



Using Dreamweaver CC

4 – Creating a Template

Now that the main page of our website is complete, we need to create the rest of the pages. Each of them will have a layout that follows the plan shown below.

<i>Logo</i>	<i>Page Heading</i>
<i>Home</i>	<i>Page content in this area</i>
<i>About Us</i>	
<i>Gallery</i>	
<i>Ordering</i>	
<i>Contact Us</i>	
<i>Links</i>	

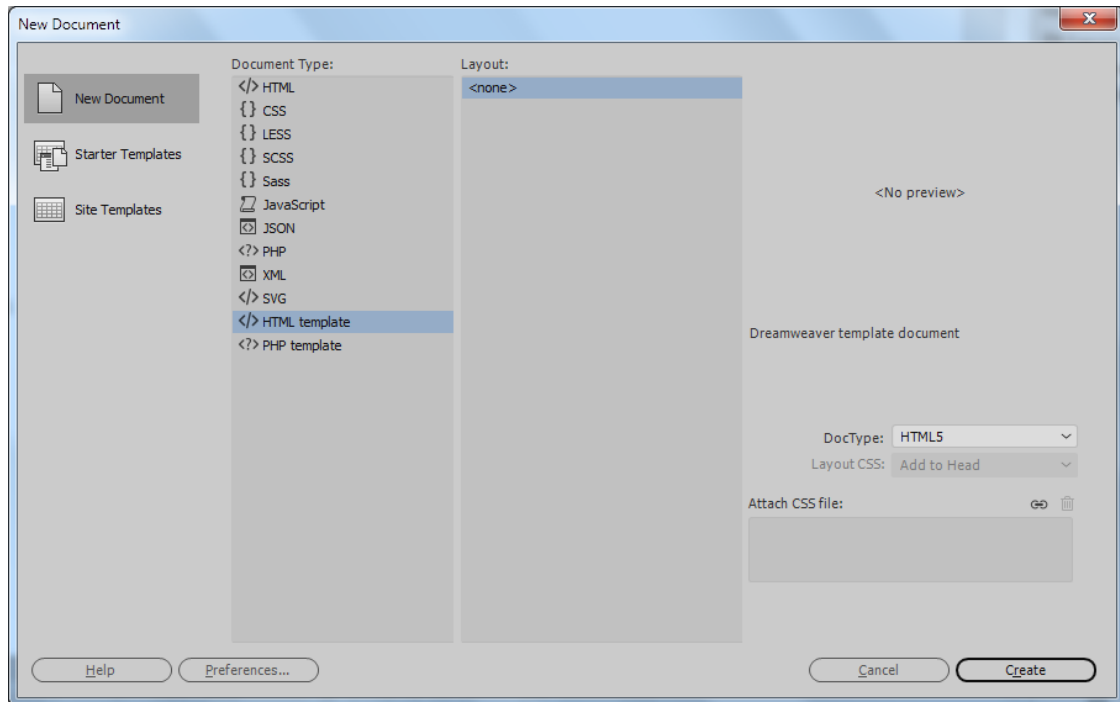
When we are creating several pages that will all be similar, we can use a template to create the basic structure and then use that template for a starting point as we create each individual page. While we create the template, we'll be using a few new techniques. Mainly, we will use CSS positioning to create the layout for the page. The CSS styles will be used to position the heading and the navigation links on the side. Earlier versions of these exercises created the page layout using a complicated arrangement of tables which used to be a common way to set up website layouts. Modern versions of Dreamweaver place a far greater emphasis on using CSS in websites in line with W3C recommendations.

Tip An alternative to creating a template is to create one page with the right layout, then use **save as** to make copies of the page under different filenames and modify the content of each one.

Exercise 1 – Starting the Template

In these exercises we will be continuing the *Francie's Flowerpot website* created in the last exercises.

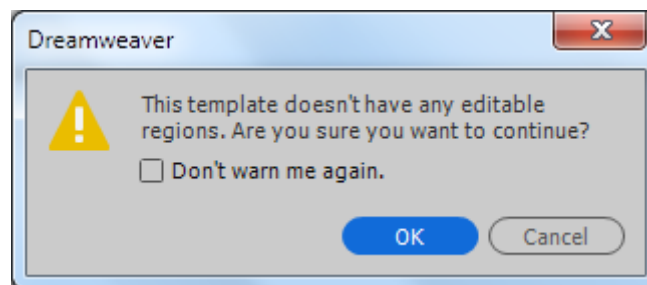
- 1) From the **File** menu select **New** ([Ctrl] [N]).



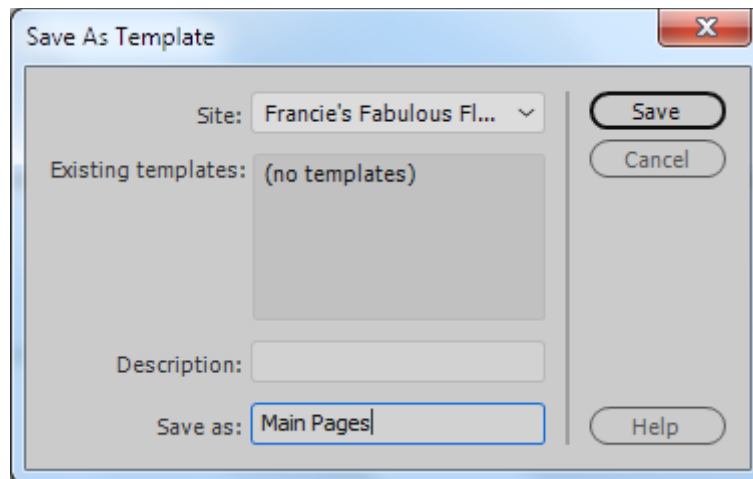
- 2) From the list of options, make sure **New Document** is selected in the left. Select **HTML template** and then click **Create**.

A new page will be created. We will use this page as the basis for our template. The first thing we will do is save the template.

- 3) From the **File** menu select **Save** or press [Ctrl] [S].

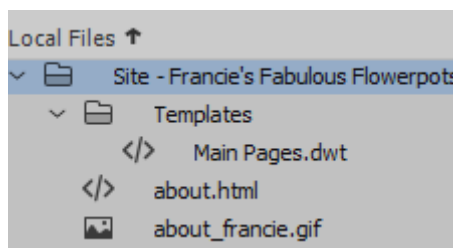


- 4) A message may appear telling us we don't have any editable regions yet. That's fine since we'll be doing that later. If you like, you can click the **Don't warn me again** option before clicking **OK**.



5) In the **Save as** area, enter *Main Pages* and click **Save**.

Note The template will be saved in a sub folder under the folder your site is in, so you can easily open the template for later modifications.



Styles in Websites

Remember than an earlier exercise stated that HTML was originally only intended to define the structure of a website. E.g. headings and paragraphs. It was never originally intended to provide a lot of information about how these page elements looked. As the world wide web became more popular, website developers wanted more control over how their websites looked. This was accomplished with the addition of HTML tags like FONT that allowed for some basic formatting options. Tags like this tended to make HTML bloated and complex.

To address this issue, the **Worldwide Web Consortium** added **CSS**. CSS stands for **Cascading Style Sheets** and is the common way for modern websites to be positioned and formatted. The idea is that HTML still defines the structure of the page, while CSS describes how the elements on the page actually look. The 'Cascading' part of the name refers to the way styles work. Style rules can cascade through your website to make it easy to control formatting throughout your whole site from one location. CSS allows you to control things such as

- Where elements will be on the page - you can be very exact with positioning
- Formatting – such as font and colour

Using styles to format your website instead of using HTML formatting tags has the following advantages:

- More precise control than HTML alone can provide over layout, fonts, colours, backgrounds, and other typographical effects.
- A way to update the appearance and formatting of an unlimited number of pages by changing just one document.
- Increased compatibility across browsers and platforms.
- Less code, smaller pages, and faster downloads.



A document in a website can be formatted by CSS in several ways.

1. Linked or Imported Stylesheets

Using this method, an extra file with a .css extension will contain all of the formatting information for your website. Then each file within your website will either link to that file or import the file. Both methods have similar results, except that using the import technique, more than one **CSS** file may be used in a single document. For example, a **CSS** file might include a style rule that specifies that level one headings will be centre aligned with large, bold navy blue text. Every HTML file that links to or imports that CSS file will have that rule applied to its level one headings so that the formatting for an entire website can be controlled from a single CSS file.

2. Embedded Styles

Using this technique, an HTML file will have a section near the top of the document that will specify style rules that will only affect that document. If the document also links to an external CSS file, then the embedded style block will have priority. For example, if an HTML document links to a CSS file that says headings will be blue, but the document also has an embedded style block saying that headings will be green, then the headings for this document will be green.

3. Inline Styles

In cases where a certain style is only needed for one part of a document, then an inline style will be used. This adds some style information to a particular HTML tag which will only affect that tag. It will also overrule any embedded or imported styles. Dreamweaver will often use embedded styles when you format some selected text in an HTML document.

Note Inline styles over-ride Embedded styles which over-ride Linked styles. E.g. If you have an external CSS file that says all of your headings are blue, but an inline style on a page says that a heading will be green, that particular heading will be green while the rest will still be blue.

Style Rules

Each CSS file, embedded style block or inline style defines its styles by making use of **Style Rules**. A Style rule consists of 2 parts. The **Selector** determines what the style rule will actually be referring to while the **Declaration** contains the style information for that selector. Look at the examples below.

```
h1 {color:navy}  
.news {color:red}
```

In the first example, the selector (h1) specifies that the style rule refers to any h1 tag in the document(s) that are affected. The declaration (color:navy) specifies that any part of the document matching the style selector will be formatted in navy blue.

