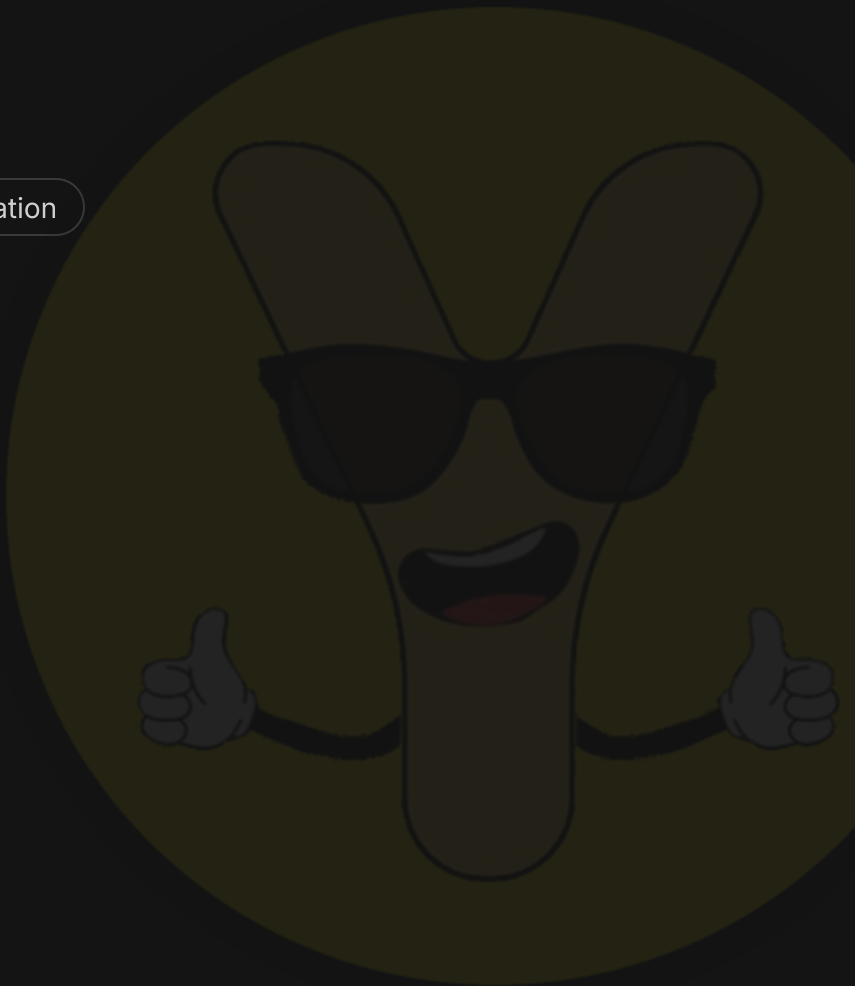
AKTU · 2ND YEAR · BCC302 / BCC402

Python Programming

Unit 5 — Data Analysis and Web Scraping

- Introduction to NumPy: arrays, operations, indexing
- Introduction to Pandas: Series, DataFrames, data manipulation
- Data visualization with Matplotlib: line plots, scatter plots, histograms
- Introduction to web scraping
- BeautifulSoup and Requests libraries
- Extracting data from HTML pages



DECODED BY

Chirag Jain

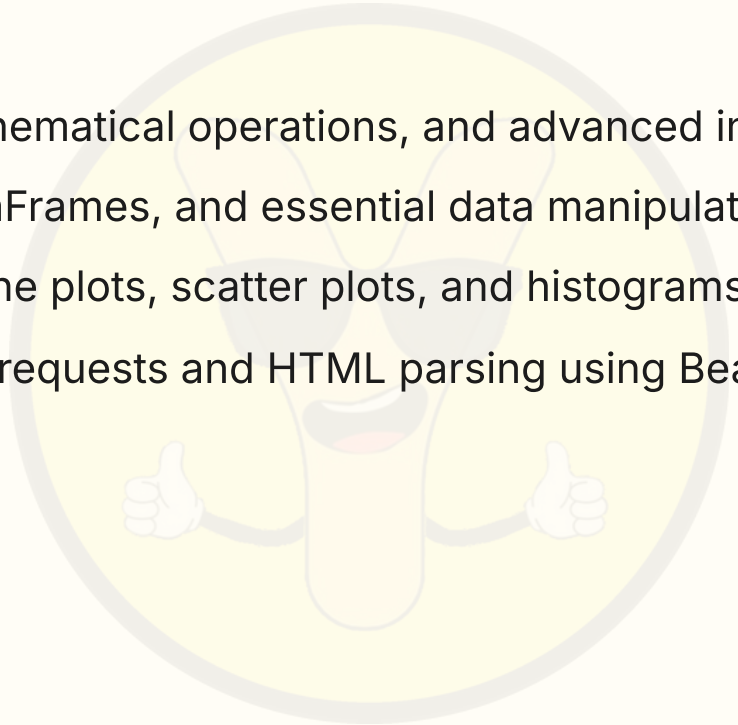
Founder, Ycotes · Gold Medalist in Artificial Intelligence, IIT Madras · taught 10,000+ students

