## Follow the instructions to make a simple, one round rock, paper, scissors game!

**Make sure to test your game thoroughly before submitting the link :)**

### Step 1: Set Up Your Python File

1. **Open your Python editor**.
2. **Start with a comment** describing the game.

```
# Rock, Paper, Scissors Game
# This program lets the player compete against the computer in a simple game.
```

### Step 2: Import the Random Module

Since the computer will make a **random selection**, import the `random` module at the top of your program after the inital comment

```
import random
```

### Step 3: Define a List for the Choices

- The game has **three valid options**: `"rock"`, `"paper"`, and `"scissors"`.
  Create a list to store these choices.

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

### Step 4: Get Player Input

- Prompt the player to **enter their choice**.
- Store the input in a variable called `player_choice`.

```
player_choice = input("Enter rock, paper, or scissors: ")
```

### Step 5: Convert Input to Lowercase

- The game should recognize **all versions** of `"rock"`, `"paper"`, and `"scissors"`, even if the player types `"ROCK"` or `"Paper"`.
- Use `.lower()` to convert the input to **lowercase**, making the game more user-friendly.

```
player_choice = input("Enter rock, paper, or scissors: ").lower()
```

### Step 6: Validate Player Input

- Check if the user **entered a valid choice**.
- If they enter something **other than rock, paper, or scissors**, print an error message and **exit the program**.
- Use `quit()` to **immediately stop the program** when an invalid choice is entered.

```
if player_choice == "rock":
    pass  # This allows the program to continue
elif player_choice == "paper":
    pass  # This allows the program to continue
elif player_choice == "scissors":
    pass  # This allows the program to continue
else:
    print("Invalid choice! Please enter rock, paper, or scissors.")
    quit()
```

### Step 7: Generate the Computer's Choice

- Use `random.randint()` to generate a **random index** from the list.
- Access the **choice using the index**.

```
computer_index = random.randint(0, 2)
computer_choice = choices[computer_index]
```

### Step 8: Print Both Choices

- Print out the user's choice and computer's choice

```
print("You chose:", player_choice)
print("Computer chose:", computer_choice)
```

### Step 9: Determine the Winner

- Compare the player's choice with the computer's choice.
- **Winning conditions:**
  - `"rock"` beats `"scissors"`
  - `"paper"` beats `"rock"`
  - `"scissors"` beats `"paper"`
- Use `if-elif-else` to determine the winner.
- Use `and` **checks if both conditions are** `True`

```
if player_choice == computer_choice:
    print("It's a tie!")
elif player_choice == "rock" and computer_choice == "scissors" :
    print("You win!")
elif player_choice == "paper" and computer_choice == "rock":
    print("You win!")
elif player_choice == "scissors" and computer_choice == "paper":
    print("You win!")
else:
    print("Computer wins!")
```

### Step 10: Optimize your code

- Looking at **Step 9**, we see that **three** `elif` **statements do the same thing** (`print("You win!")`).
- The keyword `or` allows us to check **multiple conditions in one line**.
- If **any** of the conditions are `True`, the **whole condition is** `True`.
- Use `\` to **break up long lines** into **multiple lines** for better readability.

```
if player_choice == computer_choice:
    print("It's a tie!")
elif (player_choice == "rock" and computer_choice == "scissors") or \
    (player_choice == "paper" and computer_choice == "rock") or \
    (player_choice == "scissors" and computer_choice == "paper"):
    print("You win!")
else:
    print("Computer wins!")
```

### Final Step: Test and Submit Your Code 🎮

Great job completing your Rock, Paper, Scissors game! 🚀 Before submitting, **make sure your game works correctly** by testing different cases:

✅ **Valid Inputs:**

- Enter `"rock"`, `"paper"`, and `"scissors"` → **Should play the game normally**

✅ **Invalid Inputs:**

- Enter `"hello"` or `"banana"` → **Should print an error message**

✅ **Winning and Losing Scenarios:**

- Make sure **all possible matchups** work correctly:
  - `"rock"` vs `"scissors"` → **You win**
  - `"paper"` vs `"rock"` → **You win**
  - `"scissors"` vs `"paper"` → **You win**
  - `"rock"` vs `"paper"` → **Computer wins**
  - `"paper"` vs `"scissors"` → **Computer wins**
  - `"scissors"` vs `"rock"` → **Computer wins**
  - `"rock"` vs `"rock"` → **Tie**
  - `"paper"` vs `"paper"` → **Tie**
  - `"scissors"` vs `"scissors"` → **Tie**

Your complete program should look like this:

```python
# Rock, Paper, Scissors Game
# This program lets the player compete against the computer in a simple game.
import random  # Import the random module

# Define possible choices
choices = ["rock", "paper", "scissors"]

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

# Get player choice
player_choice = input("Enter rock, paper, or scissors: ").lower()

if player_choice == "rock":
    pass  # This allows the program to continue
elif player_choice == "paper":
    pass  # This allows the program to continue
elif player_choice == "scissors":
    pass  # This allows the program to continue
else:
    print("Invalid choice! Please enter rock, paper, or scissors.")
    quit()

# Get computer choice using random index
computer_index = random.randint(0, 2)
computer_choice = choices[computer_index]  # Access choice from list

# Display choices
print("You chose:", player_choice)
print("Computer chose:", computer_choice)

# Determine the winner
if player_choice == computer_choice:
    print("It's a tie!")
elif (player_choice == "rock" and computer_choice == "scissors") or \
    (player_choice == "paper" and computer_choice == "rock") or \
    (player_choice == "scissors" and computer_choice == "paper"):
    print("You win!")
else:
    print("Computer wins!")
```