

Intro to Web Programming

Clarissa Littler

July 16, 2016

What we'll be covering?

- A short review of HTML and CSS

What we'll be covering?

- A short review of HTML and CSS
- A bare bones introduction to JavaScript

What we'll be covering?

- A short review of HTML and CSS
- A bare bones introduction to JavaScript
- Examples of how to use JavaScript to alter web sites

What we **won't** cover

- All of JavaScript

What we **won't** cover

- All of JavaScript
- How to write a server

What we **won't** cover

- All of JavaScript
- How to write a server
- How HTML and CSS work

What we **won't** cover

- All of JavaScript
- How to write a server
- How HTML and CSS work
- How to use frameworks to build a site

The point of this course

- Start you on the right track

The point of this course

- Start you on the right track
- Give a taste for what web-programming is

The point of this course

- Start you on the right track
- Give a taste for what web-programming is
- Explain the basic pieces

Client and server

Two pieces that talk to each other to make a site

Server

- Sends data to the browser
- Saves information for long term use
- Receives requests from the client

Client

- Receives data from the server
- Renders server data into a usable page
- Handles the user interface

HyperText Markup Language (HTML) is a language that uses *nested tags* to denote what elements a page has and what it should mean

Tags we'll need

- `<h1> ... <h6>`
- ``, ``, ``
- `<button>`
- `<input>`
- `<script>`
- `<style>`

What is CSS

Cascading Style Sheets

CSS provides information on the look and layout of a site

- color
- background-color
- display
- font-weight


```
.aclass {  
  color: red;  
  display: inline;  
}
```

```
#mybutton {  
    font-weight: bold;  
}
```

Tag type

```
p {  
    font-weight: 900;  
}
```

Programming

Programming is

What are programming languages?

A programming language is...

- a formal language with rules and grammar

What are programming languages?

A programming language is...

- a formal language with rules and grammar
- that has meaning as computation

What are programming languages?

A programming language is...

- a formal language with rules and grammar
- that has meaning as computation
- and can be used to talk to a computer

JavaScript in the browser

JavaScript and browsers have a special relationship

Script tag

Direct code

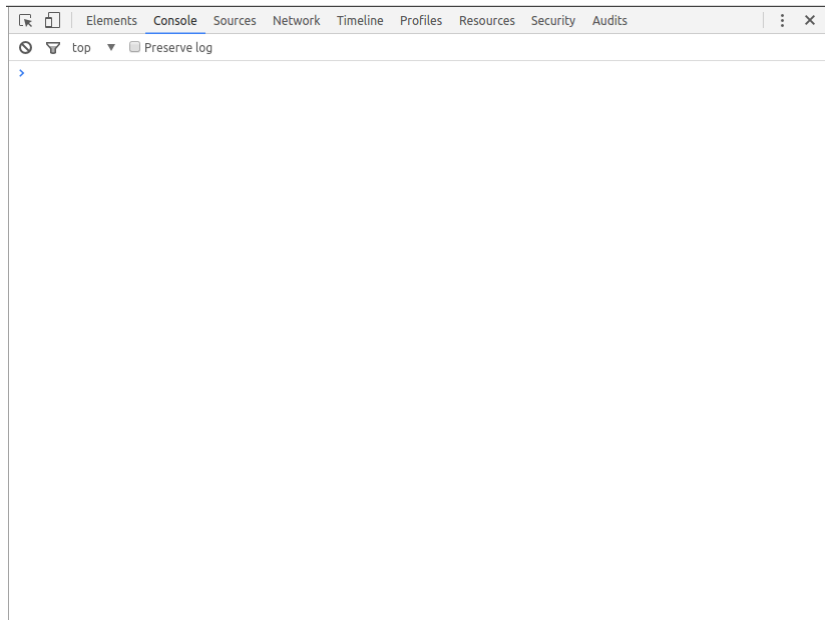
```
<!doctype html>
```

```
<html>  
  <head>  
    <script>  
      ...  
    </script>  
  </head>  
  <body>  
    ...  
  </body>  
</html>
```

Including code

```
<html>  
  <head>  
    <script src="..."></script>  
  </head>  
  <body>  
    ...  
  </body>  
</html>
```

JavaScript console



What do we need to know?

