**ACES Learn to Code**

**HTML, JavaScript, CSS**

**Details**

This Learn to Code session is geared toward those that want to learn software development for those unsure where to start. At the end of this session you will:

1. Learn the basics HTML objects
2. Learn the basic features of JavaScript and integrate them with HTML objects
3. Learn the basics of CSS
4. Create an executable JavaScript program and display it in an HTML page

**Before the Event**

1. Review - HTML CheatSheet - <http://www.simplehtmlguide.com/cheatsheet.php>
2. Review JavaScript CheatSheet - <https://www.cheatography.com/davechild/cheat-sheets/javascript/>
3. Review CSS CheatSheet - <http://www.simplehtmlguide.com/csscheatsheet.php>
4. Download a Text/HTML editor
   * <https://atom.io/>

**Extra Resources**

* HTML - <http://www.w3schools.com/html/default.asp>
* JavaScript - <http://www.w3schools.com/js/default.asp>
* CSS - <http://www.w3schools.com/css/default.asp>

**Training**

1. Create a new file and save it as ‘index.html’
2. Open the file ‘starter\_index.html’ and copy the contents to your newly created ‘index.html’ file; save these changes.
3. Navigate to your ‘index.html’ file double-click on it to open it in a web browser
4. If you see a page with “ACES Learn To Code” you are ready to move on!

**Plan**

* Create a product page for T-shirts
  + Title
  + Image
  + Product Description
  + Form
    - Choose Color
    - Size
    - Quantity
  + Order Button
* Display the user’s selection in an alert box
* Style the page using CSS
* TO DO ADD FROM ERIC’S NOTES

# HTML

## Product

The first thing a user should see when they visit this page is the product they are interested in. That means they need a visual, a title and description, a price, and a way to buy it. Everything that comes after that is additional information for the more discerning buyer.

Create a designated space for this information:

<div id=”content”>

<div id=”product”>

</div>

</div>

The div tag has no meaning at all. It is a generic, all-purpose tag typically used for grouping other tags together in such a way that it is easier to specify those elements with either CSS or JavaScript.

We’ll break this section into two columns:

<div id=”product”>

<div class=”onethird”>

</div>

<div class=”twothird”>

</div>

</div>

Tags can have attributes, which follow the format attribute=”value”. The most common attributes you’ll find are “id” and “class”. These are identifiers which give JavaScript and CSS the ability to target particular elements. The values of these attributes can be basically anything you can imagine, but it’s useful to try and be descriptive.

The difference between a “class” and an “id” is classes can be repeated. There can only be one of each “id” on a given page.

### Column 1

In the left-hand column we’ll want to put an image:

<div class=”onethird”>

<img src="images/shirts.jpg" alt="Smiley face" height="175" width="175" />

</div>

### Column 2

In the right-hand column we’ll want to put the rest of the information, and we’ll put it all in order of importance (i.e. Name, Price, Call to Action, Description).

<div class=”twothird”>

<h1>Plain Colored T-Shirts </h1>

<form id="addToCart">

</form>

<h2>Description</h2>

<p>These shirts are very shirty. They are the shirtiest shirts of any shirt that any shirtmaker in the history of shirtmaking has ever crafted.</p>

</div>

The p tag represents paragraphs. The h# tags represent headings of varying sizes (1-6; large-small).

#### Form

The form will contain the price, as well as options for quantity, color, and size.

<form id=”addToCart”>

<h2 id="price">$14.99</h2>

<p>

<label for="num">Quantity:</label>

<input type="text" id="num" name="num" size="5" />

</p>

<p>

<label for="color">Color:</label>

<select id="color" name="color">

<option value="orange">Orange</option>

<option value="blue">Blue</option>

<option value="green">Green</option>

<option value="yellow">Yellow</option> </select>

</p>

<p>

<label for="size">Size:</label>

<span class="oneline">

<label><input type="radio" name="size" id="size1" value="S" /> Small</label>

<label><input type="radio" name="size" id="size2" value="M" /> Medium</label>

<label><input type="radio" name="size" id="size3" value="L" /> Large</label>

<label><input type="radio" name="size" id="size4" value="XL" /> X-Large</label>

</span>

</p>

<p>

<input type="button" value="Order" id="submit" class="bigButton" onclick="order();" />

</p>

</form>

Each input has a label with a “for” attribute. This “for” attribute matches the “id” attribute of its paired input. When you click on the label on your screen, it will make the paired input active by selecting it/moving the cursor into it.

