Using Dreamweaver CS6

7 – Advanced Techniques

Dynamic HTML

Dynamic HTML (**DHTML**) is a term that refers to website that use a combination of **HTML**, scripting such as **JavaScript**, **CSS** and the **Document Object Model** (DOM). HTML and CSS are used to create the page, the DOM allows different elements of the page to be identified as objects and scripting is used to make changes to the page after it has loaded. This allows pages to be designed that can change depending on what the user does. For example, part of the page format or content may change when the user clicks a button or moves their mouse over something. A common use for this is to create DHTML menus for site navigation as we will be doing in this section.

When DHTML first began to be used in the late 90s, there were many problems since the creators of the two main browsers, Microsoft and Netscape implemented it in different ways. This caused many developers to be cautious about using it since it was difficult to use it in a way that would work in both browsers. In more recent years, browsers have generally become better about following standards so DHTML is far more likely to work properly in different browsers.

Dreamweaver takes a lot of the pain out of DHTML by allowing you to avoid having to write JavaScript code. We actually used a bit of DHTML with the Francies Flowerpots site when we set up the image swapping in the navigation bar. We will use Dreamweaver to create menus on the page by using a series of **layers**. Layers use either the **DIV** or **SPAN** tag in HTML (usually DIV since this produces better results in older browsers). When sections of the page are enclosed in a SPAN or DIV tag, then CSS can be used to position them anywhere on the screen with a great deal of precision. This can be used to create rectangular shaped areas on the screen that, when fit together, can form navigation menus. DHTML can then be used to hide or display these menus when needed.

Using layers for page layout

In the past, websites have often been designed using tables for page layout. Using layers with CSS enables you to position things on a page exactly without the need for tables. This technique will be used in the following exercise.

Note Positioning using styles won't be recognised in older web browsers that aren't compatible with CSS. Layers that are created using HTML's DIV tag will appear in older browsers as a series of paragraphs, one after the other so the order that they appear in the HTML code might be important if compatibility with older browsers is consideration for your site.

The Challenge

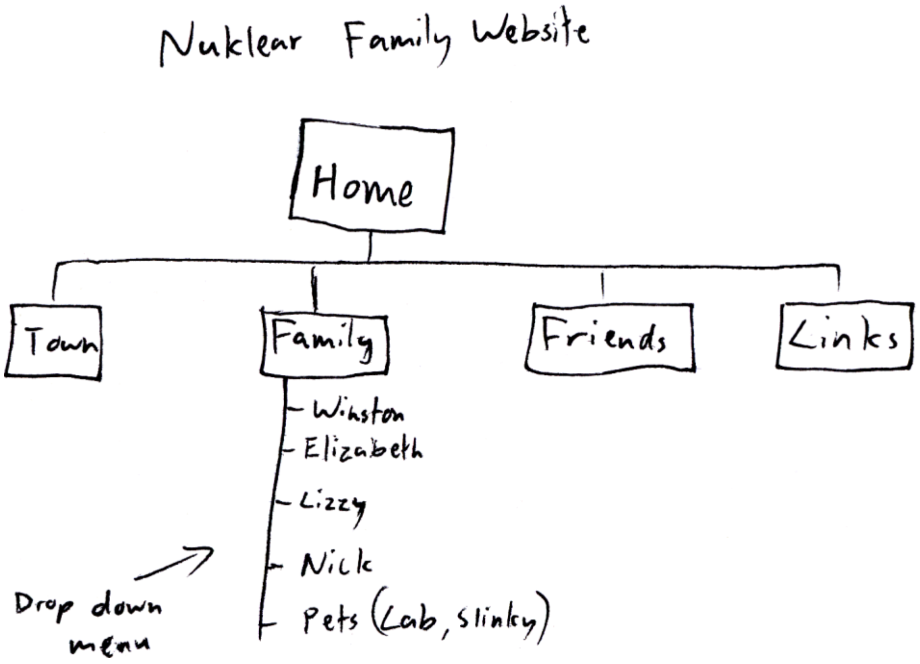
Now it's time to challenge the serious web developers among you. In this section we will create a website that will bring together skills learned in all of the previous exercises. In many sections, rather than tell you what to do, you will need to apply what you have learned previously to accomplish a certain result. If you're not sure how to do anything in this section then you may 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.

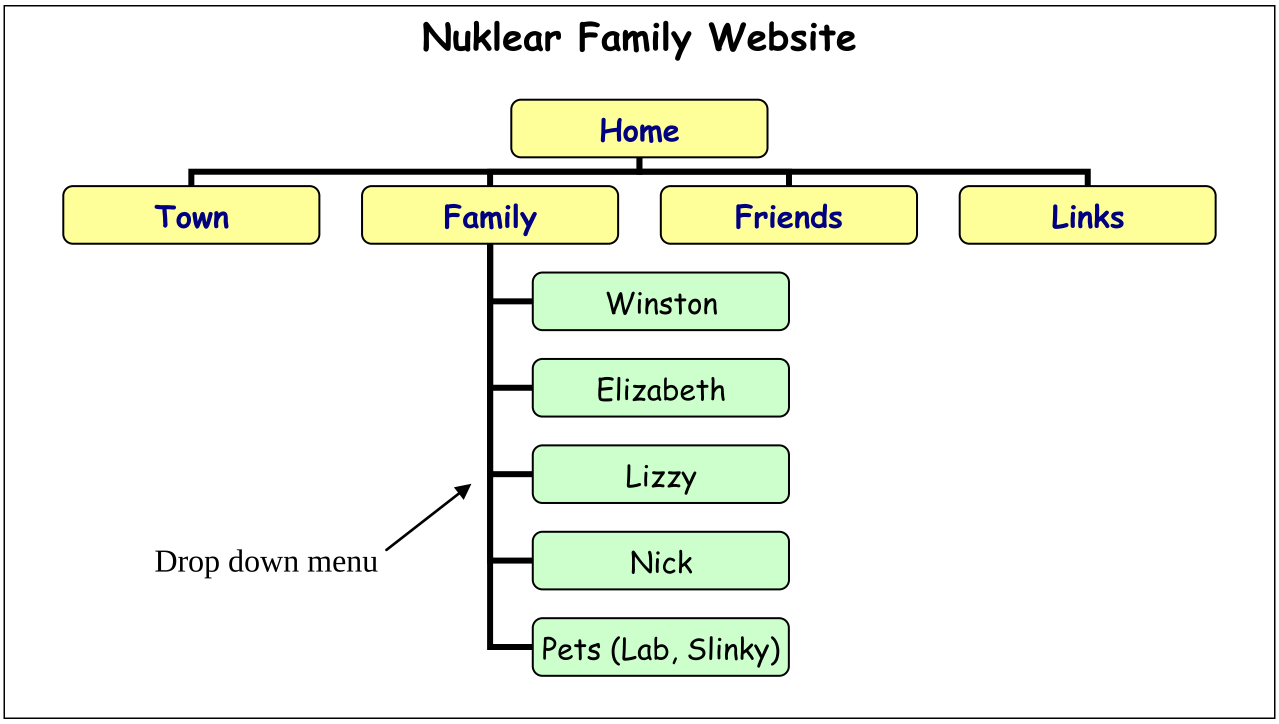
Planning

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.

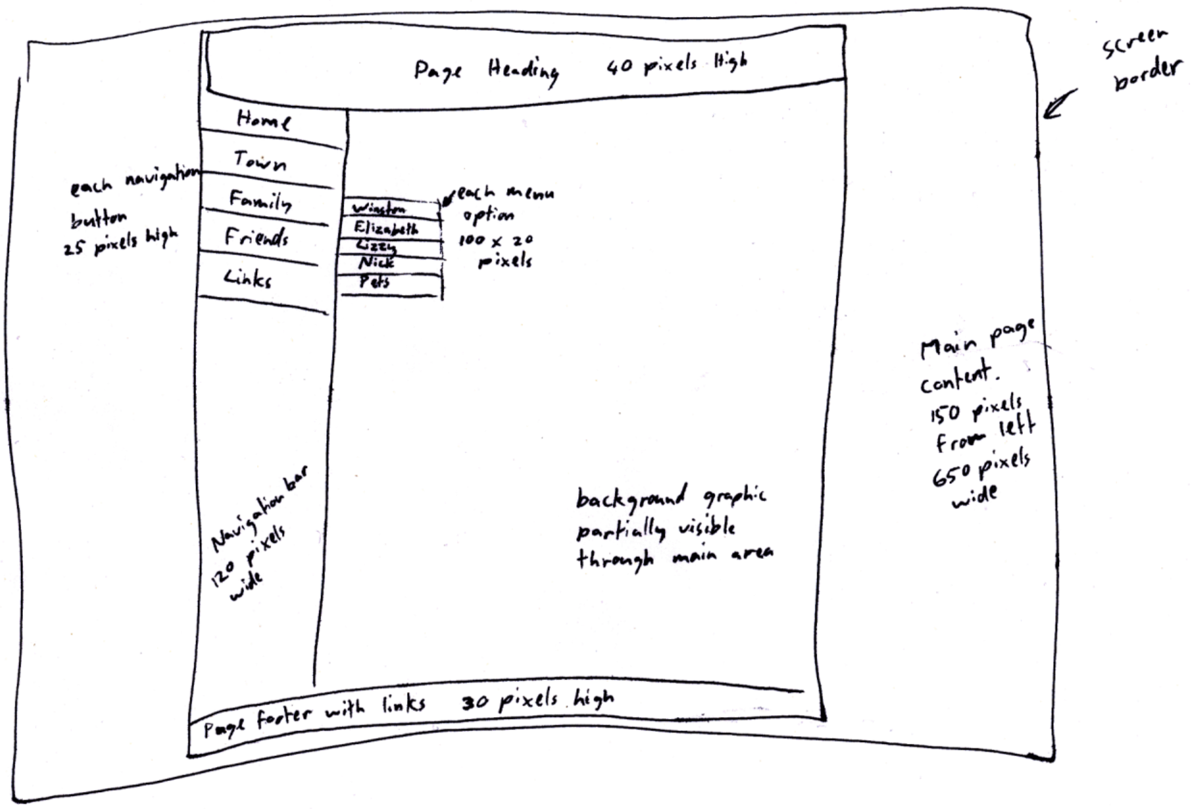


Of course, Nick Nuklear knows a thing or two about computers so he whipped up a slightly fancier version as shown on the following page (A website's planning can be anything from rough sketches to formal illustrations but the important thing is that it needs to give you a clear idea of what the site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

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 probably 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 and 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.

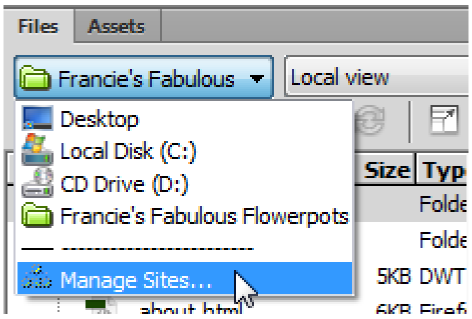

Page Heading 40 pixels High
screen border
Home
Town
Family
Friends
Links
each navigation button 25 pixels high
Winston
Elizabeth
Lizzy
Nick
Pets
each menu option 100 x 20 pixels
Main page content. 150 pixels from left 650 pixels wide
Navigation bar 120 pixels wide
background graphic partially visible through main area
Page footer with links 30 pixels high

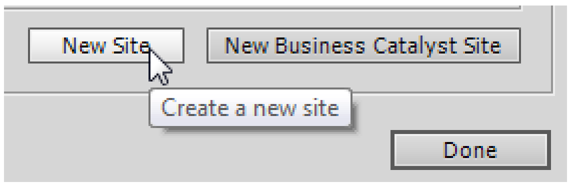
Note The level of planning shown here should be considered to be a <u>minimum</u> amount for a good website. Most professional sites would require far more detailed plans.

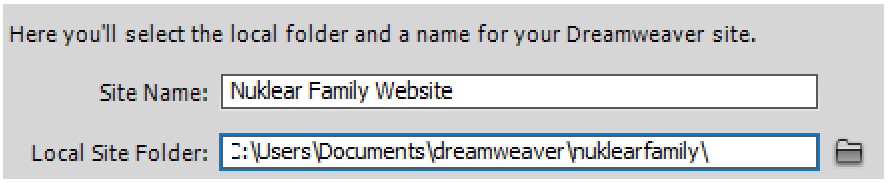
Exercise 1 – Create the Site

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 (a folder of images and one css file) are saved in the same location. Give the site a suitable name. If you can't remember how, the pictures below should give you some clues.



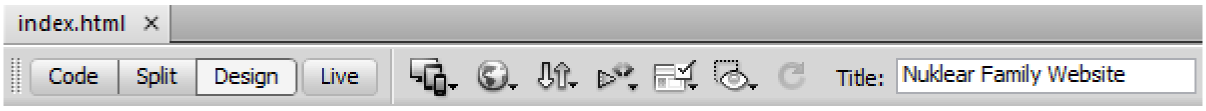



Here you'll select the local folder and a name for your Dreamweaver site.

Site Name: Nuklear Family Website

Local Site Folder: C:\Users\Documents\dreamweaver\nuklearfamily\

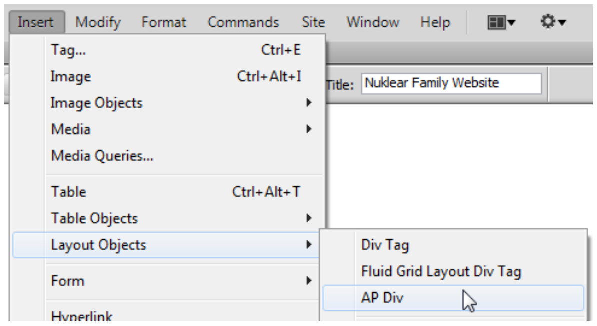
2) Once you have a site we will create a new HTML document that we will then use as the basis for our other pages (you could also create a template but we won't be creating many pages). Create a new document with the filename *index.html* and save it.
3) Set the title the page to *Nuklear Family Website*.



