CHAPTER :1

INTRODUCTION TO PYTHON

- Python is a general-purpose interpreted
- Interpreted: they do not compile. Directly run the program.
- Interactive: user can directly interact with the python shell.
- Object-oriented: Obey oops concept.

FEATURES OF PYTHON

- Easy-to-learn
- Easy-to-read
- Easy-to-maintain
- A broad standard library
- free and open source software
- Interactive Mode
- Portable

APPLICATIONS OF PYTHON PROGRAMMING

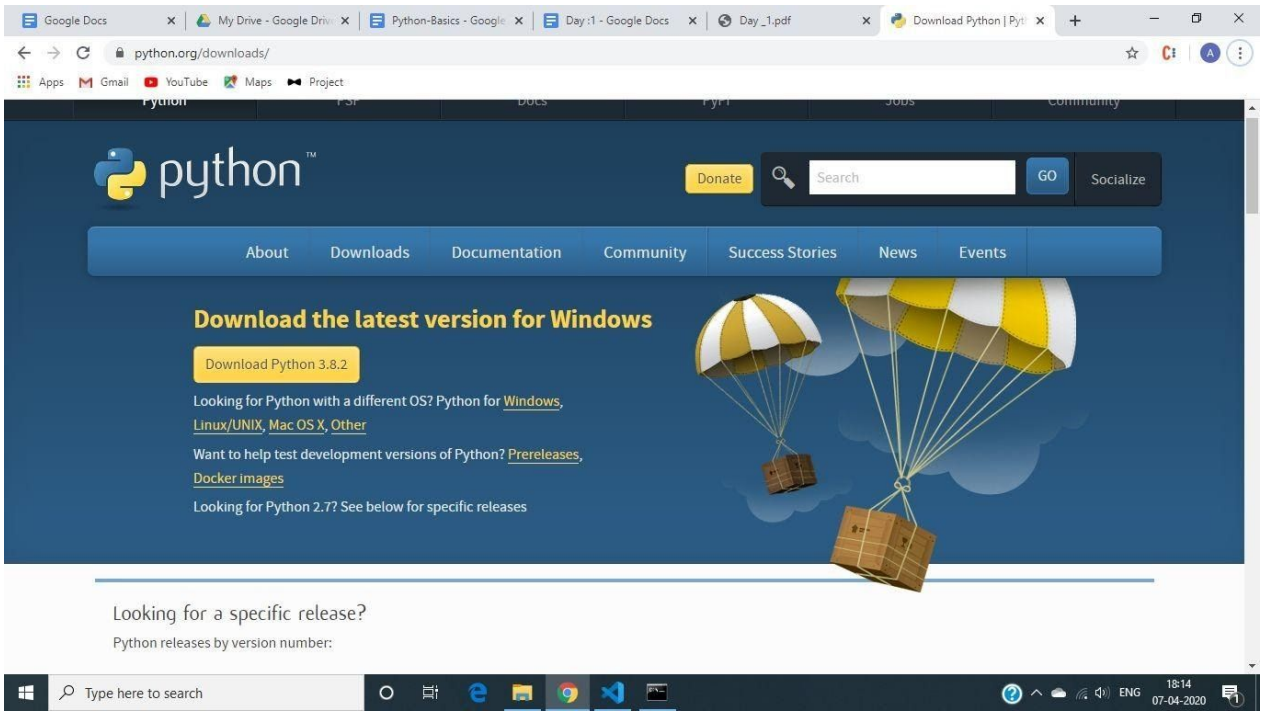
- Web and Internet Development.
- Desktop GUI Applications.
- Science and Numeric.
- Software Development.
- Education.
- Database Access.
- Network Programming.
- Games and 3D Graphics

ENVIRONMENT SETUP

- Windows Installation

Here are the steps to install Python on a Windows machine.

