CS1046 – Lab 1

Objectives:

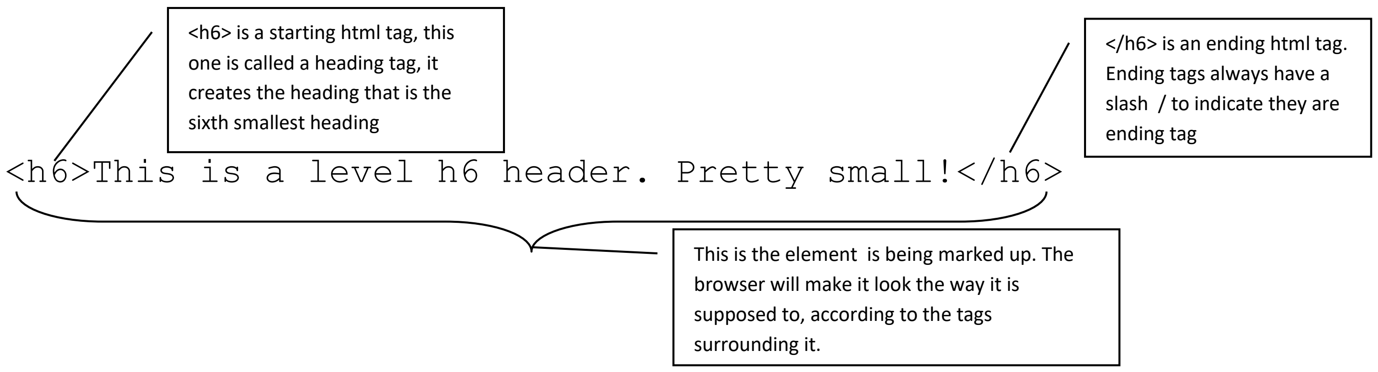
By the end of this lab you should be able to:

- View the html tags for any webpage
- Given an html file, identify at least 5 tags and explain what they are used for
- Identify an opening and closing html tag
- Open an html file in a browser
- Identify an attribute of an html tag
- Using only html tags and a simple text editor and browser, create a webpage that contains a title, a heading, an image, a link to another webpage, a list, a paragraph, a text box and a button.

Exercise 1 – Viewing, identifying and understanding html tags

1. Using the Chrome browser, go to this webpage:
   http://www.csd.uwo.ca/~lreid/cs1046/labs/lab1/backupofstanfordpage.html
   Notice that when you look at this page you can see several things such as: images, lists, links, horizontal lines. All of these elements have been created using **html tags**
   An html tag will start with an angle bracket and end with an angle back. Here is an example of an html tag:
   ```
   <h1>
   ```
   Below is another example of an html tag (the slash after the angle bracket in this html tag tells us that this tag will indicate when something stops):
   ```
   </h1>
   ```
2. Now let's look "under the hood" of a webpage. Right click on the webpage and select *View Page Source*
   You should now see the html tags and the text that makes up the webpage. This is called the html code.

3. Most html tags have a starting tag and a corresponding ending tag (always the same tag, only it now has a slash / after the angle bracket < but before the tag name). The part from the starting tag to the ending tag (including the tags and the middle part) is called an element. Elements get *marked up (*i.e. something is going to happen to that part of the content), so html is called a *markup language.* Find this part of the page source:



<h6> is a starting html tag, this one is called a heading tag, it creates the heading that is the sixth smallest heading

</h6> is an ending html tag. Ending tags always have a slash / to indicate they are ending tag

```
<h6>This is a level h6 header. Pretty small!</h6>
```

This is the element is being marked up. The browser will make it look the way it is supposed to, according to the tags surrounding it.

4. Most html tags, just like the one above have a start tag and an end tag but some don't. The one above is heading tag that makes large bolded heading. See if you can find 3 html tags in the html for the webpage you are looking at that do NOT have a corresponding ending tag.

5. If you found <li>, then you found a tag that indicates that the person who wrote this html was being a bit careless, really, the person should have included an ending tag for the <li> tag, so for example the first list should have looked like this:

```
<h2>How about a nice ordered list!</h2>
<ol>
  <li>This little piggy went to market </li>
  <li>This little piggy went to SB228 class </li>
  <li>This little piggy went to an expensive restaurant in Downtown Palo Alto </li>
  <li>This little piggy ate too much at Indian Buffet. </li>
  <li>This little piggy got lost</li>
</ol>
```

   The <ol> and </ol> tags in the box above indicate the beginning and the ending of an ordered list (ordered lists are lists that have numbers or letters but not bullets). The <li> and </li> indicate the start and end of each item in the list. (Look back at the webpage rather than the source and you should see the list and each of these list items).

Congratulations: you now can:

a. Look at webpage and then view the underlying html tags that were used to create that page
b. Identify an opening and closing html tag
c. Identify an html tag that indicates a heading
d. Identify the html tag that indicates the beginning of an ordered list and the html tag that indicates end of an ordered list.
e. Identify the html tags that indicate list items

<u>Exercise 2 – Identifying the attributes of an html tag, identifying obsolete html tags</u>

1. Open your browser and go to this page again: http://www.csd.uwo.ca/~lreid/cs1046/labs/lab1/backupofstanfordpage.html
2. Right click on the webpage and select *View Source* (or *View Page Source).*
3. Notice that besides the <li> tag, there are 2 other tags missing an ending tag. See if you can find them.
4. If you found <img src=…> and <hr>, those ones really should NOT have an ending tag. Look back at the webpage rather than the source and see if you can figure out what each of these two particular tags do.
5. Did you figure out what the <hr> tag did? It stands for horizontal rule and it draws a horizontal line at the location it is placed in the webpage.
6. An <img> tag creates an image in your document; you just have to point it to an actual jpg, gif, or png file. Let's look closely at one of the <img> tags:

```
<img align=top src="example/prettypicture.jpg" alt="Pretty
Picture">
```

   The tag above has 3 *attributes.*
   The 3 attributes for the above img tag are:
   align
   src
   alt

Attributes are additional specifications that you give to give to a tag to customize it. An attribute has an attribute *name* (e.g. src) and an attribute *value* (e.g. “example/prettypicture.jpg”). In this case, the *align* attribute tells the image to align to the top of the paragraph, the *src* attribute tells where to find the image file and the *alt* attribute tells what text to display if the person viewing your webpage decides to block out images (sometimes people with slow bandwidth don’t want to wait for images to download).