The “type” attribute of each input determines what it looks like on the screen. Text is a box you can type in, select is a dropdown menu with options, radio is an option (typically grouped together using the “name” attribute) belonging to a group so only one of the options can be selected.

## Specs

Product specs are a good place to put tabular information for a buyer that is curious about very specific details. So, just like we did for the product, lets create a designated space for this information:

<div id=”product”>

…

</div>

<hr />

<div id=”productspecs”>

</div>

The hr tag reprents a horizontal rule, or a line that separates the page.

### Table

<div id=”productspecs”>

<h2>Product Specs</h2>

<table>

<thead>

<tr>

<th width="20%">&nbsp;</th>

<th width="20%">Orange</th>

<th width="20%">Blue</th>

<th width="20%">Green</th>

<th width="20%">Yellow</th>

</tr>

</thead>

<tbody>

<tr>

<th>Product Dimensions</th>

<td>13 x 8 x 1 inches</td>

<td>12 x 7 x 0 inches</td>

<td>15 x 5 x 5 inches</td>

<td>22 x 2 x 2 inches</td>

</tr>

<tr>

<th>Shipping Weight</th>

<td>1 pounds</td>

<td>2 pounds</td>

<td>15 pounds</td>

<td>1 pounds</td>

</tr>

<tr>

<th>Barcode</th>

<td>B00B81I4IP</td>

<td>B00B81E4IO</td>

<td>B00B81A4IR</td>

<td>B00B81T4IK</td>

</tr>

<tr>

<th>Item model number</th>

<td>FTL-4A3001C</td>

<td>STL-4B3001A</td>

<td>FTP-4I3001R</td>

<td>FRT-4G3001P</td>

</tr>

<tr>

<th>Date first available</th>

<td>May 7, 1979</td>

<td>May 7, 1980</td>

<td>May 7, 1981</td>

<td>Feb 29, 2016</td>

</tr>

<tr>

<th>Domestic Shipping</th>

<td>This item is also available for shipping to select countries outside the U.S.</td>

<td>This item is also available for shipping to select countries outside the U.S.</td>

<td>This item is not eligible for domestic shipping.</td>

<td>This item is also available for shipping to select countries outside the U.S.</td>

</tr>

<tr>

<th>International Shipping</th>

<td>This item is not eligible for international shipping.</td>

<td>This item is not eligible for international shipping.</td>

<td>This item is not eligible for international shipping.</td>

<td>This item is not eligible for international shipping.</td>

</tr>

</tbody>

</table>

</div>

The table is used to contain tabular data. Tables have rows (tr), and data cells. The data cells are either headings (th) or simply data (td). Tables can be further broken down to a head section (thead), a body section (tbody), and a foot section (tfoot).

## Reviews

Product reviews are another useful, expected tool for buyers to make decisions about purchasing. Reviews need to be listed out, each containing a Name, a Rating, and Comments. A user might also want to leave a review of this product, so we’ll want to add a form as well. As before, lets create a designated space for this information:

<div id=”productspecs”>

…

</div>

<hr />

<div id=”reviews”>

</div>

### Form

<div id=”reviews”>

<h2>Reviews</h2>

<form method="post" id="reviewform">

<p>

<label for="reviewer">Name</label>

<input type="text" id="reviewer" name="reviewer" placeholder="Name" />

</p>

<p>

<label for="rating">Rating</label>

<select id="rating" name="rating">

<option value="10">10 - Outstanding.</option>

<option value="9">9 - Excellent.</option>

<option value="8">8 - Very good.</option>

<option value="7">7 - Good.</option>

<option value="6">6 - Fair.</option>

<option value="5">5 - Average.</option>

<option value="4">4 - Below average.</option>

<option value="3">3 - Poor.</option>

<option value="2">2 - Very poor.</option>

<option value="1">1 - Defies description.</option>

</select>

</p>

<p>

<label for="comments">Comments</label>

<textarea name="comments" id="comments" cols="50" rows="5"></textarea>

</p>

<p><input type="submit" id="postReview" value="Post Review" class="bigButton" /></p>

</form>

</div>

### Reviews

Below the form we need to list out each of the reviews that have already been written about the product by other users.

