

Taylor Schmidt
CSCI 446 Unit09
CoffeeScript

CoffeeScript Description

In essence, CoffeeScript is a streamlined version of JavaScript. It can accomplish all of the same tasks as JavaScript, just in a more concise and “prettier” manner. Instead of being interpreted directly at runtime like JavaScript, CoffeeScript is actually compiled into .js files and is then interpreted as straight JavaScript, so there is no need to do anything special on the part of the developer when using CoffeeScript over JavaScript.

Since CoffeeScript is basically a streamlined version of JavaScript, a developer can theoretically accomplish the same tasks in a shorter amount of time using CoffeeScript. The syntax is simple and easy to understand, doing away with things like semicolons and curly braces, making CoffeeScript more like the hip young languages of Python and Ruby over “oldschool” languages like Java, which your grandpa probably still uses. An even bigger plus is that the JavaScript which the CoffeeScript compiler creates is just as high performing and often even higher performing than the JavaScript a human would write. Definitely a win-win.

Program Description

Since it took me a very long time to get CoffeeScript working for me due to trivial first timer errors and such, I created a very simple demo program which just takes a name and password and checks whether they are correct. My page illustrates CoffeeScript’s simplistic style and ease of use (once you know the correct way to do things). Instead of a complex function with lots of syntax, CoffeeScript allowed me to basically write a function almost in English sentences, especially for if-else loops and such. To compile the .coffee file into a .js file, I needed to run the command:

coffe -b -o [Destination DIR] -c [Source DIR]

To run my demo, simply open the .html file within the unit09 directory. The correct credentials are name → “Taylor” and password → “iamcool”

Time Card:

Hours 0-1 → Researching/installing CoffeeScript

Hours 1-3 → Learning CoffeeScript

Hours 3-4.5 → Writing demo and figuring out trivial first time errors

Hours 4.5-5 → Writing this report