7. Now find the <body> tag. Notice it has an attribute also. Not all, but lots of tags will have attributes.
8. A few things to remember with attributes:
   a. Always surround attribute values with quotes. The person who wrote the webpage we are viewing was a bit careless. The individual wrote this:
      <img align=top src=example/prettypicture.jpg alt="Pretty Picture">
      But should have written this:
      <img align=”top” src=”example/prettypicture.jpg” alt="Pretty Picture">

   b. Type your attribute names in all lower case (you are not required to do this but it is the preferred method), so do this:
      <img align=”top” src=”example/prettypicture.jpg” alt="Pretty Picture">
      rather than this:
      <img ALIGN=”top” SRC=”example/prettypicture.jpg” Alt="Pretty Picture">

   c. If you need to include a double quote as part of your attribute value, then you can surround the attribute value with single quotes instead of double quotes, for example:
      <img align=”top” src=”example/puffdaddy.jpg” alt=’Sean **“**P Diddy**”** Combs’>

   d. **VERY IMPORTANT →** Be careful when you type your double quotes, some editors have two different types of double quotes, you want the simple ones, not the fancy ones, so you want this:
      <link rel="stylesheet" style="text/css" href="myfirststylesheet.css">
      rather than this:
      <link rel=“stylesheet” style=“text/css” href=“myfirststylesheet.css”>

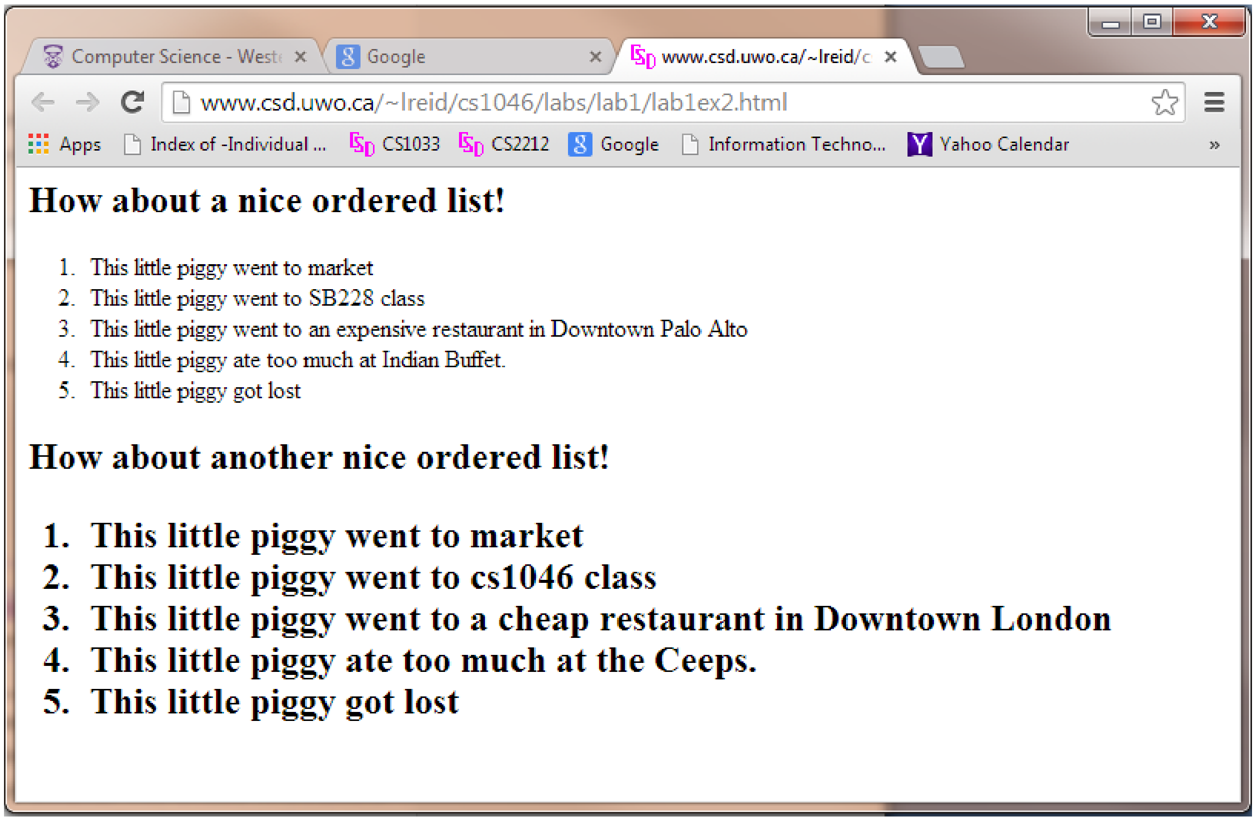
9. One more thing we should point out before leaving this example is that this page is probably a few years old because it is doing something that we don’t really want you to do any more but was common practice in the 1990s. We used to combine the tags that indicated content of the page with the tags that indicated styling of the page. By styling I mean the look and feel of the page (bolding, italics, colours). Look back at the source code and see if you can find the bold tags and the italics tags. Nowadays, we would put those tags in a separate location and handle how we “dress up the web page” a bit differently. We definitely want to have <p> tags to indicate paragraphs and <ul> tags to indicate unordered lists and <h1> tags to indicate what is a heading BUT we want to separate OUT the tags that indicate look and styling of the page. We will get to more on that in next week’s lab. For now we just want to use html to indicate the parts of a webpage and what the webpage is all about (the content).
10. To get you thinking about content, imagine you are writing a book. Don’t think about the font size or type, don’t think about the margins, just think about the content that you have written. Now consider: what are the parts of the book that you would need to indicate to your publisher (remember the publisher gets to decide on the font size, the colour of the pages, etc. BUT you get to write the book)? Try to think of at least THREE parts of a book that you could use html tags to indicate what role they play in your book!

