



BASIC PYTHON SYNTAX

1. Print to Console

```
In [5]: print("Hello, World !")  
Hello, World !
```

2. Variable Assignment:

```
In [5]: print("Hello, World !")  
Hello, World !
```

```
In [7]: a = "Pushkar sonkar "  
a
```

```
Out[7]: 'Pushkar sonkar '
```

3. Commenting:

```
In [10]: # This is single line comment
```

4. Multiline Comment:

```
In [20]: ''' This is a multi-line comment! '''  
Out[20]: ' This is a multi-line comment! '
```

5. Input from User:

```
In [25]: name = input('enter your name:')
```

6. Check Data Types:

```
In [28]: type(a)  
Out[28]: str
```

7. Type Casting:

```
In [31]: int("10"),float("10.5"),str(100)
```

```
Out[31]: (10, 10.5, '100')
```

DATA STRUCTURES

1. List (Array):

```
In [37]: my_list = [1,2,3,4,5]  
my_list
```

```
Out[37]: [1, 2, 3, 4, 5]
```

2. Access List item:

```
In [43]: my_list[3]
```

```
Out[43]: 4
```

3. List Slicing:

```
In [46]: my_list[1:4]
```

```
Out[46]: [2, 3, 4]
```

4. Add Item to List:

```
In [64]: my_list.append(6)  
my_list.append(3)  
my_list.append(2)
```

5. Remove item from List:

```
In [66]: my_list.remove(2)  
my_list
```

```
Out[66]: [1, 4, 5, 6, 6, 3, 6, 3]
```

6. Tuple:

```
In [70]: my_Tuple = (1,2,3,4)  
my_Tuple
```

```
Out[70]: (1, 2, 3, 4)
```

7. Set:

```
In [73]: my_Set = {1,2,3,4}  
my_Set
```

```
Out[73]: {1, 2, 3, 4}
```

8. Dictionary:

```
In [78]: dict ={'key1':'val1','key2':'val2'}  
dict
```

```
Out[78]: {'key1': 'val1', 'key2': 'val2'}
```

9. Access Dictionary Value:

```
In [81]: dict['key1']
```

```
Out[81]: 'val1'
```

10. Add Key-Value Pair:

```
In [84]: dict['key3']='val3'  
dict
```

```
Out[84]: {'key1': 'val1', 'key2': 'val2', 'key3': 'val3'}
```

CONTROL FLOW

1. If Statement:

```
In [90]: x=11  
if x>10: print('x is greater than 10')
```

```
x is greater than 10
```

2. If-Else Statement:

```
In [111...]: y=6  
if y>10 : print('y is high')  
else : print('y is low ')
```

y is low

3. Elif Statement:

```
In [114...]: y=6  
if y>10 : print('y is high')  
elif y>5 : print('y is avg')  
else : print('y is low ')
```

y is avg

4. For Loop:

```
In [154...]: for i in range (5):  
    print(i)
```

0
1
2
3
4

5. While Loop:

```
In [142...]: x = 0  
while x < 10 :  
    x+=1  
    print(x)
```

1
2
3
4
5
6
7
8
9
10

6. Break:

```
In [151...]: for i in range (5):
              if i==3:
                  break
              print(i)
```

```
0
1
2
```

7. Continue:

```
In [159...]: for i in range (5):
              if i==3:
                  continue
              print(i)
```

```
0
1
2
4
```

FUNCTIONS

1. Define Functions:

```
In [172...]: def my_function (x):
              print('hello from function!',x)

my_function(10)
```

```
hello from function! 10
```

2. Functions with Parameters:

```
In [175...]: def greet(name):
              print("Hello, ", name)

greet("Pushkar")
```

```
Hello, Pushkar
```

3. Return Value From Function:

```
In [179...]: def add(a, b):
```

```
    return a + b

result = add(5, 3)
print("Sum is:", result)
```

Sum is: 8

4. Lambda Function:

A lambda function is a small, anonymous function in Python (doesn't need a name). It can have any number of arguments but only one expression.

```
In [183... # lambda arguments: expression
      square = lambda x: x**2
      print(square(5))
```

25

String Manipulation

1. Find Substring

```
In [189... "Hello, World!".find("World")
```

Out[189... 7

2. Replace Substring

```
In [192... "Hello, World!".replace("World", "Python")
```

Out[192... 'Hello, Python!'

