MP1: Random Numbers

Due: Sept. 16, 2014 by 6:00pm

Overview

In MP1, you write your first JavaScript program on your own! This program will be very close to what you have seen in lecture and in lab, so it may be helpful to review the lectures slides if you get stuck.

Before getting started, you will need to setup a plain-text editor and a JavaScript interpreter in order to complete this MP. You can find information on how to do this on both PCs and Macs on the CS 105 website.

To get started on the MP, you will need to download the **`mp1.zip`** file that is provided for you on the CS 105 website. In order to work on the **`mp1.js`** found within the zip file, you must **extract** the zip file to make it a folder that is separate from the zip file itself. You will want to copy your **`mp1.js`** file into the folder that contains your JavaScript interpreter in order to make it easier to run.

Requirements

In MP1, you program must generate a random number and then ask the user to guess it. To generate the random number, you should use the code already provided in **`mp1.js`**, which is:

```
var randomNumber = Math.floor((Math.random() * 100) + 1);
```

After this line of code has been executed, the variable **`randomNumber`** will contain a number that is between 1 and 100 (incisively). That means the number may be 1, 27, 94, and even 100. The random number will never be less than 1 or greater than 100.

After generating the random number, your program must read input form the user in the form a guess at the number. When the user guesses a number, you will respond with either "Lower" or "Higher" based on if the random number is smaller/lower or larger/higher than the number they guessed until the user guesses the right number. When the user guesses the right number, you should print out "Correct!" and allow the program to exit.

Since you will be responding to the user differently each time, this requires the use of **conditionals**. Additionally, since you will be repeating the process multiple times, this requires the use of a **loop**.

Example Output

As an example, suppose your program randomly generates the number 37. The following is the result of running a solution to MP1:

```
I have thought of a random number in the range of 1 to 100.  Guess!
10
Higher
80
Lower
50
Lower
20
Higher
40
Lower
35
Higher
38
Lower
37
Correct!
```

A Few Tips

Your program will contain a **loop** in order to complete this MP. If your logic is not correct, it is possible to get stuck inside of the loop and your program might never exit. In order to force your program to exit:

- Hold down the **Control** or **Ctrl** key on your keyboard,
- Press the ‘C’ key
- …this Ctrl+C combination will force-exit your program.

It might be useful to try and test your program at different steps. It isn’t necessary to finish the entire program before testing it. Maybe just read input from the user once time and check if it’s higher or lower? Only once that works you can try adding the loop.

Submission and Grading

Submit your MP using the MP1 submission form on the CS 105 website. Due to the short nature of this MP, this MP will not have any partial credit. If your program works, it will receive full credit; if your program fails to work for any reason, it will receive no credit.