

CS 2033

Multimedia and Communications

Lab 04: Learning how to create a website with HTML and CSS and add a web form.

- Website Development -

REMEMBER TO BRING YOUR MEMORY STICK TO EVERY LAB!

INTRODUCTION TO HTML

HTML (HyperText Markup Language) is the standard language for website development. Most websites are made at least partially, if not entirely, with HTML. Tim Berners-Lee was the main pioneer behind HTML and it was he that created the first website using HTML and the first web browser to display the site around 1989-1991. Since then, there have been numerous additions and revisions to HTML that have helped website developers embed multimedia and other elements more easily into websites.

TAGS

HTML is a tag-based language and the tags are surrounded by triangular brackets < and >. Many tags must be closed with a tag which is similar to the opening tag but with a slash / before the tag element name. For example, <p>This is a paragraph.</p>

Some tags are self-contained and do not need a closing tag at the end. These are often called void elements or stand-alone elements. For example, . Notice how this void element img has a slash before the >. This indicates that it is self-closing. Browsers may not require that slash but it is best practice to include it to be safe.

Most tags can be nested within one another with no limit on the depth of the nesting. Think of Russian/Matryoshka dolls fitting inside one another as a comparison.



Closing tags must be done in the reverse order as they were added, like a mirror image. Examine the following example of the order of closing tags.

<p>This is a paragraph</p>	GOOD	p b /b /p
<p>This is a paragraph</p>	BAD	p b /p /b
<p>This is a paragraph</p>	GOOD	p b /b /p
<p>This is a paragraph</p>	BAD	p b /p /b

WEBPAGE SHELL

There are a few main tags that should be used in every webpage to create its shell, before adding any content or additional tags.

- Doctype (this is one of the few unusual tags that doesn't follow the standard tag structure)
- HTML root tag
 - Head
 - Body

Basic webpage shell! Keep this handy to reference any time you make a new page.

```
<!doctype html>

<html>
  <head>
</head>

  <body>

</body>
</html>
```

That's it! That's the shell of a webpage. Notice that I left empty lines between some of the tags and that I indented the nested tags (head and body). These do not affect the rendering of HTML code so you can leave as little or as much space as you want between lines and for line indents without changing the page at all. It's easier to follow the flow of the page when nested tags are indented so it's suggested that you follow this style but again it is not required and does not change the actual display of the webpage in a browser.

Example code snippet with indents:

```
<div>
  <div>
    <h1>Welcome to CS2033.</h1>
    <p>This is my favourite course at Western!</p>
  </div>
  <div>
    <p>I love creating websites.</p>
    <p>Starting to get the hang of HTML.</p>
  </div>
</div>
```

Example code snippet without indents:

```
<div>
<div>
<h1>Welcome to CS2033.</h1>
<p>This is my favourite course at Western!</p>
</div>
<div>
<p>I love creating websites.</p>
<p>Starting to get the hang of HTML.</p>
</div>
</div>
```

Look at the two example code snippets – one with indents and one without. Everything else is identical. The first one, with indents, is much easier to read and follow. It helps keep the levels organized so that it's easy to see how each of the elements is nested in the others.

ATTRIBUTES

Most tags use attributes to provide additional information about that element. Some attributes are required while others are optional, and certain attributes are different from one tag type to another. Attributes are added within an opening tag and usually assigned a value with an equal sign. Note that text-based attribute values need to be surrounded by quotation marks.

Examples:

<code></code>	ATTRIBUTE: src
<code><div id="banner" class="box"></div></code>	ATTRIBUTES: id and class
<code>Go home</code>	ATTRIBUTE: href

Notice that attributes are added in the opening tag but do **not** have to be mentioned in the closing tag. The `img` tag requires the `src` attribute to provide the source image filepath. The `id` attribute as shown in the second example for `div` is an optional attribute. It will be described more later on. The `a` (anchor/link) tag typically uses the `href` (hyperlink reference) attribute to provide the file or bookmark it points to.

EXERCISE 1: CREATE YOUR FIRST HTML WEBSITE

In CS1033 you used a WYSIWYG program (Kompozer or Dreamweaver) to create websites. Now you will make one with HTML only. Don't worry, you can do it!

