Lab Work Week 12

If the content of this lab is causing issues, please have a look at the video solution on webcourses!!
https://youtu.be/NxTLDd5sPJQ

Part 1: Create a Form

To do: Create a simple form (see Form lab)

- Name
- Age
- Email

And a submit button

Validating a blank field

The name field can't be blank

Validating a number

Age must be a number
For this example we will validate the inputs to make sure the user inputs a numeric value.

Step 1: Create a new project with appropriate folders similar to Example 1

Step 2: Add a form and a input for a persons age

```
<form name="myForm" onsubmit="return (validate());" method="post">

    <label for="firstname">Age: </label>
    <input type="text" id="age">

    <input type="submit" value="Click Me!!">
</form>
```

Step 3: Create the JavaScript function to check the persons age

```
// create a function
function validate() {
    var isValid = true;

    // The input we get from the form is a string
    // To try convert this to a number, multiply it by 1
    var num = document.getElementById("age").value * 1;

    if(isNaN(num)) {
        // this is not a number, tell the user!!
        alert('This is not a number!!');
        return false;
    }

    // passed all validation checks, return true
    return isValid;
}
```

isNaN is a function that will tell us if an input is a number or not.
If the value is not a number it will return false.

Validate Email Address

For this example we will validate the inputs to make sure the user inputs a valid email address.

Step 1: Create a new project with appropriate folders similar to Example 1

Step 2: Add a form and a input for a persons email address

Step 3: Create a function to validate the email address

This function is using a regular expression to check the email address. This is beyond the scope of the JavaScript we are covering, lets just use the function and trust that it performs the task for us!!
Copy and paste the following function to the JavaScript for your page.

```
function validateEmail(email) {
    var re = /^(([^<>()\[\]\\.,;:\s@"]+(\.[^<>()\[\]\\.,;:\s@"]+)*)|(".+"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/;
    return re.test(email);
}
```

Step 4: Create the JavaScript validate function for the form

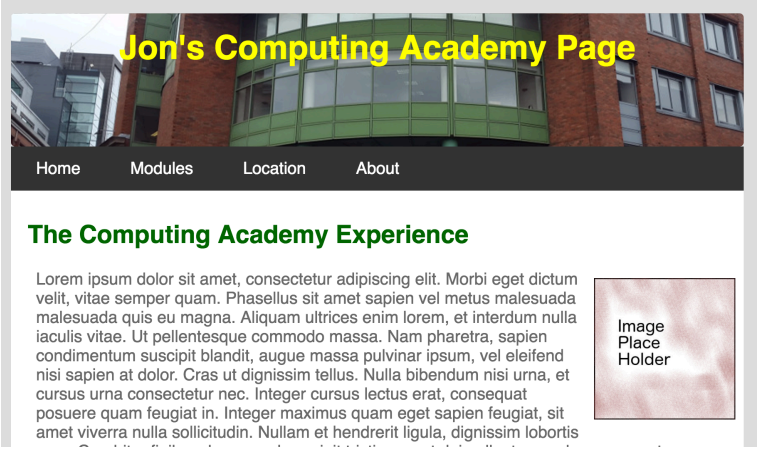
```
// create a function
function validate() {
    var isValid = true;

    // Get the email address from the form
    var email = document.getElementById("email").value;

    if(validateEmail(email)==false) {
        // this is not a valid email address, tell the user!!
        alert('This is not a valid email address!!');
        return false;
    }

    // passed all validation checks, return true
    return isValid;
}

// you can cut and paste this from the lab doc
function validateEmail(email) {
    var re = /^(([^<>()\[\]\\.,;:\s@"]+(\.[^<>()\[\]\\.,;:\s@"]+)*)|(".+"))@((\[[0-9]{1,3}\.[
        0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/;
    return re.test(email);
}
```

Step 5: Test if it is working

Part 2: Responsive Menus

Media Queries can be used to create responsive menus.
If the page is viewed on a phone, the menu bar should change to a drop-down list.

Got to webcourses and get the starter site (week 12 labs).
It should look as follows:



The menu is displaying horizontal.

When the screen size goes very small, it will be difficult for the users of the site to operate the menu.

We will try use media queries and CSS to make a menu.

Step 1:
Add the two following media queries to the bottom of the CSS file:

```
/* Query 1 */
@media (max-width:1200px) and (min-width:601px) {

}

/* Query 2 */
@media (max-width:600px) and (min-width:300px) {


}
```

We will add CSS to both media queries to get the menu to display differently depending on the size of the screen. The two queries are for screen sizes between 300 and 600, and for screens 601 to 1200.