Congratulations: you now can:

a. Identify the tag used to add a horizontal line to a webpage
b. Identify the tag used to add an image to a webpage
c. Identify attributes contained within an html tag
d. Identify the attribute name and the value given to the attribute
e. Decide when to use double quotes to enclose an attribute value and when to use single quotes
f. Identify obsolete html tags such as the italics tag
g. Conceptualize what parts of a webpage would be considered content and what parts would be considered look and feel (styling)

Exercise 3 – Debugging html files

1. Open this webpage in **CHROME** (you can use other browsers but the instructions below with be specific for Chrome so you might have to figure out on your own how to make the browser you select do the tasks we are about to try): http://www.csd.uwo.ca/~lreid/cs1046/labs/lab1/lab1ex2.html



How about a nice ordered list!

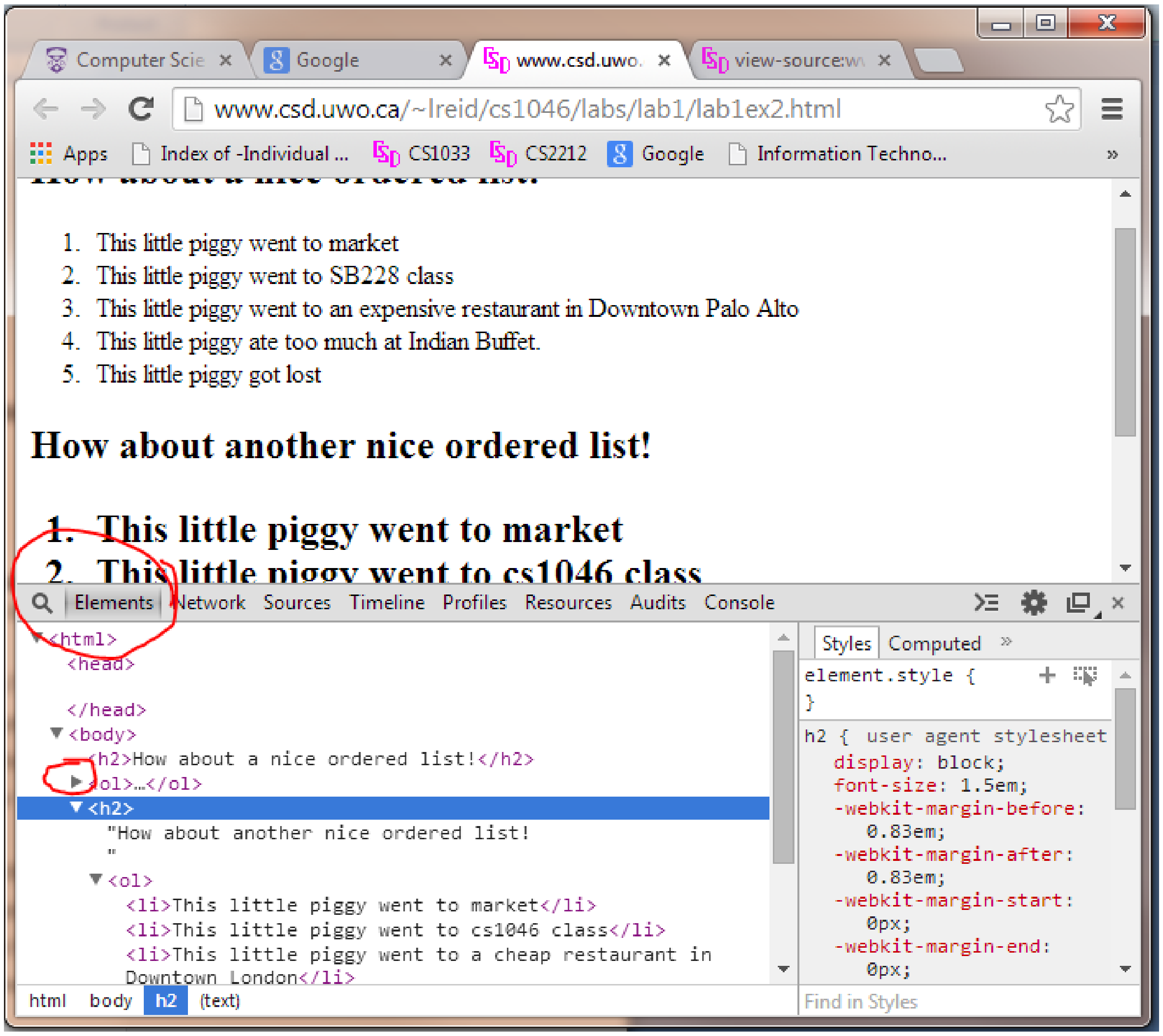
1. This little piggy went to market
2. This little piggy went to SB228 class
3. This little piggy went to an expensive restaurant in Downtown Palo Alto
4. This little piggy ate too much at Indian Buffet.
5. This little piggy got lost

How about another nice ordered list!

1. **This little piggy went to market**
2. **This little piggy went to cs1046 class**
3. **This little piggy went to a cheap restaurant in Downtown London**
4. **This little piggy ate too much at the Ceeps.**
5. **This little piggy got lost**

2. Notice that the second list has the same styling as its heading? The list and its items are bolded and large just like the heading above it.
3. Right click on the page and select *View Page Source* and, **TAKE YOUR TIME and see if you can spot the problem?**
4. Go back to the webpage view. Press *Ctrl-Shift-J* on a Windows machine or *Command-Option-J* on a Mac. This should open a little window at the bottom of Chrome. This window in Chrome is called the DevTools window. Every browser will have something similar, for example for Internet Explorer it is called Console Window and can be reached by pressing F12.
5. Look at the DevTools window and click on the *Elements* tab (seem image below). Now click(play around) with the arrows that open up the tags. These arrows show you the nesting of tags. Here, you can also see that the

second list in our example, was accidently nested inside of the <h2> tag. Actually, it wasn't even nested inside, the person who made this page FORGOT the </h2> ending tag and that messed up the whole list.



1. This little piggy went to market
2. This little piggy went to SB228 class
3. This little piggy went to an expensive restaurant in Downtown Palo Alto
4. This little piggy ate too much at Indian Buffet.
5. This little piggy got lost

How about another nice ordered list!