<div id=”reviews”>

<form>

…

</form>

<ul id="reviews">

<li class="review">

<h3>Harry Henderson</h3>

<p class="rating">8 - Very good.</p>

<p class="comment">They weren't kidding when they said these were the shirtiest shirts! I had my doubts when I placed my order--I've been disappointed with the shirtiness of shirts before--but these guys really came through.</p>

</li>

<li class="review">

<h3>Bruce Banner</h3>

<p class="rating">1 - Defies description.</p>

<p class="comment">I haven't received my shirts yet, so as far as the shirtiness of the shirts goes I'd have to say they aren't very shirty at all. There is no shirt! I just want my shirty shirt and I'm angry that I don't have it yet. You won't like me when I'm angry...</p>

</li>

</ul>

</div>

# CSS

Now that you’ve completed your page you’ll notice that it isn’t as nice looking as most of the other sites on the internet. CSS is how you fix that. So let’s create our style sheet and link it to our page.

Create a new text file and save it as “style.css” in the /style directory of your working folder.

Next, switch back to “index.html” and in the <head> section at the top of the file, modify it to link the stylesheet like this:

<head>

…

<title>ACES' Learn to Code Store</title>

<link rel="stylesheet" type="text/css" href="style/style.css" />

</head>

## Basic Styling

CSS is used by targeting specific pieces of HTML, whether by attribute or by tag, and applying rules to them that determine how they will look. CSS rules can be very loose, or very specific, as defined by you. Rules sets all follow a particular format, i.e. a target followed by a pair of curly brackets which contained a set of rules.

target {

rule-one: definition;

rule-two: definition;

rule-three: definition;

}

Comma-delimited targets (e.g. h1, h2, h3 { }) select each of the nodes and apply the same rules to all of them.

Targets with spaces or special symbols between them (e.g. table tbody td, span > p) apply to the rightmost element in the list. So, our example table tbody td is only selecting td tags that are contained inside of table tags which are in turn contained inside of table tags. So the td tag contained inside of the following HTML would be ignored:

<table><thead><td></td></thead><table>

First, we’ll add some styles to specific tags on the page, regardless of their attributes. Inside your “style.css” file add the following lines:

body {

margin: 0;

padding: 0;

background-color: #CFE1D0;

color: #2D3037;

font-family: Calibri, Candara, Segoe, 'Segoe UI', Optima, Arial, sans-serif;

}

hr {

clear: both;

margin: 1em 0;

}

h1, h2, h3 {

color: #60AC90;

}

table {

border: 0;

border-collapse: collapse;

}

table td, table th {

padding: 0.3em;

vertical-align: top;

}

table tbody tr.striped {

background-color: #CFE1D0;

}

Let’s take table for our example. The table rules in our stylesheet will apply to every <table> tag that is made in any document that links to our stylesheet. If we added the rule background-color: blue; to the table’s rules set, what would you expect to happen?

## Classes and IDs

Remember the “id” and “class” attributes we used in the HTML? Now, we can affect those nodes by targeting them inside CSS. Ids are targeted using the “#” symbol, while classes are targeted using the “.” symbol.

