DAT-119 – Python 1

**Homework 6**

As always, you need to write these programs *without searching online*. You can use the textbook, the Jupyter notebooks, conversations with your colleagues and me, and the approved resources; that’s all.

Also, as always, **be sure to follow the style guide**, including turning in a plan with your code.

1) Write a rock, paper, scissors program! (Use at least two functions, including main())

1. When the program begins, a random number between 0 and 2 is generated. If the number is 0, the computer has chosen rock; if the number is 1, the computer has chosen paper; if the number is 2, the computer has chosen scissors. Do not display the computer’s choice yet.
2. The user enters their choice of “rock,” “paper,” or “scissors” at the keyboard. Only “rock,” “paper,” or “scissors” (without commas or quotation marks) should be accepted.
3. The computer’s choice is displayed.
4. A winner is selected according to the rules:
   1. If one player chooses rock, and the other chooses scissors, rock wins (rock smashes scissors)
   2. If one player chooses scissors, and the other chooses paper, scissors win (scissors cut paper)
   3. If one player chooses paper, and the other chooses rock, paper wins (paper wraps around rock)
   4. If both players make the same choice, the game must be played again to determine the winner.

Remember to plan ahead (actually write out some of the logic in your plans … programs are going to start to get complicated enough now that trying to code without breaking things out into functions and thinking through any looping or conditional logic is a recipe for trouble). Remember to write out small pieces (make sure the water runs before you wash the dishes) -- test and save often!

**Example output:**

Welcome to the Rock, Paper, Scissors!

Please enter "rock", "paper", or "scissors", all-lowercase, no quotation marks: rock

The computer chose rock

You tied and must play again!

Please enter "rock", "paper", or "scissors", all-lowercase, no quotation marks: potato

That was not one of the options; please try again.

Please enter "rock", "paper", or "scissors", all-lowercase, no quotation marks: rock

The computer chose scissors

You win!

2) Write a random number guessing game! (Use at least two functions.)

Your program should generate a random number in the range from 1 through 100 and ask the user to guess what the number is. If they guess too high, the program should display “Too high, try again.” If they guess too low, the program should display “Too low, try again.” If they guess the correct number, the application should congratulate them, tell them how many guesses it took, and ask if they want to play again. If they do, the program should generate a new random number and let them play again. (I’m going to stop pointing this out and expect you to *do this automatically*, but I’ll say it explicitly for now: don’t accept invalid numbers as input.)

**Example output:**

I'm thinking of a number between 1 and 100. Try to guess it!

Please guess a number between 1 and 100: 105

That was not a valid guess.

Please guess a number between 1 and 100: 50

Your guess was low. Keep going!

Please guess a number between 1 and 100: 75

Your guess was high. Keep going!

Please guess a number between 1 and 100: 62

Your guess was high. Keep going!

Please guess a number between 1 and 100: 55

Your guess was high. Keep going!

Please guess a number between 1 and 100: 52

You won! It took you 5 guesses.

That was fun! Shall we play again? yes/no: no