In the second example, the selector uses a class called *news*. This means that any part of the documents that refers to the *news* class will be formatted in the colour red.

When there are several selectors that will have the same declaration, they may be grouped together. For example, if your level 1, 2 and 3 headings will all be formatted with a navy blue colour, then instead of having a separate style rule for each one they can be grouped in to a single rule as shown below.

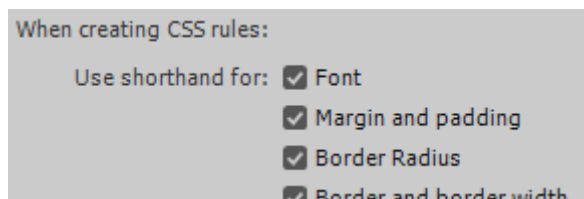
```
h1, h2, h3 {color: navy}
```

Declarations can also be grouped together. For example, if your headings are going to be bold, navy blue, 18 pixels high and formatted in Arial font, you can group the declarations in to a single rule as shown below.

```
h1 {color:#000080; font:bold 18px Arial}
```

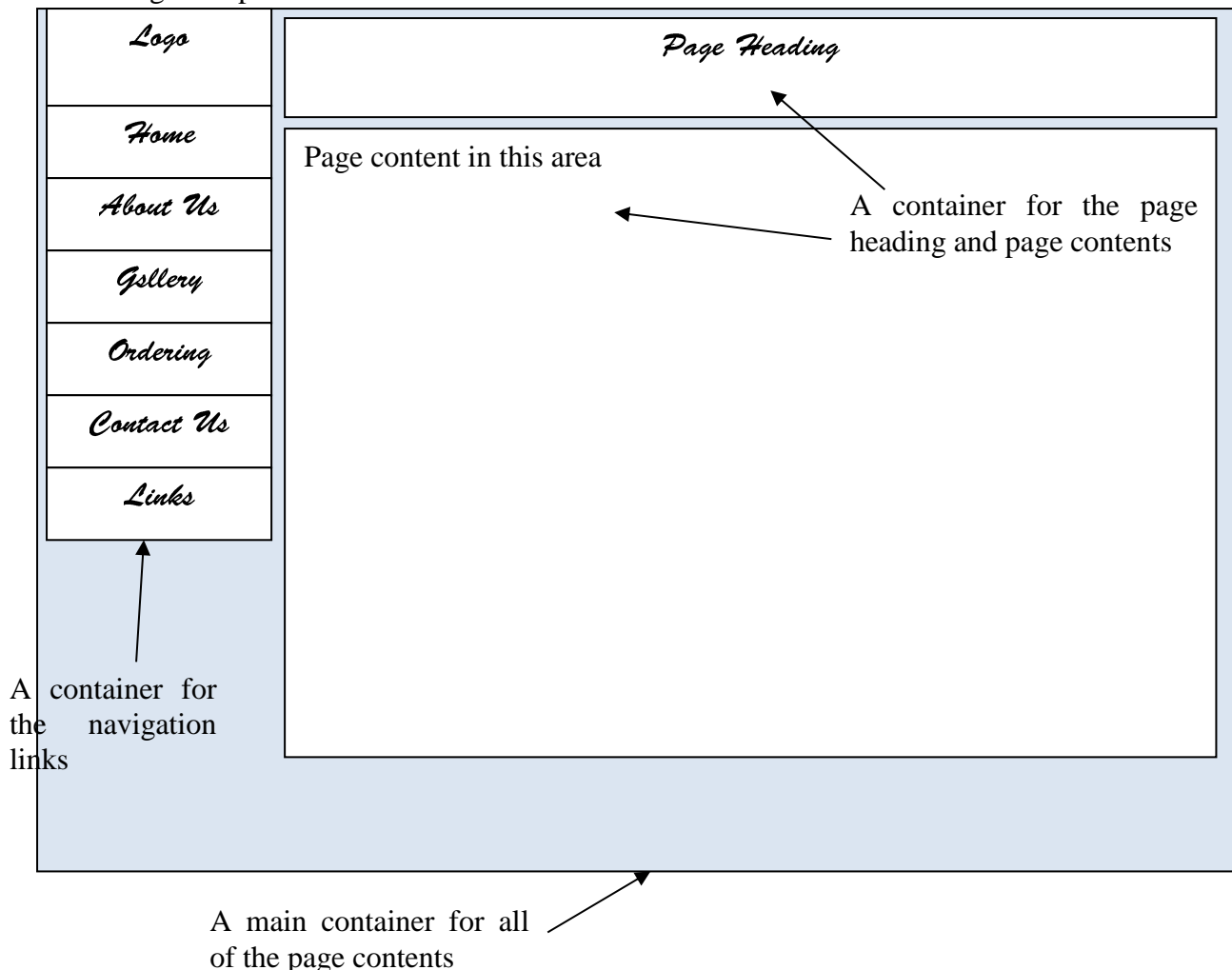
Using Dreamweaver, you can create CSS files, embedded style blocks and inline styles to take advantage of the CSS capabilities as we will do in the following exercises.

Note Dreamweaver allows you to control how CSS is written. If you want Dreamweaver to use abbreviated forms when writing CSS, you can open the Dreamweaver **Preferences** from the **Edit** menu. From the preferences you can go to the **CSS Styles** category and select the options to use shorthand.



Exercise 2 – Creating a Page Layout with CSS

Our page will be structured using certain **HTML** tags that are used to group a page in to sections. These tags create a container on the page that will then be positioned and formatted using **CSS**. Our page will be created with a series of containers inside a main container similar to the following example.



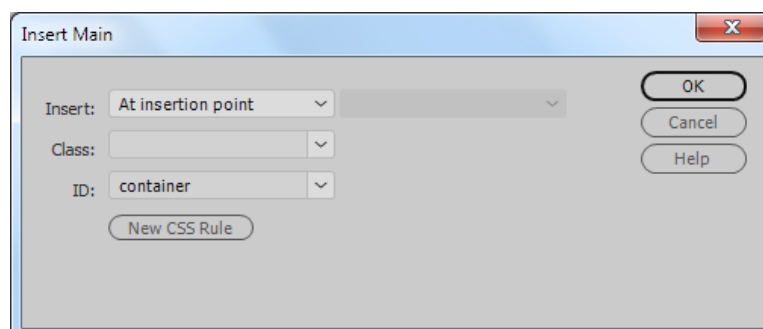
As you can see in the example, one container will be used to position the page contents on the screen. This will allow us to determine how much of the screen's width will be taken up with the page contents. Each of the other components on the page, I.e. navigation links, heading and contents will be contained inside the main container.

We'll start by creating the main container.

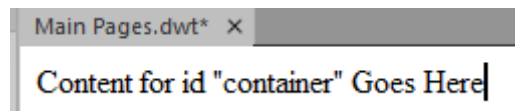
- 1) From the **Insert** menu select **Main**.

This will be used as a container for the contents of our page so we'll name accordingly.

- 2) Enter *container* for the **ID** and click **OK**.



3) Some placeholder text will be shown on the page



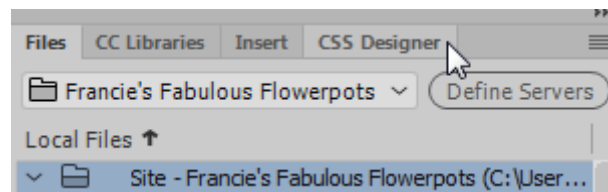
Take a look at your HTML code and you will see the **MAIN** tag with the placeholder text inside it.

```
<body>
<main id="container">Content for id "container" Goes Here</main>
</body>
```

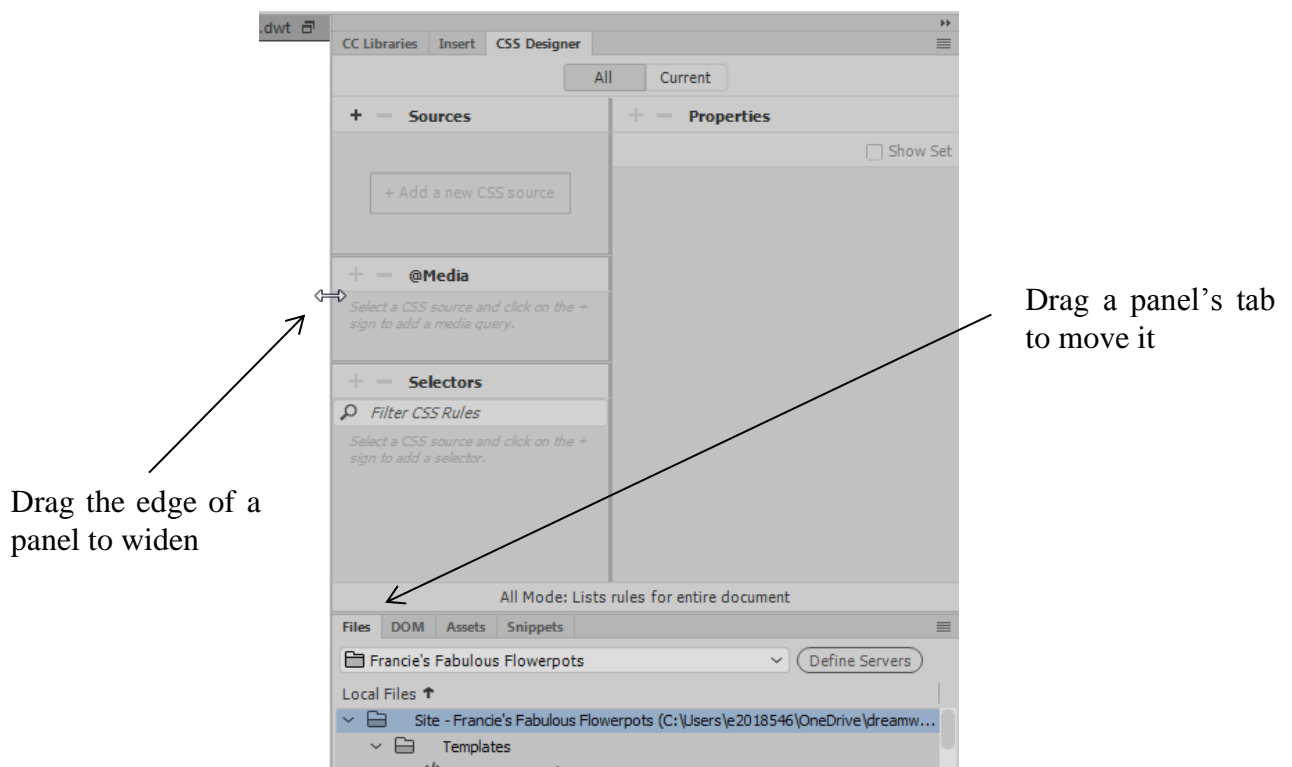
In design view you can see that you are editing a **MAIN HTML** tag which is inside the **BODY** tag. The name container will have a # in front of it since that is how HTML refers to named elements in a page.



