

Beginner CSS Tutorial

Mary Had a Little Lamb

Type3 Franz Fall 2008

Objectives

- Create a new site and set up folders for it
- Create Divs (Containers) and define them
- Define tags and apply them
- Make a class selector, define it and apply it
- Insert a photo and a background image
- Make a link
- And so much more I can't list it all...

Set up your folders

- 1 Make a folder on your desktop. Call it "yourlastname_lamb"
 - 2 Inside this folder, create two more folders, one called "images" the other called "resources".
 - 3 Go to laurafranz.com and follow the link for Typography 3. Download all images and text files for this project.
- Put text files in your "resources" folder.
 - Put the *mockup* file in your "resources" folder.
 - Put all jpegs in your "images" folder.

Create a new website

- 1 Open Dreamweaver
- 2 Create a new site
 - Site > New Site
 - name your site "yourlastname_lamb"
 - skip URL
 - Next
 - No for server technology
 - Next
 - Edit local copies on my machine
 - Browse and select folder for the site
 - Next
 - Remote Server = none (for now)
 - Next
 - Done

Create a new file in DreamWeaver

- 1 File > New, and select
 - blank page
 - Page type: XHTML 1 Transitional
 - Layout: <none>
 - Create
- 2 File > Save As
 - name your document "index.html". Save it in your "yourlastname_lamb" folder on the desktop.

Note: always name your "homepage" index.

Getting Started: Split Screens, DocType

- 1 Look at the upper left-hand corner of your screen. There are three buttons: "Code", "Split", and "Design". Click on "Split".
- 2 Your workspace is now split in half. The top part is the code area, and it already has some code in it. The bottom area is blank; this is the design area. We will work back and forth between the two.

- 3 Take a look at the code. It should read like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/  
xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<meta http-equiv="Content-Type" content="text/html;  
charset=UTF-8" />  
<title>Untitled Document</title>  
</head>  
<body>  
</body>  
</html>
```

What is this??

This is the doctype ("document type declaration"). It tells your website to recognize the browser so that your layout is presented correctly.

It informs the validator which version of (X)HTML you're using, and must appear at the very top of every web page.

DOCTYPEs are a key component of compliant web pages: your markup and CSS won't validate without them.

There are different types of doctypes; however we won't get too deep into this right now. If you're interested in learning more about them go to: <http://www.alistapart.com/articles/doctype/>

The most important thing to know is that you absolutely need this code—do not delete it!

Give your webpage a Title

- 1 Look at line 5 of the code:
<title>Untitled Document</title>

This is where you title your document. This has nothing to do with the name you save it with—what you type in here is what shows up at the top of the browser window. For now, type in:

```
<title>Your Last Name Lamb CSS</title>
```

Building your page structure with DIVs.

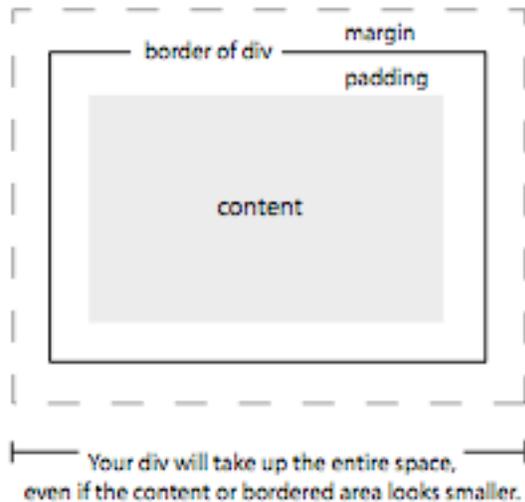
In order to put your elements into your webpage, we need to give them somewhere to “live.”

We’ll do this by making DIVs.

DIV is short for DIVision.

A `<div></div>` is a container.

It basically looks like this:



To build your page structure, we are going to make a big container that holds all the pieces together, with smaller containers that fit into it like pieces of a puzzle. We are going to call that big container “main_container.” Let’s do it now.

Creating your main container

Creating anything using CSS is a two step process. Your browser “filters” the code so you can see the visual forms. The code the browser reads is your html code. The html document contains the elements, but references the CSS document to tell the browser what the element should look like.

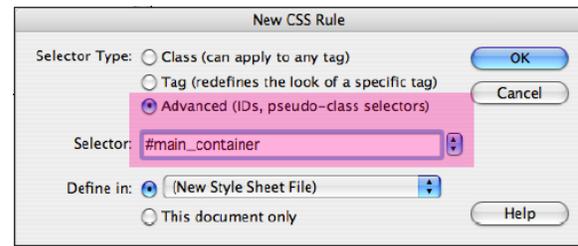
Therefore, to create the main_container div, we need to both create and define what it looks like in CSS and insert the div itself into the html.

When creating divs, I find it easier to create and define them in CSS first, then insert them into the html.

1 Look in the upper right hand corner of your screen. You should see a panel called “CSS.”

- In the upper-right hand corner of the panel is the icon for “more options.”

- Click and hold that icon
- Select “new”
- You’ll get a window that looks like this:



You are adding a new CSS rule to your document. Because you are creating a div, which is a unique element and will only be used once, you need to create what we call an ID selector.

- Selector Type: select “Advanced” (IDs, pseudo-class selectors).
- Selector: name it “#main_container” (Note: all ID selectors must start with a #. This is how the browser knows to “read” it as an ID selector.)
- Define in: select “(New Style Sheet File)”
- click “OK”

2 Save your CSS file before defining any rules.

Name your CSS file: “yourlastname_lamb.css” and save it in your folder. You will only have to do this once. The next time we add a rule, it will be much easier.

3 Once “yourlastname.css” is saved, a window will automatically pop up. Along the left you see all the ways to define how your div will look.

- Set the size of the container. Under “Box” make the width=770, the height=470, float=left, padding=0 (for all), margin=0 (for all).
- Give the container a border so we can see it when we put it into the html. Under “Border” set a solid line (for all), 1 pixel (for all), choose black for the color (for all).

4 Now, let’s insert the container into your html. All elements meant to be seen should be inserted within the `<body></body>` tags.

- in the code view, type in the following in between the body tags.