Step1:Open a Web browser and go to **https://www.python.org/downloads**



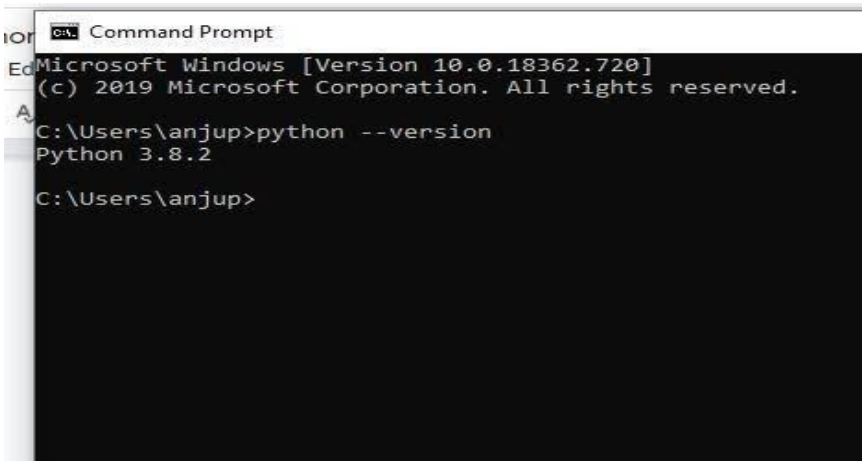
Step2:Follow the link for the Windows installer ***python-XYZ.msi*** file where XYZ is the version you need to install.

Step3: To use this installer *python-XYZ.msi*, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.

Step4: Run the downloaded file. This brings up the Python install wizard

Step5:Check the Python version using the command:-

```
python --version
```



RUNNING PYTHON

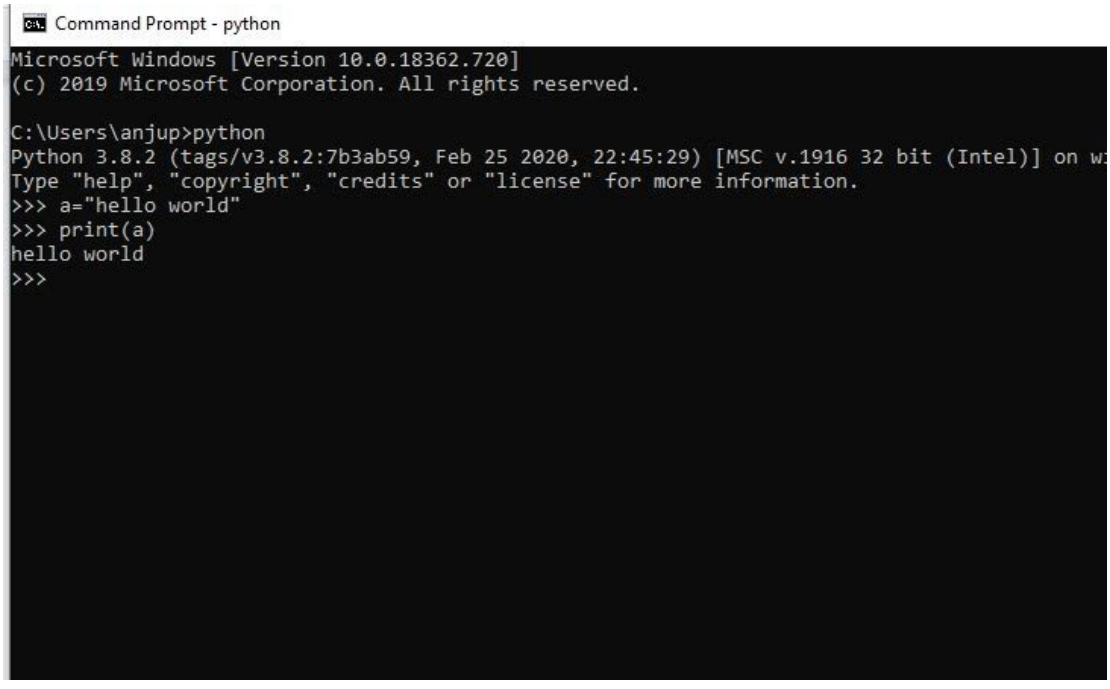
There are three different ways to start Python −

Interactive Interpreter

You can start Python from Unix, DOS, or any other system that provides you a command-line interpreter or shell window.