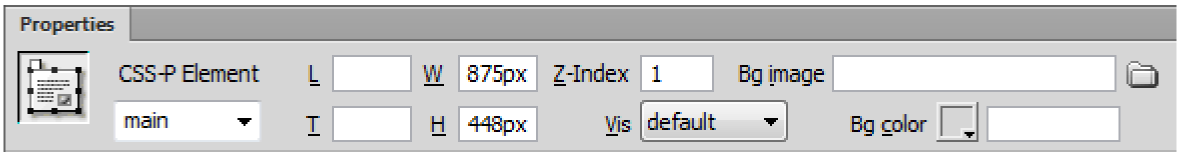
Title: Nuklear Family Website

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 DIV on your page. This will be the main container for the page content. Resize the DIV so that it is large enough to fit several layers inside it. You could use the rough plan for the page layout earlier on as a guide. The exact size doesn't matter since we will use CSS to determine exact size and positions later on.



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



CSS-P Element: main
L:
T:
W: 875px
H: 448px
Z-Index: 1
Vis: default
Bg image:
Bg color:

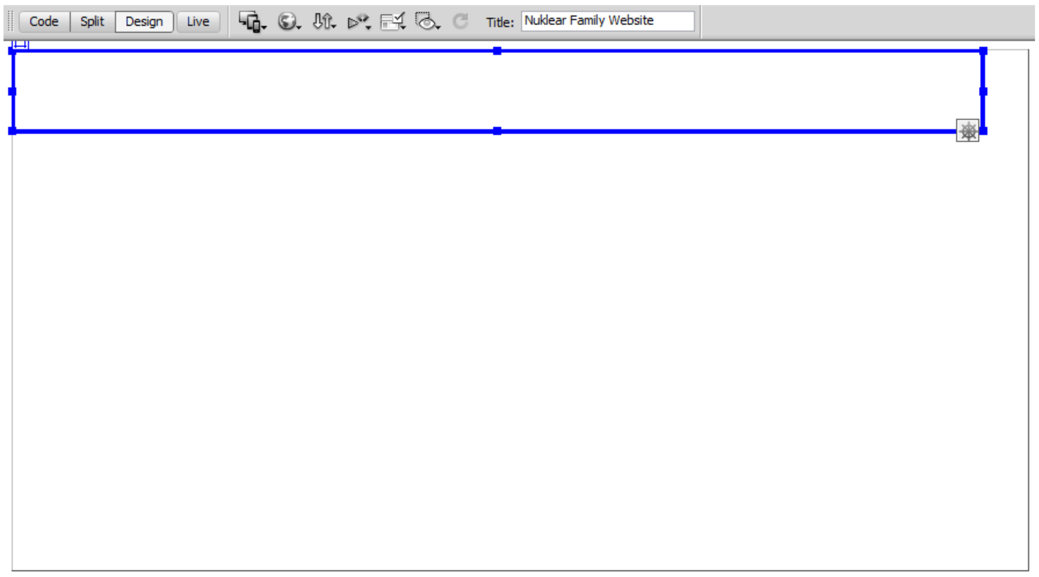
The HTML for the DIV will look something like:

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

While the CSS rule further up the page will look something like:

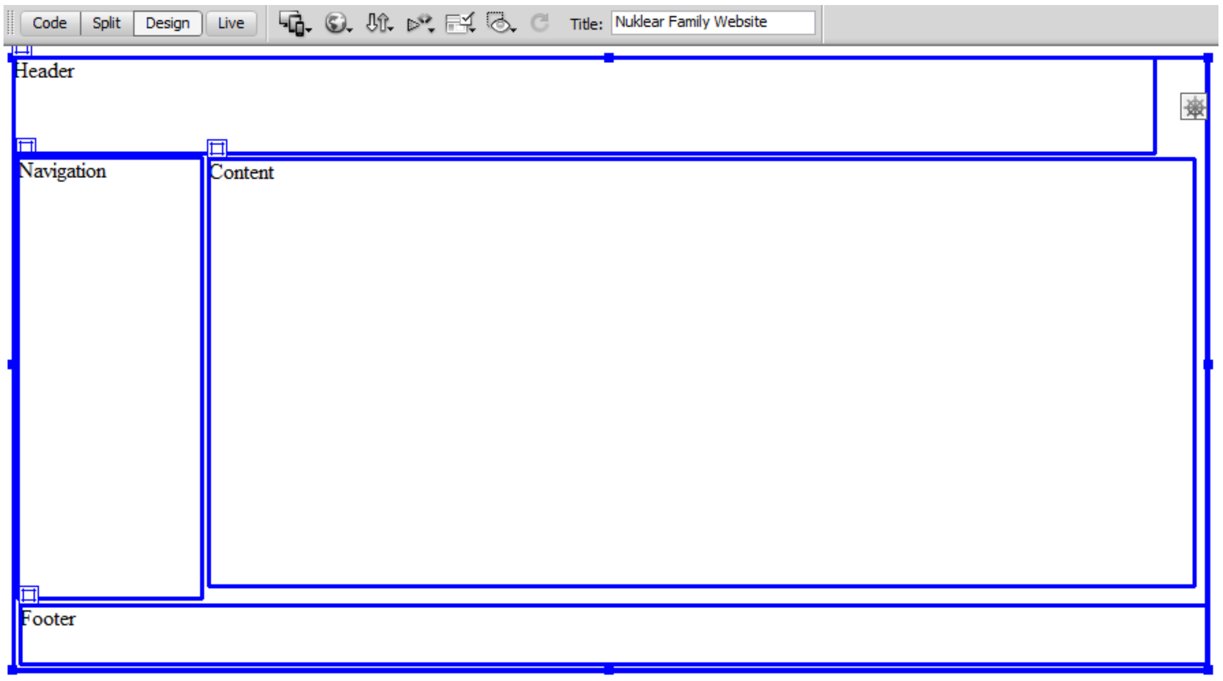
```
<style type="text/css">
#main {
    position: absolute;
    width: 875px;
    height: 448px;
    z-index: 1;
}
</style>
```

6) Click inside the main DIV. The next DIV we add will be nested inside this one.
7) Add a new DIV.
8) Move that DIV so that it is approximately positioned along the top (but still inside the main 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 3 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.

Remember to make sure that the new DIVs should be inside the main one. If you click in a blank area of the main one each time, that will help you to avoid accidentally nesting a new DIV inside one of the other ones you're making.

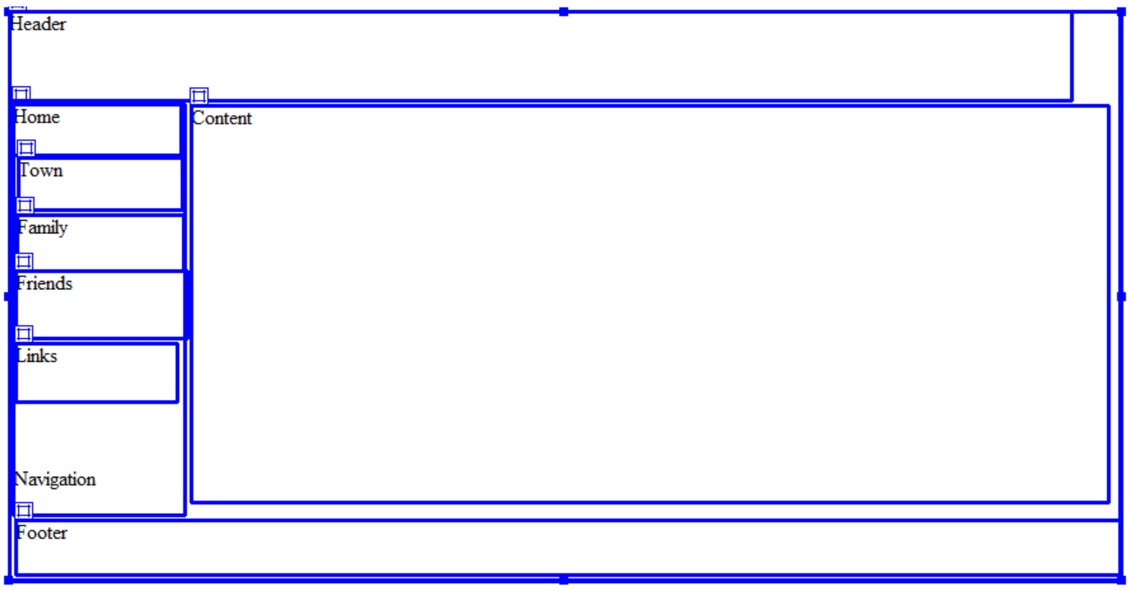


The HTML for your DIV tags should look like the example below.

```
<div id="main">
  <div id="header">Header</div>
  <div id="navigation">Navigation</div>
  <div id="content">Content</div>
  <div id="footer">Footer</div>
</div>
```

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.

10) Add 5 new layers within the navigation layer. Call the layers *home*, *town*, *family*, *friends* and *links*. Add some text within each layer as shown below. In this example, a few blanks lines have been put in the navigation DIV so that the placeholder text isn't covered up by the link DIVs but that isn't essential.



The HTML should look similar to the following. Note the way Dreamweaver uses indentation to make it easier to see which DIV tags are nested inside other DIV tags. This can make it easier to identify if you have accidentally nested a DIV in the wrong spot. If that has happened, you can easily re-arrange the lines by selecting a line and dragging it to where it should go.

```
<body>
<div id="main">
  <div id="header">Header</div>
  <div id="navigation">
    <div id="home">Home</div>
    <div id="town">Town</div>
    <div id="family">Family</div>
    <div id="friends">Friends</div>
    <div id="links">Links</div>
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>Navigation</p>
  </div>
  <div id="content">Content</div>
  <div id="footer">Footer</div>
</div>
</body>
```

11) Save the changes when done.

Exercise 2 – Creating a CSS File

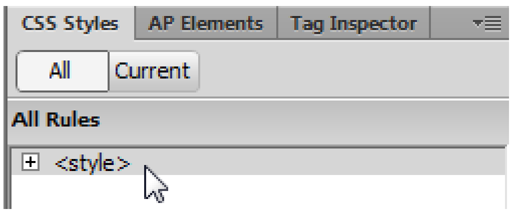
We have now created a rough layout for our page but we have left all of the formatting up to CSS. Each of our layers has been identified by its *id*. When we create our style sheet, we will specify the formatting and positioning of each layer by referring to its id. Every page using the template will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

When we created each of the DIVs in the document, they were each given a style rule in the document based on where we positioned and sized it. We don't really need those style rules since we will be creating a CSS file with the rules we want.

1) Make sure your CSS Styles panel is showing.

We'll get rid of the whole style block including all of the rules inside it. This will avoid any conflict with the rules we will create in our CSS file.

2) Right click on **<style>** within the **CSS Styles** panel and select **Delete**.

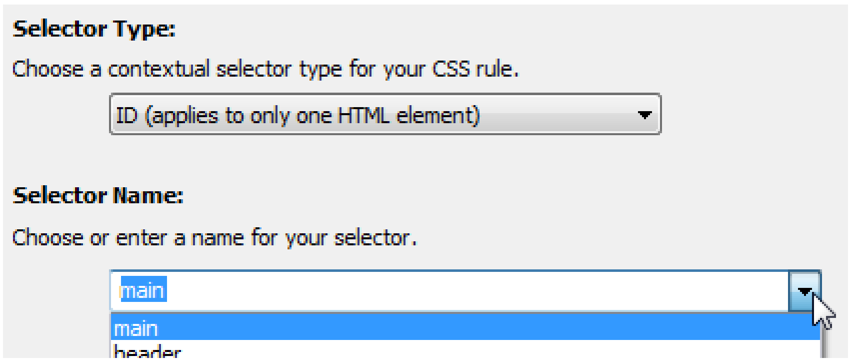


Now we have DIV tags with no CSS telling us where they will be or how they will look. The site will look pretty plain but any text in the site would still be readable. This is what you would get in an old web browser that didn't support CSS.

Our page container is called main so we will create a CSS rule for #main.

3) Click the **New CSS Rule** icon at the bottom of the **CSS Styles** panel.

4) Create a new style for an *ID* **Selector Type**, *main* as the **Selector Name** and with the **Rule Definition** in a new style sheet file. Since we already have an ID in the document called *main*, you should be able to select it from the list.



Selector Type:

Choose a contextual selector type for your CSS rule.

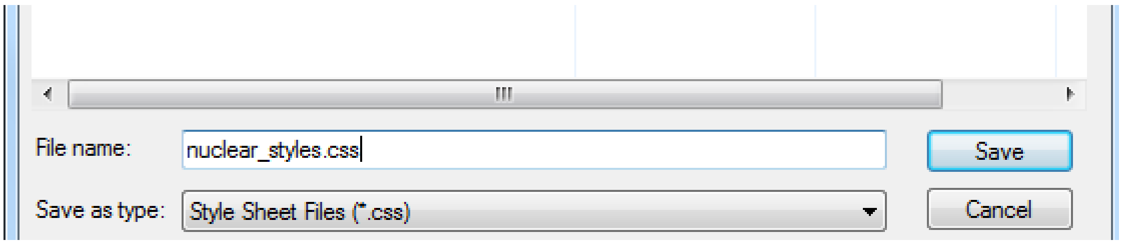
ID (applies to only one HTML element)

Selector Name:

Choose or enter a name for your selector.

main

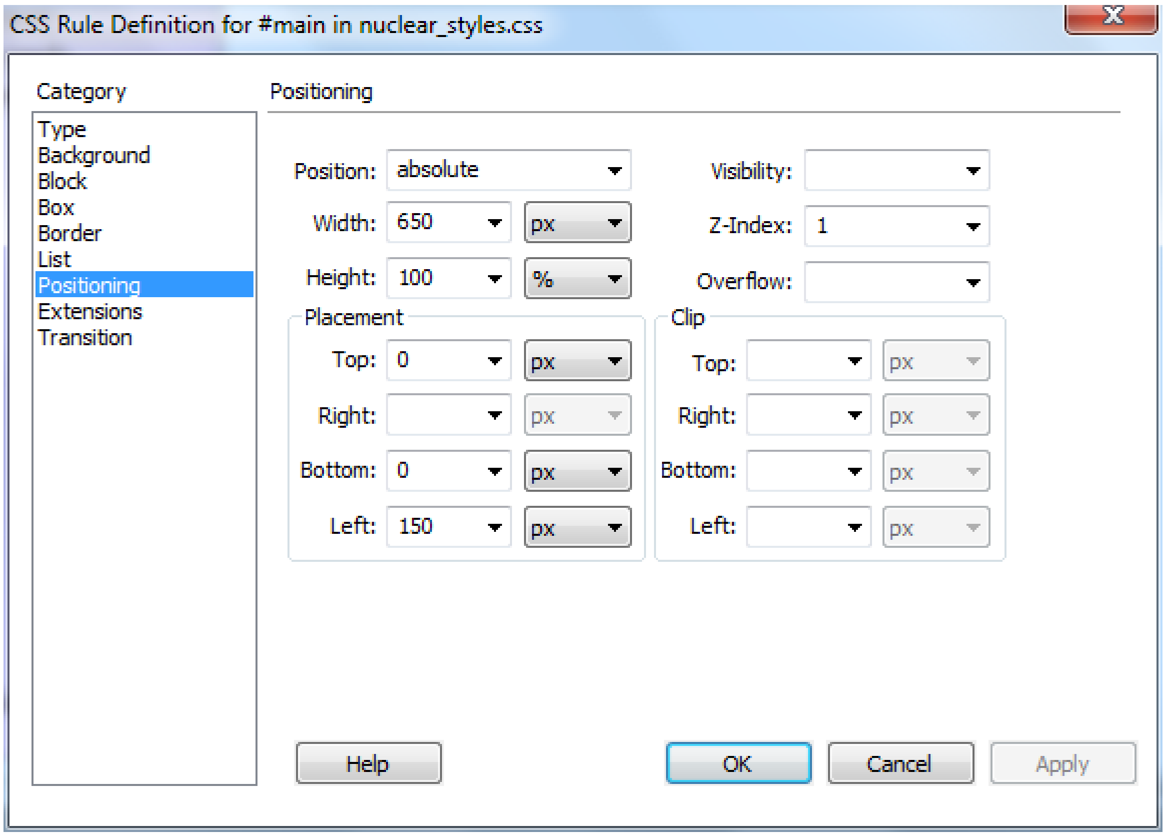
5) Click **OK** and give the style sheet a suitable name (such as nuclear_styles.css).



