JavaScript Basics

JavaScript is a cross-platform, object-oriented scripting language. JavaScript is extremely popular for a variety of reasons. It is a small and lightweight language allowing maximum flexibility for developers to take it in a bunch of different directions. JavaScript lives inside a host environment (a web browser or Node server), it can be connected to the objects of these environments to provide programmatic control over them.

Variable Declaration: JavaScript variables are containers for storing data values – imagine a cup you fill with coffee, the cup holds the coffee, a variable holds a value. All JavaScript variables must be identified with unique names. These unique names are called identifiers. Var x;

Variable Assignment: Assignment operators assign values to JavaScript variables – our cup can now have coffee poured in it, giving our variable a value to hole. The = assignment operator assignes a value to a variable. Var x=10;

Data Types: Data types are an important concept; to be able to operate on variables you need to know the data type. There are six data types that are JavaScript primites: Boolean – true or false; null – aka nothing; Number – 42 or 3.14159; String – “Coding Dojo Rocks!; Array – [1, ‘Coding’, 2, ‘Dojo’]; and Object – {first\_name: ‘Jane’, last\_name: ‘Doe’}

Conditionals: When you write code, you want to perform different actions for different decisions – hitting different code blocks based on values or conditions that have been met. You can use conditional statements in your code to accomplish this. There are the following condtional statements: if a specified condition is true, do this code in our code block; else if ito specify a new condition to test, if the first condition is false; else we execute this block of code;

Loops: There are many different kinds of loops in every programming language, but they all essentially do the same thing: they will repeat an action some number of times. Imagine you have to run a mile, well you run around the track four times and then youstop. That’s a loop!

Function: Functions are an encapsulation of a code block. When we call our function this will run that code block. Think of it as a list of instructions. As an example imagine we are putting together a desk from Ikea, we open up the instruction manual and get started, first we screw the legs to the table top; next we place the table the right way up. Done! We finished our instructions. Sadly there are a ton more tables to do so lets call our function over and over and over again.