```
<div id= “main_container”>
</div>
```

It should look like this in your code view

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "ht
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; ch
<title>divs</title>
<link href="divs_3.css" rel="stylesheet" type="text/cs
</head>

<body>

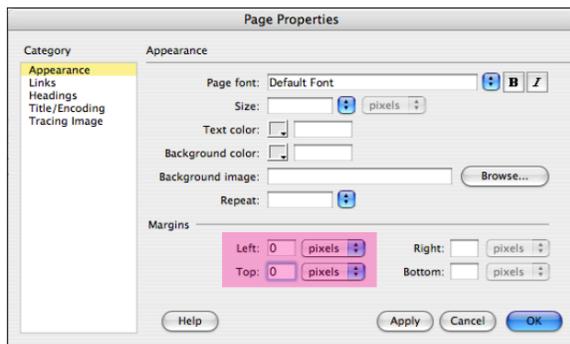
<div id= "main_container">
</div>

</body>
</html>
```

Now look at the design view. There should be 770 x 470 box with a 1 pixel black frame all the way around. You just mastered your first <div>!

Notice the box you created isn't flush with the upper left-hand corner of your design view.

- Fix that by going to modify > page properties and set the margins for left and top to 0 (zero).



Now, let's define ALL the boxes for your page. Let's start with the Navigation Bar

- 1 In the CSS window, underneath properties, click on the page + icon. (New CSS rule)
 - Select "Advanced" radial button and name the new rule "nav_container".
 - Make sure that the bottom radial button reads as "define in: yourlastname_lamb.css".
- 2 a window will automatically pop up. Along the left you will see all the ways you can define how your div will look.
 - Set the size of the container. Under "Box" make the width=770, height=30, float=none, padding=0 (for all), margin=0 (for all).
 - Give the container it's blue background color. Under "Background" type #455E78 for the background color

(Note: always use a # before a color code).

3 Now, in the code view, type in the following code within your main_container div. That means, type it in between the <div> tags for main_container.

```
<body>
<div id= "main_container">
<div id= "nav_container">
</div>
</div>
</body>
</html>
```

The design should look like this



But Yikes! In the code view, we're getting a bunch of <div> tags all nested in each other. Let's make some comments to help us identify which tag belongs to which container.

Commenting in your html document.

To type comments in html so they don't show up in your webpage, you use the following code

```
<!-- comment goes here -->
```

I always comment my code so I know what the heck I'm looking at. Enter some comments like so

```
<body>
<!-- start MAIN container-->
<div id="main_container">

<!--start NAV container-->
<div id="nav_container">
</div>
<!-- end NAV container-->

</div>
<!-- end MAIN container-->
</body>
```

Back to defining boxes. Let's do the photo.

- 1 In the CSS window, underneath properties, click on the page + icon. (New CSS rule) Select the "Advanced" radial button and name the new rule "#photo". Make sure that the bottom radial button reads as "define in: yourlastname_lamb.css".

Note: I didn't call the div "#photo_container". You can do this if you wish, but you can name it anything you want, and "#photo" works fine for our purposes.

- 2 Define the following rules for "#photo"
 - Set the size of the container. Under "Box" make the width=353, height=440, float=left, padding=0 (for all), margins=0 (for all).

