

ಜೆಎಸ್‌ಎಸ್ ಪಾಲಿಟೆಕ್ನಿಕ್
ಮೈಸೂರು - ೫೭೦ ೦೦೬



JSS POLYTECHNIC

Mysuru - 570 006

JSS Polytechnic

JSS Technical Institutions' Campus

Mysuru - 570 006

Email Id: jssp418@yahoo.co.in

Website: www.jsspolytechnic.org

WEBINAR ON PYTHON PROGRAMMING

(C20 SYLLABUS)

BY

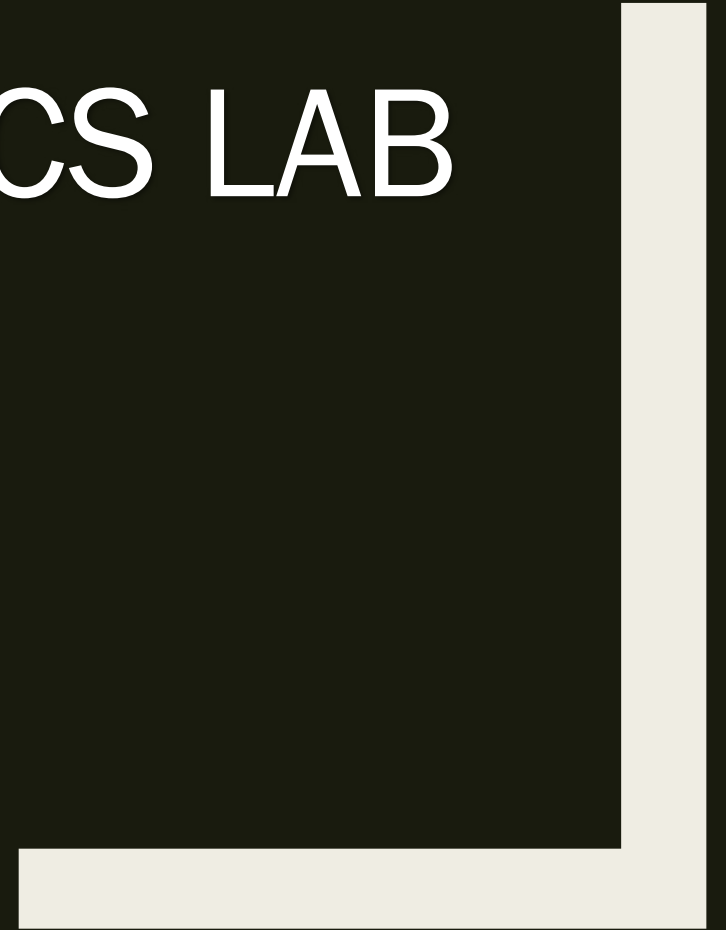
DEPARTMENT OF COMPUTER SCIENCE & ENGG

&

DEPARTMENT OF INFORMATION SCIENCE & ENGG

STATISTICS & ANALYTICS LAB

PART - 3



PYTHON PROGRAMMING



By,

Shinu Koshy
HOD
Dept. of CS & E
JSS Polytechnic Mysuru
Email: shinu@jsspolytechnic.org

Arun J
Lecturer
Dept. of IS & E
JSS Polytechnic Mysuru
Email: arunj@jsspolytechnic.org

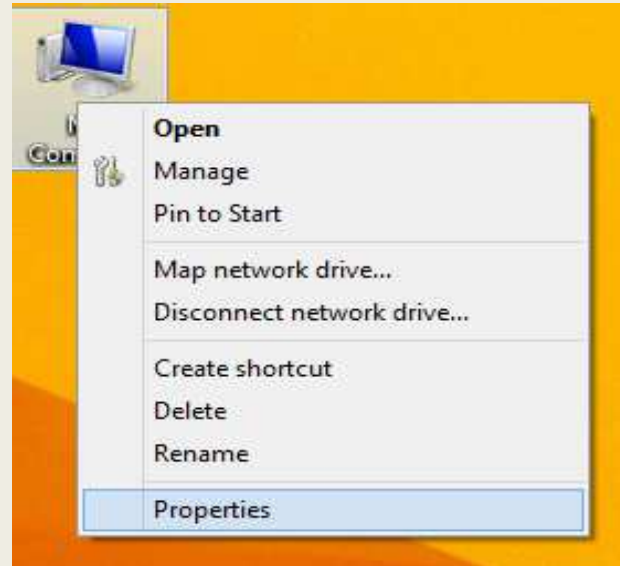
PYTHON PROGRAMMING INSTALLATION PART



Check your system OS/ Architecture

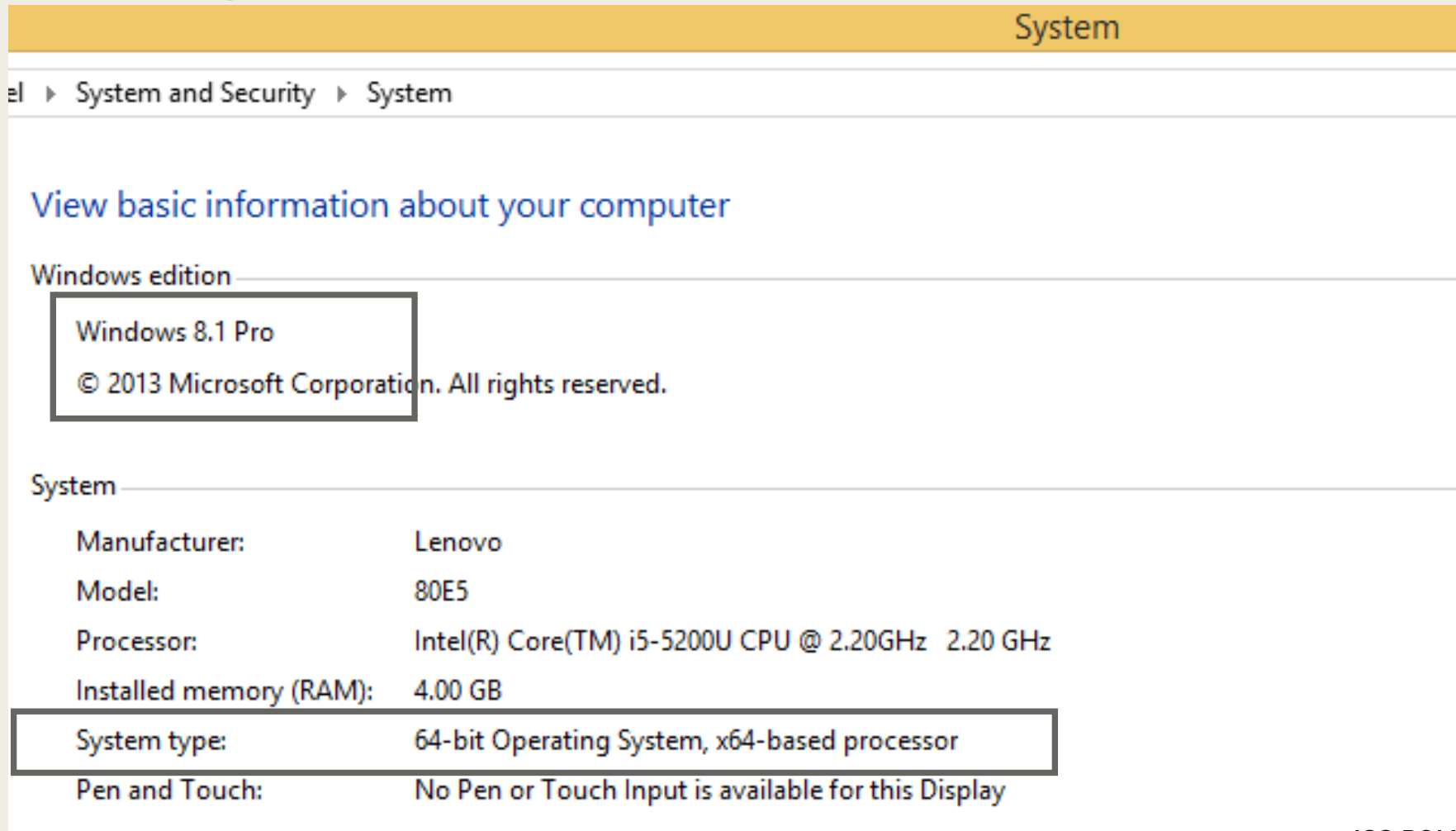
- Operating System (Windows 7 / Windows 8/ Windows 8.1 / Windows 10)
- Processor type (32 bit or 64 bit)

On the Desktop Right Click on My Computer → Select Properties



Check your system OS/ Architecture

When we Right Click on My Computer → Select Properties we will get the system configuration window



The screenshot shows the Windows System properties window. The title bar is yellow and says 'System'. The breadcrumb path is 'Control Panel > System and Security > System'. Below this is a link 'View basic information about your computer'. The 'Windows edition' section is highlighted with a black box and contains the text 'Windows 8.1 Pro' and '© 2013 Microsoft Corporation. All rights reserved.'. The 'System' section lists various hardware and software details. The 'System type' row is highlighted with a black box and shows '64-bit Operating System, x64-based processor'. The 'Pen and Touch' section at the bottom states 'No Pen or Touch Input is available for this Display'.

