Perfect — let’s make this **NumPy vs Pandas Cheat Sheet** so you can have everything in **one place** for quick interview prep and coding reference.

# 📝 **NumPy vs Pandas — Complete Cheat Sheet**

## **1. NumPy (Numerical Python)**

### **Import**

pythonCopyEditimport numpy as np

### **Core Data Structure**

* **ndarray**: N-dimensional array (homogeneous data type).

### **Creating Arrays**

pythonCopyEditnp.array([1, 2, 3]) # From list  
np.zeros((3, 3)) # 3x3 zeros  
np.ones((2, 2)) # 2x2 ones  
np.arange(0, 10, 2) # [0, 2, 4, 6, 8]  
np.linspace(0, 1, 5) # [0, 0.25, 0.5, 0.75, 1]  
np.eye(3) # Identity matrix

### **Array Operations**

pythonCopyEditarr = np.array([1, 2, 3, 4])  
arr + 10 # Add 10 to each element  
arr \* 2 # Multiply each element by 2  
np.sqrt(arr) # Square root  
np.log(arr) # Logarithm  
np.exp(arr) # Exponential

### **Indexing & Slicing**

pythonCopyEditarr[0] # First element  
arr[1:3] # Elements 1 and 2  
arr[-1] # Last element

### **Boolean Indexing**

pythonCopyEditarr[arr > 2] # Elements greater than 2

### **Matrix Operations**

pythonCopyEditA = np.array([[1, 2], [3, 4]])  
B = np.array([[5, 6], [7, 8]])  
  
A + B # Element-wise addition  
A @ B # Matrix multiplication  
A.T # Transpose  
np.linalg.inv(A) # Inverse  
np.linalg.det(A) # Determinant

### **Statistics**

pythonCopyEditarr.mean()  
arr.sum()  
arr.min()  
arr.max()  
arr.std()  
arr.var()

## **2. Pandas**

### **Import**

pythonCopyEditimport pandas as pd

### **Core Data Structures**

* **Series**: 1D labeled array.
* **DataFrame**: 2D labeled table.

### **Creating**

pythonCopyEdit# Series  
s = pd.Series([10, 20, 30], index=["a", "b", "c"])  
  
# DataFrame  
data = {"Name": ["A", "B"], "Age": [25, 30]}  
df = pd.DataFrame(data)

### **Reading & Writing Data**

pythonCopyEditpd.read\_csv("file.csv")  
pd.read\_excel("file.xlsx")  
pd.read\_json("file.json")  
  
df.to\_csv("file.csv", index=False)

### **Exploring Data**

pythonCopyEditdf.head() # First 5 rows  
df.tail() # Last 5 rows  
df.info() # Summary  
df.describe() # Stats summary  
df.shape # (rows, columns)  
df.columns # Column names  
df.dtypes # Data types

### **Selecting Data**

pythonCopyEditdf["Name"] # Single column  
df[["Name", "Age"]] # Multiple columns  
df.iloc[0] # First row (by index)  
df.loc[0] # First row (by label)  
df.iloc[0:3] # First 3 rows  
df.loc[:, "Name":"Age"] # Slice columns

### **Filtering**

pythonCopyEditdf[df["Age"] > 25]  
df[(df["Age"] > 25) & (df["Salary"] > 50000)]

### **Sorting**

pythonCopyEditdf.sort\_values(by="Age")  
df.sort\_values(by="Age", ascending=False)

### **Modifying Data**

pythonCopyEditdf["NewCol"] = df["Age"] \* 2 # New column  
df.drop("NewCol", axis=1, inplace=True) # Drop column  
df.rename(columns={"Name": "FullName"}, inplace=True)

### **Missing Data**

pythonCopyEditdf.isnull().sum() # Count missing  
df.dropna() # Drop missing rows  
df.fillna(0) # Fill missing with 0

### **Grouping & Aggregation**

pythonCopyEditdf.groupby("Department")["Salary"].mean()  
df.groupby("Dept").agg({"Salary": "sum", "Age": "max"})

### **Merging & Joining**

pythonCopyEditpd.merge(df1, df2, on="ID", how="inner")  
pd.concat([df1, df2], axis=0) # Stack rows  
pd.concat([df1, df2], axis=1) # Stack columns

### **Pivot Tables**

pythonCopyEditdf.pivot\_table(values="Salary", index="Dept", aggfunc="mean")

## **3. NumPy vs Pandas — Quick Recap Table**

| Feature | NumPy (ndarray) | Pandas (Series, DataFrame) |
| --- | --- | --- |
| Data Type | Homogeneous | Mixed types allowed |
| Data Shape | N-dimensional arrays | 1D & 2D labeled structures |
| Speed | Faster for pure math | Slightly slower (labels) |
| Use Case | Math, matrix ops | Data cleaning, analysis |
| Labels | ❌ No | ✅ Yes |
| File I/O | ❌ No | ✅ Yes |

I can also make you a **visual mindmap** showing how Pandas is built on top of NumPy and when to use each — that will make interview recall super easy.

Do you want me to prepare that next so you remember in **one glance**?