***Rock , paper , scissors in python:-***

A command line game , a traditional rock , paper , scissors is a two-player game or could be more than that in which each play simultaneously choose , either rock , paper or scissors , give that **rock crashes scissors** , **scissor cuts paper** , and **paper covers rock ,**

Based on the rules the winner decided.

In this project game in python coding we will learn couple of things:-

* Code your own **rock paper scissors** game.
* Take in user input with **input ().**
* Play several games in a row using🡪 while **loop.**
* Clean up your codes in enum & functions.
* Define more complex code with a **🡪 Dictionary.**

Using the description above we are going to need to **import** the module to simulate the computer’s choices so as to randomly computer choice & play function

One game determining who wins & who loses

We can use user input by just using the python function so let’s ask the user below what’s your choice??

And we are going to lowercase just in case somebody puts In a capital R & we do not want the computer to register that as not a rock then we are going let the computer choose between three different objects and we can do than by using RANDOM**.choice** as we mentioned above R , P , S …. For the computer every single time now.

After that let’s use some code to figure out who

Import random who wins???

Below we will see a chart where will be given all the **possibilities** that both of the **USER & COMPUTER** could input plus the result of the combination

Def play ()

User = input (“what’s your choice? ‘r’ for rock , ‘p’ for paper , ‘s’

For scissor\n("

User = user.lower()

Computer = random. Choice([r , p , s])

|  |  |  |
| --- | --- | --- |
| User | Computer | Result |
| Rock | Rock | Tie |
| Paper | Paper | Lose |
| Rock | scissors | Win |
| Paper | Rock | Win |
| Paper | Paper | Tie |
| Paper | Scissors | Lose |
| Scissors | Rock | Lose |
| Scissors | Paper | Win |
| Scissors | Scissors | Tie |

Ex:

If user == computer

Return “ you and the computer have both chosen{}. It’s a tie”.format (computer)

In case of win: we will use the helper function is win and pass how is the user 1 against the computer then if that is true iam going to return

The user x and the computer chosen y the user will win

If is\_win(user , compter)

Return “ you have chosen {} and the computer have chosen {} . you won!.format(user , computer)

Why the user win ?? let’s define the function

We see in the table above all the different combinations so if the user is rock and the computer is scissors then the user wins.

So let’s define.

Def is\_win (player , opponent):

# return true is the player beats the opponent

And we are going to make if statements if the player or the user input R and the computer choose S or If the player chose s and the computer choose P and if the player chose p and the opponent chose R , so in all those cases we will return TRUE

# winning conditions: r > s , s > p , p > r

If (player == ‘r’ and the opponent == ‘s’) or (player == ‘s’ and the opponent == ‘p’) or ( player == ‘p’ and the opponent == ‘r’)

Return true

Return false 🡪

..Which means the player has not won against the computer.

In case of lose instead of creating helper function to determine the losses we can just assume the remaining cases are losses which you could tell from the table above.

If it is not a tie or a win then that means that the player lost and at the end we are going to say the player chose x and the computer chose y

You lost.

Based on that how can we code a function that allow us to play best of n games so fisrt we are going to set up a function we can call it play best of you know that you win the best of n games when you win at least half of them and here we want to round up using math dot to get the ceiling of whatever number when we divide and by two.

For example if n three you have to win two games and if n five you will have to win three and so on

Def plat\_best\_of(n):

# play the best against the computer until someone wins the best of n games

# to win, you must win ceil(n/2) games (ie 2/3 , 3/5 , 4/7)

Player\_wins = 0

Computer\_wins = 0

Wins\_necessary = maith.ceil(n/2)

While player\_wins < wins\_necessary and computer\_wins < wins\_necessary

In this function we are going to initialize two variables player wins and the computer wins and both of those are 0 at first then we are going to initialize necessary wins which just the number you get when you divide n/2 and take the ceiling of that.

Now we will talk about loop now because we do not n=know how many iterations we need for one of us to achieve the wins necessary in order to win best out of n games.

As we can say that while loop is perfect for that as we can say if either of us have one then stop this loop otherwise keep going plus we are going to change the return function in this case we do not have the compare strings and instead of returning a tie we are going to return zero for tie and 1 for player win and negative -1 for computer win

If user == computer

Return ( 0 , user , computer )

If is\_win( user , computer)

Return ( 1, user , computer )

Return (-1 , user , computer)

And then we can print those out and show it whoever’s playing the game

When we call the play function we can set the result user and computer to whoever’s return and the result will be 0 , 1 , -1

Result ( user , computer , player ) = play()

Or positive 1 user could be r , s , p & vice versa to the computer too

And the last if statements

If it’s a tie that means the result = 0 so then we can print it is a tie

# tie

If result == 0

Print(‘It is a tie. you and the computer both chosen {}.\n’.format(user))

Otherwise win statement

# you win

Elif result == 1

Player\_wins +=1

Print(‘ you chose {} and the computer chose {}. You win \n’.format(user , computer))

Else:

Computer\_wins +=1

Print(‘you chose {} and the computer chose {}. You lost\n’.format(user , computer))

Print(\’n’)

And the end once you have broken out of this while loop if the player is greater than the computer wins that means the reason why we broke out of this loop because you won and so on.