1. This little piggy went to market
2. This little piggy went to cs1046 class

```
▼<html>
   <head>

   </head>
  ▼<body>
     <h2>How about a nice ordered list!</h2>
    ▶<ol>…</ol>
    ▼<h2>
       "How about another nice ordered list!
       "
      ▼<ol>
         <li>This little piggy went to market</li>
         <li>This little piggy went to cs1046 class</li>
         <li>This little piggy went to a cheap restaurant in
         Downtown London</li>
```

```
element.style {
}
h2 {  user agent stylesheet
   display: block;
   font-size: 1.5em;
   -webkit-margin-before:
      0.83em;
   -webkit-margin-after:
      0.83em;
   -webkit-margin-start:
      0px;
   -webkit-margin-end:
      0px;
```

6. This is called a *Syntax* error. Markup languages like HTML and programming languages like JavaScript are **VERY, VERY picky**. If you type just one little thing wrong, a number of things can happen:
   a. first of all, your page might not even compile or show at all.
   b. your whole webpage might not look right.
   c. some of your webpage won't be correct (which is the case in the example above)
7. A few things to remember to help you with html syntax and some good practices:
   a. Generally it is recommended to type html tags in lower case. So use this tag: <h1> rather than this tag: <H1>
   b. Always spell your tag correctly (Google a tag if you can't remember how to spell it, here is a great website to help you with tags and spelling: http://www.w3schools.com/tags/)
   c. Always remember your open angle bracket < and your closing angle bracket > , without those, it isn't an html tag.
   d. Most html tags have an ending tag, double check that you included the ending tag and put it in the correct location.
   e. Most tags can be nested inside each other but try to avoid ending an outer tag BEFORE first ending off the inner tag, for example:
   This is fine:

```
<li>Click here<a href=http://www.uwo.ca> for Western</a> </li>
```
This is NOT recommended (even if it works):
```
<li>Click here<a href=http://www.uwo.ca> for Western</li> </a>
```

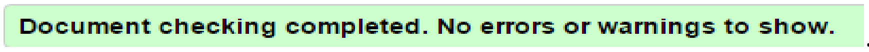
BE CAREFUL, the bad news is that, at first, syntax errors are tricky to find, the good news is that, with practice, you will get very good at spotting them quickly and fixing them early!

Congratulations: you now can:

a. Identify some syntax errors with tags
b. Open the debugging window to help you match up tags

Exercise 4 – More practice with debugging and interpreting html tags

1. Open this webpage: http://www.csd.uwo.ca/~lreid/cs1046/labs/lab1/lab1ex2a.html and look at its html (right click and select *View page source*). The link on this page works but the person who did this page did not follow one of the recommended guidelines for ***attributes***.
   a. Can you find the tag that has an attribute? (hint: it is called an anchor tag and is used to create a link)
   b. Can you figure out what they forgot?
2. Did you see the mistake? Remember that attributes almost always follow this pattern:
   attribute name then *equals sign* then *opening double quote* then *attribute value* then *closing double quote*, so usually the attribute will look similar to this:
   name="some value"
   in our case the entire tag should have looked like this:
   <a href="http://www.ceeps.com">
   but in this case, we forgot the double quotes! Some errors like this one, html can handle (it tries to be smart and figure out what we really wanted to happen) but sometimes errors like this will cause problems so be careful!
3. Notice that we kept the webpage very simple. We have a <html> at the very top to indicate this is an html file and we have its closing tag at the very last line, the </html> tag.
4. Now find the <head> and </head> tags, between these two tags is where you put things like the title, which is shown in the top tab of the browser. Also, when a user does a Google search, and your webpage is one of the webpages returned, then the title is displayed in the list of returned pages. You can also put other information between the <head> and </head> tags such as Google search keywords, etc.
5. Finally, find the <body> tag and the </body> tag. Everything in between these two tags will show up in the body of the browser window when the page is displayed. This is the way a page would look in the 1990s: fairly simple tags but of course, as time marches on, things get a bit more complicated and we now have some new tags we must include.

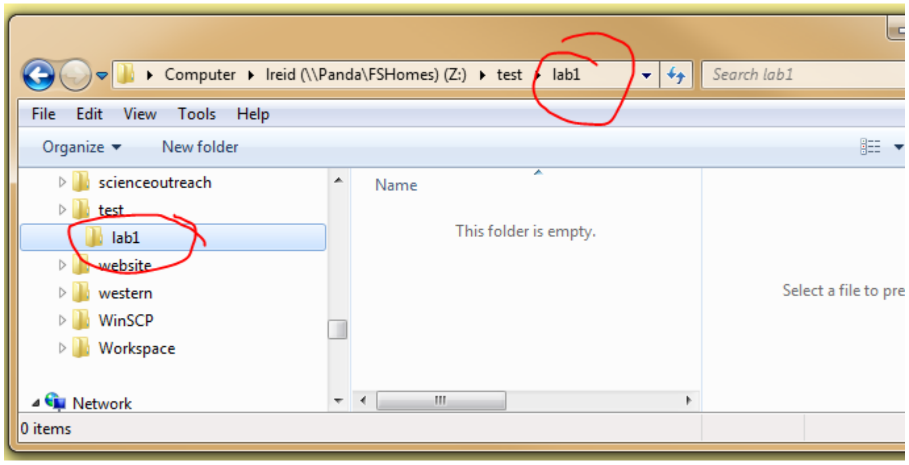
6. New standards for HTML require additional tags:
    a. The first tag in your webpage should always be:
       <!DOCTYPE html>
       The text after !DOCTYPE indicates which version of html you will be using. We are going to use the newest version: HTML5, and for this version you just put *html* after *!DOCTYPE*. BE VERY HAPPY ABOUT HTML5: if we were using the older version, HTML 4.01, we would need to put:
       <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
       GROSS ☹)
    b. In between your <head> and your </head> you must include 2 things:
        i. A title for the page using the <title> and </title> tags. The title will show up in the Chrome tab at the top of the browser.
        ii. A **<meta charset="utf-8">** tag. This tag allows you to use more than just the letters "A","B", etc. in your webpage, if for example you wanted to use Chinese characters, this tag would allow them to be displayed.
7. Open the link below, it has the standardized current html:
   http://www.csd.uwo.ca/~lreid/cs1046/labs/lab1/lab1ex2c.html
8. Notice the title of the page in the browser tab (the title is *Help*)
9. Do view page source and notice these 3 new tags: ***<!DOCTYPE html>,*** <***title>*** and ***</title>*** and ***<meta charset="utf-8">*** Try to remember to always include those tags in your pages.
10. We can check if our new page is correct (no syntax errors) by going to this site: http://validator.w3.org/ and pasting in this URL address: http://www.csd.uwo.ca/~lreid/cs1046/labs/lab1/lab1ex2c.html after the box that says Address: and then clicking on the Check button. Notice we have no errors/warning (as of January 2016 it indicated it like this → 
    Document checking completed. No errors or warnings to show.
    . Press the back button on the browser to go back to the previous screen where we pasted the address. Now paste in this address: http://www.csd.uwo.ca/~lreid/cs1046/labs/lab1/lab1ex2a.html and see how many errors were found. The errors are at the bottom of the page.
11. Finally, look at this page in your browser: http://www.csd.uwo.ca/~lreid/cs1046/labs/lab1/lab1ex2d.html
    This page has an error, view the page source and see if you can find the problem.
12. Now paste the address: http://www.csd.uwo.ca/~lreid/cs1046/labs/lab1/lab1ex2d.html into the checker: http://validator.w3.org/ It will help you find the error.

