Kelly Masuda

March 10, 2015

Unit 9 Report – CoffeeScript

I tried to do CoffeeScript, but I ended up hating the little language more than I wanted to. CoffeeScript is supposed to be a better version of Javascript such that it encourages readability, good programming practices, and easy to memorize syntaxes. I initially liked it until I tried to make a web page out of it using an older project.

Syntactically speaking, it seems that CoffeeScript makes it easier to use by removing all brackets, all semicolons, and is organized by indexes. This part I liked because trying to read someone else's JavaScript code was a bit daunting. CoffeeScript also made it easier to read by getting rid of unneeded keywords like return, function, and var.

Defining a function in CoffeeScript is easier done than said. All you have to do is type the name of the function followed by an '=' sign. The parameters will go between the '()' symbols then an '->' will define the actual parts of your method. If a variable is left at the bottom by itself then that means to return that variable, like it is done in Ruby.

String interpolation is done the same way it is done in Ruby, with '#{var}'. Since I decided to not integrate what I've learned here into an actual web project, I learned that 'console.log' outputs whatever you want to see to the console.

As far as arrays are concerned, there are several ways to initialize one. You can initialize an array using a for loop with a range (or another array) all on a single line. You can initialize them like you'd address them in Ruby.

Iterating through arrays is the most interesting thing I've discovered during my CoffeeScript field trip. Assuming I have a function:

```
        printWarriors = (id, num) -> console.log "#{id} #{warrior}"
```

Let's also assume I have two arrays of n items each, one that has a list of names, the other simply being an array of of numbers from 1 to n. I can print to console the id number and the warrior's name from two different array lists, all on a single line:

```
        printWarriors idnum[i], warrior for warrior, i in warriors when
warriors[i] isn't "Velaseriat"
```

So I lied. Let us add an extra element in our array of names with the entry being "Velaseriat". What the above line does is given a multi-parameter function, we will need multiple arrays (it actually doesn't have to be the same length), and we will crank out the contents of that function until the longest array exhausts itself. Since printWarriors has two parameters, we have to stuff it with two arrays.

So if I got anything useful from CoffeeScript, it is the iteration of multiple arrays to get a result out of a function that accepts multiple parameters. This alone may have justified switching from JavaScript to CoffeeScript.