

Lab 10 – Forms, HTML5 and JavaScript

Example 1 – Create a simple form

Step 1 – Download the Form1 zip file from Webcoueses

Extract its content.

Open index.html for editing

Step 2:

Add the following form to the <div id=content> Form goes here </div>

```
<h3>Registration page</h3>
```

```
<form action="server-script-url" name="regForm">
```

```
</form>
```

Step 3: Add the form controls.

- Firstname
- Lastname
- Address
- Set Password
- Confirm Password
- Class
- Gender (Male / Female)
- Travel (Walk / Bus / Car / Train / Bicycle / Helicopter / Jet)

This will be described in the following steps

Step 4: Add Inputs for the Firstname and Lastname

```
<label>Firstname</label><input type="text" name="firstname"><br>  
<label>Lastname</label><input type="text" name="lastname"><br>
```

Step 5: Add the address field

```
<label>Address</label><input type="text" name="address"><br>
```

Step 6: Add the password fields

```
<label>Password</label><input type="password" name="password1"><br>  
<label>Password</label><input type="password" name="password2"><br>
```

Step 7: Add the class drop-down box

```
<label>Class</label><select>
  <option value="dt228">DT228</option>
  <option value="dt282">DT282</option>
  <option value="dt211">DT211</option>
  <option value="dt265">Dt265</option>
</select><br>
```

Step 8: Add the radio buttons for male/female

```
<label>Gender</label><br>
<input type="radio" name="gender" value="male" checked> Male<br>
<input type="radio" name="gender" value="female"> Female<br>
```

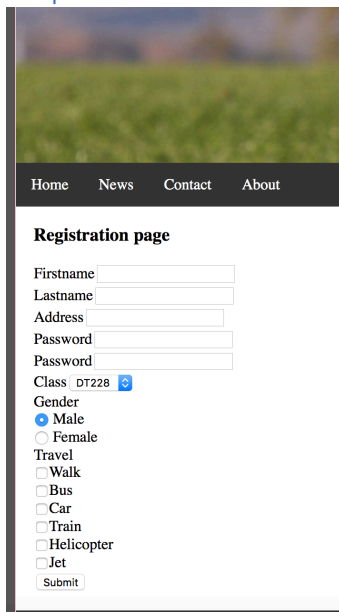
Step 9: Add the checkbox for the travel options

```
<label>Travel</label><br>
<input type="checkbox" name="vehicle" value="Walk">Walk<br>
<input type="checkbox" name="vehicle" value="Bus">Bus<br>
<input type="checkbox" name="vehicle" value="Car">Car<br>
<input type="checkbox" name="vehicle" value="Car">Train<br>
<input type="checkbox" name="vehicle" value="Helicopter">Helicopter<br>
<input type="checkbox" name="vehicle" value="Jet">Jet<br>
```

Step 10: Add the submit button

```
<button type="submit">Submit</button>
```

Step 11: Save the file and open in a browser. It should now look as follows:



Home News Contact About

Registration page

Firstname

Lastname

Address

Password

Password

Class DT228

Gender

☒ Male

☐ Female

Travel

☐ Walk

☐ Bus

☐ Car

☐ Train

☐ Helicopter

☐ Jet

What is wrong with the form??

Its styling just doesn't look good, we will use CSS to try make the form look better.

Example 2: Styling the form.

We will use CSS to try make the form look better.

Step 1: Update the HTML format for the form controls:

Eg. For the firstname:

```
<label>
    <span>Firstname:</span>
    <input class="generalInput" type="text" name="firstname">
</label>
```

Follow this structure for all inputs in the form.

