girldevelopit

Girl Develop It:
Intro to JavaScript

with:
Sara Chipps
by:
Sara Chipps and Alexis Goldstein

Introductions

- Before we begin, we'd like to learn a little bit more about everyone here!
- Can you please introduce yourself, and tell us why you're here and what you hope to learn.

Goals for Today

- Learn what JavaScript is and what it can do
- Learn what functions and variables are
- Work with JavaScript and build our first interface

Why use JavaScript?

- To add interactivity to web pages.

It can:
• Put dynamic text into HTML pages.
• React to events.
• Validate form data.
• Create cookies (but not the yummy kind).

What is JavaScript

- A scripting language
- Adds interactivity to a web page
- Allows access to HTML objects
- Animates, adds, updates, slides, collapses, creates plugins, and more!

Jargon Alert! Scripting language

- A scripting language is a light-weight programming language.

- They may use another program (in JavaScript's case, the browser) to do the work and simply tell it what to do.

- They often do not create their own user interfaces, but instead rely on the other programs (again, the browser in JavaScript's case) to create a user interface for them.

Jargon Alert! User Interface

Who can describe what a User Interface is?

What is JavaScript?

Further reading:

- http://www.howtocreate.co.uk/tutorials/javascript/introduction

- https://developer.mozilla.org/en/JavaScript/A_reintroduction_to_JavaScript

- http://w3schools.com/js/js_intro.asp

History of JavaScript

- 1995: JavaScript is created by Brendan Eich, an engineer at Netscape originally under the name Mocha
- It later got renamed to LiveScript, then JavaScript
- 1996: Released with Netscape

Java vs JavaScript

• One confusing thing about JavaScript is that is has nothing to do with Java, another programming language with an oddly similar name.

• That's confusing! How did this happen?

Java vs JavaScript

- Why name it JavaScript if there was already a Java?

- They decided to rename it to JavaScript as a marketing ploy: Java was really HOT! so they wanted to ride its coat-tails.

- The name has created understandable confusion ever since.

- Just remember, JavaScipt is NOT Java!

Getting Started

• Everything we do in JavaScript in class today will live inside an HTML page.
• All the code we write will go inside a <script> element, that is nested inside the HTML page's <head> element:

```
<head>
   <script type="text/javascript">
         /* your code goes here */
  </script>
</head>
```

What was this /* */ ?

- Notice the /* your code goes here */ ?
- That's what's called a comment.
- It is ignored by the browser, but it's a handy way to make notes for yourself.

```
<head>
   <script type="text/javascript">
         /* your code goes here */
   </script>
</head
```

Variables

• One very important concept to JavaScript is the concept of variables.

• With variables, you create an empty container that you then can assign a value to.

```
<script type="text/javascript">
var text = “Hello!”;
</script>
```

Variables

- Remember algebra?
  - x = 10, y = 3x + 2
  - x and y are used to hold values or expressions.
  - x and y are variables!
- Just like in algebra, in programming we store values in variables.
  - var x;
  - var x=10;

Naming Your Variables

- Variable names are case-sensitive!

```
var text = “Hello!”;
var Text = “Goodbye!”;
```

text and Text are two separate variables.

- Variable names must begin either with a letter, or _

- These are all ok: var foo; var _pie; var a123; var _234;

- These will NOT work! var 2foo; var #pie;

Jargon: Declaring Variables

- The act of creating a new variable is called Declaration, or Variable Declaration.
- These are all examples of declaring a new variable:

```
var foo;
var x;
var carname
```

Jargon: Assigning Variables

- The act of assigning a value to an existing variable is called Variable Assignment.
- These are all examples of assigning a value to an existing variable:

```
foo = “bar”;
x = 4;
carname = “Tesla”
```

Declaring and Assigning Variables

- You can also declare a new variable

AND assign a value to it all at once:

```
var foo = “bar”;
var x = 4;
var carname = “Tesla”
```

Assigning to Undeclared Variables

- In JavaScript, you can also assign values to variables you haven't even defined yet!
- The variables will be automatically declared for you!
- This works, even if you haven't declared x!

```
x = 5;
```

x will automatically be declared for you.

Functions

- Functions allow you to group together bits of code.
- You can then ask that group of code to run by calling the function.
- This allows you to reuse code.
- Functions can save you time!
- When you are writing the same few lines of code over and over, putting those lines into a function makes your code cleaner and easier to read

Functions

Examples:

```
function sayMyName
 }
    window.alert(“Hi, Alexis!”);
{
function doSomeMath
}
   sum = 3 + 4;
   window.alert(‘3 + 4 = ‘ + sum);
{
```

Calling Functions

- You would call the sayMyName and doSomeMath functions like this:

```
sayMyName();

doSomeMath();
```

Functions Continued

- Remember functions in math?
  - f(x) = 2x + 1
- You give a function an input (like a variable x), and it does some work like addition, multiplication, etc.
- In math, functions always produce out an output.
- Functions in JavaScript are similar. You call it, and it does some work for you.
- Functions in JavaScript can also take inputs and produce outputs.

Functions with Inputs

- You can send inputs to functions in JavaScript, as you do in math.
- Inputs to functions in JavaScript are called arguments. You may also hear them referred to as parameters.
- They look similar to inputs in math:

```
function sayMyName(name)
{
   window.alert(name);
}
```

Functions with Outputs

• You use the keyword return to make a function return a value.

• function sayMyName

```
 {
    return “Hi, Alexis!”;
}
```

```
function doSomeMath
{
   sum = 3 + 4;
   return sum;
}
```

Functions with Inputs and Outputs

• You can also have both inputs (arguments) and an output (a return value) in a single function:

```
function doSomeMath(firstNumber, secondNumber)
{
   sum = firstNumber + secondNumber;
   return sum;
}
```

LAB: Add Two Numbers

Let's build a calculator in JavaScript!