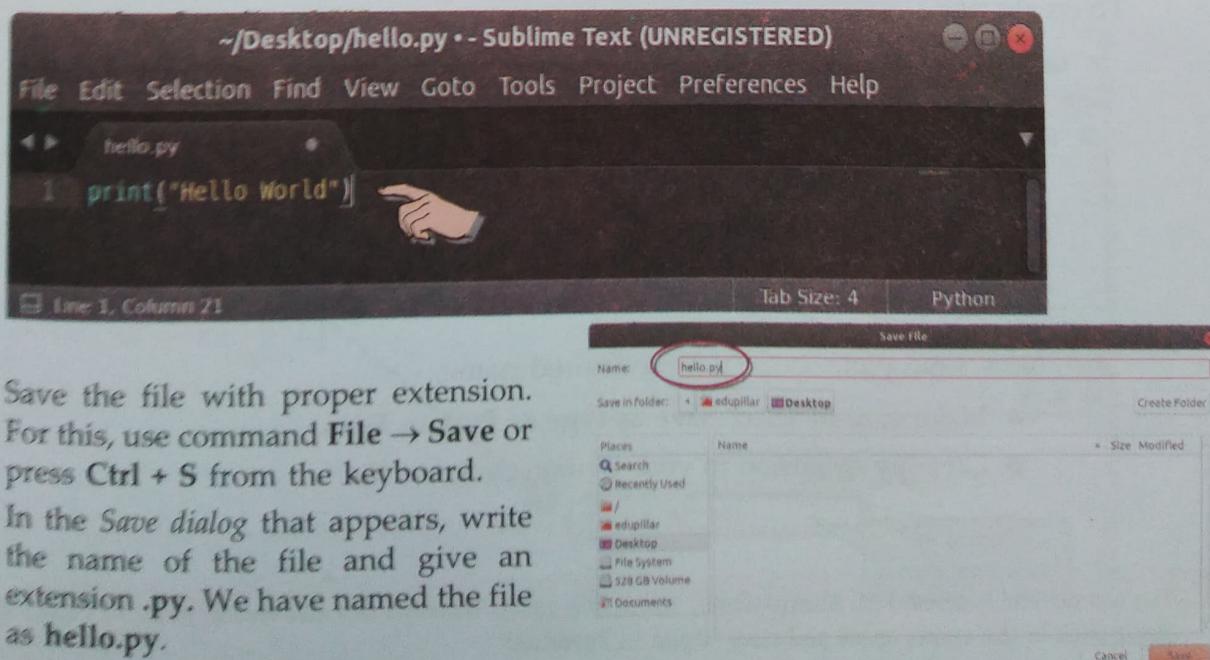
But here, we will be working with *Sublime Text* 3.

Let us see, how we can write a simple *Hello World* program in *Sublime Text* and compile and run it using the terminal. (Please note that *Sublime Text* editor must be installed on the machine before you can use it).

#### Coding in Text Editor (Sublime Text)

- Open Sublime Text. Press the super key (windows key) and search for Sublime Text. You will find an option to open sublime text. Click on the icon and it will appear.
- Create a New File. Either use command File → New or press **Ctrl + N** on the keyboard.
- A new untitled file will open. Write the hello world code :

```
print("Hello World")
```



- Save the file with proper extension. For this, use command File → Save or press **Ctrl + S** from the keyboard.

In the *Save dialog* that appears, write the name of the file and give an extension **.py**. We have named the file as **hello.py**.

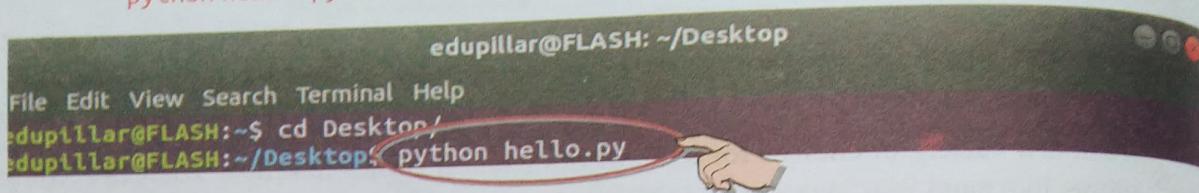
<sup>7</sup> *Vim* and *emacs* have their GUI version as well.