The completed form should look as follows:

```
<form action="server-script-url" name="regForm">
    <label>
        <span>Firstname:</span>
        <input class="generalInput" type="text" name="firstname">
    </label>

    <label>
        <span>Lastname:</span>
        <input class="generalInput" type="text" name="lastname">
    </label>

    <label>
        <span>Address:</span>
        <input class="generalInput" type="text" name="address">
    </label>

    <label>
        <span>Password:</span>
        <input class="generalInput" type="password" name="password1">
    </label>

    <label>
        <span>Password:</span>
        <input class="generalInput" type="password" name="password2">
    </label>

    <label>
        <span>Class:</span>
        <select class="generalInput">
            <option value="dt228">DT228</option>
            <option value="dt282">DT282</option>
            <option value="dt211">DT211</option>
            <option value="dt265">Dt265</option>
        </select>
    </label>

    <label>
        <span>Gender:</span><br>
        <input class="clickInput" type="radio" name="gender" value="male" checked> Male<br>
        <input class="clickInput" type="radio" name="gender" value="female"> Female<br>
    </label>
    <br>

    <label>
        <span>Travel:</span><br>
        <input class="clickInput" type="checkbox" name="vehicle" value="Walk">Walk<br>
        <input class="clickInput" type="checkbox" name="vehicle" value="Bus">Bus<br>
        <input class="clickInput" type="checkbox" name="vehicle" value="Car">Car<br>
        <input class="clickInput" type="checkbox" name="vehicle" value="Car">Train<br>
        <input class="clickInput" type="checkbox" name="vehicle" value="Helicopter">Helicopter<br>
        <input class="clickInput" type="checkbox" name="vehicle" value="Jet">Jet<br>
    </label>
    <br>
    <button type="submit">Submit</button>
</form>
```

Step 2: Now we can style the updated form elements:

Add the following to your CSS file:

```
89  h3 {
90      color: green;
91      font-size: 30px;
92  }
93
94  span {
95      color: blue;
96      font-size: 20px;
97      font-weight: bold;
98  }
99
100  .generalInput {
101      width: 375px;
102      display: block;
103      border: 1px solid #999;
104      height: 25px;
105      margin-bottom: 20px;
106  }
107
108  .clickInput
109  {
110      border: 1px solid #999;
111      margin-bottom: 7px;
112      margin-right: 10px;
113  }
---
```

We have 4 style rules to style the elements in the form.

Go through the CSS rules and try understand what is happening.

Step 3: Save and reload the page

The form should now look as follows:

Registration page

Firstname:

Lastname:

Address:

Password:

Password:

Class:

Gender:

☒ Male

☐ Female

Travel:

☐ Walk

☐ Bus

☐ Car

☐ Train

☐ Helicopter

☐ Jet

Example 3: HTML5 Validation

We will now add some HTML5 Validation to the form.

First make a backup copy of your html form page, we will be using this later without JavaScript validation!!

Step 1:

The firstname and lastname are required, the user must enter their firstname and lastname. With HTML5 form validation, we can use the **required** keyword.

Update the form inputs for firstname and lastname to add the required keyword:

```
<label>
  <span>Firstname:</span>
  <input class="generalInput" type="text" name="firstname" required>
</label>

<label>
  <span>Lastname:</span>
  <input class="generalInput" type="text" name="lastname" required>
</label>
```

Step 2: Add an email field to the form:

We will add a email field to the form to allow the user enter their email address.

When adding the input to the form, make sure to set the type to email.

```
<label>
  <span>Email:</span>
  <input class="generalInput" type="email" name="email">
</label>
```

Save and reload the page. Enter an incorrect email address, the form should tell you that it is incorrect. Note: we have not set the email field as required, if you enter nothing the form will just submit!!

Step 3: Add a date field to the form

```
<label>
  <span>Date:</span>
  <input class="generalInput" type="date" name="date" required>
</label>
```

Save and reload the page. Try a few different browsers to see if they validate the date field!! It should work in Chrome!!

Example 4: JavaScript Quiz

We will create a simple quiz using JavaScript to demonstrate the basic operation of how JavaScript can manipulate the content of a webpage. The webpage will hold 3 questions. When the user clicks one of the questions buttons, the question will appear on the webpage. The user must provide the correct answer to the question. The user will be told if the answer is correct or not.