We're going to create a simple site for a fictional Western-themed band called 44 Mustangs.

What you'll learn in this exercise:

- How to create an HTML file in a plain text editor
- Adding common text-based tags
- Setting attributes in HTML tags
- Adding images with relative paths
- The difference between `` and ``

1. Navigate to your USB drive folder (F:) and into the **cs2033** subfolder (should have been created in the last lab).
2. Create a folder called **lab04** inside **cs2033**.
3. Within **cs2033**, create a subfolder called **images**.
4. Open the link: <http://www.csd.uwo.ca/~bsarlo/cs2033b/labs/Lab04/images/> and download the images: **guitar.jpg** and **mustangs.png**. Copy them into **cs2033 > lab04 > images**.
5. Open Brackets. For assignments on your own computers, you may use a different plain text editor if you choose. Never use Microsoft Word for this.
6. In Brackets, click File > New to create a new file.
7. Type in the basic webpage shell as provided above. Make sure everything is typed in exactly as it is given (empty lines and indents don't really matter though).
8. Within the head section, set the title (which appears in the browser tab) with the following line of code. Notice how they will automatically add the closing tag (`</title>`) when you

finish typing the opening tag. They are helping save you time by closing tags. Code:
<title>44 Mustangs – Home</title>

9. Your code should look like this so far:

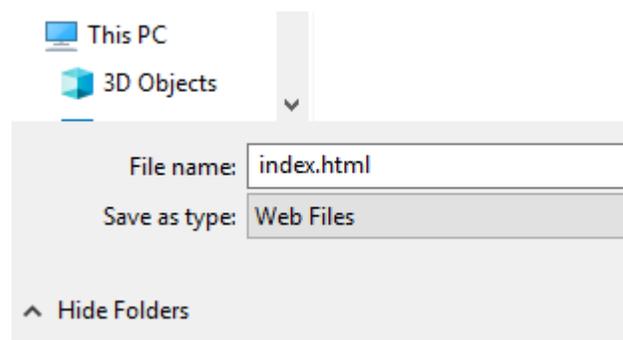
```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>44 Mustangs – Home</title>
5  </head>
6
7
8  <body>
9
10
11 </body>
12
13 </html>
```

10. Now click into the body section of the code so we can begin adding content to the site.

11. Create a large header for the band name. Type <h1> and then between that and the automatic closing </h1> tag, add the band name: 44 Mustangs

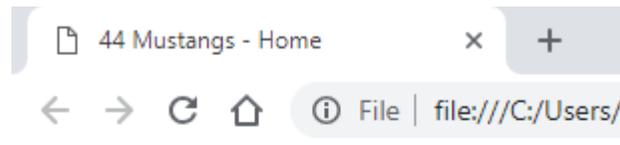
```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>44 Mustangs – Home</title>
5  </head>
6
7
8  <body>
9      <h1>44 Mustangs</h1>
10
11 </body>
12
13 </html>
```

12. Let's take a look at the start of our site. Click File > Save As. Then navigate to the folder **lab04** you created in **cs2033** (but not inside the images subfolder!). Before pushing Save, change the filename to index.html since this will be the homepage of the site. (NOTE: Only use Save As the first time saving a file; later just click Save or use Ctrl+S to save it).



13. Open the **lab04** folder and you should see your index.html file in there now. Right-click on it and select Open With > Google Chrome. This will open the webpage in Chrome. It looks

plain and empty now but the two things we added should be displayed: the title 44 Mustangs – Home in the tab, and the header 44 Mustangs at the top of the page.