### Compiling and Running the Code using the Terminal

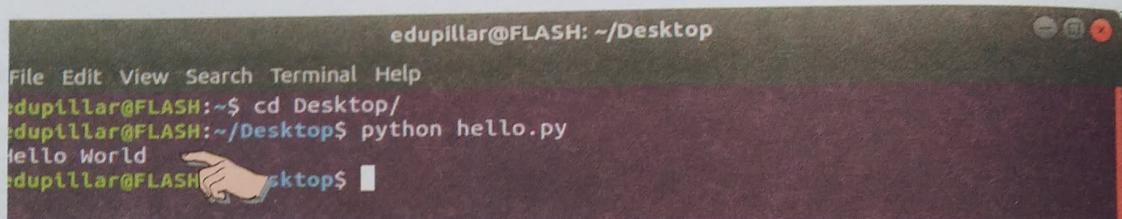
- (i) Open the terminal. Press **Ctrl + Alt + T** from the keyboard. A command prompt will appear.
- (ii) Go the folder where you saved the file<sup>8</sup>.
- (iii) Type the following command in front of the prompt and press enter :
- python <filename>.py

That is we should type following command to compile and run our program *hello.py*<sup>9</sup>:

python hello.py



- (iv) It will compile and run your program. You will see the output of the program in the terminal window.



### 6.5 UNDERSTANDING FIRST PROGRAM/SCRIPT

Let us create our first program – a simple “Hello World” program. (We are using Spyder IDE for this. You can use any of the above mentioned options.)

