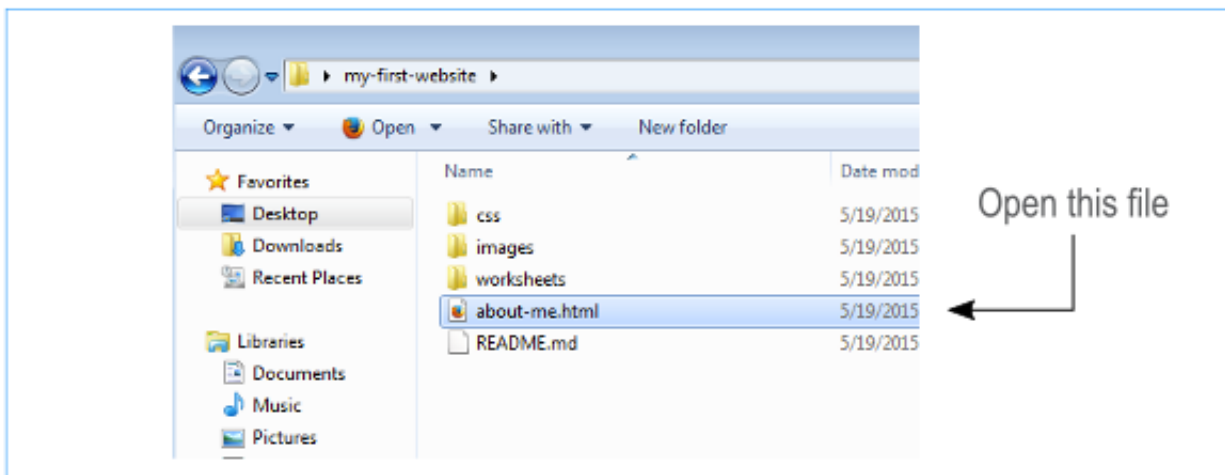
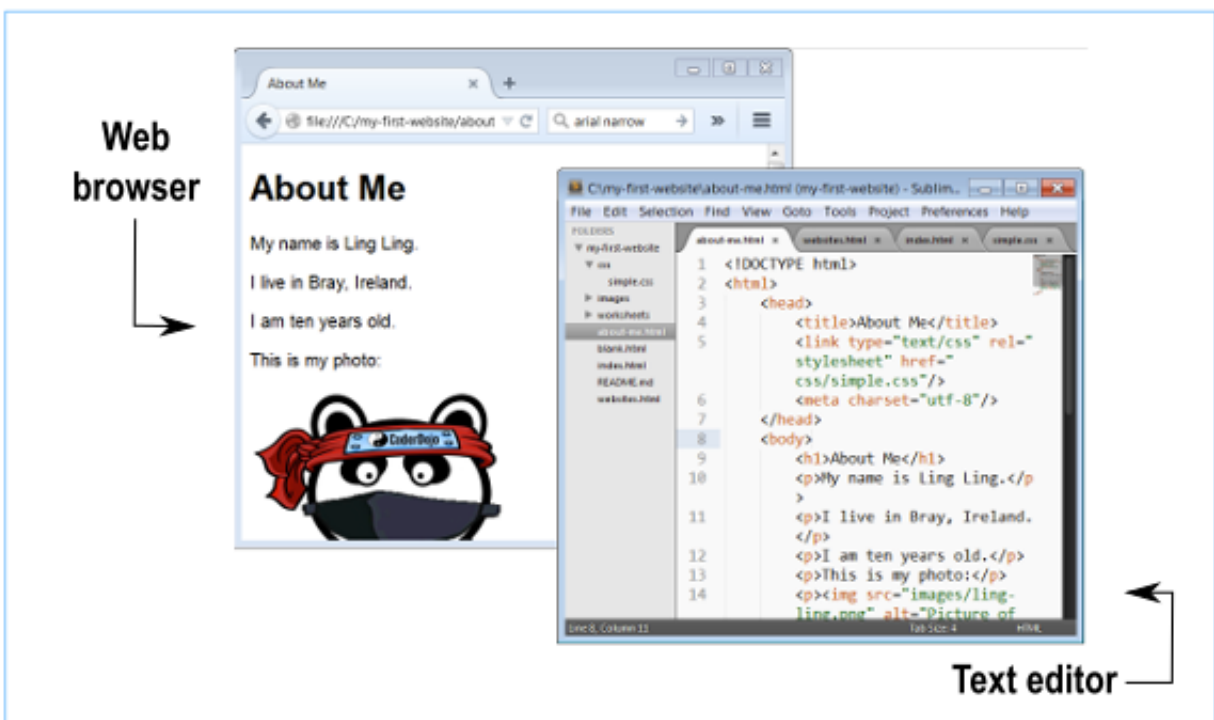


