



Using Dreamweaver

9 – Challenge

Now it's time to challenge the serious web developers among you. In this section we will create a website that will bring together skills learned in all of the previous exercises. In many sections, rather than tell you what to do, you will need to apply what you have learned previously to accomplish a certain result. If you're not sure how to do anything in this section then you have either skipped previous exercises or forgotten what you have learned in them. Either way, best to make sure your skills are up to scratch before you try to tackle what follows. For those who have easily done everything in the previous sections, here's a test of your skills.

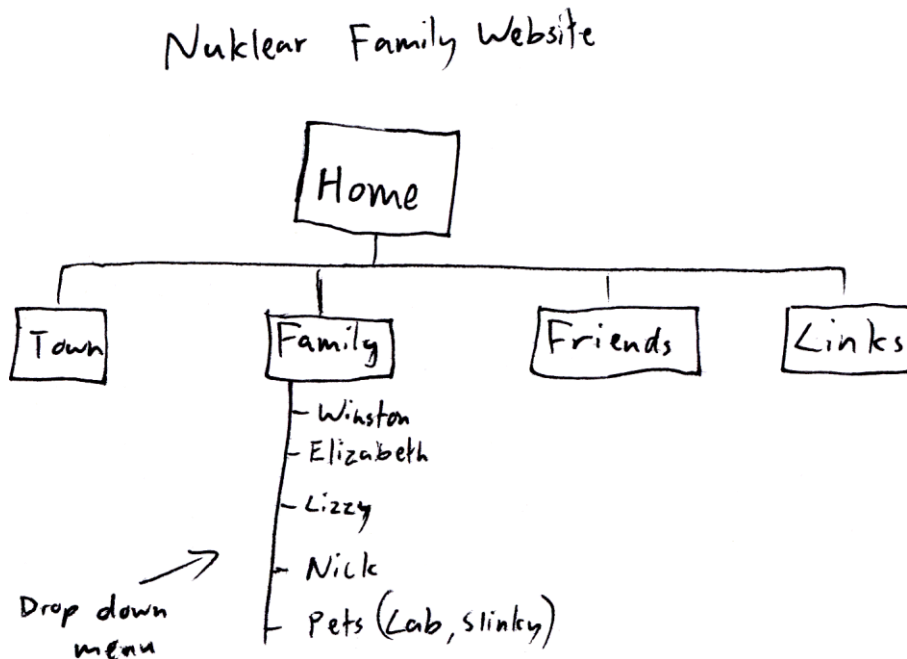


The Task

Your task in this section is to create a website for the Nuklear family. They want you to create a website to tell the world about them. Are you up to the task? Time to find out.

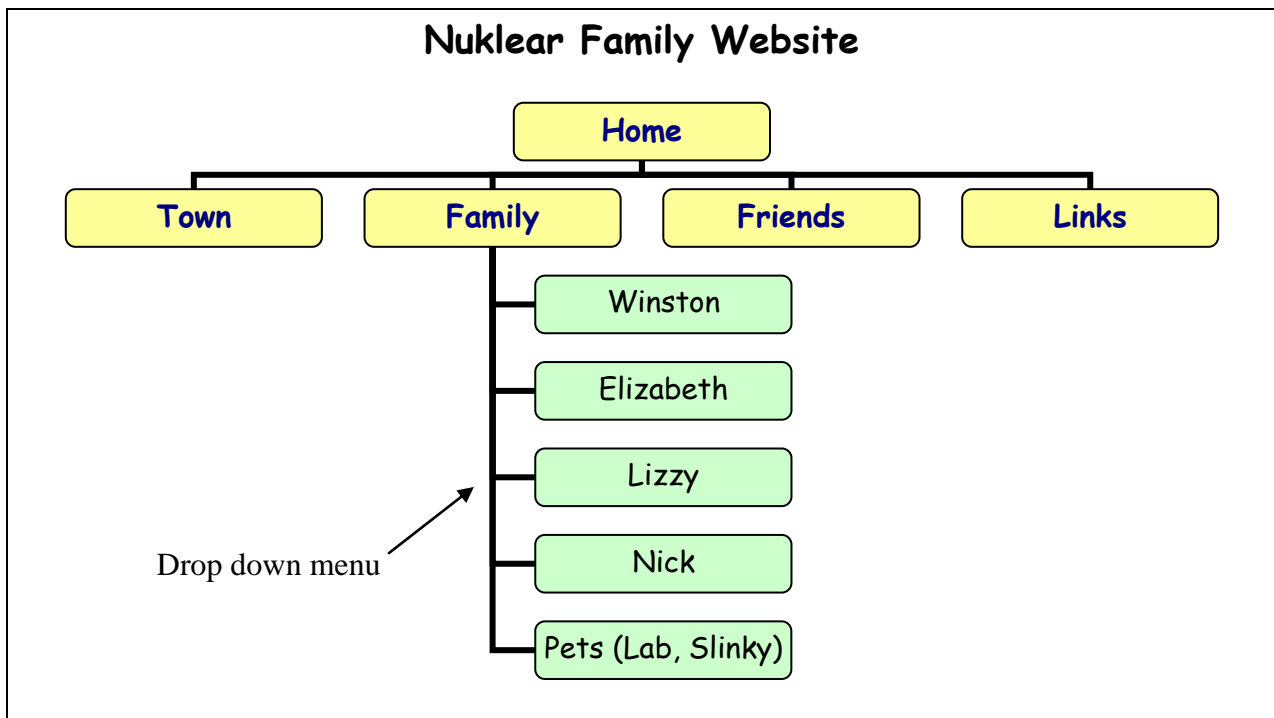
Planning

Most good websites begin with some good planning and this one will be no different. The Nuklear family have thought long and hard about what they want in their website and have given you a rough **Site Map** as shown below.

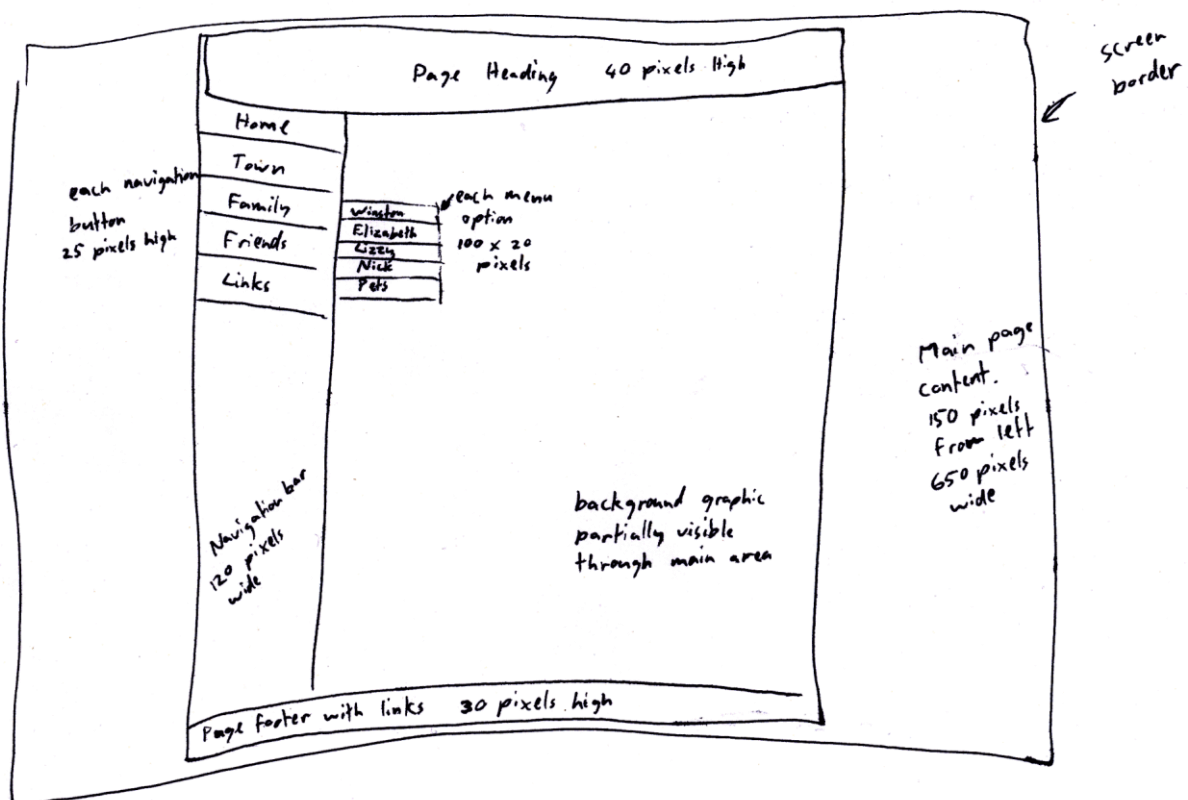


Of course, Nick Nuklear knows a thing or two about computers so he whipped up a slightly fancier version as shown on the following page. A website's planning can be anything from rough sketches to formal illustrations but the important thing is that it needs to give you a clear idea of what the site will contain.

