



Python Game Programming: Rock Paper Scissors

In this tutorial, you are going to make a “Rock Paper Scissors” game. Below are the game rules:

- Rock smashes scissors.
- Paper covers rock.
- Scissors cut paper.

Step 1: Open a text editor

Step 2: Use the import statement

```
import random
```

The import statement loads module contents for later access and use. When an import statement is executed, the standard built-in `__import__()` function is called. The random module is a built-in module to generate random variables. We are going to use it to make computer choose a random value among rock, paper, and scissors each game we play.

Step 3: Print introduction line

```
print("Welcome to Rock, Paper, Scissors!\n")
```

Make sure to use double quotes because strings must be enclosed in double quotes for the data to be recognized as a string, not a number or variable name.

A backslash n (`\n`) is called a new line and used inside the print statement to create a new line as its name indicates.

Step 4: Set win counters

```
player_wins = 0
```

```
computer_wins = 0
```

Assign a value of 0 to win counter variables.

Step 5: Use While statement

```
while True:  
    player = input("Enter a choice (rock, paper, scissors): ")  
    choices = ["rock", "paper", "scissors"]  
    computer = random.choice(choices)  
    print(f"\nYou chose {player}, computer chose {computer}.")
```

While True means loop forever. In other words, the while loop will run as long as the While condition is True. You can force the loop to end by using the break statement.

Create an input variable where players can enter a value among rock, paper, scissors.

Create a list containing the rock, paper, scissors values with brackets. We will call the list "choices".

Then, we are going to use the random module here. Create a random.choice variable. The choice() function here is used to randomly select an item from a list.

Print the game status information. The f with curly braces is called f-strings, and it lets you include the value of Python expressions inside a string.

Step 6: Use If statement

```
if player == computer:  
    print(f"Both players selected {player}. It is a tie!")  
elif player == "rock":  
    if computer == "scissors":  
        print("Rock smashes scissors. You win!")  
        player_wins+=1  
    else:  
        print("Paper covers rock. You lose.")  
        computer_wins+=1
```

```

elif player == "paper":
    if computer == "rock":
        print("Paper covers rock. You win!")
        player_wins+=1
    else:
        print("Scissors cuts paper. You lose.")
        computer_wins+=1

elif player == "scissors":
    if computer == "paper":
        print("Scissors cuts paper. You win!")
        player_wins+=1
    else:
        print("Rock smashes scissors. You lose.")
        computer_wins+=1

```

Note that an if block can go under another if block.

Step 7: Print the win counters

```

print("You have "+str(player_wins)+" wins")
print("The computer has "+str(computer_wins)+" wins")

```

The str() function converts the specified value into a string. Instead of the str() function, you can use f-strings as follows:

```

print(f"You have {player_wins} wins")
print(f"The computer has {computer_wins} wins")

```

Note. See the difference between the str() block and the f-string block. Two pairs of double quotes and two + operators are used only in the str() block. This is called String Concatenation, which is the process of merging two or more strings. Since the win counters are not originally strings, we have to separate the strings *You have* and *wins* and concatenate them.

Step 8: Ask to quit or play again

```
repeat = input("\nPlay again? (yes/no): ")

if repeat.lower() != "yes":

    print("Thanks for playing!")

    break
```

Create a repeat variable where players can enter a value between yes or no. The lower() method returns the lowercase string from the given string.

The exclamation mark with an equal sign (!=) is called not equal to operator. If players enter any value other than yes, the while statement stops operating because of the following break statement.

The break statement terminates the current loop. It can be used in both while and for loops.

Congratulations! You've just built a rock-paper-scissors game in Python.