Step 2: Update the 1200 media query to look as follows:

```
/* Query 1 */
@media (max-width:1200px) and (min-width:601px) {
        nav ul {
          list-style-type: none;
          margin: 0;
          padding: 0;
          overflow: hidden;
          background-color: #333;
        }

        nav li {
          float: left;
        }

        nav li a {
          display: block;
          color: white;
          text-align: center;
          padding: 15px 30px;
          text-decoration: none;
        }

        nav li a:hover {
          background-color: #111;
        }
}
```

Hint: The content of the media query is the same as the original CSS for the nav.

Step 3: Update the HTML markup
We will add a hyperlink to the nav section, this will appear when the menu goes small.
Menu

```
<nav>
  <a id="menu">Menu</a>
  <ul class="menulinks">
    <li><a class="active" href="index.html">Home</a></li>
    <li><a href="modules.html">Modules</a></li>
    <li><a href="location.html">Location</a></li>
    <li><a href="about.html">About</a></li>
  </ul>
</nav>
```

Give the UL tag a class name of menulinks

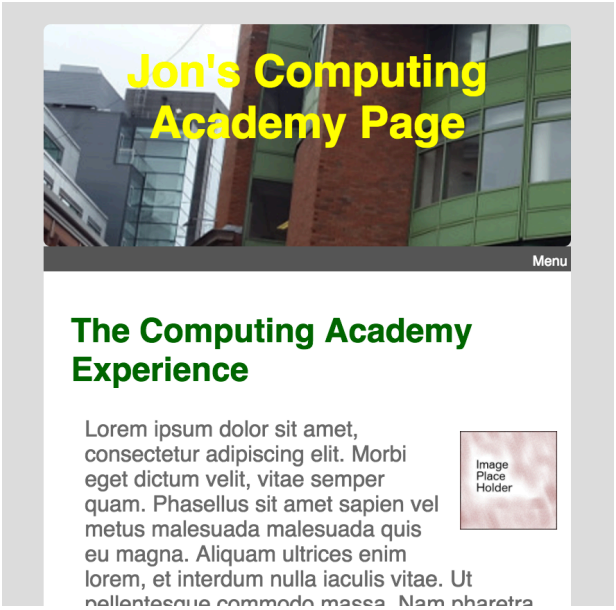
Step 4: Delete the following from the CSS file:

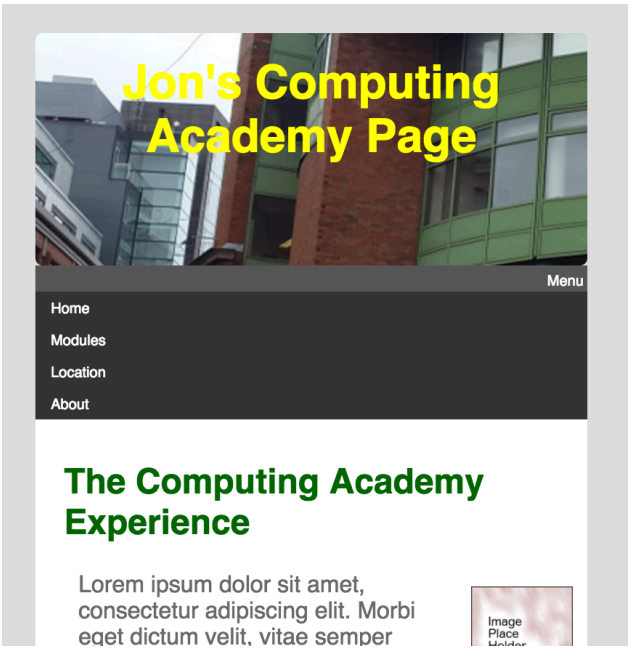
```
nav ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
}

nav li {
  float: left;
}

nav li a {
  display: block;
  color: white;
  text-align: center;
  padding: 15px 30px;
  text-decoration: none;
}

nav li a:hover {
  background-color: #111;
}
```

Step 5: Add a style rule for the 300 to 600 window.

```
@media (max-width:600px) and (min-width:300px) {
  #menu {
    visibility: visible;
  }

  nav a {
    padding: 3px;
    font-size: 10px;
  }

  nav {
    text-align: right;
    min-height: 15px;
    background-color: #555555;
    color: white;
  }

  nav ul {
      text-align: left;
      margin: 0;
      padding: 0;
      width:100%;
      list-style-type: none;
      display: none;
  }

  nav:hover .menulinks {
    display: block;
    font-size: 10px;
  }

  nav ul li a {
      text-decoration: none;
      color: white;
      padding: 5px 11px;
      background-color: #333333;
      display:block;
  }

  nav ul li a:visited {
      color: white;
  }

  nav ul li a:hover {
      color: white;
      background-color: black;
  }

}
```

Step 6: Save and test





Move the page around to see the menu style change.