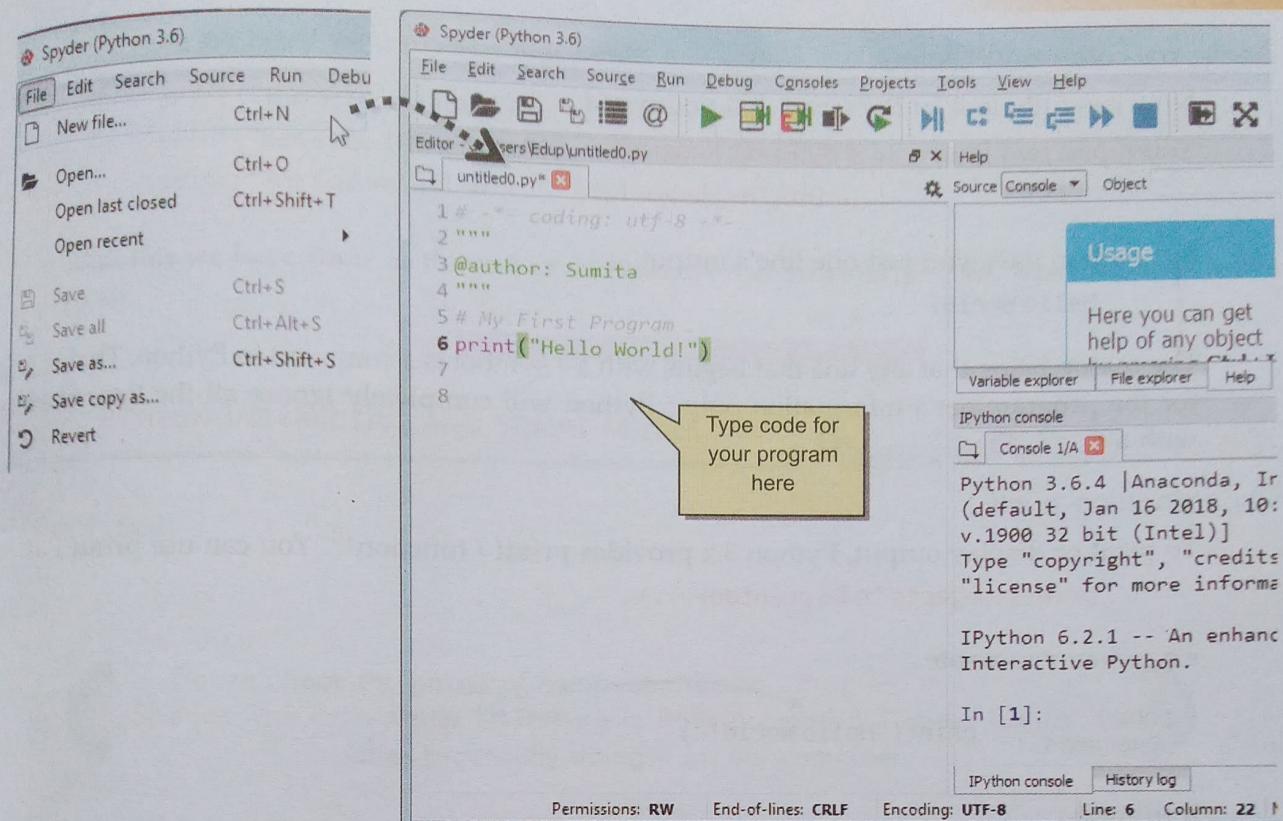
To create this,

1. Start Spyder IDE or any other editor of your choice.
  
2. Start new file (**File → New File**) and type the following text in the editor window :
 

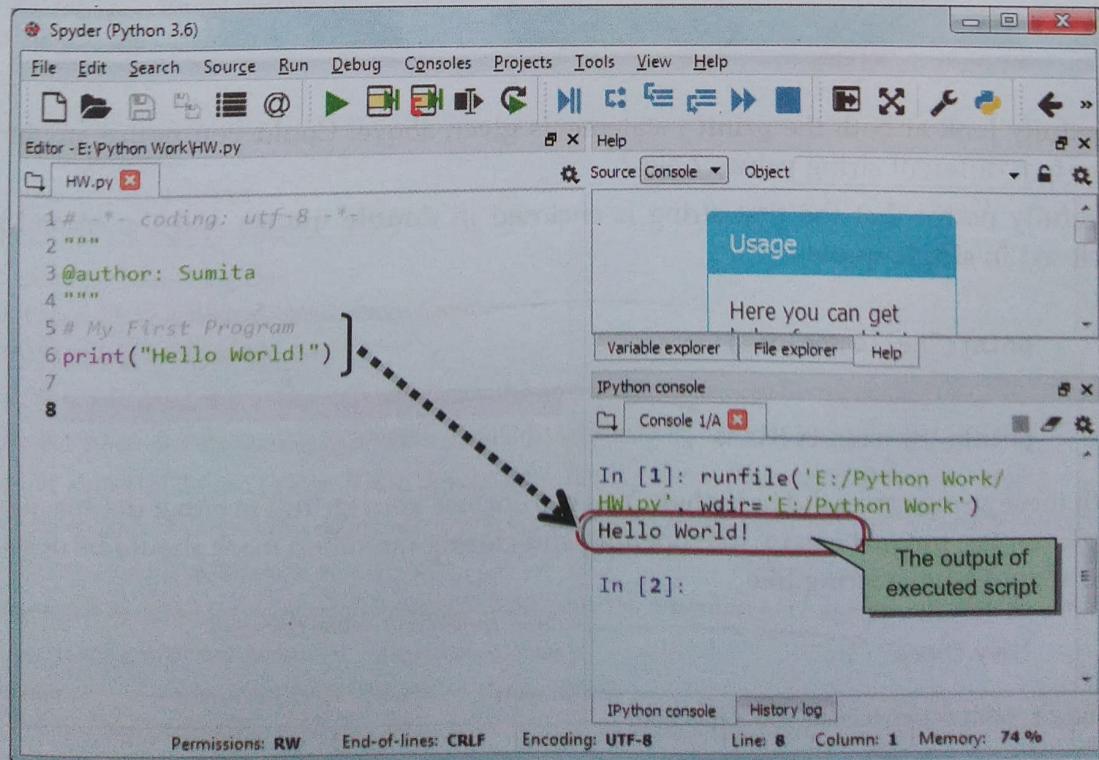
```
#My First Program
print("Hello World!")
```
  
3. Now save your script with a desired name.
  - Make sure to select **Save as type as Python Files**.
  - Give **.py** extension to your python program file.

8. You can use the command **cd**. Alternatively, open files (press Windows key and search for files.) Go to the correct folder. Right click in the empty space and click **“Open in Terminal”**.
9. There is no concept of extensions in Linux however, sublime text uses these extensions to identify the format and highlights the syntax accordingly. You can externally set the syntax format as well. But we are not covering that.

# Path Wala



4. Now run your script by clicking Run icon [▶] or by clicking Run → Run command or by pressing F5. (Firstly, click in IPython console window)
5. It will show you the output in the console window pane (see below).