Note Before you begin, make sure you have the images for these exercises saved in the location where you intend to create the Nuklear Family Website.



After deciding on the content of the site, the family decided on exactly how they wanted each page to look. A rough design plan for the home page is shown below. More detailed or neater plans for each page could be created and definately would be for a professional site. Note the precision of the size and position for certain elements on the page. Standard old HTML isn't capable of that level of precision, but CSS is. Since the majority of browsers now support CSS and its use is encouraged by the WC3, we will rely on it for our page formatting and layout. HTML for the structure of the page with CSS for the presentation (layout and formatting). Content and presentation separate as it should be.

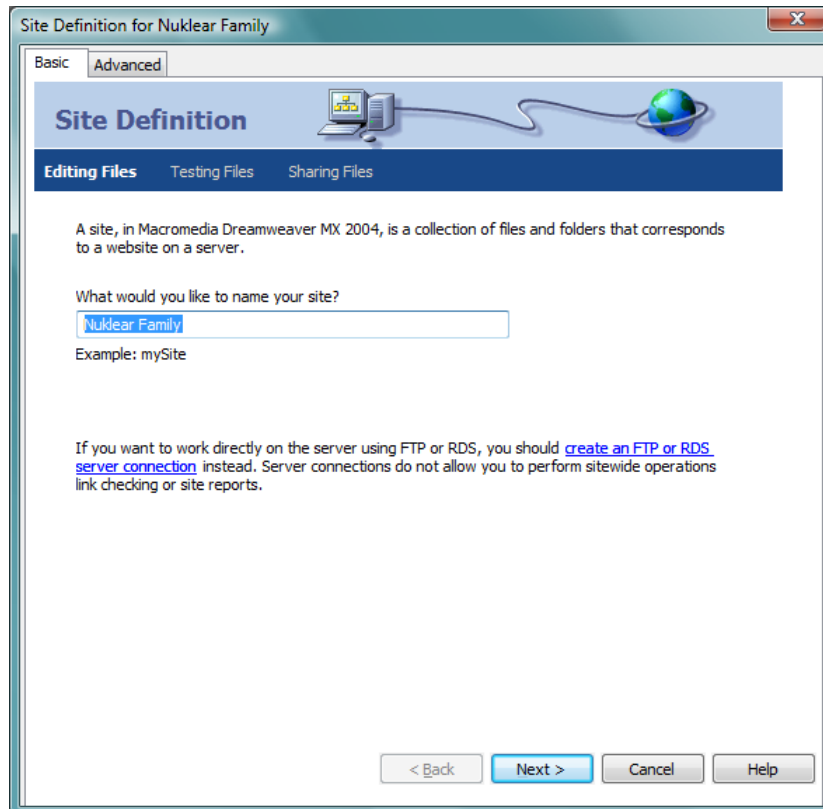


Note The level of planning shown here should be considered to be a minimum amount for a good website. Most professional sites would require far more detailed plans.

Exercise 1 – Create the Site

Assuming all the site planning is complete, we can now begin to create the site itself in dreamweaver.

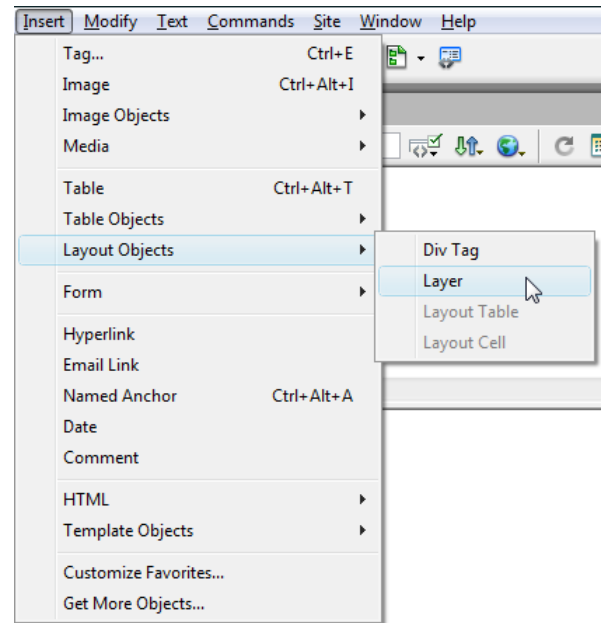
- 1) Create a new site, selecting a suitable location for the files in the site to be saved in. Make sure the exercise files for this exercise (several images and one css file) are saved in the same location. Give the site a suitable name.



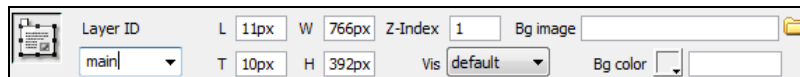
- 2) Once you have a site we will create a new HTML document that we will then use as the basis for our other pages (you could also create a template but we won't be creating many pages). Create a new document with the filename *index.html* and save it.
- 3) Set the title the page to *Nuklear Family Website*.

The layout of the page will be created by using layers. Layers make use of HTML's **DIV** (division) tag to create a section of the page which can then be positioned and formatted using CSS.

- 4) Insert a layer (or `ap_div` depending on dreamweaver version) on your page. This layer will be the main container for the page content. Resize the layer so that it is large enough to fit several layers inside it. The exact size doesn't matter since we will use CSS to determine exact size and positions later on, but for now it should take up most of your page area.



- 5) Make sure the layer is still selected. Change the layer name to *main*.



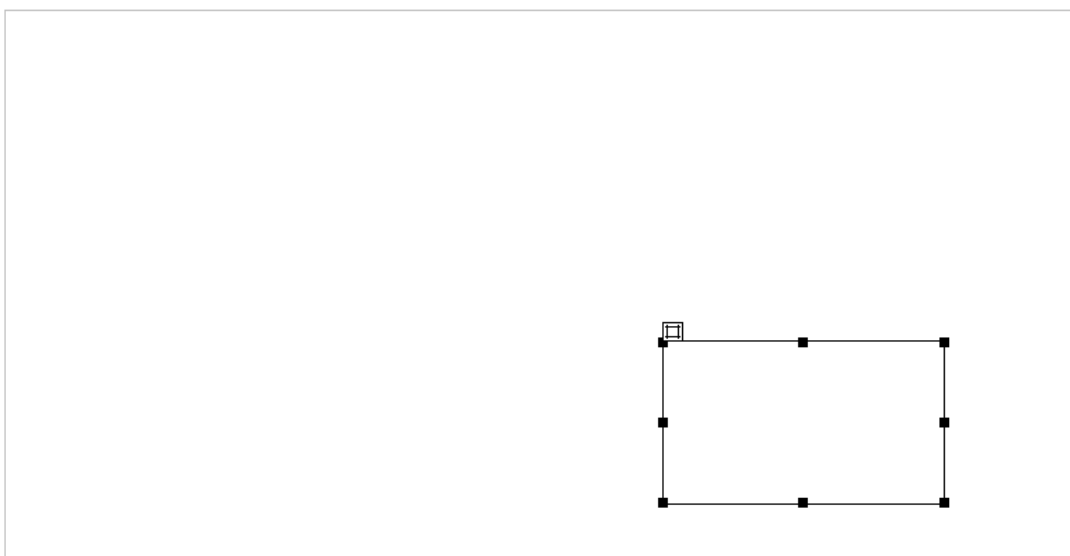
The HTML for the layer will look something like:

```
<div id="main" style="position:absolute; width:526px; height:264px; z-index:1"></div>
```