3. Split String

```
In [195... "Hello, World!".split(",")
```

Out[195... ['Hello', ' World!']

FILE HANDLING

1. Open a File

```
In [ ]: file = open("example.txt", "r")
```

2. Read File

```
In [ ]: content = file.read()
```

3. Read Line by Line

```
In [ ]: lines = file.readlines()
```

4. Write to a File

```
In [209...]: file = open("example.txt", "w")
file.write("Hello, World!")
```

```
Out[209...]: 13
```

5. Close a File

```
In [216...]: file.close()
```

LIST COMPREHENSION

1. Basic List Comprehension

```
In [223...]: squares = [x**2 for x in range(5)]
squares
```

```
Out[223...]: [0, 1, 4, 9, 16]
```

2. List Comprehension with Condition

```
In [228...]: evens = [x for x in range(10) if x % 2 == 0]
evens
```

```
Out[228... [0, 2, 4, 6, 8]
```

ERROR HANDLING

1. Try-Except Block

```
In [234... 
```

```
try:  
    with open("example.txt", "r") as file:  
        content = file.read()  
        print(content)  
except FileNotFoundError:  
    print("File not found")
```

```
Hello, World!
```

2. Finally Block

```
In [238... 
```

```
try:  
    x = 10 / 0  
except ZeroDivisionError:  
    print("Error: Division by zero")  
finally:  
    print("This runs always")
```

```
Error: Division by zero  
This runs always
```

WORKING WITH LIBRARIES

1. Importing a Library

```
In [242... 
```

```
import math
```

2. Using a library function

```
In [245... 
```

```
print(math.sqrt(16))
```

```
4.0
```

3. Install a Library (run in terminal, not inside Python)

```
In [248... pip install pandas  
Requirement already satisfied: pandas in /opt/anaconda3/lib/python3.12/site-packages (2.2.2)  
Requirement already satisfied: numpy>=1.26.0 in /opt/anaconda3/lib/python3.12/site-packages (from pandas) (1.26.4)  
Requirement already satisfied: python-dateutil>=2.8.2 in /opt/anaconda3/lib/python3.12/site-packages (from pandas) (2.9.0.post0)  
Requirement already satisfied: pytz>=2020.1 in /opt/anaconda3/lib/python3.12/site-packages (from pandas) (2024.1)  
Requirement already satisfied: tzdata>=2022.7 in /opt/anaconda3/lib/python3.12/site-packages (from pandas) (2023.3)  
Requirement already satisfied: six>=1.5 in /opt/anaconda3/lib/python3.12/site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)  
Note: you may need to restart the kernel to use updated packages.
```

4. Import Specific Function

```
In [253... from math import sqrt  
print(sqrt(25))
```

5.0

NUMPY FOR NUMERICAL OPERATIONS

```
In [256... import numpy as np
```

1. Create NumPy Array

```
In [273... arr = np.array([1,2,3,4,5])  
arr
```

Out[273... array([1, 2, 3, 4, 5])

2. Array Reshaping

```
In [276... reshaped = arr.reshape(5, 1)  
print(reshaped)
```

```
[[1]
 [2]
 [3]
 [4]
 [5]]
```

3. Array Operations

```
In [279... arr2 = np.array([10, 20, 30, 40, 50])
      print(arr + arr2)
```

[11 22 33 44 55]

4. Array Slicing

```
In [282... print(arr[1:4])
```

[2 3 4]

5. Array Statistics

```
In [285... print("Mean:", np.mean(arr))
      print("Median:", np.median(arr))
      print("Std Dev:", np.std(arr))
```

