Coin Toss 1

Programming is learning about different tools, and then creatively putting those tools together to cohesively make a whole program that does what you want to do. So far the tools you've learned are:

1. How to add javaScript to your html code
2. How to write a function in js
3. How to call a function in your html using onClick() and buttons
4. How to write things out using an alert box
5. How to click on an image in your html code to call a function in your javaScript
6. How to add a parameter to your function and how to call a function with a value for the parameter
7. How to create and use a variable
8. How to use prompt box to put a value into a variable
9. What an if condition is, what an else-if condition is, and what an else condition is
10. How to modify existing html using document.getElementById()
    a. Using document.getElementById to change the src and the alt of an existing element
    b. Sending the id of an html element into the function to be used by document.getElementById()
11. Combining variables with strings using the + sign
12. How to generate random numbers
13. How to read information from existing html using document.getElementById()
14. How to use document.getElementById() to modify the style of an existing element
15. What the innerHTML is and how to read and modify that using document.getElementById()
16. When innerHTML can be complex, and how to update complex innerHTMLs
17. How to join to the end of a variable or string using +
18. The difference between using + with numbers and + with strings of characters
    a. Using parseInt to change a string of characters to a number
19. Modifying the length of a list using if and prompts
20. An intro to a while loop

You've learned a lot! While there are many more tools available for progamming (we've only been learning for ½ a semester), there are only 2 more tools needed for the final project.

Contents

Global Variables:

Coin Toss

Step 1: In your html file, create an image. Give the image an id. You can make the image be of anything at all – I made mine be a closed curtain (like in a magic trick, with something hiding behind it) but it really doesn't matter what it is.

Step 2: In your html file, create a paragraph. Give the paragraph an id. Again, I don't really care what it says now – a function will change its content shortly

Step 3:. Create a button. Make the value "play game" and make the onClick call a function PlayGame().

Now in your .js:

Step 4: Create a function PlayGame() Note that this function doesn't need a parameter.

Step 5: Inside the function, generate a random number between 0 and 2 (not including 2) and place the random number in a variable as follows:

```
x = Math.floor(Math.random() * 2)
```

Step 6: Add to the function an if condition such that if x holds 1 (the random number generated was 1), change the image in step 1's src to be that of a coin head and create a variable y that holds the word, 'heads' (fyi, the id of my image is 'cid'):

```
if (x === 0) {
        y = 'heads'
        document.getElementById('cid').src = "coinhead.jpg"
}
```

Step 7: create an else condition that sets y to tails, and changes the .src of the image to be that of a cointail

Step 8: Below the else condition (at the bottom of the function) use document.getElementById and the y variable to change the paragraph created in step 2 to "computer generated " + y

Step 9: Save and test. When you click on a button, either a coin head should show up and the paragraph "computer generated heads" should show up, or a coin tail should show up and the paragraph should say, "computer generated tails"

Let's make it a game:

Step 10: Back to the html: add another image next to the first image. Give it a different id. I put my two images in a table with 1 row and 2 data cells so they'd appear next to each other, but you don't have to.

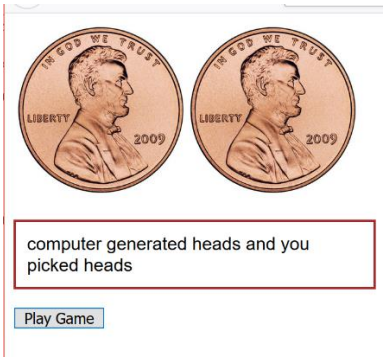
Step 11: And back to the function PlayGame() – add a prompt at the very top of the function that asks the user, "pick heads or tails"). Save the response in a variable (a new one, not the x or y you've already used).

Step 12: Now add a new if condtion either above or below the one you created in steps 6 and 7. The if condition should work as follows: if the new variable from 11 holds 'heads', change the src of the image you created in step 10 to the coin head, else change the src to the coin tail.