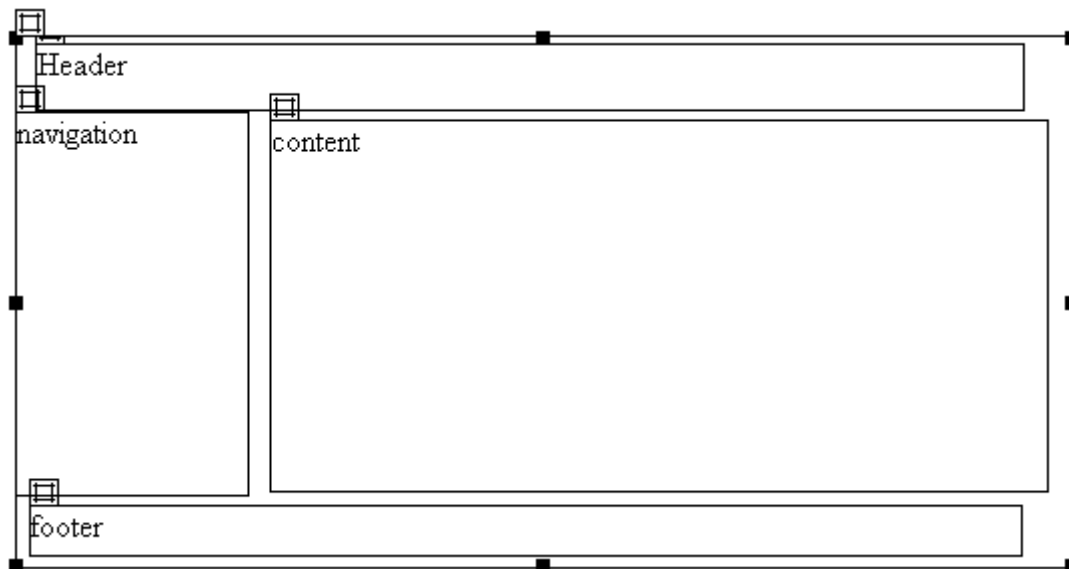
Or perhaps simply

```
<div id="main"></div>
```

- 6) Click inside the layer and add a few blank lines. These aren't essential but give us some room to work with. They can be deleted later.
- 7) Click inside the first line of the layer and add a new layer (this layer will be nested inside the main layer).
- 8) Drag that layer to the side (but still inside the main layer) so that you have room to create an additional layer.



- 9) Create 3 additional layers so that there are 4 in total. Each one should be inside the main layer. Name each of the layers *header*, *navigation*, *content* and *footer* (You could name them using the layer properties or simply change the **id** property in the **DIV** tag from code view). Remember that at this stage the size and position of the layers is not important though for ease of editing, you may want to arrange them to be in the approximate positions they will be once we are finished. Refer to the plan on page 2 or the example below. It could help to temporarily type the name of the layer in to each one to help identify them while you are editing. You can easily remove that text later. Make sure that each layer is inside the Main layer (click inside it first), but not inside one of the other layers before creating each layer.



Later on we will add additional layers for the menu but for now, the last layers we will add will be for the links within the navigation layer. These layers will all be nested within the navigation layer so make sure you click in that layer first before creating them.

- 10) Add 5 new layers within the navigation layer (you may want to add some blank lines first). Call the layers *home*, *town*, *family*, *friends* and *links*. Add some text within each layer as shown below.

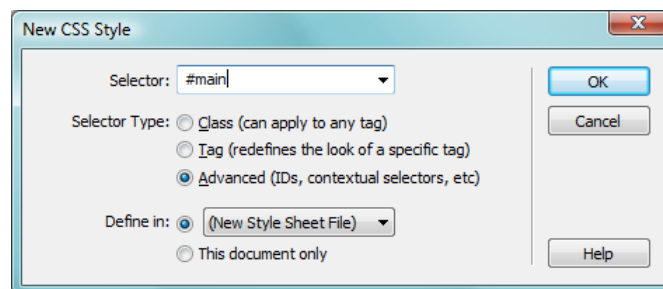


- 11) Save the changes when done.

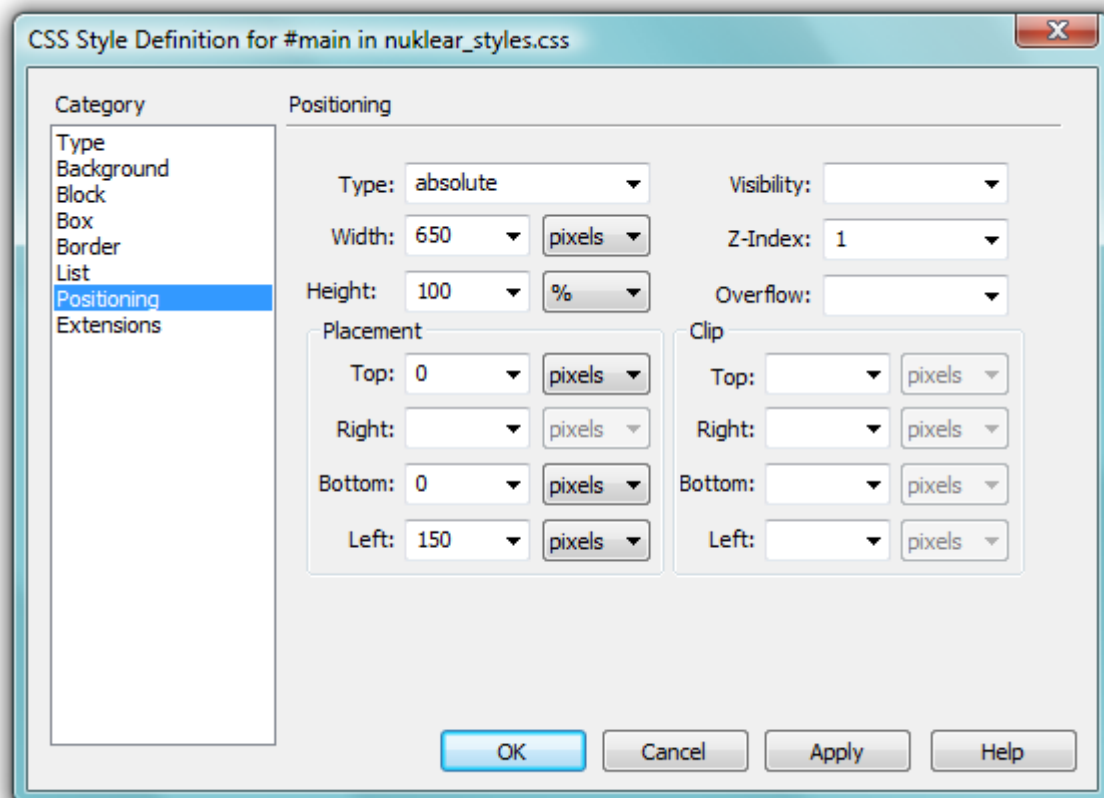
Exercise 2 – Creating a CSS File

We have now created a rough layout we can use as a template for our page but we have left all of the formatting and positioning up to CSS. Each of our layers has been identified by its *id*. When we create our style sheet, we will specify the formatting and positioning of each layer by referring to its *id*. Every page using this layout will have its layers formatted using the style rules in our CSS file so each page can have consistent formatting and positioning of its elements.

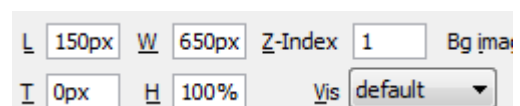
- 1) Select the **CSS Styles** tab.
- 2) Create a new style for main (an ID used for one of our layers) with the style defined in a new style sheet file. IDs are referenced in CSS by putting a # followed by the name of the style. The options should look like the example below.



- 3) Give the style sheet a suitable name (such as nuclear_styles.css).
- 4) Set the positioning properties for the main ID as shown below and click **OK** to apply the changes (650 pixels width, 150 from left and z-index set to 1).



It may be necessary to delete the positioning properties for the layer from the properties below, or any styles previously created by Dreamweaver, so they can be replaced with the properties from the styles you create.

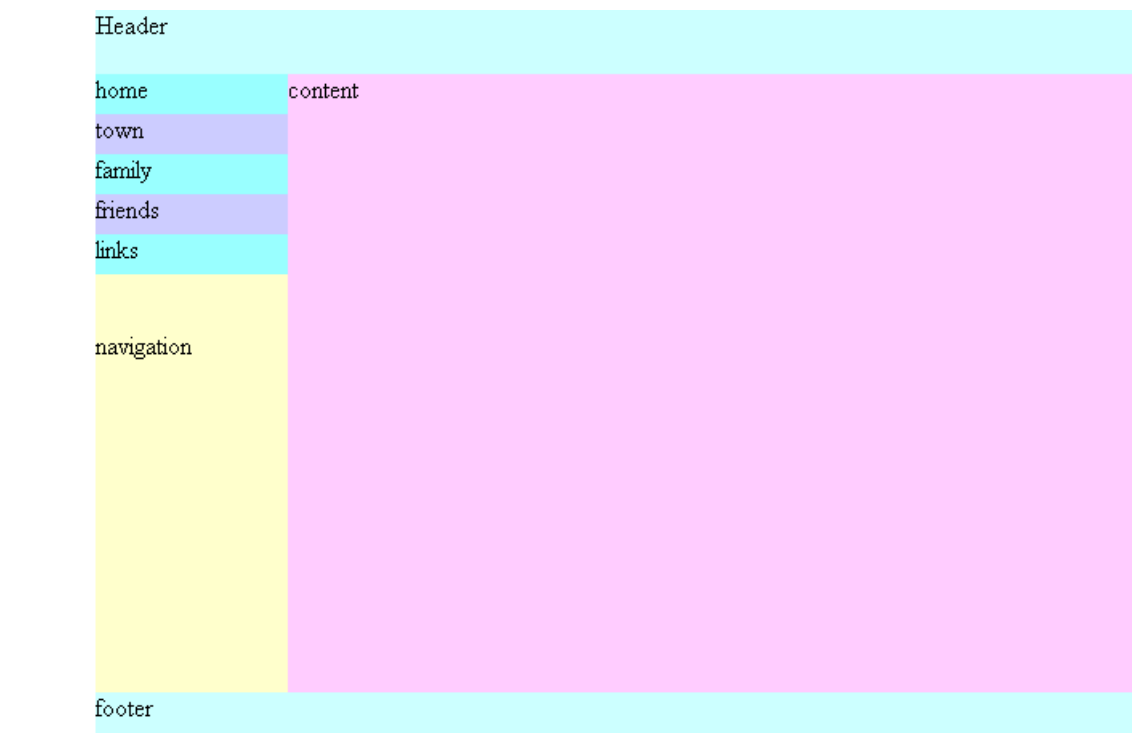


- 5) Create additional style ID classes for the remaining layers with the attributes listed below. Each should have positioning type set to **absolute**. In addition, the **Overflow** property of the *content* frame should be set to *auto* to allow for a scrollbar if there is too much content to fit in that area.