44 Mustangs

14. Now go back into Brackets so we can add more code. It's a good idea to keep your page open in Chrome while working so that you can come back frequently and refresh it to see your updates without having to re-open it every time.
15. After the h1 header line, add an h2 header (slightly smaller than h1) which should say "The official page of the Western-themed pop/rock band"
16. Save the file and refresh or re-open the page in Chrome.
17. Experiment with the header sizes by changing h2 to h3, then h4, then h5, and lastly h6. Save and reload the page in the browser after each header change to see the size different. (NOTE: remember the change both the opening and closing tag together). After looking at these differences, revert to h2 to make it big again.
18. Now let's add an image to the page. Under the h2 line, add the following code:
``
19. Save it and reload the page in the browser. Does the picture show up? What do you see? Why do you think this is occurring? Now can you see why we set the alt text on images?
20. The path we used in the img src attribute was "guitar.jpg" which is the correct filename. However, our picture is stored in the images folder so we need to specify that in the filepath. To make working filepaths, everything should be relative which means indicating how to get to the specified file (guitar.jpg) from the current location (lab04). Therefore we need the path to be "images/guitar.jpg" so that it has to look in images for the image.
21. Once you fix the image source path, save it and reload and then the picture should appear.
22. The webpage should also include a list of the band members. Create an ordered list with `` or an unordered list with ``. List items are then added within a list tag with the `` tags. Do you think the list of band members should be an ordered or unordered list?
23. Get started with the following code, which you can enter below the img tag.
`
 Brent Thomas
 Myra "Lungs" Campos
`
24. Save and reload the page. Then add the other band members into that list: Val Smith, Omar Patel, and Tony "Drum Machine" Rudd.
25. These next two pictures should be very similar to your code and browser display now.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>44 Mustangs - Home</title>
5   </head>
6
7
8   <body>
9     <h1>44 Mustangs</h1>
10    <h2>The official page of the Western-themed pop/rock band</h2>
11
12    
13    <ul>
14      <li>Brent Thomas</li>
15      <li>Myra "Lungs" Campos</li>
16      <li>Val Smith</li>
17      <li>Omar Patel</li>
18      <li>Tony "Drum Machine" Rudd</li>
19    </ul>
20  </body>
21
22 </html>
--
```

44 Mustangs

The official page of the Western-themed pop/rock band



- Brent Thomas
- Myra "Lungs" Campos
- Val Smith
- Omar Patel
- Tony "Drum Machine" Rudd

EXERCISE 2: ADDING CSS TO YOUR WEBSITE

You have a good start on this website but as you probably noticed already, it's a little boring. That's because it has no CSS to style it yet, so we'll add styles to it!

What you'll learn in this exercise:

- Adding internal CSS in a webpage
- Creating style rule-sets for tag elements
- Determining how to apply a rule to one element and not another of the same type
- Assigning an ID to an HTML element
- Creating style rule-sets for a specific ID
- Embedding Google Fonts

1. In index.html in Brackets, put your cursor at the very end of the <title> line and hit Enter.
2. Now write the following code:
`<style type="text/css">`
3. Again you will see that they added the closing </style> tag beside it. Click in between the two and hit Enter a bunch of times to create space within the style tag.
4. We should start by setting the page's background colour to brown. Click into the line immediately under the open style tag and create an empty rule-set for the body element:
`body {`

`}`

This rule-set applies to the body tag which means any styles we add within this set (inside these curly brackets) will be applied to the main page.

5. We can set the background colour to brown simply by adding this line within the rule-set:
`background-color: brown;`

NOTE: you must use American spelling "color" instead of "colour" in CSS

6. Save and reload your page in the browser. The colour is a bit too red for what we wanted.
7. Back in Brackets, change "brown" to the hexadecimal code #6F4137 (include the pound sign # with the code) and then save and reload the page. That brown looks better.
8. Below the body rule-set, create a rule-set for the h1 element. Within the h1 rule-set, change the font colour to white (color: white) and the font size to 40px (font-size: 40px).

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>44 Mustangs - Home</title>
5  <style type="text/css">
6      body {
7          background-color: #6F4137;
8      }
9  h1 {
10     color:white;
11     font-size:40px;
12 }
13
14 </style>
15 </head>
```

9. Save and reload the page.

10. The font is a little boring as well. There's a cool font called Acme that the band found on Google Fonts that would look better for their site. Remember that Google Fonts can be linked/embedded in a site so that they will be loaded from any device. Google Fonts give both the HTML to link to the font and the CSS style rule for applying the font.
11. The first step is to link the font in our HTML. The following line must be copied anywhere into the head section of our HTML (not within the style tag area though):
<link href="https://fonts.googleapis.com/css?family=Acme" rel="stylesheet">
12. The other step is to add the style rule that indicates where this font should be applied. Let's apply it to the entire site, so it can go within the body style rule-set:
font-family: 'Acme', sans-serif;
13. Save and reload the page.



