Rock, Paper, Scissors Game

Welcome to our interactive Python notebook where you'll learn to create a simple yet fun game: Rock, Paper, Scissors! This game is not only entertaining but also a great way to practice basic programming concepts.

```
import random

def play():
    user_choice = input("Choose 'r' for rock, 'p' for paper, 's' for scissors:\n")
    computer_choice = random.choice(['r', 'p', 's'])

    if user_choice == computer_choice:
        return "It's a tie!"

    if is_win(user_choice, computer_choice):
        return 'You won!'

    return 'You lost!'

def is_win(player, opponent):
    # Return True if player wins
    if ((player == 'r' and opponent == 's') or
        (player == 'p' and opponent == 'r') or
        (player == 's' and opponent == 'p')):
        return True
    return False

print(play())
```

How to Run the Code

To execute the code in the cell above and start the game:

1. **Click** inside the code cell (the cell with the Python code).
2. Look for the **Run** button in the toolbar at the top of this notebook. It looks like a play symbol (▶).
3. **Click** the **Run** button.

The game will then ask for your input directly below the cell. Type your choice ('r' for rock, 'p' for paper, 's' for scissors) and press **Enter**.

If you make a mistake or want to play again, simply click the **Run** button again to restart the game. Enjoy!

Example Game Playthrough

- **Prompt**: Choose 'r' for rock, 'p' for paper, 's' for scissors:

- **User Input**: `r`
- **Program Output**: You won! (Assuming the computer chose scissors.)

This example demonstrates a single round of the game, showcasing the possible interaction and outcome based on the user's choice.

```python
# Detailed Code Explanation:
# This section reiterates the game's code with added detailed commentary
# to elaborate on the programming logic and design decisions. This is an
# excellent opportunity to delve deeper into understanding Python.

import random

def play():
    # User makes a choice from 'r', 'p', 's'. We capture this choice here.
    user_choice = input("Choose 'r' for rock, 'p' for paper, 's' for scissors:\n")

    # The computer's choice is made randomly from the same set of options.
    computer_choice = random.choice(['r', 'p', 's'])

    # Compare choices to check for a tie.
    if user_choice == computer_choice:
        return "It's a tie!"

    # Use the is_win function to determine if the user wins.
    if is_win(user_choice, computer_choice):
        return 'You won!'

    # If it's not a tie and the user didn't win, then the user loses.
    return 'You lost!'

def is_win(player, opponent):
    # Winning conditions:
    # Rock ('r') beats Scissors ('s'), Scissors ('s') beat Paper ('p'),
    # and Paper ('p') beats Rock ('r').
    if ((player == 'r' and opponent == 's') or
        (player == 'p' and opponent == 'r') or
        (player == 's' and opponent == 'p')):
        return True
    return False

# Start the game
print(play())
```

Understanding the Game Logic

- **Making Choices**: Both the player and the computer make a choice. The player's choice is made through input, while the computer's choice is randomly generated.
- **Determining the Outcome**: The game's outcome is determined by comparing the choices. The rules are straightforward: Rock crushes scissors, scissors cut paper, and paper covers rock.

- **Winning Logic**: We use a separate function `is_win` to simplify understanding the conditions under which the player wins against the computer.