Step 1: Create the basic HTML page for the Quiz

Add the following markup to a html page. You can use the starter template for this weeks lab work.

```
<h1>Jono's Super Quiz Page</h1>
<button>Q1</button>
<button>Q2</button>
<button>Q3</button>

<h2>Question:</h2>
<h3 id="question"></h3>

<label>
  <span>Enter your answer:</span>
  <input class="generalInput" type="text" id="answer" required>
</label>

<h3 id="correct"></h3>

<button>Submit</button>
```

Step 2: Add some JavaScript to the page:

Create a file named site.js in the Assets/JavaScript folder.

Add the Script to the page:

```
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Jon's Quiz Page</title>
  <link rel="stylesheet" href="Assets/CSS/site.css">
  <script src="Assets/JavaScript/site.js"></script>
</head>
```

Step 3: Add the questions and Answers to the JS file

```
1  // Create an array to hold the quiz questions
2  var questions = [];
3
4  // Create an array to hold the quiz answers
5  var answers = [];
6
7  //Set the questions and answers
8
9  // First Question
10 questions[0] = "Who is the Flash?";
11 answers[0] = "Barry Allen";
12
13 // Second Question
14 questions[1] = "Where is Superman from?";
15 answers[1] = "Krypton";
16
17 // Third Question
18 questions[2] = "What is Bruce Banners alter ego?";
19 answers[2] = "The Hulk";
20
21 var currentQuestion;
```

Step 4: Display the question on the page

Now that we have the questions stored in an array, we will need to display the question on the page for the user to see!! There are 3 button on the page: Q1, Q2, Q3

When the user clicks the Q1 button, we want question to appear etc...

We need to bind an event to the buttons, when the button is pressed we want something to happen!! The onclick attribute can be used for this. When the buttons are clicked we will call a JavaScript function to display the question on the page.

Add the following function to the JS file:

```
23 function setQuestion(questionNumber) {
24     currentQuestion = questionNumber;
25     document.getElementById("question").innerHTML = questions[questionNumber];
26 }
```

The function setQuestion takes in a questionNumber, this will be used to identify the question to display from the questions array. The variable currentQuestion will be set to the questionNumber that is passed into the function, we will need this later to check the answer. document.getElementById("question").innerHTML is use to access the element on the page, we will get the question from the array and add this text to the element on the page for display.

Update the Question buttons on the page to look as follows:

```
<button onclick=" setQuestion(0); ">Q1</button>
<button onclick=" setQuestion(1); ">Q2</button>
<button onclick=" setQuestion(2); ">Q3</button>
```

Reload the page, click the questions buttons, the quiz questions should be displaying on the page.

Step 5: Get the answer from the user

Next we want to allow the user to enter their answer in the input box and click submit. When submit is pressed we want to check if the answer is correct and then let the user know.

First we will create a JS function that will get the input from the form field when the submit button is pressed. Add the following to the JS file:

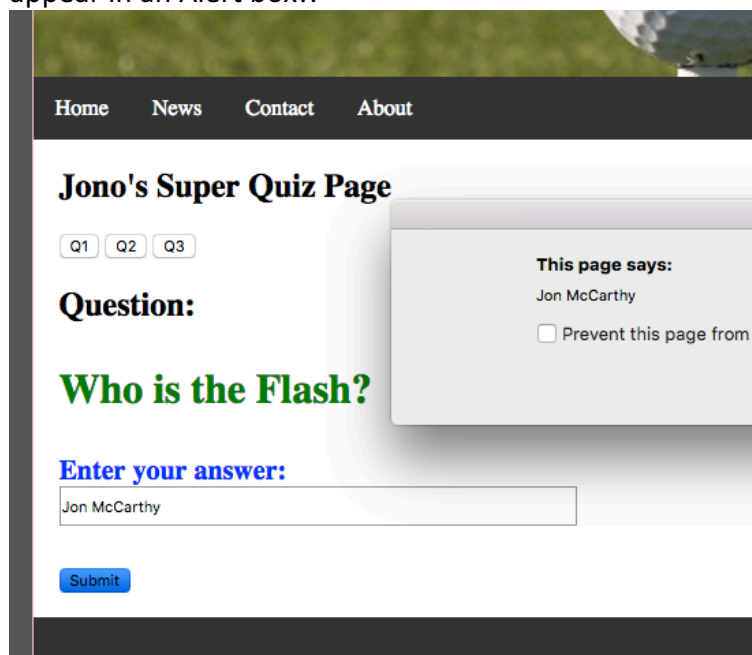
```
function checkAnswer() {
    var tmpAnswer = document.getElementById("answer").value;
    alert(tmpAnswer);
}
```

When this function is called, we will use `document.getElementById("answer")` to access the input element on the page with an id of "answer". The `.value` will get the text the user entered and we will store it in a variable called `tmpAnswer`. Alert will display the users text for us to see!!

