# 1. Dice Roller Simulation

This Python program simulates rolling a pair of dice, with added functionality to specify the number of dice to roll. Each time the program runs, it randomly generates the results of the dice rolls and displays them to the user. The program keeps track of how many times the user has rolled the dice during the session and provides a user-friendly experience.

**Features:**

* Simulates rolling any number of dice.
* Generates random results for each die roll.
* Keeps count of the total number of rolls in a session.
* User-friendly prompts and error handling.

**How to Use:**

1. Run the program.
2. Enter 'y' to roll the dice.
3. Specify the number of dice to roll.
4. View the results and the total count of rolls.
5. Repeat or exit by entering 'n'.

**Example:**

Python

Roll the dice? (y/n): y

How many dice would you like to roll? 2

You rolled: 3, 5

Total rolls so far: 1

Roll the dice? (y/n): n

Thank you for playing!

You rolled the dice 1 times during this session.

**Requirements:**

* Python 3.x

**Author:** Abhijeet Patil

# 2. Enhanced Number Guessing Game

This Python program is an enhanced version of the classic number-guessing game. The computer randomly selects a number within a user-defined range, and the player attempts to guess the number. The program hints if the guess is too high or too low, and includes several additional features to enhance the gameplay experience.

**Features:**

1. **Customizable Number Range**: Players can specify the minimum and maximum values for the number range before the game starts, allowing for customizable difficulty levels.
2. **Limited Guesses**: The game limits the number of guesses a player can make. If the player runs out of attempts, the correct number is revealed.
3. **Best Score Tracking**: The program keeps track of the fewest attempts it took to guess the number correctly and displays the "best score" at the end of each game.

**How to Play:**

1. Run the program.
2. Enter the minimum and maximum values for the number range.
3. Enter the maximum number of guesses allowed.
4. Start guessing the number within the specified range.
5. Receive hints if the guess is too high or too low.
6. Continue guessing until the number is correctly guessed or the maximum attempts are reached.
7. View your score and the best score so far.
8. Option to play again or exit the game.

**Example:** Python

Welcome to the Enhanced Number Guessing Game!

Enter the minimum value for the number range: 1

Enter the maximum value for the number range: 100

Enter the maximum number of guesses allowed: 10

I have selected a number between 1 and 100. Can you guess what it is? You have 10 attempts.

Attempt 1: Enter your guess: 50

Too high! Try again.

Attempt 2: Enter your guess: 25

Too low! Try again.

...

Congratulations! You guessed the correct number in 6 attempts.

Your best score so far is 6 attempts.

Do you want to play again? (y/n): n

Thank you for playing!

**Requirements:**

* Python 3.x

**Author:** Abhijeet Patil

# 3. Rock, Paper, Scissors Game

This Python program simulates the classic game of Rock, Paper, Scissors. It offers multiple game modes, including single-player vs. computer and two-player modes. The game keeps track of the number of wins, losses, and ties, and declares the overall winner in a best-of-three format. Additionally, the program includes an option to restart the game for a new session.

**Features:**

1. **Game Modes**:
   * Single Player vs. Computer
   * Two Players
2. **Best-of-Three Rounds**:
   * The first player (or computer) to win two out of three rounds is declared the overall winner.
3. **Statistics Tracking**:
   * Keeps a tally of wins, losses, and ties for both players.
   * Displays detailed game statistics at the end of each session.
4. **Game Restart Option**:
   * Players can choose to restart the game entirely for a new session.

**How to Play:**

1. Run the program.
2. Choose the game mode: Single Player vs. Computer or Two Players.
3. Players take turns to input their choices (rock, paper, or scissors).
4. The program displays the choices using emojis and determines the winner of each round.
5. Continue playing until a best-of-three winner is declared.
6. View the game statistics and choose to restart or exit.

**Example:** Python

Welcome to Rock, Paper, Scissors!

Choose the game mode: (1) Single Player vs Computer, or (2) Two Players: 1

Player 1, enter your choice (r for rock, p for paper, s for scissors): r

Player 1 choose: 🪨 Rock

Computer choose: 📃 Paper

Computer wins this round!

...

Player 1 is the overall winner!

Game Over! Here are the stats: Player 1 wins: 2, Computer wins: 1, Ties: 0

Thank you for playing!

Do you want to restart the game? (y/n): y

**Requirements:**

* Python 3.x

**Author:** Abhijeet Patil

# 4. QR Code Generator

This Python program allows users to generate QR codes based on input text or URLs. It provides options to customize the color of the QR code and background to match user style or branding. Additionally, users can generate multiple QR codes at once by providing a list of texts or URLs, each saved with a unique filename.