theycotes.com

Unit 5 Overview: Data Analysis & Web Scraping

This unit covers numerical computing, tabular data handling, visualization, and extracting data from websites.

- Introduction to NumPy: arrays, mathematical operations, and advanced indexing.
- Introduction to Pandas: Series, DataFrames, and essential data manipulation techniques.
- Data visualization with Matplotlib: line plots, scatter plots, and histograms.
- Web scraping fundamentals: HTTP requests and HTML parsing using BeautifulSoup.



Introduction to NumPy

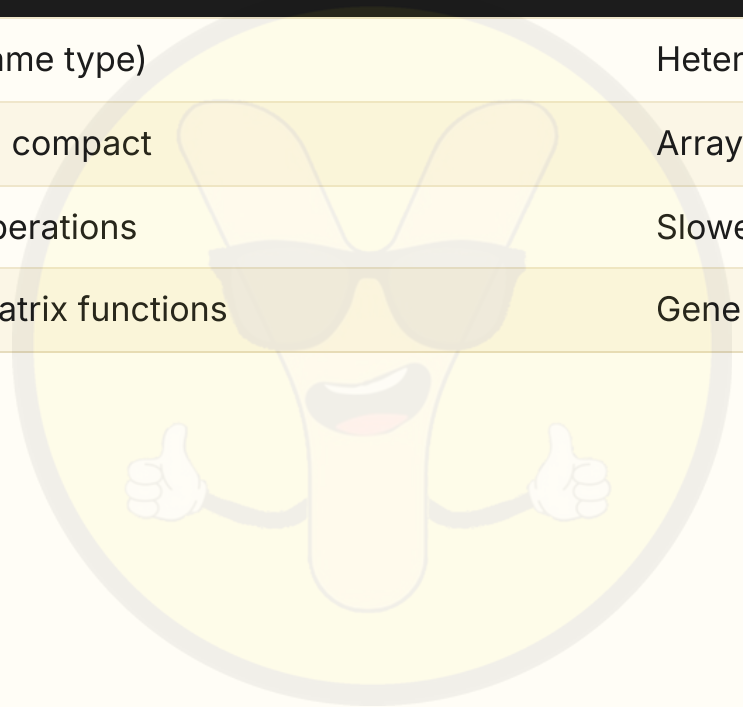
NUMPY (NUMERICAL PYTHON)

A foundational Python library for scientific computing, designed to handle large multi-dimensional arrays and matrices of numerical data efficiently.

- Provides the ndarray object for homogeneous data types.
- Implemented in C, offering execution speeds significantly faster than standard Python lists.
- Forms the underlying structural foundation for Pandas, SciPy, and Scikit-learn.

NumPy Arrays vs Python Lists

Feature	NumPy Array	Python List
Data Type	Homogeneous (same type)	Heterogeneous (any type)
Memory	Contiguous block, compact	Array of pointers, fragmented
Performance	Fast vectorized operations	Slower iteration via loops
Functionality	Mathematical & matrix functions	General-purpose container



