**Task 1:**

* **BeautifulSoup** – Great for beginners and simple tasks. Pairs well with requests to pull HTML and then parse it.
* **Scrapy** – More advanced. It’s a full-fledged framework ideal for large-scale scraping and complex sites.
* **Selenium** – Useful if you need to interact with JavaScript-heavy pages (e.g., clicking buttons or scrolling).

**import requests**

**from bs4 import BeautifulSoup**

**url = "https://example.com"**

**response = requests.get(url)**

**soup = BeautifulSoup(response.text, 'html.parser')**

**for title in soup.find\_all('h2'):**

**print(title.get\_text())**

**Task 2:**

**Goals of EDA**

* **Ask Questions**: What are we trying to find out?
* **Understand the Dataset**: What do the variables represent? Any missing values or outliers?
* **Spot Trends & Patterns**: Are there correlations? Clusters? Seasonal trends?
* **Validate Assumptions**: Are numeric variables normally distributed? Are categories balanced?
* **Clean the Data**: Often EDA and data cleaning go hand-in-hand.

**import pandas as pd**

**import seaborn as sns**

**import matplotlib.pyplot as plt**

**# Load your data**

**df = pd.read\_csv("your\_data.csv")**

**# Quick overview**

**print(df.info())**

**print(df.describe())**

**# Check for missing values**

**print(df.isnull().sum())**

**# Visualize distributions**

**sns.histplot(df['column\_name'])**

**plt.show()**

**# Correlation heatmap**

**sns.heatmap(df.corr(), annot=True, cmap="coolwarm")**

**plt.show()**

**Task 3:**

 The brain processes **visuals 60,000x faster** than text.

 Good visuals reveal **patterns, trends, and outliers** instantly.

 They help **simplify complex data** for non-technical audiences.

### Types of Visuals & When to Use Them:

| **Chart Type** | **Best For** |
| --- | --- |
| Line Chart | Trends over time |
| Bar Chart | Comparing categories |
| Histogram | Distribution of numeric data |
| Scatter Plot | Correlations and relationships |
| Box Plot | Spread and outliers in data |
| Heatmap | Correlation matrices, intensities |
| Pie Chart (sparingly) | Showing part-to-whole ratios (simple data) |

**Design Principles for Impact:**

* **Clarity > Complexity**: Avoid clutter; every element should serve a purpose.
* **Color Wisely**: Use consistent color schemes and ensure accessibility (e.g., colorblind-friendly palettes).
* **Tell a Story**: Highlight what matters — use titles, annotations, and tooltips if needed.
* **Interactive Visuals**: Libraries like Plotly, Altair, or dashboards with Streamlit/Power BI can really bring your story to life.

**Libraries to Try in Python:**

* **Matplotlib / Seaborn** – static, clean visuals for EDA
* **Plotly** – interactive plots with hover/click features
* **Altair** – declarative syntax, great for rapid exploration
* **Dash or Streamlit** – build interactive dashboards to share