CSCE 120: Learning To Code

Manipulating Data I
Hacktivity 3.2

Introduction

Prior to engaging in this hacktivity, you should have completed all of the pre-class activities as outlined in this module. At the start of class, you will be randomly assigned a partner to work with on the entirety of this hacktivity as a *peer programming* activity. Your instructor will inform you of your partner for this class.

One of you will be the driver and the other will take on the navigator role. Recall that a driver is in charge of the keyboard and computer while the navigator is in charge of the handout and directing the activity. However, you are *both* responsible for contributing and discussing solutions. If you were a driver/navigator in the prior activity, switch roles for this activity.

1 Oddities

Computers are finite machines. Sometimes the way that numbers are stored in memory can lead to odd or unexpected results. We'll explore these a few of these issues in the following exercises.

Create a new JavaScript source file in LightTable

1.1 Undefined Values

1. Create a variable and set it to a positive value.
2. Create another variable and set it equal to the first variable multiplied by another variable that you have not previously used

3. Print out the value of the variable to the console. What is it?
4. Discuss with your partner why you get each result that you do.

1.2 Division By Zero

1. Create a variable and set it to a non-zero value.
2. Create a variable and set it equal to zero.
3. Attempt to divide the first by the second and place the value into another variable and print it out.
4. Discuss with your partner why you get each result that you do.

1.3 Invalid Numbers

1. Call the square root function in the math library with a variable whose value is -1 and print the result. What is it?
2. Call the logarithm function in the math library with a variable whose value is zero and print the result. What is it?
3. Call the sine function in the math library with a variable whose value is 90 and print the result. What is it?
4. Discuss with your partner why you get each result that you do.

1.4 Limits of Integers

1. Create a variable and assign it a value of 9007199254740992.
2. Create another variable and set it equal to the first plus one.
3. Create another variable and set it equal to the first plus two.
4. Print out all your variables and determine if it worked. Discuss within your group what could have gone wrong.

Advanced Tip: If you want to explore this issue further, read the following blog post: http://blog.vjeux.com/2010/javascript/javascript-max_int-number-limits.html. Several libraries exist for dealing with large numbers in JavaScript, notably: http://www-cs-students.stanford.edu/~tjw/jsbn/.

1.5 Limits of Floating Point Numbers

1. Write the following code snippet:

```
var y = 0.1 * 0.2;

console.log(x);

x = y * 10000;
console.log(x);

x = y * 100000;
console.log(x);
```

2. Before you execute the code, predict what the values will be.
3. Execute this snippet and observe the results. Do they match your predictions?
4. Discuss with your partner possible reasons for why you got the results you did.

2 Program: Computing Air Distance

Create a new JavaScript file and write a program that prompts the user for the latitude and longitude of two locations (an origin and a destination) on the globe (so four values). These numbers are in the range $[-180, 180]$ (negative values correspond to the western and southern hemispheres). Your program should then compute the air distance between the two points using the Spherical Law of Cosines. In particular, the distance d is

$$d = \arccos\left(\sin(\varphi_1)\sin(\varphi_2) + \cos(\varphi_1)\cos(\varphi_2)\cos(\Delta)\right) \cdot R$$

- φ_1 is the latitude of location A, φ_2 is the latitude of location B
- Δ is the difference between location B's longitude and location A's longitude
- R is the (average) radius of the earth, 6,371 kilometers

Note: the formula above assumes that latitude and longitude are measured in radians r, $-\pi \leq r \leq \pi$. Recall that you can convert degrees to radians using the following:

$$\frac{d \cdot \pi}{180}$$

Your program should then output the air distance between the two locations.

Test your program by computing the distance between each pair of the following locations:

- Memorial Stadium: 40.8206N, 96.7056W
- Wrigley Field: 41.9483N, 87.6556W
- Tokyo Japan: 35.6833N, 139.6833E

3 Finish an App: Currency Conversion

We have provided a basic currency exchange app that makes real-time queries to a currency exchange webservice (which provides the most recent market currency exchange rates). However, it is incomplete. It will be your task to examine the code and determine where the problem is and fix it.

1. Download (or "clone") the app from GitHub: https://github.com/cbourke/CurrencyApp
2. Try out the application using a few different currencies: Japanese Yen (JPY), British Pound (GBP), and Euro (EUR). You'll notice that the result is always zero.
3. Open and examine the HTML and JavaScript files.
4. Together, explore the code and trace how clicking the button works. It is not necessary that you completely understand all the details yet.
5. Find the point in the code in which you should use the input value (in dollars) and the exchange rate (provided by the webservice) to compute the result.
6. Add the necessary code to correctly display the result and rerun the program.
7. How much is $100 in Yen, Pounds, and Euros?
8. Try some other currencies