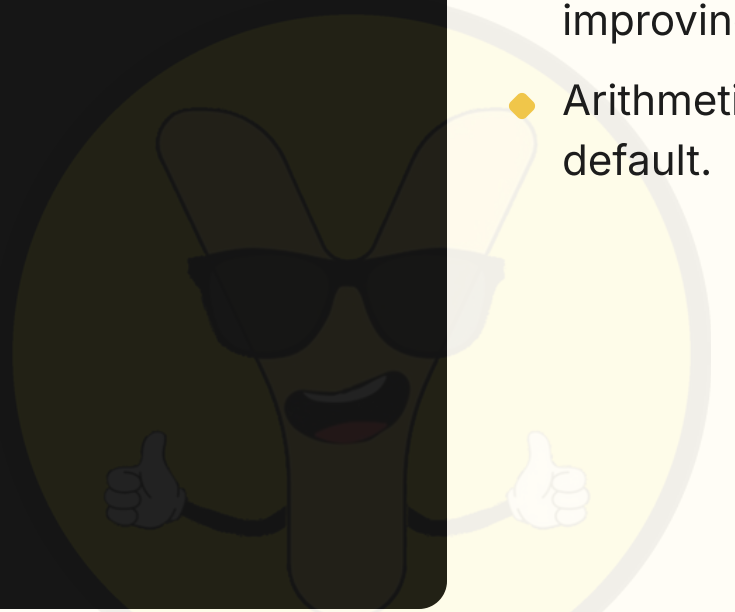
NumPy Array Operations

```
import numpy as np

# Create 1D and 2D arrays
a = np.array([1, 2, 3])
b = np.array([[1, 2], [3, 4]])

# Vectorized addition
c = a + 10
print(c)  # Output: [11 12 13]

# Matrix dot product
dot_prod = np.dot(b, b)
print(dot_prod)
```

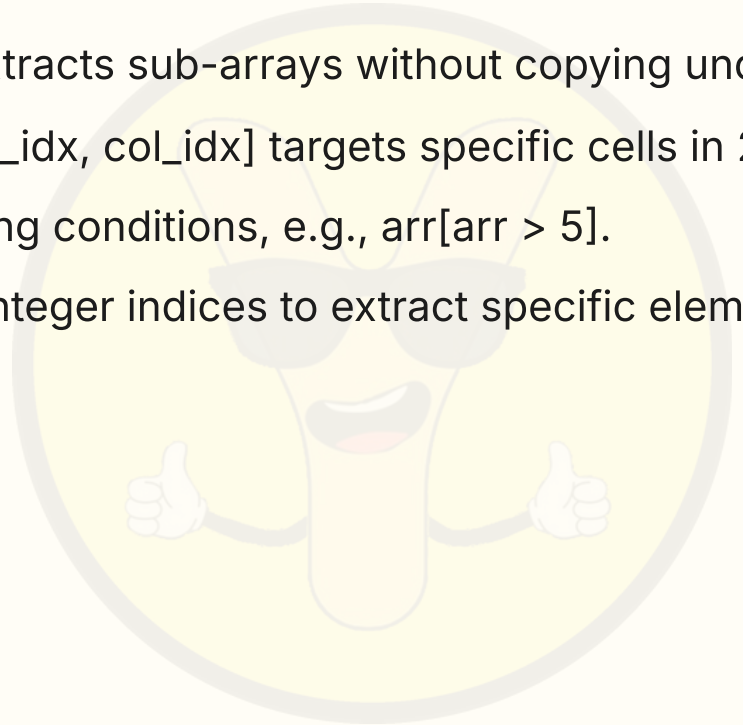


- Vectorization eliminates explicit for-loops, improving readability and speed.
- Arithmetic operations apply element-wise by default.

NumPy Indexing and Slicing

Accessing and modifying specific elements or subsets of multidimensional arrays.

- Basic Slicing: arr[start:stop:step] extracts sub-arrays without copying underlying data.
- Multi-dimensional indexing: arr[row_idx, col_idx] targets specific cells in 2D arrays.
- Boolean Indexing: Filtering data using conditions, e.g., arr[arr > 5].
- Fancy Indexing: Passing arrays of integer indices to extract specific elements.



