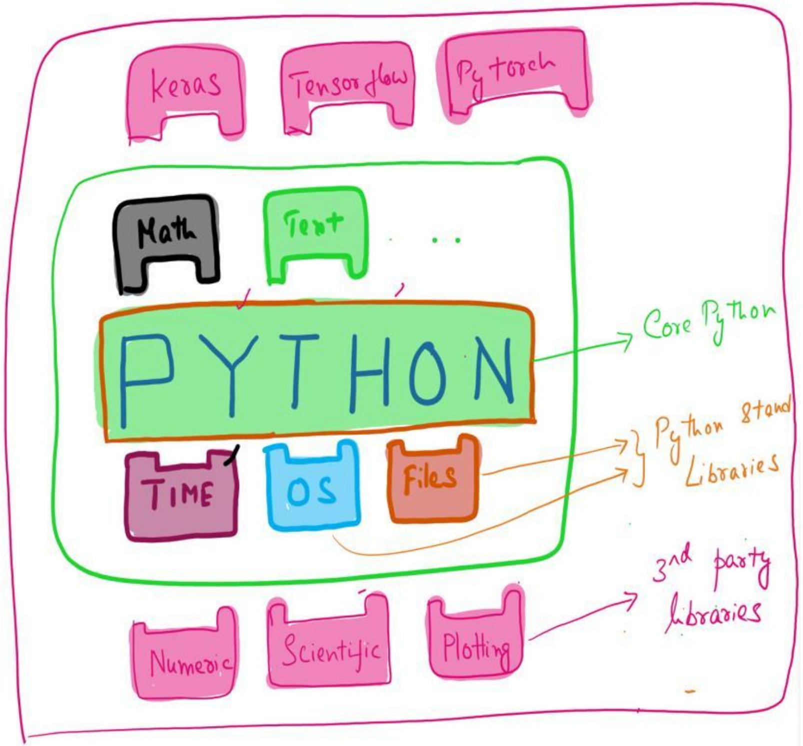
Chapter 4
Introduction to Python Libraries

Chapter Content

- 1. Python Libraries
- 2. Use of Libraries in Python Program
- 3. The Python Standard Library
- 4. The 3 rd Party Libraries

1. Python Library

- A Python library is a collection of related modules containing bundles of code that can be used repeatedly in different programs.
- It makes Python Programming simpler and convenient for the programmer when we needn't to write the same code again and again for different programs



2. Use of Libraries in Python Program

- Importing the libraries:

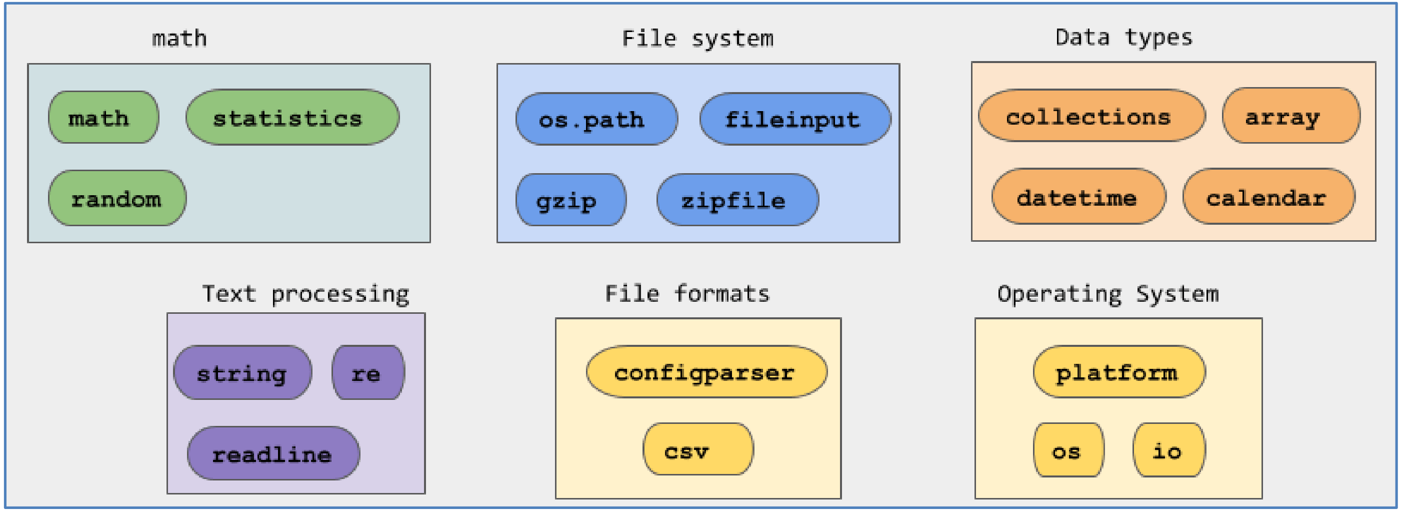
```
# Importing math library
import math
A = 16
print(math.sqrt(A))
```

