



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

Advanced libraries in Python

Assignment - 3

Name:- Yash Sharma

College:- SOCET

Semester:- 4

Branch:- Btech CSE

Enrollment No.:- 2301031800112

Subject:- Python Programming

Introduction

- Python is a versatile language with a vast ecosystem of libraries.
- Advanced libraries help in data science, machine learning, web development, automation, and more.
- This presentation covers some of the most powerful libraries in Python.

NumPy (Numerical Computing)

Purpose: NumPy (Numerical Python) is essential for scientific computing and data processing.

Key Features:

Supports multi-dimensional arrays and matrices.

Provides mathematical and statistical functions.

Optimized for performance compared to Python lists.

Example Code:

```
import numpy as np
arr = np.array([1, 2, 3])
print(arr * 2) # Outputs: [2 4 6]
```

Pandas - Data Analysis & Manipulation

Purpose: Pandas is used for working with structured data (like CSV, Excel, databases).

Key Features:

Provides Series (1D) and DataFrame (2D) structures for handling data.

Supports data cleaning, transformation, and analysis.

Example Code:

```
import pandas as pd
df = pd.DataFrame({'Name': ['Alice', 'Bob'], 'Age': [25, 30]})
print(df)
```

Matplotlib & Seaborn - Data Visualization

Purpose:

Matplotlib: Used for basic charts like line plots, bar charts, scatter plots.

Seaborn: Built on Matplotlib, provides advanced visualizations and better aesthetics.

Key Features:

Helps in visualizing trends, relationships, and patterns in data.

Example Code:

```
import matplotlib.pyplot as plt
plt.plot([1, 2, 3], [4, 5, 6])
plt.title("Simple Line Graph")
plt.show()
```

Scikit-Learn - Machine Learning

Purpose: Scikit-Learn provides tools for machine learning and data mining.

Key Features:

Supports classification, regression, clustering, dimensionality reduction.
Includes algorithms like Decision Trees, SVM, Random Forest, etc.

Example Code:

```
from sklearn.linear_model import LinearRegression  
model = LinearRegression()
```

TensorFlow & PyTorch - Deep Learning

Purpose: Used for building neural networks and deep learning models.

Key Features:

TensorFlow: Optimized for large-scale machine learning; used in industry applications.

PyTorch: More flexible and user-friendly, widely used in research.

Example Code:

```
import torch  
x = torch.tensor([1.0, 2.0])  
print(x)
```

Slide 8: Flask & Django - Web Development

Purpose: These frameworks help build web applications using Python.

Key Differences:

Flask: Lightweight, minimal, best for small applications.

Django: Feature-rich, follows the "batteries-included" approach, best for large applications.

Example Code (Flask):

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def home():
    return "Hello, Flask!"
if __name__ == "__main__":
    app.run(debug=True)
```


Selenium & BeautifulSoup - Web Scrapping & Automation

Purpose: These libraries help in extracting data from websites.

Key Features:

Selenium: Automates browser interactions for testing or data extraction.

BeautifulSoup: Parses HTML and XML documents to extract data.

Example Code (BeautifulSoup):

```
from bs4 import BeautifulSoup
soup = BeautifulSoup('<p>Hello</p>', 'html.parser')
print(soup.p.text) # Outputs: Hello
```

OpenCV - Computer Vision

Purpose: OpenCV is used for image processing, facial recognition, and object detection.

Key Features:

Supports image transformations, filtering, and detection of objects.

Works with real-time video processing.

Example Code:

```
import cv2  
img = cv2.imread('image.jpg')  
cv2.imshow('Image', img)  
cv2.waitKey(0)
```

Requests - HTTP Requests & APIs

Purpose: The requests library is used for making HTTP requests to web services.

Key Features:

Helps in consuming APIs and web scraping.

Supports GET, POST, PUT, DELETE requests.

Example Code:

```
import requests  
response = requests.get('https://api.github.com')  
print(response.json()) # Outputs JSON response from GitHub API
```

Conclusion

Python has a vast collection of powerful libraries that help in various domains.

Choosing the right library can significantly speed up development and improve efficiency.

Keep exploring and practicing to master advanced Python libraries!