- Bare bones JavaScript

What do we need to know?

- Bare bones JavaScript
- Arithmetic

What do we need to know?

- Bare bones JavaScript
- Arithmetic
- Strings

What do we need to know?

- Bare bones JavaScript
- Arithmetic
- Strings
- Variables

What do we need to know?

- Bare bones JavaScript
- Arithmetic
- Strings
- Variables
- Objects

What do we need to know?

- Bare bones JavaScript
- Arithmetic
- Strings
- Variables
- Objects
- Arrays

What do we need to know?

- Bare bones JavaScript
- Arithmetic
- Strings
- Variables
- Objects
- Arrays
- Functions

What do we need to know?

- Bare bones JavaScript
- Arithmetic
- Strings
- Variables
- Objects
- Arrays
- Functions
- Iteration

Numbers

- 1
- 0.5
- -20
- ...

Operations

- +
- -
- *
- ...

Strings

Strings are text-as-data

```
"this is a string"  
'this is also a string'  
"even this 'is a string'"
```

I have a friend, let's call her "Cassandra"...

Variables function both as storage containers and pronouns

Creating Variables

```
var *name-of-variable* = *initial-value-in-it*;
```

```
var numberOfToes = 10;
```

```
var musicalIllNeverHaveTickets
```

Assigning variables

`*name-of-variable* = *new-value*`

```
var musicalIllNeverHaveTickets
```

```
var musicalIllNeverHaveTickets
```

Sequencing code

```
10 + 10;  
20 + 20;  
var these = "that";
```


Functions in math

$$f(x) = x + 10$$

Functions in JavaScript

```
function f(x) {  
    return x + 10;  
}
```

Using functions

`console.log`

```
console.log("chicken");  
console.log("fish");  
console.log(10 + 20);
```

Multi-argument functions

- Phone books

Objects

- Phone books
- Contact lists

Objects

- Phone books
- Contact lists
- Mall directories

Objects

- Phone books
- Contact lists
- Mall directories
- Dictionaries

Making Objects

```
var obj = {prop1 : 0, prop2 : 1};  
var otherObject = {};
```

Objects

```
var obj = {prop1 : 0, prop2 : 1, prop3 : "thing"};  
  
console.log(obj);  
console.log(obj.prop1);  
console.log(obj.prop2);  
console.log(obj.prop3);
```

Objects

```
var obj = {};  
console.log(obj.numberOfChickens);  
obj.numberOfChickens = 2;  
console.log(obj.numberOfChickens);
```

Arrays

- to-do lists

Arrays

- to-do lists
- book shelves

Arrays

- to-do lists
- book shelves
- instructions

Arrays

```
var list = [10,11,12];  
console.log(list[0]);  
console.log(list[1]);  
console.log(list[2]);  
list[0] = 20;  
console.log(list[0]);  
console.cog(list.length);
```

Iteration

For

What is the Document Object Model?

The DOM

The document object model (DOM) is the representation of the web page as *JavaScript objects*

Putting the document in DOM

`document` is the object that holds most of the important methods

When to load code

```
window.onload = function () {  
    ...  
};
```

Creating elements in code

Creating elements in code

- `document.createElement`

Creating elements in code

- `document.createElement`
- `document.createTextNode`

Creating elements in code

- `document.createElement`
- `document.createTextNode`
- `document.body`

Creating elements in code

- `document.createElement`
- `document.createTextNode`
- `document.body`
- `*element*.appendChild`

Creating elements

```
<!doctype html>
<html>
  <head>
    <script>
      window.onload = function () {
        var newHeadline = document.createElement("h1");
        var textNode = document.createTextNode("This is a head");
        newHeadline.appendChild(textNode);
        document.body.appendChild(newHeadline);
      };
    </script>
  </head>
  <body>
  </body>
</html>
```

