**Use Case: Design an application for Simulating Random Coin-Flips and Dice-Rolls Using NumPy**

**Problem Statement:**

* Design a Python application that simulates **random coin flips** and **dice rolls** using the **NumPy** library.
* The objective is to understand how randomness works in probability-based experiments and visualize the distribution of outcomes using **Matplotlib**.

**Solution:**

In probability theory:

* A **coin flip** has **two possible outcomes**:
  + Head (H)
  + Tail (T)  
    Each outcome has an **equal probability of 0.5** in a fair coin.
* A **dice roll** has **six possible outcomes** (1 to 6).  
  Each outcome has an **equal probability of 1/6 ≈ 0.1667** for a fair die.

To simulate randomness:

* **NumPy** provides np.random.randint() to generate random integers.
* Large numbers of trials (e.g., 1000 or more) make the results approach the **expected theoretical probabilities** due to the **law of large numbers**.

Finally, **Matplotlib** can be used to plot the outcomes, making it easier to visualize how randomness distributes over many trials.

**Python Program:**

import numpy as np

import matplotlib.pyplot as plt

# Number of experiments

num\_flips = 1000

num\_rolls = 1000

# --- Coin Flip Simulation ---

# 0 represents Tails, 1 represents Heads

coin\_flips = np.random.randint(0, 2, num\_flips)

# Counting outcomes

heads = np.count\_nonzero(coin\_flips == 1)

tails = np.count\_nonzero(coin\_flips == 0)

# --- Dice Roll Simulation ---

# Values between 1 and 6

dice\_rolls = np.random.randint(1, 7, num\_rolls)

# Counting occurrences of each face

dice\_counts = np.bincount(dice\_rolls)[1:] # ignore index 0

# --- Visualization using Matplotlib ---

plt.figure(figsize=(10, 4))

# Plot for Coin Flips

plt.subplot(1, 2, 1)

plt.bar(['Heads', 'Tails'], [heads, tails], color=['gold', 'lightblue'], edgecolor='black')

plt.title("Coin Flip Simulation (1000 flips)")

plt.ylabel("Count")

# Plot for Dice Rolls

plt.subplot(1, 2, 2)

plt.bar(np.arange(1, 7), dice\_counts, color='lightgreen', edgecolor='black')

plt.title("Dice Roll Simulation (1000 rolls)")

plt.xlabel("Dice Face")

plt.ylabel("Count")

plt.tight\_layout()

plt.show()

# Display results

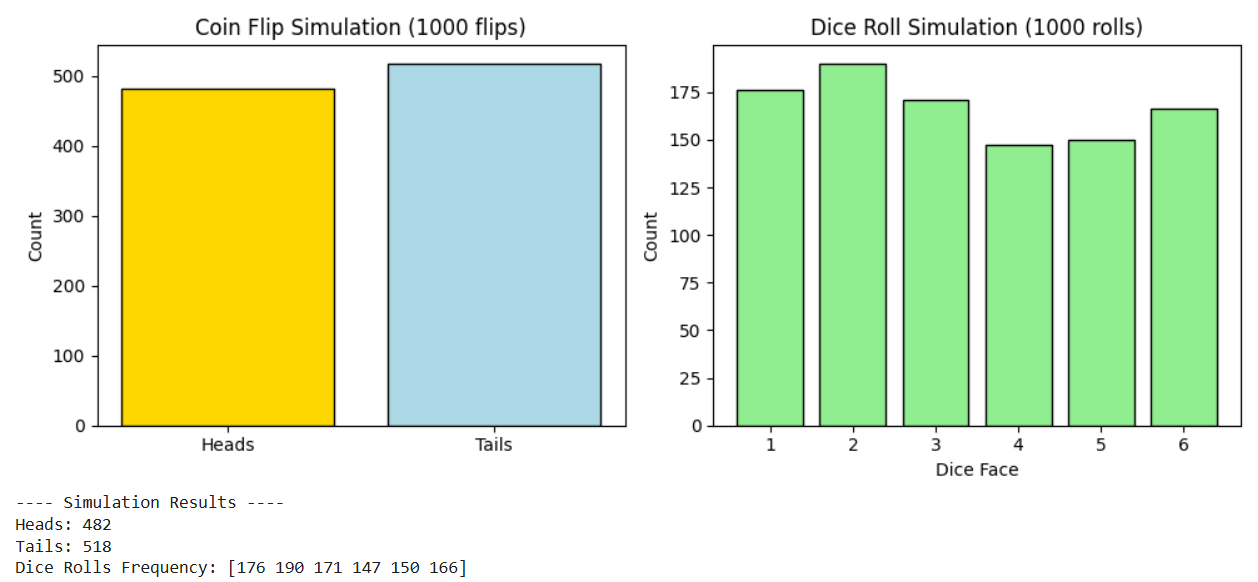
print("---- Simulation Results ----")

print(f"Heads: {heads}")

print(f"Tails: {tails}")

print("Dice Rolls Frequency:", dice\_counts)

**Output:**

****