Step 13: Add to the document.getElementById for the paragraph (that currently writes out, 'computer picked '+y) what the user picked as well. My line looks like this (my paragraph's id was 'p1' and my user variable was z):

```
document.getElementById('p1').innerHTML = "comp generated " + y + " and you picked " + z
```

Step 14: Save and test. Now 2 images should show up – one representing the computer's random number of 0 or 1, with 0 being a coin head and 1 being a coin tail, and one representing what the user picked. In addition, the paragraph should print out what both picked. Mine looks like this (with a bit of style added to the paragraph to give it a border, and both the user picking heads and the computer generating the random number 0):



Step 15: let's indicate whether you won or not: In the function, add another if condition. This one should be below all the if conditions you've created so far. It is checking to see if what the user picked is what the computer generated. So if what the computer generated was heads, the y variable will hold heads, and if the user chose heads, the variable (my z variable) will hold heads, so they'll both be the same (y === z). If the computer generated tails, y will hold tails and if the user chose tails, the variable (my z again) will hold tails, so, again, y will be equal to z. So we can add a condition below all the ifs and around the document.getElementById() as follows:

```
if (y === z) {
        document.getElementById('p1').innerHTML = "You won!  Computer generated " + y + " and you picked " + z
}
else {
        document.getElementById('p1').innerHTML = "You lost.  Computer generated " + y + " and you picked " + z
}
```

What this does is check to see whether what the user picked and what the user generated were the same. If they're the same, it prints to the paragraph, "You won! …"

Otherwise it prints out, "You lost. …"

Almost done this part!

Global Variables:

Have you noticed that we keep using the same variable and parameter names in each function? We keep using x, y, z, and par again and again in every function we write. That's because variables within a function only belong to that function. The best way I have of describing this is that when I say, "professor" of CISC103, and "professor" of ENG110, the professor is unique to the class. The variables x, y, z, etc. are unique to each function, and thus the value inside of x in one function doesn't carry to another function. Equally, every time you click on a paragraph or button that calls that function, the variables are made fresh, as if they didn't exist before. To show you what I mean, try the following:

Step 16: Add another paragraph to your html code below the 'p1' paragraph. I gave this paragraph an id of 'wins' and also started out the paragraph's text with "Win Count: " (you'll change this in the .js playGame() function).

Step 17: in the function playGame(), at the very top of the function, add a variable wincount (yep, I named this variable multiple letters. I want to be clear what this variable holds, so I actually gave it a name that reflects what its purpose is inside the function. Initialize the wincount variable to be 0. The top of my function looks like this:

```
function PlayGame(par) {
        wincount = 0
        x = Math.floor(Math.random() * 2)
        z = prompt("Guess heads or tails")
        …
```

Step 18: In the if condition that checks whether you've won or lost (from step 15, above), if the user won (or if (y===z)) increase the count of the wincount variable as follows:

```
wincount = wincount + 1
```

Step 19: below the if condition (the very last line of the function, right above the function's closing }), change the paragraph with the id 'wins' that you created in step 16 using

```
document.getElementById('wins').innerHTML = "Win count: " + wincount
```

Step 20: Save and test your code by clicking again and again on the button in your html code. While there should be a paragraph that says "Win count: 0" or "Win count: 1" in the browser, this shouldn't be working the way you want it to.

What you'd really like is for the win count to keep going up every time you win. It isn't because every single time you click on the button on your web site, the function PlayGame() is called and the very first thing that happens is that wincount is created FOR THIS FUNCTION ONLY and set to be 0. Every. Single. Time. So if you win, the newly created wincount, which was set back to 0, is then increased to 1, and you get "Win count: 1" . If you didn't win, wincount isn't increased, which means you'll get, "Win count: 0", even if you've won 20 times before!

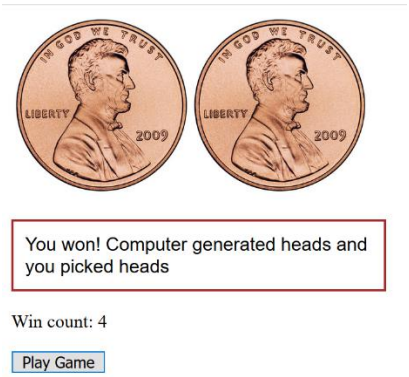
The problem is, wincount belongs to one calling of the function PlayGame. If that doesn't make sense, realize that every time you call the function, wincount gets set to 0 again. That's not what we want.