Below is the entire code:

```
import random

print("Welcome to Rock, Paper, Scissors!\n")

player_wins = 0

computer_wins = 0

while True:

    player = input("Enter a choice (rock, paper, scissors): ")

    choices = ["rock", "paper", "scissors"]

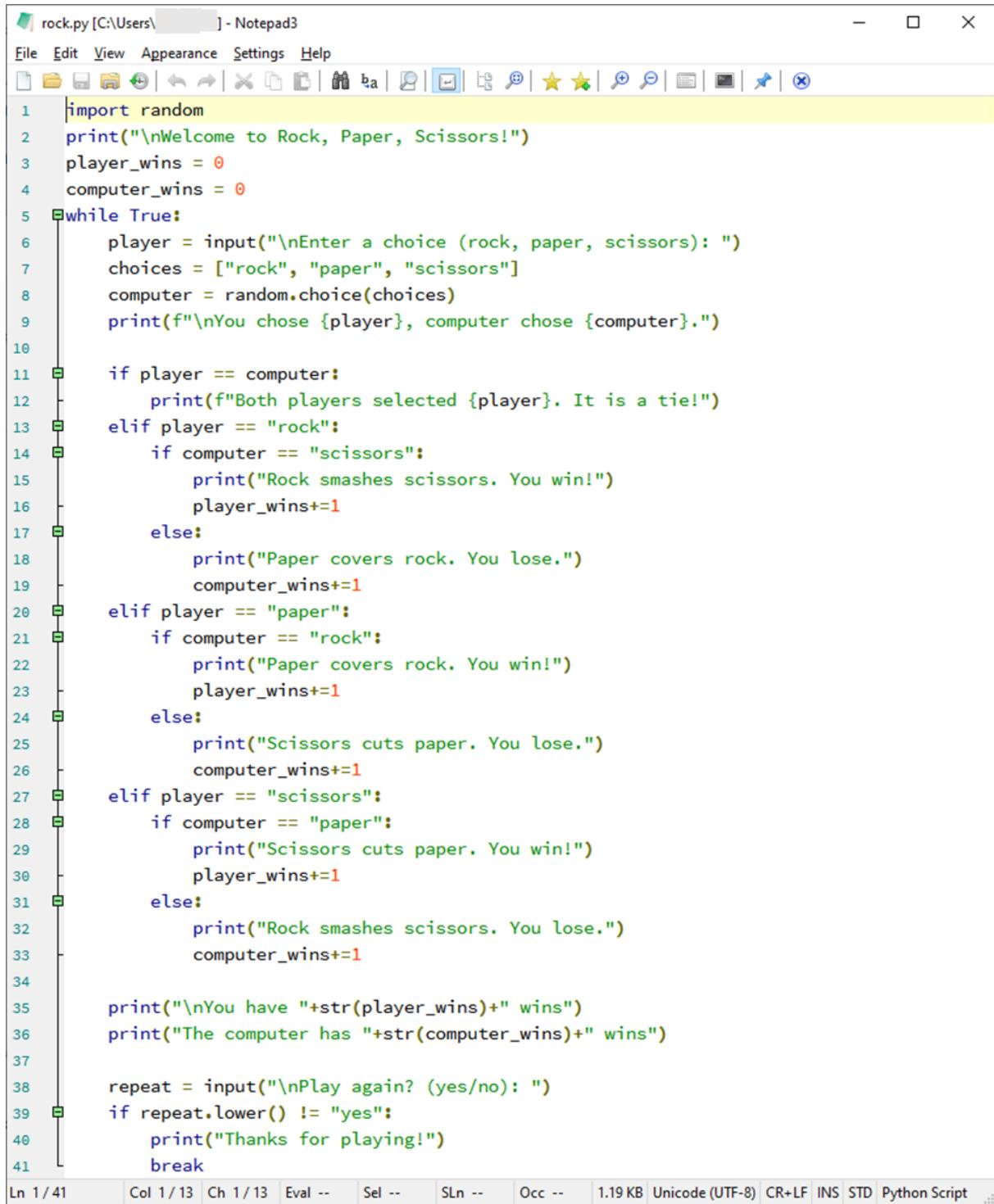
    computer = random.choice(choices)

    print(f"\nYou chose {player}, computer chose {computer}.")
```

```
if player == computer:  
    print(f"Both players selected {player}. It is a tie!")  
  
elif player == "rock":  
    if computer == "scissors":  
        print("Rock smashes scissors. You win!")  
        player_wins+=1  
  
    else:  
        print("Paper covers rock. You lose.")  
        computer_wins+=1  
  
elif player == "paper":  
    if computer == "rock":  
        print("Paper covers rock. You win!")  
        player_wins+=1  
  
    else:  
        print("Scissors cuts paper. You lose.")  
        computer_wins+=1  
  
elif player == "scissors":  
    if computer == "paper":  
        print("Scissors cuts paper. You win!")  
        player_wins+=1  
  
    else:  
        print("Rock smashes scissors. You lose.")  
        computer_wins+=1  
  
print("You have "+str(player_wins)+" wins")  
print("The computer has "+str(computer_wins)+" wins")
```

```
repeat = input("\nPlay again? (yes/no): ")
if repeat.lower() != "yes":
    print("Thanks for playing!")
```