- Importing specific items from a library module:

```
# Importing specific items
from math import sqrt, sin
A = 16
B = 3.14
print(sqrt(A))
print(sin(B))
```

3. Python Standard Library

- Python Standard library plays a very important role to access to the system functionalities of python
- https://docs.python.org/3/library/
- The library contains more than 200 core prebuilt pretested modules (written in C) that are provided as part of the default python installation itselt

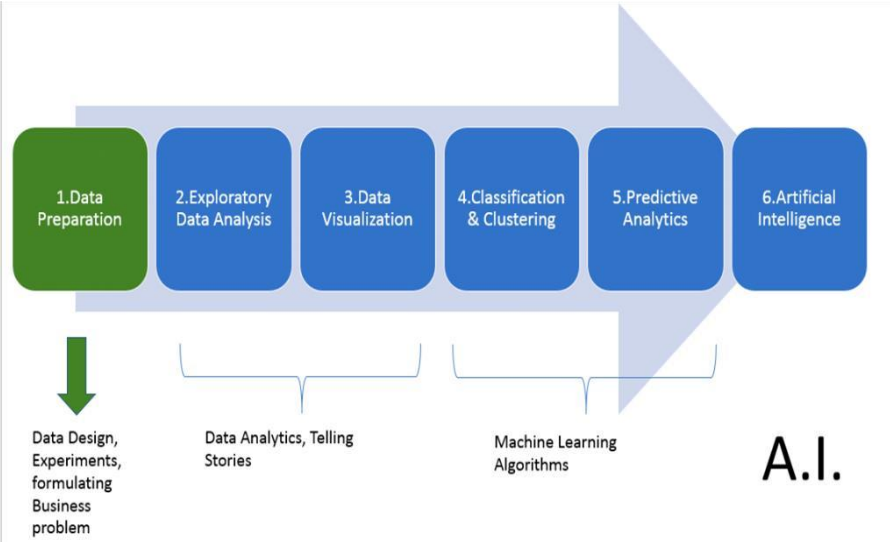


4. The 3rd Party Libraries

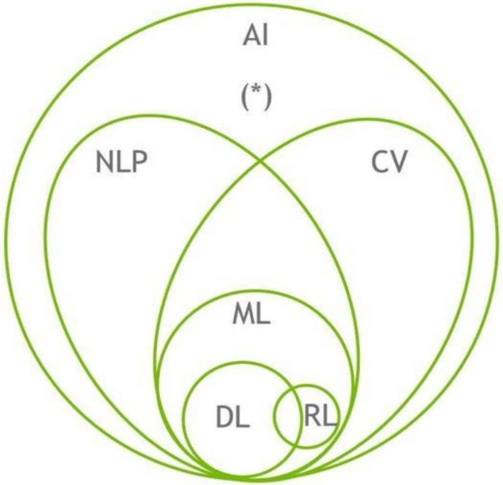
- Are open-source libraries (packages) developed by somebody else
- and made available for download using a standard interface.
- used to create applications and models in a variety of fields, e.g. machine learning, data science, data visualization, image and data manipulation, and many more.
- Find, install and publish Python packages with the Python Package Index https://pypi.org/
- Using pip to install packages

```
pip install <packages>
```