>>Interactive Mode Programming

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



>>**Script Mode Programming:**

Let us write a simple Python program in a script.

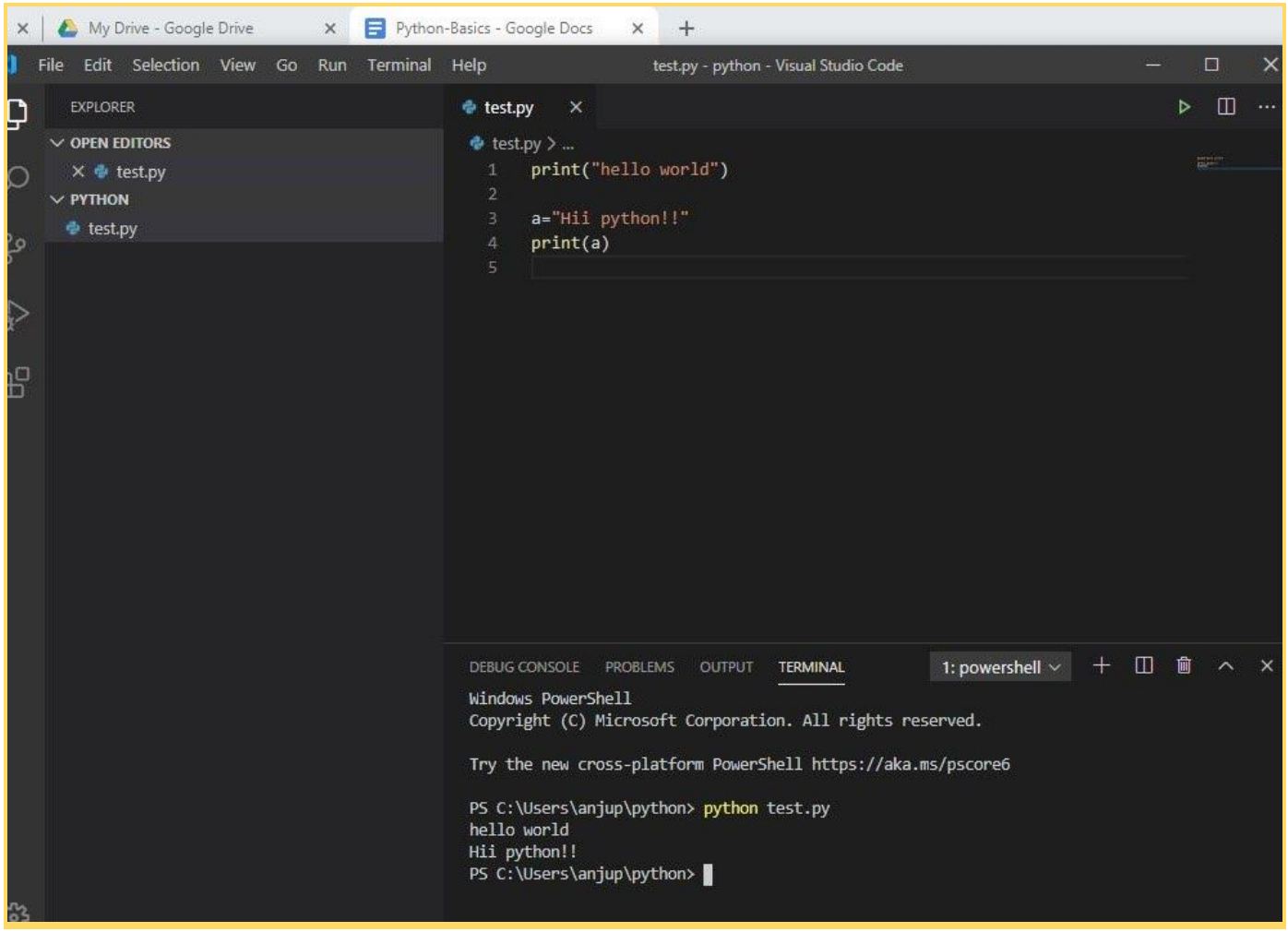
Python files have extension .py.