File name: nuclear_styles.css

Save as type: Style Sheet Files (*.css)

6) Set the **Positioning** properties for the main ID as shown below (Position absolute, 650 pixels width, 100% height, 0 from top & bottom, 150 from the left and z-index set to 1).


CSS Rule Definition for #main in nuclear_styles.css

Category: Type, Background, Block, Box, Border, List, Positioning, Extensions, Transition

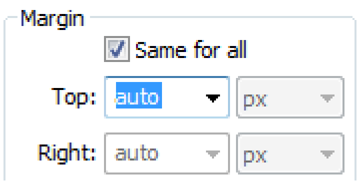
Positioning

Position: absolute
Width: 650 px
Height: 100 %
Visibility:
Z-Index: 1
Overflow:

Placement
Top: 0 px
Right: px
Bottom: 0 px
Left: 150 px

Clip
Top: px
Right: px
Bottom: px
Left: px

7) Set the **Box** properties so the **Margin** is set to **auto** and click **OK** to apply the changes.

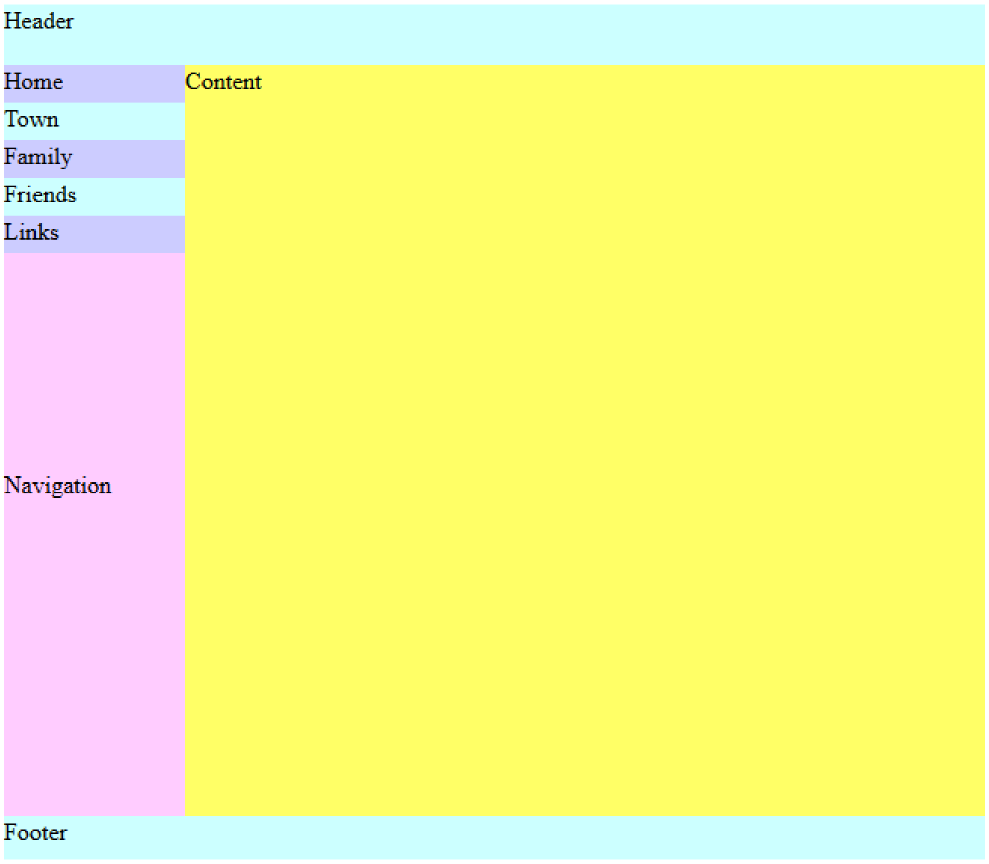

Margin
☑ Same for all
Top: auto px
Right: auto px

8) 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 **DIV** so you can see the effect of their positioning a little as you edit. This makes it a lot easier to see where any problems may lay.

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

9) Save and preview the page (Make sure your CSS file is 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 DIVs.

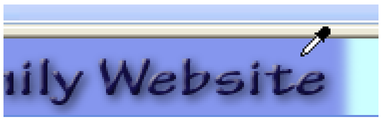


Exercise 3 – Adding Some Content

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.
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 shown below.

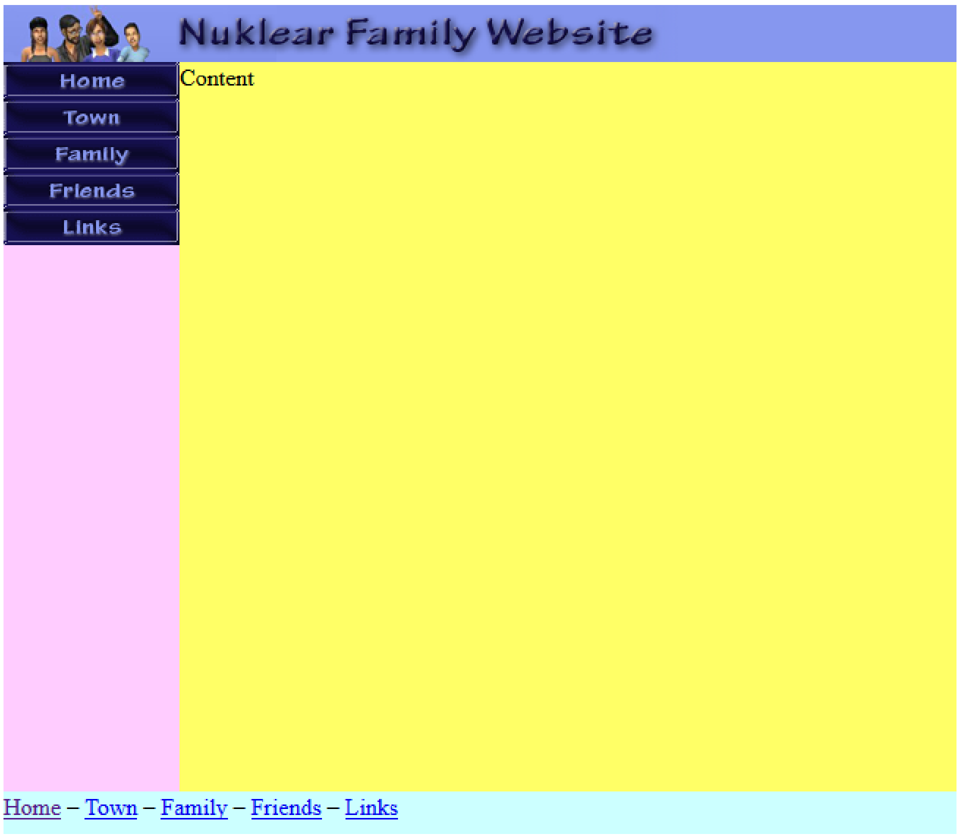
Home – Town – Family – Friends – Links

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

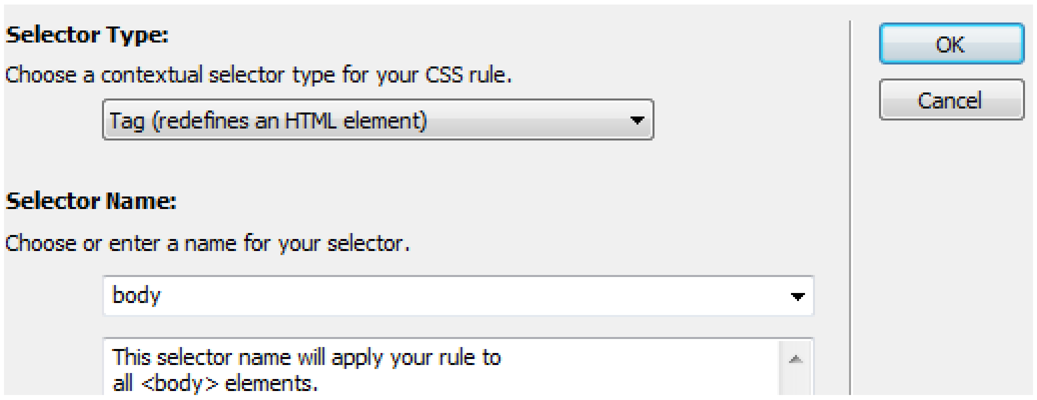
5) Remember to save the file regularly.
6) In each of the DIV layers within the navigation layer, add one of the navigation images as appropriate, replacing the text in each layer.

Remember to add appropriate **ALT** text for each image. You can delete the blank lines and text in the navigation layer. You can also remove the background colours from the CSS rules for these navigation DIV layers. When you save and preview the page it should look similar to the following example.



Exercise 4 – CSS Formatting

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

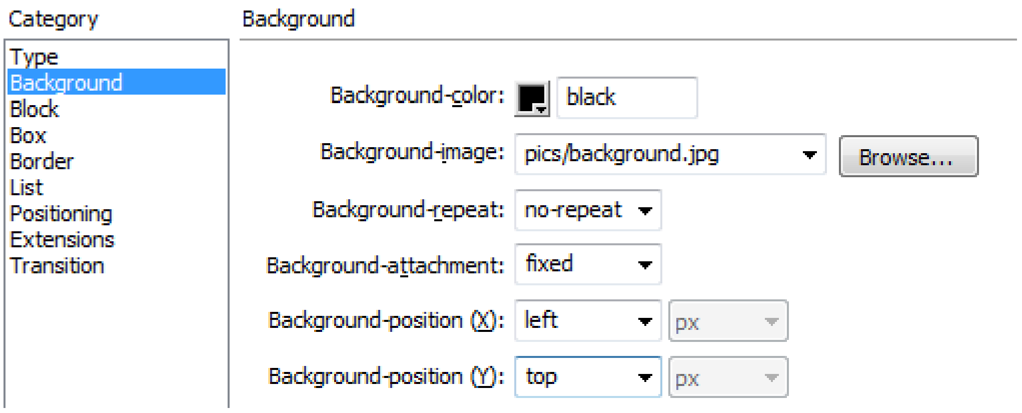


Selector Type:
Choose a contextual selector type for your CSS rule.
Tag (redefines an HTML element)

Selector Name:
Choose or enter a name for your selector.
body

This selector name will apply your rule to all <body> elements.

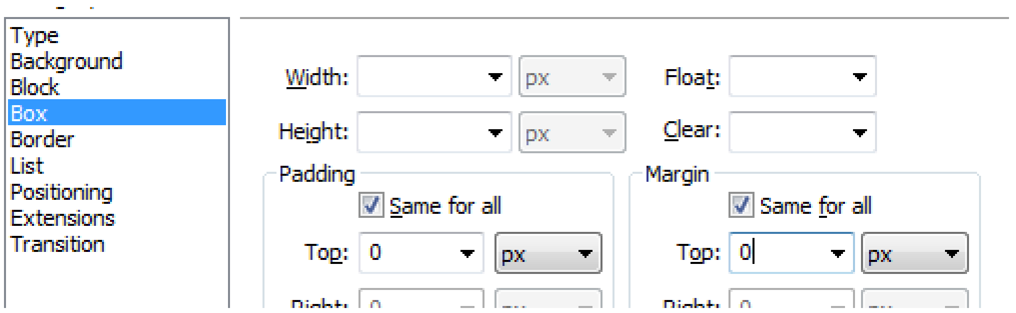
8) Set the **Background** properties as shown



Background

Background-color: black
Background-image: pics/background.jpg
Background-repeat: no-repeat
Background-attachment: fixed
Background-position (X): left px
Background-position (Y): top px

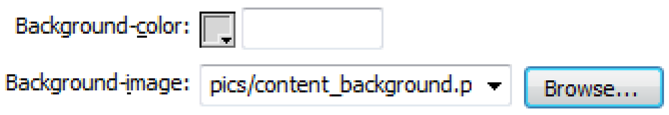
9) Set the **Box** settings so that there is no **Padding** or **Margin** and click **OK** to apply your new style rule.



Width: px
Height: px
Float:
Clear:
Padding ☑ Same for all
Top: 0 px
Margin ☑ Same for all
Top: 0 px

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 have become 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 as it fills the area.

10) 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.



Background-color:

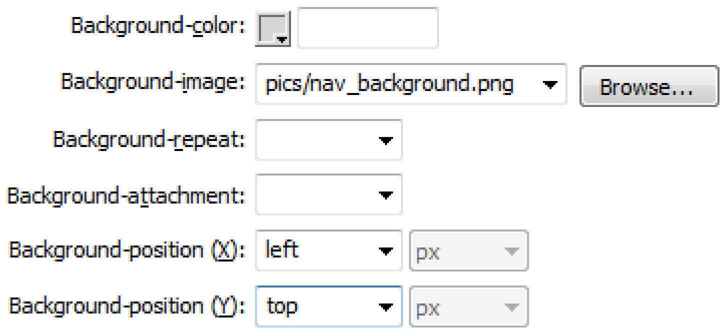
Background-image: pics/content_background.p

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/ .