3 Now, in code view, add this to your html mark-up:

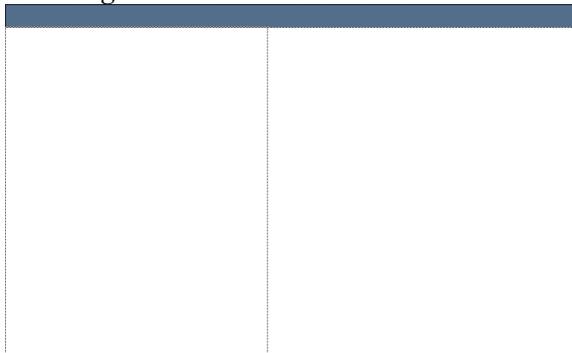
```
<body>
<!-- start MAIN container-->
<div id="main_container">

<!--start NAV container-->
<div id="nav_container">
</div>
<!-- end NAV container-->

<!-- start PHOTO container -->
<div id="photo">
</div>
<!-- end PHOTO container-->

</div>
<!-- end MAIN container-->
</body>
```

The design view should look like this



You might not see anything. The div is empty and doesn't have a background color or border. It would help if we could see what we're building, so if you can't see a light dotted border around the div in design view, turn your CSS Layout Outlines on:

- View > Visual Aids > CSS Layout Outlines

Define the column with the title in it

1 In the CSS window, underneath properties, click on the page + icon. (New CSS rule) Select the "Advanced" radial button and name the new rule "#title_column". Make sure that the bottom radial button reads as "define in: yourlastname_lamb.css".

- 2 Define the following rules for "#title_column"
 - Set the size of the container. Under "Box" make the width=160, height=auto, float=left, padding=0 (for all), margins=0 (for all).

3 Now, in code view, add this to your html mark-up:

```
<body>
<!-- start MAIN container-->
<div id="main_container">

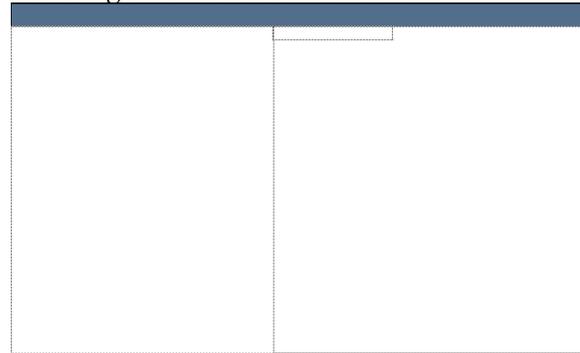
<!--start NAV container-->
<div id="nav_container">
</div>
<!-- end NAV container-->

<!-- start PHOTO container -->
<div id="photo">
</div>
<!-- end PHOTO container-->

<!-- start TITLE COLUMN container -->
<div id="title_column">
</div>
<!-- end TITLE COLUMN container-->

</div>
<!-- end MAIN container-->
</body>
```

The design view should look like this



Define the column with the lyrics in it

1 In the CSS window, underneath properties, click on the page + icon. (New CSS rule) Select the "Advanced" radial button and name the new rule "#lyrics_column". Make sure that the bottom radial button reads as "define in: yourlastname_lamb.css".

- 2 Define the following rules for "#lyrics_column"
 - Set the size of the container. Under "Box" make the width=213, height=auto, float=right, padding=0 (for all), margins=0 (for all).

3 Now, in code view, add this to your html mark-up:

```
<body>
<!-- start MAIN container-->
<div id="main_container">

<!--start NAV container-->
<div id="nav_container">
</div>
<!-- end NAV container-->

<!-- start PHOTO container -->
```

```

<div id="photo">
</div>
<!-- end PHOTO container-->

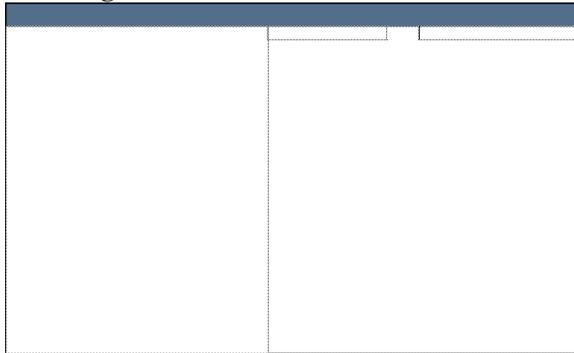
<!-- start TITLE COLUMN container -->
<div id="title_column">
</div>
<!-- end TITLE COLUMN container-->

<!-- start LYRICS COLUMN container -->
<div id="lyrics_column">
</div>
<!-- end LYRICS COLUMN container-->

</div>
<!-- end MAIN container-->
</body>

```

The design view should look like this



WAIT! There's not text or images in the containers yet, but there is something definitely wrong. I think the text columns should be lower on the page. In the mock-up, the text does not start right under the navigation bar. Let's get some text in there and see what's happening.

