

# HTML5 + JavaScript

Dylan McClure  
NHMFL Optical Microscopy



## Abstract

Tutorials written in Java as Java Applets were ported to HTML5 and JavaScript. Interactivity was provided via JavaScript using Canvas, ThreeJS and the built-in JavaScript tools while programming in and learning about the powerful modal text editor, vim.

## HTML, CSS, and JavaScript

### Overview

Almost all web pages today are comprised of HTML, CSS and Javascript. These languages are written in plain text but are structured in such a way that web browsers can interpret. Each of them that tells the web browser how to do a specific thing.

### HTML Hypertext Markup Language

HTML is used to represent content – like images and text. It is comprised of just one thing: tags. Tags can store other tags or plain text. An opening and closing tag comprise an element.

### CSS Cascading Style Sheet

CSS is tells the browser how to color, position, and otherwise style HTML elements.

### JS JavaScript

JavaScript is a very powerful addition to HTML and CSS, and it can reference both the HTML and CSS elements. It can also listen to and respond to mouse and keyboard interaction.

## CSS

mark	type	description
<code>class</code>	selector	selects html class
<code>id</code>	selector	selects html id
<code>property</code>	property	property that governs how an element looks or behaves; “color” sets background
<code>value</code>	value	describes the property; “absolute” positions without contextual awareness

```

class .mainscene {
  position: relative;
  width: 400px;
  height: 400px;
  background: #002b36;
  margin: 0 auto;
}
.tutorialTitle {
  text-align: center;
  width: 100%;
  line-height: 400px;
  color: #839496;
}
#myCan {
  position: absolute;
  left: 0px;
  top: 0px;
  width: 100%;
  height: 100%;
}

```

## CSS

```

1 <!doctype html> }doctype
2
3 <head>
4   <link rel="stylesheet" type="text/css" href="styles/myStyle.css" />
5 </head>
6
7 <body>
8   <div class="mainscene">
9     <div class="tutorialTitle">Tutorial Name!</div>
10    <canvas id="myCan" width="400" height="400"></canvas>
11  </div>
12
13  <script type="text/javascript" src="js/jquery.min.js"></script>
14  <script type="text/javascript" src="js/myJs.js"></script>
15 </body>

```

mark	type	description
<code>doctype</code>	tag	special syntax that denotes this is html5
<code>head</code>	element	place for metadata and loading CSS
<code>body</code>	element	place for main content of web page
<code>href</code>	attribute	loads “myStyle.css” from folder “styles/”
<code>inner content</code>	inner-HTML	text or other elements can be placed between opening and closing tags
<code>script</code>	elements	how to load external scripts