Our other page elements will all go inside the main container but before we create them, we'll add a CSS rule that specifies how we want our container to be positioned on the page. We could do it later but adding the position now might help to visualise how the end result is going to look. To edit the CSS we'll use the **CSS Designer** panel.

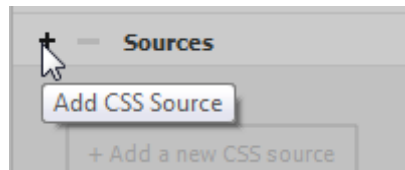


Tip If you've got a big enough monitor it helps to resize your **CSS Designer** panel so that it's wide enough to fit all of the sections. It can also help to move your Files panel in to the bottom group of tabs so that you can have the **Files** and **CSS Designer** panels both showing at the same time.

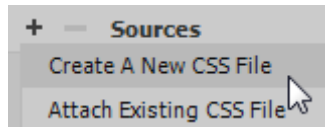


When we create a CSS rule we need to decide where it will be. Since we are about to create rules that we want to be used in all of the pages on our site, we want them to be in a CSS file that all our site's pages can refer to. That means we need to create that file.

- 6) In the **Sources** section of the **CSS Designer** panel click on **Add CSS Source**.

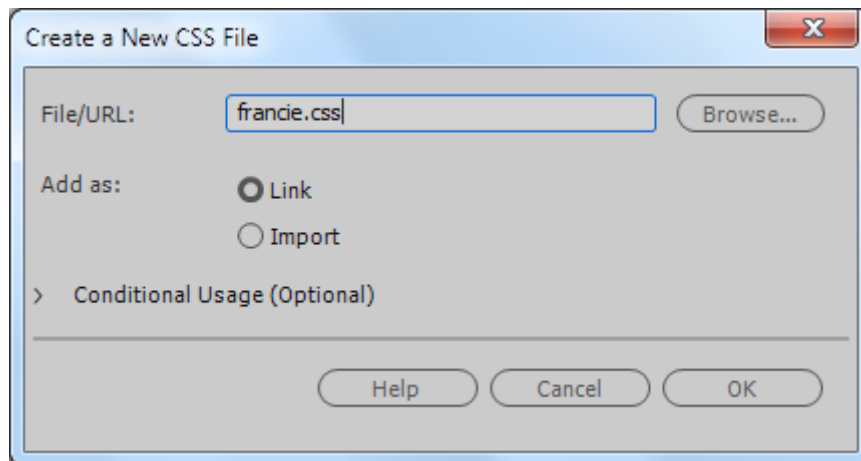


- 7) Click **Create A New CSS File**.

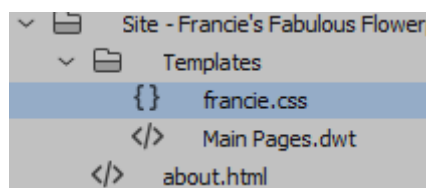


- 8) Click the browse button and make sure the location is set to the same location the rest of your website is saved in.

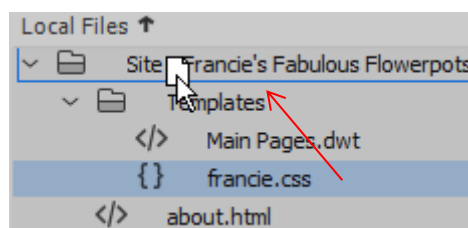
- 9) For the filename enter *francie.css* and click **OK**.



A CSS file will now appear in your file list.



- 10) If it has appeared in your templates folder, drag it to your main folder to ensure it is in the same location as your other pages. Otherwise there may be problems later when you create pages based on this template.



If you look in your HTML you will see a line that refers to that file.

```
<link href="francie.css" rel="stylesheet" type="text/css">
```


Now that we have a CSS file we can create CSS rules in that file that can be used by all of the documents that are referring to that file.

11) Click the **+** icon next to **Selectors**.

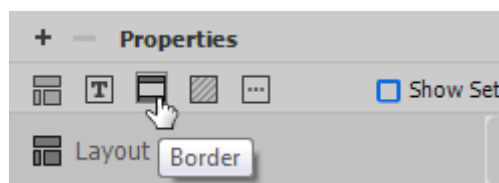
12) Make sure the selector is set to `#container`.

Usually it is filled in based on what you have selected on the page. The selector means that the style rules we create will apply to anything on the page that uses the `#container` ID.



13) In the **Properties** section of the **CSS Designer** make sure the ☐ **Show Set** option is unticked. Otherwise the panel will only show options that are already in use.

The CSS properties are grouped in to a few categories which can be changed using the buttons along the top.

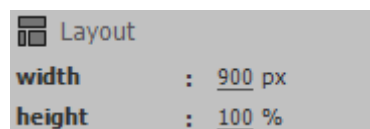


14) Make sure the **Layout** category (the first one) is selected.

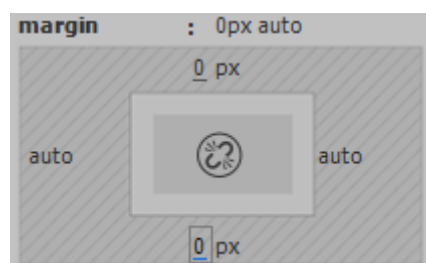
15) Move your mouse over the space next to **width** and the word **auto** will be visible. Double-click on that and change the value to `900px` (900 pixels wide).



16) Change the **height** property to `100%`.

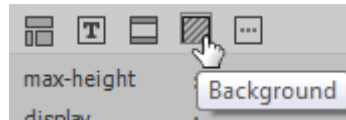


17) Scroll down until you can see the **margin** attribute. Change the margin options so that they appear as below. You can click on the left and right margin options and select **auto** from the top of the list that appears.



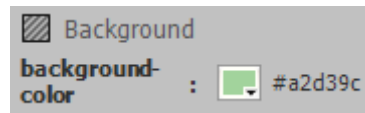
This means that there will be no gap at the top and bottom of the container. On the left and right sides, any extra space will be automatically allocated on both sides, leaving it centre aligned.

18) Change to the Background category.



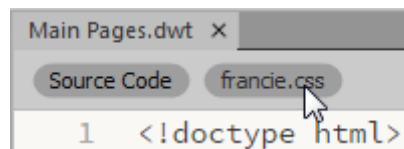
When we add other sections of the page we'll give some of them a background colour. For now we'll add a background colour to our container to make it easier to see but we'll remove it later.

19) For the **background-colour** enter #a2d39c which will give us a light green. Later we'll remove You can use the small box next to the value to open a colour picker but in this case we know the exact colour we want so we can type it in directly.

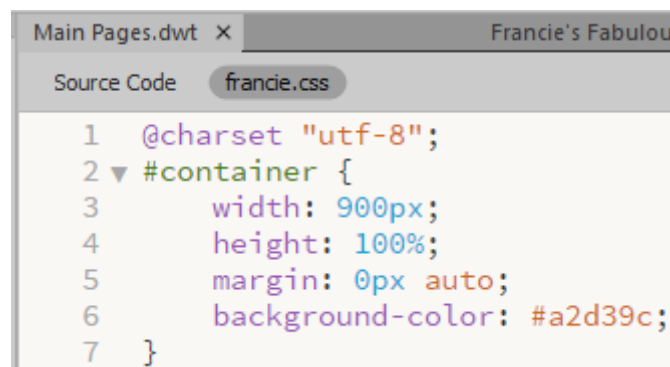


It's a lot easier to see the shape of the box now that it has a background colour. You will now be able to see it stretching to exactly 900 pixels width.

If you have a look at the HTML code nothing will have changed since the changes have all been in the CSS file.

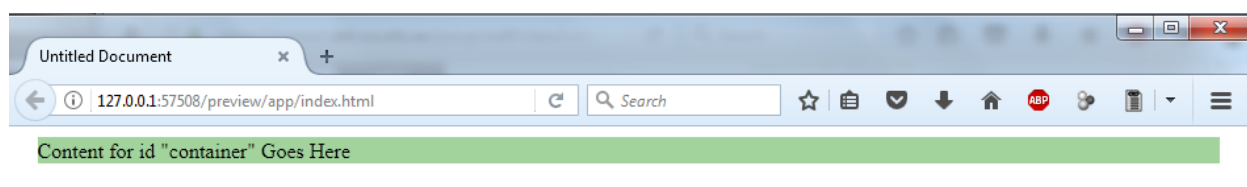


20) Switch to code view and then click the name of the CSS file at the top to view the CSS code instead of the HTML code. It will look similar to the following example.