System	
View basic information about your computer	
Windows edition	
Windows 8.1 Pro © 2013 Microsoft Corporation. All rights reserved.	
System	
Manufacturer:	Lenovo
Model:	80E5
Processor:	Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz 2.20 GHz
Installed memory (RAM):	4.00 GB
System type:	64-bit Operating System, x64-based processor
Pen and Touch:	No Pen or Touch Input is available for this Display

Versions of Python

The table below illustrates the different versions of Python to be installed based on the Operating System (OS)

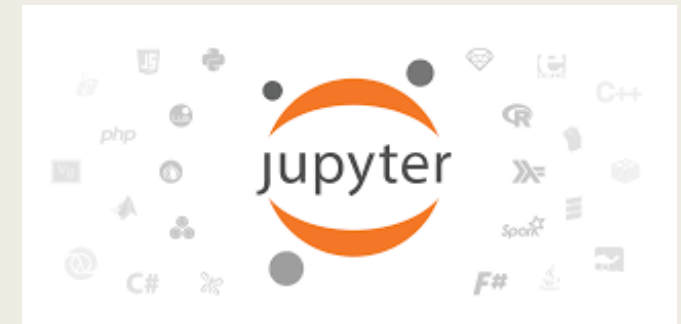
Operating System (OS)	Python Version
Windows 10	Latest Version (Python 3.9.1)
Windows 8.1	Latest Version (Python 3.9.1)
Windows 8	Python 3.8.6
Windows 7	Python 3.8.6
Windows Vista	Python 3.6
Windows XP	Python 3.4

Software Requirements

- Python (Version → Based on your system architecture)

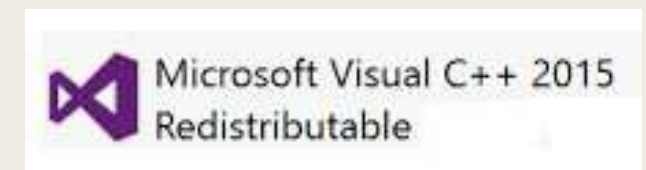


- Jupyter Notebook



Optional:

- Microsoft Visual C++ 2012/2015 Redistributable
(If Not already installed)



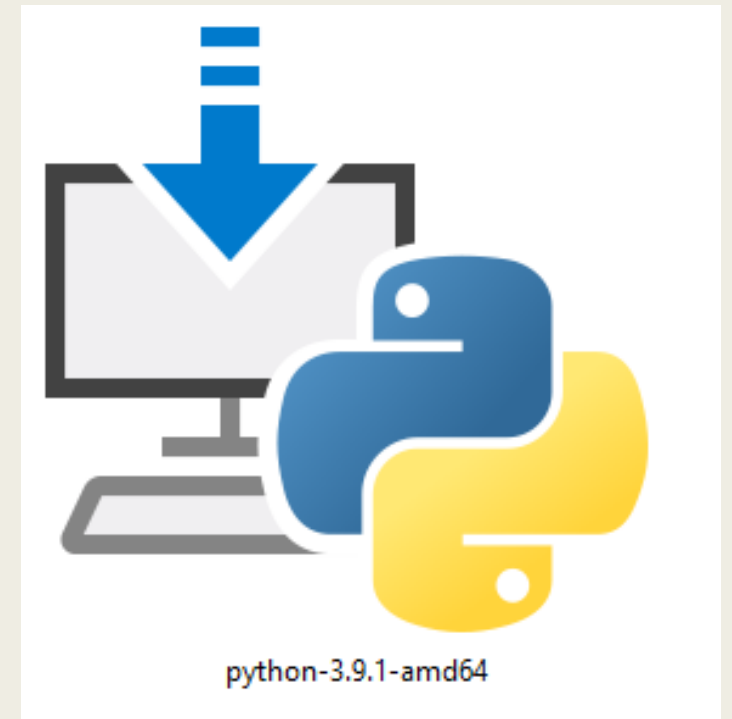
How to Download Python on Windows

- Download Python software from <https://www.python.org/downloads/>



How to Install Python on Windows

- Go to the folder where Python has been downloaded.
- You can see python-3.9.1.exe.
- Double click on it to run.



How to Install Python on Windows (Contd..)

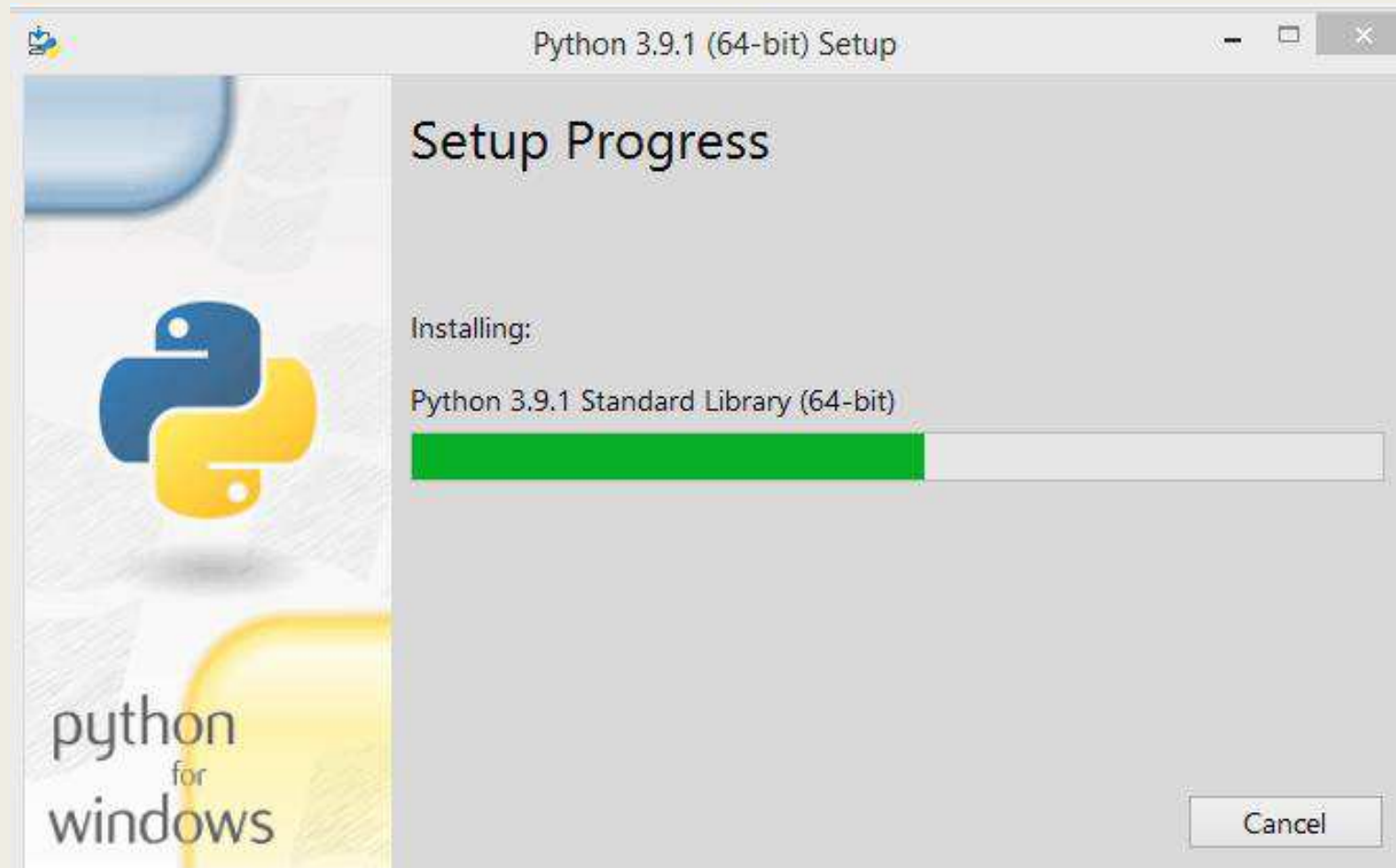
- After clicking Run, you can see a window like what you are noticing now:

Note: Must enable the check box (Add Python 3.9 to PATH)



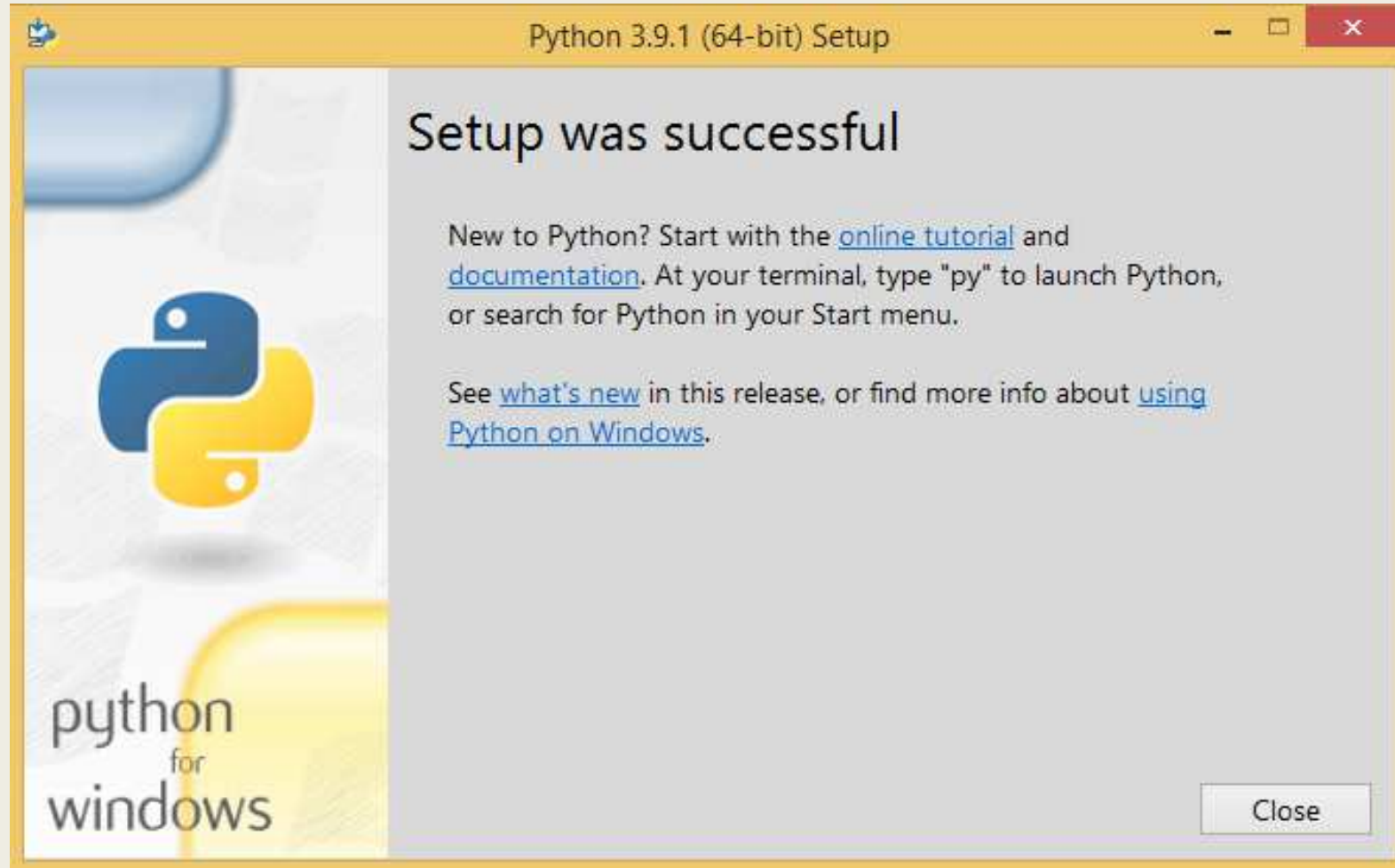
How to Install Python on Windows (Contd..)