11) Change the background colour for footer style ID to the same colour you set for the header style ID.

12) 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.



Background-color:

Background-image: pics/nav_background.png

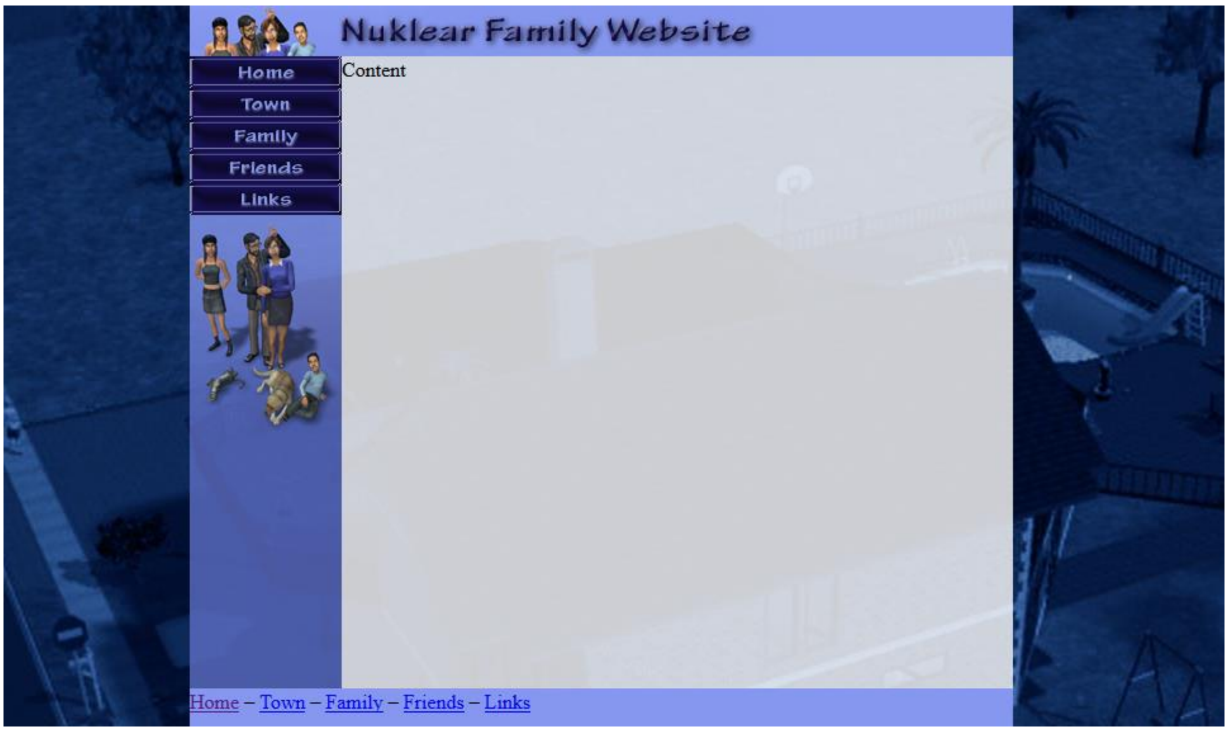
Background-repeat:

Background-attachment:

Background-position (X): left px

Background-position (Y): top px

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



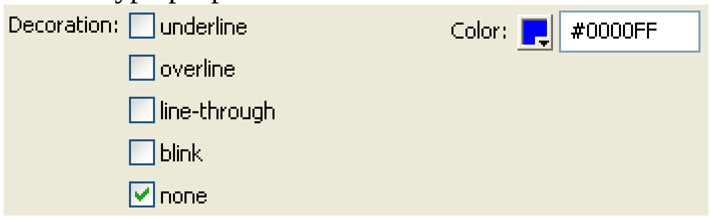
Tip You could change the layout of the entire site now simply by editing the styles. For instance, if you wanted the menu to be across the top of the page, all you'd have to do would be to edit the positioning information for the styles. That will change the layout for the entire site. Some sites even 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.

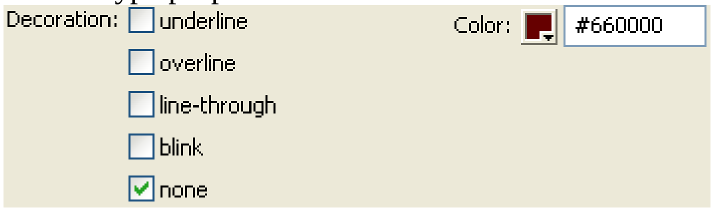
1) Create some new style rules for the hyperlink compound classes as described below.

a:link – type properties



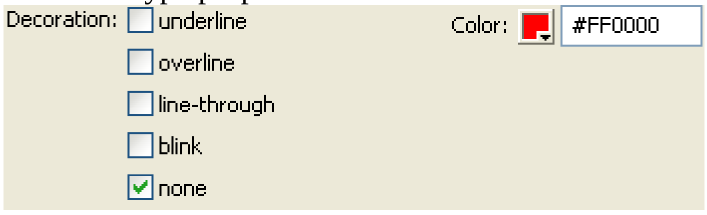
Decoration: ☐ underline ☐ overline ☐ line-through ☐ blink ✓ none
Color: 🟦 #0000FF

a:link – type properties



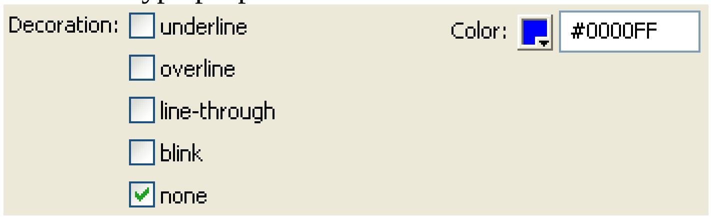
Decoration: ☐ underline ☐ overline ☐ line-through ☐ blink ✓ none
Color: 🟥 #660000

a:active – type properties



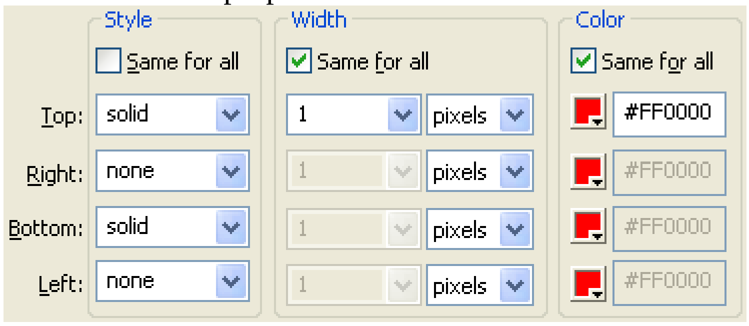
Decoration: ☐ underline ☐ overline ☐ line-through ☐ blink ✓ none
Color: 🟥 #FF0000

a:hover – type properties



Decoration: ☐ underline ☐ overline ☐ line-through ☐ blink ✓ none
Color: 🟦 #0000FF

a:hover – border properties



	Style	Width		Color
	☐ Same for all	✓ Same for all		✓ Same for all
Top:	solid	1	pixels	#FF0000
Right:	none	1	pixels	#FF0000
Bottom:	solid	1	pixels	#FF0000
Left:	none	1	pixels	#FF0000

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

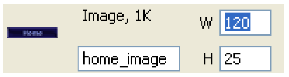
3) Edit the styles for the **BODY** tag so that the following style rules are added:

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

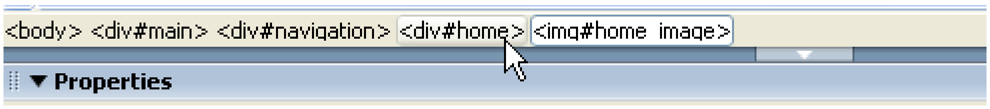
1) Select the Home image in the navigation area.



Image, 1K
W 120
home_image
H 25

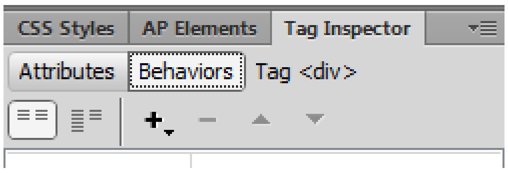
2) Set the **ID** property of the image to *home_image*.
3) Add appropriate IDs for each of the other images.
4) Select the home layer which the image is inside of.

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.

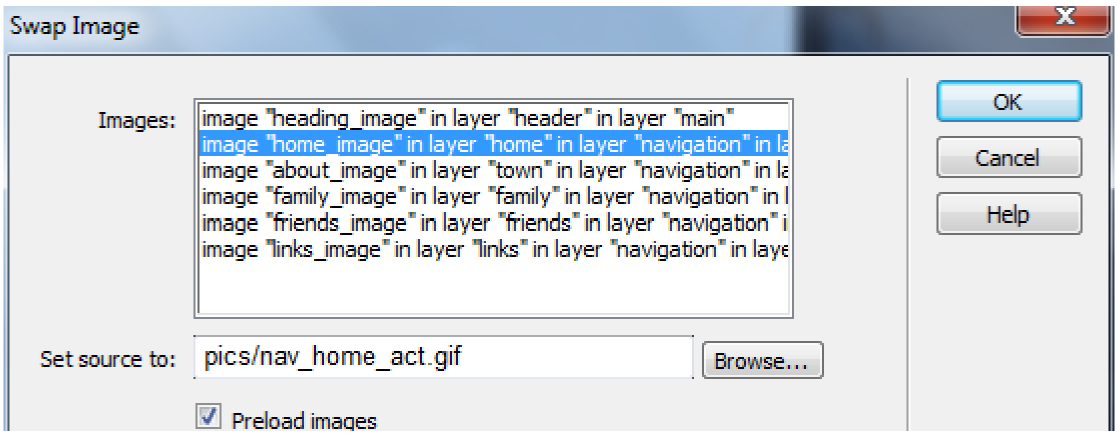


<body> <div#main> <div#navigation> <div#home> <img#home_image>

5) Display the **Tag Inspector** panel if it is not already showing and select the **Behaviours** tab.



6) Add a **swap image** behaviour so that *nav_home.gif* is swapped with *nav_home_act.gif* on mouse over.



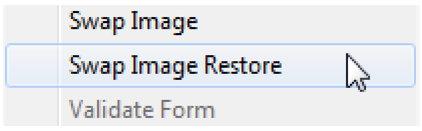
Swap Image

Images:
image "heading_image" in layer "header" in layer "main"
image "home_image" in layer "home" in layer "navigation" in la
image "about_image" in layer "town" in layer "navigation" in la
image "family_image" in layer "family" in layer "navigation" in l
image "friends_image" in layer "friends" in layer "navigation" i
image "links_image" in layer "links" in layer "navigation" in laye

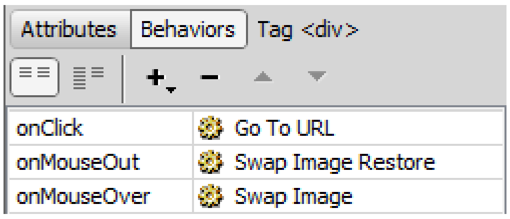
Set source to: pics/nav_home_act.gif

Preload images [✓]

7) Add a **Swap Image Restore** behaviour.



8) Add similar behaviours for the other navigation layers (remember to add a **Swap Image Restore** behaviour for each one as well. Also remember to make sure you have the DIV around the image selected and not the image itself when you add the behaviours).
9) Add a **Go to URL** behaviour for the home layer so that it links to *index.html*. Make sure the behaviour is set to **onClick**.



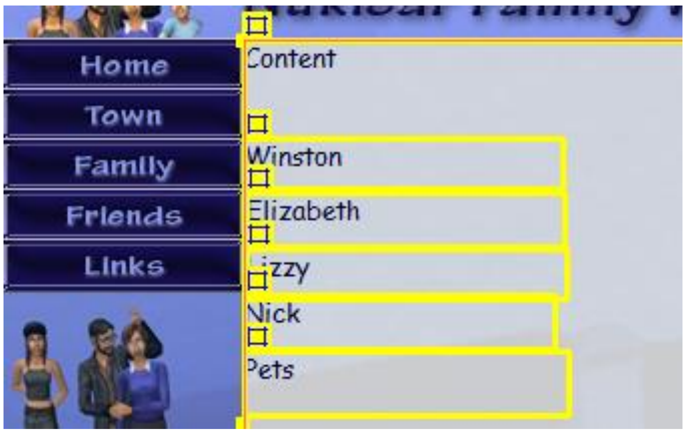
onClick	Go To URL
onMouseOut	Swap Image Restore
onMouseOver	Swap Image

10) Add similar behaviours for the other navigation DIVs. Remember we haven't saved the other pages yet but the file names will be **town.html**, **family.html**, **friends.html** and **links.html**.

11) Create new layers within the content layer for *Winston*, *Elizabeth*, *Lizzy*, *Nick* and *Pets* as shown below. Give each layer a suitable **ID** name.

If clicking on the content DIV only selects the border of the layer rather than allowing you to edit its content, try double-clicking on it instead.

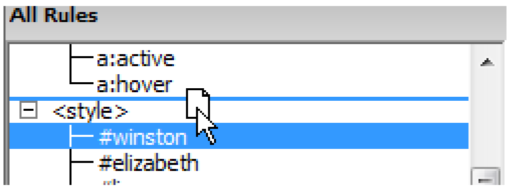
Note They don't necessarily have to be inside the content layer but if you place them within a different layer you will need to adjust the **top** and **left** style properties to get them in the right spot.



12) Modify the style rules for each of the new DIVS as shown below.

