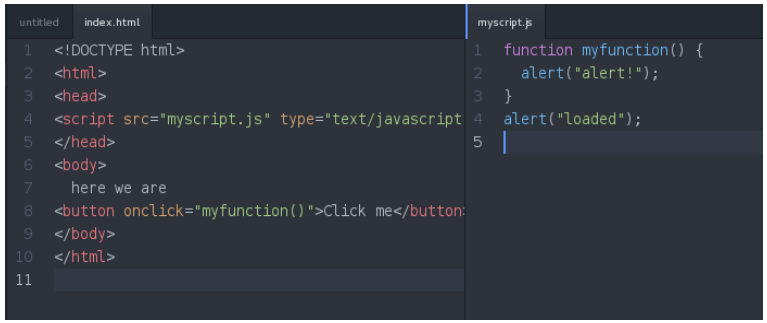
EECS 1012: Introduction to Computer Science

September 30, 2016

From HTML to JavaScript

- Certain HTML tags support user interaction (selection). Such as buttons
  - <button onclick="myfunction()">Click Me</button>
- When clicked invokes the myfunction in your JavaScript



Note

- atom does not display alert's
- browsers' do
- the html2apk application does too

Slightly more complex

- We can make the callback code/ button handler very complex.
- Lets make it **change** the way the viewed page.

document.getElementById()

- refers to the HTML element with that id
  - remember id? <tag id="foo"> this stuff </tag>
- var x = document.getElementById("foo");
  - obtains a reference to that portion of the HTML page
- x.innerHTML = "that stuff" changes it

index.html | myscript.js | index.html Preview

```
<!DOCTYPE html>
<head>
<script src="myscript.js" type="text
</head>
<bod>
<button onclick="myfunction()">Click
<div id="debug"> this text </div>
</body>
</html>
```

Click me
this text

Note typo above. Can you find it?

```
function myfunction() {
  alert("alert!");
  var x = document.getElementById("debug");
  x.innerHTML = "that text";
}
```

JavaScript validator

- JSLint (removes fluff from code)
- It is very picky.

JavaScript

- Large, complex language
  - It looks a bit like Java, acts a bit like C, and adds its own special wrinkles
- We will not cover all of it, nor will we cover it in a formal manner. Rather, the course will provide multiple examples of how things are done, with formal descriptions added as needed.

An example

- Goal here is to understand the code (not necessarily to be able to write this from scratch)

Canvas (as an example)

```
<DOCTYPE! html>
<html>
<head>
<script type="text/javascript" src="canvas.js"></script>
</head>
<body>
<canvas width="640" height="480" id="mycanvas"> </canvas>
</body>
</html>
```

Canvas

```
<DOCTYPE! html>
<html>
<head>
<script type="text/javascript" src="canvas.js"></script>
</head>
<body>
<canvas width="640" height="480" id="mycanvas"> </canvas>
</body>
</html>
```

The HTML <canvas> element is used to draw graphics on a web page.

The graphic to the left is created with <canvas>. It shows four elements: a red rectangle, a gradient rectangle, a multicolor rectangle, and a multicolor text.

```
function drawCanvas() {
    alert("drawCanvas called");
    var id = document.getElementById("mycanvas");
    var cx = id.getContext("2d");
    cx.beginPath();
    cx.moveTo(0,0);
    cx.lineTo(639,479);
    cx.closePath();
    cx.stroke();
}

window.onload = drawCanvas;
```

```
function drawCanvas() {
    alert("drawCanvas called");
    var id = document.getElementById("mycanvas");
    var cx = id.getContext("2d");
    cx.beginPath();
    cx.moveTo(0,0);
    cx.lineTo(639,479);
    cx.closePath();
    cx.stroke();
}

window.onload = drawCanvas;
```

Syntax

```
window.onload = funcRef;
```

- funcRef is the handler function to be called when the window's load event fires.

```
function drawCanvas() {
    alert("drawCanvas called");
    var id = document.getElementById("mycanvas");
    var cx = id.getContext("2d");
    cx.beginPath();
    cx.moveTo(0,0);
    cx.lineTo(639,479);
    cx.closePath();
    cx.stroke();
}

window.onload = drawCanvas;
```

invoke the method getElementById() on the document object

```
function drawCanvas() {
    alert("drawCanvas called");
    var id = document.getElementById("mycanvas");
    var cx = id.getContext("2d");
    cx.beginPath();
    cx.moveTo(0,0);
    cx.lineTo(639,479);
    cx.closePath();
    cx.stroke();
}

window.onload = drawCanvas;
```

invoke the getContext method on the canvas element
This is a drawing context

```
function drawCanvas() {
    alert("drawCanvas called");
    var id = document.getElementById("mycanvas");
    var cx = id.getContext("2d");
    cx.beginPath();
    cx.moveTo(0,0);
    cx.lineTo(639,479);
    cx.closePath();
    cx.stroke();
}

window.onload = drawCanvas;
```

Large number of drawing context methods that 'draw' on the canvas