Tip for the time being, you may find it convenient to change the background colour of each **id** so you can see the effect of their positioning a little easier as you edit. This makes it easier to see where any problems with layout may be occurring.

| Name | Width | Height | Top | Right | Bottom | Left | Z-Index |
|-------------|-------|--------|-------|-------|--------|-------|---------|
| #header | 650px | 40px | 0 | | | 0 | 2 |
| #navigation | 120px | | 40px | | 30px | 0 | 3 |
| #content | 530px | | 40px | 0 | 30px | 120px | 3 |
| #footer | 650px | 30px | | | 0 | 0 | 4 |
| #home | 120px | 25px | 0 | | | 0 | 5 |
| #town | 120px | 25px | 25px | | | 0 | 5 |
| #family | 120px | 25px | 50px | | | 0 | 5 |
| #friends | 120px | 25px | 75px | | | 0 | 5 |
| #links | 120px | 25px | 100px | | | 0 | 5 |

- 6) Save and preview the page (use **File - Save all** since your css file will need to be saved also). It should look something like the example below (remember the colour is only temporary to make it easier to edit – you may prefer not to colour your layers).



Note If the content section looks small in the editor don't be concerned. When you preview it in the browser it will probably look the right size. The Dreamweaver editor will sometimes only show a layer as a small layer if it doesn't have much content. This is why it's important to preview your site in a browser as the editor doesn't always give you an accurate idea of how it will look.

Exercise 3 – Adding Some Content

Time to add some content to our page.

- 1) Delete the text in the heading layer and insert the image *heading_home.png*, adding appropriate **ALT** text for the image (such as *Nuklear Family Website*).
- 2) Change the background properties for the heading **ID** style so that the background of the layer matches the background of the header image.

Hint You can use the colour picker to sample the from the image on your page.



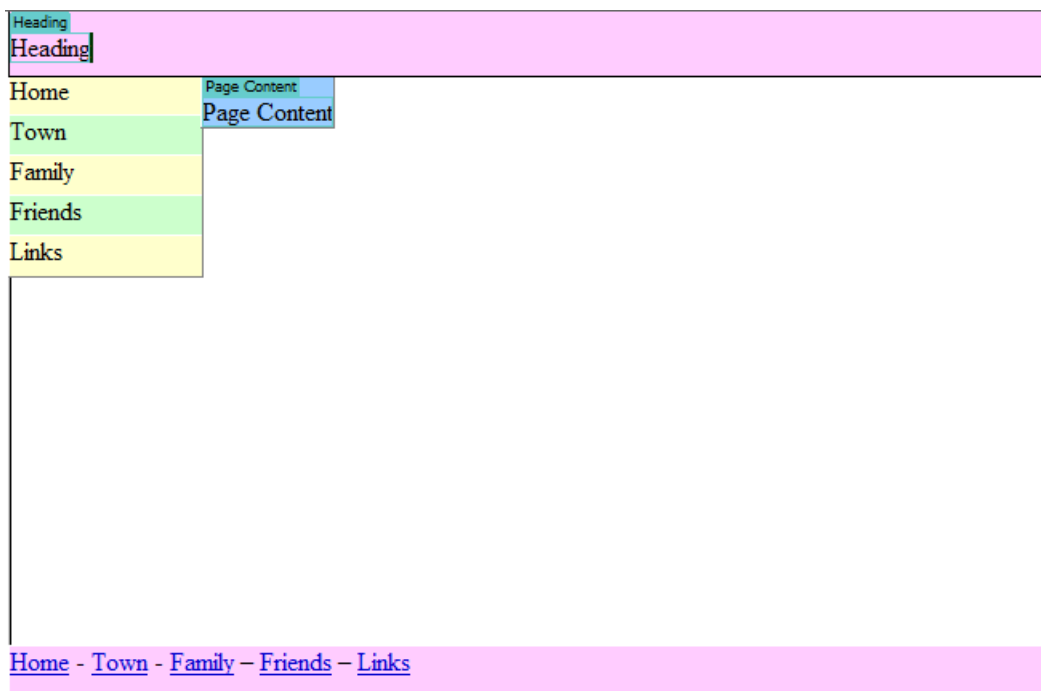
- 3) Replace the text in the footer layer with the row of text as shown below.

Home – Town – Family – Friends – Links

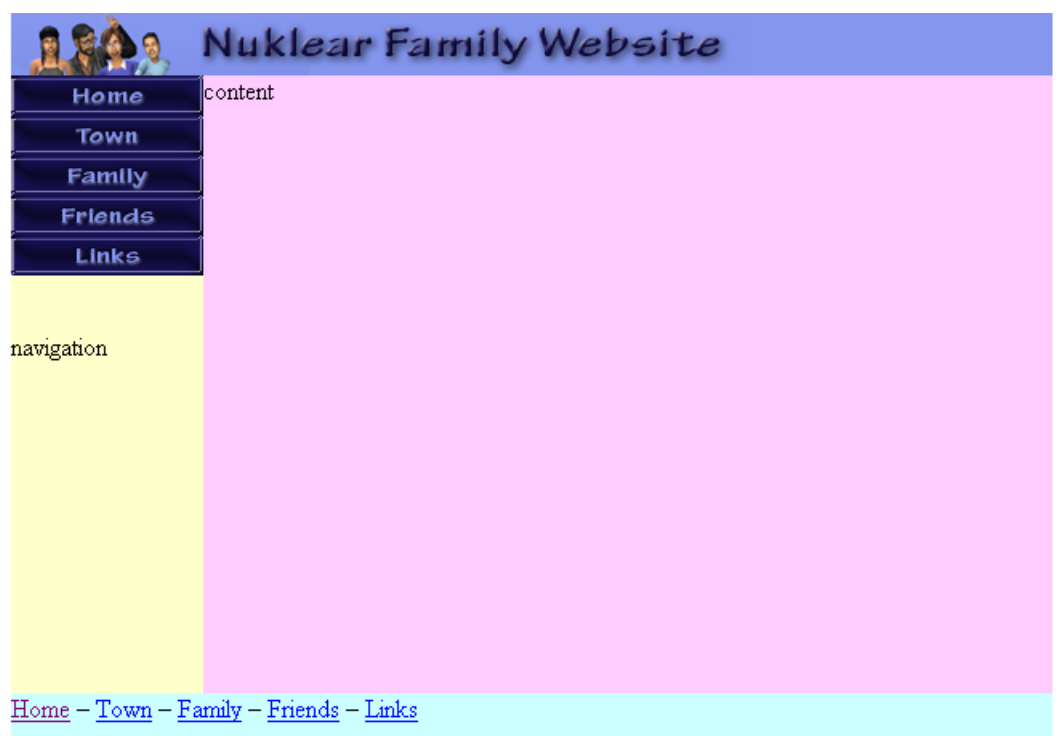
- 4) Each of the words will be a link to a page that we will create later based on this page. Select each 'link' and place the appropriate filename in the Link property to create the links.

| | |
|---------|--------------|
| Home | index.html |
| Town | town.html |
| Family | family.html |
| Friends | friends.html |
| Links | links.html |

- 5) Remember to save the file regularly.