Mean: 3.0
Median: 3.0
Std Dev: 1.4142135623730951

PANDAS FOR DATA HANDLING

```
In [288... import pandas as pd
```

1. Create DataFrame

```
In [291... df = pd.DataFrame({"Name": ["Alice", "Bob"], "Age": [25, 30]})
      print(df)
```

	Name	Age
0	Alice	25
1	Bob	30

2. Read CSV File

```
In [298... df = pd.read_csv('IPL.csv')
```

df

Out[298...]

	match_id	date	venue	team1	team2	stage	toss_winner	team3	team4
0	1	March 26,2022	Wankhede Stadium, Mumbai	Chennai	Kolkata	Group	Kolkata		
1	2	March 27,2022	Brabourne Stadium, Mumbai		Delhi	Mumbai	Group		Delhi
2	3	March 27,2022	Dr DY Patil Sports Academy, Mumbai	Banglore	Punjab	Group	Punjab		
3	4	March 28,2022	Wankhede Stadium, Mumbai	Gujarat	Lucknow	Group	Gujarat		
4	5	March 29,2022	Maharashtra Cricket Association Stadium,Pune	Hyderabad	Rajasthan	Group	Hyderabad		
...
69	70	May 22,2022	Wankhede Stadium, Mumbai	Hyderabad	Punjab	Group	Hyderabad		
70	71	May 24,2022	Eden Gardens, Kolkata	Gujarat	Rajasthan	Playoff	Gujarat		
71	72	May 25,2022	Eden Gardens, Kolkata	Banglore	Lucknow	Playoff	Lucknow		
72	73	May 27,2022	Narendra Modi Stadium, Ahmedabad	Banglore	Rajasthan	Playoff	Rajasthan		
73	74	May 29,2022	Narendra Modi Stadium, Ahmedabad	Gujarat	Rajasthan	Final	Rajasthan		

74 rows × 20 columns

3. View Data

In [303...]

df.head()

Out[303...]

	match_id	date	venue	team1	team2	stage	toss_winner	toss
0	1	March 26,2022	Wankhede Stadium, Mumbai	Chennai	Kolkata	Group	Kolkata	
1	2	March 27,2022	Brabourne Stadium, Mumbai	Delhi	Mumbai	Group	Delhi	
2	3	March 27,2022	Dr DY Patil Sports Academy, Mumbai	Banglore	Punjab	Group	Punjab	
3	4	March 28,2022	Wankhede Stadium, Mumbai	Gujarat	Lucknow	Group	Gujarat	
4	5	March 29,2022	Maharashtra Cricket Association Stadium,Pune	Hyderabad	Rajasthan	Group	Hyderabad	

4. Basic Statistics

In [306...]

```
df.describe()
```

Out[306...]

	match_id	first_ings_score	first_ings_wkts	second_ings_score	second_in
count	74.000000	74.000000	74.000000	74.000000	74
mean	37.500000	171.121622	6.135135	158.540541	6
std	21.505813	29.048355	2.222699	29.299207	2
min	1.000000	68.000000	0.000000	72.000000	1
25%	19.250000	154.250000	5.000000	142.750000	4
50%	37.500000	169.500000	6.000000	160.000000	6
75%	55.750000	192.750000	8.000000	176.000000	8
max	74.000000	222.000000	10.000000	211.000000	10