14. Now create a rule-set for the img element and set the width to 800px.
15. Save and reload the page. The guitar image should be larger now.
16. The band also wants to add the Western Mustangs logo near the bottom of the site to show their #purplepride. In the HTML, put your cursor after the closing list tag and add an image the same way you added the guitar image. In this case, the src is "images/mustangs.png" and the alt and title can both be set to "Western Mustangs".
17. Save and reload the page. What is wrong with the image?
18. Remember in step #14 you created a rule-set for the img element and set the image width to 800px? Well this applies to all img elements so the Western Mustangs logo is being set to 800px too which is much too large. If we remove that rule-set, then the guitar image won't be the size we want though, so how can we apply the width style to the guitar image but not the Western Mustangs logo?

19. We can use classes or IDs to accomplish this. These two options are similar but classes are better suited if there might be multiple elements that the style set should apply to, while the ID option is for a singular instance only.
20. In this case we will use an ID, although either one would be fine. In the HTML, find the line where you added the guitar image. Within that tag, add the following attribute and value: `id="main-pic"`

```
30 <body>
31   <h1>44 Mustangs</h1>
32   <h2>The official page of the Western-themed pop/rock band</h2>
33
34   
35 <ul>
36   <li>Brent Thomas</li>
37   <li>Myra "Lungs" Campos</li>
38   <li>Val Smith</li>
39   <li>Omar Patel</li>
40   <li>Tony "Drum Machine" Rudd</li>
41 </ul>
42
43   
44
45 </body>
```

21. Since the guitar image now has an ID attached to it, we can apply a style rule-set to that individual ID instead of applying it to all the `img` elements at once.
22. In the CSS, remove the line that says `img {` which is currently opening the `img` rule-set.
23. In place of that line, type: `#main-pic {`
24. This tells it that this rule-set, which is currently setting the width to 800px, to apply to the ID "main-pic".
25. Save and reload the page. Now you should see the Mustangs logo at its proper, small size and the guitar picture maintains its larger size.
26. It would be nice to have the content shifted to the middle of the page. To do this, we will put all the content inside a `div` and then center that `div`. Create a `div` immediately after the opening `body` tag and give the `div` the ID "content" (`<div id="content">`). The automatically generated closing `div` tag appears right there but we need to move it down further so that the content is enclosed in the `div`. So remove that automatically made closing `div` tag and then click immediately before the closing `body` tag and add a closing `div` tag there (you can either cut and paste it or delete and re-type it).

```
30 <body>
31
32 <div id="content">
33
34   <h1>44 Mustangs</h1>
35   <h2>The official page of the Western-themed pop/rock band</h2>
36
37   
38 <ul>
39   <li>Brent Thomas</li>
40   <li>Myra "Lungs" Campos</li>
41   <li>Val Smith</li>
42   <li>Omar Patel</li>
43   <li>Tony "Drum Machine" Rudd</li>
44 </ul>
45
46   
47
48 </div>
49
50 </body>
```

27. To keep up the easy-to-read indenting, you can quickly indent everything within the new div by highlighting all of it and hitting Tab. You should see all of it indent in at once.
28. Go up to the styles again and add a rule-set for the ID "content". Within it, set the width to 50% (not pixels) and set the margin to auto (margin: auto) which means center it.

```
6 ▾      <style type="text/css">
7 ▾          body {
8             background-color: #6F4137;
9             font-family: 'Acme', sans-serif;
10          }
11 ▾      h1 {
12          color:white;
13          font-size:40px;
14      }
15 ▾      #main-pic {
16          width:800px;
17      }
18 ▾      #content {
19          width:50%;
20          margin:auto;
21      }
22
23      </style>
```

29. Save and reload the page. The content should be in the middle. The content itself is not actually centered, but the div containing it is centered. This is ideal because remember that it does not look good to center most text unless it is very short.
30. The band name at the top is ok to center so go back to the h1 tag and add the following style rule to it:
text-align:center;
31. This band name can also be made larger. In the h1 tag, change the font size up to 80px.
32. Save and reload the page. There is a lot more that could be done but this was enough to give you a start on adding CSS into a webpage for tag elements and IDs.