Type the following source code in a test.py file as shown in the fig.

print(“Hello World!!”)

a=”Hello World”

print(a)



KEYWORDS IN PYTHON

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

LINES AND INDENTATION

- Python provides no braces to indicate blocks of code for class and function definitions or flow control. Blocks of code are denoted by line indentation, which is rigidly enforced.
- The number of spaces in the indentation is variable, but all statements within the block must be indented the same amount. For example −

```
i f True:
    print "True"
 else:
    print "False"
```

COMMENTS IN PYTHON

A hash sign (#) that is not inside a string literal begins a comment. All characters after the # and up to the end of the physical line are part of the comment and the Python interpreter ignores.

Following triple-quoted string is also ignored by Python interpreter and can be used as a multiline comments:

```
#print("this is a single line comment")

''' 

This is a multi line

comment.

'''
```

DATA TYPES

- Numbers
- Strings
- List
- Tuple
- Dictionary
- Boolean
- Sets

PYTHON NUMBERS

Number data types store numeric values. Number objects are created when you assign a value to them.

For example −

```
a=20
b=30.5
c=3+4j
print(type(a))
print(type(b))
print(type(c))
```

Output:

<class 'int'>

<class 'float'>

<class 'complex'>

- Python supports integers, floating point numbers and complex numbers. They are defined as int, float and complex classes in Python.
- Integers and floating points are separated by the presence or absence of a decimal point. 5 is integer whereas 5.0 is a floating point number.
- Complex numbers are written in the form, x + yj, where x is the real part and y is the imaginary part.
- We can use the type() function to know which class a variable or a value belongs to

Example:

```
a = 5
print(type(a))
# Output: <class 'int'> (Integers)

print(type(5.0))
# Output: <class 'float'>  (Floating Point)
c = 5 + 3j
print(c+3)
# Output: (8+3j)  (Complex number)
```

2.STRINGS

- Strings in Python are identified as a contiguous set of characters represented in the quotation marks.
- Python allows for either pairs of single or double quotes.
- Subsets of strings can be taken using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.
- The plus (+) sign is the string concatenation operator and the asterisk (*) is the repetition operator.

For example −

```
mystr = 'Hello World!'
print(mystr)
```

Output: Hello World!

IN-BUILT STRING HANDLING FUNCTIONS

strip(): removes white space from beginning or end.

len( ) : method returns the length of a string.

lower() : method returns the string in lower

upper(): method returns the string in upper case:

replace() : method replaces a string with another string

split() :method splits the string into substrings if it finds an instance of separator.

The above fig. shows the examples of string methods in python.

```
a="  haii welcome  "
print(a.strip())        #removes white space from beginning and end

print(len(a))           #prints length of the string

b="PYTHON IS A GREAT LANGUAGE"
print(b.lower())        #prints b in lower case

print(a.upper())        #prints a in uppercase

c="I like Rose"
print(c.replace("Rose","Jasmine"))      #replaces the string "Rose" to "Jasmine"

print(c.split())        #Splits the string c with whitespace into a list
print(a[2:5])           #Accessing elements using slicing operator
```

Output:

haii welcome
16
python is a great language
HAII WELCOME
I like Jasmine
['I', 'like', 'Rose']
hai