Congratulations: you now can:

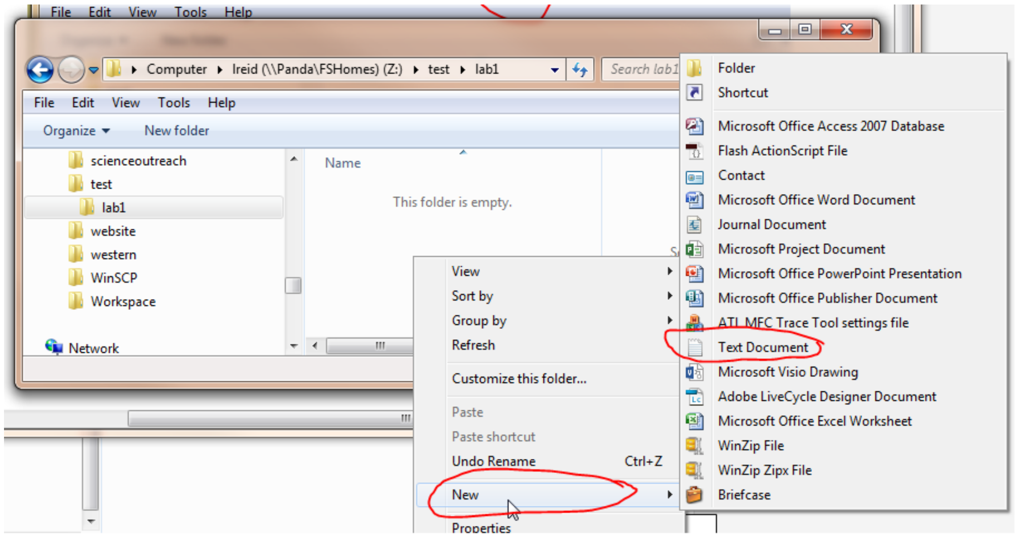
a. Identify the tag that tell which version of html the page is using, the tag that tells title of the page and the tag that indicates we want to use a character set that includes other languages
b. Identify the starting and ending html tags and the starting and ending tag for the body of the page
c. Use a website to check the syntax of the html on a given page.

Exercise 5 -- Creating your first webpage using HTML tags:

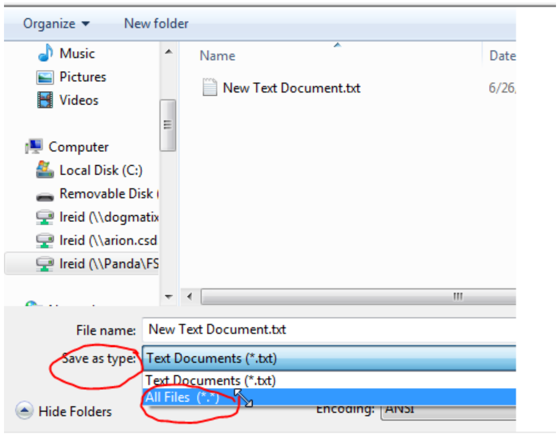
1. Let's make an introduction and menu for a fake restaurant in London called Duck Duck Duck Goose Cafe. We will type in some content and then mark it up with html tags to make it look a bit better.
2. Create a new folder/directory called *lab1*
   This is where we are going to store our work.



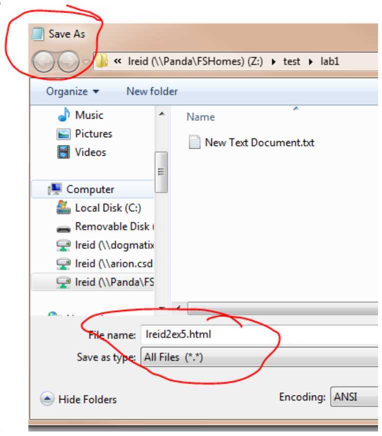
3. Go into that folder and right click then select New>Text Document. Open that new file, this will open a text editor like Notepad (TextEdit on Mac).



4. Immediately do *File>Save As* Change the *Save as Type* dropdown box to *All Files* and give the File name as yourwesternuseridex5.html (eg. jsmith2ex5.html):



File name: New Text Document.txt
Save as type: Text Documents (*.txt)
Text Documents (*.txt)
All Files (*.*)

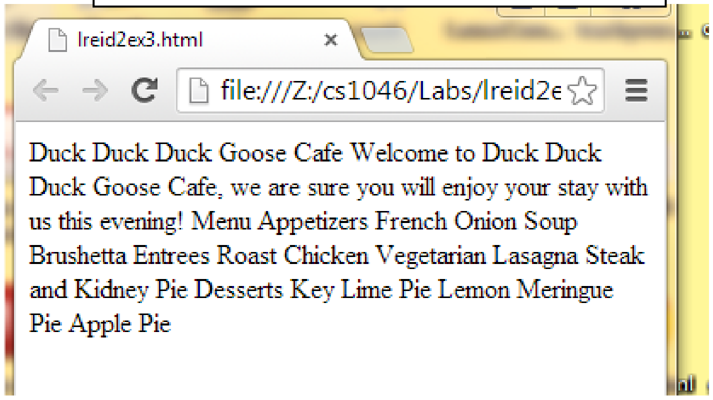


File name: lreid2ex5.html
Save as type: All Files (*.*)
Encoding: ANSI

5. Do not type any tags in initially, instead just type (or copy and paste) in the following text and then save your file →

```
Duck Duck Duck Goose Cafe

Welcome to Duck Duck Duck Goose Cafe,
we are sure you will enjoy your stay with us
this evening!

Menu

Appetizers

French Onion Soup
Brushetta


Entrees

Roast Chicken
Vegetarian Lasagna
Steak and Kidney Pie

Desserts

Key Lime Pie
Lemon Meringue Pie
Apple Pie
```