Before we see this working we need to update the submit button to call the `checkAnswer` function:

```
<button onclick=" checkAnswer(); ">Submit</button>
```

Reload the page, click a question and then answer the question. The text entered should appear in an Alert box!!



Step 6: Next we need to check the users, see if it is correct and then inform the user.

In step 5 we got the answer and displayed it in an alert box, now we will use this answer to see if it is correct and then inform the user.

Update the checkAnswer function to look as follows:

```
28 function checkAnswer() {  
29     var tmpAnswer = document.getElementById("answer").value;  
30     if (tmpAnswer == answers[currentQuestion]) {  
31         document.getElementById("correct").innerHTML = 'Correct, you are good!!';  
32     } else {  
33         document.getElementById("correct").innerHTML = 'Wrong!!!!!!';  
34     }  
35 }
```

We are using an if/else to check if the answer is correct.

document.getElementById("correct").innerHTML will access the element on the page with an id of correct, we will use this to let the user know if the answer is correct.



Home News Contact About

Jono's Super Quiz Page

Q1 Q2 Q3

Question:

Who is the Flash?

Enter your answer:

Wrong!!!!!!

Submit

Example 5 – JavaScript Form Validation

We can validate the inputs from a form. The process to do this is very similar to the operating of the Quiz game we created in Example 4

Step 1: Create a form

Re-Create the following form (if you saved a version of this form earlier, you can use this!! If not, start typing!!!)

```
<form action="server-script-url" name="regForm">
  <label>
    <span>Firstname:</span>
    <input class="generalInput" type="text" name="firstname">
  </label>

  <label>
    <span>Lastname:</span>
    <input class="generalInput" type="text" name="lastname">
  </label>

  <label>
    <span>Address:</span>
    <input class="generalInput" type="text" name="address">
  </label>

  <label>
    <span>Password:</span>
    <input class="generalInput" type="password" name="password1">
  </label>

  <label>
    <span>Password:</span>
    <input class="generalInput" type="password" name="password2">
  </label>

  <label>
    <span>Class:</span>
    <select class="generalInput">
      <option value="dt228">DT228</option>
      <option value="dt282">DT282</option>
      <option value="dt211">DT211</option>
      <option value="dt265">Dt265</option>
    </select>
  </label>

  <label>
    <span>Gender:</span><br>
    <input class="clickInput" type="radio" name="gender" value="male" checked> Male<br>
    <input class="clickInput" type="radio" name="gender" value="female"> Female<br>
  </label>

  <label>
    <span>Travel:</span><br>
    <input class="clickInput" type="checkbox" name="vehicle" value="Walk">Walk<br>
    <input class="clickInput" type="checkbox" name="vehicle" value="Bus">Bus<br>
    <input class="clickInput" type="checkbox" name="vehicle" value="Car">Car<br>
    <input class="clickInput" type="checkbox" name="vehicle" value="Train">Train<br>
    <input class="clickInput" type="checkbox" name="vehicle" value="Helicopter">Helicopter<br>
    <input class="clickInput" type="checkbox" name="vehicle" value="Jet">Jet<br>
  </label>
  <br>
  <button type="submit">Submit</button>
</form>
```

Step 2: Update the form tag

Update the form tag to add the onsubmit attribute.

When the form is submitted, we will call a JavaScript function to perform some form validation.

```
<form action="server-script-url" name="regForm" onsubmit="return( validate() );">
```

Step 3: Create the validate() function

The function validate will perform all of the JavaScript form validation for us. If the form validation checks are ok, the function will return true, if everything is not ok, return false. If we return false this will stop the form from submitting.