Add the following lines to your “style.css” file:

#sitewrapper {

width: 930px;

margin: 15px auto 0;

padding: 15px;

background-color: #fff;

overflow: hidden;

}

#wrappershadow {

width: 940px;

margin: 0 auto 35px;

background-color: #999;

opacity: 0.5;

padding: 3px 0;

}

#header {

border-top: 1px solid #738A84;

border-bottom: 20px solid #738A84;

padding: 15px;

}

#product {

margin: 0 0 1em;

overflow: hidden;

}

#product .onethird {

text-align: center;

}

#product .onethird img {

margin-top: 2em;

}

#footer {

border-top: 15px solid #738A84;

padding: 10px 0;

}

.onethird {

width: 290px;

padding: 0 10px;

float: left;

}

.twothird {

width: 600px;

padding: 0 10px;

float: left;

}

#reviews {

list-style: none;

margin: 1em 0;

padding: 0;

}

.review {

border-bottom: 1px solid #738A84;

}

#productspecs table {

width: 100%;

}

#productspecs table th {

text-align: left;

}

#content form {

background-color: #efefef;

border: 1px solid #738A84;

border-radius: 10px;

padding: 0 1em;

}

#content form p > label {

display: inline-block;

width: 75px;

}

#content form p > textarea {

vertical-align: top;

}

.bigButton {

font-size: 16px;

font-weight: bold;

padding: 0.5em 1em;

background-color: #53CA9F;

border: 1px solid #60AC90;

border-radius: 10px;

color: #fff;

cursor: pointer;

}

.bigButton:hover {

background-color: #738A84;

border: 1px solid #2D3037;

}

# JavaScript

**Index.html**

1. Download the starter folder
2. Open the index.html file in a text editor
3. Start by creating a header with the <h1> </h1> tags
   1. In between the tags enter a title for the page
4. Add an image by typing the following
   1. <img src="images/shirts.jpg" alt="Smiley face" height="175" width="175">
   2. Change the height and with to different values to display the image as you want
   3. Open the index.html file in an internet browser to check that it works
5. Add a form by typing <form id="addToCart"> </form>
   1. All of the code to make a order form will go in between these tags
6. Add the input for # of shirts they want to order
   1. Number of shirts:<input type="text" id="num"> </input> <br>
   2. <br> is a line break which puts everything that comes after it on a new line
7. Add a dropdown for the shirt colors
   1. Color <select id="color"> <option value="orange">Orange</option> <option value="blue">Blue</option><option value="green">Green</option><option value="yellow">Yellow</option></select> <br>
   2. The select tag represents the entire dropdown and the option tags are the different items in the dropdown list.
8. Add radio buttons for size
   1. <input type="radio" name="size" id="size1" value="S"> Small<br><input type="radio" name="size" id="size2" value="M"> Medium<br><input type="radio" name="size" id="size3" value="L"> Large<br><input type="radio" name="size" id="size4" value="XL"> X-Large<br>
   2. All radio buttons with the same name are grouped together so that the user can only select one of them
9. Lastly, add a order button
   1. <input type="button" value="Order" id="submit" onclick="order();"> </input>
   2. the onclick=”order();” will call a function we are about to write that will display

**Javascript.js**

1. Open the JS folder and inside open the javascript.js
2. Create a function for order()
   1. Function order(){ }
   2. All the code for this function will go inside the curly braces
   3. This function will get called when the user clicks the order button
3. First get the number of shirts the user ordered
   1. var num = document.getElementById('num').value;
   2. var num creates a variable called num which we will store data in
   3. document.getElementById('num') will find the HTML object that has an ID = ‘num’
4. Get the value for radio buttons
   1. var size;

if (document.getElementById('size1').checked) {

size = document.getElementById('size1').value;}

else if (document.getElementById('size2').checked) {

size = document.getElementById('size2').value;}

else if (document.getElementById('size3').checked) {

size = document.getElementById('size3').value;}

else if (document.getElementById('size4').checked) {

* 1. size = document.getElementById('size4').value;}
  2. This code checks each radio button to see if it checked and if it is get the value associated with that button.

1. Get the value of the drowpdown menu
   1. var x = document.getElementById('color').selectedIndex;
      1. This stores the option tag that is selected
   2. var color = document.getElementsByTagName("option")[x].value
      1. This gets the value of that selected option
2. Alert the choices to the user
   1. alert(num + " " + size +" " + color);
3. Save and check that everything works!

**CSS**

**Extra Credit**

**Maybe convert to JQUERY instead of javascript**

Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers that are multiples of both three and five print "FizzBuzz".