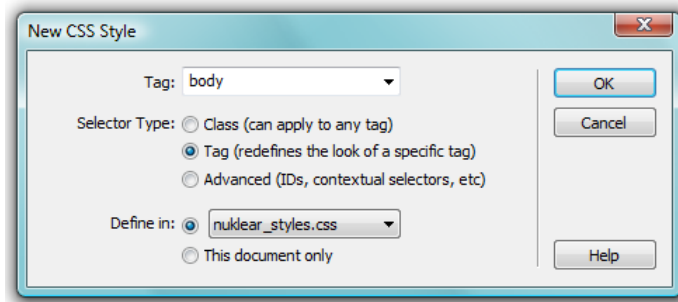


In each of the layers within the navigation layer, add one of the navigation images as appropriate, replacing the text in each layer. Remember to add appropriate **ALT** text for each image. When you save and preview the page it should look similar to the following example.

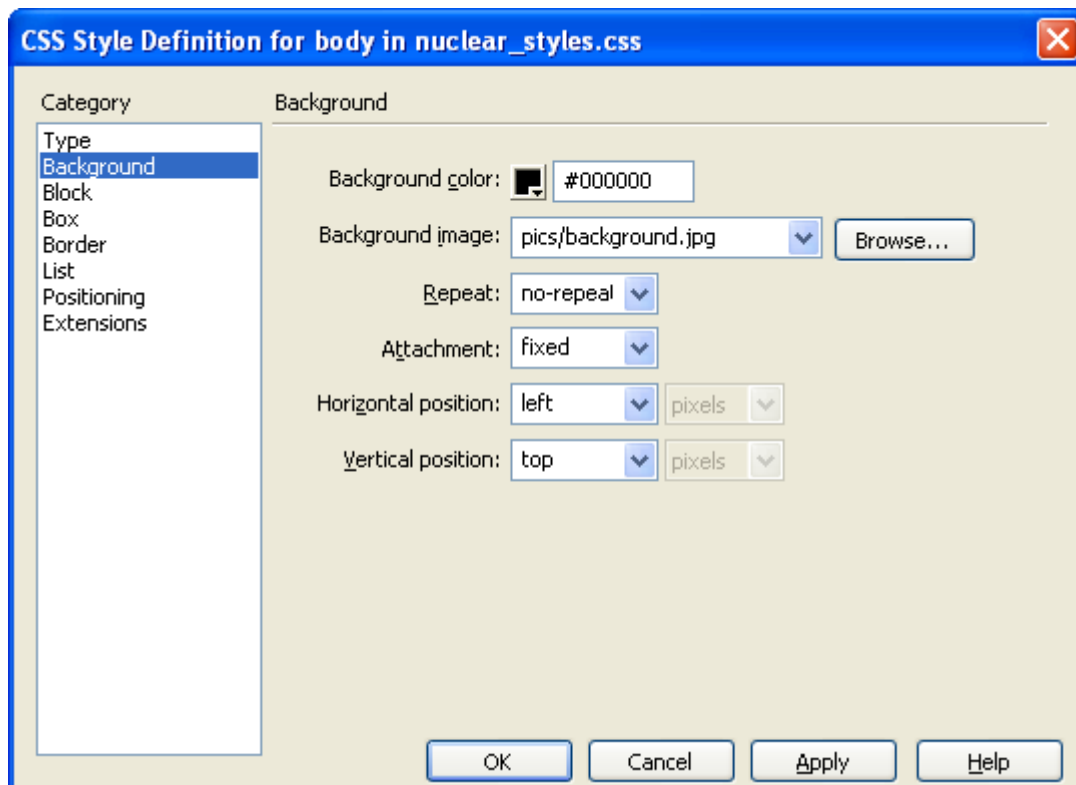


Exercise 4 – CSS Formatting

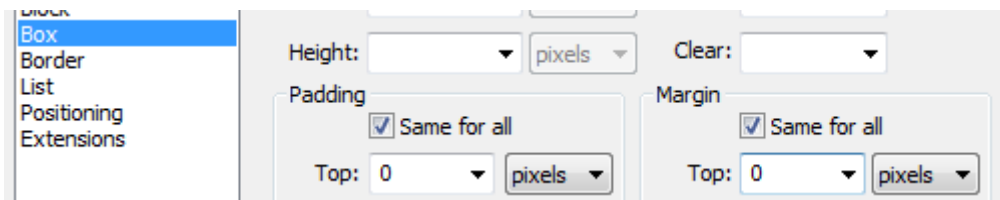
- 1) Create a new style rule in your **CSS** file for the *body* **HTML** tag.



- 2) Set the Background properties as shown



- 3) Set the Box settings so that there is no Padding or Margin and click **OK** to apply your new style rule.



Earlier we temporarily applied background colours to our layers to assist with editing but we will remove those later, so we will need to ensure that the dark coloured text in the content area can still be read on the dark coloured background. Traditionally, HTML only allowed for the use of **JPEG** and **GIF** files but recent browsers also recognise the use of **PNG** files for images which are becoming increasingly common. One advantage of a **PNG** file is that it allows for partial transparency. We will take advantage of this by setting the background of the content layer to use a **PNG** file that is partially transparent so that the background of the page will show through. The image we will use is 1 pixel in height and width so it will be a small file size even as it repeats to fill the area.

- 4) Change the background property of the content layer's style to remove any background colour you may have set earlier and set *content_background.png* as the background image.

This will create a background for the section that is easy to read text over but still shows the background of the page. This will become more obvious as we add content to that layer on each page that is created from this template.

Tip Some older web browsers such as Internet Explorer 6 support PNG files but don't support their transparency properties. This problem may be solved by adding a simple script to your page. For details and instructions go to <http://homepage.ntlworld.com/bobosola/>.

- 5) Change the background colour for footer style ID to the same colour you set for the header style ID.
- 6) Delete any extra text and blank lines from within the Navigation layer and change the background properties for the Navigation style rule so that it matches the following.

- 7) Preview the page and it should look similar to the example below



- 8) You can remove any background colours you added to your navigation layers. They are probably not visible behind the images but you might not want those colours to show while a visitor to your site waits for an image to load.


Tip You could change the layout of the entire site now simply by editing the styles. For instance, if you wanted the menu to be across the top of the page, all you'd have to do would be to edit the positioning information for the styles. That will change the layout for the entire site. Some sites even have more than one CSS file so that the layout and format can be quickly changed simply by changing the CSS file with a script on the page.

Exercise 5 – Modifying Formatting with CSS Rules

Time to establish some rules for formatting within our site.

1) Create some new style rules as described below.

a:link – type properties

Decoration: ☐ underline Color:  #0000FF


☐ overline

☐ line-through

☐ blink

☒ none

a:visited – type properties

Decoration: ☐ underline Color:  #660000


☐ overline

☐ line-through

☐ blink

☒ none

a:active – type properties

Decoration: ☐ underline Color:  #FF0000


☐ overline

☐ line-through

☐ blink

☒ none

a:hover – type properties

Decoration: ☐ underline Color:  #0000FF





☐ overline

☐ line-through

☐ blink

☒ none

a:hover – border properties

| | Style | Width | Color |
|---------|--|--|---|
| Top: | <input type="checkbox"/> Same for all solid | <input checked="" type="checkbox"/> Same for all 1 pixels | <input checked="" type="checkbox"/> Same for all  #FF0000 |
| Right: | none | 1 pixels |  #FF0000 |
| Bottom: | solid | 1 pixels |  #FF0000 |
| Left: | none | 1 pixels |  #FF0000 |

2) Edit the styles for the **footer** layer so that **font-size** is *smaller* and **text-align** is *center*.

3) Edit the styles for the **BODY** tag so that the following style rules are added:

font-family: "Comic Sans MS", Arial, sans-serif;

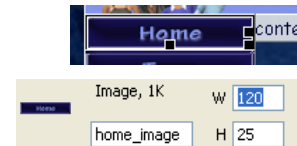
font-size: 13px;

color: #000033;

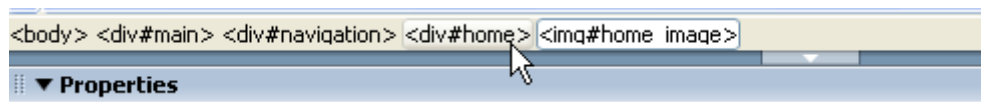
Tip You could add these directly by typing/pasting them in to the **BODY** section of the **CSS** file.

Exercise 6 – Adding a Menu

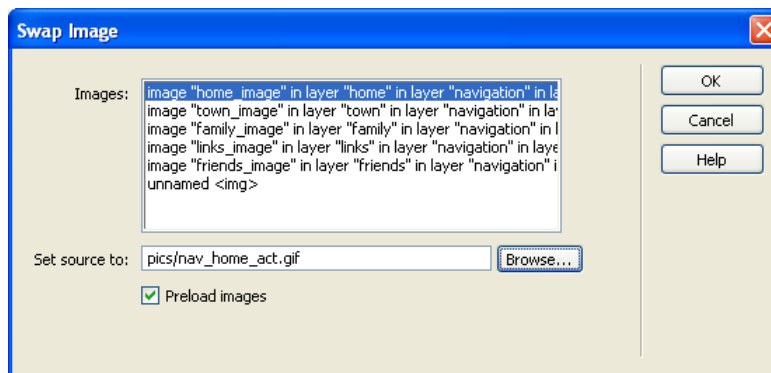
- 1) Select the Home image in the navigation area.
- 2) Set the **ID** property of the image to *home_image*.
- 3) Add appropriate IDs for each of the other images.
- 4) Select the home layer.