We want wincount to continue existing even when the PlayGame is done running each time. Equally, we want to originally set wincount to 0, but then every time we call PlayGame, we don't want it to be reset to 0. In order to do this, we can make the variable OUTSIDE of any function, at the top of the .js file. So it would look like this:

```
wincount = 0
function PlayGame(par) {
        x = Math.floor(Math.random() * 2)
        z = prompt("Guess heads or tails")
        …
```

Step 21: Place the wincount variable outside and at the top of your javascript file, and set it to 0 (as demonstrated above).

Step 22: Save the code, and test it by clicking on the button on your web page multiple times. After playing 7 times and losing 3 times and winning 4 times, my results now look like this:



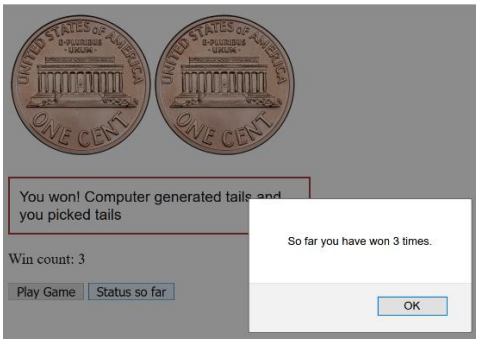
To show that you can use the variable wincount in other functions, let's create another function as follows:

Step 23: in your html code, add a button below the Play Game button that has as a value, "Status so far" and using onClick, calls a function, Status('a').

Step 24: In your .js file, create a second function below the PlayGame function called Status(par). In the function, use an alert box to print out the current number of wins as follows:

```
alert("So far you have won " + wincount + " times.")
```

Step 25: Save your code and test it by clicking on the Play Game multiple times. After you've played a couple of times, click on the Status so far button. An alert box should pop up showing you how many times you've won so far. Mine looks like this:



This shows that we can use global variables, or variables created above all the functions, in every function below it. Here we've used wincount in 2 separate functions, and it didn't lose its value between functions.

Congrats! You just wrote your first game

Your Turn:CoinToss(8 pts)

1. Make sure you have the above html code and .js functions running and you understand it!

Extra Credit (8 pts):War

There's a card game called War. In war, you get a random card, and the other player (in this case, the computer) gets a random card. Note that in this case both cards are random. If your card is higher, you win. If the computer's card is higher, the computer wins. If it's the same, then there's a tie. In this game, an ace beats a king, which beats a queen, which beats a jack, which beats a 10… Equally, you can assume there's only one suit for the purpose of images (otherwise you'd need to download 52 different images). Write the card game War, which is similar to the coin toss game above, and display both your card and the computer's card. Keep track of the number of times you win.

Extra Credit 2 (20 pts):BlackJack

Write the card game Black Jack. This extra credit might be challenging and requires a solid understanding of previous tools we've learned. In this game, you play against the dealer (the computer, in this case). Your goal is to get closer to 21 without going over. If the dealer goes over 21, you automatically win. If you go over 21, you automatically lose. Otherwise, if the dealer goes over 21, you win. Otherwise, whoever comes closer to 21 wins. I'm okay with there being ties, although in real blackjack if it's a tie the dealer wins.

For this game you'll have to make a table of 5 possible cards for yourself and 5 possible cards for the dealer. You'll have to make a variable to keep track of your score, and a separate variable to keep track of the dealer's score. You could use an if condition to ask 3 times if the user wants more cards, after the user gets their initial 2 cards, or you could use a while loop for this. You'll have to make some sort of rule for the dealer that if the dealer's total is under 17, they will get another random card. Again, if it helps, you can assume only one suit. You'll also need a global variable to keep track of how many times you've won.