1. Introduction to Javascript

For this lab, you will practice using functions to modify attributes of elements.

2. Facts, Words, Skills

The main ideas for this stage in our learning how to build a web-based slide show are:

- A webpage consists of a collection of elements
- Each element on a webpage can have several attributes
- The attributes of an element are specified in the element’s tag
- Different types of elements have different attributes
- The head of a webpage can contain a script section
- The script section of a page contains functions
- The elements on a page can receive events
- Elements on a page can call functions when they receive events
- The functions can modify attributes of elements on the page

The vocabulary words you must understand by the end of this lab are:

- element
- tag attributes
- link
- table
- table row
- table data
- built-in attributes
- style attributes
- color codes
- events
- functions
- script

The skills you must master during this lab are:

- How to set element attributes to control the look of a page
- How to write functions in the <head> of a webpage
- How to tell an element to call a function
- How to tell a function to modify attributes of elements

3. Setting Up: Create a Directory on Your Website

a. Create a directory called lab07 in your website.
b. In that directory make directories called small and big
c. Copy 3 small and big images from some other project into small and big

4. Remember Scratch?

For the last part of the course, we shall write scripts to animate web pages. We began the course writing scripts to animate sprites. In Scratch, we had a stage that displays sprites. In HTML, we have a page that displays elements.

How did we get a Scratch stage to become animated? Easy, we wrote scripts that modified attributes of sprites. We could control the color, the size, the direction, the position, etc, of sprites by writing code. And we executed that code by attaching scripts to events.
5. Javascript Works Sort of Like Scratch

Javascript brings to web pages what Scratch brings to sprites on a stage. Javascript uses the idea of events that trigger code and code that modifies attributes. Here’s the picture:

We put code somewhere on the page, and we connect code to events. Today, we work with a sample page. Copy this HTML file from the course area by typing:

```bash
cp /comp/10IDI/files/lab07/lab07a.html lab07a.html
```

This file contains:

```html
<html><head>
<!-- lab07a.html my first javascript code -->
<title>Javascript demo1</title>
<script type='text/javascript'>
// this is how we define functions in javascript
function newcolor()
{
    alert("Hi!");
    document.body.style.backgroundColor = 'red';
}
</script>
</head>
<body style='background-color:blue; color:white;'><div style='text-align: center; width: 350;'><h3 style='color: cyan; text-align: right; border-style: double; border-color: papayawhip; padding: 4px; background-color: plum;'>Javascript Demo</h3><input type='button' value=' click me ' onclick='newcolor()'/><img src='small/1.jpg' id='pic1' border='2' width='100'/>
</div></body>
</html>
```

**Step 1. Put Code in head of Page**

To program action on a web page dynamic, put code in the head of the page. The code is enclosed by `<script>` tags. The code consists of functions. These functions are just like scripts in the Unix shell and like scripts in Scratch. Each function is a chunk of instructions that is given a name.

**Step 2. Attach functions to Events on Elements**

In Scratch, we attach scripts to events by putting a special puzzle piece at the top. In Javascript, we tell the elements what to do when an event happens to the element. Look at the body part of the page. The button element has an attribute called `onclick`. This `onclick` attribute is set to the function call `newcolor()`. When the button is clicked (computer lingo: when the button receives the click event), the function called newcolor is called.

**Step 3. Have Functions Modify Elements**

That newcolor function then does two things. The first thing it does is to call a javascript function called `alert()`. What does that do?

The next thing the newcolor function does is to set something called `document.body.style.backgroundColor` to the value ‘red’. What does that do? It changes an attribute of an element of the page. This is just like a mouse click changing the color of a sprite. An element receives an event, the event causes some code to be executed. The code modifies an attribute of an element. Same idea, different syntax.
6. **An IMPORTANT NOTE** on Style Components

In the style part of the body tag, we write `background-color: blue`, but in the `newcolor()` function, we set the attribute by writing `backgroundColor = "red"`. Why do we use a dash in the HTML part of the page and use a capital C and no dash in the Javascript part of the page? It turns out that Javascript uses slightly different names for style components. If the HTML version of a style component has a dash, the Javascript version has no dash and an upper case letter for the second word. Javascript thinks that a dash means subtraction, and HTML does not know about subtraction.