Putting in our content

1 Flow in the lyrics

- open the lamb_text document
- highlight and copy the lyrics text
- in Dreamweaver, in the design view, click your cursor in the lyrics column
- paste in the text

2 Flow in the Title column text

- open the lamb_text document
- highlight and copy the title column text
- in Dreamweaver, in the design view, click your cursor in the title column
- paste in the text

Lets look at the code view. You'll see the text was automatically entered into the code within the appropriate div. The `
` tag means "break" or what we call a soft return. The `<p></p>` tags define a "paragraph" and all text within the tags are part of the same paragraph. Note: if we didn't have `
` and `<p></p>` tags, the text would just keep going on the same line, because the browser wouldn't know to put in a line break!

```

<!--start NAV container-->
<div id="nav_container">
</div>
<!-- end NAV container-->

<!--start PHOTO container-->
<div id="photo">
</div>
<!-- end PHOTO container-->

<!-- start TITLE COLUMN container -->
<div id="title_column">
<p>Mary had a little lamb<br />
Traditional Children's song<br />
Written By Unknown</p>
Copyright Unknown</div>
<!-- end TITLE COLUMN container-->

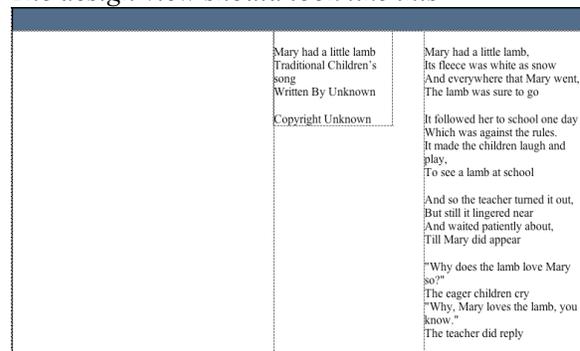
<!-- start LYRICS COLUMN container -->
<div id="lyrics_column">
<p>Mary had a little lamb,<br />
Its fleece was white as snow<br />
And everywhere that Mary went,<br />
The lamb was sure to go</p>
<p>It followed her to school one day<br />
Which was against the rules.<br />
It made the children laugh and play,<br />
To see a lamb at school</p>
<p>And so the teacher turned it out,<br />
But still it lingered near<br />
And waited patiently about,<br />
Till Mary did appear</p>
<p>&quot;Why does the lamb love Mary so?&quot;<br />
The eager children cry<br />
&quot;Why, Mary loves the lamb, you know.&quot;<br />
The teacher did reply</p>
</div>
<!-- end LYRICS COLUMN container-->

</div>
<!-- end MAIN container-->

</body>
</html>

```

The design view should look like this



Yikes. That's not what we want. We'll have to go in and set the type correctly. We need the right margins, font, size, leading, color, etc. We'll do that in a minute, but first, let's put in the photo.

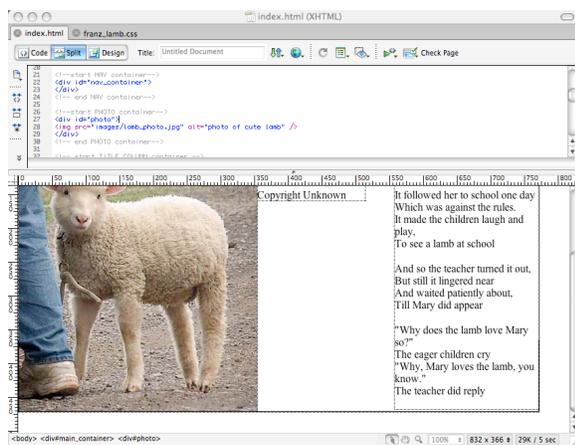
Putting in our photo

We can't just copy and paste the photo into the DW document like we did the text. But DW does have an easy way to insert an image.

Usually, one is able to put the cursor in the div they want the photo to live in, and insert the photo. I can't seem to do that this time around, so we are going to insert it directly into the code.

- put your cursor in the code, between the photo div tags.
- go to insert > image
- find the image "lamb_photo.jpg" and choose it
- the Alternate Text window will automatically come up. Give the photo a short description for those who cannot see the photo.

In split mode view, the code for the image has been inserted in the photo div, and the image is visible in the design view.



OK, now let's take care of the type!

To make type look a specific way, we need to both create and define a rule about what it looks like in CSS and insert the tag for that rule into the html.

When I work with text, I find it is easier to put the text into the html document first, then tag the type, then do the CSS to define what the tagged type should look like.

Tag the title first, typing in the code

- 1 In code view, find the title
- find the tag directly in front of the M (it should be a <p>), and change it to <H1>
- find the tag directly after the b (it should be a
) and change it to </H1>

You have just told the browser that your title is tagged as a "Headline 1." Yuck. The default H1 is not what we want. Let's make a CSS rule to define what H1 type should look like.

Define the H1 tag

- 1 In the CSS window, underneath properties, click on the page + icon. (New CSS rule) Select the "tag" radial button and select the tag "h1" from the list of options (accessed via the button with the arrows). Make sure that the bottom radial button reads as "define in: yourlastname_lamb.css".
- 2 Define the following rules for "h1"
 - Use font=Georgia, size=30, line height=36, decoration=none, weight=normal, color=#7B8A45

Tag the text under the title, using the properties window (bottom of screen)

- 1 In design view, highlight all of the text under the title
- in the properties window, choose format: Heading 2 from the format menu

You have just told the browser that this text is tagged as a "Headline 2." If you look at the code, you'll see the <h2></h2> tags around that text. Yuck. The default H2 is not what we want. Let's make a CSS rule to define what H2 type should look like.