4. The 3rd Party Libraries: Data Science & AI


1.Data Preparation
2.Exploratory Data Analysis
3.Data Visualization
4.Classification & Clustering
5.Predictive Analytics
6.Artificial Intelligence
Data Design, Experiments, formulating Business problem
Data Analytics, Telling Stories
Machine Learning Algorithms
A.I.

Data mining from data sources.



AI = Artificial Intelligence
NLP=Natural Language Processing
CV=Computer Vision
ML=Machine Learning
DL=Deep Learning
RL=Reinforcement Learning

(*)=We would have more ellipses t
NLP or CV) representing Robotics,
Speech, and Planning, Scheduling
systems. But it would look very m
ahead and imagine they are there

4. The 3rd Party Libraries: Data Mining

- There are several Python tools for scraping data used in Python machine learning models.
- These libraries are known for web crawling, data scraping and arrange it into the required format

4. The 3rd Party Libraries: Data Mining

- There are mainly two ways to extract data from a website:
  - Use the API of the website (if it exists). For example, Facebook has the Facebook Graph API which allows retrieval of data posted on Facebook.
  - Access the HTML of the webpage and extract useful information/data from it. This technique is called web scraping or web harvesting or web data extraction.

4. The 3rd Party Libraries: Data Mining

- Example 1

install: pip install requests
pip install bs4

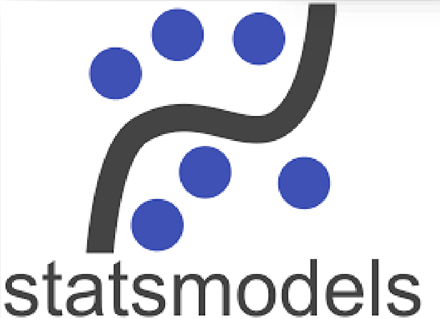
```
from bs4 import BeautifulSoup
import requests
# Fetch the web page
url = "https://example.com"
response = requests.get(url)
data = response.text
# Parse the HTML content
soup = BeautifulSoup(data, 'html.parser')
# Extract specific elements
titles = soup.find_all('h1')
for title in titles:
    print(title.text)
```

4. The 3rd Party Libraries: Data Analytics

- There are several Python tools for processing and visualizing data.

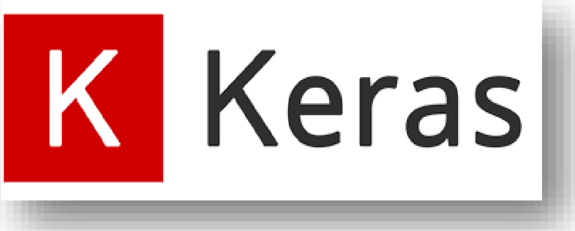
4. The 3rd Party Libraries: Machine Learning (Traditional Algorithms)

- Provide the implementation of traditional Machine Learning
- Algorithms:
  - Classification: SVM, Random Forest, Decision Tree, etc…
  - Clustering: K-Mean, etc ...
  - Except neural networks.



