# 01 · Python for ML: NumPy & pandas

TutorialHub cheat sheet

## Python essentials

| Need | Code |
|---|---|
| list / dict | `xs = [3, 1, 2]` · `d = {"east": 119}` |
| comprehension | `[p * q for p, q in zip(prices, qty)]` · `[x for x in xs if x > 0]` |
| function | `def mean(xs): return sum(xs) / len(xs)` |
| f-string | `f"{value:.2f}"` |
| environment | `python3 -m venv ~/ml && source ~/ml/bin/activate && pip install -r requirements.txt` |

## NumPy

| Task | Code |
|---|---|
| create | `np.array([[1, 2], [3, 4]])` · `np.zeros((3, 4))` · `np.arange(10)` · `np.linspace(0, 1, 5)` |
| random (seeded) | `rng = np.random.default_rng(42); rng.normal(size=5)` |
| inspect | `X.shape` · `X.dtype` · `X.ndim` |
| select | `X[1]` row · `X[:, 2]` column · `X[X > 6]` mask · `X[[0, 2]]` rows 0 and 2 |
| reduce | `X.mean(axis=0)` per column · `X.sum(axis=1)` per row · `X.max()` everything |
| reshape | `a.reshape(3, 4)` · `a[:, None]` makes (n,) → (n,1) |
| matrix multiply | `A @ B` |
| copy | slices are views; `X[1:3].copy()` is a copy |

**Broadcasting:** compare shapes right to left; each pair must be equal or 1. `(3,4) - (4,)` ✓ · `(3,4) - (3,)` ✗ → use `(3,1)`.

## pandas

| Task | Code |
|---|---|
| load / save | `pd.read_csv("f.csv")` · `df.to_csv("out.csv", index=False)` · `df.to_parquet("out.parquet")` |
| inspect | `df.shape` · `df.info()` · `df.describe()` · `df.isna().sum()` · `df["c"].value_counts()` |
| select | `df["price"]` · `df[["a", "b"]]` · `df.loc[df.price > 10, "qty"]` · `df.iloc[0, :3]` |
| new column | `df = df.assign(revenue=df.price * df.qty)` |
| missing | `df.dropna(subset=["customer_id", "price"])` · `df["coupon"].fillna("none")` |
| group | `df.groupby("month")["revenue"].agg(["count", "sum", "mean"])` |
| join | `orders.merge(customers, on="customer_id", how="left")` |
| sort | `df.sort_values("revenue", ascending=False)` |

| how= | keeps |
|---|---|
| inner | keys in both tables |
| left | every left row (NaN where no match) |
| right | every right row |
| outer | everything |

## Plots (Matplotlib)

| Question | Code |
|---|---|
| one column's shape | `ax.hist(x, bins=30)` |
| two columns | `ax.scatter(x, y, s=6)` |
| over time | `ax.plot(t, y)` |
| save | `fig.savefig("plot.png", dpi=120)` |

## Traps

- `X.mean()` averages everything; features need `X.mean(axis=0)`.
- `df.dropna()` drops rows with NaN in **any** column — name the required ones.
- An inner join silently drops unmatched rows; print `len(df)` before and after every merge.

- Chained assignment `df[mask]["c"] = v` may not stick — use `df.loc[mask, "c"] = v`.
- Notebook results only count after **Restart & Run All**.