Warm-up: Rock-Paper-Scissors

1. Insert the words elif, else and if into the gaps in the code so that it runs:

```
import random

player = input("Please enter R, P or S (for [R]ock, [P]aper and
[S]cissors) ")
computer = random.choice('RPS')

____ player == 'R' and computer == 'P':
    print("Computer wins")
____ player == 'R' and computer == 'S':
    print("Player wins")
____:
    print("it's a draw")
```

2. Extend the program, so that it also works for other choices.

3. Fix one bug in each if-statement:

```
elif player.upper() not in 'RPS':
    print('Invalid input. Please enter R,P or S.')

elif player == computer
    print('You chose the same as I did')

if player = 'S':
    print('You chose "scissors".')

else:
print('You chose something else than "scissors"')
```

Exercises

- Write a script that checks if a string input is a palindrome (is the same read backwards).

- The following for loop searches for 33 in the data. Modify the code, so that it uses a while loop instead.

```
data = [5, 7, 33, 12, 4, 3, 18]

found = False
for n in data:
    if n == 33:
        found = True

print("The value 33 has been found: {}".format(found))
```

- The following while loop counts numbers higher than 10. Change the code so that it uses a for loop instead.

```
data = [4, 7, 11, 1, 3,  15]

i = 0
high = 0
while i < len(data):
    if data[i] > 10:
        high += 1
    i += 1

print(f"There are {high} values higher than 10")
```

- Create a script that takes a list of 10 integers as input and checks it for duplicates. Use conditions and loops for that.

- Create a script that asks the user to type an integer input, then, another, then another, etc... - when the user types "0", the script has to stop and display the number of odd values and the number of even values that were entered by the user since the last “0”.

- The user gives an integer n between 2 and 12, the program gives the number of ways to make n by throwing two dice
  ((3, 6) and (6, 3) count as two different ways)

- Now, do the same with 3 dice and an integer between 3 and 18

- Finally, create a function with 2 input arguments : (number_of_dice, integer) that returns a list of tuples with all the ways the dice can create the number.