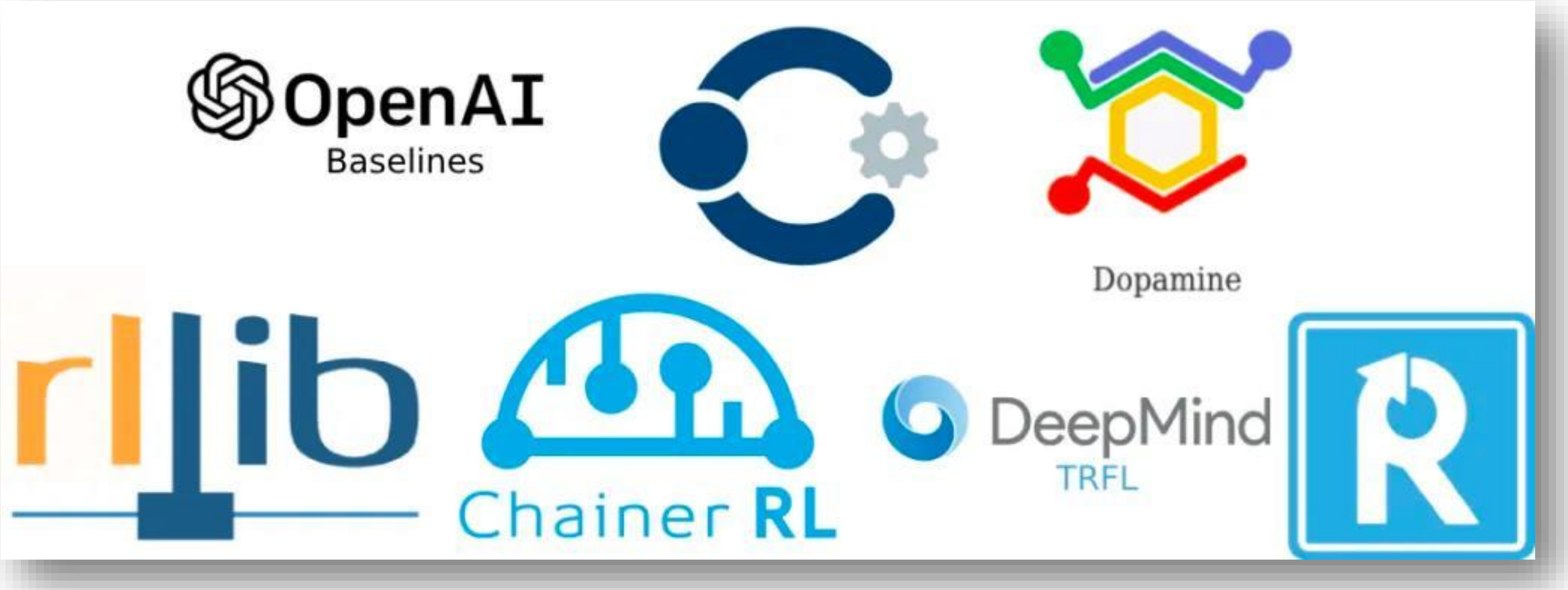
4. The 3rd Party Libraries: Deep Learning

TensorFlow



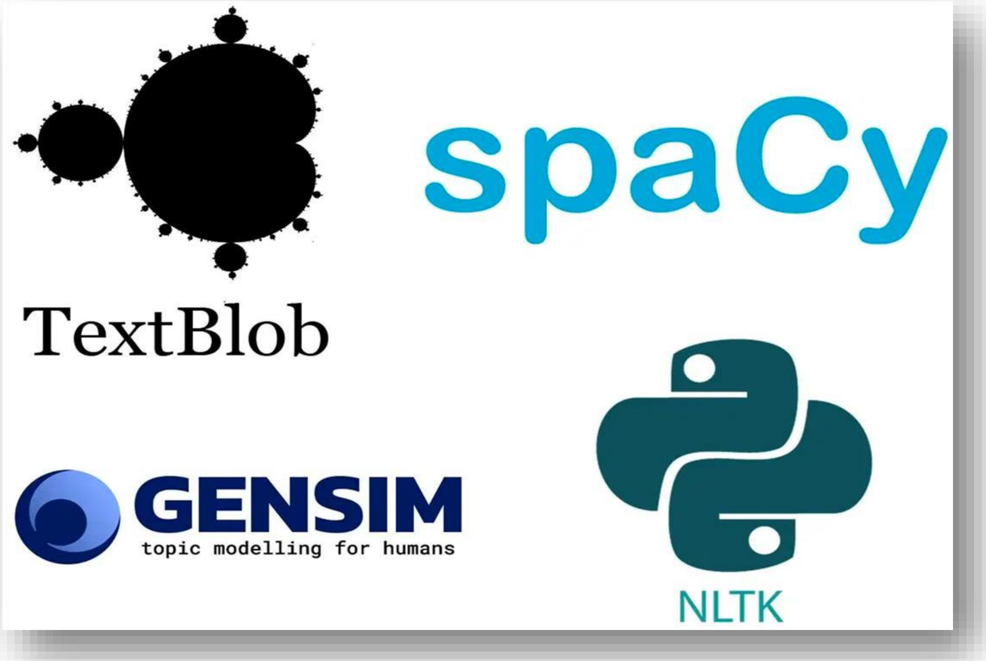
4. The 3rd Party Libraries: Reinforcement Learning