Tip Remember that you can select the image and then use the bar along the bottom of the editing area to select elements of the page that are around that image.

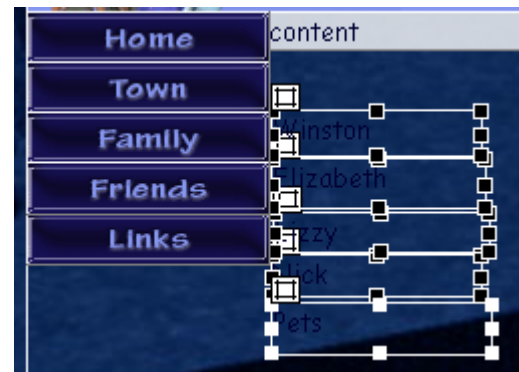


- 5) Display the **Behaviours** palette if it is not already showing.
- 6) Add a **swap image** behaviour so that *nav_home.gif* is swapped with *nav_home_act.gif* on mouse over.



- 7) Add similar behaviours for the other navigation layers (remember to add swap image restore behaviours for each one as well).
- 8) Add a **Go to URL** behaviour for the home layer so that it links to *index.html*.
- 9) Add similar behaviours for the other navigation.
- 10) Create new layers within the content layer for *Winston*, *Elizabeth*, *Lizzy*, *Nick* and *Pets* as shown below. Give each layer a suitable **ID** name.

Note They don't necessarily have to be inside the content layer but if you place them within a different layer you will need to adjust the **top** and **left** style properties to get them in the right spot.



- 11) Create style rules for each of the new layers as shown below.

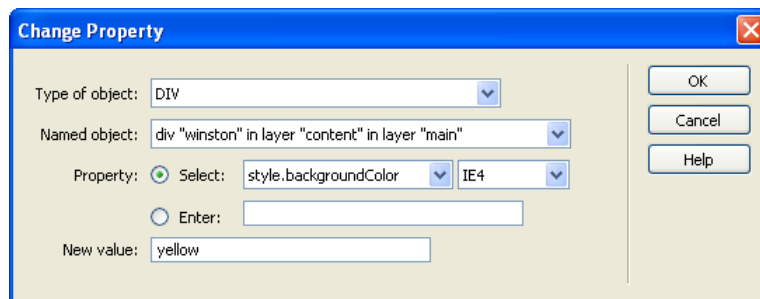
```
#winston
background-color: #8798F0;
border: 1px solid #000033;
position: absolute;
visibility: visible;
z-index: 7;
height: 18px;
width: 90px;
left: 0px;
top: 55px;
text-align: center;
```

```
#elizabeth
background-color: #8798F0;
border: 1px solid #000033;
position: absolute;
visibility: visible;
z-index: 7;
height: 18px;
width: 90px;
left: 0px;
top: 73px;
text-align: center;
```

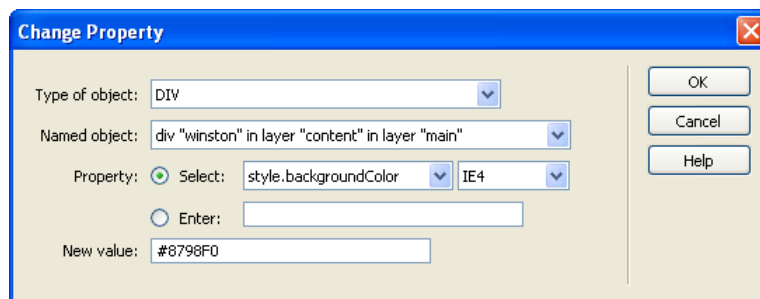
```
#lizzy
background-color: #8798F0;
border: 1px solid #000033;
position: absolute;
visibility: visible;
z-index: 7;
height: 18px;
width: 90px;
left: 0px;
top: 91px;
text-align: center;
```

| | |
|----------------------------|----------------------------|
| #nick | #pets |
| background-color: #8798F0; | background-color: #8798F0; |
| border: 1px solid #000033; | border: 1px solid #000033; |
| position: absolute; | position: absolute; |
| visibility: visible; | visibility: visible; |
| z-index: 7; | z-index: 7; |
| height: 18px; | height: 18px; |
| width: 90px; | width: 90px; |
| left: 0px; | left: 0px; |
| top: 109px; | top: 127px; |
| text-align: center; | text-align: center; |

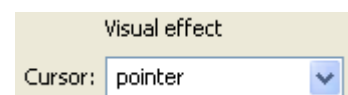
- 12) For each of the layer in the menu, create a **Go to URL** action so that when they are clicked, they will go to pages that will be created later. The files that each should go to are *winston.html*, *elizabeth.html*, *lizzy.html*, *nick.html* and *pets.html*.
- 13) For each of the menu layers, create a new **Change Property** behaviour so that the background colour changes to yellow on mouse over.



- 14) Create another **Change Property** behaviour for each layer so that the background colour changes back to its original colour on mouse out.

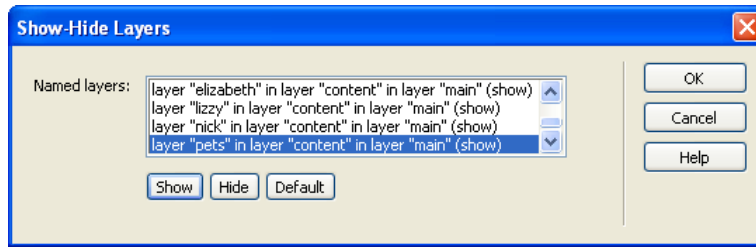


Tip You may want to add a style rule for each of the menu and navigation layers so that the mouse will change to a hand shape when it goes over the layer. Under the **style extensions** category, set the **Cursor** option to *pointer*.

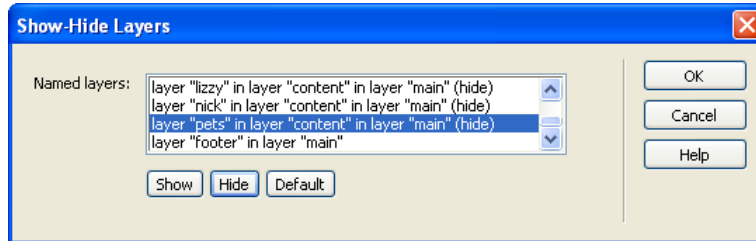


Now to finish the menu, we need to add some behaviours so that the menu will show when the mouse is over the Family layer or any of the menu layers. We will also need to add some behaviours so that the menu layers are hidden when the mouse goes off them.

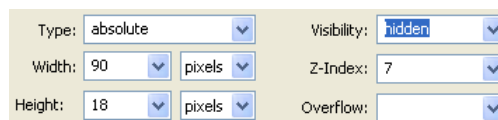
- 15) Add a **Show-hide Layers** behaviour for the *Family* layer that will show each of the layers in the menu (later we will change them so that they are not showing to start with). Make sure the behaviour is set to on mouse over.



- 16) Add a similar behaviour for each of the layers in the menu so that the menu layers will remain showing when the mouse is over each one of them.
- 17) Now for each of the Family and the menu layers, add a behaviour that will hide all of the menu layers on mouse out.



- 18) The last thing you need to do for the menu is to change the style properties for each of the menu layers so that they will be hidden when the page loads.



- 19) Save and preview your page to test the menu (remember that the links go to files that haven't been created yet but we're about to change that).



Exercise 7 – Adding more Pages

- 1) Save a copy of this page as *town.html*.
- 2) Change the **Title** of the page to *Nuklear Family's Town*.



- 3) Change the image in the header to *heading_town.png* and change the **ALT** text for the image.