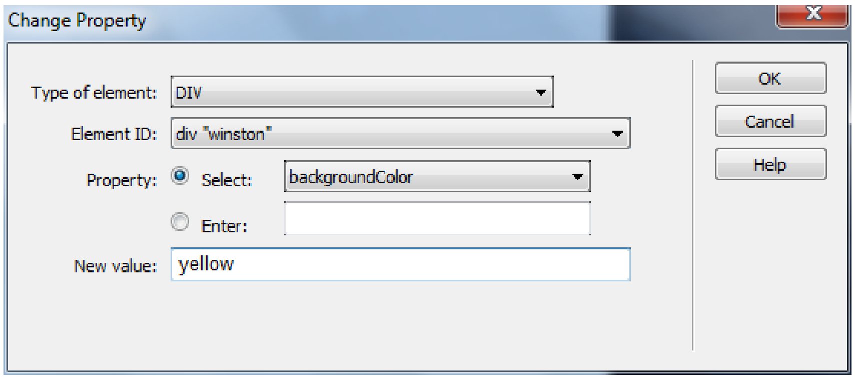
```
#winston
background-color: #8798F0;
border: 1px solid #000033;
position: absolute;
visibility: visible;
z-index: 7;
height: 18px;
width: 90px;
left: 0px;
top: 55px;
text-align: center;

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

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

Once you have modified the style rules, move each of the new DIVs in to the css file.

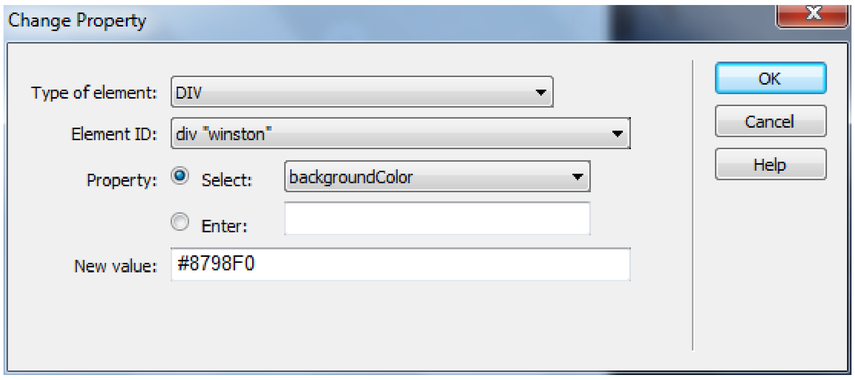


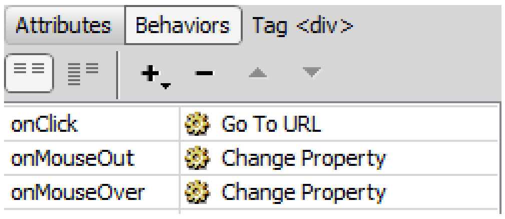
13) For each of the layer in the menu, create a **Go to URL** action so that when they are clicked, they will go to pages that will be created later. The files that each should go to are *winston.html*, *elizabeth.html*, *lizzy.html*, *nick.html* and *pets.html*.

14) For each of the menu layers, create a new **Change Property** behaviour so that the background colour changes to yellow. It should be set to change in the **Behaviours** panel **onMouseOver**.


Change Property
Type of element: DIV
Element ID: div "winston"
Property: [✓ Select:] backgroundColor
Enter:
New value: yellow

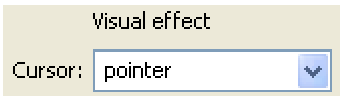
15) Create another **Change Property** behaviour for each layer so that the background colour changes back to its original colour of **#8798F0** that activates **onMouseOut**.


Change Property
Type of element: DIV
Element ID: div "winston"
Property: [✓ Select:] backgroundColor
Enter:
New value: #8798F0



onClick	Go To URL
onMouseOut	Change Property
onMouseOver	Change Property

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*.

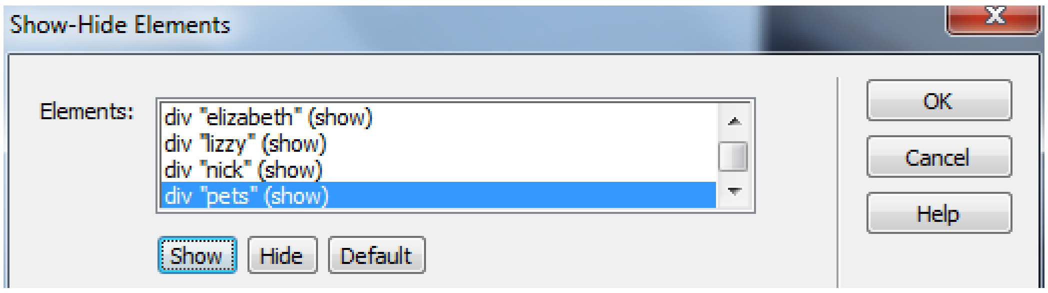

Visual effect
Cursor: pointer

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.

16) Preview your page to make sure everything is looking ok so far.

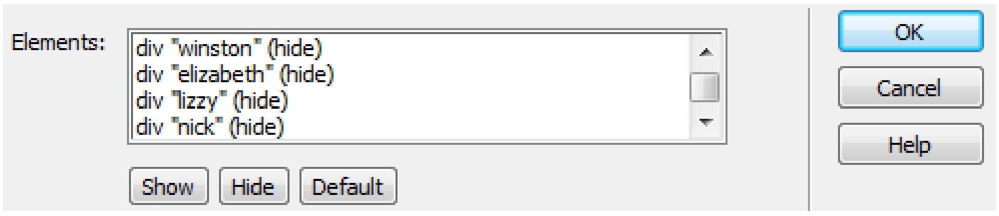
Now we will add some behaviours for our menu. A behaviour will be added to the Family DIV that will set all of the menu layers to Show when the mouse is over the Family layer. Another behaviour will be added that will set the menu to hide when the mouse goes off the Family layer. We will also add behaviours to each of the menu layers to ensure they are still set to show when the mouse is over one of them.

17) Select the Family DIV and add a **Show-hide Elements** behaviour that will show each of the layers in the menu – **Winston**, **Elizabeth**, **Lizzy**, **Nick** and **Pets** (later we will change them so that they are not showing to start with). Make sure the behaviour is set to **onMouseOver**.

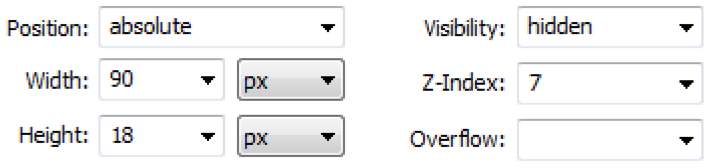

Show-Hide Elements

Elements:
div "elizabeth" (show)
div "lizzy" (show)
div "nick" (show)
div "pets" (show)

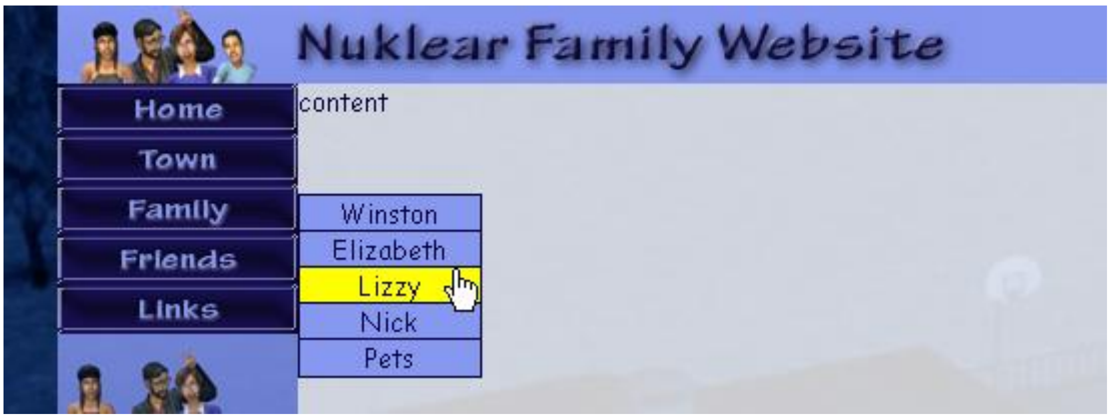
18) Add another behaviour for the Family DIV that will hide all of the menu layers **onMouseOut**.


Elements:
div "winston" (hide)
div "elizabeth" (hide)
div "lizzy" (hide)
div "nick" (hide)

19) Select the **Winston** DIV. Add a **Show-hide Elements** behaviour that sets each of the family menu layers to **Show** when the mouse is over it the Winston layer. This will prevent those layers from vanishing when the user moves their mouse off the Family DIV and on to the menu.
20) Add another **Show-hide Elements** behaviour that will hide the menu layers **onMouseOut**.
21) Repeat steps 19 and 20 for each of the other four menu DIVs. This will make sure that they stay visible as long as the mouse is over one of them.
22) 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. Now they will only appear when the mouse goes over the Family navigation button.


Position: absolute
Width: 90 px
Height: 18 px
Visibility: hidden
Z-Index: 7
Overflow:

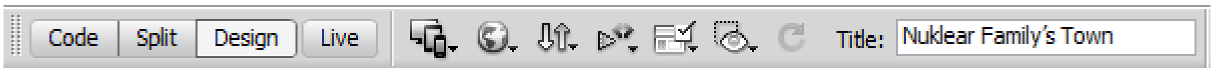
23) Save and preview your page to test the menu (remember that the links go to files that haven't been created yet but we're about to change that).



By now the Code for your page will be looking a lot more complex as the HTML is being combined with programming script for the behaviours.

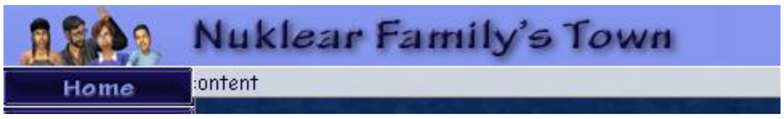
Exercise 7 – Adding more Pages

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

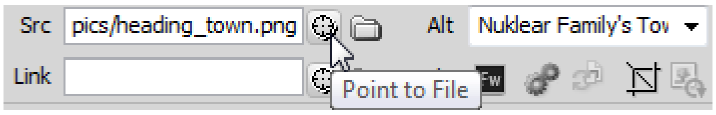


Title: Nuklear Family's Town

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



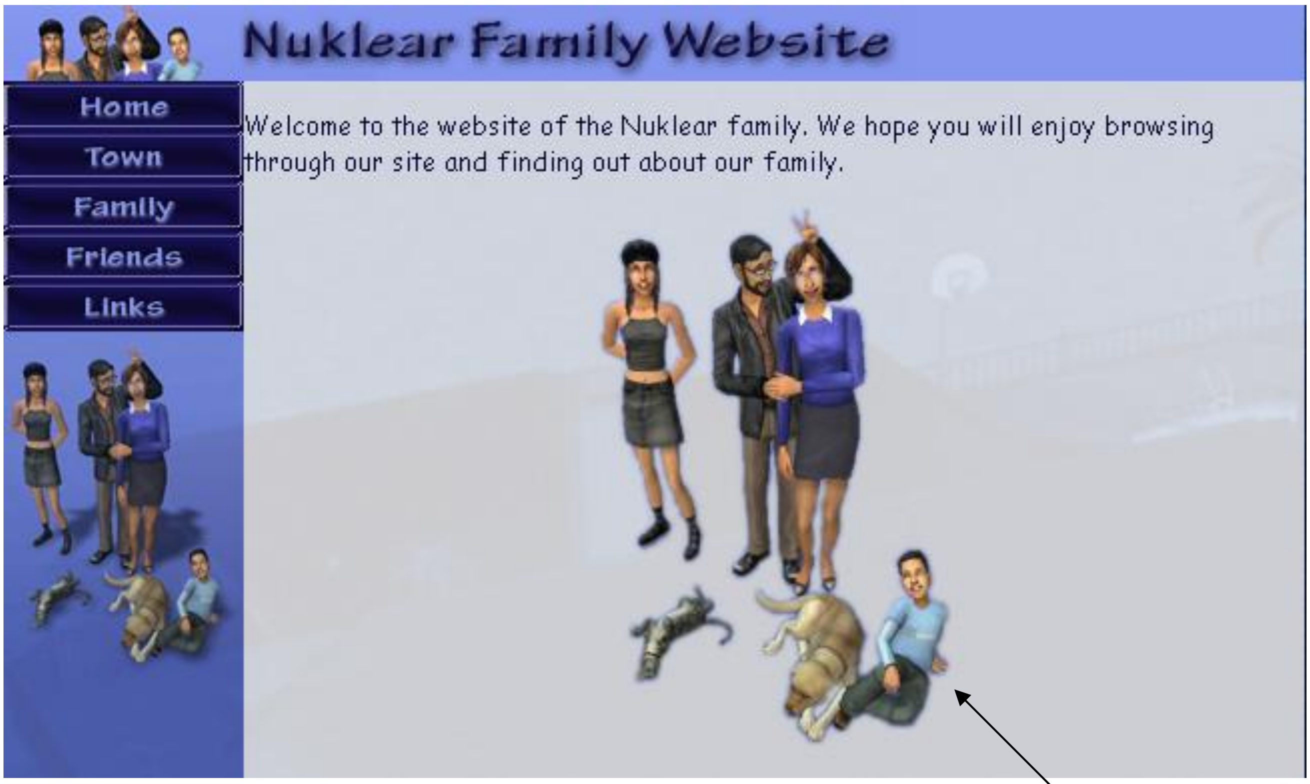
Tip You can use the Point to file to change a file for an image just as you can use it to link to files.



Src pics/heading_town.png

Alt Nuklear Family's Tov

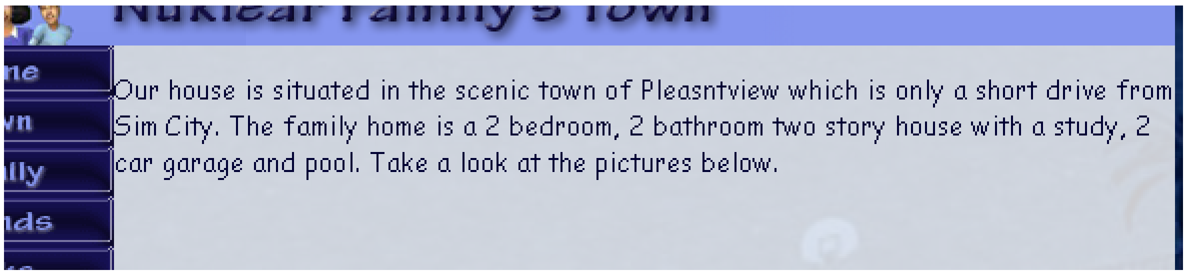
4) Make copies of this page to create *family.html*, *friends.html* and *links.html*.
5) Change the page title and header image in each.
6) Open the *family.html* page and save a copy of it as *winston.html*.
7) Change the page title to *Nuklear Family – Winston*.
8) In the content section, replace the existing text with *Winston* and make it a level 1 heading. You may need to double-click the content layer to edit its contents (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.



Welcome to the website of the Nuklear family. We hope you will enjoy browsing through our site and finding out about our family.

Image filename - main_family.png

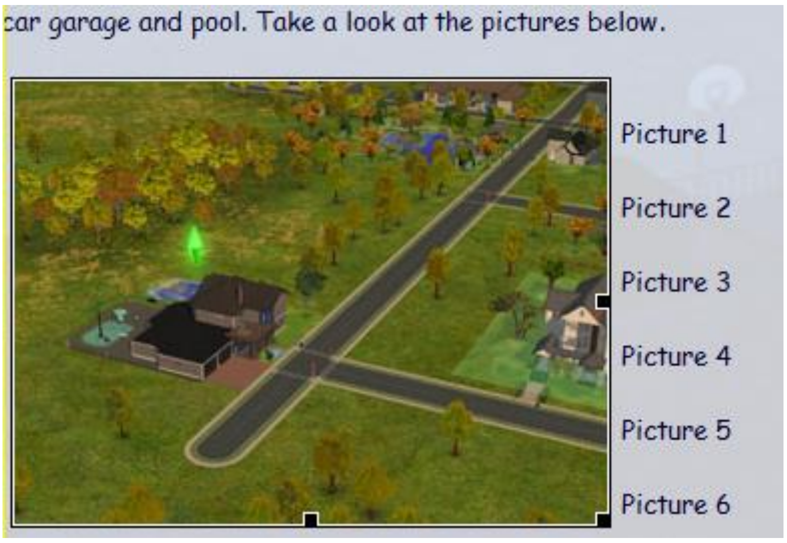
11) Add the following content to the Town page.



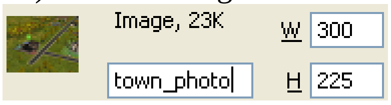
Our house is situated in the scenic town of Pleasntview which is only a short drive from Sim City. The family home is a 2 bedroom, 2 bathroom two story house with a study, 2 car garage and pool. Take a look at the pictures below.

12) Add the image *town1.jpg* below the text as shown below along with the text next to the image. You will need to set the image to align left so that the text can wrap around it to the right.

Tip If you want a bit of space between the image and the text, right click the image, choose **Edit Tag <img>** and then adjust the **horizontal space** property.



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.

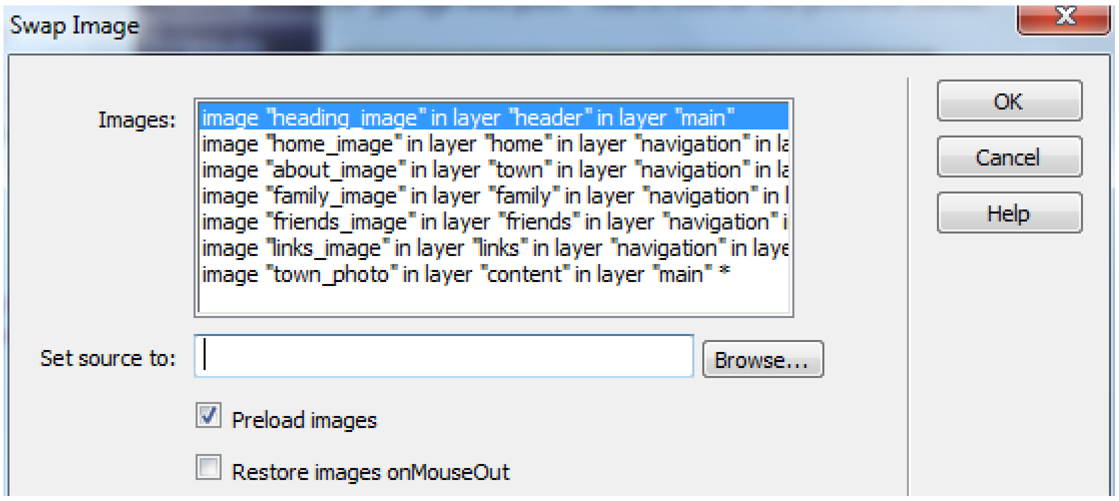
14) Edit the HTML to add anchor tags to the text next to the picture. We won't add any anchors though. The anchors will be used in a behaviour.

