

Create Objects

Creating a NumPy array

```
arr = np.array([1, 2, 3, 4, 5])
```

Creating a Pandas DataFrame

```
df = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]})
```

Indexing and Selection:

Indexing and slicing NumPy array

```
arr[0] Access element at index 0
```

```
arr[2:4] Slice elements from index 2 to 4
```

Indexing and slicing Pandas DataFrame

```
df['A'] Access column 'A'
```

```
df.loc[0] Access row at label 0
```

```
df.iloc[0] Access row at index 0
```

Operations

Arithmetic operations with NumPy arrays

```
np.sum(arr) Sum of all elements
```

```
np.mean(arr) Mean of all elements
```

Arithmetic operations with Pandas DataFrame

```
df.sum() Sum of all elements (column-wise)
```

```
df.mean() Mean of all elements (column-wise)
```

Missing Data Handling

Numpy

NumPy does not have built-in support for missing data

Handling missing data in Pandas DataFrame

```
df.isnull() Detect missing values
```

```
df.dropna() Drop rows with missing values
```

```
df.fillna(value) Fill missing values with specified value
```

Grouping

Grouping data in a Pandas DataFrame and calculates the Mean value for each group

```
df.groupby('column_name').mean()
```

Grouping data in a Pandas DataFrame and calculates the Sum of values for each group

```
df.groupby('column_name').sum()
```

Aggregation

Aggregating data in a Pandas DataFrame

```
df.groupby('column_name').agg({'column_name': 'sum'})
```

Example: Sum of 'values' column for each 'category'

```
df.groupby('category').agg({'values': 'sum'})
```

Filtering

Filtering data in a Pandas DataFrame

```
df[df['column_name'] > threshold]
```

Example: Filtering rows where 'values' column is greater than 10

```
df[df['values'] > 10]
```

Indexing