Introduction to Pandas

PANDAS

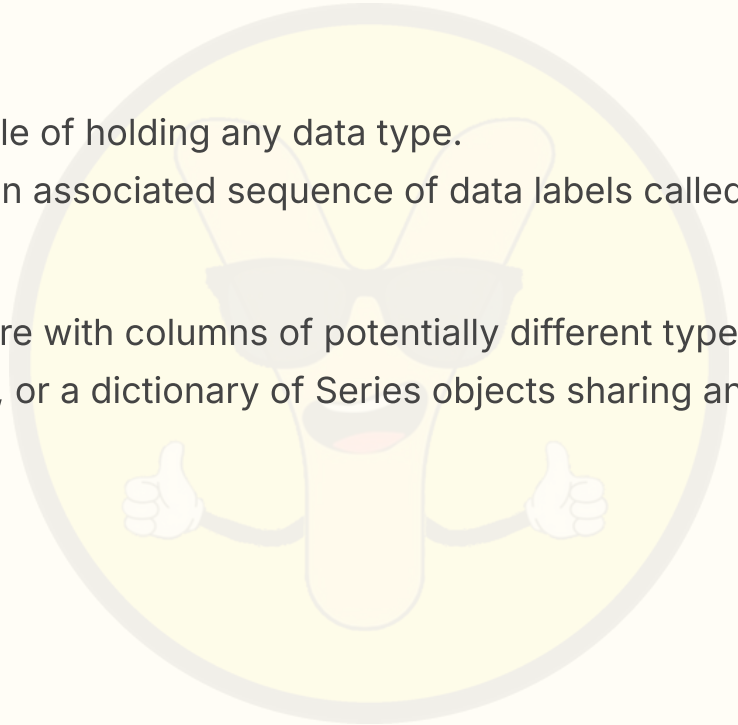
A high-level data manipulation library built on top of NumPy, providing fast, flexible, and expressive data structures designed to work with structured data.

- Core data structures are Series (1D) and DataFrame (2D).
- Excels at handling missing data, alignment, grouping, and merging.
- Supports reading and writing diverse file formats like CSV, Excel, and SQL databases.

Pandas Core Data Structures

Understanding Series and DataFrames.

- Series
  - A one-dimensional labeled array capable of holding any data type.
  - Consists of a sequence of values and an associated sequence of data labels called the index.
- DataFrame
  - A two-dimensional labeled data structure with columns of potentially different types.
  - Analogous to a spreadsheet, SQL table, or a dictionary of Series objects sharing an index.

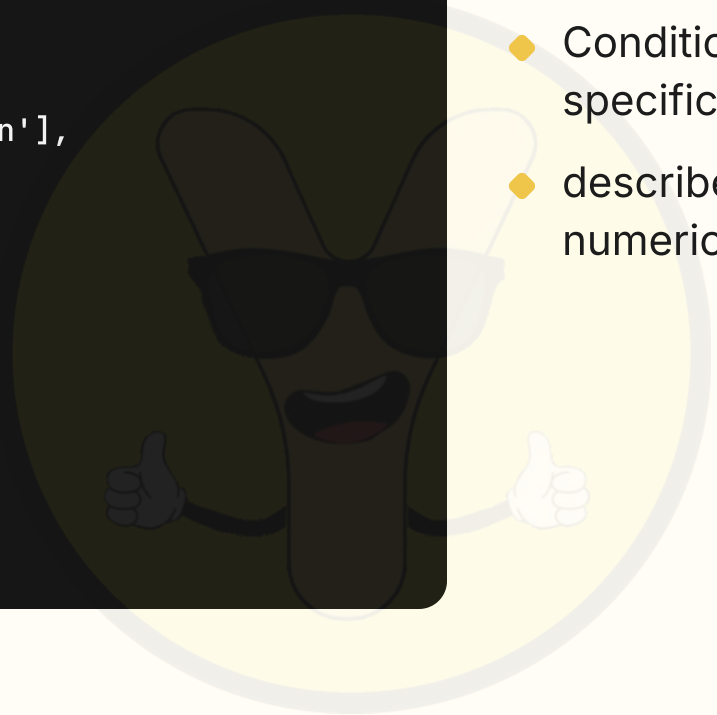


Pandas Data Manipulation

```
import pandas as pd

# Creating a DataFrame
data = {'Name': ['Aman', 'Priya', 'Rohan'],
        'Age': [21, 22, 20],
        'Marks': [85, 90, 78]}
df = pd.DataFrame(data)

# Data manipulation methods
print(df.head(2))
print(df[df['Marks'] > 80])
df['Percentage'] = df['Marks'] * 1.0
print(df.describe())
```