21) Return to design view.

22) Now that we're editing 2 files (the HTML file and the CSS file) you would need to make sure that both are saved before previewing in a browser. From the **File** menu select **Save All** and then preview the page in a web browser.



If you want to make any further changes to the #container style rule you just need to make sure it's selected in the Selectors section of the **CSS Designer**.

Exercise 3 – Creating Nested Tags

Now that we have created a Container that will hold our page content, we will create an additional section for the page layout that will be inside that main division. In HTML 4 this was commonly accomplished using a series of DIV tags that each had a CSS rule attached. IN HTML 5, new tags have been added for page elements. The main tag we already added is one of these. Other examples that we will be using are:

- Header
- Nav
- Section
- Footer

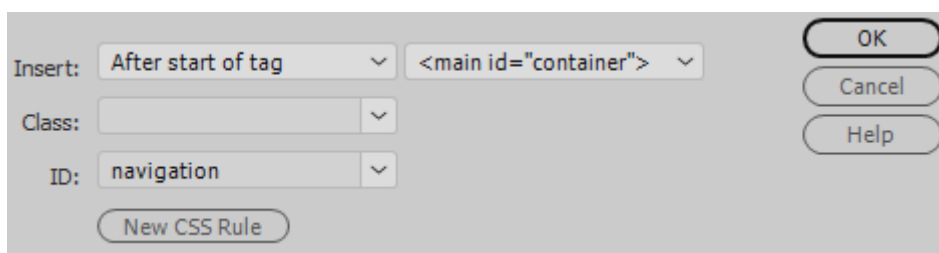
A benefit to using these elements to lay out our page is that it makes it a lot easier for search engines and text-to-speech browsers to understand how the page is organised.

Each of these elements will be placed inside the container we have already created.

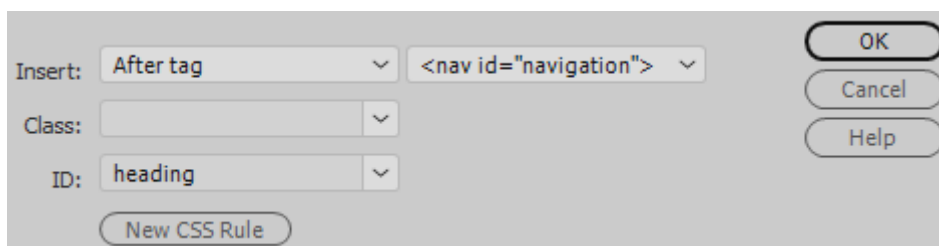
- 1) Click before the placeholder text to make sure we are inside that container.

Content for id "container" Goes Here

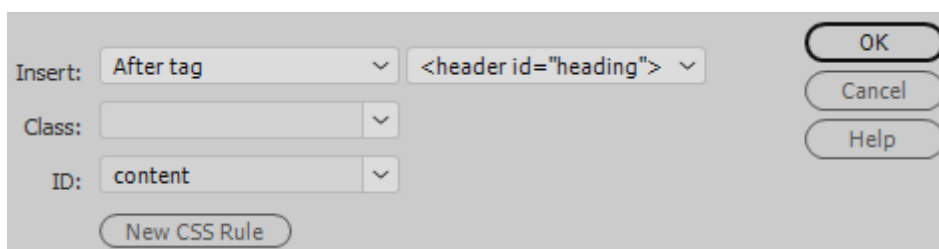
- 2) From the **Insert** menu select **Navigation**.
- 3) Enter *navigation* for the **ID** and make sure the top two options are set as follows. This will ensure that the navigation will be placed just after the container begins. I.e. it will be the first thing inside the page container.



- 4) Click **OK**.
- 5) From the **Insert** menu select **Header**.
- 6) Set the options as follows so that it goes after the header with *heading* as the **ID** and click **OK**.

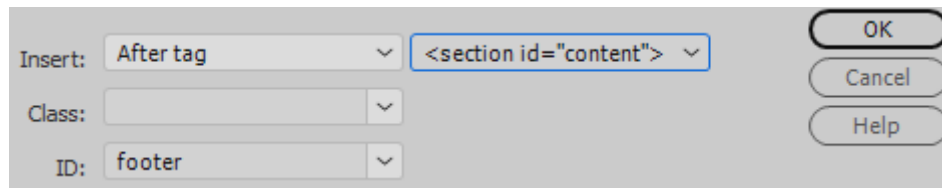


- 7) From the **Insert** menu select **Section**.
- 8) Enter *content* for the **ID** and make sure it is going after the heading.

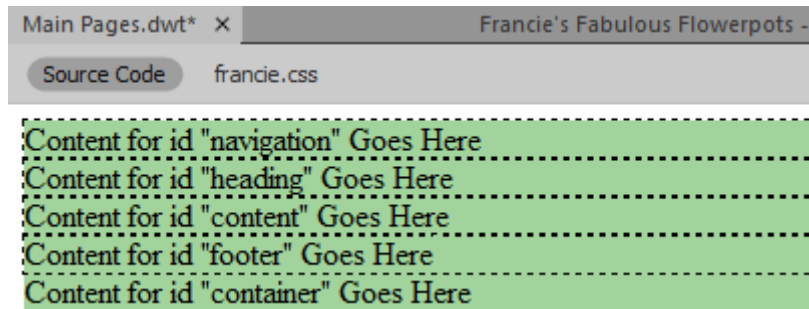


9) From the **Insert** menu select **Footer**.

10) Enter *footer* for the **ID** and make sure it is going after the content



The design view should now look like the following.



11) Check your HTML code (**Ctrl ~** shortcut to switch). It should look like the following example. If it's not, you can edit the code direction. You should have the **header**, **navigation**, **section** and **footer** tags all inside the **main** tag.

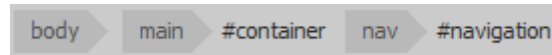
```
13 ▼ <body>
14 ▼ <main id="container">
15     <nav id="navigation">Content for id "navigation" Goes Here</nav>
16     <header id="heading">Content for id "heading" Goes Here</header>
17     <section id="content">Content for id "content" Goes Here</section>
18     <footer id="footer">Content for id "footer" Goes Here</footer>
19     Content for id "container" Goes Here</main>
20 </body>
```

12) Return to design view and save the changes to your work.

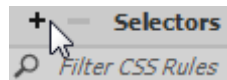
Exercise 4 – Creating CSS Rules for Page Layout

Now that we've added tags that will contain the 4 sections of the page (inside our container), we'll add some CSS rules that will position each of those section.

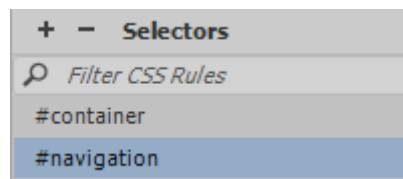
- 1) Click inside the navigation part of the page to make sure it is selected. Remember you can look at the bottom of the page to check. If it is the selected page element then the bottom of the page will show the following.



- 2) In the **CSS Designer** panel click the **+** icon next to **Selectors** to create a new selector.

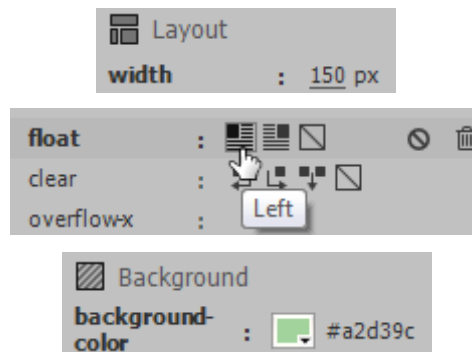


- 3) The name of the new selector should be `#navigation`. If it is `#container #navigation` then simply delete the first part.

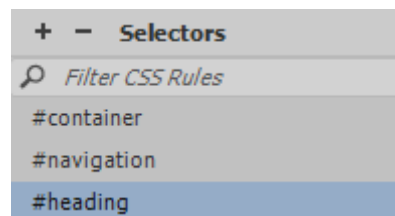


- 4) With the `#navigation` selected, choose the following CSS properties.

width	150px
float	Left (this will mean that this page element will float on the left side of the page while other page elements will wrap to the side of it)
color	<code>#a2d39c</code>

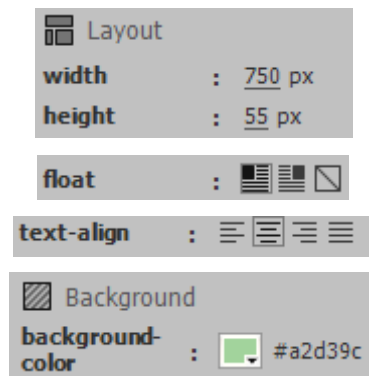


- 5) Click in the *heading* section to make sure it is selected and create a new CSS selector.



- 6) Add the following CSS properties

width	750px
Height	55px
float	left
text-align	Center (in the text category)
background-color	<code>#a2d39c</code>

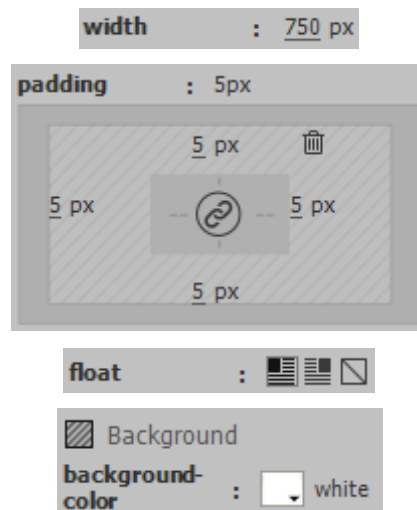


- 7) Create a new CSS selector for *#content* and make sure it is selected



- 8) Add the following CSS properties.