Your installation should now begin:



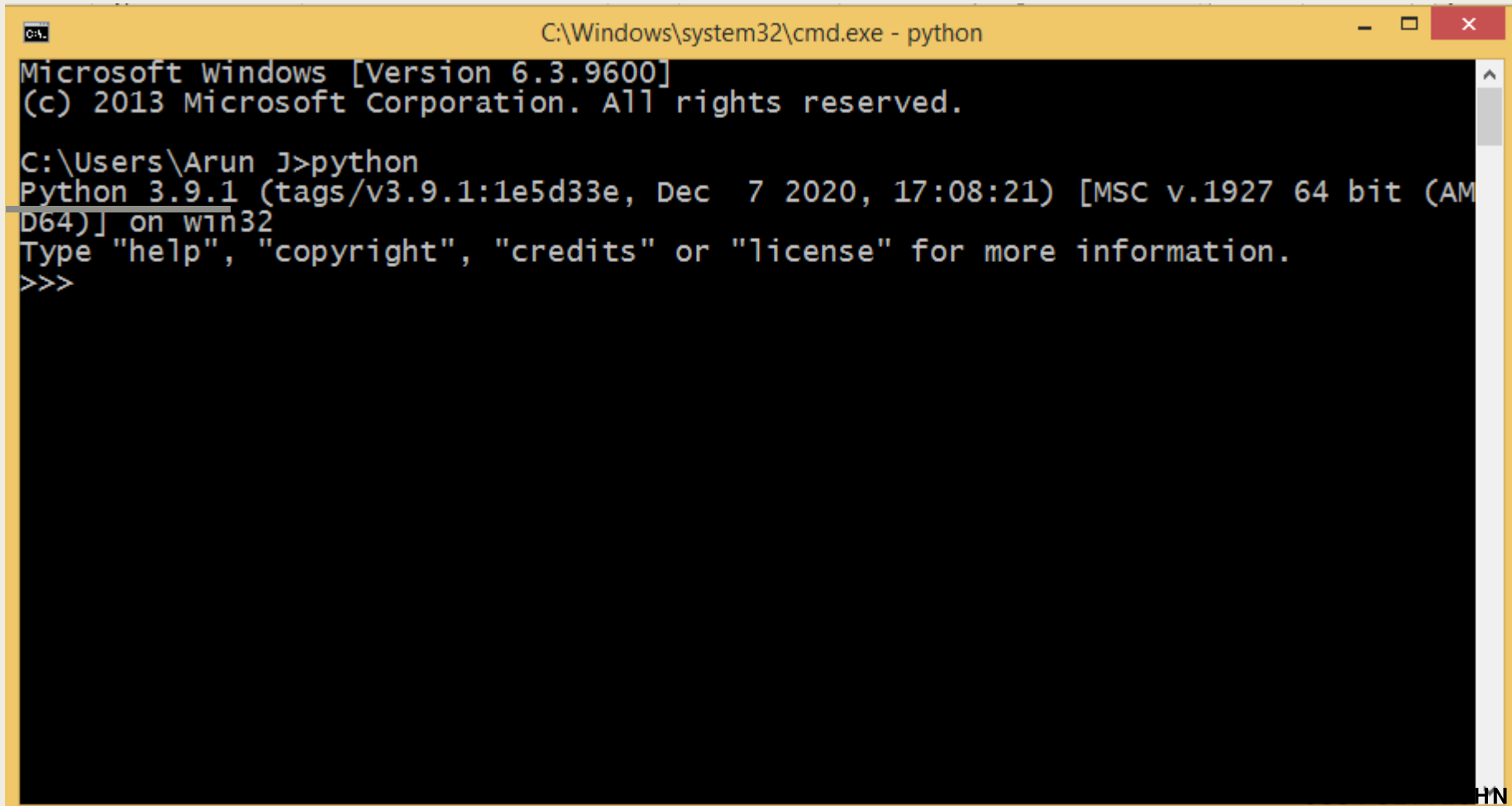
How to Install Python on Windows (Contd..)

After a short period of time, your setup would be completed:



Checking Python installed or not

After successful installation of python software its necessary to check in the command prompt by executing “**python**” command.

A screenshot of a Windows Command Prompt window. The title bar reads 'C:\Windows\system32\cmd.exe - python'. The window content shows the following text: 'Microsoft Windows [Version 6.3.9600] (c) 2013 Microsoft Corporation. All rights reserved. C:\Users\Arun J>python Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>>'. The prompt is now at the interactive shell '>>>'.

```
C:\Windows\system32\cmd.exe - python
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Arun J>python
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

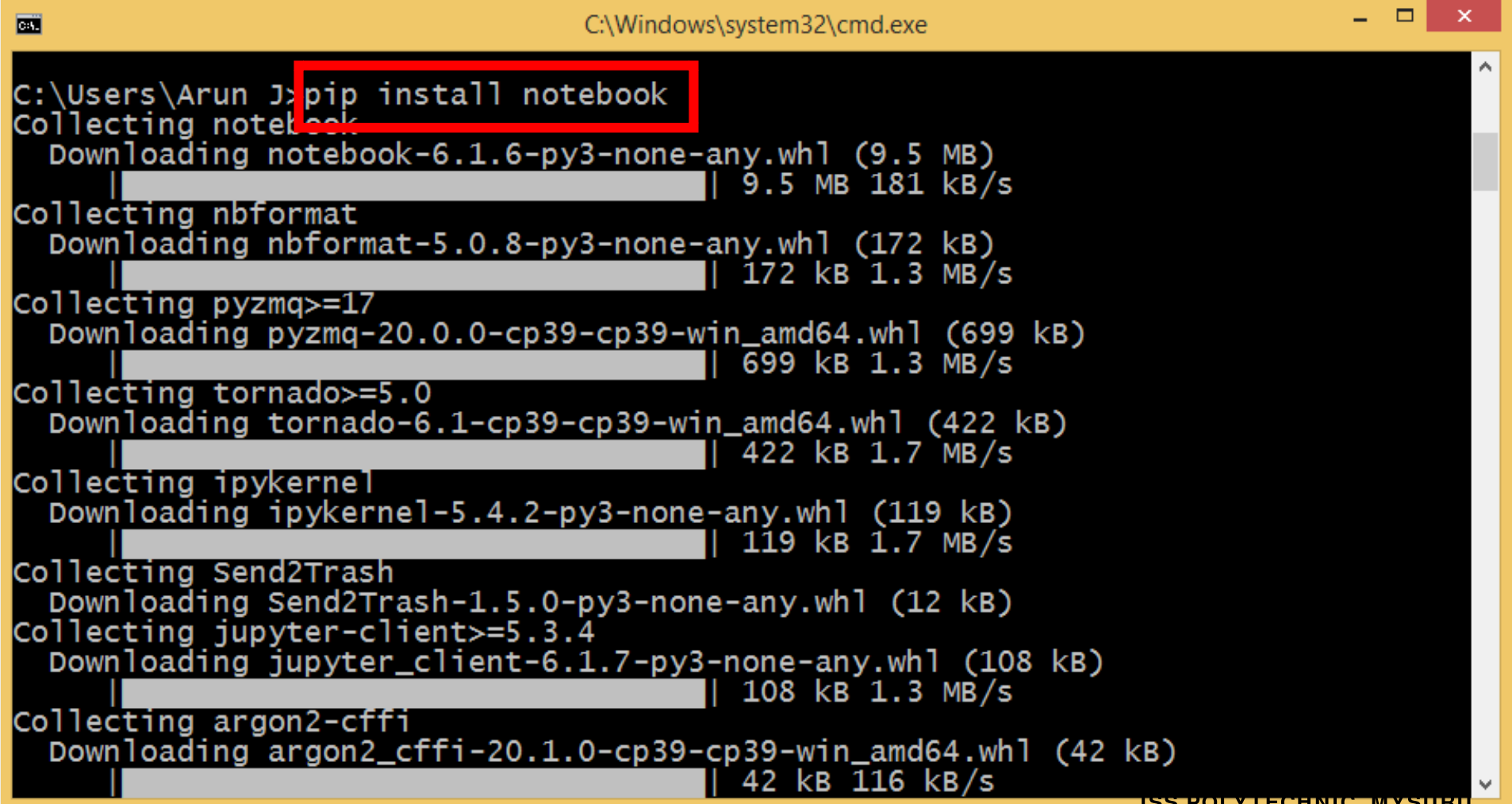
About Jupyter Notebook

- The Jupyter Notebook is an interactive way of running the Python code.
- It is an open-source web application that allows us to create and share documents that contain live code, equations, visualizations and narrative text.



How to install Jupyter Notebook

Go to command prompt and type: **pip install notebook**

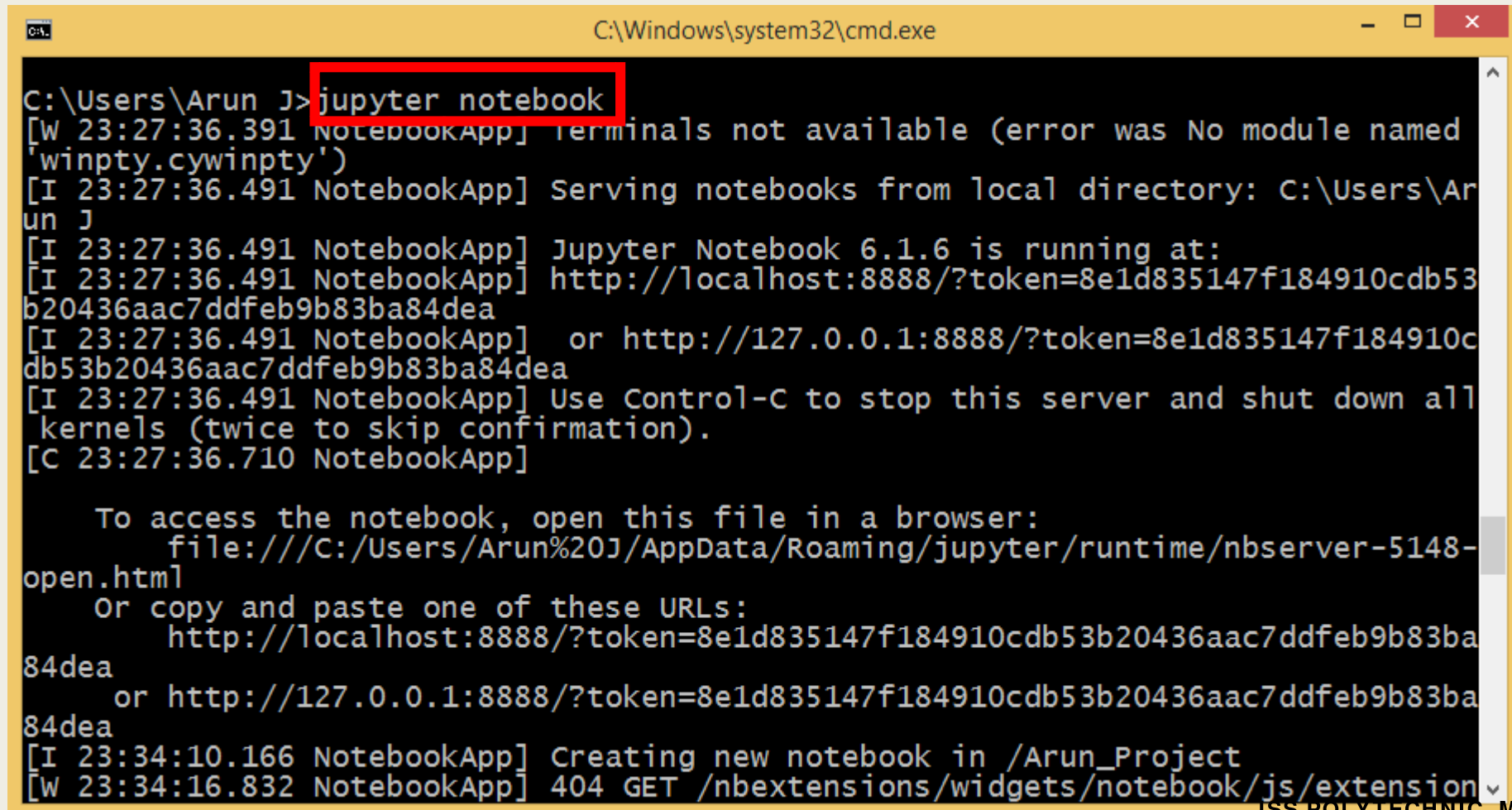