We're going to build a new web page! We're going to make ours about "Books" but you can make yours about whatever you like!

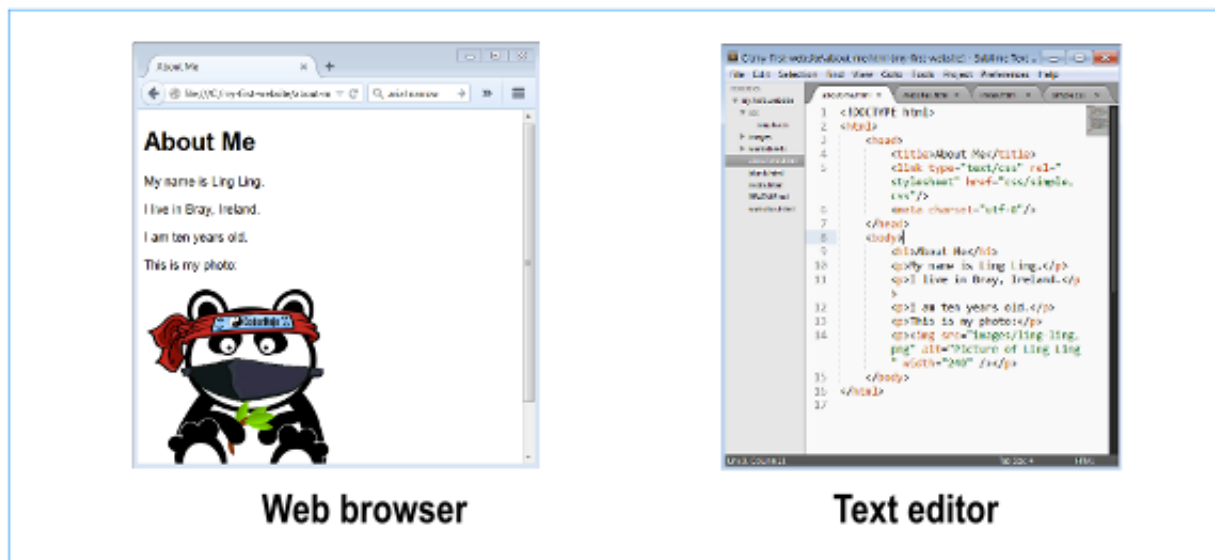
- 1 Download the HTML code from http://kata.coderdojo.com/wiki/My_First_Website
Go into the **my-first-website** folder and **open** the **about-me.html** file. It opens in your web browser.



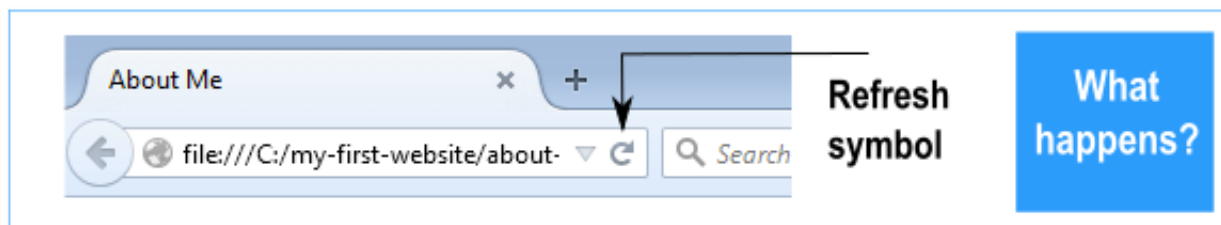
- 2 Now open the same file in a plain text editor such as Notepad ++ or Sublime Text