Define the H2 tag

- 1 In the CSS window, underneath properties, click on the page + icon. (New CSS rule) Select the "tag" radial button and select the tag "h2" from the list of options (accessed via the button with the arrows). Make sure that the bottom radial button reads as "define in: yourlastname_lamb.css".
- 2 Define the following rules for "h2"
 - Use font=Georgia, size=12, style=italic, line height=18, decoration=none, weight=normal, color=#000000 (black)

Tag the lyrics, using the properties window

- 1 In design view, highlight all of lyrics
 - in the properties window, choose format: paragraph
- You have just told the browser that this text is tagged as a "paragraph." If you look at the code, you'll see the <p></p> tags around the paragraphs. The
 tags do not change.. Once again, the default <p> is not what we want. Let's make a CSS rule to define what paragraph type should look like.*

Define the paragraph tag

- 1 In the CSS window, underneath properties, click on the page + icon. (New CSS rule) Select the "tag" radial button and select the tag "p" from the

list of options. Make sure the bottom radial button reads “define in: yourlastname_lamb.css”.

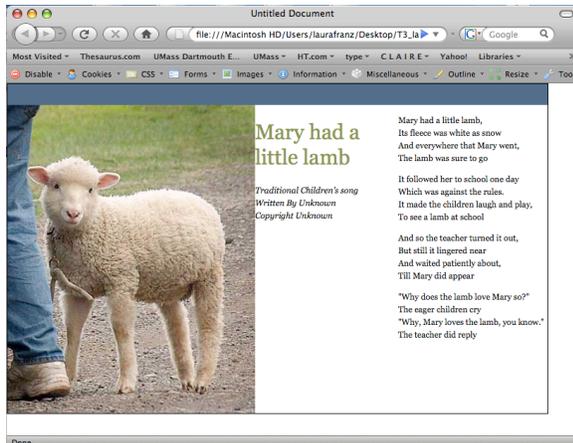
2 Define the following rules for “p”

- Use font=Georgia, size=12, line height=18, decoration=none, weight=normal, color=#000000 (black)

Let’s take a look at what we’ve got so far

- Go up to the “world” icon, click and hold, and choose a browser to preview the page in.

You should see



The type isn’t bad, but the margins aren’t correct, and what’s with those quotation marks?

Fix the hideous quotation marks

In code view, use the following code wherever you want double left quotation marks

“

wherever you want double right quotation marks

”

wherever you want single left quotation marks

‘

wherever you want single right quotation marks

’

Fix the margins on the columns

- In the CSS panel, double-click on #title_column
- Go to “Box” and change the margins by turning off “same for all” and setting the top margin = 84 pixels, left margin = 12 pixels.
- Click OK.
- In the CSS panel, double-click on #lyrics_column
- In “Box,” set the top margin = 78 pixels, and the right margin = 12 pixels. Keep the other margins at 0. Click OK.

Fix the vertical (paragraph) spacing.

- In the CSS panel, double-click on h1
- Go to “Box” and set bottom padding = 18 pixels. Set all other padding and margins to 0.
- Click OK.

You’ve just told the browser the space below all <h1> paragraphs is 18 pixels. The space above all <h1> paragraphs is 0. Notice how the title now tucks up to the top of the div.

- In the CSS panel, double-click on h2
- Go to “Box” and set all padding and margins to 0.
- Click OK.

You’ve just told the browser the space above and below all <h2> paragraphs is 0.

- In the CSS panel, double-click on p
- Go to “Box” and set bottom padding = 12 pixels. Set all other padding and margins to 0.
- Click OK.

You’ve just told the browser the space below all <p> paragraphs is 12 pixels. The space above all <p> paragraphs is 0. Notice how the lyrics now tuck up to the top of the div, but there is good space between the stanzas.

Let’s add the ornament, using two skills you’ve already learned: inserting an image, and setting padding below the text, so the image is properly placed.

- Put your cursor at the end of the italic type in the title div
- Hit return
- Insert > Image
- Find “ornament.gif” and choose it
- Type in the alternative text, and hit OK

Notice how the ornament is set tight to the text above it. Let’s fix that by adding padding to the bottom of the text.

- In the CSS panel, double-click on h2
- Go to “Box” and set the bottom padding = 12. Leave all other padding and margins at 0.
- Click OK.

You’ve just told the browser the space below all <h2> paragraphs is 12 pixels. The ornament is now properly placed.

We’ve got one more thing to take care of in this column. We need to set a couple of words as small caps with a loose tracking.

We need to create a “Class” Selector.

Why a class selector? Like ID selectors, class selectors can be used to “override” additional

formatting onto text that has been formatted using a tag (like p, h1, h2...)

