

◇ NumPy Detailed Cheat sheet

Import NumPy

```
import numpy as np
```

- np is an alias (industry standard).
- NumPy is used for **numerical computing** and **multi-dimensional arrays**.
- Faster than Python lists because it is implemented in C.

Creating Arrays

◊ Basic Array

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

Creates a 1D array.

◊ Zeros

```
np.zeros((2,3))
```

Creates 2x3 matrix filled with 0.

◊ Ones

```
np.ones((3,3))
```

Creates 3x3 matrix filled with 1.

◊ Range

```
np.arange(0, 10, 2)
```

Like Python `range()`, but returns NumPy array.

◊ Evenly Spaced Values

```
np.linspace(0, 1, 5)
```

Creates 5 equally spaced numbers between 0 and 1.

◊ Random Numbers

```
np.random.rand(3,3)
```

3x3 matrix with random values between 0 and 1.

❖ Used heavily in ML for weight initialization.

Array Attributes

```
arr.shape    # (rows, columns)
arr.ndim      # number of dimensions
arr.dtype     # data type
arr.size      # total elements
```

Example:

```
arr = np.array([[1, 2, 3], [4, 5, 6]])
```

- shape → (2,3)
- ndim → 2
- size → 6

❖ Important in ML when checking feature dimensions.

Indexing & Slicing

◊ Access Element

```
arr[0]
```

◊ Slice

```
arr[0:2]
```

◊ 2D Access

```
arr[:, 1]    # all rows, column 1
```

❖ NumPy slicing returns **view**, not copy (important for memory optimization).

Reshaping

```
arr.reshape(2, 3)
```

Changes shape without changing data.

```
arr.flatten()
```

Converts multi-dimensional array into 1D.

❖ Used in ML before feeding data into models.

Mathematical Operations

NumPy supports **vectorized operations** (no loops needed).

```
arr + 2  
arr * 3
```

◊ Universal Functions (ufuncs)

```
np.sqrt(arr)  
np.mean(arr)  
np.sum(arr)  
np.std(arr)
```

❖ Vectorization makes NumPy much faster than Python loops.

Aggregation

```
np.min(arr)  
np.max(arr)  
np.argmin(arr)  
np.argmax(arr)
```

- `argmin()` → index of minimum value
- `argmax()` → index of maximum value

❖ Used in optimization and ML evaluation.

Linear Algebra

```
np.dot(a, b)
```

Matrix multiplication.

```
np.linalg.inv(a)
```

Matrix inverse.

```
np.linalg.det(a)
```

Determinant.

❖ Core for ML algorithms like Linear Regression.

◇ Pandas Detailed Cheat sheet

Pandas is built on top of NumPy and used for **data manipulation & analysis**.

Import Pandas

```
import pandas as pd
```

Creating DataFrame

◊ From Dictionary

```
data = {'name': ['A', 'B'], 'age': [25, 30]}\ndf = pd.DataFrame(data)
```

◊ From CSV

```
pd.read_csv("file.csv")
```

◊ From Excel

```
pd.read_excel("file.xlsx")
```

❖ DataFrame = table (rows + columns).

Inspecting Data

```
df.head()          # first 5 rows\ndf.tail()         # last 5 rows\ndf.info()         # structure + null values\ndf.describe()     # statistical summary\ndf.shape          # (rows, columns)\ndf.columns        # column names
```

❖ First step in any Data Science project = EDA (Exploratory Data Analysis).

Selection

◊ Single Column

```
df['column']
```

◊ Multiple Columns

```
df[['col1', 'col2']]
```

◊ Label-based

```
df.loc[0]
```

◊ Index-based

```
df.iloc[0:3]
```

📌 loc → labels

📌 iloc → index positions

Filtering

```
df[df['age'] > 25]
```

Multiple conditions:

```
df[(df['age'] > 25) & (df['salary'] > 50000)]
```

📌 Used in real-world dashboards and business analytics.

Adding & Dropping

◊ Add Column

```
df['new_col'] = value
```

◊ Drop Column

```
df.drop('column', axis=1)
```

◊ Handle Missing Values

```
df.dropna()  
df.fillna(0)
```

📌 Data cleaning is 60–70% of Data Science work.

GroupBy

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

Specific column:

```
df.groupby('column')['sales'].sum()
```

📌 Used for:

- Sales by region
- Revenue by category

- Customer segmentation

Sorting

```
df.sort_values('column')
df.sort_values('column', ascending=False)
```

☛ Used in ranking, leaderboards, reports.

Merging & Joining

◊ Merge (SQL JOIN)

```
pd.merge(df1, df2, on='id')
```

◊ Join

```
df1.join(df2)
```

☛ Very important for combining datasets.

Exporting

```
df.to_csv("output.csv", index=False)
df.to_excel("output.xlsx", index=False)
```

☛ Used for reporting and sharing results.

NumPy	Pandas
Numerical computing	Data manipulation
Arrays	DataFrames
Fast math operations	Data cleaning & analysis
Used in ML models	Used before ML models

Author

Ganesh B. Patil
+91 9022982252