```
C:\Windows\system32\cmd.exe

C:\Users\Arun J>pip install notebook
Collecting notebook
  Downloading notebook-6.1.6-py3-none-any.whl (9.5 MB)
    |-----| 9.5 MB 181 kB/s
Collecting nbformat
  Downloading nbformat-5.0.8-py3-none-any.whl (172 kB)
    |-----| 172 kB 1.3 MB/s
Collecting pyzmq>=17
  Downloading pyzmq-20.0.0-cp39-cp39-win_amd64.whl (699 kB)
    |-----| 699 kB 1.3 MB/s
Collecting tornado>=5.0
  Downloading tornado-6.1-cp39-cp39-win_amd64.whl (422 kB)
    |-----| 422 kB 1.7 MB/s
Collecting ipykernel
  Downloading ipykernel-5.4.2-py3-none-any.whl (119 kB)
    |-----| 119 kB 1.7 MB/s
Collecting Send2Trash
  Downloading Send2Trash-1.5.0-py3-none-any.whl (12 kB)
Collecting jupyter-client>=5.3.4
  Downloading jupyter_client-6.1.7-py3-none-any.whl (108 kB)
    |-----| 108 kB 1.3 MB/s
Collecting argon2-cffi
  Downloading argon2_cffi-20.1.0-cp39-cp39-win_amd64.whl (42 kB)
    |-----| 42 kB 116 kB/s
```

How to Run Jupyter Notebook

In command prompt give the command as: **jupyter notebook**



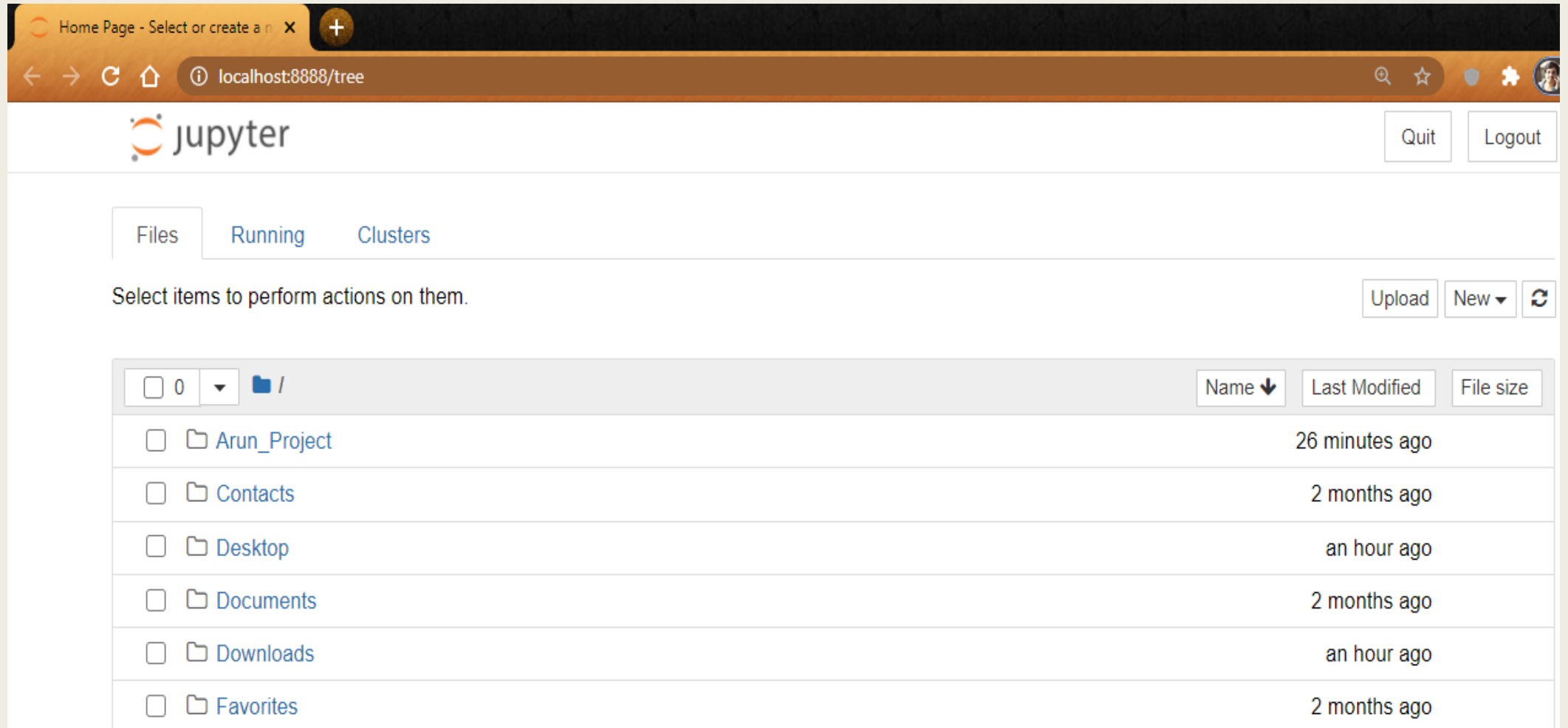
```
C:\Windows\system32\cmd.exe

