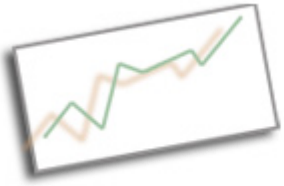

Coding Workshop

Dr. Cindy Royal
Texas State University
School of Journalism and Mass Communication

Coding Workshop

This workshop is an introduction of HTML basics, CSS, HTML Forms and JavaScript. At the end of the workshop, you will have an appreciation of all you can accomplish developing Web applications.

In the TRACS modules, we covered all the concepts you need to know. Here is a review.

HTML: Hypertext Markup Language; the language of the Web. Structures information on a Web page.

CSS: Cascading Stylesheets; provides the design and layout of a Web page; a single CSS page can control an entire site and make for easy and consistent maintenance.

JavaScript: a programming language that works well in a browser, plays nice with HTML. It's not the same as Java. It is primarily a client-side language that runs on the users computer, allowing you to create interactive applications.

Programming is problem solving. This workshop will provide you with many tools, but as you continue with programming, the approach you take to solving the problem is what is most important. We'll be thinking through the algorithms (steps), and we'll execute the algorithm with code.

HTML/CSS Concepts

Tags or Elements – things like<p>, <h1>, <img>
Attributes – options that the element can use. Use with the value (in quotes) of the option. For example <a href=http://www.google.com>Google</a>.

Ids and Classes – use the # or . to create ids and classes. Use an id once on a page, use classes as often as you'd like.

Styles – create styles within { }. For example:

```
p {
color: gray;
font-family: Arial;
}
```

Setting Up an HTML Page

We are going to be working on a page that will ultimately contain an interactive quiz. To start, we will create the page with the questions. We'll then add the form elements. Later, we'll make it interactive, so it can score the quiz.

In a text editor like Text Wrangler, start an HTML page as follows. Name it quiz.html. Save frequently as you go through the workshop

```
<html>
<head>
<title>JavaScript Quiz</title>
</head>
<body>
<h1>JavaScript Quiz</h1>
<p>What is your year in school?</p>
<p>Which is the best description of a variable?</p>
<p>Which is the best description of the substring method?</p>
<p>Which is the best description of concatenation?</p>
<p>Which is the best description of an if statement?</p>


</body>
</html>
```

Open this page in a browser. See how it looks. Right now, it's just basic text.

Basic CSS

Add this in the head section of the document. We'll add styles within this style section.

```
<style>
body {
background-color: lightblue;
font-family: Helvetica;
}


</style>
```

Creating Sections on a Page

We want to create two sections, a main section and a header within the page to contain an image. Create <div id = “main”> to contain all the content on the page. Close it before the </body>.

We’ll create a header section to hold the heading and a background image.

```
<body>
<div id=main”>
<div id="header"><h1>JavaScript Quiz</h1></div>


---all the rest of the content on the page---


</div>
</body>
```

Below the quiz in the html, add this code to create paragraphs to hold our answers. The ids will change via JavaScript later with getElementById when the user submits the form.

```
<p>Your grade is: <span id="grade">__</span></p>
<p id="grade2"></p>
```

CSS for Layout

Now that you have these sections created, let’s add some CSS to place and style them.

Add this in the <style> section of the page. We’ll discuss what each does.

```
#main {
width: 80%;
max-width: 950px;
border: 1px gray solid;
margin: auto;
padding: 10px;
background-color: white;
border-radius: 10px;
}

#header {
margin-top: 0;
border: 2px solid black;
padding: 5px;
height: 250px;
background: beige;
background-image: url("code.jpg");
color: white;
}
```

HTML Forms

The page looks pretty good like this. We've created sections with design and layout. Next we need to create the form elements for the quiz.

For the items in the form, we need to surround by the <form> element.
After the header, but before the first question, include this:

```
<form  id="form1" name="form1" action=" ">
```

Close the form with </form> at the end of the questions.

There area variety of form elements that use the <input> tag.
<input type="text"> - This captures a short text input.
<input type="radio"> - This creates a radio button (can choose one in a group). Use with name attribute for the grouping.
<input type="checkbox"> - This creates a checkbox (can choose more than one in a group). Use with name attribute for the grouping.

You can also use the <textarea> for a longer text comment.

You can use <select> and <option> to create a dropdown.

For the first question, let’s create a dropdown:

```
<p>What is your year in school?</p>
<select id="thedropdown">
   <option value="0">Choose One</option>
  <option value="1">Freshman</option>
  <option value="2">Sophomore</option>
  <option value="3">Junior</option>
  <option value="4">Senior</option>
  <option value="4">Other</option>
</select>
```