```
   <p><img src="pics/town1.jpg" alt="Our Town" name="town_photo" width="300" height="225" id="town_photo" align="left" hspace="5" /> </p>
   <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><img src="pics/town1.jpg" alt="Our Town" name="town_photo" width="300" height="225" id="town_photo" align="left" hspace="5" /> </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*. Untick the **Restore images onMouseOut** option. Set it to **onclick** and if an on **mouseout** behaviour has been created, remove it. We want it to stay on that image until another link is clicked.

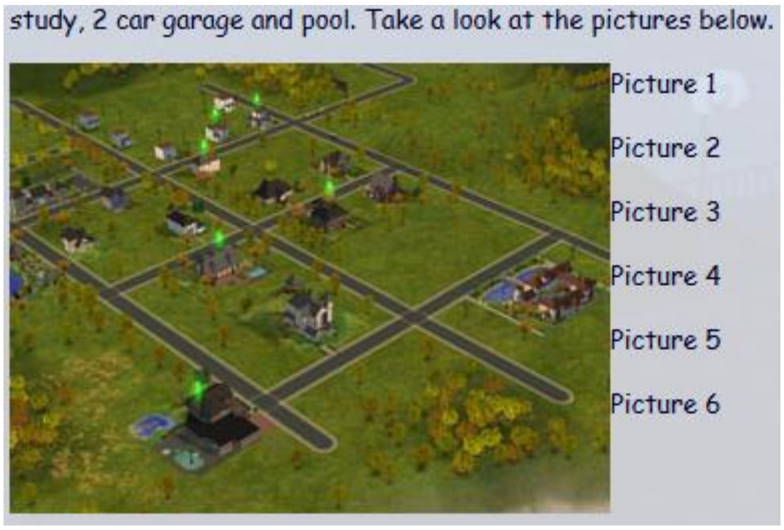

Swap Image

Images:
image "heading_image" in layer "header" in layer "main"
image "home_image" in layer "home" in layer "navigation" in la
image "about_image" in layer "town" in layer "navigation" in la
image "family_image" in layer "family" in layer "navigation" in l
image "friends_image" in layer "friends" in layer "navigation" i
image "links_image" in layer "links" in layer "navigation" in laye
image "town_photo" in layer "content" in layer "main" *

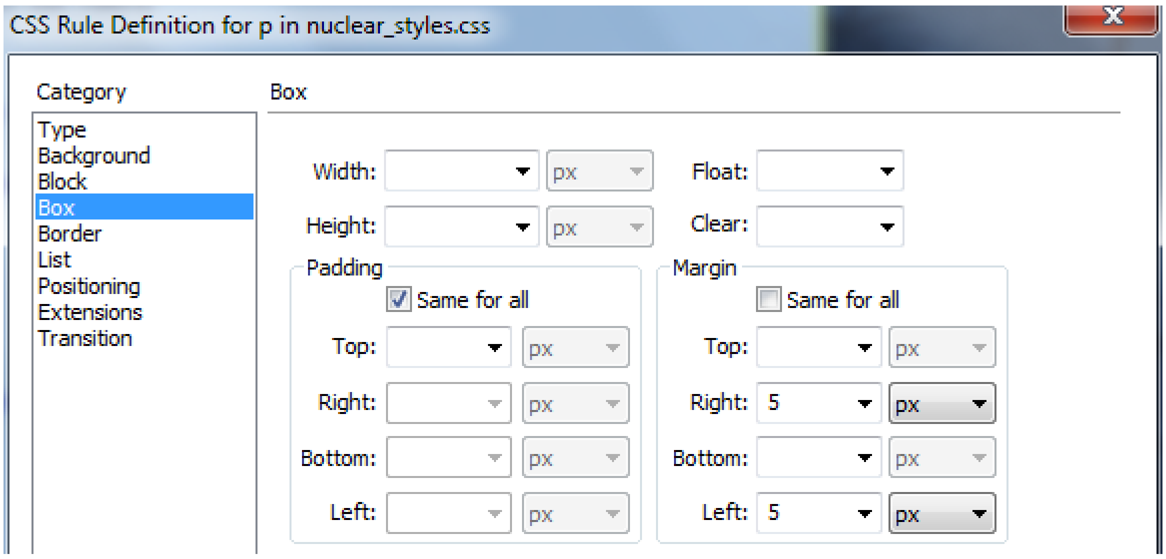
Set source to:

[✓] Preload images
[ ] Restore images onMouseOut

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.


study, 2 car garage and pool. Take a look at the pictures below.
Picture 1
Picture 2
Picture 3
Picture 4
Picture 5
Picture 6

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.


CSS Rule Definition for p in nuclear_styles.css

Category: Type, Background, Block, **Box**, Border, List, Positioning, Extensions, Transition

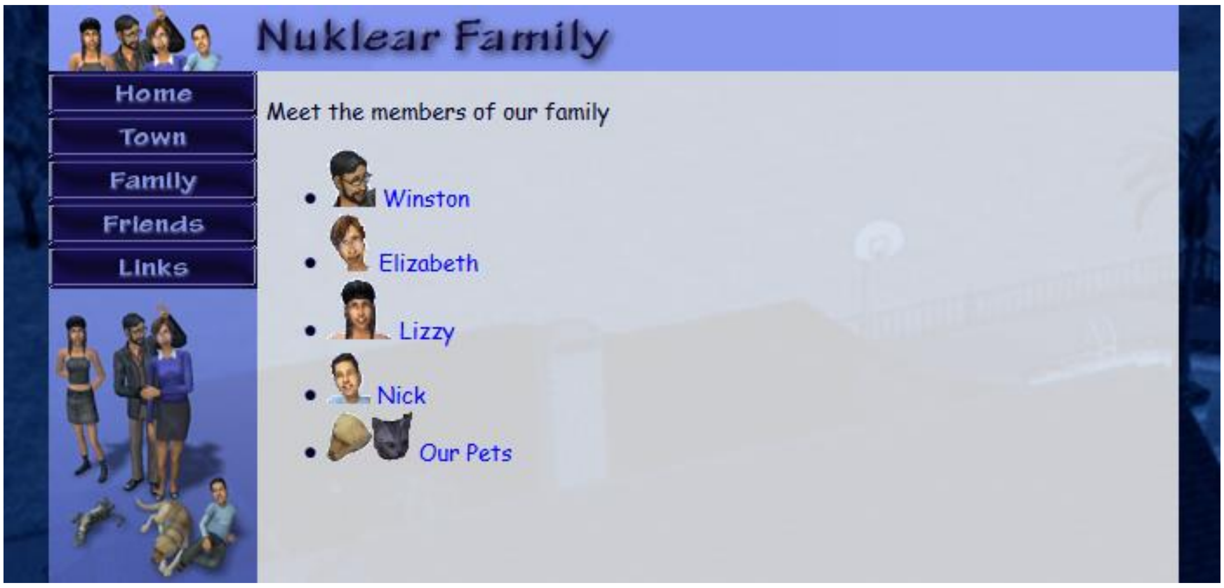
Box

Width: px | Float:
Height: px | Clear:

Padding: [✓] Same for all
Top: px
Right: px
Bottom: px
Left: px

Margin: [ ] Same for all
Top: px
Right: 5 px
Bottom: px
Left: 5 px

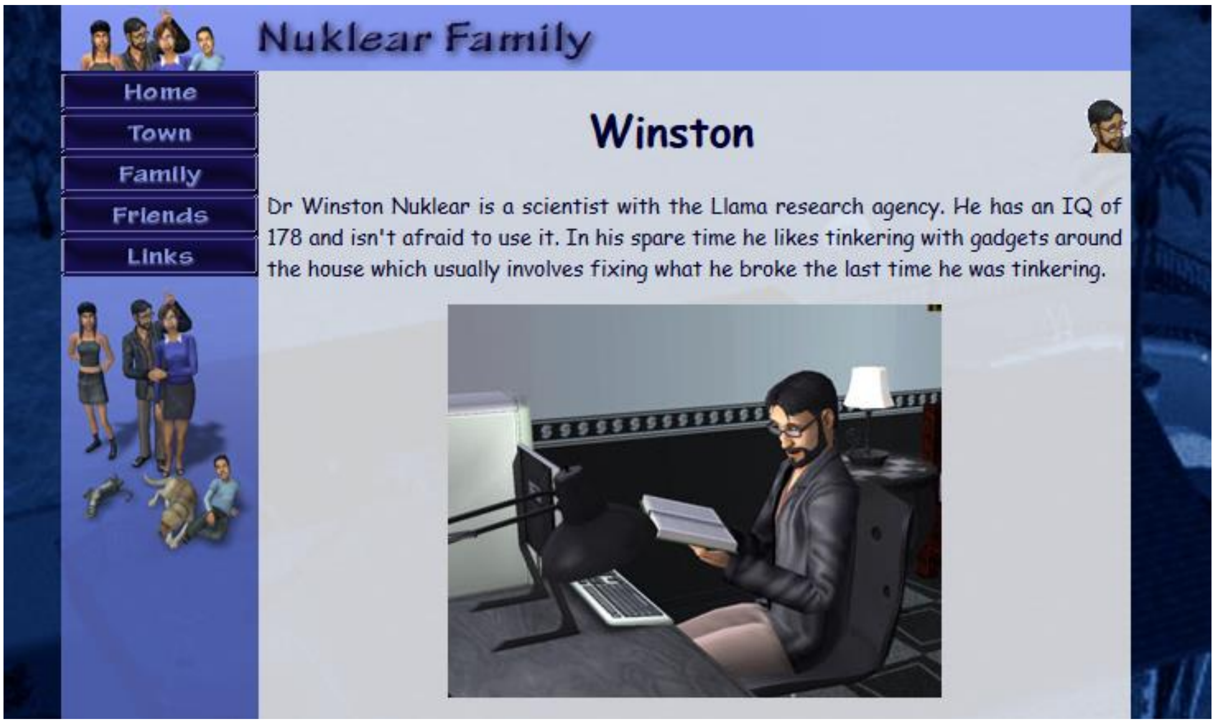
17) Edit the *family.html* page so that it looks like the following. Each family member's name should be a link to their page and should be in a unordered (bulleted) list. The images to use are *icon_winston.png*, *icon_elizabeth.png*, *icon_lizzy.png*, *icon_nick.png* and *icon_pets.png*.