C:\Users\Arun J>jupyter notebook
[W 23:27:36.391 NotebookApp] terminals not available (error was No module named
'winpty.cython')
[I 23:27:36.491 NotebookApp] Serving notebooks from local directory: C:\Users\Arun J
[I 23:27:36.491 NotebookApp] Jupyter Notebook 6.1.6 is running at:
[I 23:27:36.491 NotebookApp] http://localhost:8888/?token=8e1d835147f184910cdb53b20436aac7ddfeb9b83ba84dea
[I 23:27:36.491 NotebookApp] or http://127.0.0.1:8888/?token=8e1d835147f184910cdb53b20436aac7ddfeb9b83ba84dea
[I 23:27:36.491 NotebookApp] Use Control-C to stop this server and shut down all
kernels (twice to skip confirmation).
[C 23:27:36.710 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/Arun%20J/AppData/Roaming/jupyter/runtime/nbserver-5148-
open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=8e1d835147f184910cdb53b20436aac7ddfeb9b83ba84dea
or http://127.0.0.1:8888/?token=8e1d835147f184910cdb53b20436aac7ddfeb9b83ba84dea
[I 23:34:10.166 NotebookApp] Creating new notebook in /Arun_Project
[W 23:34:16.832 NotebookApp] 404 GET /nbextensions/widgets/notebook/js/extension
```


Jupyter Notebook window

Jupyter Notebook interface:



Creating a Folder in Jupyter Notebook

- To Create a Folder in Jupyter Notebook Click on **NEW** Button → Select “**Folder**”
- Untitled Folder will be created. Rename it By clicking **RENAME** button



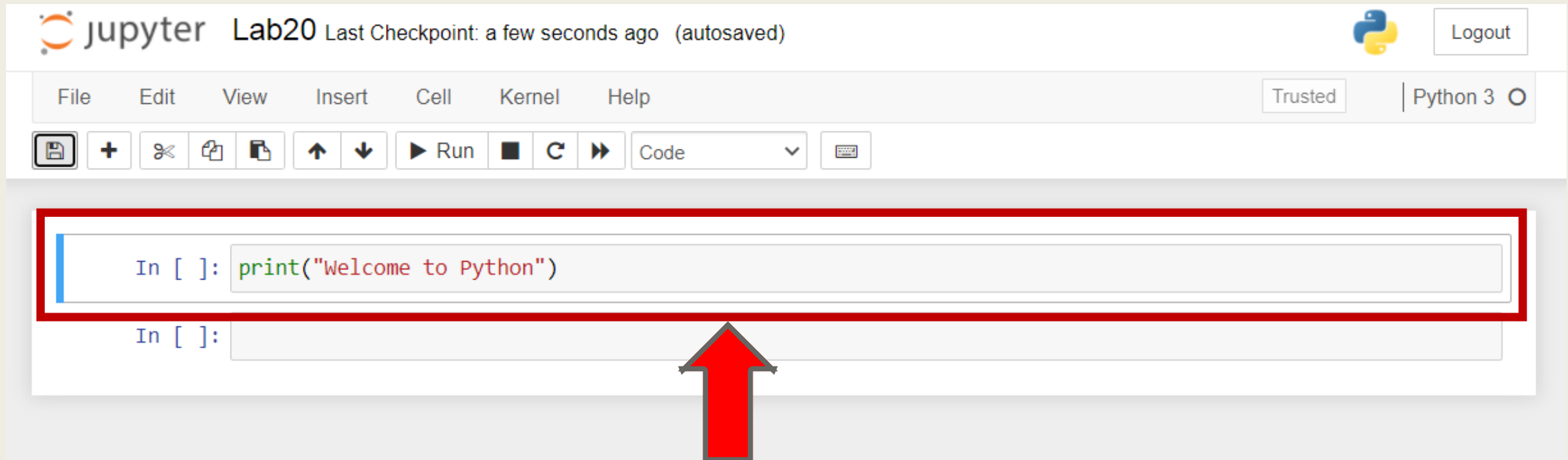
Creating a Python file in Jupyter Notebook

- To Create a File in Jupyter Notebook Click on **NEW** Button → Select “**Python 3**”



Where to write Python code on Jupyter Notebook

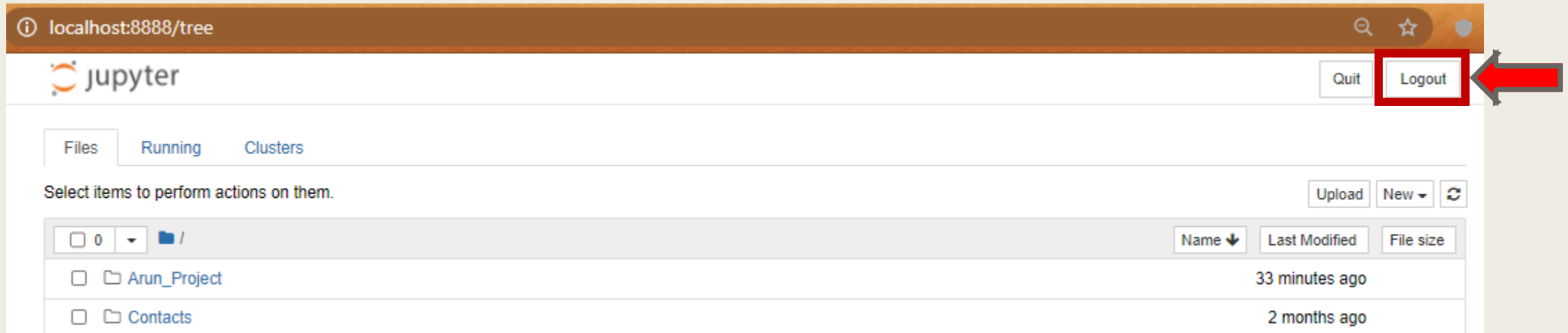
When we open a python file in jupyter notebook, we will notice that it contains a **cell**



Type Python Code here

Shutting down Jupyter Notebook

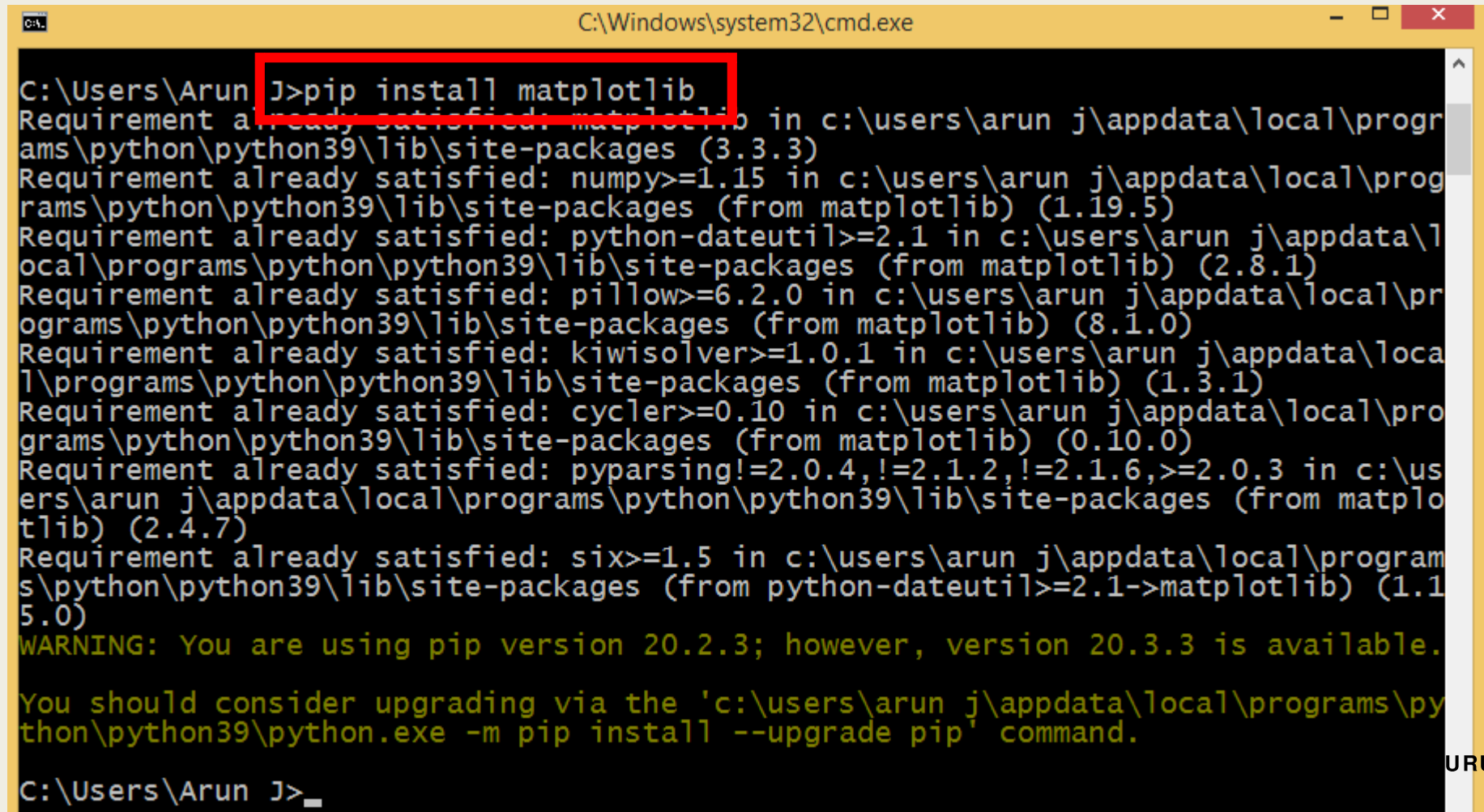
- Click on **Logout** button on Jupyter Notebook



- Press Ctrl+c in command prompt

Installing matplotlib Package

- To Execute Lab Cycle 25 & 26 (Graphs) This Package has to installed.
- In command prompt type: **pip install matplotlib**



```
C:\Windows\system32\cmd.exe

C:\Users\Arun J>pip install matplotlib
Requirement already satisfied: matplotlib in c:\users\arun j\appdata\local\programs\python\python39\lib\site-packages (3.3.3)
Requirement already satisfied: numpy>=1.15 in c:\users\arun j\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (1.19.5)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\arun j\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (2.8.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\arun j\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (8.1.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\arun j\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cyclor>=0.10 in c:\users\arun j\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\users\arun j\appdata\local\programs\python\python39\lib\site-packages (from matplotlib) (2.4.7)
Requirement already satisfied: six>=1.5 in c:\users\arun j\appdata\local\programs\python\python39\lib\site-packages (from python-dateutil>=2.1->matplotlib) (1.15.0)
WARNING: You are using pip version 20.2.3; however, version 20.3.3 is available.
You should consider upgrading via the 'c:\users\arun j\appdata\local\programs\python\python39\python.exe -m pip install --upgrade pip' command.