Add the following function to the JS file:

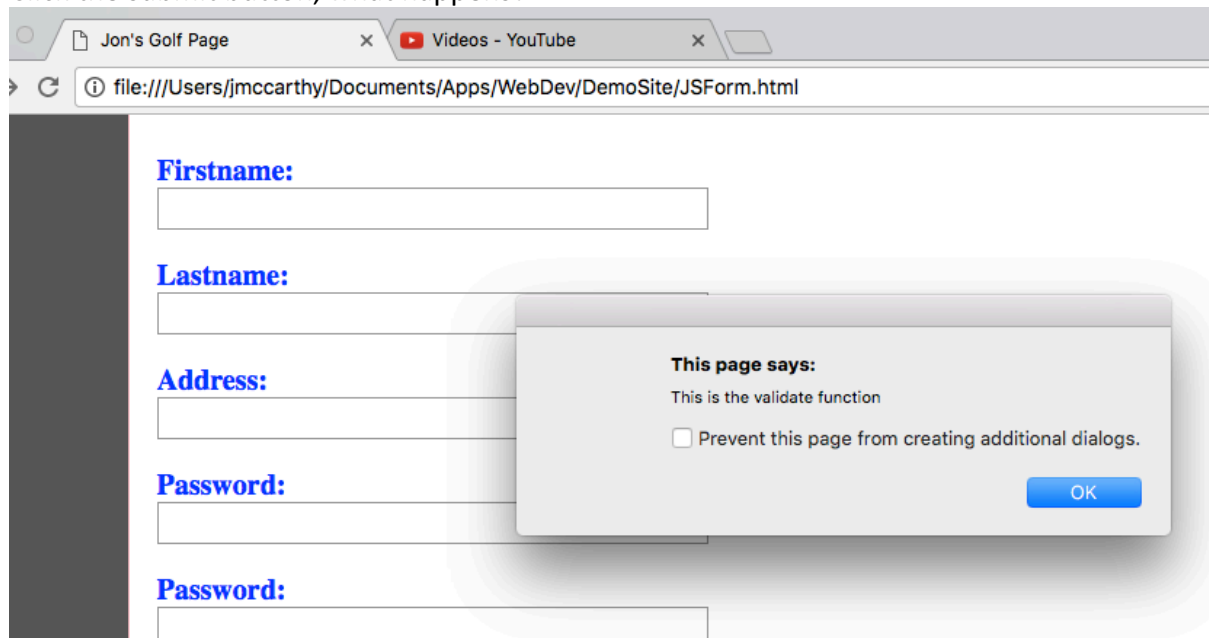
```
38 function validate() {  
39     alert('This is the validate function');  
40     return false;  
41 }
```

Add the JS file to the page, if you haven't already done so:

```
3 <head>  
4 <meta charset="utf-8">  
5 <meta name="viewport" content="width=device-width, initial-scale=1">  
6 <title>Jon's Golf Page</title>  
7 <link rel="stylesheet" href="Assets/CSS/site.css">  
8 <script src="Assets/JavaScript/site.js"></script>  
9 </head>
```

Save all open files and refresh the page.

Click the submit button, what happens?



Step 4: Validate the firstname and lastname

The user must enter a firstname and a lastname.

The checks will be performed in the validate() function using if statements.

If the input is not as expected, stop the form submission and tell the user!!

```
38 function validate() {  
39 |  
40 // check that the firstname field is not blank  
41 if(document.regForm.firstname.value == "") {  
42     alert("Firstname can't be blank!!");  
43     return false;  
44 }  
45  
46 return true;  
47 }  
48
```

Line 40: if the content in the firstname input box is a blank string, stop the form submitting and tell the user with an alert.

Save, refresh and try submit!!

