1. **Difference between Errors and Exceptions**

----

* Errors are issues in the code that typically prevent the program from running (e.g., syntax errors).
* Exceptions are runtime issues that can be caught and handled to avoid program crashes.

* In Python, both errors and exceptions are related to problems that occur during the execution of a program,but they have different meanings and usage.

Error:-

----

* An error is a more general term for a problem that occurs in your program.
* It typically refers to issues that arise from incorrect code or unexpected conditions that can't be easily handled.

* For example, a SyntaxError occurs

when the Python code is not written correctly
(e.g., missing a parenthesis or a colon).

Ex:- print("Hello World"

Exception:

----

* An exception is a specific type of error that occurs during runtime (while the program is running).
* It represents an abnormal condition or event that can be caught and handled using a try-except block.
* Exceptions can be handled gracefully, allowing the program to recover or give a meaningful message rather than crashing.
* Examples of exceptions include ZeroDivisionError, FileNotFoundError, ValueError, etc.

Example for Exception Handling

—---

```
try:
    print("code start")

# if error occur the it goes in except block
except:
    print("an error occurs")
    print(1 / 0)


# final code in finally block
finally:
    print("welcome")
```

Output:-

—---------------------

```
code start
welcome
```

2) write any 4 python packages with examples

1. NumPy (Numerical Python)

NumPy is a powerful package used for scientific computing.
It provides support for arrays, matrices, and many mathematical functions.

Example:

```
import numpy as np

# Create a NumPy array
arr = np.array([1, 2, 3, 4, 5])

# Perform element-wise operations
arr_squared = arr ** 2
print(arr_squared)
```

Output:

```
[ 1  4  9 16 25]
```

NumPy is commonly used for operations involving arrays and matrices, as well as mathematical functions like linear algebra, random number generation, and more.

2. Pandas

Pandas is a package primarily used for data manipulation and analysis.
It provides data structures like Series (1D) and DataFrame (2D) that are ideal for handling structured data.

Example:

```
import pandas as pd

# Create a DataFrame
data = {
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Age': [25, 30, 35],
    'City': ['New York', 'Los Angeles', 'Chicago']
}

df = pd.DataFrame(data)
```

```
# Print the DataFrame
print(df)
```

Output:

```
—--------------------------------
      Name  Age         City
0    Alice   25     New York
1      Bob   30  Los Angeles
2  Charlie   35      Chicago
```

Pandas is excellent for handling tabular data, filtering, aggregating, and performing operations on large datasets.

3. Matplotlib

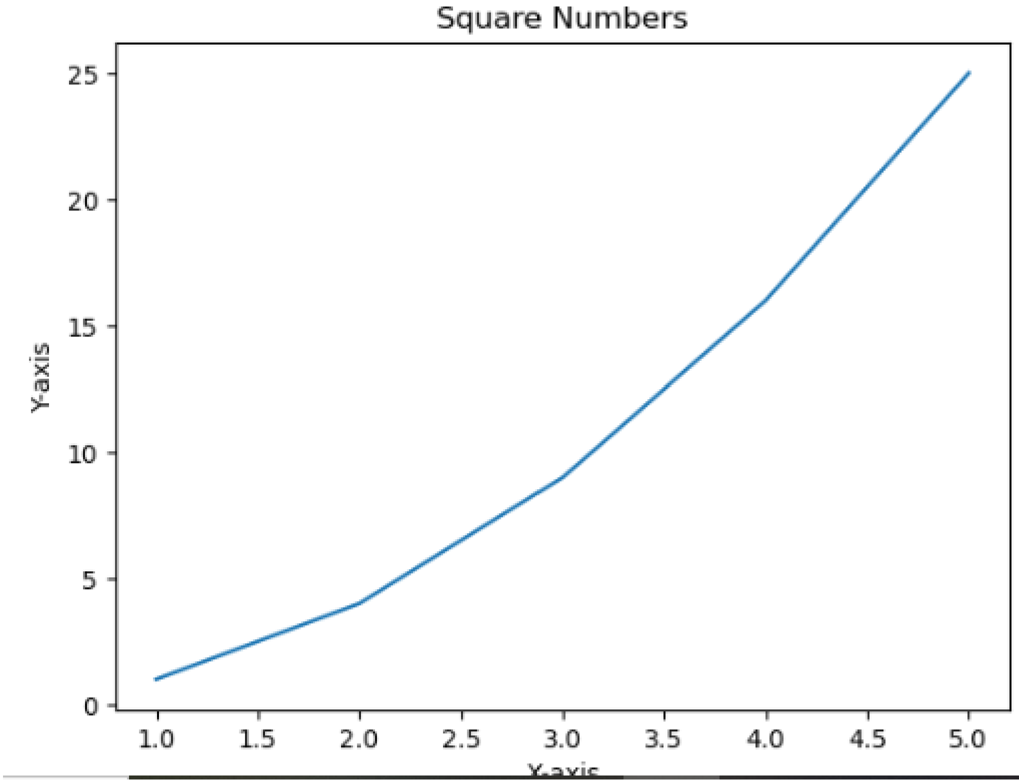
Matplotlib is a plotting library that provides
a wide range of tools for creating static, animated,
and interactive visualizations in Python.

Example:

```
import matplotlib.pyplot as plt

# Create some data
x = [1, 2, 3, 4, 5]
y = [1, 4, 9, 16, 25]

# Plot the data
plt.plot(x, y)

# Add labels and title
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Square Numbers')
```

```
# Show the plot
plt.show()
```

Output:



A plot with x on the horizontal axis and y on the vertical axis, showing a curve for the squares of the numbers.

Matplotlib is widely used for creating visualizations like line charts, bar charts, histograms, and scatter plots.

4. Requests

——----------------------

Requests is a simple and easy-to-use HTTP library that allows you to send HTTP requests, such as GET, POST, and more.
It is widely used for making web requests in Python.

```
import requests

# Send a GET request to a website
response =
requests.get('https://jsonplaceholder.typicode.com/posts')

# Check if the request was successful
if response.status_code == 200:
    # Parse JSON response
    posts = response.json()
    # Print the first post's title
    print(posts[0]['title'])
else:
    print('Failed to retrieve data')
```

Summary:

NumPy: Used for numerical computations and working with arrays.
Pandas: Great for data manipulation and analysis, especially with structured data.
Matplotlib: A powerful library for creating visualizations and plots.
Requests: Used to send HTTP requests and work with web data