C:\Users\Arun J>
```

INTRODUCTION TO PYTHON PROGRAMMING



Why Programming ?

- Computer has to be given instructions to perform a task
- **Program** : a set of instructions given to a computer
- Program to be written in a formal language
- Programming language has a set of rules(syntax)
- Examples of programming languages – C , Java , Python

Why Python?

- User friendly and easy for beginners to learn
- It is free(open source)
- Commonly used trending language
- Used in Artificial Intelligence, Game development and Image Processing etc.

History of python

- First released in 1991
- Created by Dutch Programmer Guido Van Rossum
- Van Rossum wanted name to be short, unique and slightly mysterious.
- Inspired by his favorite British Comedy Group called Monty Python



Guido Van Rossum

Keywords/Reserved words

- Words with a special meaning in a programming language
- Are predefined
- Cannot be used as variable names, function names etc.
- 33 keywords
- Usually written in lowercase
- Except True/False/None is written in camel case (First letter Capital)

Keywords/Reserved words (Contd..)

Ex:

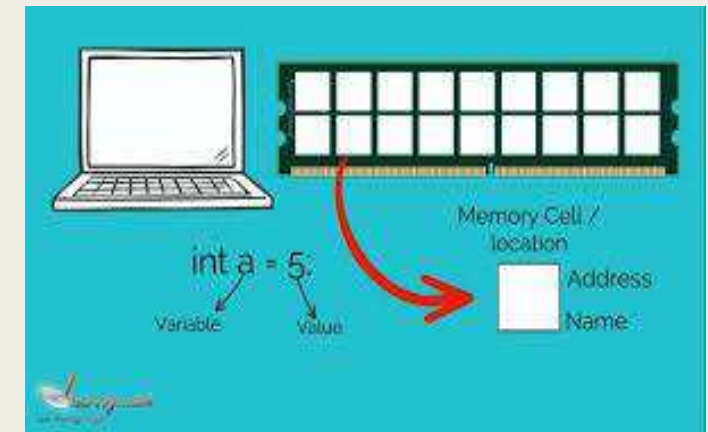
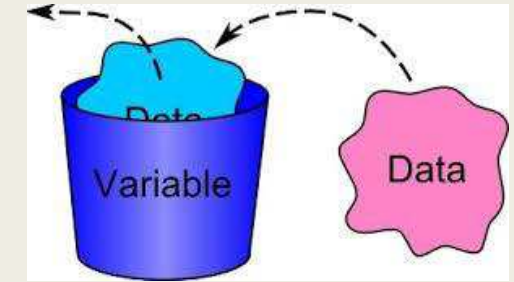
- *if*
- *else*
- *for*
- *while*

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

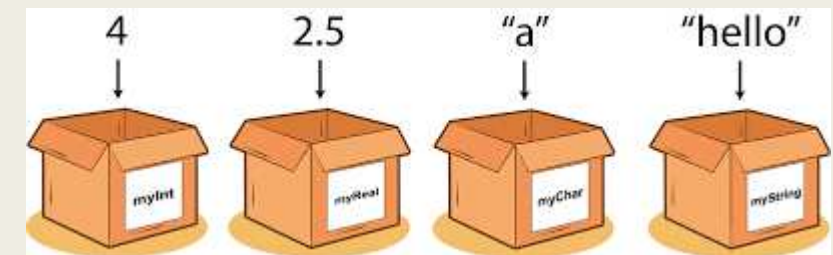
What is a Variable ?

Variables are like containers to store data

- Data is stored inside memory in a computer.
- Each memory location has an address.
- Difficult to remember the address
- Hence a name will be given to a memory location.
- These names are called Variables.



Use different variables for different data



Rules for Variables

- A variable name must start with alphabets or underscore (a-z / A-Z or _).
- A variable name cannot start with a number.
- A variable name can only contain alpha-numeric characters and underscores (a-z/A-Z, 0-9, and _)
- Variable names are case-sensitive (age, AGE and aGe are three different variables)
- Space is not allowed
- Cannot be keywords

Ex: `x = 5`
`x_y = "India"`
`Num1=34.888`

Question



Which of the following are valid Python variable names:

a) Emp_Salary

b) age@

c) Num2

d) 2Num

e) Emp Salary

f) _Empsalary

Answer



Which of the following are valid Python variable names:

- a) **Emp_Salary**
- b) **age@** (@ not allowed)
- c) **Num2**
- d) **2Num** (cannot start with number)
- e) **Emp Salary** (Space not allowed)
- f) **_Empsalary**

Printing / Displaying Output

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

Hello

#printing string

```
x=100  
print(x)
```

100

#printing variable value

Arithmetic Operators

Operator	Name	Example
+	Addition	$4+2=6$
-	Subtraction	$4-2=2$
*	Multiplication	$4*2=8$
/	Division(Quotient)	$4/2=2$
%	Modulus(Remainder)	$4\%2=0$
**	Power	$4**2=8$

Data Types

Name	Description
Integer	Whole numbers. Ex: 8, -70, 4567
Floating point	Numbers with decimal point. Ex: 6.77 , -7.80
String	Set of characters. (a-z, 0-9, any special character) Ex: "hello" , 'India' , "r5@39\$_2" , '087'

Comments

- Comments are added to provide explanation of the program
- Used to give information about the Python program.
- Comments can be used to make the code more readable.
- Not executed by computer
- Symbol used for comment is #

Example:

```
#Write Python program to add 2 numbers
```

```
#created on 11.01.2021
```

```
num1=100
```

```
num2=200
```

```
sum1=num1+num2
```

```
print("The sum of 2 integers is ", sum1)
```

if statement / Decision making

if statement used to execute based on a condition being satisfied

Examples:

- If age of a person is ≥ 18 ; Can Vote otherwise cannot Vote
if age ≥ 18
Can Vote
else
Can't Vote
- if PIN correct I can do transaction in ATM
- if student marks ≥ 35 ; PASS else FAIL

if statement

Python has following Comparison Operators

Operators	Meaning	Example	Result
<	Less than	5<2	False
>	Greater than	5>2	True
<=	Less than or equal to	5<=2	False
>=	Greater than or equal to	5>=2	True
==	Equal to	5==2	False
!=	Not equal to	5!=2	True

They are used in "if statements" and “loops”.

How to use if statement

```
marks = 33
if marks >= 35 :
    print("PASS")
else :
    print("FAIL")
```



If statement, without indentation gives an error

```
marks = 33
if marks >= 35 :
print("PASS")
else :
print("FAIL")
```



Loops

- In some situations few statements may have to be repeated several times – Loops.
- Printing 'welcome' 25 times.

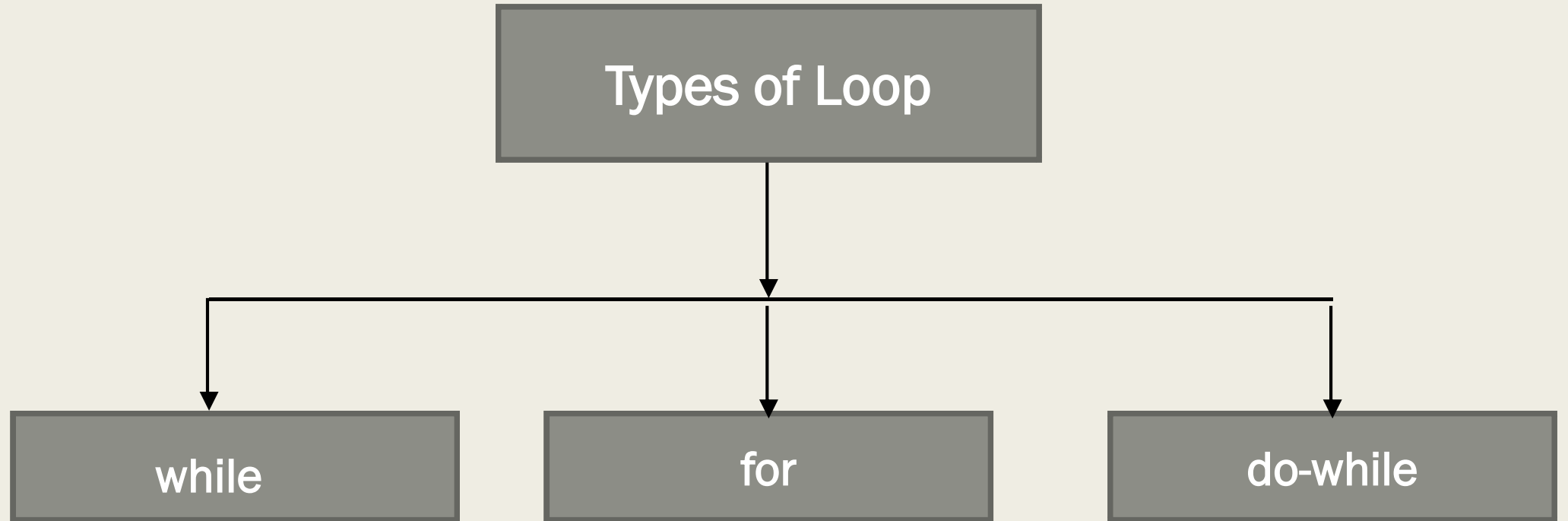
```
print("welcome")  
print("welcome")  
print("welcome")  
print("welcome")  
print("welcome")  
  