EXERCISE 3: CREATING A WEB FORM

Web forms are easy to create in HTML and to style with CSS. We will create a 2nd page for this website and experiment with web form inputs in it.

What you'll learn in this exercise:

- Linking one page to another
- Adding an HTML form
- Some of the common input types

1. In Brackets, click File > New like you did earlier to create the first page. Right away click Save As and give it the name contact.html and ensure it is in the same folder as index.html. Push Save.
2. Enter the basic webpage shell code into the new file.
3. Add the title tag to set the page title: 44 Mustangs - Contact

4. Below the opening body tag of contact.html, add a link to the homepage with this code:
`Home`
5. Now copy that same line of code for adding a link and go back into index.html (click it under "Working Files") and paste this line immediately after the Mustang logo image tag. Change the href value in this one to "contact.html" and the link text to "Contact Us" so that each page links to the other one.
6. Save both files and reload the index.html in the browser. Click the "Contact Us" link to open the new (boring) contact page and from there click "Home" to get back to the home page.
7. In the homepage in the browser, the link is right beside the Mustangs logo which doesn't look good. Go into the CSS and add a rule-set for links/anchors:


```
a {
  display:block;
  margin:10px;
}
```
8. The link is on its own line below the logo. Click again on "Contact Us".
9. This contact page needs contact information. The first thing we should add is an email address link. In Brackets, go back into contact.html.
10. Click anywhere on the homepage link code and hit Ctrl+D to duplicate it, so you have 2 consecutive identical lines. In the second line, remove the path "index.html" and replace it with "mailto:example@gmail.com" (NOTE: this is a fictional email address so do not actually send an email to it). The prefix "mailto:" in that address tells the browser that this is an email address so when it is clicked, it will try to open Outlook or another email program.
11. In that link with the email address, change the link text to "Email us now!".

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>44 Mustangs - Contact</title>
5
6  </head>
7
8
9  <body>
10     <a href="index.html">Home</a>
11     <a href="mailto:example@gmail.com">Email us now!</a>
12
13 </body>
14
15 </html>

```

12. Save and reload contact.html. Notice the links are side by side. Once again, we can add CSS and copy the same anchor rule-set as was used in the index file.

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>44 Mustangs - Contact</title>
5      <style type="text/css">
6          a {
7              display:block;
8              margin:10px;
9          }
10
11     </style>
12 </head>

```

13. Back to creating the contact information. After the email link line of code, hit Enter and type this self-closing tag `<hr />` to add a horizontal rule/line.

14. Hit Enter after that and create a web form with the `<form>` tag.
15. This creates the shell of a web form but right now it is still empty so we will need to add input fields to it. Between `<form>` and `</form>`, hit Enter several times to make space.

```
15 ▾ <body>
16     <a href="index.html">Home</a>
17     <a href="mailto:example@gmail.com">Email us now!</a>
18     <hr />
19
20 ▾     <form>
21
22
23
24
25
26
27         </form>
28
29     </body>
```

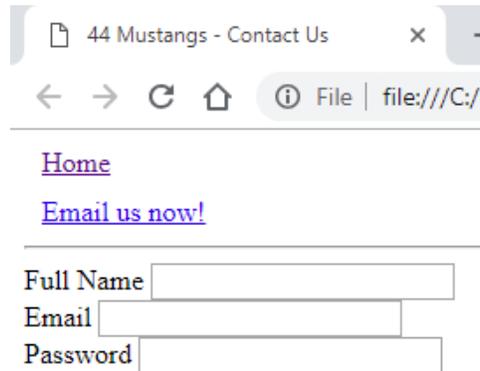
16. Let's start with a text input for the user's name. Under `<form>`, add this code:
`<input type="text" name="fullname" />`
17. Copy the same line two more times on consecutive lines. In the 2nd input line, change the name value to "email" and in the 3rd, set it to "password"

```
20 ▾     <form>
21
22         <input type="text" name="fullname" />
23         <input type="text" name="email" />
24         <input type="text" name="password" />
25
26
27
28
29     </form>
```