5. Filter Data

In [311...]

```
print(df[df["first_ings_score"] > 25])
```

	match_id	date	venue			
0	1	March 26,2022	Wankhede Stadium, Mumbai			
1	2	March 27,2022	Brabourne Stadium, Mumbai			
2	3	March 27,2022	Dr DY Patil Sports Academy, Mumbai			
3	4	March 28,2022	Wankhede Stadium, Mumbai			
4	5	March 29,2022	Maharashtra Cricket Association Stadium, Pune			
..			
69	70	May 22,2022	Wankhede Stadium, Mumbai			
70	71	May 24,2022	Eden Gardens, Kolkata			
71	72	May 25,2022	Eden Gardens, Kolkata			
72	73	May 27,2022	Narendra Modi Stadium, Ahmedabad			
73	74	May 29,2022	Narendra Modi Stadium, Ahmedabad			
	team1	team2	stage			
	toss_winner	toss_decision	first_ings_score			
0	Chennai	Kolkata	Group	Kolkata	Field	131
1	Delhi	Mumbai	Group	Delhi	Field	177
2	Banglore	Punjab	Group	Punjab	Field	205
3	Gujarat	Lucknow	Group	Gujarat	Field	158
4	Hyderabad	Rajasthan	Group	Hyderabad	Field	210
..
69	Hyderabad	Punjab	Group	Hyderabad	Bat	157
70	Gujarat	Rajasthan	Playoff	Gujarat	Field	188
71	Banglore	Lucknow	Playoff	Lucknow	Field	207
72	Banglore	Rajasthan	Playoff	Rajasthan	Field	157
73	Gujarat	Rajasthan	Final	Rajasthan	Bat	130
	first_ings_wkts	second_ings_score	second_ings_wkts	match_winner		
0	5	133	4	Kolkata		
1	5	179	6	Delhi		
2	2	208	5	Punjab		
3	6	161	5	Gujarat		
4	6	149	7	Rajasthan		
..	
69	8	160	5	Punjab		
70	6	191	3	Gujarat		
71	4	193	6	Banglore		
72	8	161	3	Rajasthan		
73	9	133	3	Gujarat		
	won_by	margin	player_of_the_match	top_scorer	highscore	
0	Wickets	6	Umesh Yadav	MS Dhoni	50	
1	Wickets	4	Kuldeep Yadav	Ishan Kishan	81	
2	Wickets	5	Odean Smith	Faf du Plessis	88	
3	Wickets	5	Mohammed Shami	Deepak Hooda	55	
4	Runs	61	Sanju Samson	Aiden Markram	57	
..
69	Wickets	5	Harpreet Brar	Liam Livingstone	49	
70	Wickets	7	David Miller	Jos Buttler	89	
71	Runs	14	Rajat Patidar	Rajat Patidar	112	
72	Wickets	7	Jos Buttler	Jos Buttler	106	
73	Wickets	7	Hardik Pandya	Shubman Gill	45	
	best_bowling	best_bowling_figure				

```
0      Dwayne Bravo      3--20
1      Kuldeep Yadav    3--18
2      Mohammed Siraj     2--59
3      Mohammed Shami     3--25
4      Yuzvendra Chahal   3--22
...
69     Harpreet Brar      3--26
70     Hardik Pandya     1--14
71     Josh Hazlewood    3--43
72     Prasidh Krishna    3--22
73     Hardik Pandya     3--17
```

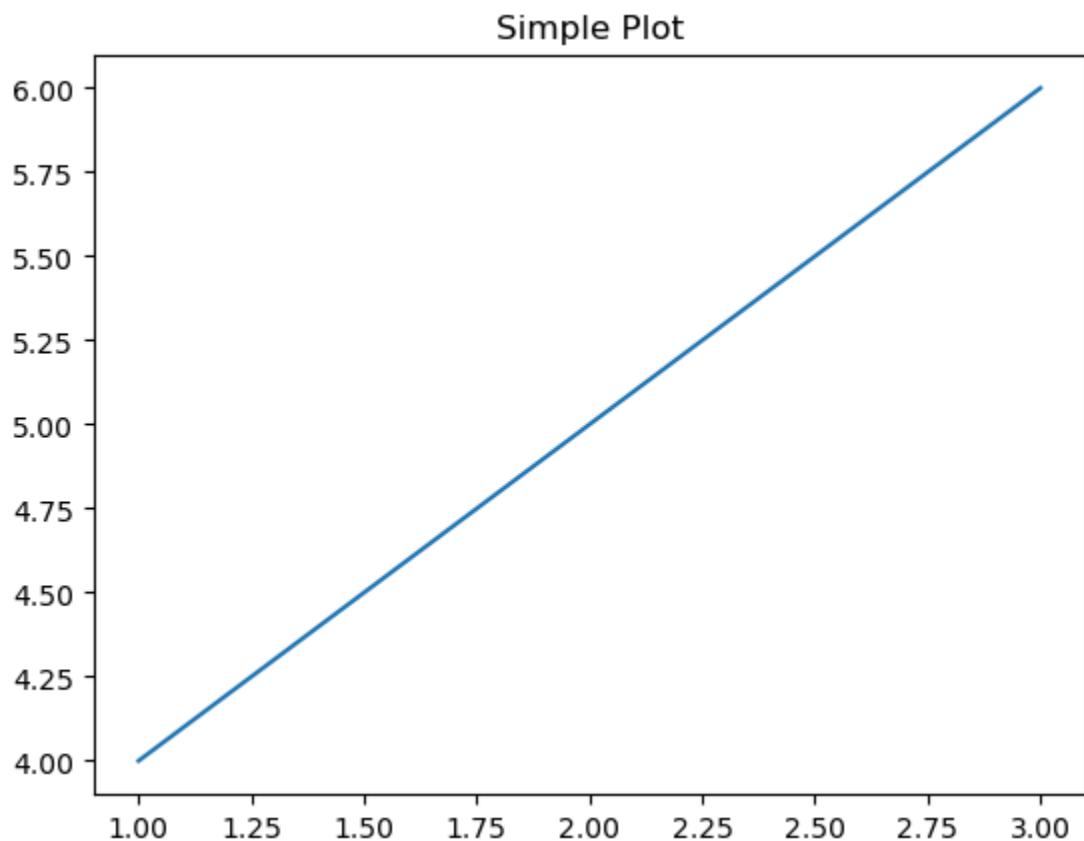
[74 rows x 20 columns]