- - - -  
  
- - - -
```



```
Loop (25 times)  
print("welcome")
```

- Performing Scientific experiments for many iterations.
- Sum of 1st 10 natural numbers (1+2+3+4+5+6+7+8+9+10)

Types of loop in python



While Loop

Repeats a statement or group of statements while a given condition is TRUE. It tests the condition before executing the loop body.

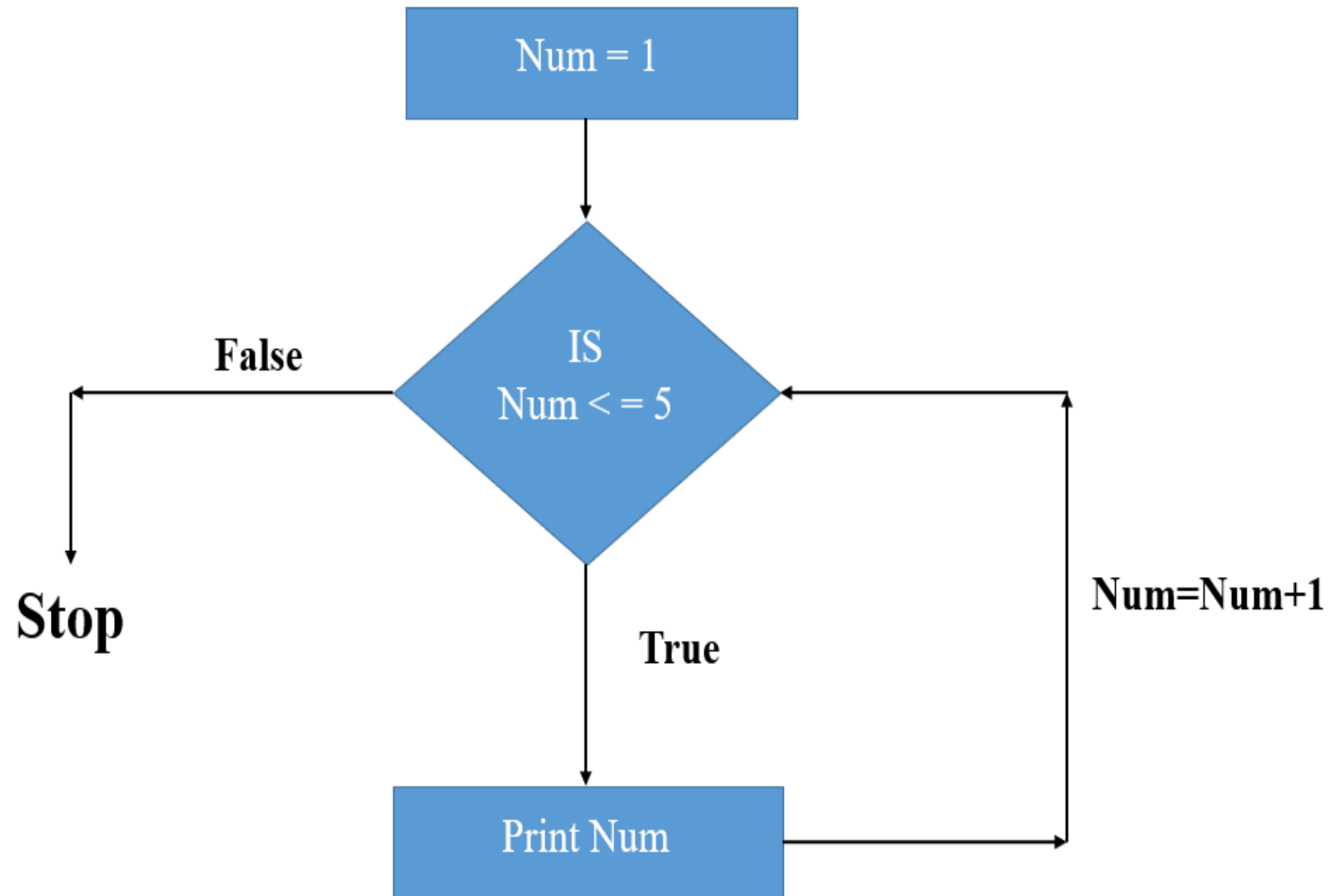
How Loops work

Example: To print 1 to 5

```
Num = 1  
while Num <= 5:  
    print (Num)  
    Num = Num + 1
```

OUTPUT

1 2 3 4 5



Array/List

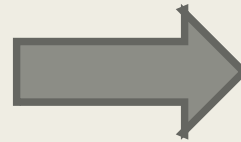
An array holds many data values in a single variable

```
student1 = "Ramesh"
```

```
student2 = "Suresh"
```

```
student3 = "Priya"
```

For 50 Students



Array Concept

```
students= ["Ramesh", "Suresh", "Priya", ... ..]
```

for Loop

A **for** loop is used for iterating over a list of items

With the **for** loop we can execute a set of statements, once for each item in a list

Example. Print all items in the **list**, one by one

for loop for One List

```
fruits = ["Apple", "Mango", "Cherry"]  
for x in fruits:  
    print(x)
```

OUTPUT:

```
Apple  
Mango  
Cherry
```

for loop for two lists

```
name = ["Smith", "John", "Mary", "Dave"]  
marks = [40, 88, 20, 65]  
for (i, j) in zip(name, marks):  
    print(i, j)
```

OUTPUT:

```
Smith 40  
John 88  
Mary 20  
Dave 65
```

Working With Files

- Files are used to store data in computer memory permanently.
- The data can be used later - Reusability
- Many file handling operations can be performed in python programming

Ex:

- ✓ Reading from a file
- ✓ Writing to a file

Reading Data from a File

Steps:

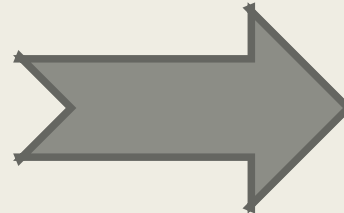
- Opening the file - Use the `open()` function
Ex: `file = open("student", "r")`

`r` – read
`student` – file name
- Reading from file - Use the `readlines()` function
Ex: `file.readlines()`
- Closing a file - Use `close()` function
Ex: `file.close()`

Functions – Making things Easy

To find variance without function

```
import math
sample=(5,8,10,8,8,5)
sum=0
var=0
for x in sample:
    sum=sum+x
    mean=sum/6
print("mean=",mean)
for x in sample:
    var=var+(x-mean)**2
var=var/5
print("variance = ",var)
```



To find variance with function

pvariance(sample data)

Functions

- Used to make a complex task easy
- Group of related statements that performs a specific task.
- Help us divide a big program into a simple form
- Python has many inbuilt functions.

Ex:

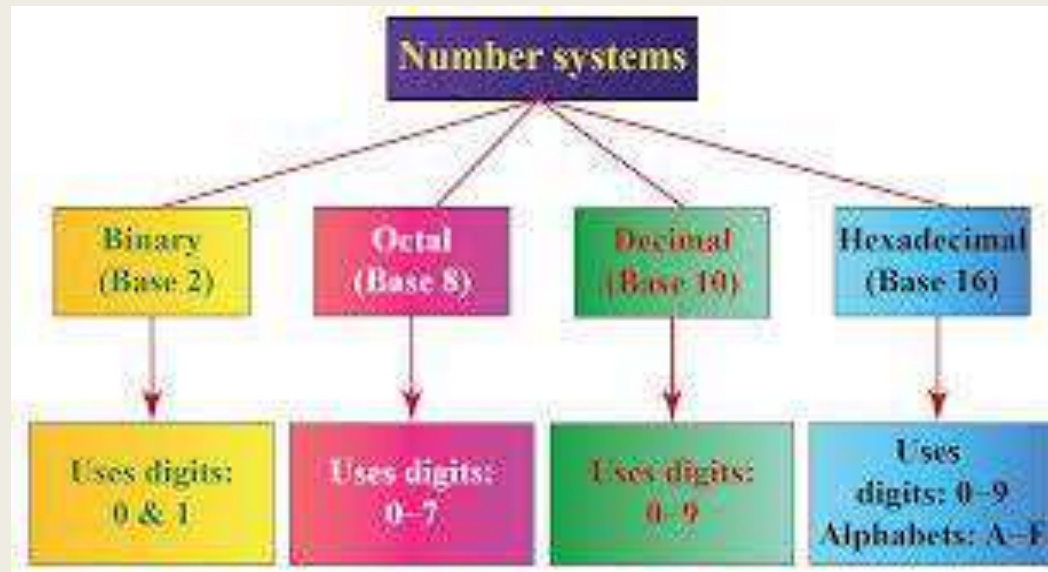
To find variance	-	pvariance(data)
To draw a pie chart	-	pie(data)
To draw a bar chart	-	bar(data)

Number System

Python offers inbuilt functions to handle number system.

Functions for conversion between number systems



- **bin(dec)** -> function to convert decimal to binary
- **oct(dec)** -> function to convert decimal to octal
- **hex(dec)** -> function to convert decimal to hexadecimal





To find Standard deviation and Variance

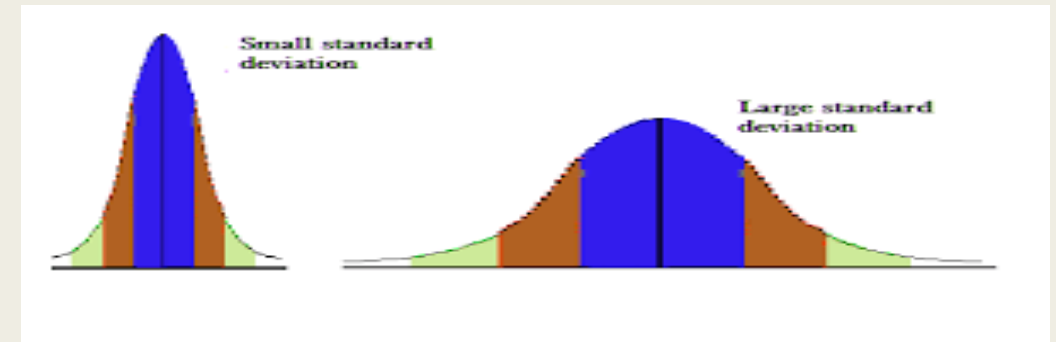
- Variance and standard deviation are related to statistics,
- It measures how much the data in a group is spread out from the mean value.

Population Standard Deviation Formula


$$\sigma = \sqrt{\frac{\sum (x - \mu)^2}{N}}$$


Population Variance Formula


$$\sigma^2 = \frac{\sum_{i=1}^n (x_i - \mu)^2}{N}$$




Statistics Library

- Statistics Library has many inbuilt functions.

Ex:

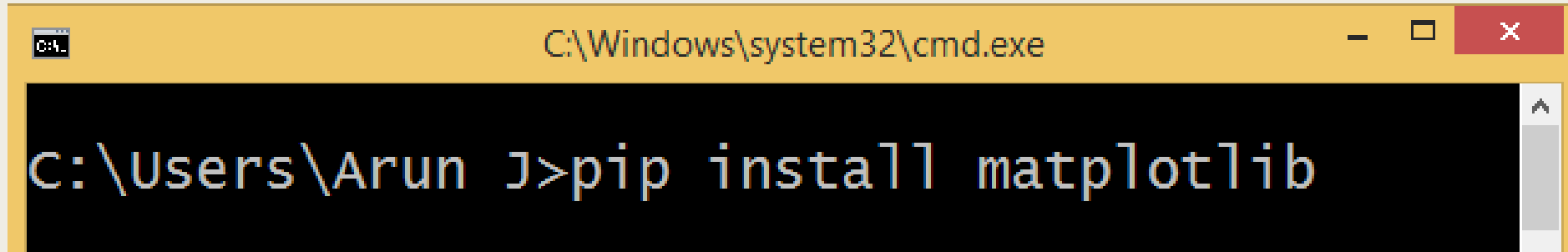
```
statistics.pvariance(data)  
statistics.pstdev(data)  
statistics.mean(data)
```

- In order to work with statistical function