The screenshot shows a web browser window with two tabs: 'Jon's Golf Page' and 'Videos - YouTube'. The address bar shows the file path: 'file:///Users/jmccarthy/Documents/Apps/WebDev/DemoSite/JSForm.html'. The web page contains a form with the following fields:

- Firstname:** (empty text input field)
- Lastname:** (empty text input field)
- Address:** (empty text input field)
- Password:** (empty text input field)
- Password:** (empty text input field)

An alert dialog box is overlaid on the form, displaying the message: 'This page says: Firstname can't be blank!!'. Below the message is a checkbox labeled 'Prevent this page from creating additional dialogs.' and an 'OK' button.

Step 5:

Create another two rules to deal with the Lastname and the address.

Hint: you will need two more if statements.

Step 6:

The alerts don't look particularly good. Let's try get rid of them.

Update the label for the firstname to include a new span to display the error message:

```
<label>
  <span>Firstname:</span><span class="errorMessage" id="firstnameMessage"></span>
  <input class="generalInput" type="text" name="firstname">
</label>
```

Next add a style rule for the error message:

Add this to the end of your style sheet:

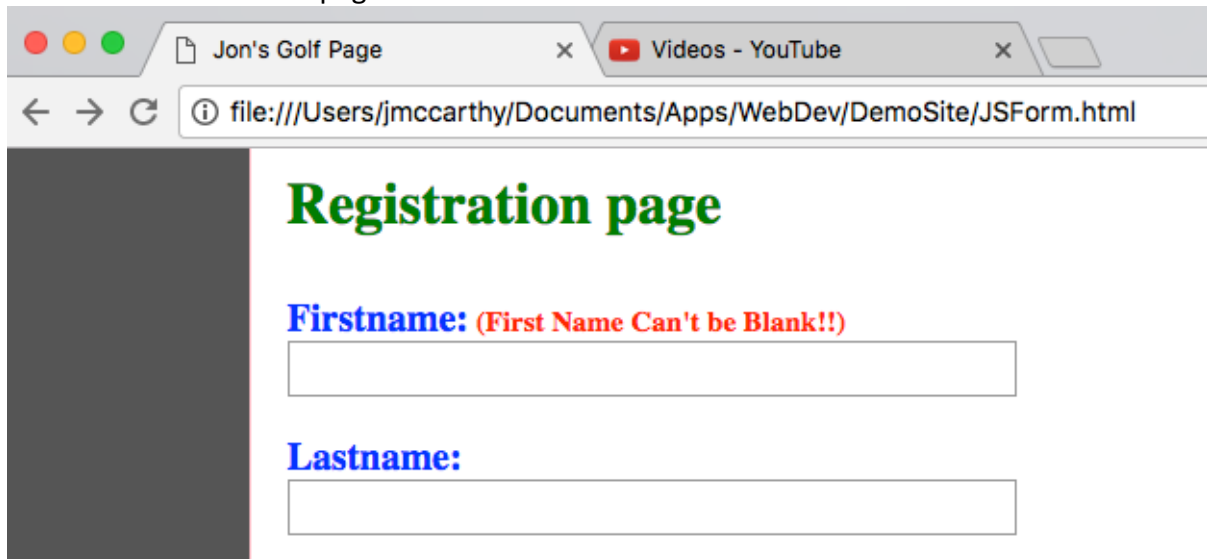
```
.errorMessage {
  color: red;
  font-size: 15px;
  font-weight: bold;
}
```

Update the validate function to display a message in the span with an id of firstnameMessage:

```
function validate() {
  // check that the firstname field is not blank
  if(document.regForm.firstname.value == "") {
    document.getElementById("firstnameMessage").innerHTML = " (First Name Can't be Blank!!)";
    return false;
  }

  return true;
}
```

Save all files and reload page and submit:



The screenshot shows a web browser window with two tabs: "Jon's Golf Page" and "Videos - YouTube". The address bar displays the file path: "file:///Users/jmccarthy/Documents/Apps/WebDev/DemoSite/JSForm.html". The page content features a dark grey sidebar on the left and a main area with the heading "Registration page" in green. Below the heading, the "Firstname:" label is in blue, followed by a red error message "(First Name Can't be Blank!!)". A text input field is positioned below the error message. The "Lastname:" label is also in blue, with an empty text input field below it.