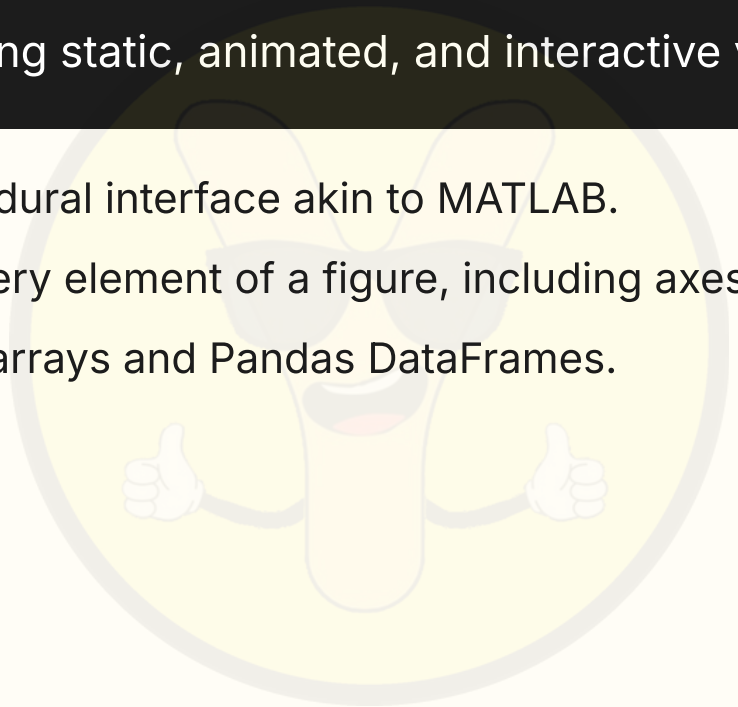
- head(n) returns the first n rows.
- Conditional filtering selects rows meeting specific criteria.
- describe() generates summary statistics for numerical columns.

Introduction to Data Visualization

MATPLOTLIB

A comprehensive library for creating static, animated, and interactive visualizations in Python.

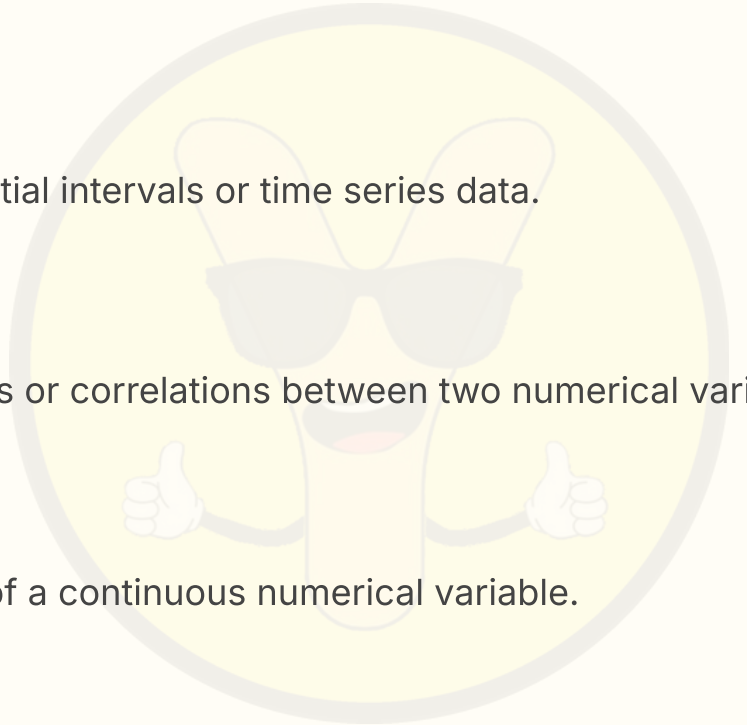
- pyplot submodule provides a procedural interface akin to MATLAB.
- Allows fine-grained control over every element of a figure, including axes, labels, and legends.
- Integrates seamlessly with NumPy arrays and Pandas DataFrames.



Matplotlib Basic Plot Types

Choosing the correct plot type for data exploration.

