Rock, Paper, Scissors!

This was my second time working with python. I decided I wanted to make some sort of quick easy game and had the goal at the beginning of having a game that could be played in a loop as I had never tried using a loop in python before. I learned a good amount doing this project especially when it comes to the importance of organizing code, how loops work and just reenforcing the basics I had learned before.

![A screenshot of a computer program

Description automatically generated with medium confidence]()I used a tutorial found at (<https://realpython.com/python-rock-paper-scissors/>) to help me get started. The first phase of the project was just getting the basic structure working. Starting with the basics of the game. Getting the user input out of 3 options, rock paper or scissors. Generating a move by the computer by using an option at random. I had used variables like this before so this was fairly quick and easy. Then setting a table of all the possible moves and outcomes, printing the outcome of the game to show a winner. Below image is the first successful run of code where we had a working game.

This ran the basic game, would show you a winner, but didn’t allow you to play multiple times in a row without closing and reloading the script as well as it did not keep score which is something I wanted to figure out how to do. So for phase two I had two goals, make it loop and find a way to keep score. The tutorial I used to get going didn’t cover how to create a loop very well, but after doing some more searching on google I learned my first lesson that I will consider a valuable takeaway, indentions matter! I didn’t realize until 5 minutes of messing around with it that the whole code was breaking because I didn’t have things properly indented. After getting the loop working, I was able to add lines to keep score. After accomplishing both of the main goals I had for this step I realized if I typed in my option wrong such as misspelling my move for example, that the scoreboard wouldn’t update, and I realized I needed a better way to display that your move was invalid or not accepted by the game. So I added the input error variable to check when printing results. At this point I wondered about adding a system to check for common misspelling or different ways to type the move and have the game automatically accept close or similar answers by having a library maybe to compare against to count as a move, like for example ROCK, Rock, rockk, could be accepted as rock. But I was coding this on my lunch break in an hour of free time I had and decided I was happy with just having it loop back through so they could try again so I could finish this project and move on to the next one. The below image shows several round in ![A screenshot of a computer

Description automatically generated]()a row, keeping score over each round and how it will handle an error on the user input.

I was able to learn quite a bit even on such a short simple project. My biggest takeaways were the importance of the order of your code, proper indention, and also this was the most if/elif/else style of statements I have ever use so managing them properly and not messing up the syntax was good practice. Most of all though I was happy to implement a loop properly for the first time.