## Analysing Script and Output

Now carefully look at the script that you typed.

You typed two lines : # My First Program  
print("Hello World")

But Python gave you just one line's output as:

Hello World!

The reason being that any line that begins with a `#` symbol is a comment in Python. That is, it is for the programmer's information only ; Python will completely ignore all the lines starting with a `#`.

## Understanding print( )

To print or display output, Python 3.x provides `print()` function<sup>10</sup>. You can use `print()` as  
`print(<objects to be printed>...)`

e.g., when you wrote

```
print()  
statement 1    print("Hello World!")
```

*String "Hello World!" is the object to be printed*

it printed :

Hello World!

Similarly, to print other strings you may give something like.

```
print( )  
statement?    print('My name is Misha')
```

*String object 'My name is Misha' to be printed.*

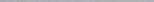
it will print

My name is Misha

Carefully look at both the `print( )` statements given above. Could you notice anything other than two different string values ? No ?

Carefully notice that the first string is enclosed in **double quotes** and the second string is enclosed in **single quotes**.

```
print("Hello World!")
```



```
print('My name is Misha')
```

*String enclosed in single quotes*

Both these strings are valid in Python. You can enclose your strings in either double quotes or in single quotes, but just ensure that opening and closing quotation mark should be of same type. You cannot have a string like

**'Hey there'** ← Error! opening and closing Quotation marks do not match

10. In Python 2.x, print is a statement, not a function.

Following are some valid `print()` statements.

```
print('The Golden Ratio')
print("has same letters as")
print('The God Relation')
```

With this we have come to the end of this chapter. Let us quickly revise what we have learnt so far.



### PYTHON INTERACTIVE AND SCRIPT MODES

Progress In Python 6.1 & 6.2

*Start Python*

:

Please check the practical component-book – *Progress in Computer Science with Python* and fill it there in PriPs 6.1 and 6.2 under Chapter 6 after practically doing it on the computer.

>>>❖<<<

## LET US REVISE

- ❖ Python was developed by Guido Van Rossum in February 1991.
- ❖ Python offers following advantages
  - ⌚ easy to use
  - ⌚ expressive
  - ⌚ complete
  - ⌚ cross-platform
  - ⌚ free and open source
- ❖ Python also has these limitations :
  - ⌚ Not the fastest language
  - ⌚ Lesser libraries than C, Java, Perl
  - ⌚ Not strong on Type-binding.
- ❖ In Python, one can work in two different ways : (i) Interactive mode, (ii) Script mode.
- ❖ Popular Python distributions are CPython (default), Anaconda distribution etc.
- ❖ Popular Python IDEs are Python IDLE, Spyder IDE, Jupyter Notes, Pycharm IDE etc.
- ❖ Interactive mode does not save commands in form of a program and also, output is sandwiched between commands.
- ❖ Interactive mode is suitable for testing code.
- ❖ Script mode is useful for creating programs and then run the programs later and get complete output
- ❖ Python is an interpreted language.
- ❖ Python's interactive interpreter is also called Python Shell.

**O**bjective Type Questions**O**TQs**M**ultiple Choice Questions

1. Python is a/an \_\_\_\_\_ language.
  - (a) High level
  - (b) Object oriented
  - (c) procedural
  - (d) difficult
  
2. Python uses a/an \_\_\_\_\_ to convert source code to object code.
  - (a) Interpreter
  - (b) Compiler
  - (c) Combination of Interpreter and compiler
  - (d) Special virtual engine
  
3. Python code can run on a variety of platforms, it means Python is a \_\_\_\_\_ language.
  - (a) Graphical
  - (b) Cross-platform
  - (c) independent
  - (d) all of these
  
4. Python programs are typed in
  - (a) Interactive mode
  - (b) Script mode
  - (c) A combination of interactive and script modes
  - (d) All of these
  
5. The \_\_\_\_\_ mode of Python gives instant result of typed statement
  - (a) Interactive mode
  - (b) Script mode
  - (c) Combination of interactive and script modes
  - (d) All of these
  
6. Which of the following is not a Python IDE ?
  - (a) IDLE
  - (b) Spyder
  - (c) Jupyter Notes
  - (d) Sublime Text
  
7. To print the value of a variable, Python uses
  - (a) Print statement
  - (b) Print( ) function
  - (c) print statement
  - (d) print( ) function
  