**Features:**

1. **Customizable Colors**:
   * Users can choose the color of the QR code and the background color.
2. **Multiple QR Code Generation**:
   * Generate multiple QR codes at once by providing a list of texts or URLs.
   * Each QR code is saved with a user-specified unique filename.

**How to Use:**

1. Run the program.
2. Enter the desired color for the QR code and the background color.
3. Input the text or URLs to generate QR codes, separated by commas.
4. Provide filenames for each QR code, separated by commas.
5. The program will generate the QR codes and save them with the specified filenames.

**Example:**

Welcome to the QR Code Generator!

Enter the color for the QR code (e.g., 'black', 'blue', 'red'): blue

Enter the background color for the QR code (e.g., 'white', 'yellow'): white

Enter the text or URLs to generate QR codes (separated by commas): https://example.com, Hello World, https://another-example.com

Enter the filenames for each QR code (separated by commas): example1.png, hello\_world.png, another\_example.png

QR code saved as example1.png

QR code saved as hello\_world.png

QR code saved as another\_example.png

All QR codes have been generated successfully!

**Requirements:**

* Python 3.x
* qrcode and Pillow libraries

**Installation:**

bash

pip install qrcode[pil]

**Author:** Abhijeet Patil

# 5. Currency Converter

This Python program is a comprehensive currency converter that allows users to convert money from one currency to multiple currencies simultaneously using fixed exchange rates. It includes an expanded list of available currencies and maintains a history of conversions made during the session.

### Features:

1. **Multi-Currency Conversion**:
   * Convert an amount from one currency to several different currencies simultaneously.
2. **Expanded Currency List**:
   * Includes additional currencies such as EUR, JPY, GBP, INR, CAD, AUD, CHF, CNY, and SEK.
3. **Conversion History**:
   * Keeps a history of the most recent conversions made during the session and displays it at the end.

### How to Use:

1. Run the program.
2. Enter the amount to convert.
3. Input the currency to convert from (e.g., USD, EUR, JPY, GBP, INR).
4. The program will display the equivalent amount in multiple currencies.
5. Continue with additional conversions or exit the program to view the conversion history.

### Example:

Currency Converter

Enter the amount to convert: 100

Enter the from currency (e.g., USD, EUR, JPY, GBP, INR): USD

100.0 USD is equivalent to:

96.00 EUR

15326.00 JPY

80.00 GBP

8677.00 INR

142.00 CAD

159.00 AUD

91.00 CHF

731.00 CNY

1080.00 SEK

Do you want to perform another conversion? (y/n): n

Conversion History:

100.00 USD = 96.00 EUR

100.00 USD = 15326.00 JPY

100.00 USD = 80.00 GBP

100.00 USD = 8677.00 INR

100.00 USD = 142.00 CAD

100.00 USD = 159.00 AUD

100.00 USD = 91.00 CHF

100.00 USD = 731.00 CNY

100.00 USD = 1080.00 SEK

### Requirements:

* Python 3.x

### Author: Abhijeet Patil

# 6. Quiz Game

This Python program is a flexible and interactive quiz game that allows users to answer multiple-choice questions from various categories and difficulty levels. The questions are loaded from a JSON file, making it easy to update and add new content.

### Features:

1. **Multiple Categories**: Choose from different categories of questions, such as history, science, and general knowledge.
2. **Difficulty Levels**: Select the difficulty level (easy, medium, hard) to match your knowledge and challenge yourself.
3. **Shuffled Questions**: Questions are shuffled to add randomness to the quiz.
4. **Input Validation**: Ensures that users provide valid inputs for category, difficulty, and answers.
5. **Exit Option**: Provides an option to exit the quiz at any point.

### How to Use:

1. Run the program.
2. Enter the name of the JSON file with the quiz questions (e.g., 'questions.json').
3. Choose a category from the available options.
4. Select a difficulty level from the available options.
5. Answer the multiple-choice questions.
6. Your final score will be displayed at the end of the quiz.

### Example:

Welcome to the Quiz Game!

Enter the name of the JSON file with the quiz questions (e.g., 'questions.json'): questions.json

Available categories: history, science, general\_knowledge, geography

Choose a category: science

Available difficulty levels: easy, medium, hard

Choose a difficulty level: medium

What is the atomic number of carbon?

A. 6

B. 8

C. 12

D. 14

Your answer (A/B/C/D) or 'Q' to quit: A

Correct!

Your final score is: 1/1

Thank you for playing the quiz!

### Requirements:

* Python 3.x

### Author: Abhijeet Patil