6. Double click on this file icon inside the lab1 folder, the file should automatically open in a browser. It should look similar to this:



Duck Duck Duck Goose Cafe Welcome to Duck Duck Duck Goose Cafe, we are sure you will enjoy your stay with us this evening! Menu Appetizers French Onion Soup Brushetta Entrees Roast Chicken Vegetarian Lasagna Steak and Kidney Pie Desserts Key Lime Pie Lemon Meringue Pie Apple Pie

7. **Not at all what we want ☹!** We need to mark it up with HTML. Open the file up again with Notepad (you might have to right click on it and select ***Open With*** and then select ***Notepad***). Remember that first we must add all the tags we are required to include at the beginning of an html file. To do so, paste the text in the textbox (on the right) above the Duck Duck Duck… text that already exists inside your file.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>DDDG Cafe in London, Ontario</title>
</head>
<body>
```

8. Next add the tags `</body>` and `</html>` to the very end of the file. Then save it again and double click on it again to view it in the browser. It will still not look great but now the tab in the browser should at least have a title.

9. Now let's try to format it. Probably we want the name of the restaurant to be the largest text on the webpage. So, in notepad, add heading one(<h1>) tags around the element that represents the restaurant name, like this
   `<h1>Duck Duck Duck Goose Cafe</h1>`
10. The first sentence should be a paragraph, so put **`<p>`** and **`</p>`** paragraph tags around it.
11. Now make the word Menu the second largest heading with **`<h2>`** and **`</h2>`** tags around it.
12. Make the three courses heading 3 tags **`(<h3>`** and **`</h3>)`** (i.e., Appetizers, Entrees, and Desserts)
13. Now let's make the items in the courses lists. We need an unordered list tag **`<ul>`** and **`</ul>`**around ALL the items in the menu for each of the appetizers, entrees and desserts. So for the appetizers, it will look like this:

```
<h3>Appetizers</h3>

<ul>
French Onion Soup
Brushetta
</ul>
```

14. Now we need to make each item a list element like this:

```
<ul>
<li>French Onion Soup</li>
<li>Brushetta</li>
</ul>
```

15. Now make the Entrees and Desserts into unordered lists also.
16. Save your file in Notepad and double click on the file again (or do ***refresh*** in the browser after you have saved your changes) to see if the menu webpage looks a bit better now.
17. Now we will put a link into our webpage. We want people to be able to click on the word Goose and get more information about geese. Move your cursor to just to the left of the word Goose in the main title and add this tag:
   `<a href="http://www.allaboutbirds.org/">`
   Just to the right of the word Goose, put the closing tag **`</a>.`** The line should look like this:
   `<h1>Duck Duck Duck <a href="http://www.allaboutbirds.org"> Goose </a>Cafe</h1>`
18. Now let's add an image into our document. Move your cursor to near the end of the document, just above the **`</body>`** tag but after your **`</ul>`** tag. Type in the following tag:
   `<img src="http://www.csd.uwo.ca/~lreid/cs1046/labs/lab1/duckduck.gif">`
   Save your file and view it again in the browser. Make sure the link and the image are working. NOTE: I cut and pasted the line above, the <img src=..> line and it did NOT work for me. That is because the above line has BAD quotes. If something doesn't work and you cut and pasted it, try retyping the quotes, that will often fix it. Here is the line with bad quotes:

```
<img src=”http
```

   and here is the line with good quotes (just delete and retype them in notepad or textedit):

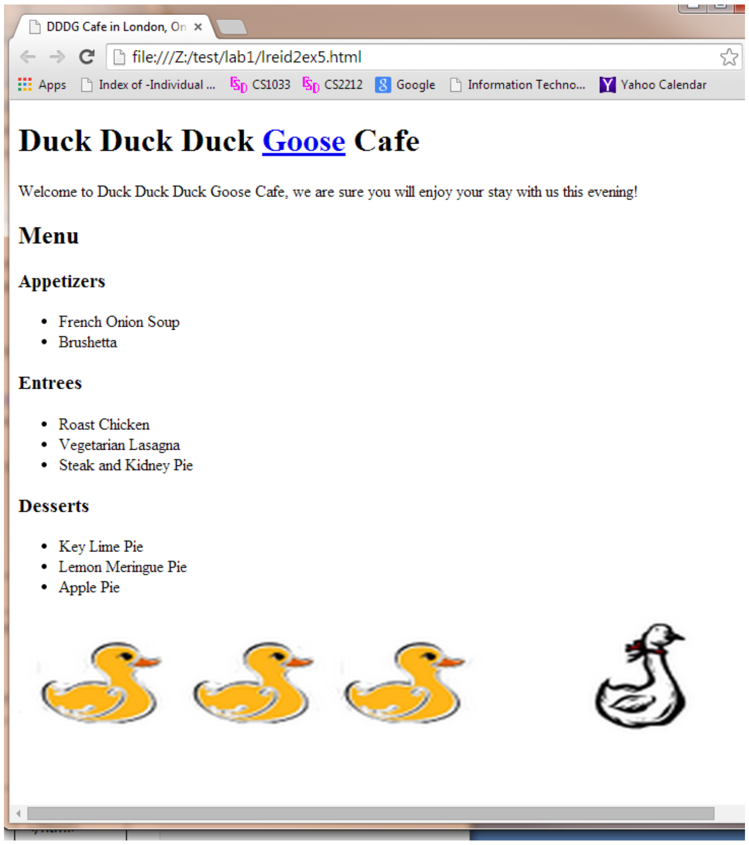
```
<img src="http
```

19. Let's try putting some more attributes on the **<img>** tag. Remember that attributes always have an attribute name equals an attribute value and the value should be enclosed in quote. Try adding two more attributes to the img tag by add the red attributes below to the **<img>** tag like this:
   `<img src="http://www.csd.uwo.ca/~lreid/cs1046/labs/lab1/duckduck.gif" width="800" height="20">`

20. Save the file and look at it again in the browser. Notice the image got distorted. The original image was 300 pixels by 60 pixels. If we want to keep it so it doesn't get distorted we need to keep the correct ratio, so instead change it to:

```
<img src="http://www.csd.uwo.ca/~lreid/cs1046/labs/lab1/duckduck.gif"
width="600" height="120">
```

21. If you have some errors, don't forget, you can go here: http://validator.w3.org and click on the 3rd tab at the top called Validate by Direct Input and paste your code into the box and then hit the Check button and it will check for errors. When you are done, if you completed the steps correctly, your page should look sort of like this:



Duck Duck Duck Goose Cafe

Welcome to Duck Duck Duck Goose Cafe, we are sure you will enjoy your stay with us this evening!

Menu

Appetizers

- French Onion Soup
- Brushetta

Entrees

- Roast Chicken
- Vegetarian Lasagna
- Steak and Kidney Pie

Desserts

- Key Lime Pie
- Lemon Meringue Pie
- Apple Pie

Congratulations: you now can:

a. Create a simple webpage using only html tags and content and a text editor such as Notepad
b. Save and open your simple webpage in a browser to check it
c. Make changes to your webpage using Notepad, save the changes and test it by reloading your page
d. Create a list of items using html tags
e. Add an image to a webpage using html tags and change the size of the image.

Exercise 6 -- Building a webpage from scratch:

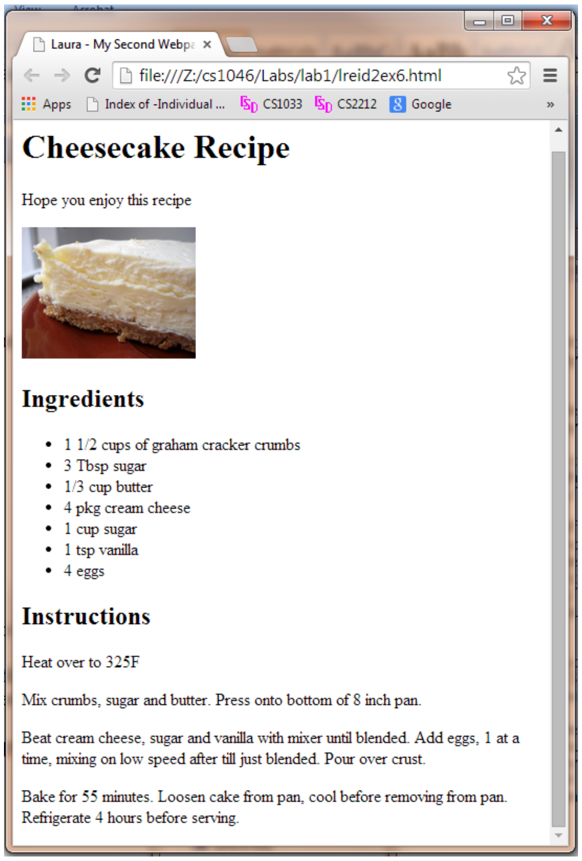
1. In your *lab1* folder, open Notepad and do File>Save As and save this new file as *yourwesternuseridex6.html* (eg. jsmith2ex6.html)
2. Add the following lines to the top of this file:
   In between the <title> and the </title> , put your name and the text: *- My Second Webpage*. For example:
   <title>Laura – My Second Webpage</title>

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title> </title>
</head>
<body>
```

3. Add the following lines to the bottom of this file:

```
</body>
</html>
```

4. Now, in the body area (i.e. between the <body> tag and the </body> tag), add the necessary tags to create a page that looks similar to the webpage below. Note: the image of the cheesecake can be found at this URL: http://www.csd.uwo.ca/~lreid/cs1046/labs/lab1/slice.jpg



Cheesecake Recipe

Hope you enjoy this recipe

Ingredients

- 1 1/2 cups of graham cracker crumbs
- 3 Tbsp sugar
- 1/3 cup butter
- 4 pkg cream cheese
- 1 cup sugar
- 1 tsp vanilla
- 4 eggs

Instructions

Heat over to 325F

Mix crumbs, sugar and butter. Press onto bottom of 8 inch pan.

Beat cream cheese, sugar and vanilla with mixer until blended. Add eggs, 1 at a time, mixing on low speed after till just blended. Pour over crust.

Bake for 55 minutes. Loosen cake from pan, cool before removing from pan. Refrigerate 4 hours before serving.

5. Save your file and check it out in a browser to make sure your webpage looks similar to the one to the right →

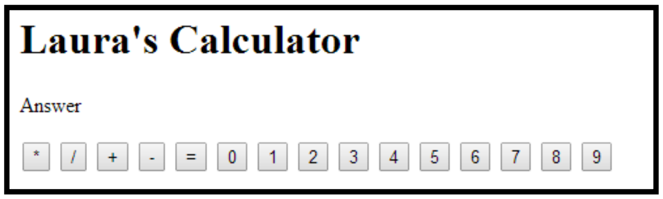
Congratulations: you now can:

a. Create a simple webpage using only html tags and content and a text editor such as Notepad by yourself!
b. Save and open your simple webpage in a browser to check it

Exercise 7 -- Using Components to get Input with HTML:

So far, the pages we have built have been fairly static. The only thing that anyone who saw our page could do, other than read it/look at it, was click on a link. We want to create webpages that do things and in order to do so, we need to get input from the user. We are going to use controls to allow the user to enter data.