Plus, class selectors can be used more than one time (unlike the unique ID selectors we used to make the divs).

You might ask, why not just use a tag? Class selectors can be used to format just a couple of words in a line. The tags (p, h1, h2, etc.) all automatically insert a line break with the closing tag. When you want to format text without a line break, use a class selector.

Creating a class selector to format text. Let's call it "subhead"

As usual, this is a two-part process. To create a "subhead" class, we need to both create and define what it looks like in CSS and mark the appropriate text as "subhead" in the html.

When creating classes, I find it easier to create and define them in CSS first, then mark the text in the html. As you've probably figured out by now, we can always go back and change what we've made.

Define the subhead class selector

1 In the CSS window, underneath properties, click on the page + icon. (New CSS rule) Select the "Class" radial button and name the new rule "subhead". Make sure that the bottom radial button reads as "define in: yourlastname_lamb.css".

2 We only need to define the changes from the <h2> tag to the subhead: size, style, case, letterspacing.

Define the following rules for "subhead"

- In "Type" make size=10, style=normal, case=uppercase.
- In "Block" make letterspacing=1 pixel (watch out! don't use ems! the space will be huge!)
- Click OK

Notice the Class shows up in your CSS list. It has a "." in front of the word "subhead." This is normal. Just like we needed to add a # in front of IDs for the browser to recognize an ID selector, we need a [dot] in front of Classes so the browser will recognize them. It's sort of weird that DW automatically adds one and not the other, but there it is.

Apply the subhead class, typing it in the code.

- in code view, find the words you want to set as "subhead" and add these tags to your html
``
- So, to apply the subhead class to "written by" your html would look like this
`Written By`

Apply the subhead class, using the properties window (at bottom of the screen).

- in design view, find the words you want to set as "subhead" and highlight them
- go down to the properties window and choose the style "subhead" from the style menu
- DW will automatically write the code for you.
- So, if you apply the subhead class to "copyright" and then look in the code view, your html will look like this
`Copyright`

We've only got one last bit of text to attend to... the Navigation! (links at top).

- Go to the lamb_text document and copy the text for the links
- Go to DW design view, click in the nav_container and paste the links text into the container.

Tag the navigation text, using the properties window (bottom of screen)

- In design view, highlight all of the navigation text
- in the properties window, choose format: Heading 3 from the format menu

You have just told the browser that this text is tagged as a "Headline 3." If you look at the code, you'll see the <h3></h3> tags around that text. Yuck. The default H3 is not what we want. Let's make a CSS rule to define what H3 type should look like.

Define the H3 tag

1 In the CSS window, underneath properties, click on the page + icon. (New CSS rule) Select the "tag" radial button and select the tag "h3" from the list of options (accessed via the button with the arrows). Make sure that the bottom radial button reads as "define in: yourlastname_lamb.css".

2 Define the following rules for "h3"

- Use font=Verdana, size=12, style=normal, line height=18, decoration=none, weight=normal, color=#FFFFFF (white)
- In "Box" set top padding=4, left padding=64, all other padding=0, and all margins=0.

Give the separation lines between the links more space... using a class selector

1 In the CSS window, underneath properties, click on the page + icon. (New CSS rule) Select the “Class” radial button and name the new rule “nav_lines”. Make sure that the bottom radial button reads as “define in: yourlastname_lamb.css”.

2 We only need to define the changes from the <h3> tag to the subhead: margins. Define the following rules for “nav_lines”

- In “Box” make the left and right padding=6. Leave everything else the same.
- Click OK

Apply the “nav_lines” class, using the properties window (at bottom of the screen).

- in design view, highlight one of the separation lines in the navigation text.
- go down to the properties window and choose the style “nav_lines” from the style menu
- DW will automatically write the code for you.
- So, if you apply the nav_lines class to a separation line and then look in the code view, your html will look like this
`|`
- Repeat for all the navigation separation lines.

Make a link by linking “UMassD Home” to the UMassD Homepage.

- In design view, highlight “UMassD Home”
- In the properties window type “http://www.umassd.edu” where it says link.
You needed to type in the entire URL including the http:// because you are linking to an external webpage. It will be easier linking to another page in your own site. Also, since you are linking to an external site, it’s good to have it come up in a new window, so let’s give it a target.

- for target, choose “_blank”
- DW did the code for you. Look in your code view to see it. The link lives between the <a href> tags.
`UMassD Home`

Excellent. You just made your first link. Now we just have to make it look the way we want it to.

Define the <a> tags, starting with “a:link”

1 In the CSS window, underneath properties, click on the page + icon. (New CSS rule) Select the “Advanced” radial button and

choose “a:link” from the pop-up options. Make sure that the bottom radial button reads as “define in: yourlastname_lamb.css”.