Finding elements

Finding elements

- `document.getElementById`

Finding elements

- `document.getElementById`
- `document.getElementsByTagName`

Finding elements

- `document.getElementById`
- `document.getElementsByTagName`
- `*element*.firstChild`

Finding elements

- `document.getElementById`
- `document.getElementsByTagName`
- `*element*.firstChild`
- `*node*.nodeValue`

```
<body>
  <ol id="list1">
    <li>This is a list</li>
  </ol>
  <ol id="list2">
    <li>This is our second list</li>
  </ol>
</body>
```



```
window.onload = function () {  
    var newItem = document.createElement("li");  
    var newText = document.createTextNode("item in the second list");  
    newItem.appendChild(newText);  
    var secondList = document.getElementById("list2");  
    secondList.appendChild(newItem);  
};
```

getElementsByTagName

```
<!doctype html>
<html>
  <head>
    <script src="getElementsByTagName.js"></script>
  </head>
  <body>
    <ol id="list1">
      <li>This is a list</li>
    </ol>
    <ol id="list2">
      <li>This is our second list</li>
    </ol>
  </body>
</html>
```

getElementsByTagName

```
window.onload = function () {  
    var lists = document.getElementsByTagName("ol");  
  
    for(var i = 0; i < lists.length; i++){  
        var list = lists[i];  
        var newItem = document.createElement("li");  
        var newText = document.createTextNode("new element");  
        newItem.appendChild(newText);  
        list.appendChild(newItem);  
    }  
};
```

Changing CSS properties

```
<!doctype html>
<html>
  <head>
    <script>
      window.onload = function () {
        var h = document.getElementById("headline");
        h.style.color = "red";
      }
    </script>
  </head>
  <body>
    <h1 id="headline">This is a headline!</h1>
  </body>
</html>
```

Changing the CSS class

```
<!doctype html>
<html>
  <head>
    <style>
      .reddish {
        color: red;
      }
    </style>
    <script>
      window.onload = function () {
        var h = document.getElementById("headline");
        h.classList.add("reddish");
      };
    </script>
  </head>
  <body>
    <h1 id="headline">This is a headline</h1>
```

Events

Events connect user interfaces to code

Listening to events

```
<!doctype html>
<html>
  <head>
    <script>
      window.onload = function () {
        var h = document.getElementById("headline");

        h.addEventListener("mouseover", function () {
          this.style.color = "red";
        });

        h.addEventListener("mouseleave", function () {
          this.style.color = "black";
        });
      };
    </script>
  </head>
```

Collapsing list

```
<body>
  <div id="content">
    <h3>Our list is below here</h3>
    <ol id="list">
      <li>First item</li>
      <li>Second item</li>
      <li>Third item</li>
      <li>Fourth item</li>
    </ol>
  </div>
</body>
```


Collapsing list

```
window.onload = function () {  
    var list = document.getElementById("list");  
    var div = document.getElementById("content");  
    div.addEventListener("mouseover", function () {  
        list.style.display = "block";  
    });  
    div.addEventListener("mouseleave", function () {  
        list.style.display = "none";  
    });  
};
```

To-do list

```
<body>
  <h1>Welcome to your to-do list</h1>
  <ol id="list">
</ol>
  <input id="input" type="text"></input>
  <button id="add">Add element</button>
</body>
```

To-do list

```
window.onload = function () {  
    var inputElement = document.getElementById("input");  
    var todoList = document.getElementById("list");  
    var addButton = document.getElementById("add");  
  
    addButton.addEventListener("click", function () {  
        var itemText = document.createTextNode(inputElement.value);  
        var newItem = document.createElement("li");  
        newItem.appendChild(itemText);  
        todoList.appendChild(newItem);  
        inputElement.value = "";  
    });  
  
    inputElement.addEventListener("focus", function () {  
        inputElement.style.fontWeight = "bold";  
    });  
};
```

What's left

- More