Width 740px
Float Left
Padding 5px
background-color white (this is in the background category of the CSS properties)



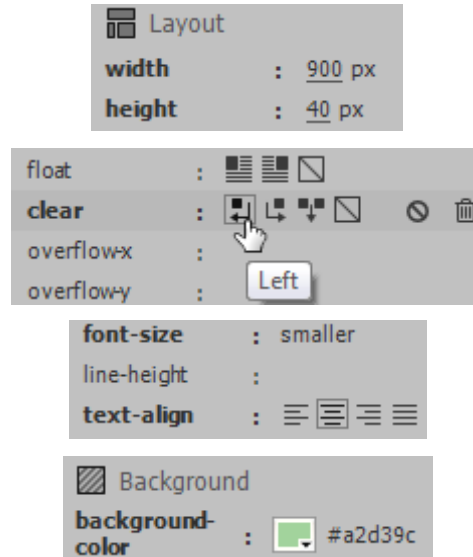
Note This section of the page will take up 750 pixels width but we have only put 740 for the width. Why? Because we also put 5px for padding so that will add an extra 5 pixels on each side. That gives us 750 total width.

- 9) Create a new CSS selector for *#footer* and make sure it is selected.



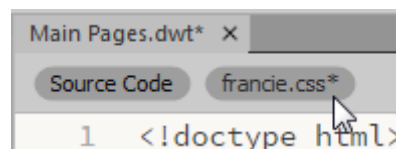
- 10) Add the following CSS properties.

Width	900px
Height	40px
Clear	left (this will clear and wrapping from another page element to make sure this starts on the left)
font-size	smaller
text-align	center
background-color	#a2d39c



If you check your code there won't be any changes since all of these changes have been in the CSS.

11) In code view make sure the CSS file is selected.



12) Save changes to the CSS file and check that the CSS rules are similar to the following example.

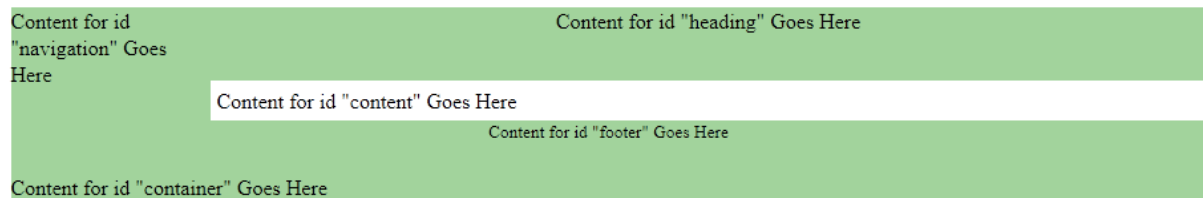
```

1  @charset "utf-8";
2  ▼ #container {
3      width: 900px;
4      height: 100%;
5      margin: 0px auto;
6      background-color: #a2d39c;
7  }
8  ▼ #navigation {
9      width: 150px;
10     float: left;
11     background-color: #a2d39c;
12 }
13 ▼ #heading {
14     width: 750px;
15     height: 55px;
16     float: left;
17     text-align: center;
18     background-color: #a2d39c;
19 }
```

```
20 ▼ #content {
21     width: 740px;
22     padding: 5px;
23     float: left;
24     background-color: white;
25 }
26 ▼ #footer {
27     width: 900px;
28     height: 40px;
29     clear: left;
30     font-size: smaller;
31     text-align: center;
32     background-color: #a2d39c;
33 }
```

13) Preview your page. It should look similar to the following example.

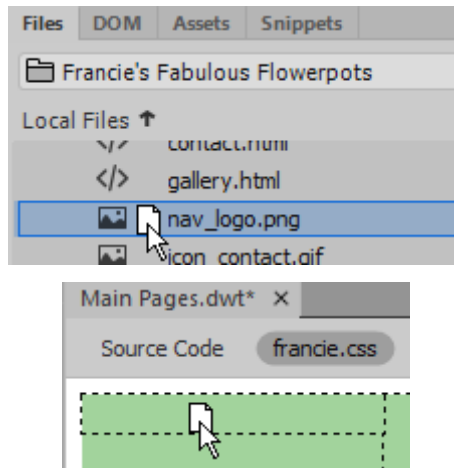
Not pretty so far but we've got the basic layout in place.



Exercise 5 – Completing the Navigation

Now we will add the individual components of the navigation bar. This will consist of **images** including several that will each be a link to a page within the site.

- 1) Delete the *Content for id "navigation" Goes Here* placeholder text in the navigation area.
- 2) Add the image called `nav_logo.png`. An easy way is to simply drag it from the **Files** panel.



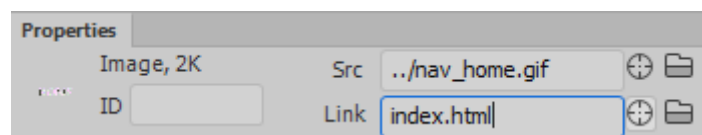
- 3) Enter *Francie's Flowerpots* as the **Alt** text for the image.
- 4) Click next to the image.



- 5)
- 6) Add the image called `nav_home.gif` and enter *Home* for the **Alt** text.
- 7) We want the next image to be placed right after this one. With the image still selected press your right arrow key so that the next one is ready to be placed right after it.
- 8)

This image is going to link to the home page.

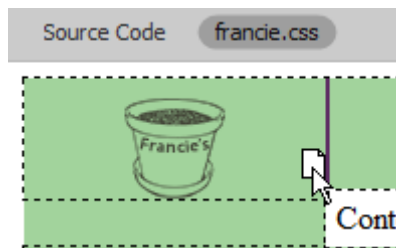
- 9) While the image is still selected, enter `index.html` for the **Link** property.



- 10) We want the next image to be placed right after this one. With the image still selected press your right arrow key so that the next one is ready to be placed right after it.



11) The next image to insert is *nav_home.gif*. You could add it from your **Insert** menu or by dragging it from the files panel. If you drag it from the files panel, make sure you drag it to a position right next to the image that is already there.



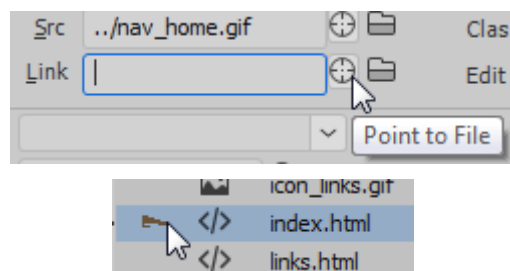
12) Enter *Home Page* for the **Alt** text.

13) Add the remaining navigation images in the following order with appropriate Alt text and link filenames for each one.

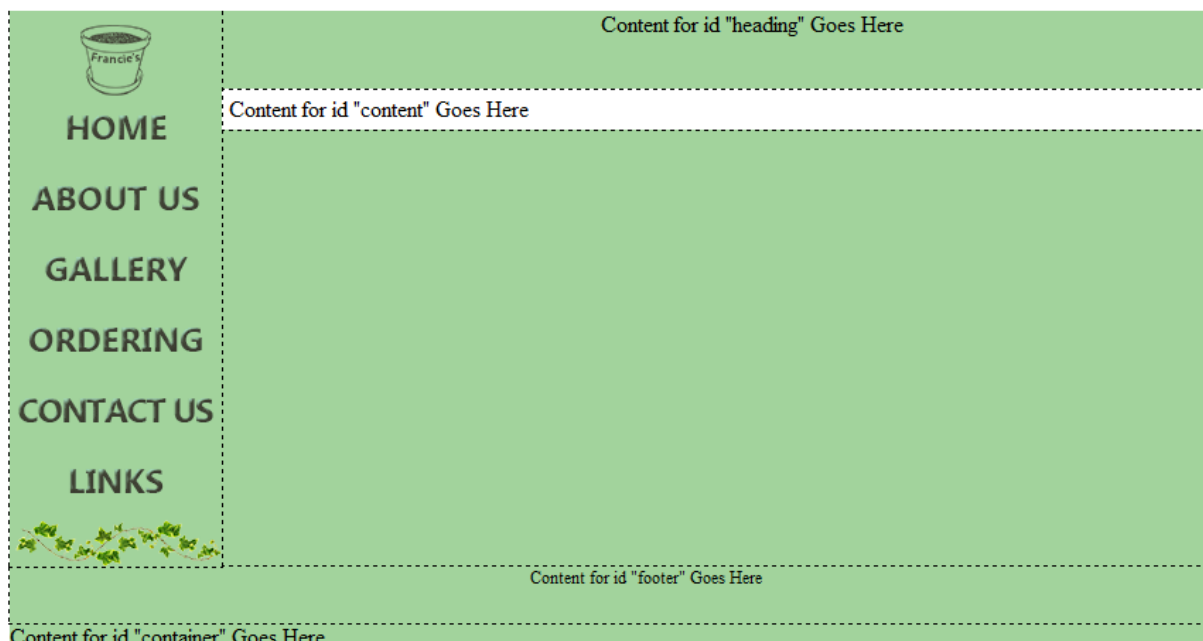
Filename	Alt text	Link
nav_home.gif	Home Page	index.html
nav_about.gif	About Us	about.html
nav_gallery_act.gif	Gallery	gallery.html
nav_ordering.gif	Ordering	ordering.html
nav_contact.gif	Contact Us	contact.html
nav_links.gif	Links	links.html
nav_bottom.png		

14) Click on the *Home* image.