2 Define what the links should look like.

- Use font=Verdana, size=12, style=normal, line height=18, decoration=none, weight=normal, color=#FFFFFF (white)
- Click OK

You just told the browser what your links should look like. Next, we need to define what links look like when a cursor “hovers” over them, when a mouse click “activates” one, and when a link has been “visited.”

Define “a:hover”

1 In the CSS window, underneath properties, click on the page + icon. (New CSS rule) Select the “Advanced” radial button and choose “a:hover” from the pop-up options. Make sure that the bottom radial button reads as “define in: yourlastname_lamb.css”.

2 Define what the links should look like when the mouse hovers over them.

- Use font=Verdana, size=12, style=normal, line height=18, decoration=underline, weight=normal, color=#FFFFFF (white)
- Click OK

Define “a:active”

1 In the CSS window, underneath properties, click on the page + icon. (New CSS rule) Select the “Advanced” radial button and choose “a:active” from the pop-up options. Make sure that the bottom radial button reads as “define in: yourlastname_lamb.css”.

2 Define what the links should look like when they are activated by a click.

- Use font=Verdana, size=12, style=normal, line height=18, decoration=underline, weight=normal, color=#C0AB90
- Click OK

Define “a:visited”

1 In the CSS window, underneath properties, click on the page + icon. (New CSS rule) Select the “Advanced” radial button and choose “a:visited” from the pop-up options. Make sure that the bottom radial button reads as “define in: yourlastname_lamb.css”.

- 2 Define what the links should look like when they have been visited.
- Use font=Verdana, size=12, style=normal, line height=18, decoration=none, weight=normal, color=#FFFFFF
 - Click OK

Test your links in Firefox

- Go up to the “world” icon, click and hold, and choose a browser to preview the page in.

Add a background image

We still have to put in the running sheep. You’ll notice they run behind the photo container, the title column, and the lyrics column. That means they live in the main container. The easiest way to do this so they don’t conflict with the divs is to make them a background image in the main container.

- In the CSS panel, double-click on “#main_container”
- in “Background” find and choose the image “running_lamb.gif”
- Repeat=repeat-x
- Horizontal position=left
- Vertical position=36 pixels
- Click OK

Take another look in Firefox

- Go up to the “world” icon, click and hold, and choose a browser to preview the page in.

It should look like this



We’re going to add a couple of fine touches. Make the page horizontally center in the browser.

- In the CSS panel, double-click on “#main_container”
- in “Box” turn off the “same for all” under

margins, and set the left margin=auto and the right margin=auto. This will make the left and right margins adjust as the browser window gets bigger and smaller.

- Click OK.

Add a background color around the page

- Go to Modify > Page Properties
- In “Appearance” click on the square next to Background Color.
- An eyedropper appears
- Move the eyedropper over to the photo of the lamb and choose a color you think works well with the blue navigation bar, the photo, and the green title and ornament.
- Click OK.

Yikes! The background color is showing through most of the main_container! We want it white.

Change the background color of your main_container.

- In the CSS panel, double-click on “#main_container”
- In “Background” set your background color to white.
- Click OK.

Take another look in Firefox

- Go up to the “world” icon, click and hold, and choose a browser to preview the page in.

When I looked at mine in Firefox, I didn’t like the black border around the page. I had put it there so I could see the div, but now I want to get rid of it. So, I just removed the borders from my “#main_container” rule in the CSS panel.

Now it looks like this



Put your completed tutorial on the server

- Move the entire folder “yourlastname_lamb” into your server space.

It is very important that you move the entire folder over! Your page won't work without the images!

- Open your server space the same way you open your student storage space
- To view the completed space online, go to: “www.des.umassd.edu/yourusername/yourlastname_lamb”

Look at the CSS code

We haven't looked at the CSS code closely.

In DW, notice there are two tabs at the top of your work area.

Each tab allows you to access a file you have open in DW.

When working in your “index.html” file, you've been creating code in your “lastname_lamb.css” file as well.