```
import statistics
```

Matplotlib Library

- **Matplotlib** is a plotting library for **Python** programming
- An amazing visualization library in **Python** for 2D plots of arrays
- Install matplotlib using command prompt using below command.

A screenshot of a Windows Command Prompt window. The title bar is yellow and contains the text 'C:\Windows\system32\cmd.exe'. The main area has a black background with white text. The command 'C:\Users\Arun J>pip install matplotlib' is entered at the prompt. A vertical scrollbar is visible on the right side of the command area.

```
C:\Windows\system32\cmd.exe  
C:\Users\Arun J>pip install matplotlib
```

- Use below statement in your program to use matplotlib library

import matplotlib.pyplot as plt

LAB CYCLE

PROGRAMS

Lab Cycle

#19. Write a python program to convert Decimal to Binary, Octal and Hexadecimal.

```
dec=15
b=bin(dec)
o=oct(dec)
h=hex(dec)
print("Decimal number is : ",dec)
print("Binary value is : ",b)
print("Octal value is : ",o)
print("Hexadecimal value is : ",h)
```

OUTPUT:

```
Decimal number is : 15
Binary value is : 0b1111
Octal value is : 0o17
Hexadecimal value is : 0xf
```

Lab Cycle

#20. Write a python program to add 2 integers and two strings and print the result

```
num1=100
num2=200
sum1=num1+num2
print("The addition of two integers is ", sum1)

str1="JSS "
str2="Polytechnic"
sum2=str1+str2
print("The addition of two strings is ", sum2)
```

OUTPUT:

```
The addition of two integers is  300
The addition of two strings is  JSS Polytechnic
```

Lab Cycle

#21. Write a Python program to find the sum of first 10 natural numbers

```
num = 1
sum = 0
while(num <=10):
    sum = sum + num
    num = num + 1
print("The sum of first 10 natural numbers is: ", sum)
```

OUTPUT:

```
The sum of first 10 natural numbers is: 55
```

Lab Cycle

#22. Write a python program to find whether the given number is ODD or EVEN

```
num=101
if num % 2 == 0 :
    print("The Number is EVEN")
else :
    print("The Number is ODD")
```

OUTPUT:

```
The Number is ODD
```

Lab Cycle

#23. Python program to find the variance and standard deviation for the given data.

```
import statistics

sample = [130, 137, 136, 142, 135]

var=statistics.pvariance(sample)
print("The Variance is ", var)

sd=statistics.pstdev(sample)
print("The Standard Deviation is ", sd)
```

OUTPUT:

```
The Variance is  14.8
The Standard Deviation is  3.847076812334269
```

Lab Cycle

#24. Write a python program to display student marks from the record.

Student

Student Name	:	Student Marks
Ramesh	:	90
Suresh	:	95
Priya	:	100

```
file=open("student","r")
lines=file.readlines()

for x in lines:
    print(x)
file.close()
```

OUTPUT:

Student Name	:	Student Marks
Ramesh	:	90
Suresh	:	95
Priya	:	100

Lab Cycle

#25. Write a python program to create a labeled bar graph using matplotlib.pyplot

```
import matplotlib.pyplot as plt
```

```
x=["Ramesh", "Priya", "Suresh", "Akshay","Roopa"]
```

```
y=[74,23,80,34,56]
```

```
plt.title("Students Marks Bar Graph")
```

```
plt.xlabel("Student Name")
```

```
plt.ylabel("Marks")
```

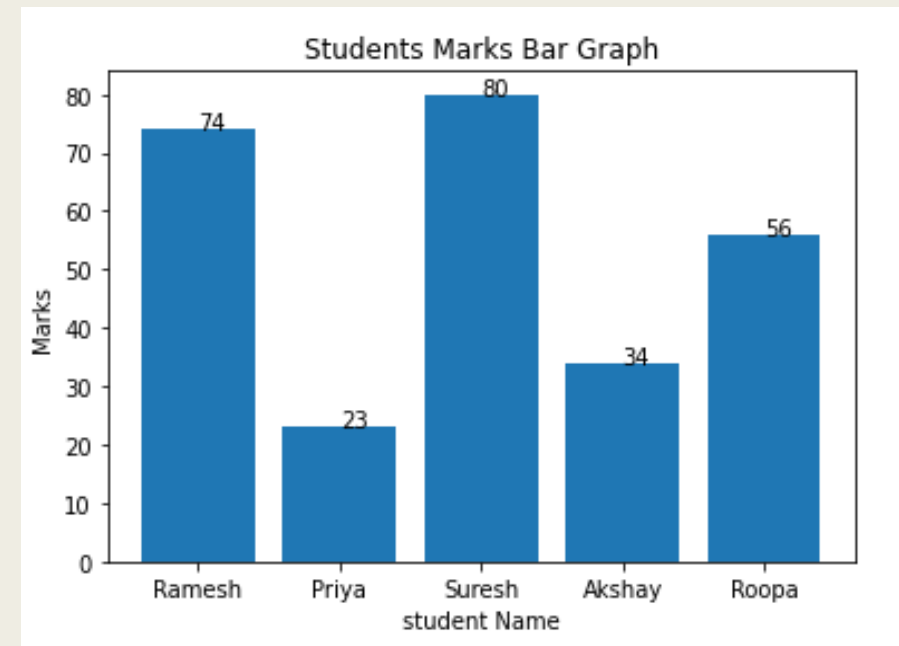
```
plt.bar(x,y)
```

```
for (i,j) in zip(x,y):
```

```
    plt.annotate(j,(i,j))
```

```
plt.show()
```

OUTPUT:



Lab Cycle

#26 Write a python program to create a labeled pie graph using matplotlib.pyplot

```
import matplotlib.pyplot as plt
```

```
x=["Ramesh", "Priya", "Suresh", "Akshay","Roopa"]
```

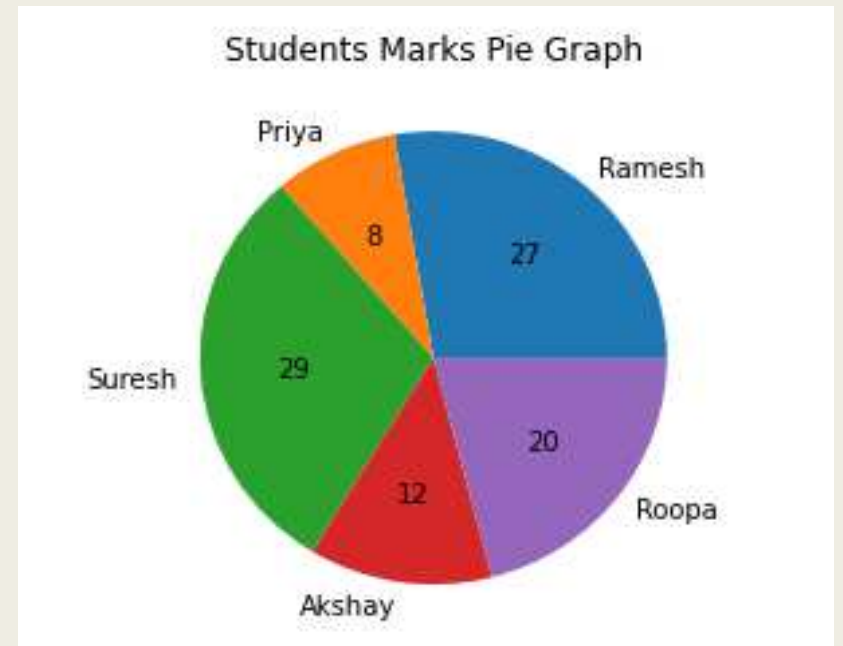
```
y=[74,23,80,34,56]
```

```
plt.title("Students Marks Pie Graph")
```

```
plt.pie(y,labels=x,autopct='%d')
```

```
plt.show()
```

OUTPUT:



Bibliography

- College Website Link

<http://www.jsspolytechnic.org>

- To Download Python

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

- Older Versions of python

<https://www.python.org/downloads/release/python-386/>

- Documentation on Jupyter notebook

<https://www.codecademy.com/articles/how-to-use-jupyter-notebooks>

- Documentation on Python

<https://www.tutorialspoint.com/python/index.htm>