**Assignment 3**

Solve these JavaScript looping problems.

**The “Coin Flip” Game Redux**

In this part of the assignment you will explore what you’ve learned so far about variables, for loops, conditionals, operators, math, and more to create a different version of the “coin flip game”. To complete the game follow the steps outlined below:

1. Begin your application by declaring a variable called *coinFlip*. Do not assign the variable a number just yet.
2. Create a for loop that loops a certain amount of times. The amount of times that the for loop executes will be collected from the user via a prompt.
3. Within the for loop assign a randomly generated number to *coinFlip.* You will have to use the same Math formula that you used in the previous assignment to get this number.
4. Use a conditional statement to check the result of the coin flip. If it’s 0, write out “Heads” into the console window. If it’s 1, write out “Tails” into the console window.

Note: Unlike the previous assignment, you’re not prompting the user for their choice. You’re simply displaying the coin flip results in the console.

**The “Coin Flip Streak” Game**

In this part of the assignment you will explore what you’ve learned so far about variables, do while loops, conditionals, operators, math, and more to create a “coin flip streak game”. The point of this simple game is to see what kind of streak you can get to (how many times “Heads” comes up in a row before “Tails” does to end the game). To complete the game follow the steps outlined below:

1. Begin your application by declaring a variable called *coinFlip*. Do not assign the variable a number just yet.
2. Create a do while ldooop.
3. Within the do while loop assign a randomly generated number to *coinFlip.* You will have to use the same Math formula that you used in the previous assignment to get this number.
4. Use a conditional statement to check the result of the coin flip. If it’s 0, write out “Heads” into the console window. If it’s 1, write out “Tails” into the console window.
5. Set the condition of the do while loop to end once the coinFlip becomes “Tails”.

**Looping a Triangle**

Write a loop that displays the following triangle within a console window:

#  
##  
###  
####  
#####  
######  
#######do

**Odd or Even?**

Write a loop that will iterate from 0 to 15. For each iteration it will check if the current number is odd or even, and display a message in the console window.

Sample Output:  
"0 is even"   
"1 is odd"   
"2 is even"

**Marco! Polo!**

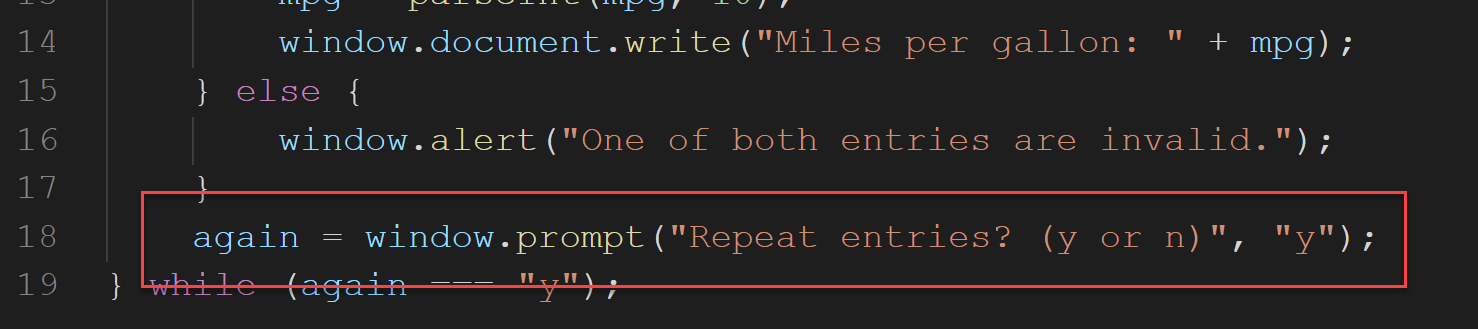
Create an application which iterates numbers from 1 to 100. For multiples of 3, print "Marco!" instead of the number and for multiples of 5, print "Polo!" instead of the number. For numbers which are multiples of both 3 and 5 print "Marco! Polo!". This is a VERY common JavaScript interview question and you should know how to do this. You will need to take advantage of the modulus operator to accomplish this task.

**Countdown**

Write an application that counts down to 0 based on the number that a user passes into a prompt. If the number is 10, it should count down from 10 to 0. If the number is 100, it should count down from 100 to 0 and so on.

**Validate Repeat Entry in Lab 5 (Extra Credit 5 points)**

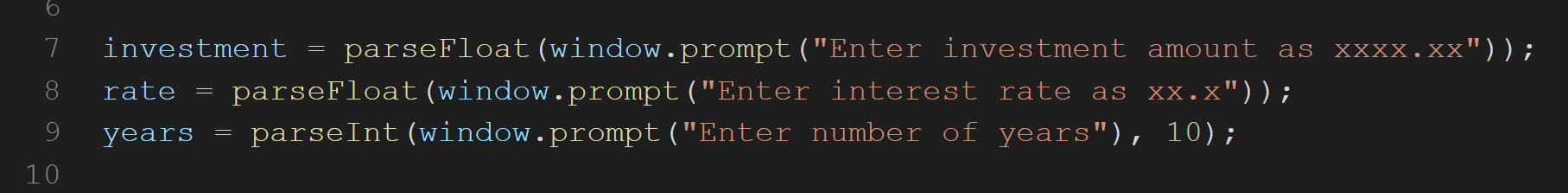
There’s a line of code in lab 5 that asks the user if they want to repeat their entry. The code resembled the image outlined below.



Your job in this part of the assignment is to validate that entry. If it’s anything other than “y” or “n”, the prompt should simply reappear until the user enters a valid entry.

**Validate Numeric Entries in Lab 6 (Extra Credit 5 points)**

There are 3 lines of code in lab 6 where a user must enter their investment amount, the rate, and the number of years they would like to invest for. The code resembled the image outlined below.



Your job in this part of the assignment is to validate those entries. Within investment, the number should be numeric. Within rate, the number should be numeric and between 0 and a realistic rate like 6%. For years, the number should be numeric and between 1 and 30.