8. You don't have to pay for **Python** and you can view its source code too. It means Python is \_\_\_\_\_.
  - (a) Free and open source
  - (b) freeware
  - (c) open source
  - (d) shareware

**F**ill in the Blanks

1. Python is a \_\_\_\_\_ level language.
2. Python's two working modes are : \_\_\_\_\_ mode and \_\_\_\_\_ mode.
3. The shortcut key to run a Python program from script mode is \_\_\_\_\_.
4. Python programs/scripts are stored in files with \_\_\_\_\_ extension.
5. Python's default distribution's IDE is called \_\_\_\_\_.

## True/False Questions

1. Python is the fastest language.
2. Python code is compiled before running.
3. You can create programs in Python's interactive mode.
4. You can create programs in Python's script mode.
5. Python is an interpreted language.

**NOTE :** Answers for OTQs are given at the end of the book.

## Solved Problems

1. Who developed Python Programming Language ?

**Solution.** Guido Van Rossum in 1990s developed Python programming language.

2. Is Python an Object Oriented language ?

**Solution.** Yes, Python is an Object Oriented language.

3. 'Python is an interpreted high level language'. What does it mean to you ?

**Solution.** '**Python is a high level language**' means it is programmer-friendly i.e., easy to program and comprehend.

'**Python is an interpreted language**' means it requires an interpreter (not compiler) to execute its code line by line – one statement at a time.

4. Python programming language got its name from which show.

**Solution.** Python programming language was named after a British TV show namely 'Monty Python's Flying Circus'.

5. What does a cross platform language mean ?

**Solution.** A cross platform language means it can run well on variety of platforms like Windows, Linux/Unix, Macintosh etc. etc.

6. Python is a Free and Open Source language. What do you understand by this feature ?

**Solution.** It means – to download Python, one needs not pay anything, because it is **Free**. And its source-code is also available, which can be modified/improved etc., because it is **open-source**.

7. What is the difference between interactive mode and script mode in Python ?

**Solution.** In *interactive mode*, instructions are given in front of Python prompt (e.g., >>> or In[ ]: prompts) in Python Shell. Python carries out the given instruction and shows the result there itself.

In *script mode*, Python instructions are stored in a file generally with .py extension and are executed together in one go as a unit. The saved instructions are known as *Python script* or *Python program*.

8. What will be the output of following code :

```
#This is a sample program
#to output simple statements
#print ("Such as")
print("Take every chance.")
print("Drop every fear.")
```

Pick the correct output from the following choices and give reason. This is a sample program to output simple statements

- |   |   |  |
|---|---|--|
| (a) Such as<br>Take every chance.<br>Drop every fear. | (b) Such as<br>Take every chance.<br>Drop every fear. | (c) Take every chance.<br>Drop every fear. |
|---|---|--|

**Solution.** The correct output is (c).

**Reason being :** the code lines beginning with a # sign are comments. They are just for information and ignored by the Python interpreter. Hence, the third line `#print ("Such as")` will also be ignored by Python interpreter. Thus, the Python interpreter will give output of only `print()` statements.

9. Which of the following are not valid strings in Python ?  
 (a) "Hello"    (b) 'Hello'    (c) "Hello"    (d) 'Hello"    (e) {Hello}

**Solution.** Strings (c), (d) and (e) are not valid strings in Python.

## GLOSSARY

**Python Shell**      Interactive interpreter of Python.

**Source Code**      Complete program instructions.

## Assignments

### Type A : Short Answer Questions/Conceptual Questions

1. When was Python released ?
2. Who was Python's developer and which two languages contributed to Python as a programming language ?
3. What is a cross-platform software ?
4. What are the advantages of Python programming language ?
5. What are some limitations of Python programming language ?
6. In how many different ways, can you work in Python ?
7. What are advantages/disadvantages of working in Interactive mode in Python ?
8. What are the advantages/disadvantages of working in script mode in Python ?

### Type B : Application Based Question

1. Write instructions to get the following result :

Math is Fun so don't be resistant

Just learn the rules, the rules are consistent

And most important, you must be persistent !

Adding fractions, get common denominators.

Multiply by missing factors to get the denominators.

Add numerators only, NOT denominators.

Do it in both interactive mode and script mode.