1. In your lab1 folder, open Notepad and do File>Save As and save this new file as yourwesternuseridex7.html (eg. jsmith2ex7.html)
2. Add these lines to this file:
```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title> Calculator</title>
</head>
<body>
```
3. Between the <body> tag and the </body> tag we are going to try to draw a simple integer calculator with buttons. Add a heading with**`<h1>`** tags called *Your name's Calculator*
4. Then put a **`<p>`** tag with the text: *Answer*
5. Underneath your <p> tag, start making the buttons using this tag: **`<button>`** and **`</button>.`** We know we need a button for the digits 0 to 9 and for multiply, add, subtract, divide and equals. Thus your html should be starting to look like this:
6. Once you have the 10 digit buttons and the 4 operator buttons and the equals button html tags, save your file. Then view it in the browser. Notice that:
   a. the buttons don't do anything yet when we click them and
   b. it is not formatted properly:

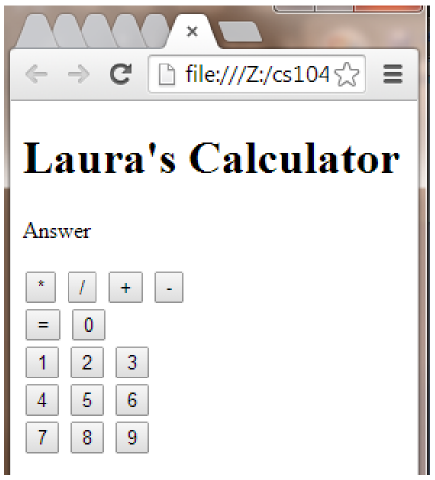


Laura's Calculator

Answer

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title> Calculator</title>
</head>
<body>
<h1>Laura's Calculator</h1>

<p>Answer</p>
<button>*</button>
<button>/</button>
<button>+</button>
<button>-</button>
<button>=</button>
<button>0</button>
<button>1</button>
<button>2</button>
<button>3</button>

….
</body>
</html>
```

7. We will fix this when we learn about CSS, but for now we can put some `<br>` tags in (they are line break tags) to make the page look a bit more like a calculator. Add four **`<br>`** tags after each of these buttons: -, 0, 3, and 6



Laura's Calculator

Answer

8. We will still need to use JavaScript to make it actually work. Try clicking on the buttons, notice that they don't do anything yet. That is because we have not written the JavaScript for it yet. In the next exercise we will look at some of the other html tags you can use to get input from the user.

Congratulations: you now can:

a. Add a button to your webpage using an html tag
b. Cause a line break in your content using an html tag

Exercise 8 -- Using components to get Input with HTML:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title> Getting Input</title>
</head>
<body>
<h1>Laura's Contest</h1>




</body>
</html>
```

1. In your lab1 folder, open Notepad and do File>Save As and save this new file as yourwesternuseridex8.html (eg. jsmith2ex8.html)
2. Add the code to the right to your page. Change the text *Laura's* to your own name. Save the file.
3. Add a *textbox* that will give the user a place to type in his/her name, by adding the following code after the **`<h1>`** tag but before the **`</body>`** tag:
   `<br>City:<input type="text" name="city" id="formCity" value="Your Home Town" size="20"><br>`
4. Save your file and view it in the browser. Then change the *size* attribute to have a value of 120 and watch what happens. Change the *value* attribute to have the *"London"* rather than *"Your Home Town".* Reload the page, again watch what happens.
5. Now we will create a dropdown box of provinces. Add the following tags right after the second <br> code in the above steps.

```
Province<select name="prov" id="whichprov">
<option>Ont</option>
<option>Que</option>
</select>
```

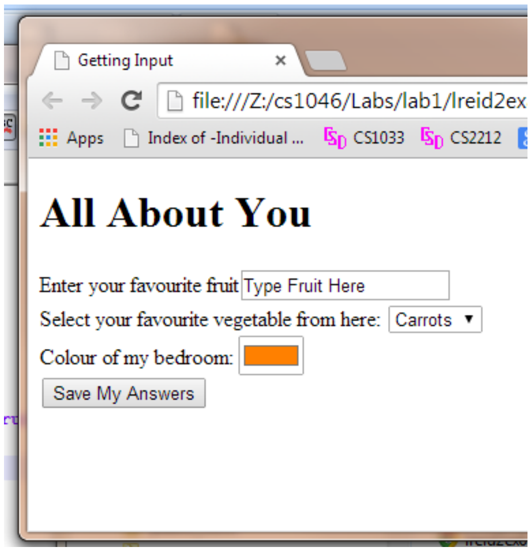
6. Save your file and view it in the browser. Add 2 more provinces to the list (you will need to the use the <option> and </option> tags). Save your work, make sure that it works in the browser.
7. After the attribute id="*whichprov*" but before the > add the following:
   ***size="3"* multiple>** so it looks like this now:
   Province<select name="prov" id="whichprov" size="3" multiple>
8. Save your file and reload it again in the browser, see the difference? (the list box now shows 3 provinces at once, rather just showing one)
9. The input components we will likely use the most are the text boxes and the buttons but just for fun, let us look at 2 more components that can be added to the webpage:
10. Right under the </select> tag, add the following tag:
   <input type="date" name="mycal" value="2015-02-19" id="mybirthday">
11. Now add this tag:
   <p>Your favourite colour:<input type="color" name="myfav" id="myfavcol">
12. Notice that all these tools we are using to get input have an attribute called ***id.*** The Id attribute is VERY important. We are going to use that attribute A LOT later on when we need to figure out how get the input from the user (like which province he/she selected) into our JavaScript.
13. Save your file and try it out. Note: some of these controls do not work in IE, try it out in Chrome.

Congratulations: you now can:

a. Create a text box where a user viewing a webpage can type in data
b. Create a dropdown box that will allow a user to select an item on a webpage
c. Add a date control to a webpage that allows a user to select a date
d. Add a colour control to a webpage that allows a user to pick a colour.

Exercise 9 -- Try it yourself:

1. In your lab1 folder, open Notepad and do File>Save As and save this new file as yourwesternuseridex9.html (eg. jsmith2ex9.html)
2. Write the HTML to create this webpage (pick different vegetables for your list):



All About You

Enter your favourite fruit Type Fruit Here
Select your favourite vegetable from here: Carrots
Colour of my bedroom:
Save My Answers

3. Add one other type of input component. Google and see what other types you can find (there are lots: radio buttons, calendars, multiline text boxes, you can add any component you want, just do a search for: *html tags input type=* and see what comes up and add one of the types to your page)
4. Save the file and make sure that it looks correct in a browser.

Congratulations: you now can:

a. Create your own webpage that allows for user input
b. Search the web for different types of controls that can be added to a webpage
c. Add new controls to your webpage.

TURN IN THIS WORK INTO OWL IN ORDER TO GET YOUR LAB MARK.

1. Go into owl and for Lab 1, submit the file you created for exercise 7 and exercise 9.