- There are libraries designed to have all the necessary tools to both implement and test Reinforcement Learning models



4. The 3rd Party Libraries: NLP Machine Learning

- There are many Python libraries that can be used for practically implementing natural language processing (NLP) and text mining tasks.



4. The 3rd Party Libraries: Computer Vision

- Python provides several computer vision libraries and frameworks for developers to help them automate tasks, which includes detections and visualisations.

4. The 3rd Party Libraries: Example

- Example 2

Install: pip install speedtest-cli

```python
import speedtest as st
def Speed_Test():
    test = st.Speedtest()
    down_speed = test.download()
    down_speed = round(down_speed / 10 ** 6,  2)
    print("Download Speed in Mbps: ", down_speed)

    up_speed = test.upload()
    up_speed = round(up_speed / 10 ** 6,  2)
    print("Upload Speed in Mbps: ", up_speed)

    ping = test.results.ping
    print("Ping: ", ping)

Speed_Test()
```

```
# Result

Download Speed in Mbps: 20.57

Upload Speed in Mbps: 18.47

Ping: 26.177
```

- Example 3

Install: pip install wifi-qrcode-generator

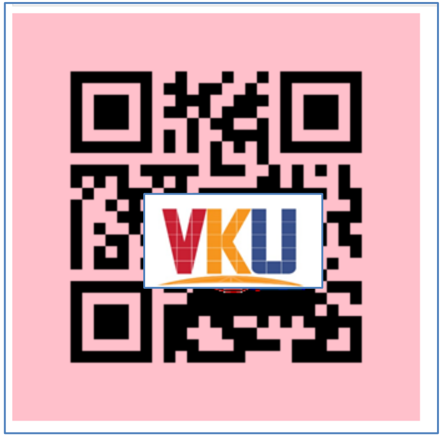
```
from wifi_qrcode_generator import wifi_qrcode

wq = wifi_qrcode("danang123", hidden=False,
authentication_type="WPA",password="ptc555" )
qrcodeimage = wq.make_image()
qrcodeimage.save("qrcode.png")
```

4. The 3rd Party Libraries: Example

- Example 4

Install: pip install qrcode
pip install pillow

```
import qrcode
from PIL import Image

url = https://www.vku.udn.vn   # Generate QR code for a URL

qr = qrcode.QRCode(version=1, error_correction=qrcode.constants.ERROR_CORRECT_L,
box_size=8, border=5)
qr.add_data(url)
qr.make(fit=True)

# Create an image with logo
image = qr.make_image(fill_color="black", back_color="pink")

logo = Image.open("logovku.png") # Add logo to the QR code
logo_size = img.size[0] // 4

# resizing with anti-aliasing
logo = logo.resize((logo_size, logo_size), Image.LANCZOS)
image.paste(logo, ((img.size[0] - logo.size[0]) // 2, (img.size[1] - logo.size[1]) // 2))

image.save("qr_code.png") # Save the image
Image.open("qr_code.png") # Open the image
```