MATPLOTLIB FOR PLOTTING

```
In [314]: import matplotlib.pyplot as plt
```

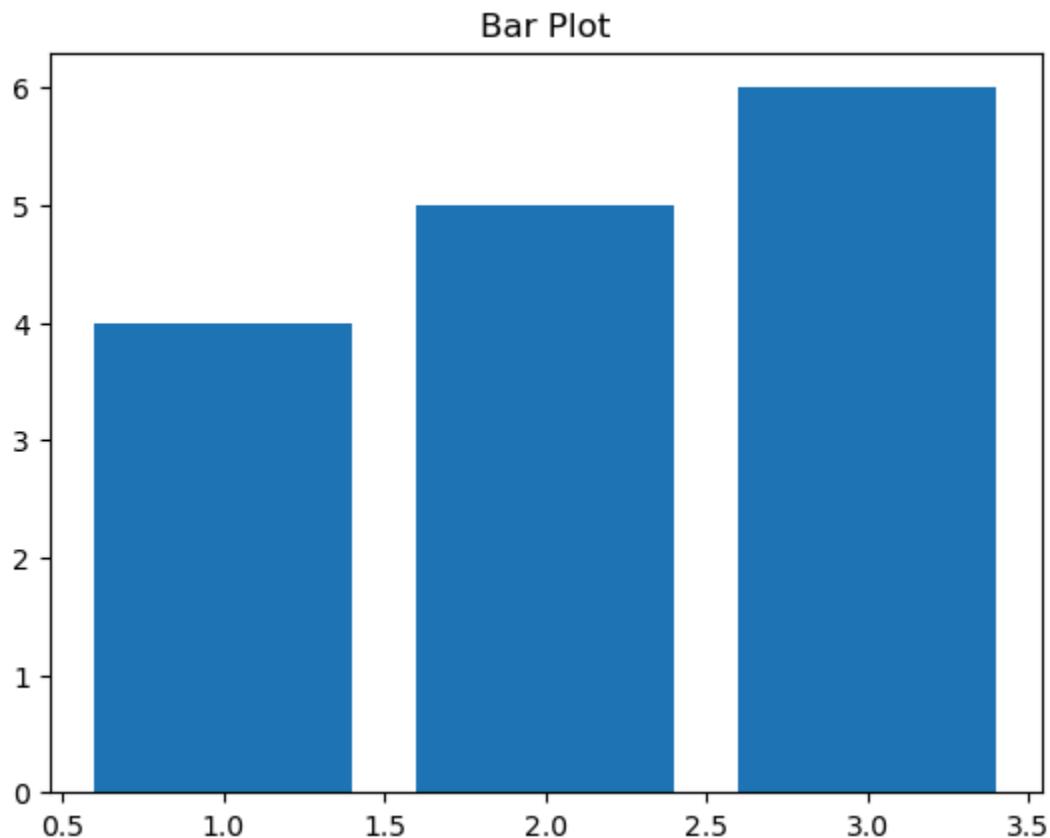
1. Simple Plot

```
In [317]: plt.plot([1, 2, 3], [4, 5, 6])
plt.title("Simple Plot")
plt.show()
```