## JS

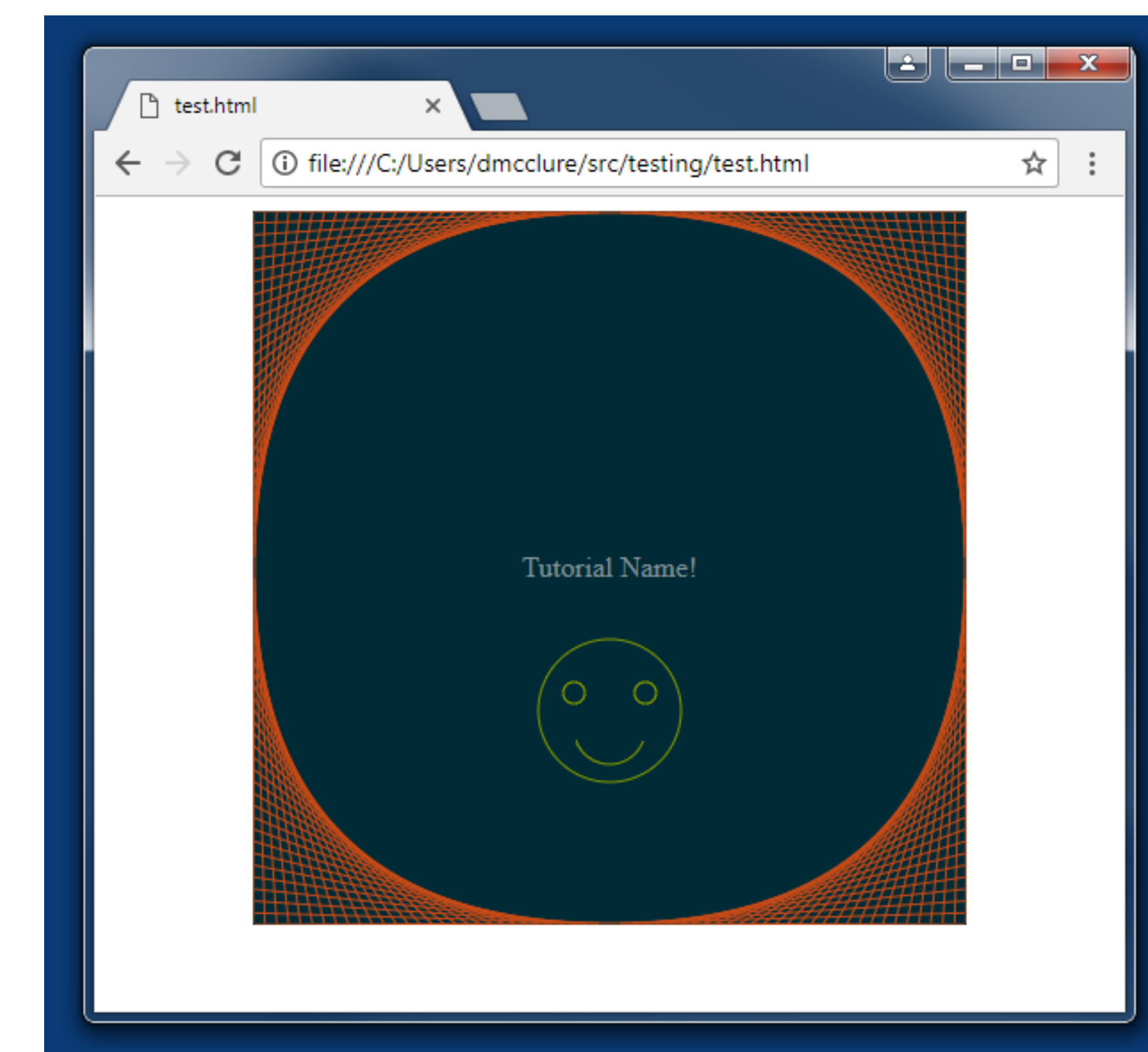
```

1 function main() {
2   var canvas = document.getElementById("myCan"), ctx = canvas.getContext("2d"),
3       canWidth = myCan.width,
4       canHeight = myCan.height;
5
6   drawCanvas();
7
8   function drawCanvas() {
9     var steps = 15, stepW = canWidth / steps / 2, stepH = canHeight / steps / 2;
10    ctx.strokeStyle = "#cb4b16"; /* nice orange */
11    ctx.beginPath(); /* start drawing a new object */
12    for (var i = 0, steps = steps + 1; i < steps; i++) {
13      /* top-left web */
14      ctx.moveTo(1, canHeight / 2 - i * stepH);
15      ctx.lineTo(i * stepW, 1);
16      /* top-right web */
17      ctx.moveTo(i * stepW + canWidth / 2, 1);
18      ctx.lineTo(canWidth - 1, i * stepH);
19      /* bottom-right web */
20      ctx.moveTo(canWidth - 1, canHeight / 2 + i * stepH);
21      ctx.lineTo(canWidth - i * stepW, canHeight - 1);
22      /* bottom-left web */
23      ctx.moveTo(canWidth / 2 - i * stepW, canHeight - 1);
24      ctx.lineTo(1, canHeight - i * stepH);
25    }
26    ctx.stroke(); /* finally, stroke all the lines drawn */
27  }
28 }
29
30 window.onload = main;

```

mark	description
<code>globals</code>	defines variables for entire scope of “main” function
<code>lines</code>	relevant code that draws lines on canvas’s context

## Result



## Vim

### Overview

Since software developers are editing text a lot, text editors tend to be very important to them.

### Vim Vi iMproved

Vim is a very powerful tool for developers that is designed to do one thing: edit text efficiently. Despite its old code base, it is quite powerful. It was released in 1991 but has grown considerably since then into a mature and powerful editor.

### Modality normal insert command visual select ex

In vim, you can be in one of six modes. The most common are “normal” and “insert” mode. In normal mode you can perform actions. Each action is mapped to a key, a combination of keys, or a sequence of key strokes. You can remap all of these keys in any of the modes.

<b>normal</b>	mode in which most movements and commands are performed; any other mode can also be accessed from <b>normal</b> mode
<b>insert</b>	allows characters to be inserted into buffer as they are typed. <code>&lt;esc&gt;</code> returns to <b>normal</b> mode
<b>command</b>	entered by typing <code>:</code> – this mode allows user to use vimscript and run built-in or shell commands
<b>visual</b>	block, line, or in-line select that defines a visual selection for using commands on a region
<b>select &amp; ex</b>	<b>ex</b> – for compatibility; <b>select</b> – less-useful <b>visual</b>