It should look something like this.

```
@charset "UTF-8";
#main_container {
    margin: 0px auto;
    padding: 0px;
    height: 470px;
    width: 770px;
    border: 1px none #000000;
    background-image: url(images/running_lamb.gif);
    background-repeat: repeat-x;
    background-position: left 36px;
    background-color: #FFFFFF;
}
h1 {
    font-family: Georgia, "Times New Roman", Times, serif;
    font-size: 30px;
    line-height: 36px;
    font-weight: normal;
    text-decoration: none;
    color: #7B8A45;
    margin: 0px;
    padding: 0px 0px 18px;
}
h2 {
    font-family: Georgia, "Times New Roman", Times, serif;
    font-size: 12px;
    font-style: italic;
    line-height: 18px;
    font-weight: normal;
    text-decoration: none;
    color: #000000;
    margin: 0px;
    padding: 0px 0px 12px;
}
.subhead {
    font-size: 10px;
    text-transform: uppercase;
    letter-spacing: 1px;
    font-style: normal;
}
p {
    font-family: Georgia, "Times New Roman", Times, serif;
    font-size: 12px;
    line-height: 18px;
    font-weight: normal;
```

```

        color: #000000;
        text-decoration: none;
        margin: 0px;
        padding: 0px 0px 12px;
    }
    #nav_container {
        background-color: #455e78;
        margin: 0px;
        padding: 0px;
        float: none;
        height: 30px;
        width: 770px;
    }
    #lyrics_column {
        margin: 78px 12px 0px 0px;
        padding: 0px;
        float: right;
        height: auto;
        width: 213px;
    }
    #photo {
        margin: 0px;
        padding: 0px;
        float: left;
        height: 440px;
        width: 353px;
    }
    #title_column {
        margin: 84px 0px 0px 12px;
        padding: 0px;
        float: left;
        height: auto;
        width: 160px;
    }
    }
    h3 {
        font-family: Geneva, Arial, Helvetica, sans-serif;
        font-size: 12px;
        line-height: 18px;
        color: #FFFFFF;
        text-decoration: none;
        padding: 4px 0px 0px 64px;
        font-weight: normal;
        font-style: normal;
        background-color: #455E78;
        border-top-style: none;
        border-right-style: none;
        border-bottom-style: none;
        border-left-style: none;
        margin: 0px;
    }
    .nav_lines {
        padding-right: 6px;
        padding-left: 6px;
    }
    }
    a:link {
        font-family: Verdana, Arial, Helvetica, sans-serif;
        font-size: 12px;
        font-style: normal;
        line-height: 18px;
        font-weight: normal;
        color: #FFFFFF;
        text-decoration: none;
    }
    }
    a:hover {
        font-family: Verdana, Arial, Helvetica, sans-serif;
        font-size: 12px;
    }

```

```

font-style: normal;
line-height: 18px;
font-weight: normal;
color: #FFFFFF;
text-decoration: underline;
}
a:active {
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 12px;
font-style: normal;
line-height: 18px;
font-weight: normal;
color: #C0AB90;
text-decoration: underline;
}
a:visited {
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 12px;
font-style: normal;
line-height: 18px;
font-weight: normal;
color: #FFFFFF;
text-decoration: none;
}
}

```

Each bit of the code defines what something (tag, class, ID) looks like. For example, directly above is the definition of what “a:visited” looks like. You’ll see the name (a:visited) followed by an opening bracket. The bracket means: description starts here. Then there is a list of all the formats we applied to a:visited. Finally, a closing bracket means: description ends here. The browser uses those brackets to know what descriptions go with which rule. The name of the next rule would start on the very next line of code.

If you weren’t using DreamWeaver, you could type this code in by hand. It’s very picky (miss a colon, a semi-colon, or a bracket, and it doesn’t work), but as you can see, it’s very much like English. You can probably understand it.

Now let’s look at the HTML

Again, if you weren’t using DreamWeaver, you could type in the html by hand. It’s picky as well, but if you study it carefully, you can probably understand what is going on. It should look something like this:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Franz Lamb CSS</title>
<link href="franz_lamb.css" rel="stylesheet" type="text/css" />
<style type="text/css">
<!--
body {
margin-left: 0px;
margin-top: 0px;
background-color: #997F66;
}
-->
</style></head>

```

```

<body>

<!-- start MAIN container-->
<div id="main_container">

<!--start NAV container-->
<div id="nav_container">
  <h3>Hear the song <span class="nav_lines">|</span> See more lamb pictures <span class="nav_lines">|</span>
  Browse delicious lamb recipes <span class="nav_lines">|</span> <a href="http://www.umassd.edu"
  target="_blank">UMassD Home</a></h3>
</div>
<!-- end NAV container-->

<!--start PHOTO container-->
<div id="photo"></div>
<!-- end PHOTO container-->

<!-- start TITLE COLUMN container -->
<div id="title_column">
  <h1>Mary had a little lamb</h1>
  <h2>Traditional Children's song<br />
  <span class="subhead">Written By</span> Unknown<br />
  <span class="subhead">Copyright</span> Unknown</h2>
  <p></p>
</div>
<!-- end TITLE COLUMN container-->

<!-- start LYRICS COLUMN container -->
<div id="lyrics_column">
  <p>Mary had a little lamb,<br />
  Its fleece was white as snow<br />
  And everywhere that Mary went,<br />
  The lamb was sure to go</p>
  <p>It followed her to school one day<br />
  Which was against the rules.<br />
  It made the children laugh and play,<br />
  To see a lamb at school</p>
  <p>And so the teacher turned it out,<br />
  But still it lingered near<br />
  And waited patiently about,<br />
  Till Mary did appear</p>
  <p>&ldquo;Why does the lamb love Mary so?&rdquo;<br />
  The eager children cry<br />
  &ldquo;Why, Mary loves the lamb, you know.&rdquo;<br />
  The teacher did reply</p>
</div>
<!-- end LYRICS COLUMN container-->

</div>
<!-- end MAIN container-->

</body>
</html>

```