LEARN PYTHON

Rock, Paper, Scissors

In this project, we'll build Rock-Paper-Scissors!

The program should do the following:

1. Prompt the user to select either Rock, Paper, or Scissors.
2. Instruct the computer to randomly select either Rock, Paper, or Scissors.
3. Compare the user's choice and the computer's choice.
4. Determine a winner (the user or the computer).
5. Inform the user who the winner is.

Happy coding!

Rock, Paper, Scissors

1. Begin by writing a multi-line comment that describes what this program will do.

2. Since the computer will select Rock, Paper, or Scissors randomly, we will need the `randint` function – which is not built-in.

   Use a function import to import `randint` from the `random` module.

3. We've imported the code that we will use later. Let's move on!

   Create a list called `options` and store `"ROCK"`, `"PAPER"`, and `"SCISSORS"` as strings.

4. The user will either win or lose in the game, so the program will need to print win/lose messages to the user later.

   Create a dictionary called `message` with three key-value pairs:

   - key `"tie"` points to value `"Yawn it's a tie!"`
   - key `"won"` points to value `"Yay you won!"`
   - key `"lost"` points to value `"Aww you lost!"`

5. Let's write a function that decides who the winner is.

   Create a function called `decide_winner`. The function should take two parameters: `user_choice` and `computer_choice`.

6. Let's start building the `decide_winner` function.

   First, print the `user_choice`, the first parameter, using string formatting.

7. On the next line, print the `computer_choice`, the second parameter, use string formatting.

8. How do we determine the result?

   Start by adding an `if` statement that checks if the `user_choice` is equal to the `computer_choice`. This means it's a tie!

   Inside the `if` statement, print the message to the user informing them of the tie. The message is stored under the `"tie"` key in `message` dictionary.

9. Now it's time to think of the scenarios in which the user wins:

   - **User:** Rock, **Computer:** Scissors
   - **User:** Paper, **Computer:** Rock
   - **User:** Scissors, **Computer:** Paper

   Let's take care of the first bullet point!

   Add an `elif` statement that checks if the user selects Rock `and` the computer selects Scissors.

   Remember, we can compare `user_choice` and `computer_choice` with the items in `options` by their index.

   Inside the statement, print the winning message, which is the value stored under the `"won"` key in `message` dictionary.

10. Perfect! But that takes care of only one scenario where the user wins.

    Add two more `elif` statements that print the winning message when the user wins.

    You can use the scenarios from task 9 to help you.

11. Finally, we've taken care of all of the cases where the user could win. But what if none of these conditions are met?

    If it's not a tie, and the user did not win, that only means the user lost!

    Add an `else` block and print the losing message inside of it.

12. Great! We have the function that will decide who the winner is between the user and the computer, but we haven't written a function that actually starts the game. Let's do that now.

    Create a new function called `play_RPS()`.

13. Inside the function, we'll have to prompt the user for their selection.

    Prompt them with the message: `"Enter Rock, Paper, or Scissors: "`

    Store their input in a variable called `user_choice`.

14. Convert the user's choice to uppercase in case they type in lowercase `rock`, `paper`, or `scissors`.

15. The computer has to play too! Remember, the computer's choice has to be random, so we'll make use of `randint` to accomplish that.

    Remember, this is how the `randint` function works: `randint(low, high)`

    On the next line, create a variable called `computer_choice`.

    Set the variable equal to an item in `options` at a random index (0 to 2).

16. Great! The user has now submitted their choice and the computer has also made a random choice. It's time to determine a winner. Thankfully, we already wrote a function that can do that.

    On the next line, call the `decide_winner` function. Pass in `user_choice` as the first argument and `computer_choice` as the second argument.

17. Our program won't run unless we call the correct function! On the next line, call the `play_RPS()` function. Make sure it's outside of any other function.

18. You worked really hard to create this game. Now it's time to sit back and be amazed at how far you've come!

First, click Save. Then, in the terminal, type the following command and press Enter:

```
python RPS.py
```

You have built a Rock, Paper, Scissors program. Congrats!