Python Library

pandas

The Top Two most useful Libraries in Python, particularly for Data science ,forming the foundation for many data analysis, scientific computing task and general Programming

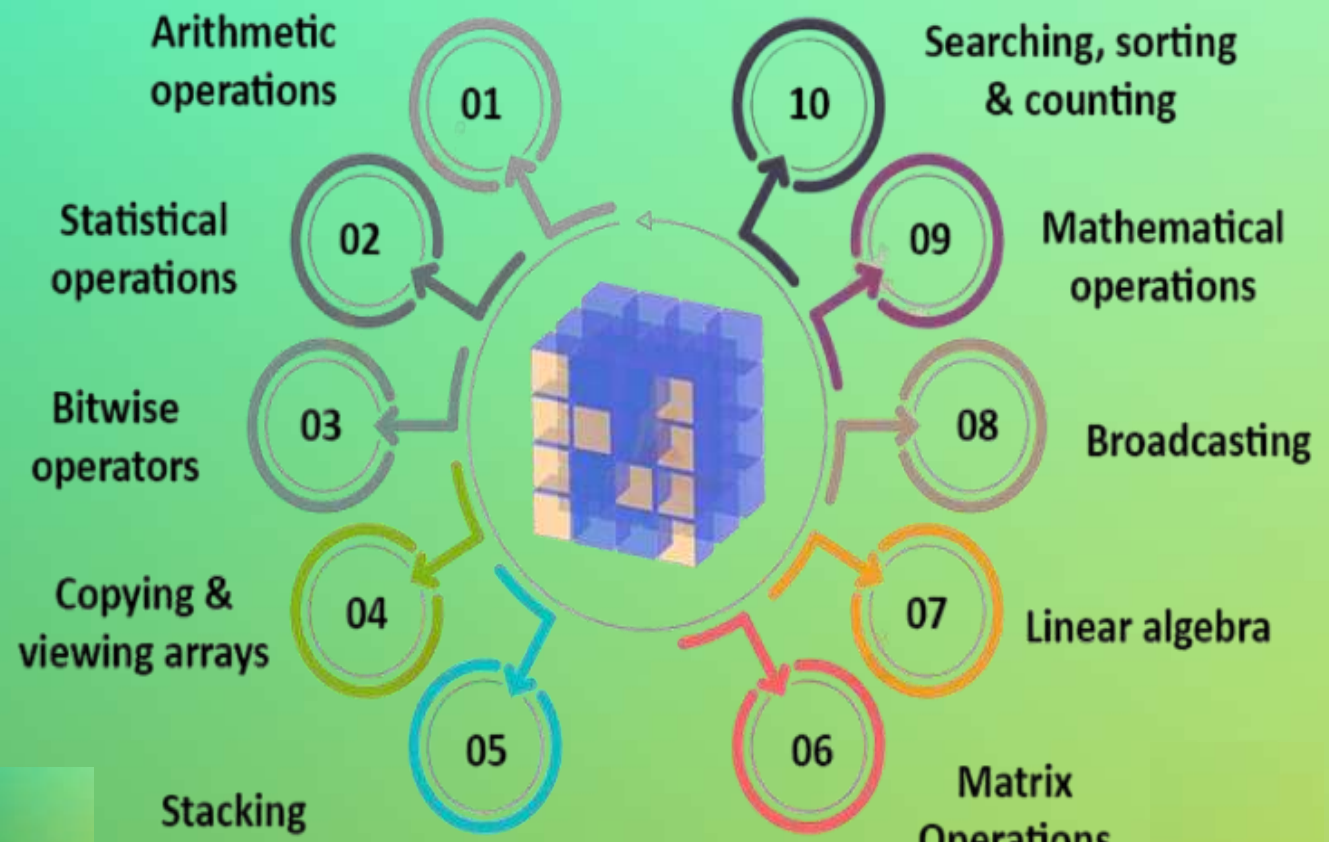
NumPy

Uses of Pandas

pandas

- Data cleaning
- Data Exploration
- Data Transformation
- Time Series Analysis
- Data Aggregation
- Data Input / Output
- Handling Categorical Data
- Financial Analysis
- Machine Learning Preparation

Uses of NumPy



Pandas is a powerful library in Python used for data manipulation and analysis. The line import pandas as pd imports the library and allows you to use the shorthand pd to reference it in your code. Here are some key features and uses of pandas:

1. Data Structures: Pandas provides two main data structures:

- **Series**: A one-dimensional labeled array that can hold any data type.
- **DataFrame**: A two-dimensional labeled data structure with columns of potentially different types, similar to a table in a database or a spreadsheet.

2. Data Manipulation: You can easily manipulate data with functions to:

- Filter and sort data.
- Handle missing values.
- Merge and join datasets.

3. Data Analysis: It provides tools for statistical analysis, including aggregating and summarizing data.
4. Data Input/Output: Pandas can read from and write to various file formats, such as CSV, Excel, and SQL databases.

Here's a simple example of how to use pandas:

```
import pandas as pd

# Create a DataFrame
data = {'Name': ['Alice', 'Bob', 'Charlie'], 'Age': [24, 27, 22], 'City': ['New York', 'Los Angeles', 'Chicago']}
df = pd.DataFrame(data)


# Display the DataFrame
print(df)

# Filter the DataFrame for ages greater than 23
filtered_df = df[df['Age'] > 23]
print(filtered_df)
```

This code creates a DataFrame, displays it, and then filters for rows where the age is greater than 23

numpy

The line **import numpy as np** is used to import the NumPy library in Python, which is a fundamental package for scientific computing. Here's a breakdown of its features and usage:

Key Features of NumPy

1. **N-dimensional Arrays**: NumPy provides the ndarray object, which is a fast and flexible container for large data sets in Python. You can create arrays of any dimension (1D, 2D, 3D, etc.).

2. **Mathematical Functions**: It includes a wide variety of mathematical functions for operations on arrays, including element-wise operations, statistical functions, linear algebra, and more.

3. **Performance**: NumPy arrays are more efficient in terms of memory and speed compared to Python lists, especially for large data sets.

4. **Integration**: It integrates well with other libraries like pandas, SciPy, and matplotlib, making it a core component of the scientific Python ecosystem.

Here's a simple example of using NumPy:

```
import numpy as np

# Create a 1D array
arr = np.array([1, 2, 3, 4, 5])
print("1D Array:", arr)

# Create a 2D array (matrix)
matrix = np.array([[1, 2, 3], [4, 5, 6]])
print("2D Array:\n", matrix)

# Perform element-wise operations
squared = arr ** 2
print("Squared Array:", squared)

# Calculate the mean
mean_value = np.mean(arr)
print("Mean:", mean_value)
```

In this example, we create both a 1D and a 2D array, perform element-wise squaring, and compute the mean of the 1D array.