JavaScript 2a: Random Numbers

So that your html and javaScript files don't get overwhelmingly confusing, please start a new html and javascript file for these second set of js tutorials.

Contents

Random numbers

Random Numbers Explained

Every computer game in the world uses random numbers. Things have to happen randomly or it's not much of a game. Random numbers are used any time you want something to happen unexpectedly – for instance, you may want random pictures to show up in a display, or you may want random words of encouragement to pop up in an app. Another example is in generated synthetic voices. If you just use a pure sin wave (a smooth sound wave), the resulting sound has that sort of tinny, synthetic sound to it. To make something sound more like a real voice, you need to add some breathiness. To add breathiness to a smooth sound wave, you just generate small random numbers and add them to the wave.

These are just a few of many uses for random numbers.

Previously you learned about variables, and about 2 different ways to put values into variables. You can do:

```
x = 3
```

and that puts 3 into the variable x. Once 3 is in x, every time you use x, it is the same as if you were using the number 3.

You can also do:

```
y=prompt("What is your age?")
```

and when the user types in their age, it goes into the variable y. Again, once the age is in y, every time you use y, it is as if you are using the age that was typed in.

You can also put random numbers into variables. Every language lets you do this in a slightly different way. JavaScript lets you do this using:

x = Math.floor(Math.random() * *10*)

This generates a random number between 0 and *10 (not including 10)* and puts that random number into the variable x.

The number 10 in the above example is what specifies that the largest random number to be generated will be 9. You can always change this number. For instance, if you have the following random number generator:

x = Math.floor(Math.random() * *3*)

The possible random numbers that x might hold would be 0, 1, or 2 (not 3 - the smallest number is 0, and the largest is always one less than the number after the * (which is multiplication in javaScript).

Now, what if you want a random number between 5 and 15? You'd do the following:

x = Math.floor(Math.random() * *10*) + 5

The first part generates a random number between 0 and 9, and then, whatever that random number is, 5 is then added to it. So, for instance, if the random number generated is 0, then after 5 is added, x will hold 5. If the random number generated is 4, then, after 5 is added, x will hold 9, and if the random number generated is 9, then x will hold 14.

Thus the smallest number the random number can generate, which is 0, plus 5, will make the smallest random number 5, and the largest number the random number can generate, which is 9, plus 5,will make 14, so the range is from 5 to 14, which is what we wanted.

You can just memorize how to use this, or you can read the following explanation:

x = Math.floor(Math.random() * *9*)

1. Math.random() generates a random number between 0 and 1 (e.g., 0.4).
2. This number is multiplied by the range (in this case 9):
    a. Math.random() * 9 gives us 3.6
3. Math.floor(3.6) rounds down to the nearest integer, or 3
    a. It chops off everything after the decimal point
4. Finally, you need a variable to put all this in.
    a. x = 3

Let's try random numbers*:*

Step 1: In your html file create a button. Make the button call a function when it is clicked on. Give the button a value. You can make up the name of the function and the value displayed in the function.

Step 2: In javascript file, create a function with the exact same name as the function that is called when you click on the button from step 1.

Step 3:. In the function, generate a random number between 0 and 6 and place it in a variable

Step 4: Use an alert box to show the random number generated.

Now save both files and test.

If it isn't working:

Make sure you've:

- Linked the javascript file to the html file using <script src = …></script> (like in the first tutorial)
- Have the function name in the html file and in the js file EXACTLY the same
- Have both an opening and closing { } at the beginning and ending of the function
- Make sure you have a small f in floor and a small r in random, but a large M for Math.
- Make sure that you have the parentheses exactly right – two opening and two closing parentheses in the Math.floor(Math.random()*7)

Once you have this working, let's expand a bit:

Weather Prediction

First, let's use the random number to bring up a random alert, in this case for the weather.

(Let's be honest – at some point it's crossed everyone's mind that the weather report very well might just be randomly generated)

The above function currently has a variable that will hold a random number, and that random number can be either 0, 1, 2, 3, 4, or 5 (not 6).

Step 5: Now in your .js: after the random number has been generated and placed into a variable, add an if condition.

Step 6: In the if condition, check to see if the random number variable is equal to 0, and if so, use an alert to show a type of weather ("sunny", "thunderstorms","tornados","windy", etc.). (see video)

Step 7: add an else if for if the random number is 1, and then alert with a different weather condition

Step 8: add an else if for if the random number is 2, and then alert with yet another weather condition

Step 9: continue to add else if conditions for random numbers 3 through 5, each with an alert of a different weather condition

Step 10: save and make sure it's working.

If it isn't working 2:

- Double-check all of If it isn't working 1
- Make sure that after each if and else if you've got an opening and closing ( )
- Make sure that when checking to see if your variable is a number, you use === (not =)
- Maker sure that each if and else if has an opening and closing { }
- Make sure the variable that you're using in the if, i.e., if (x === 0) {…. } is exactly the same as the variable that holds the random number, i.e., x = Math.floor(Math.random() *3)

Your Turn (20 pts):

Problem 1: (5 pts) Get the weather prediction function working

Problem 2: (7 pts)Magic 8 ball

You've all played with the magic 8 ball, in which you ask the ball a question and then shake it and an answer to your question pops up? You might ask, "Will I win the lottery?" and then shake the ball and the answer appears that might say something like, "It is highly unlikely!"? You're going to write this. Note that the typical magic 8 ball has 8 possible answers, and they are:

- As I see it, yes.
- Ask again later.
- Better not tell you now.
- Cannot predict now.
- Concentrate and ask again.
- Don't count on it.
- It is certain.
- It is decidedly so.

But feel free to be creative and come up with 8 of your own potential answers.

Create a magicEight function in your javascript file.

Inside the function, use a prompt box to ask the user to enter a question. Place a variable on the right to hold the answer. NOTE: you will not use the variable that holds the answer again. THERE IS NO CORRELATION BETWEEN THE QUESTION AND THE ANSWER THAT POPS UP!

Now below the prompt box, generate a random number between 0 and 8 (again, note that the possible random numbers generated will be 0,1,2,3,4,5,6, and 7, but not 8) and make sure you place the random number in a variable, to the left of the random number generator.

Now, like you did with the weather predictor, use the random number generated to bring up one of the 8 possible answers. So you might have something like,

```
if (x === 0) {
        alert("As I see it, yes.")
}
else if (x === 1) {…
```

Add a button to your html code that calls this magicEight function when you click on it

Test and make sure the function is working.

Problem 3: (8 pts) Lucky Guess

This problem is similar to the problem from the last lab, where you had one button that was the right button, and all the other buttons were wrong. But now the "right" button is going to be determined randomly.

In your html code, add a table with 6 buttons (more if you like). Give each button a value of 0,1,2,3,4, and then 5, respectively. Make each button call the same function when clicked on. However, when the button is clicked on and the function is called, make sure each button calls the function with its number as input to the parameter. So you might have something like this:

<input type = "button" value = "0" onClick = "Guessfunc(0)">

Please don't copy, or if you do, make sure you retype the quotes. Quotes in Microsoft word don't work in javaScript because they're funky.

Do this for each button.

Now in your javascript file, write a function with the same name as the oen being called by the above buttons. Make sure it has a parameter to hold the number being passed in.

Inside the function, generated a random number between 0 and 6, and make sure you have a variable to hold that random number on the left side of the random number generator.

Now use an alert box to show the random number

Now use an if condition to check to see if the random number is the same as the parameter. If it is, use an alert box to say, "You guessed correctly!!"

Otherwise, have an alert box that says, "Sorry, wrong button".

Save your files. Test and make sure it works.