15) Use the point to file option in properties to link it to index.html



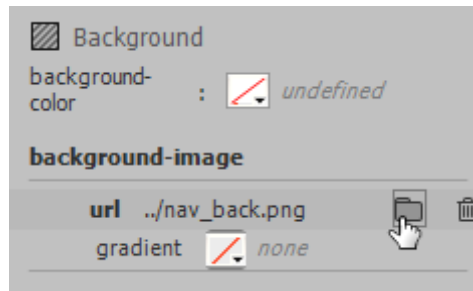
16) Add links for each of the other images as indicated in the table above.



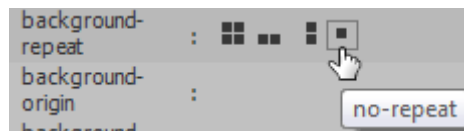
The last thing we'll add in the navigation is a background image which we'll add in the CSS.

17) Select the *#navigation* selector in the **CSS Designer** panel.

18) In the **Background** category click the browse button next to **background-image** and select the file named *nav_back.png*.



19) Set the **background-repeat** property to **no-repeat** to make sure it doesn't repeat at the bottom of the navigation section.

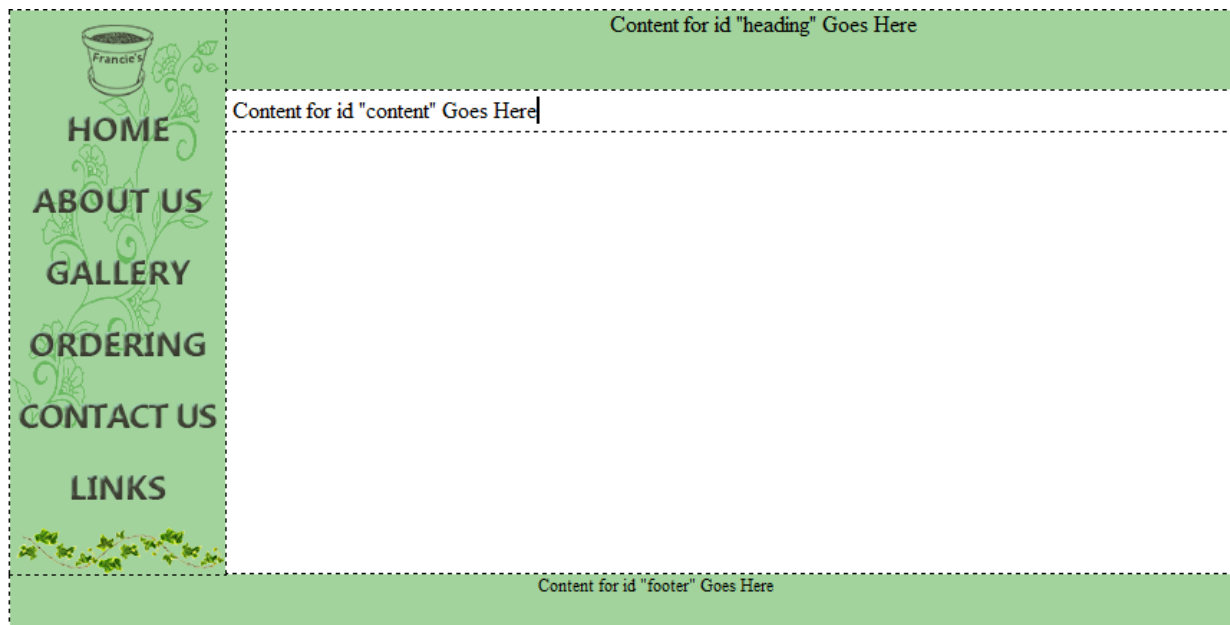
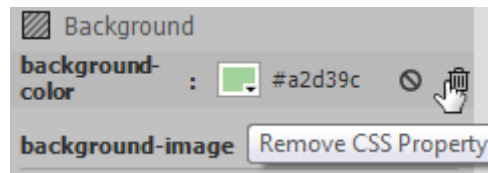


Now that we have our main sections created we no longer need to have any content in the container since it's done its job of holding everything else together.

20) Delete the *Content for id "container" Goes Here* placeholder text.

21) In the **CSS Designer** panel select *#container*.

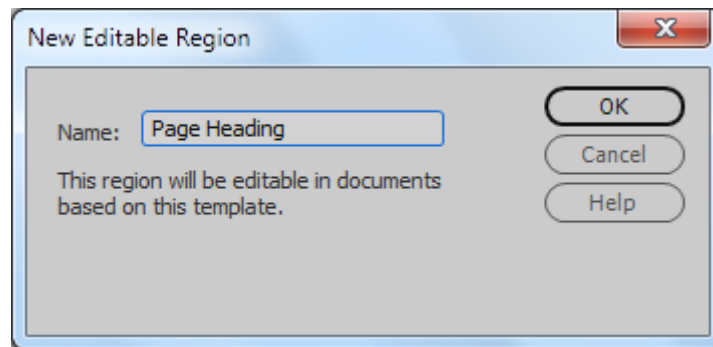
22) In the **Background** category, move your mouse over the **background-color** property and click the bin icon to remove the background colour.



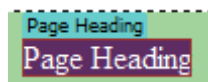
Exercise 6 – Editable Areas

We are going to create several pages based on this template. Some things like the navigation will be the same on each page. Other things like the heading and content should be different on each page. We will use editable regions to tell the template which parts of the page can be changed when a new page is made using the template.

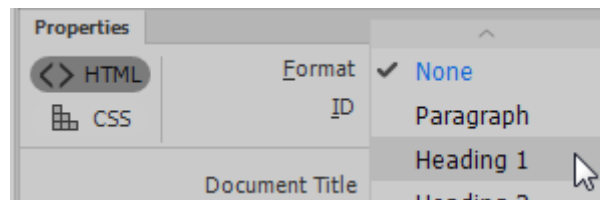
- 1) Delete the *Content for id "heading" Goes Here* placeholder text in the heading section.
- 2) From the **Insert** menu select **Template** and then **Editable Region**.
- 3) Enter *Page Heading* for the name of the editable region and click **OK**.



Dreamweaver will now add a special region to your page. When you create a new page based on your template, only parts of the page that are inside an editable region can be modified. The rest will be the same for all pages based on the template.

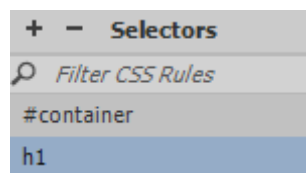


- 4) Select the editable region if it isn't already selected.
- 5) In the **Properties Panel** change the format to *Heading 1*.



We'll add a new CSS rule to format our level 1 headings.

- 6) Create a *h1* CSS selector.



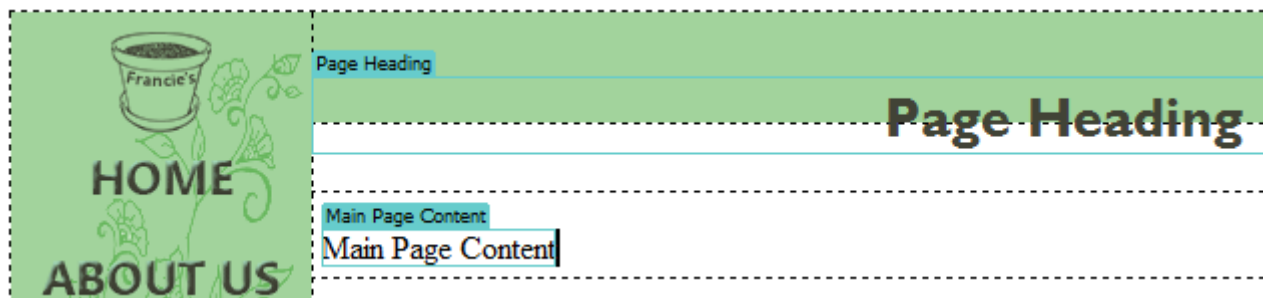
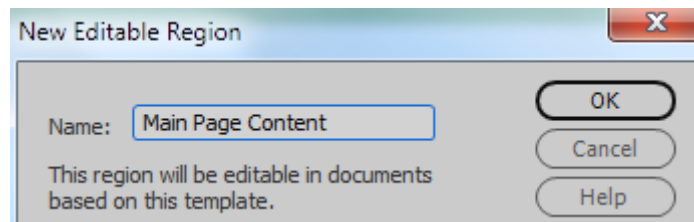
- 7) Choose the following CSS properties.

color (text colour)	#414336
font-size	28px
font-family	"Gill Sans", "Gill Sans MT", ...



The other section of the document that will be different on each page of the site is of course the content section so we need to add another editable area there.

- 8) Delete the *Content for id "content" Goes Here* placeholder text.
- 9) From the **Insert** menu select **Template** and then **Editable Region**.
- 10) Enter *Main Page Content* for the editable region name.