Break



The screenshot shows a window titled "rock.py [C:\Users\] - Notepad3". The window contains a Python script for a Rock, Paper, Scissors game. The code is color-coded: green for strings, blue for keywords, and red for numbers. The script starts by importing random and printing a welcome message. It initializes player_wins and computer_wins to 0. A while loop continues until the user types "no". Inside the loop, it prompts the user for their choice and generates a random choice for the computer. It then prints a message indicating the result and updates the win counts. Finally, it prints the total wins for both player and computer, and asks if the user wants to play again.

```
rock.py [C:\Users\ ] - Notepad3
File Edit View Appearance Settings Help
1 import random
2 print("\nWelcome to Rock, Paper, Scissors!")
3 player_wins = 0
4 computer_wins = 0
5 while True:
6     player = input("\nEnter a choice (rock, paper, scissors): ")
7     choices = ["rock", "paper", "scissors"]
8     computer = random.choice(choices)
9     print(f"\nYou chose {player}, computer chose {computer}.")
10
11    if player == computer:
12        print(f"Both players selected {player}. It is a tie!")
13    elif player == "rock":
14        if computer == "scissors":
15            print("Rock smashes scissors. You win!")
16            player_wins+=1
17        else:
18            print("Paper covers rock. You lose.")
19            computer_wins+=1
20    elif player == "paper":
21        if computer == "rock":
22            print("Paper covers rock. You win!")
23            player_wins+=1
24        else:
25            print("Scissors cuts paper. You lose.")
26            computer_wins+=1
27    elif player == "scissors":
28        if computer == "paper":
29            print("Scissors cuts paper. You win!")
30            player_wins+=1
31        else:
32            print("Rock smashes scissors. You lose.")
33            computer_wins+=1
34
35    print("\nYou have "+str(player_wins)+" wins")
36    print("The computer has "+str(computer_wins)+" wins")
37
38    repeat = input("\nPlay again? (yes/no): ")
39    if repeat.lower() != "yes":
40        print("Thanks for playing!")
41        break
```

```
C:\ Command Prompt
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ >python rock.py

Welcome to Rock, Paper, Scissors!

Enter a choice (rock, paper, scissors): rock

You chose rock, computer chose scissors.
Rock smashes scissors. You win!

You have 1 wins
The computer has 0 wins

Play again? (yes/no): yes

Enter a choice (rock, paper, scissors): paper

You chose paper, computer chose scissors.
Scissors cuts paper. You lose.

You have 1 wins
The computer has 1 wins

Play again? (yes/no): yes

Enter a choice (rock, paper, scissors): scissors

You chose scissors, computer chose scissors.
Both players selected scissors. It is a tie!

You have 1 wins
The computer has 1 wins

Play again? (yes/no): no
Thanks for playing!

C:\Users\ >
```

Written by Ryan Clow, Digital Scholarship Training Specialist, 02/2023

This work is licensed under the Creative Commons Attribution 4.0 International License.
To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0>