- Line Plots
  - plt.plot(x, y)
  - Ideal for visualizing trends over sequential intervals or time series data.
- Scatter Plots
  - plt.scatter(x, y)
  - Used to observe and show relationships or correlations between two numerical variables.
- Histograms
  - plt.hist(data, bins)
  - Represents the frequency distribution of a continuous numerical variable.

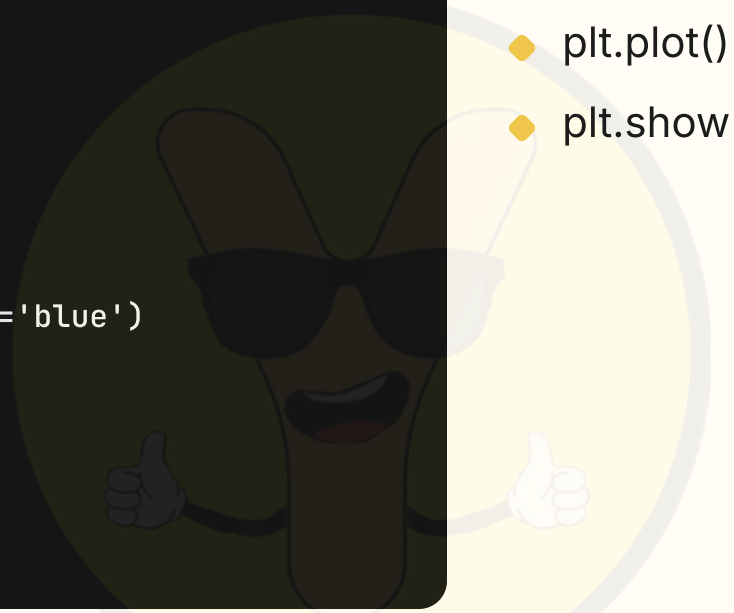


Matplotlib Plotting Example

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 10, 100)
y = np.sin(x)