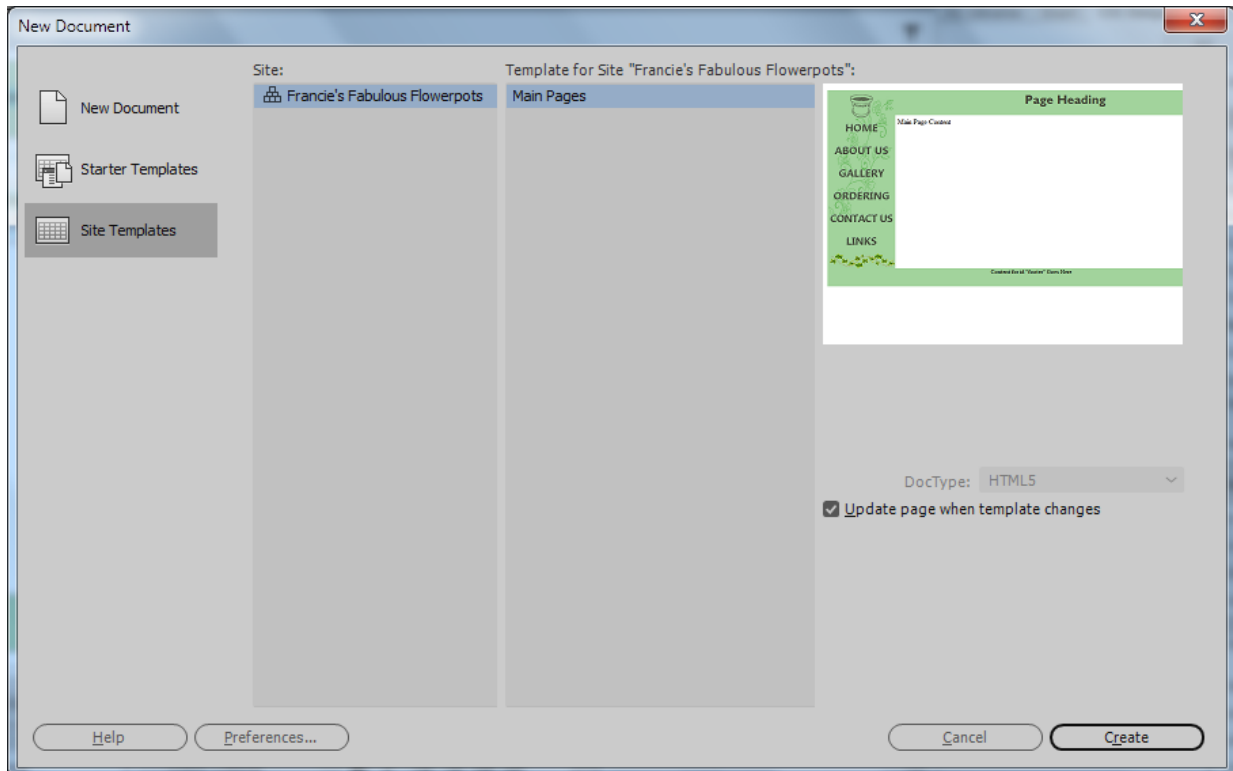


- 11) Save changes to the document.

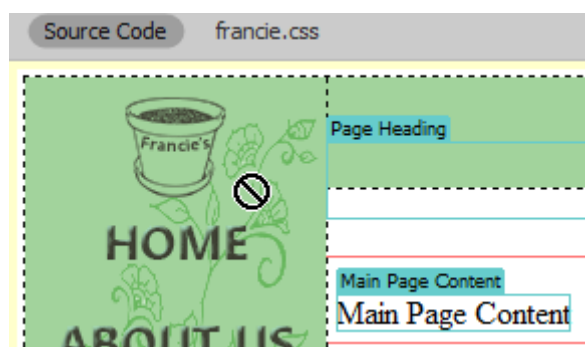
Exercise 7 – Using the Template

We have already created blank documents for the other pages to test our links, but we won't keep those ones. Instead, we'll create new pages using the template and then save the new pages over the top of the old ones.

- 1) From the **File** menu select **New**. [Ctrl] [N]
- 2) When the new file dialog appears, select the **Site Templates** option on the side.



- 3) Any Dreamweaver sites you have created will be listed in the middle section. Make sure your *Francie's Flowerpots* site is selected and the *Main Pages* template is selected.
- 4) Click **Create** to make a new page based on the template.
- 5) Move your mouse outside the editable areas and a cross will appear on your mouse.



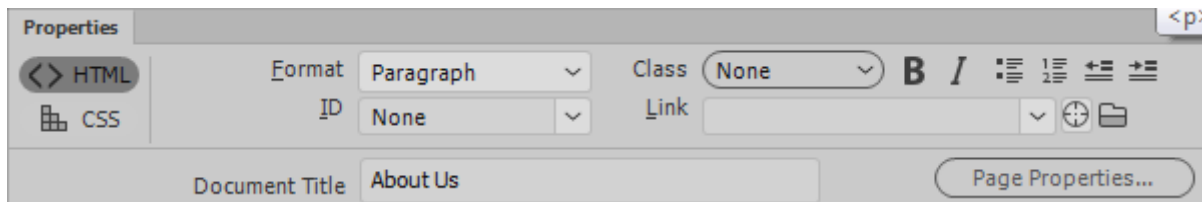
- 6) Click in the editable area for the heading. Delete the text that is there and type *About Us*.
- 7) Click in the *Main* editable area. Replace that text with the following.

Francie's Flowerpots is a business located in Western Australia. All of our pots are handmade using quality materials and techniques. We deliver to anywhere in the world. A sample of our range of pots can be viewed from the Gallery page. Other types of pots or custom made pots are available on request. Visit our Contact page to request a complete catalogue or more information about what's in the gallery.

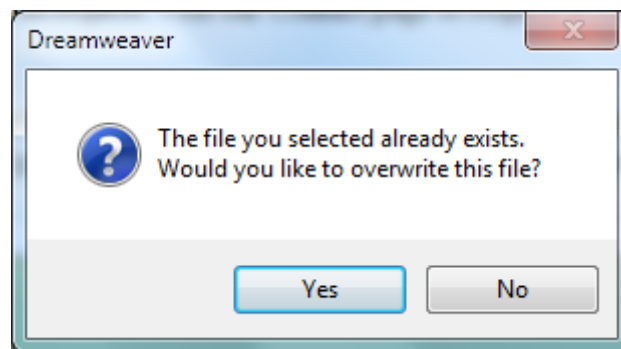
(Press Enter to start a new paragraph here)

Francie began making her own pots in the early 90s and began selling them to interested friends as a hobby. Since then the business has grown and we now have a team of skilled potters working to please our customers. We're certain that we have just the right pot for you.

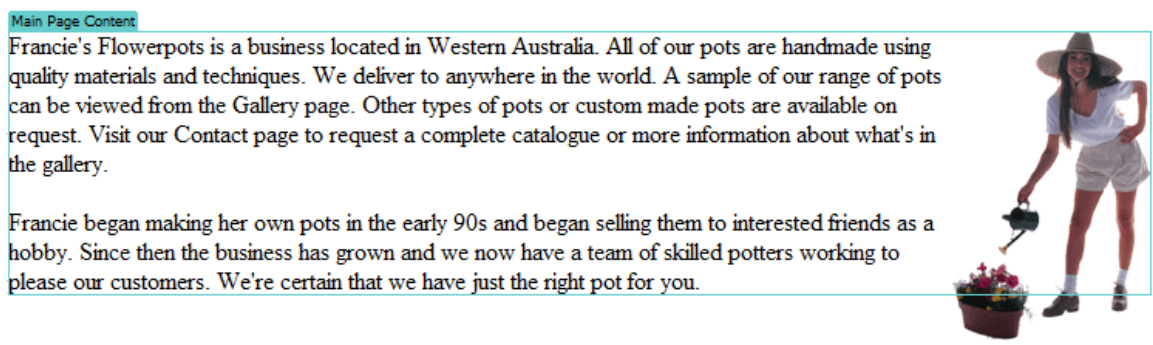
- 8) Change the **Title** of the page to *About Us*.



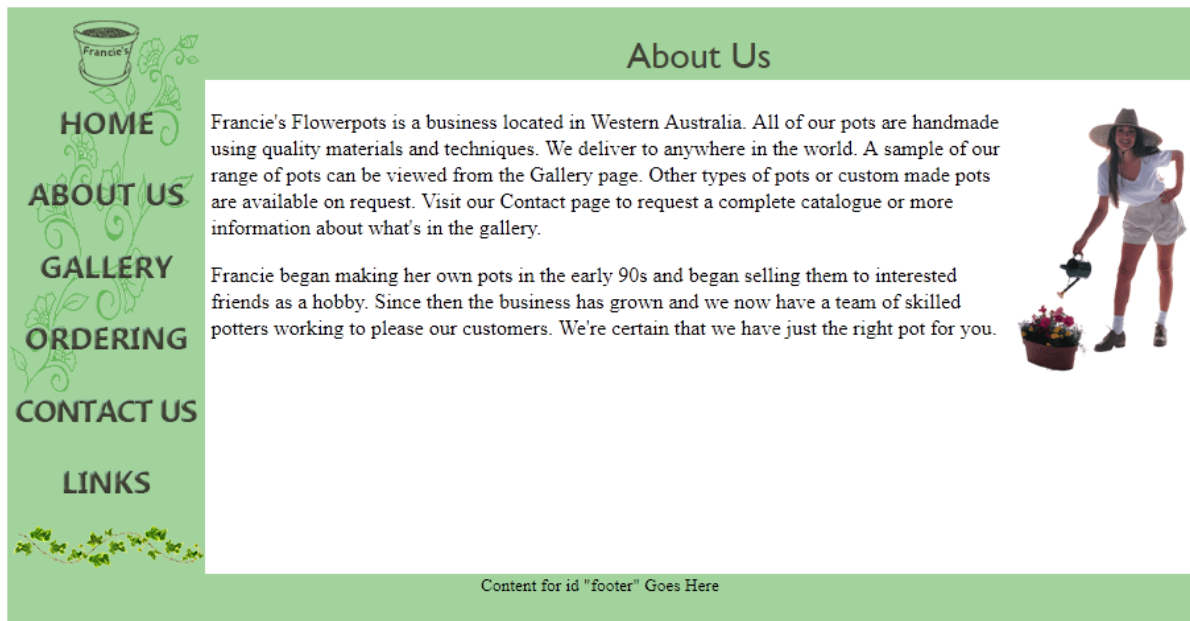
- 9) Save the page as *about.html*. When you are prompted to replace the existing file, click **Yes**.