Meet the members of our family

- Winston
- Elizabeth
- Lizzy
- Nick
- Our Pets

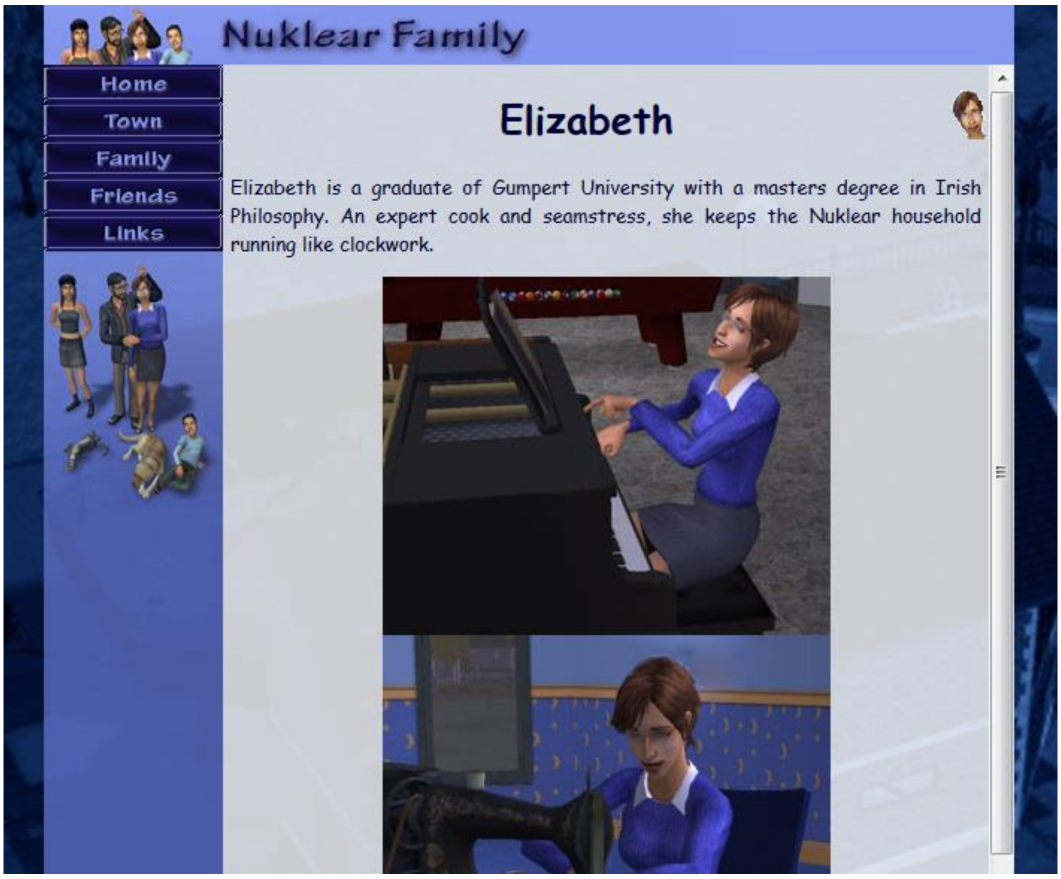
18) Modify the file *winston.html* so it looks like the following. You will need to create a style rule for **H1** to make it look like the heading in the example and you will also need to change the style rule for paragraphs so that text alignment is justified. Of course changing the style rule to make paragraphs justified aligned might mean you'll need an inline style to make the paragraph with the picture centre aligned.



Winston

Dr Winston Nuklear is a scientist with the Llama research agency. He has an IQ of 178 and isn't afraid to use it. In his spare time he likes tinkering with gadgets around the house which usually involves fixing what he broke the last time he was tinkering.

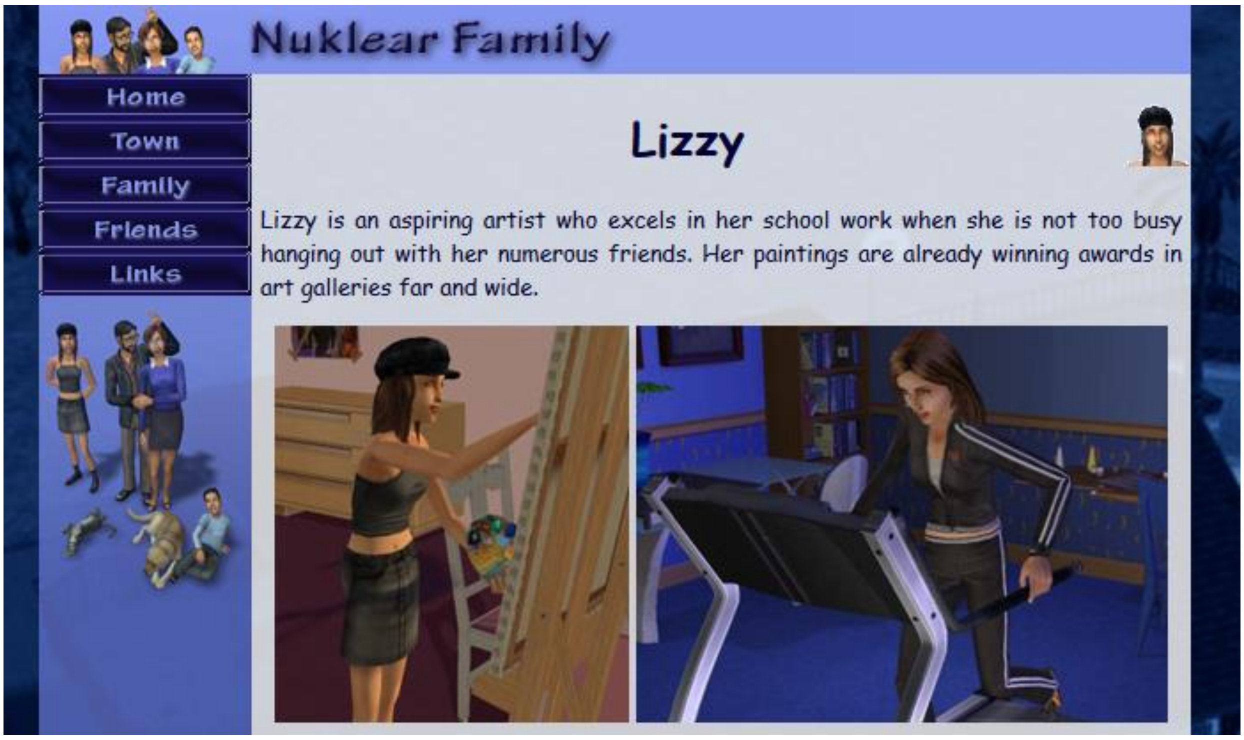
19) Edit the page *elizabeth.html* so that it appears like the following example.



Elizabeth

Elizabeth is a graduate of Gumpert University with a masters degree in Irish Philosophy. An expert cook and seamstress, she keeps the Nuklear household running like clockwork.

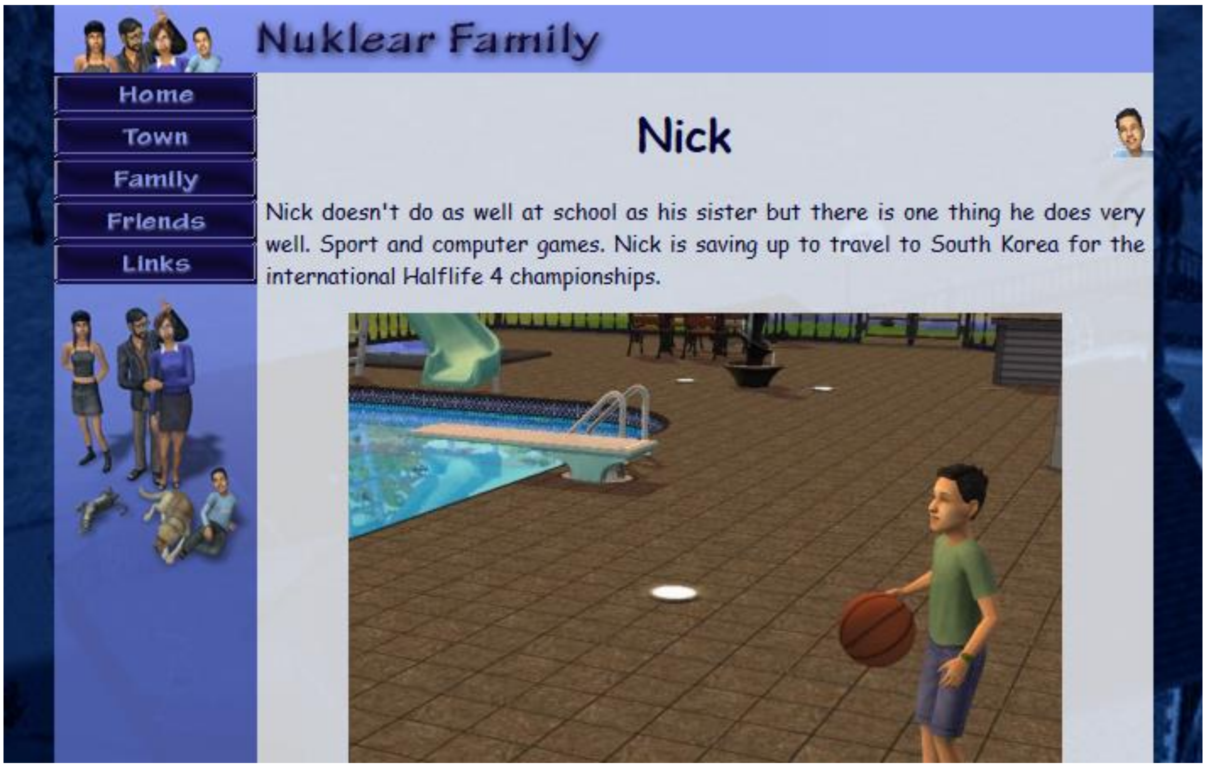
20) Modify *lizzy.html* so that it looks like the following.



Lizzy

Lizzy is an aspiring artist who excels in her school work when she is not too busy hanging out with her numerous friends. Her paintings are already winning awards in art galleries far and wide.

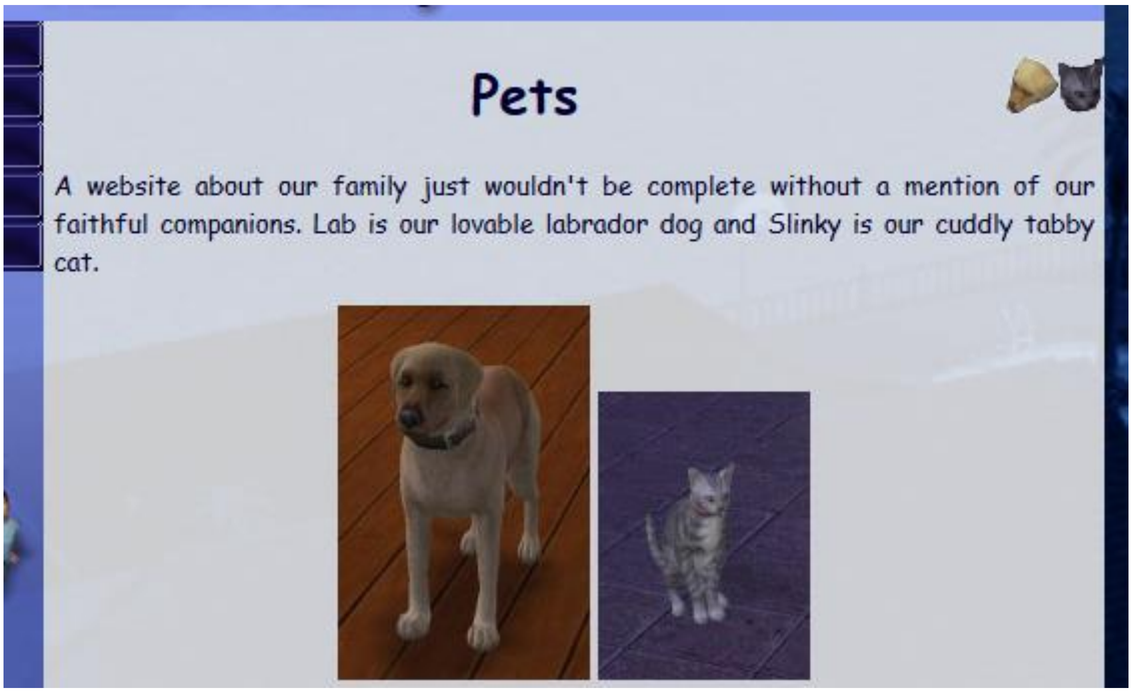
21) Modify *nick.html* so that it looks like the following.



Nick

Nick doesn't do as well at school as his sister but there is one thing he does very well. Sport and computer games. Nick is saving up to travel to South Korea for the international Halflife 4 championships.