plt.figure()
plt.plot(x, y, label='Sine Wave', color='blue')
plt.title('Simple Line Plot')
plt.xlabel('X Axis')
plt.ylabel('Y Axis')
plt.legend()
plt.show()
```



- plt.figure() creates a new figure canvas.
- plt.plot() draws lines connecting data points.
- plt.show() renders the plot to the screen.

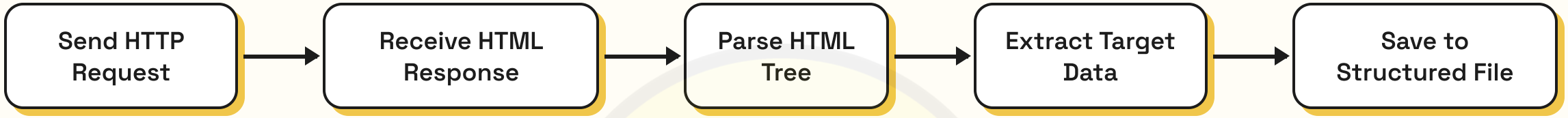
Introduction to Web Scraping

WEB SCRAPING

An automated technique used to extract large amounts of data from websites, saving the extracted information into a structured format like a CSV or database.

- Involves fetching the webpage HTML content using HTTP protocols.
- Parses the raw HTML document to target and extract specific elements.
- Essential when websites do not provide a public API for data access.

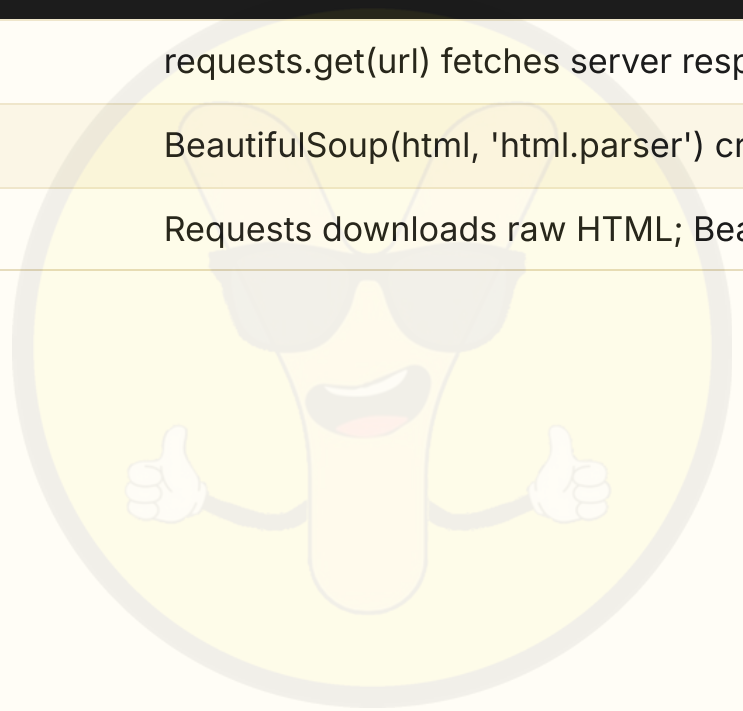
Web Scraping Workflow



The standard lifecycle of a web scraping script.

Requests vs BeautifulSoup

Library	Primary Role	Key Method / Feature
Requests	Fetching web pages	requests.get(url) fetches server response
BeautifulSoup	Parsing HTML/XML	BeautifulSoup(html, 'html.parser') creates parse tree
Collaboration	Combined Usage	Requests downloads raw HTML; BeautifulSoup extracts info

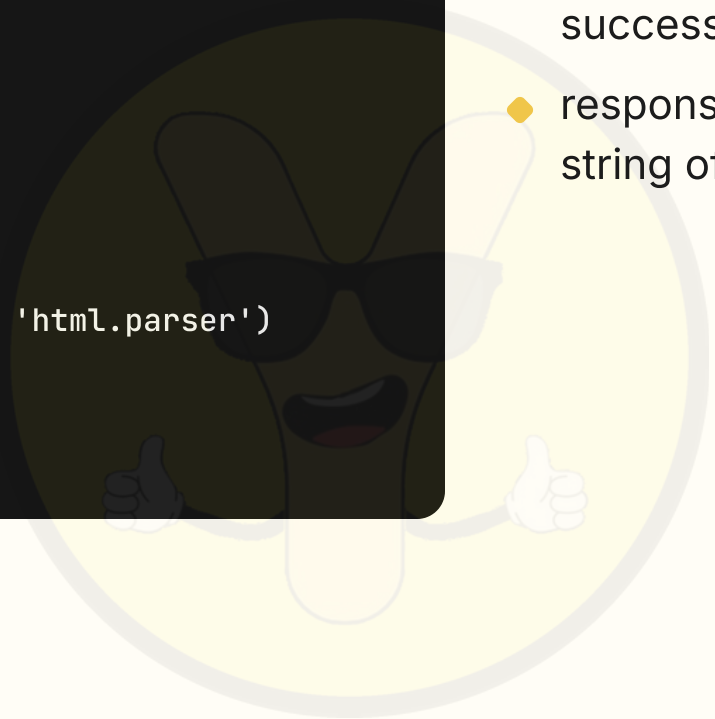


Requests and BeautifulSoup Basics