18. Save and reload the page.
19. There are some problems with this form as you may have noticed. For one, the 3 input boxes are beside each other which isn't appealing. Even worse, there is no indication of which textbox is for what information. Users wouldn't know where to add their name and what other information is required. We need to add labels to make it more clear.
20. Firstly, add an ID in each of the 3 input fields. The ID name can be the same as the name for each one (i.e. "fullname", "email", and "password", respectively).
21. Now we can add labels. Before each of these inputs, hit Enter and add the following line of code, but change the ID and label text for the 2nd and 3rd ones to the corresponding inputs:
`<label for="fullname">Full Name</label>`

```
<label for="fullname">Full Name</label>
<input type="text" name="fullname" id="fullname" /><br/>
<label for="email">Email</label>
<input type="text" name="email" id="email" /><br/>
<label for="password">Password</label>
<input type="text" name="password" id="password" /><br/>
```

22. Save and reload the page. Now you should see the labels appear. Click on each of the labels to see the corresponding textbox light up and become active.
23. We still want the inputs to be placed on separate lines. This can be done with CSS but it takes a few steps which we will learn more about next week. For now, just add a line break after each input line using the `
` tag.
24. Now you should see something like this:



25. This is better but we should add space between the lines now. The CSS for doing this is easy! Create a rule-set for the input tag (`input { }`) and within it, add the following line:
`margin-bottom:10px;`
26. This means to add 10px of space beneath each input element. If we had used `margin:10px` instead of `margin-bottom:10px`, it would add 10px around all 4 sides. We only need the space beneath each one, so keep the line at `margin-bottom:10px`;
27. Save and reload to see the space added between the inputs.
28. Type a couple random letters into each of the fields (don't bother typing in your name and email since the form doesn't do anything yet, so just test it quickly)
29. Notice the password field shows the characters you type. This is usually not done since passwords are supposed to be private and hidden. The reason it shows it is we used a regular text box for the password. Go down to the HTML line where we add the password field and change the type value to "password" instead of "text"
30. The updated file should look like this:

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>44 Mustangs - Contact</title>
5  <style type="text/css">
6      a {
7          display:block;
8          margin:10px;
9      }
10 input {
11     margin-bottom:10px;
12 }
13
14 </style>
15 </head>
16
17
18 <body>
19     <a href="index.html">Home</a>
20     <a href="mailto:example@gmail.com">Email us now!</a>
21     <hr />
22
23 <form>
24
25     <label for="fullname">Full Name</label>
26     <input type="text" name="fullname" id="fullname" /><br/>
27     <label for="email">Email</label>
28     <input type="text" name="email" id="email" /><br/>
29     <label for="password">Password</label>
30     <input type="password" name="password" id="password" /><br/>
31
32
33
34
35 </form>
36
37 </body>
38
39 </html>

```

31. Save and reload the page. Type a few characters in the password field. See the difference?
32. Now, this is meant to be a contact form so we actually don't need a password. If it was a sign up or login page, then passwords would be beneficial but not here. Remove the label and input lines for the password.
33. In its place, we will add some radio buttons.
34. Add a paragraph (p) element that says "Select your favourite album"
35. Beneath that, add the following lines to create the first radio button option:


```

<input type="radio" name="album" id="firstmustang " />
<label for="firstmustang">The First Mustang</label>

```
36. Save and reload the page to see the radio button appear with the label beside it.
37. We'll add two more radio buttons for the band's other albums:


```

<input type="radio" name="album" id="moonlighttrail " />
<label for="moonlighttrail">Moonlight Trail</label>
<input type="radio" name="album" id="purplepeople " />
<label for="purplepeople">Purple People</label>

```
38. Save and reload the page to see all 3 radio buttons. Click on each of the radio buttons (either the text label or the little circle button) to select them. Notice that selecting an album automatically deselects the previous selection, so you can only have at most one radio button of a set selected at a time.
39. The next part of the form is asking fans which songs they like. In this case, the question is not asking for their favourite one but rather for a set of multiple selections. So instead of

using radio buttons, this question will work better with checkboxes since they aren't restricted to one selection.

40. Under all the radio button code, add another paragraph tag, which should read "Which songs do you like?"
41. Then add the following lines of code to create checkboxes for the song options:

```
<input type="checkbox" name="song" value="rainy" />  
<label for="rainy">Rainy Daze</label>  
<input type="checkbox" name="song" value="first" />  
<label for="first">The First Mustang</label>  
<input type="checkbox" name="song" value="spoke" />  
<label for="spoke">(Jamming) In The Spoke</label>  
<input type="checkbox" name="song" value="valleyhill" />  
<label for="valleyhill">The Valley And The Hill</label>
```
42. Your entire form code should now look like this:

```
<form>  
  
  <label for="fullname">Full Name</label>  
  <input type="text" name="fullname" id="fullname" /><br/>  
  <label for="email">Email</label>  
  <input type="text" name="email" id="email" /><br/>  
  
  <p>Select your favourite album</p>  
  <input type="radio" name="album" id="firstmustang" />  
  <label for="firstmustang">The First Mustang</label>  
  <input type="radio" name="album" id="moonlighttrail" />  
  <label for="moonlighttrail">Moonlight Trail</label>  
  <input type="radio" name="album" id="purplepeople" />  
  <label for="purplepeople">Purple People</label>  
  
  <p>Which songs do you like?</p>  
  <input type="checkbox" name="song" value="rainy" />  
  <label for="rainy">Rainy Daze</label>  
  <input type="checkbox" name="song" value="first" />  
  <label for="first">The First Mustang</label>  
  <input type="checkbox" name="song" value="spoke" />  
  <label for="spoke">(Jamming) In The Spoke</label>  
  <input type="checkbox" name="song" value="valleyhill" />  
  <label for="valleyhill">The Valley And The Hill</label>  
  
</form>
```

43. Save and reload the page. Click around on the song options to select and deselect them. Notice how you can select as many or as few as you want; there is not a limit like on radio buttons.
44. The last input field we need is a large textbox for the fans to leave comments to the band. The text input we used for name and email is too small for this. The textarea element is better suited for longer amounts of text.
45. Add a paragraph element that says, "Leave us a comment!"
46. Under that message, use the following line to create a textarea element:

```
<textarea name="comments"></textarea>
```
47. Save and reload the page. Look in the lower-right corner of the newly added textarea. That is a handle that you can click and drag to resize it.
48. All we need to finish the form is a submission button.
49. Under the textarea, add the following line to add a button:

```
<input type="submit" name="submit" value="Send Now!" />
```

50. Save and reload the page. Notice that the words "Send Now!" are inside the button and that we didn't need a label beside it, unlike the other input fields. You may also see that the button is beside the textarea which doesn't look good. We'll need to add some CSS.
51. Click in the CSS in the head of contact.html.
52. We want to make style changes to the submit button, but not the same changes to other fields. There are a few options to do this. We could assign a class or ID to the submit button and create styles for that class/ID. However, there's another option that allows us to add styles to all submit buttons without affecting other form elements. Add the following CSS:


```
input[type=submit] {
  display:block;
}
```
53. Let's change more styles on the button. Within this same rule-set, set the background colour to purple, the font colour to white, the width to 200px, and the height to 50px.
54. Save and reload the page. Leave the inputs empty and click the button. Nothing really happens because we haven't told it how to handle anything (that's for another week!). However you should notice that the URL now has a bunch of code in it which represents the information entered. If you enter text in the name and email and click the button again, you should see that text appear within the gobbledygook in the URL.
55. Suppose we wanted to make the email field required. Find the HTML tag for inserting the email input field. Add the attribute "required" inside that tag. So that line should now say:


```
<input type="text" name="email" id="email" required /><br/>
```
56. Save and reload the page again. Now leave everything empty (or at least leave the email box empty) and click the button. Do you see what happens now? You should see:

[Home](#)
[Email us now!](#)

Full Name

Email

Selected  Please fill out this field.

The First Mustang Moonlight Trail Purple People

Which songs do you like?

Rainy Daze The First Mustang (Jamming) In The Spoke The Valley And The Hill

Leave us a comment!

This concludes this lab session. Call your TA over to check your work and receive your mark for this lab.

REMEMBER TO REMOVE YOUR MEMORY STICK FROM YOUR MACHINE AND PUT IT IN YOUR BACKPACK! (don't forget it)! 😊