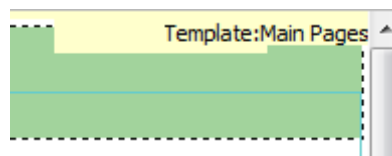
- 10) Click at the beginning of the first line (before the text *Francie's Flowerpots is a business*) and insert the image *about_francie.gif*.
- 11) For the alternate text type *About Francie*.
- 12) Right-click on the image and then select **Align** and **Right** so that it will be aligned to the right of the page.



- 13) Save and preview the document.



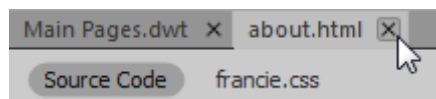
Note in the top right corner of the editing window, you might see a small tip indicating that this page is based on a template.



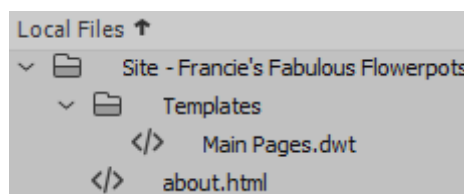
Exercise 8 – Updating the Template

When we make changes to a template, it will also change any pages based on that template.

- 1) Close the about.html file by clicking on the close button next to the filename or by pressing **[Ctrl] [W]**.



- 2) If you don't still have the template open, open it. Remember that the template is saved in a folder underneath your other files. If you have it showing in your file list you can double click it to open it. You can also use the recent files list under the **File menu**.



- 3) Delete the *Content for id "footer" Goes Here* placeholder text in the footer section.
- 4) We will add a small text navigation line similar to the one on the home page. Enter the text shown below.
- 5) About Us - Gallery - Ordering - Contact Us - Links
- 6) Click at the end of the line you typed and press **Shift Enter** to add a line break.

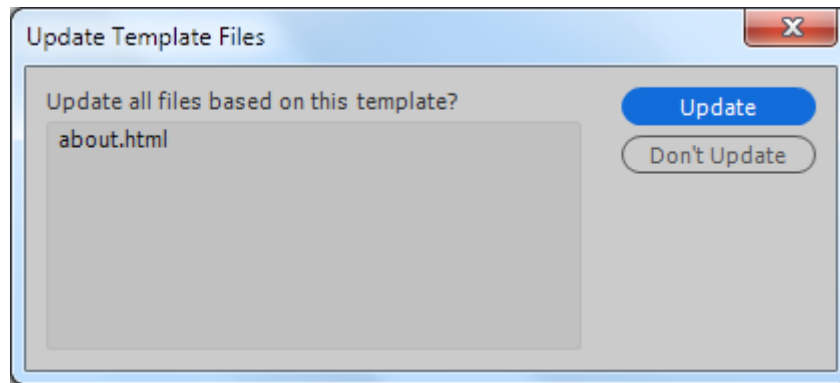
We want the line to start with a copyright symbol.

- 7) From the **Insert** menu select **HTML, Character** and **Copyright**.
- 8) After the copyright symbol type *Francie's Flowerpots* followed by the current year.
- 9) Create links to the appropriate pages so that it looks like the example below.

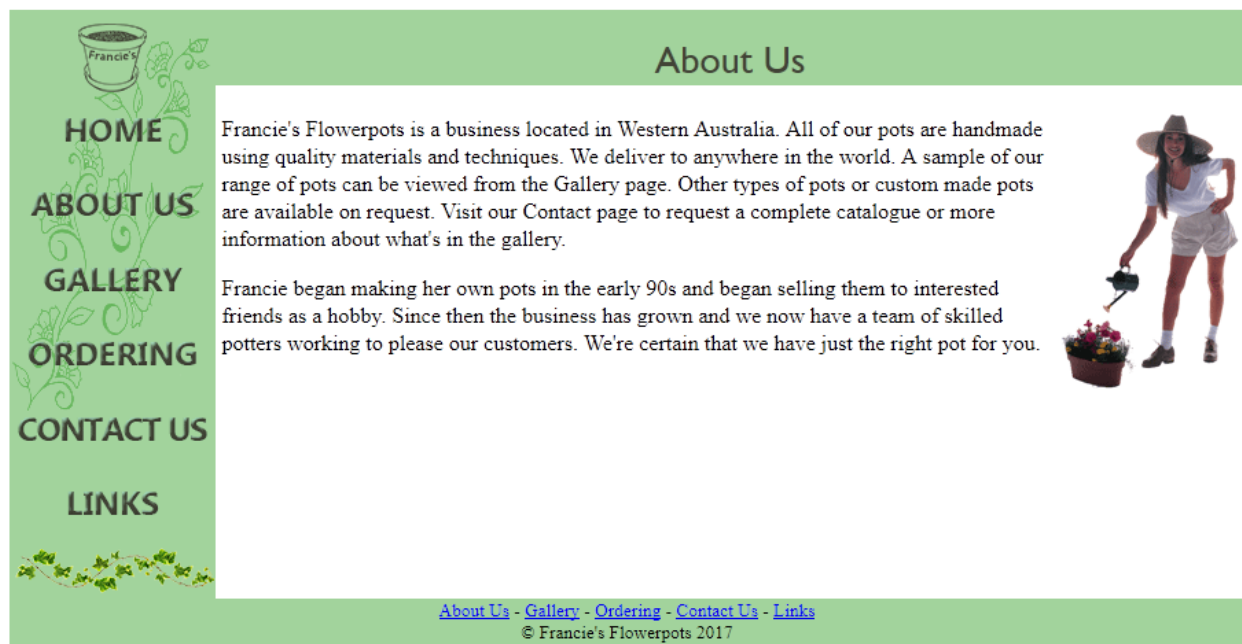
[About Us](#) - [Gallery](#) - [Ordering](#) - [Contact Us](#) - [Links](#)
© Francie's Flowerpots 2017

- 10) From the **File** menu select **Save All**.

Since you have modified the template, you will be asked if you want to update pages based upon that template.



- 4) Click the **Update** button to confirm. Another message will appear once the update is done. You can close that message.
- 5) Open the about.html page. You will see that the changes to the template have been applied to this page.
- 6) Preview the page in your web browser.



Shortcuts Learned

[Ctrl] [N]	Create a new document
[Ctrl] [Alt] [V]	Insert an editable region in to a template
[Ctrl] [W] or [Ctrl] [F4]	Close the current document
[Ctrl] [Q] or [Alt] [F4]	Exit Dreamweaver

Code Reference

By the time you complete this section your template's code should look similar to the examples below.

HTML

```

1  <!doctype html>
2  ▼ <html>
3  ▼ <head>
4    <meta charset="utf-8">
5    <!-- TemplateBeginEditable name="doctitle" -->
6    <title>Untitled Document</title>
7    <!-- TemplateEndEditable -->
8    <!-- TemplateBeginEditable name="head" -->
9    <!-- TemplateEndEditable -->
10   <link href="../francie.css" rel="stylesheet" type="text/css">
11   </head>
12
13 ▼ <body>
14 ▼ <main id="container">
15   <nav id="navigation"><a href="../index.html"></a><a href="../about.html"></a><a href="../gallery.html"></a><a
   href="../ordering.html"></a><a href="../contact.html"></a><a href="../links.html"></a></nav>
16 ▼ <header id="heading"><!-- TemplateBeginEditable name="Page Heading" -->
17   <h1>Page Heading</h1>
18   <!-- TemplateEndEditable --></header>
19   <section id="content"><!-- TemplateBeginEditable name="Main Page Content" -->Main Page
   Content<!-- TemplateEndEditable --></section>
20   <footer id="footer"><a href="../about.html">About Us</a> - <a
   href="../gallery.html">Gallery</a> - <a href="../ordering.html">Ordering</a> - <a
   href="../contact.html">Contact Us</a> - <a href="../links.html">Links</a><br>
21   &copy; Francie's Flowerpots 2017 </footer>
22 </main>
23 </body>
24 </html>

```

CSS

```
1  @charset "utf-8";
2  ▼ #container {
3      width: 900px;
4      height: 100%;
5      margin: 0px auto;
6  }
7  ▼ h1 {
8      color: #414336;
9      font: 28px "Gill Sans", "Gill Sans MT", "Myriad Pro", "DejaVu Sans Condensed",
      Helvetica, Arial, sans-serif;
10 }
11
12 ▼ #navigation {
13     width: 150px;
14     float: left;
15     background-color: #a2d39c;
16     background-image: url(nav_back.png);
17     background-repeat: no-repeat;
18 }
19 ▼ #heading {
20     width: 750px;
21     height: 55px;
22     float: left;
23     text-align: center;
24     background-color: #a2d39c;
25 }
26 ▼ #content {
27     width: 740px;
28     padding: 5px;
29     float: left;
30     background-color: white;
31 }
32 ▼ #footer {
33     width: 900px;
34     height: 40px;
35     clear: left;
36     font-size: smaller;
37     text-align: center;
38     background-color: #a2d39c;
39 }
```