```
import requests
from bs4 import BeautifulSoup

url = 'https://example.com'
response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.text, 'html.parser')
    print(soup.title.text)
else:
    print('Failed to retrieve webpage')
```

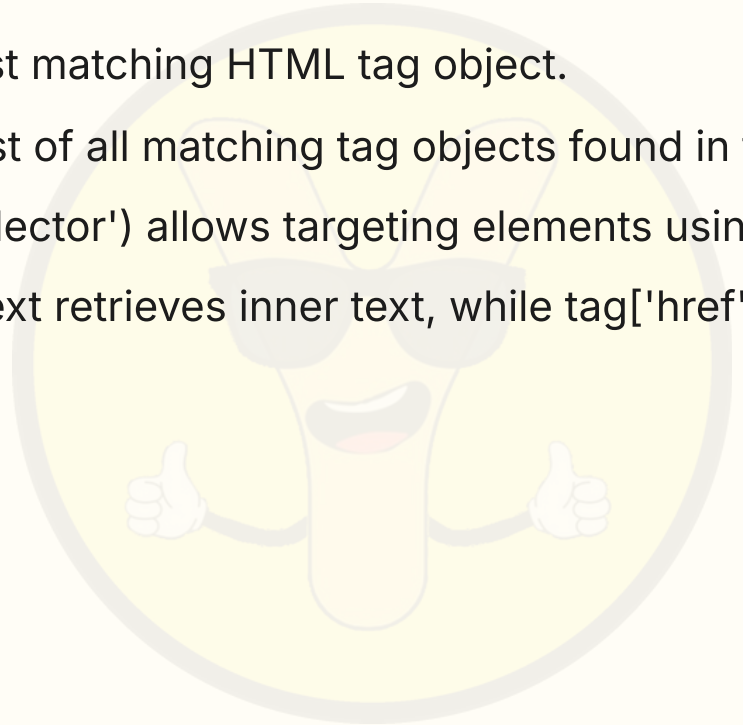


- response.status_code 200 indicates a successful HTTP connection.
- response.text contains the raw HTML source string of the page.

Extracting Data from HTML Pages

Methods to locate and retrieve content from BeautifulSoup parse trees.

- find(tag, attributes): Returns the first matching HTML tag object.
- find_all(tag, attributes): Returns a list of all matching tag objects found in the document.
- CSS Selectors: soup.select('css_selector') allows targeting elements using standard CSS syntax.
- Extracting text and attributes: tag.text retrieves inner text, while tag['href'] extracts attribute values.

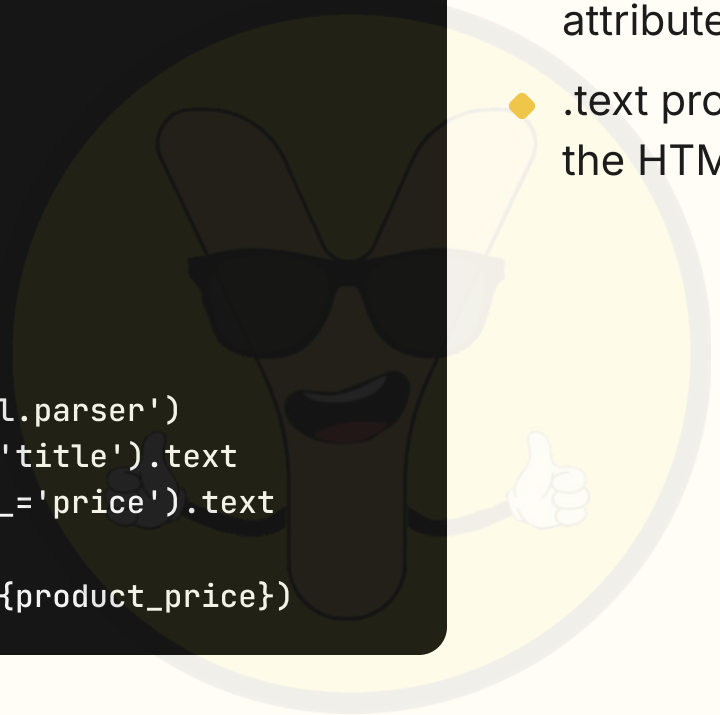


HTML Data Extraction Example

```
html_content = '''
<html><body>
  <div class="product">
    <h2 class="title">Laptop</h2>
    <span class="price">$999</span>
  </div>
</body></html>
'''

soup = BeautifulSoup(html_content, 'html.parser')
product_title = soup.find('h2', class_='title').text
product_price = soup.find('span', class_='price').text

print(Product: {product_title}, Price: {product_price})
```



- class_ keyword argument handles the class attribute in Python.
- .text property extracts the clean string inside the HTML tags.

Exam corner

- **Q1** Differentiate between NumPy arrays and Python lists with respect to memory and performance.
- **Q2** Explain the core data structures in Pandas: Series and DataFrame.
- **Q3** Write a NumPy code snippet to perform matrix multiplication and vectorized addition.
- **Q4** What is data visualization? Explain the usage of line plots, scatter plots, and histograms in Matplotlib.
- **Q5** Describe the role of Requests and BeautifulSoup libraries in web scraping.
- **Q6** Write a Python script using BeautifulSoup to extract all hyperlink URLs from a sample HTML page.
- **Q7** Explain the importance of checking HTTP status codes (like 200, 404) during web scraping.

Unit 5 done. You are exam-ready.

Revise the Exam corner once more, then open Unit 1 on theycotes.com. Every AKTU subject, unit by unit, free. Notes by Chirag Jain, Ycotes.