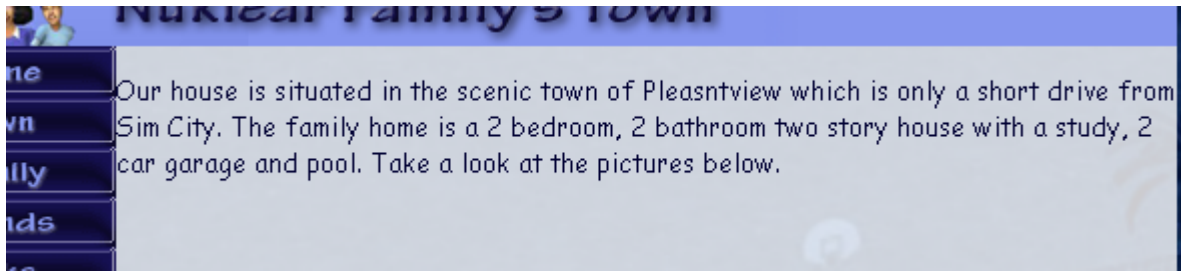


- 4) Make copies of this page to create *family.html*, *friends.html* and *links.html*.
- 5) Change the page title and header image in each.
- 6) Open the *family.html* page and save a copy of it as *winston.html*.
- 7) Change the page title to *Nuklear Family – Winston*.
- 8) In the content section, replace the existing text with *Winston* and make it a level 1 heading (be careful not to delete the menu layers if you have placed them in the content section).
- 9) Make copies of this file for *elizabeth.html*, *lizzy.html*, *nick.html* and *pets.html*. Change the page title and content in each one.
- 10) Add some content to the main page as shown below.

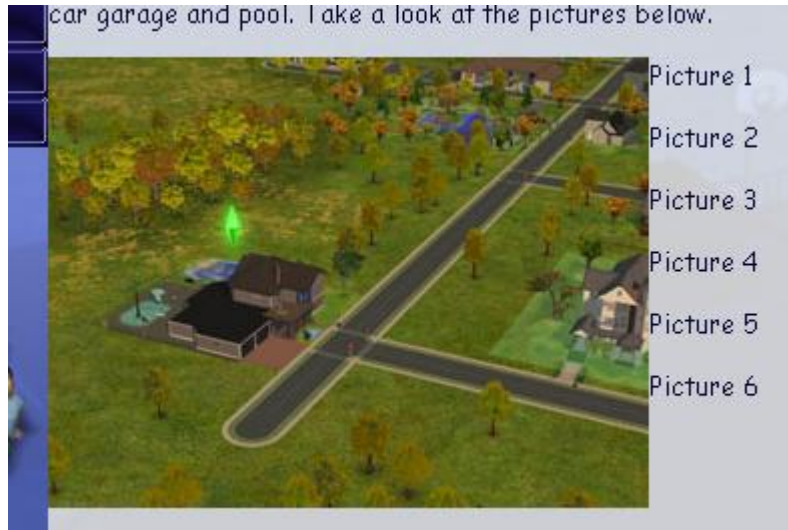


Image filename - main_family.png

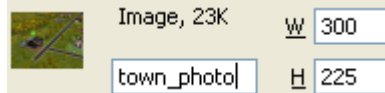
11) Add the following content to the Town page.



12) Add the image *town1.jpg* below the text as shown below along with the text next to the image (left align the image).



13) Give the image the ID *town_photo*.



For the next part we're going to add some HTML so make sure your code view is showing.

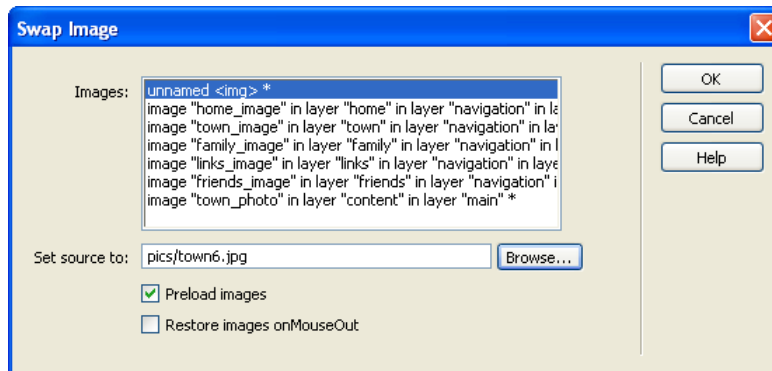
14) Edit the HTML to add anchor tags to the text next to the picture. We won't add any anchors though. The anchors will be used in a behaviour.

```
<p>
Picture 1 </p>
<p>Picture 2</p>
<p>Picture 3</p>
<p>Picture 4</p>
<p>Picture 5</p>
<p>Picture 6</p>
```

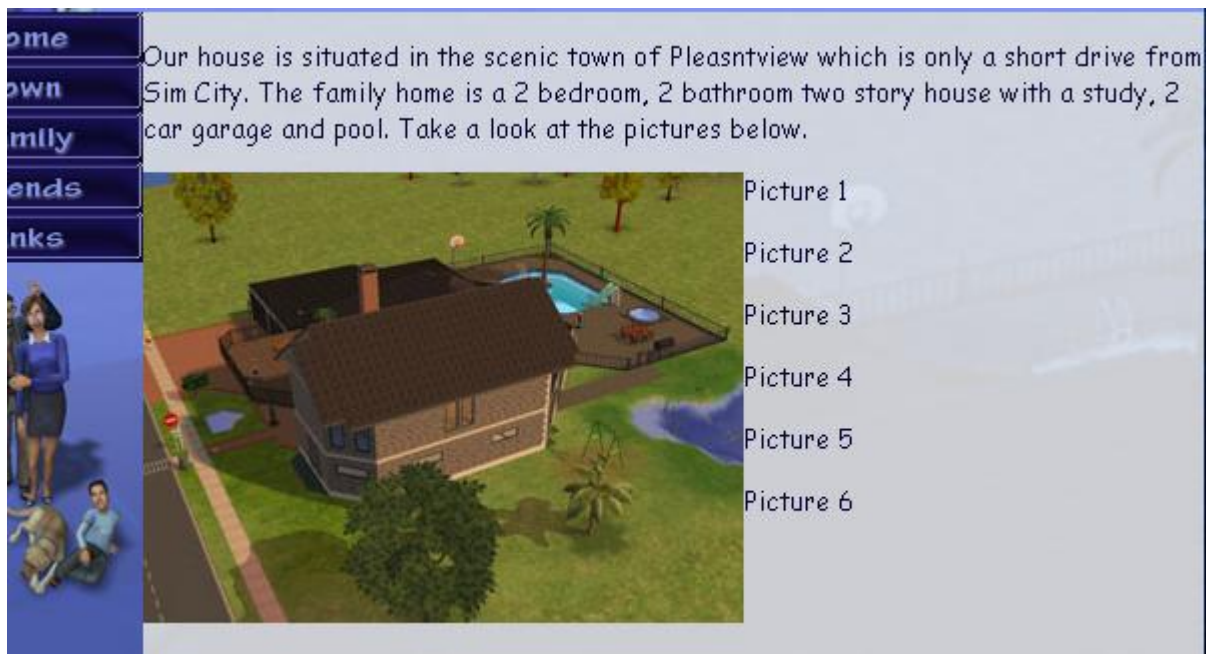
→

```
<p>
<a id="pic1">Picture 1</a></p>
<p><a id="pic2">Picture 2</a></p>
<p><a id="pic3">Picture 3</a></p>
<p><a id="pic4">Picture 4</a></p>
<p><a id="pic5">Picture 5</a></p>
<p><a id="pic6">Picture 6</a></p>
```

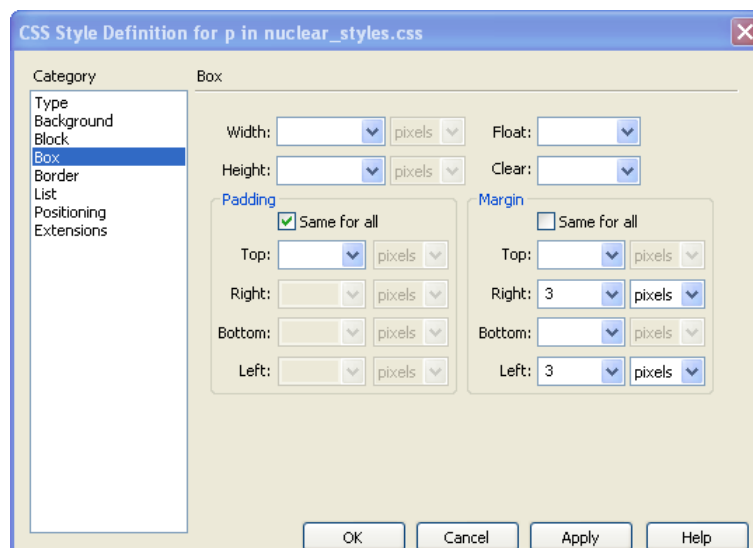
- 15) For the last link (Picture 6) select the text and create a **Swap Image** behaviour that changes the picture to *town6.jpg*. Set it to **onclick** and if an **on mouseout** behaviour has been created, remove it.



- 16) Add similar behaviours for each of the other links so that when each link is clicked, a different picture will be displayed. Save and preview to see the results.