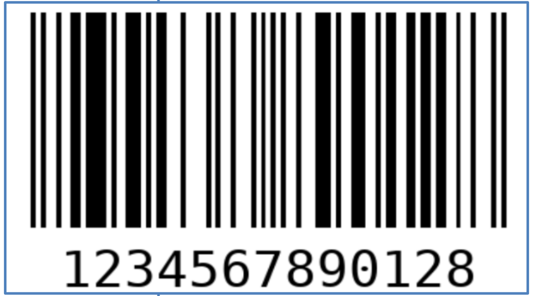


4. The 3rd Party Libraries: Example

- Example 5

Install: pip install python-barcode

```
import barcode

from barcode.writer import ImageWriter
from IPython.display import Image, display

barcode_format = barcode.get_barcode_class('ean13')
barcode_number = '1234567890128'
barcode_image = barcode_format(barcode_number,
writer=ImageWriter())
barcode_filename = 'barcode_image'
barcode_image.save(barcode_filename)

display(Image(filename=f'{barcode_filename}.png'))
```



4. The 3rd Party Libraries: Example

Install: pip install pickle

- Example 6

```
import pickle

my_dict = {'name': 'John', 'age': 30,
                           'city': 'New York'}
with open('dict.pkl', 'wb') as file:
      pickle.dump(my_dict, file)

with open('dict.pkl', 'rb') as file:
      loaded_dict = pickle.load(file)

print("Loaded Dictionary:", loaded_dict)

# Loaded Dictionary:  {'name': 'John',
'age': 30, 'city': 'New York'}
```

Saving and Loading
a Dictionary

```
import pickle

my_list = ['apple', 'banana', 'cherry']
with open('list.pkl', 'wb') as file:
      pickle.dump(my_list, file)

with open('list.pkl', 'rb') as file:
      loaded_list = pickle.load(file)

print("Loaded List:", loaded_list)


#Result: # Loaded List: ['apple',
'banana',  'cherry']
```

Saving and Loading
a List

```
import pickle

my_tuple = (10, 20, 30, 'Hello')
with open('tuple.pkl', 'wb') as file:
      pickle.dump(my_tuple, file)

with open('tuple.pkl', 'rb') as file:
      loaded_tuple = pickle.load(file)

print("Loaded Tuple:", loaded_tuple)


# Resulte: Loaded Tuple: (10, 20, 30,
'Hello')
```

Saving and Loading
a Tuple

4. The 3rd Party Libraries: Example

- Example 7

Install: pip install pickle

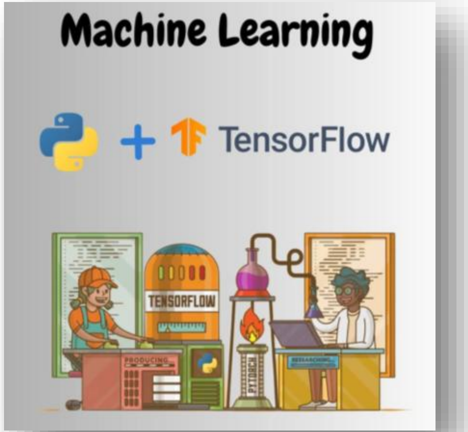
```
import pickle
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def __repr__(self):
        return f"Person(name={self.name}, age={self.age})"

person = Person('Alice', 25)
with open('person.pkl', 'wb') as file:
    pickle.dump(person, file)

with open('person.pkl', 'rb') as file:
    loaded_person = pickle.load(file)

print("Loaded Person:", loaded_person)
# Loaded Person: Person(name=Alice, age=25)
```

