# Rock, Paper, Scissors Project

To display some of my programming language and skills, I decided to go through a step-by-step breakdown of how to create and run this programme. I will cover the key components and logic used in the implementation of this game which can be enjoyed by family and friends.

**1. Importing Required Modules**

Firstly, I added the import.random function. This allows the computer to generate random choices for the game.

A close-up of a word

Description automatically generated

**2. Defining the User Functions**

Next, I defined the user functions. Using the input function, the game will prompt the user to make a choice between ‘r’ (rock), ‘p’ (paper), or ’s’ (scissors). With the lower function, the game will automatically change the letter casing to lower-case. This is in-case the user inserts an upper case letter.

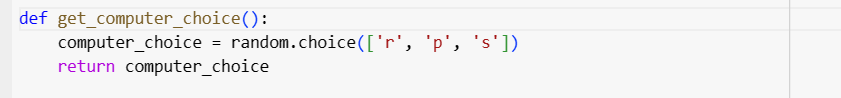
I used the while True function to ensure the user chooses the correct choice and does not input the wrong letter by accident (typo). This is done by inserting the if and else statement as shown below.

A screenshot of a computer

Description automatically generated

**3. Defining the Computer Functions**

Below is a screenshot of the next step. I am now defining the computer choice by using the def function in python.



This function randomly selects a choice from an assigned list for the computer using random.choice ( ['r', 'p', 's'] ).

**4. Determining the Winner Function**

Using the determine\_winner()` function in python will allow the game to determine a winner based on the choice of the user and the computer generated choice. There is also a return function which will help return the statement – It’s a tie! If the user and computer make the same choice, or either “You win! or, Computer wins!”

A screenshot of a computer program

Description automatically generated

This part of the game takes the user's choice (`user\_choice`) and the computer's choice (`computer\_choice`) as python arguments.

Python compares the choices based on the rules of Rock, Paper, Scissors to determine the winner:

- Rock (`'r'`) beats Scissors (`'s'`).

- Paper (`'p'`) beats Rock (`'r'`).

- Scissors (`'s'`) beats Paper (`'p'`).

- If both players choose the same item, it's a tie.

**5. The Main Game Function**

This function is the main game loop and is essential for the game to run:

def play\_game() is the function used to define how the game runs, which is shown below. To start playing the game, use the play\_game() function to run the game after you have passed the codes that define the game, which is also shown below at the end of the codes in the screenshot.

The game is designed to commence with a welcome message that then prompts the user to make a choice out of the 3 specific options. The game then displays the computer’s choice followed by the results. The game will ask if you would like to play again after the end of each result. The game ends when the user chooses no when prompted with the following question: “Do you want to play again? (yes/no)”

A screenshot of a computer program

Description automatically generated

**Time to Play**

1. Run the Python script.

2. Follow the prompt to choose rock (`r`), paper (`p`), or scissors (`s`) by entering the corresponding letter.

3. The computer will randomly choose its move.

4. The game will determine the winner based on the choices and display the result.

5. After each round, you can choose to play again or exit the game.

This step-by-step explanation covers the essential components of a basic Rock, Paper, Scissors game in Python. Feel free to modify and expand upon this implementation to add features such as keeping score, improving user interface, or implementing more complex game strategies!

Thanks

Yemi Atta

Below, I tested the game and challenged the computer. On my first attempt, I finally defeated the computer on my 6th attempt, which happens to be my favourite number.

Please leave a comment and let me know how many attempts it took you to defeat the computer.

A screenshot of a computer program

Description automatically generated