Tip To avoid the text and pictures appearing squashed up next to the navigation area, you may want to add a style rule for your paragraphs **<P>** tag so that there is a bit of a left and right margin.



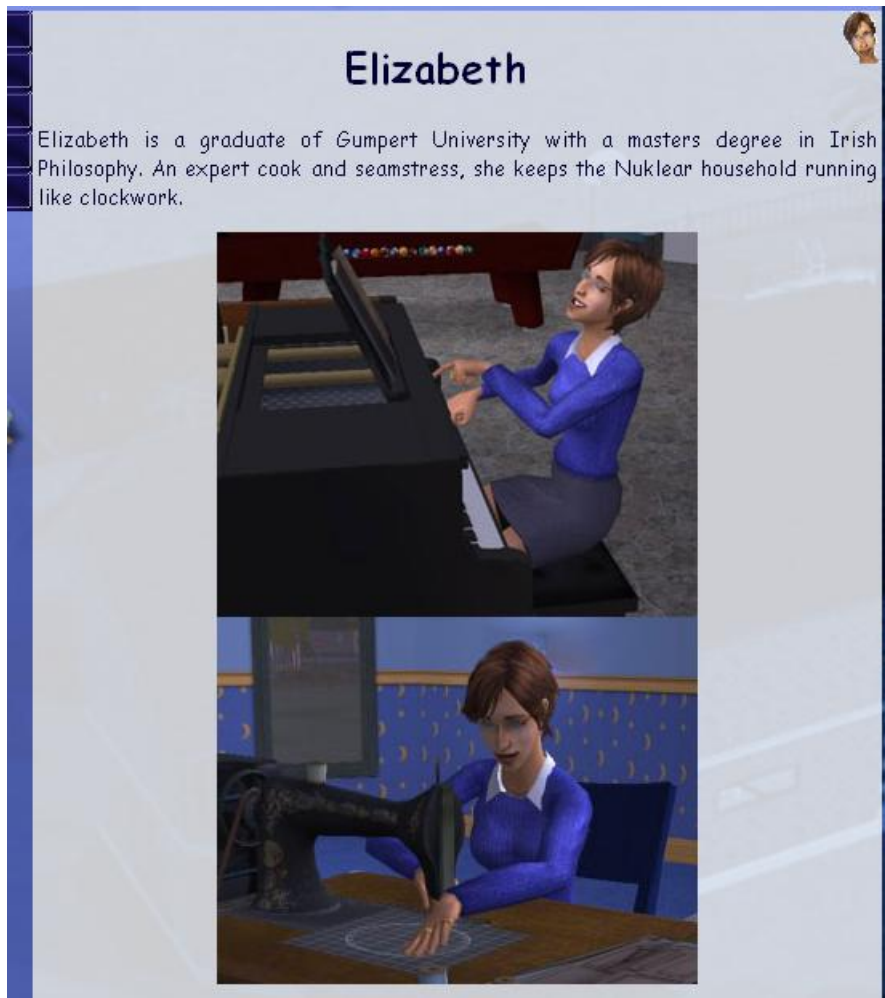
17) Edit the *family.html* page so that it looks like the following. Each family member's name should be a link to their page. The images to use are *icon_winston.png*, *icon_elizabeth.png*, *icon_lizzy.png*, *icon_nick.png* and *icon_pets.png*.



18) Modify the file *winston.html* so it looks like the following (you will need to create a style rule for **H1** to make it look like the heading in the example and you will also need to change the style rule for paragraphs so that **text-align** in the **Block** category is justified).



19) Edit the page *elizabeth.html* so that it appears like the following example.



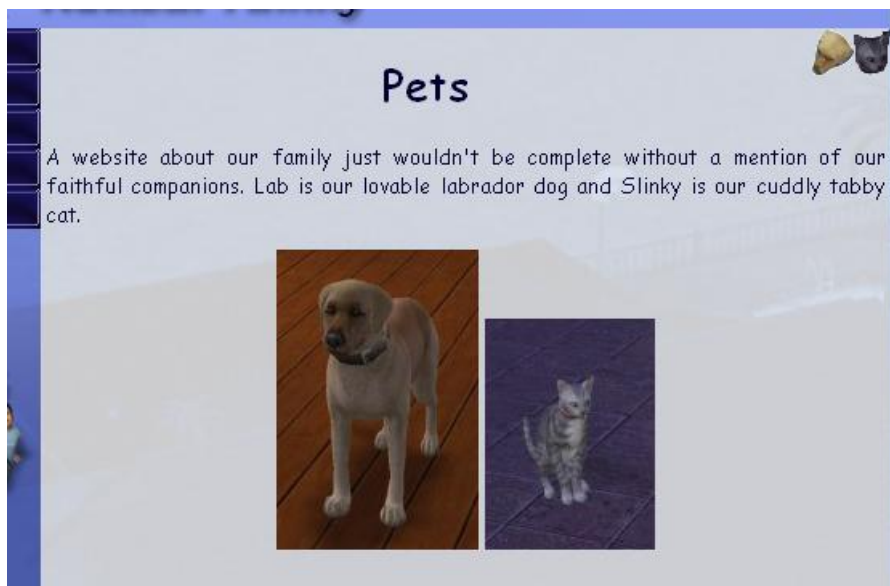
20) Modify *lizzy.html* so that it looks like the following.



21) Modify *nick.html* so that it looks like the following.



22) Modify *pets.html* so that it looks like the following.



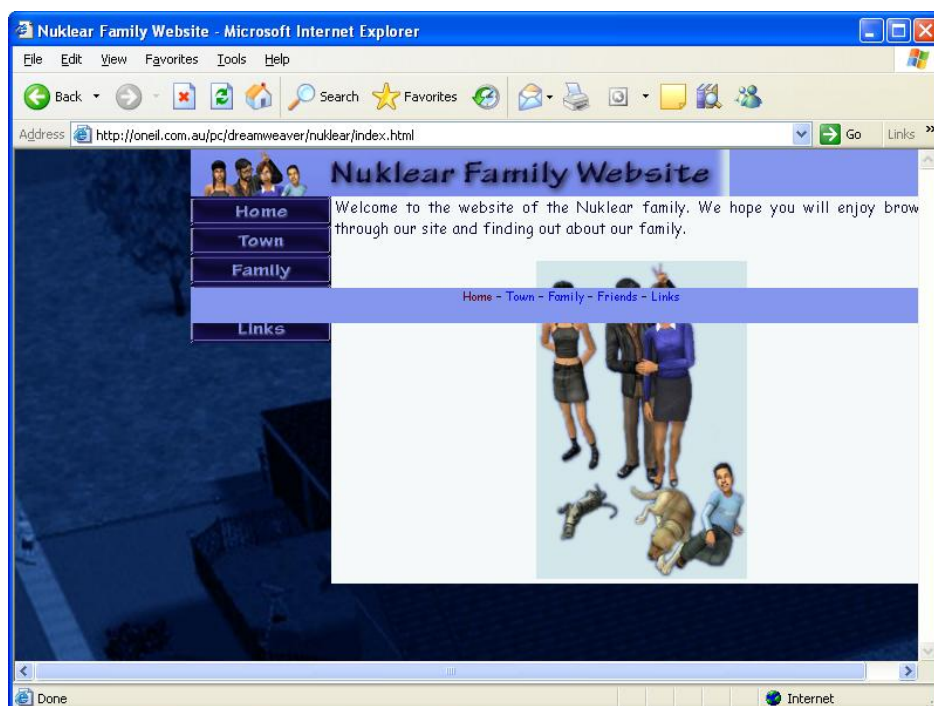
23) Edit the friends page to look similar to the following example. Add names and descriptions for the images (a table might be useful for layout).



24) Edit the links page to look like the following and add some of your own favourite links in the bullet points.



Note Some older browsers such as Internet Explorer 6 don't render styles correctly. When this happens some elements on the page might not appear as they are meant to. For example with our website, the bottom navigation may not appear at the bottom of the page in Internet Explorer 6 and may instead appear halfway down the page. You may need to make some adjustments to the styles to allow for this, such as changing the **z-index** attribute of that section to a lower number so that it appears beneath the main section rather than covering it and getting in the way. Below is the finished site as it would appear in Internet Explorer 6 without modification (note the lack of support for transparent PNG images).



Exercise 8 – Changing Style Sheets

Now to really demonstrate the power of styles, let's try using a different version of our stylesheet that has different attributes for some of our style rules.

- 1) From the exercise files, make sure that the file `alternate_styles.css` is saved in the same folder as your main website.
- 2) In your HTML code, locate the line that links to the stylesheet.

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

- 3) Change it so that it refers to the alternate version of the stylesheet.

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

Save and preview the page to see the differences. The only change we've made is referring to a different stylesheet but that alone has made significant changes to the formatting and layout of the page.