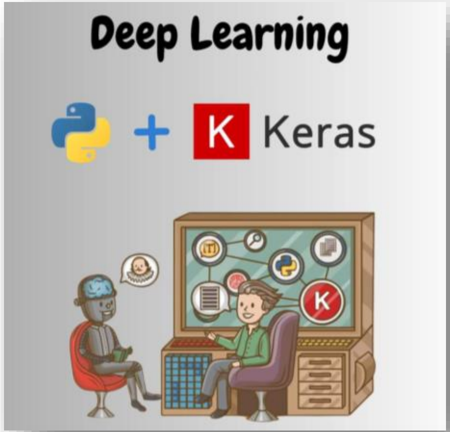
Saving and Loading a Class

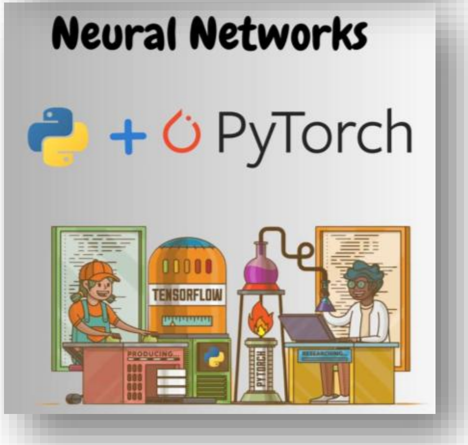
```
import pickle
list_data = [1, 2, 3]
dict_data = {'a': 1, 'b': 2}
string_data = "Hello, World!"

with open('multiple.pkl', 'wb') as file:
    pickle.dump(list_data, file)
    pickle.dump(dict_data, file)
    pickle.dump(string_data, file)

with open('multiple.pkl', 'rb') as file:
    loaded_list = pickle.load(file)
    loaded_dict = pickle.load(file)
    loaded_string = pickle.load(file)

print("Loaded List:", loaded_list)
print("Loaded Dictionary:", loaded_dict)
print("Loaded String:", loaded_string)
```

```
Loaded List: [1, 2, 3]
Loaded Dictionary: {'a': 1, 'b': 2}
Loaded String: Hello, World!
```