For the rest of the questions, the selections will be the same, but the name and correct answer will change. For each one, add the following:

```
<p>Which is the best description of a variable?</p>
<input type="radio" name="variable" value="0" />Identifies a portion of a string.<br />
<input type="radio" name="variable" value="0" />A method to join strings.<br />
<input type="radio" name="variable" value="25" />Allows you to store information so it can be reused throughout the program<br />
<input type="radio" name="variable" value="0" />Allows you to make a decision based on a condition.<br />

<p>Which is the best description of the substring method?</p>
<input type="radio" name="sub" value="25" />Identifies a portion of a string.<br />
<input type="radio" name="sub" value="0" />A method to join strings.<br />
<input type="radio" name="sub" value="0" />Allows you to store information so it can be reused throughout the program<br />
<input type="radio" name="sub" value="0" />Allows you to make a decision based on a condition.<br />

<p>Which is the best description of concatenation?</p>
<input type="radio" name="con" value="0" />Identifies a portion of a string.<br />
<input type="radio" name="con" value="25" />A method to join strings.<br />
<input type="radio" name="con" value="0" />Allows you to store information so it can be reused throughout the program<br />
<input type="radio" name="con" value="0" />Allows you to make a decision based on a condition.<br />

<p>Which is the best description of an if statement?</p>
<input type="radio" name="ifstate" value="0" />Identifies a portion of a string.<br />
<input type="radio" name="ifstate" value="0" />A method to join strings.<br />
<input type="radio" name="ifstate" value="0" />Allows you to store information so it can be reused throughout the program<br />
<input type="radio" name="ifstate" value="25" />Allows you to make a decision based on a condition.<br />
```

Go through and add these for each questions, modifying the value to have 25 points for the correct answer and 0 for incorrect. And give each a unique name.

Finally, we need a Submit button. Include this before the </form>

```
<input type="submit" value="Submit">
```

Adding Interactivity with JavaScript

So far, we have created a nice HTML page with some styling. We have a form in it with several questions and answers. But now we need to add the functionality to the form with JavaScript. We'll do a little practice with these concepts in the Chrome console first, the return to our quiz.

See the JavaScript handout.

JavaScript for the Quiz Application

Open the <script> element after the elements on the page, but before the </body> closing tag. Create the function that happens when we submit the form.

```
<script>
document.getElementById("form1").onsubmit=function() {



return false; // required to not refresh the page; just leave this here
} //this ends the submit function

</script>
```

Create a variable for the year from the dropdown.

```
year=document.getElementById("thedropdown").value;
```

To get the value of a radio button, it's a little more complex. We have to evaluate each answer in a loop to determine which one is checked.

Here is what I did for the first one:

```
variablevar = document.form1.variable;
          for(i=0; i<variablevar.length; i++) {
          if (variablevar[i].checked) {
          vscore = parseInt(variablevar[i].value);

          }
          }//end for
```

Create one of these for each of the four radio questions. It should look as follows. Note that the form provides a string, but we ultimately want to add these scores together as integers. So, the final line stores the value in a variable and parses it into an integer (parseInt).

```
variablevar = document.form1.variable;
for(i=0; i<variablevar.length; i++) {
        if (variablevar[i].checked) {
        vscore = parseInt(variablevar[i].value);
        }
}//end for

subvar = document.form1.sub;
for(i=0; i<subvar.length; i++) {
        if (subvar[i].checked) {
        sscore = parseInt(subvar[i].value);
        }
}//end for

convar = document.form1.con;
for(i=0; i<convar.length; i++) {
        if (convar[i].checked) {
        cscore = parseInt(convar[i].value);
        }
}//end for

          ifvar = document.form1.ifstate;
          for(i=0; i<ifvar.length; i++) {
          if (ifvar[i].checked) {
          iscore = parseInt(ifvar[i].value);
          }
}//end for
```

Then we need to add up the score and place it somewhere on the html page.

At the end of the script, add these lines to calculate the score.

```
result = vscore + sscore + cscore +iscore;
document.getElementById("grade").innerHTML = result;
```

The second line above will take the sum of the four scores and add modify the #grade element in the html page.

Additional Functionality
We didn’t do anything with the Year dropdown. For this, let’s just make sure that they choose something. We will start out with an if statement that provides a line if they don’t choose something. Include this line after the year variable is created.

```
if (year == 0) {result2 = "<strong>Please select a year</strong>"; }
```

Then everything else on the page will be wrapped in the else code

```
else {

--everything else we have coded in the script--

}
```

We also have a #grade2 id in the html. This is meant to hold a comment that changes based on the grade. Put this below the “grade” area, but still within the else statement.

```
if (result == 0) {result2 = "I don't think you studied."};
if (result == 25) {result2 = "You need to spend more time. Try again."};
if (result == 50) {result2 = "I think you could do better. Try again.”};
if (result == 75) {result2 = "So close. Try again.”};
if (result == 100) {result2 = "Excellent! You're a JavaScript pro!”};
```

And include this right after the else closes, but still in the submit function, before the return false statement.

```
document.getElementById("grade2").innerHTML = result2;
```

Or you can get really fancy and add some memes. Notice I used single quotes in the img html, because we can’t have quotes inside of the quotes already used for the JavaScript string. The images are in the folder provided from the GitHub repository.

```
if (result == 0) {result2 = "I don't think you studied.<br /><img src='0.jpg'
width='300' />"};
if (result == 25) {result2 = "You need to spend more time. Try again.<br
/><img src='25.jpg' width='300' />"};
if (result == 50) {result2 = "I think you could do better. Try again.<br
/><img src='50.jpg' width='300' />"};
if (result == 75) {result2 = "So close. Try again.<br /><img src='75.jpg'
width='300' />"};
if (result == 100) {result2 = "You're a JavaScript pro!<br /><img
src='100.jpg' width='300' />"};
```

Test your program. What do you like about it? What changes would you make?

You’ve created an interactive quiz using HTML, CSS and JavaScript. Congrats!