7. Your Turn

Now, modify this page so it does the following new things:

- a. Change the first button so it says "Red"
- b. Add a new button that says "Blue" and changes the background to blue

8. Debugging Javascript

In Firefox, look at the menu bar and find the Tools item, and from there, select Error Console. A window will open. Click the Clear button to clear the history. This window displays any problems with your Javascript code. Click on the error messages to see the offending lines of code.

9. Elements, Attributes, Events, Functions

We now have two buttons, each responds to clicks, and each calls a function that modifies an attribute of the body.

What if we wanted to change the background by clicking on the image? Can an image respond to clicks? Yes!

- c. Add an `onclick` attribute to the `img` element that changes the background to green.

Other Events

In Scratch, we saw lots of events we could handle. Javascript has lots, too:

<table>
<thead>
<tr>
<th>Javascript Attributes for Event Handling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
</tr>
<tr>
<td>onAbort</td>
</tr>
<tr>
<td>onblur</td>
</tr>
<tr>
<td>onChange</td>
</tr>
<tr>
<td>onClick</td>
</tr>
<tr>
<td>onDblClick</td>
</tr>
<tr>
<td>onError</td>
</tr>
<tr>
<td>onFocus</td>
</tr>
<tr>
<td>onkeydown</td>
</tr>
<tr>
<td>onkeypress</td>
</tr>
<tr>
<td>onkeyup</td>
</tr>
<tr>
<td>onLoad</td>
</tr>
<tr>
<td>onmousedown</td>
</tr>
<tr>
<td>onmousemove</td>
</tr>
<tr>
<td>onmouseout</td>
</tr>
<tr>
<td>onmouseover</td>
</tr>
<tr>
<td>onmouseup</td>
</tr>
<tr>
<td>onreset</td>
</tr>
<tr>
<td>onresize</td>
</tr>
<tr>
<td>onselect</td>
</tr>
<tr>
<td>onsubmit</td>
</tr>
<tr>
<td>onunload</td>
</tr>
</tbody>
</table>

Changing Attributes of Elements: The Need for id

We have written functions that change the background color of the body. We can use Javascript to change any
attribute of any element. All we need to do is to give the element an identifier so we can tell the browser what element we want to change. In this page, the image has an id of pic1.

**Do These Things**

d. Add a third button marked "200" that changes the image width to 200

e. Add a fourth button marked "50" that changes the image height to 50

f. Modify the IMG element so when you click the image the border becomes 20
   Note: the background should still turn green when you click the image

g. Add code so the h3 text becomes left-justified and the border color becomes yellow when the mouse is over the image

h. Add code so the h3 text becomes right-justified and the border color returns to papayawhip when the mouse leaves the image

i. Add code so the "Red" button label changes to "Hi!" when you click the h3 header

j. Add code so the "Red" button label returns to "Red" when you click the "Blue" button

**10. Extra Credit: Tables and JavaScript**

If you get done with some time to spare, and you would like some more practice, try this problem.

Write an HTML file called lab07-b.html that uses a table to create the page shown at left. Instead of the three images shown put IMG tags to display small/1.jpg, small/2.jpg, and small/3.jpg. If your files end in .JPG, be sure your to use the upper case names in your html.

The table has three rows. Row one has a centered caption. Row two has three images in separate cells. Give each of those cells height=100 and width=100. Row three has one big cell. The contents of that cell should be

```html
<IMG height='300' width='300' id='bigpic' />
```

Also, make the background of the page be lightblue.,

Also, give each IMG a border='3' attribute .

The Problem: Modify the three image tags and add Javascript so that when the user moves the mouse over any of the three small images, the larger version of that image appears in the big cell at the bottom of the table.