Summary


Machine Learning
+ TensorFlow
TENSORFLOW
PRODUCING
PYTORCH
RESEARCHING

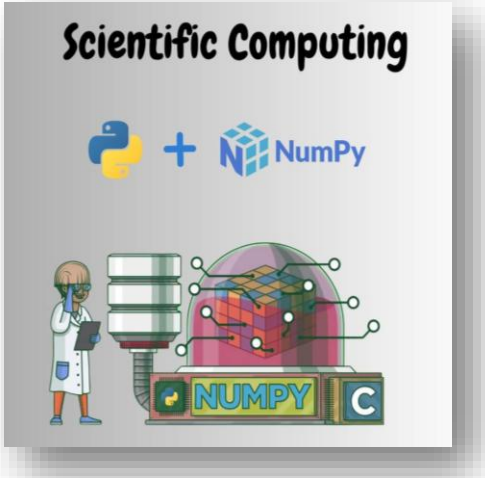

Deep Learning
+ K Keras
K

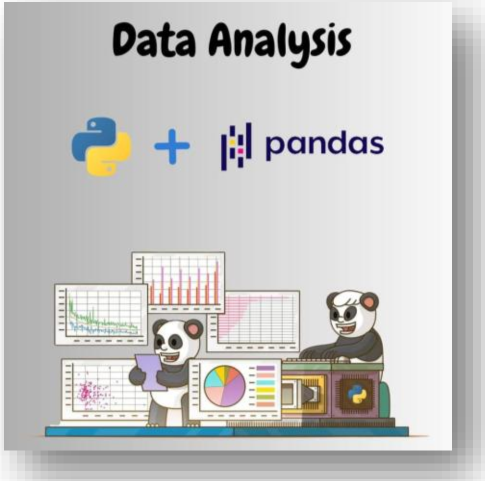

Neural Networks
+ PyTorch
TENSORFLOW
PRODUCING
PYTORCH
RESEARCHING

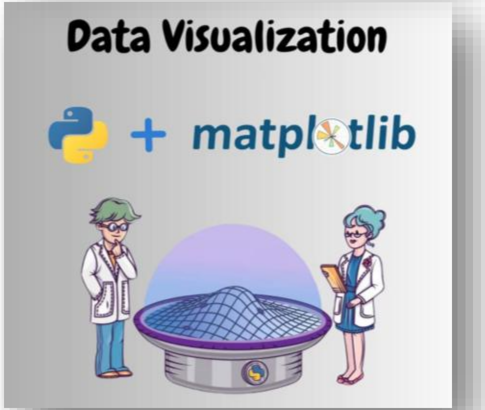
Computer Vision
+ OpenCV

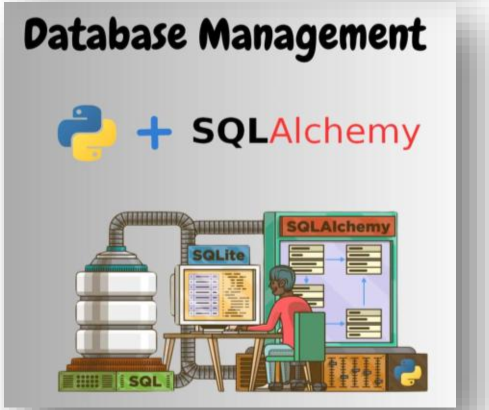
23

Summary

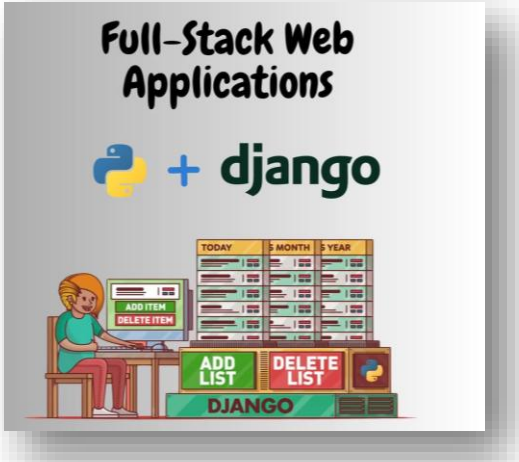

Scientific Computing
+ NumPy
NUMPY
C

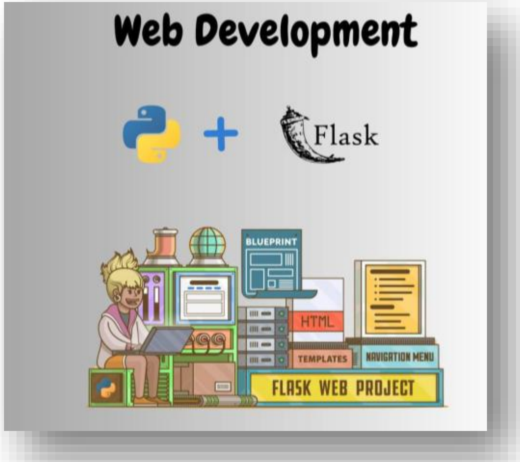

Data Analysis
+ pandas


Data Visualization
+ matplotlib


Database Management
+ SQLAlchemy
SQLAlchemy
SQLite
SQL

Summary


Full-Stack Web Applications
+ django
TODAY
5 MONTH
5 YEAR
ADD ITEM
DELETE ITEM
ADD LIST
DELETE LIST
DJANGO


Web Development
+ Flask
BLUEPRINT
HTML
TEMPLATES
NAVIGATION MENU
FLASK WEB PROJECT

Web Scraping
+ Scrapy
DATA
SCRAPY
SCRAPING...
MONGODB
STORING DATA...
CRAWL

Automated Testing
+ Se Selenium

The end of Chapter