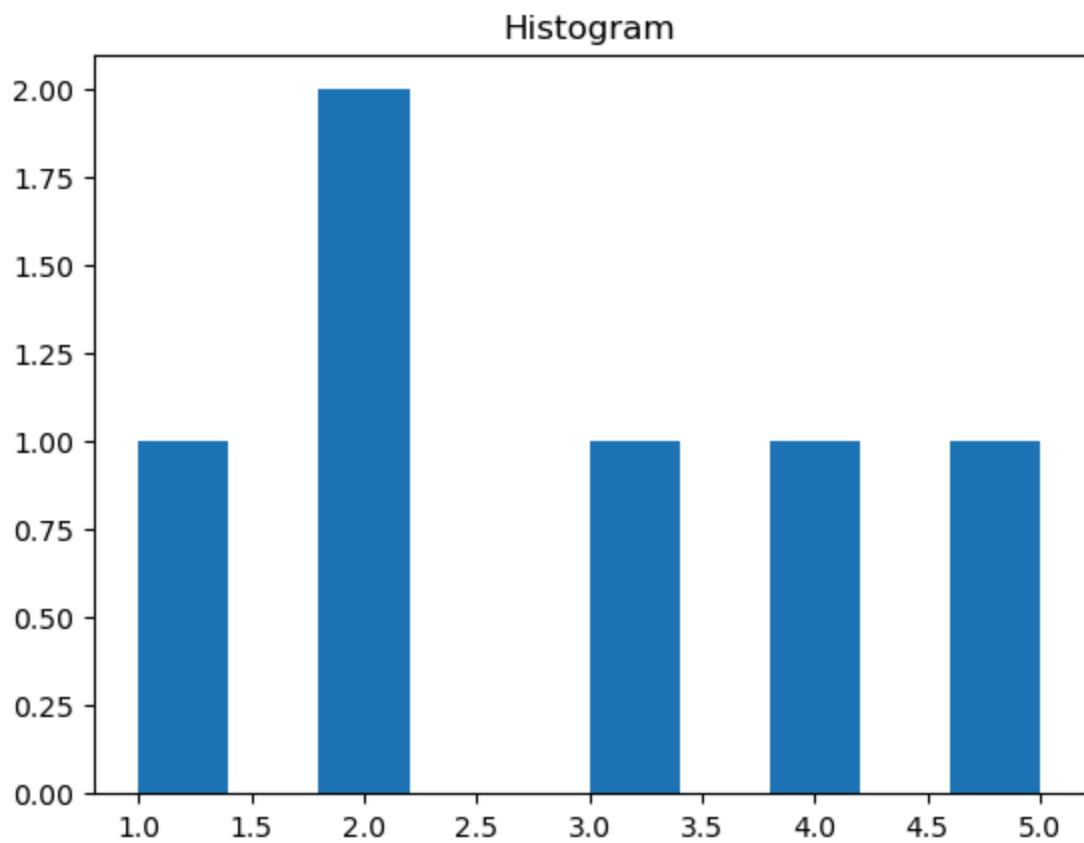
2. Bar Plot

```
In [320...]: plt.bar([1, 2, 3], [4, 5, 6])
plt.title("Bar Plot")
plt.show()
```



3. Histogram

```
In [323...]: plt.hist([1, 2, 2, 3, 4, 5])
plt.title("Histogram")
plt.show()
```



4. Scatter Plot

```
In [326]: plt.scatter([1, 2, 3], [4, 5, 6])
plt.title("Scatter Plot")
plt.show()
```

Scatter Plot