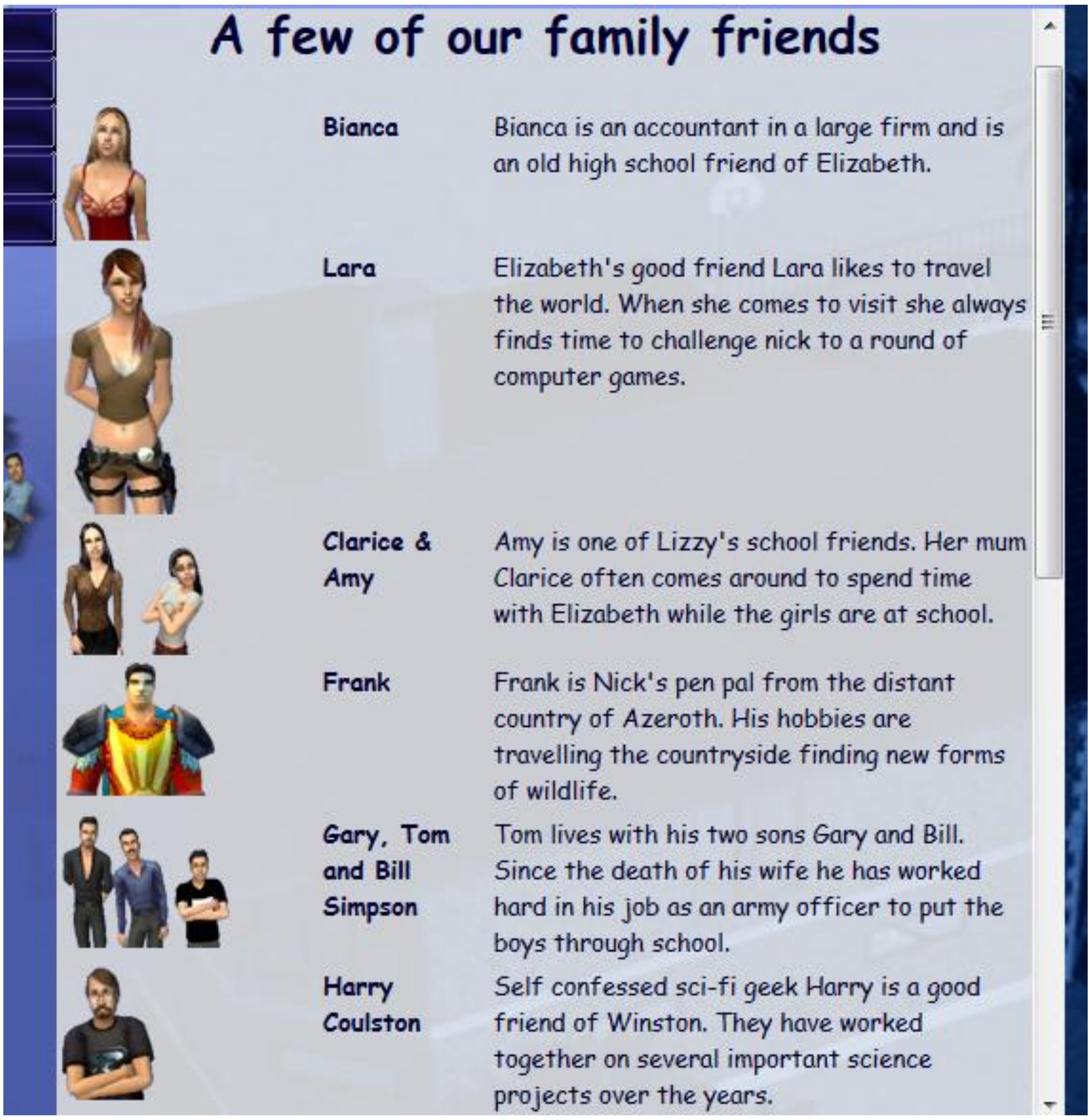
22) Modify *pets.html* so that it looks like the following.



Pets

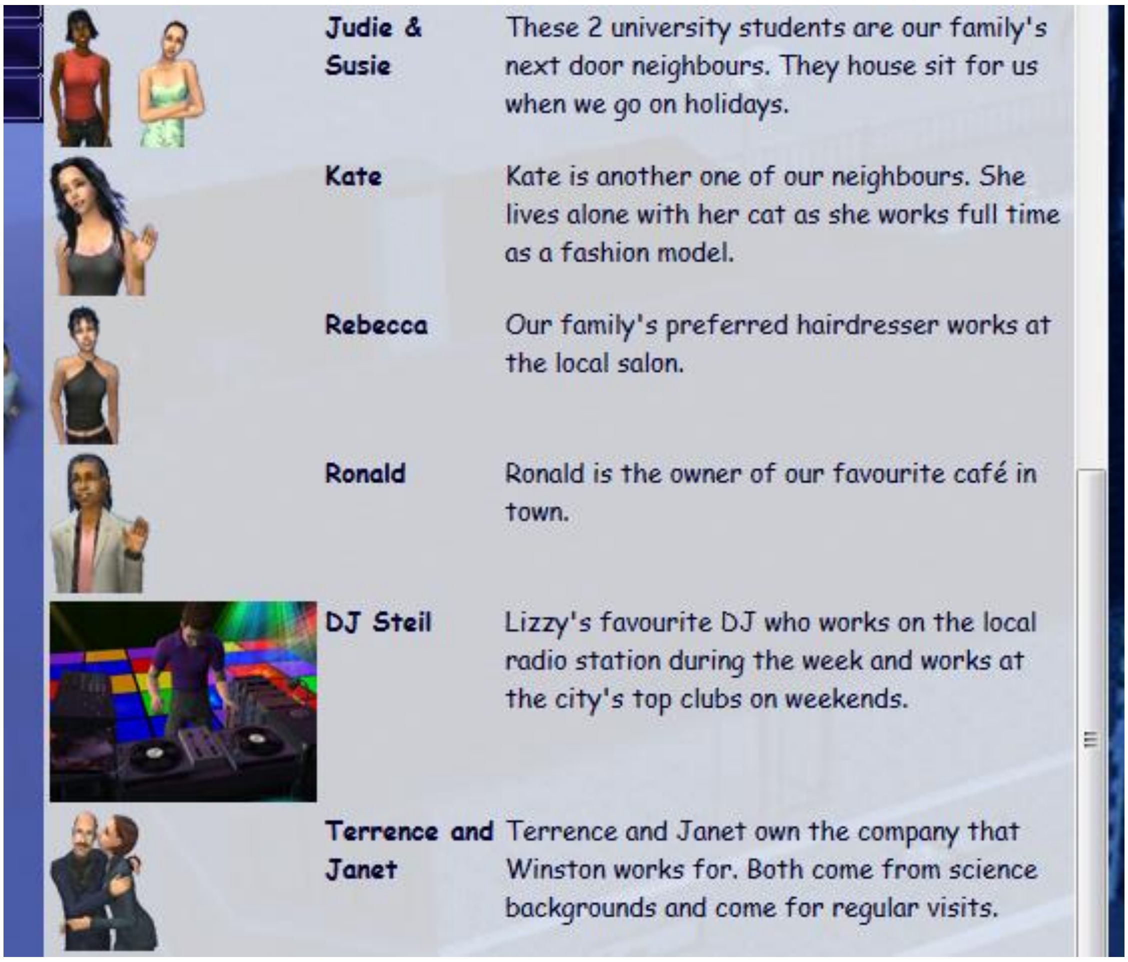
A website about our family just wouldn't be complete without a mention of our faithful companions. Lab is our lovable labrador dog and Slinky is our cuddly tabby cat.

23) Edit the friends page to looks similar to the following example. Add names and descriptions for the images (a table might be useful for layout). You could add a style rule for your table cells (TD tag) that sets their vertical alignment to top.



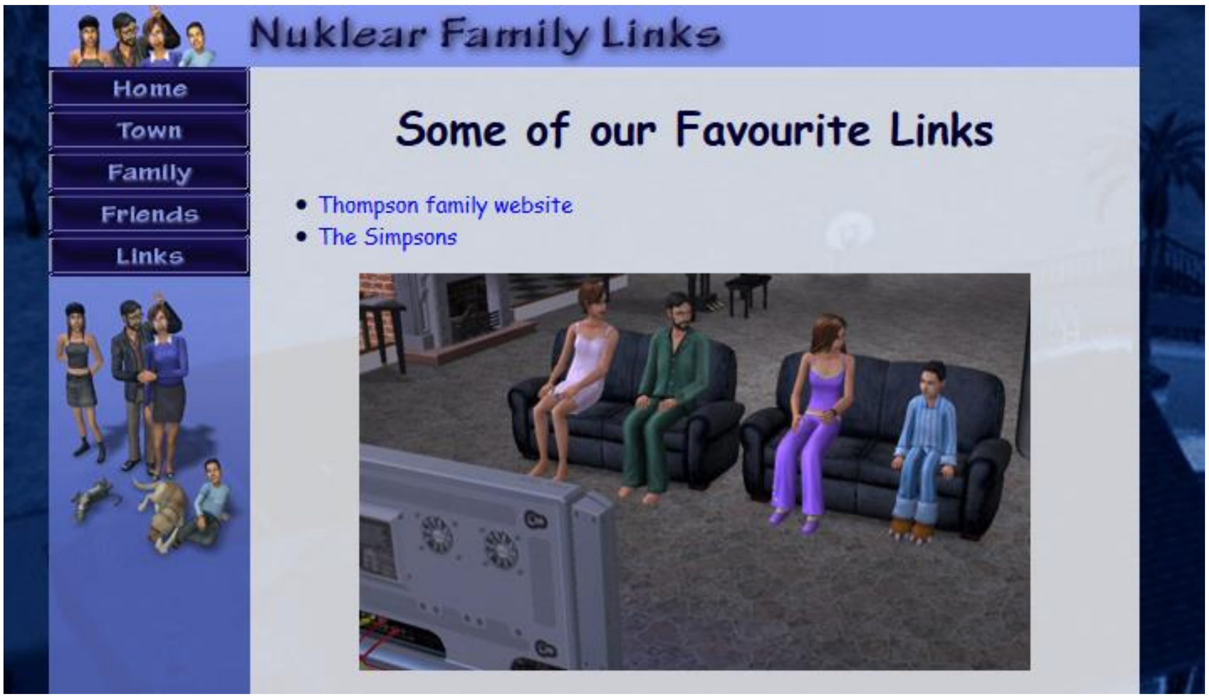
A few of our family friends

	Bianca	Bianca is an accountant in a large firm and is an old high school friend of Elizabeth.
	Lara	Elizabeth's good friend Lara likes to travel the world. When she comes to visit she always finds time to challenge nick to a round of computer games.
	Clarice & Amy	Amy is one of Lizzy's school friends. Her mum Clarice often comes around to spend time with Elizabeth while the girls are at school.
	Frank	Frank is Nick's pen pal from the distant country of Azeroth. His hobbies are travelling the countryside finding new forms of wildlife.
	Gary, Tom and Bill Simpson	Tom lives with his two sons Gary and Bill. Since the death of his wife he has worked hard in his job as an army officer to put the boys through school.
	Harry Coulston	Self confessed sci-fi geek Harry is a good friend of Winston. They have worked together on several important science projects over the years.



	Judie & Susie	These 2 university students are our family's next door neighbours. They house sit for us when we go on holidays.
	Kate	Kate is another one of our neighbours. She lives alone with her cat as she works full time as a fashion model.
	Rebecca	Our family's preferred hairdresser works at the local salon.
	Ronald	Ronald is the owner of our favourite café in town.
	DJ Steil	Lizzy's favourite DJ who works on the local radio station during the week and works at the city's top clubs on weekends.
	Terrence and Janet	Terrence and Janet own the company that Winston works for. Both come from science backgrounds and come for regular visits.

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



Some of our Favourite Links

- Thompson family website
- The Simpsons

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.

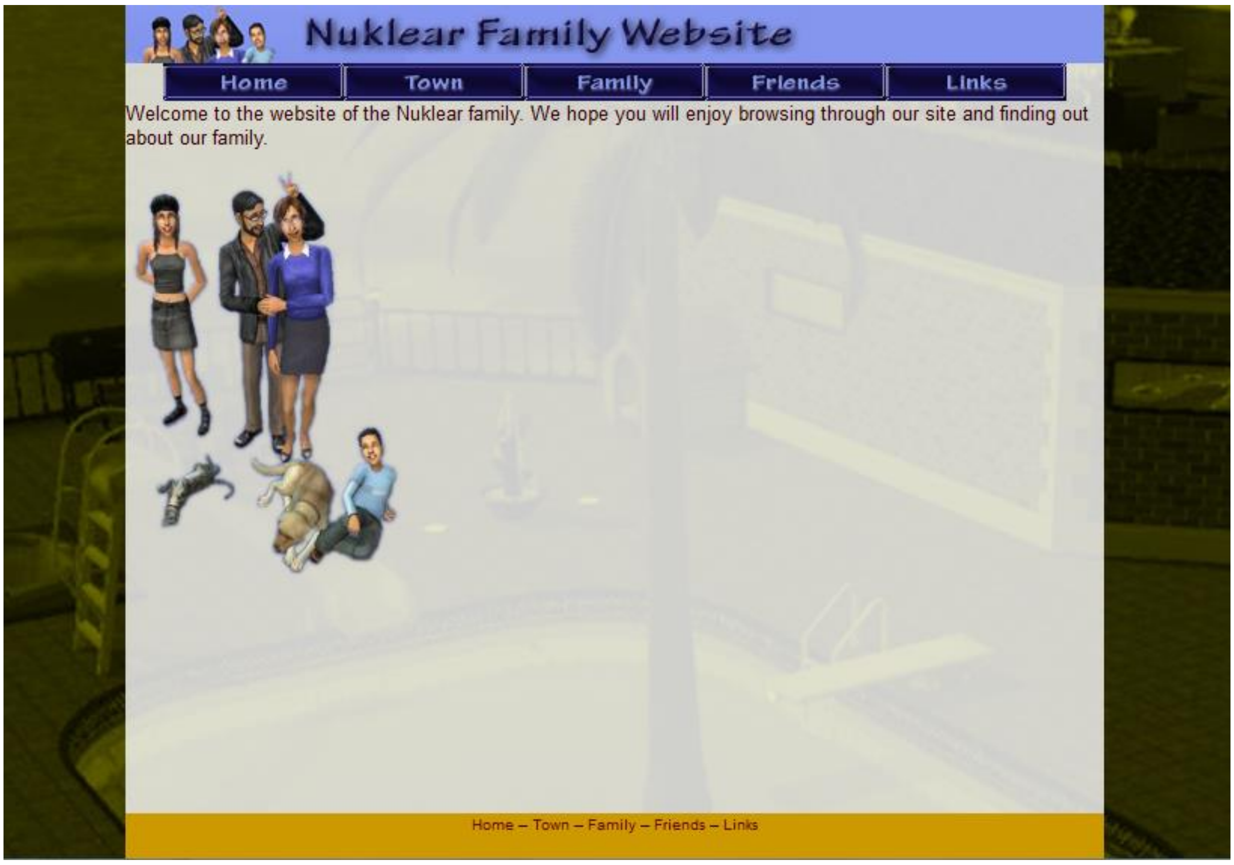
1) Open any one of the pages in your site (they all currently use the same stylesheet).
2) From the exercise files, make sure that the file alternate_styles.css is saved in the same folder as your main website.
3) In your HTML code, locate the line that links to the stylesheet.

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

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

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

5) 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.



Welcome to the website of the Nuklear family. We hope you will enjoy browsing through our site and finding out about our family.

These days it is common for websites to be viewed on more than just computers. Different stylesheets can be created for different devices. E.g. one stylesheet that is used when someone views your website on a personal computer. Another stylesheet that will be used when someone views your website on a mobile device, such as a phone or a tablet.