- 3 Arrange the browser window and the text editor so they are alongside one another on your screen



- 4 Change the text in the page so that it's about you instead of Ling Ling
- 5 After you have changed some text, save the file in your text editor (File > Save) and then refresh the web page in your browser window. To do this click the refresh symbol (or press F5).



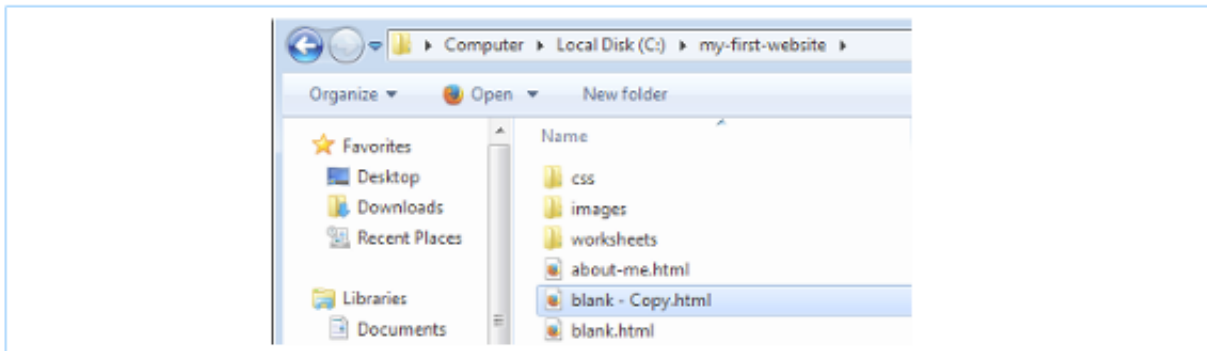
- 6 Now put in a new paragraph of text. This needs to go between the `<p>` and `</p>` tags. Like this:
`<p>I am learning how to make a website at CoderDojo.</p>`
`<p>` is the start tag for a paragraph element, `</p>` is the end tag.

- 7 Again, **save** and **refresh**.

You can see now that a web page is just text typed into a text file. How the text appears on the web page is controlled by the tags.

What happens if you use **h1** or **h2** elements instead of **p**?
 What happens if you put some words inside **strong**? Like this:
`<p>My name is Ling Ling</p>`

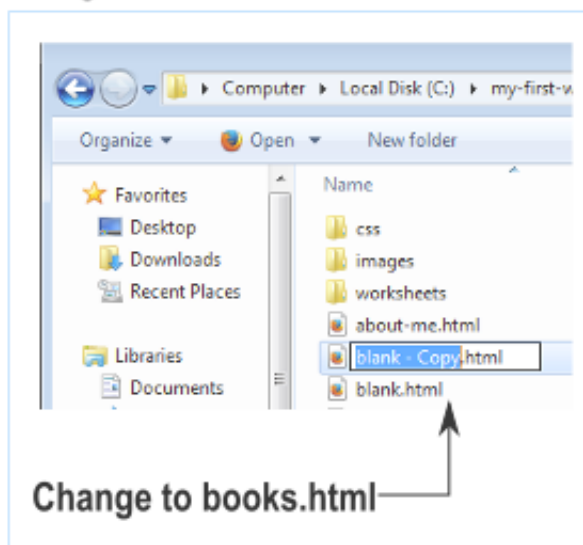
- 1 Go into the **my-first-website** folder and copy the **blank.html** file.
To copy the file, select it, then hold down the **CTRL** and **C** keys at the same time.
- 2 Paste the copy of the file into the same folder.
To paste the file, hold down the **CTRL** and **V** keys at the same time.
After you have pasted the file you will see this:



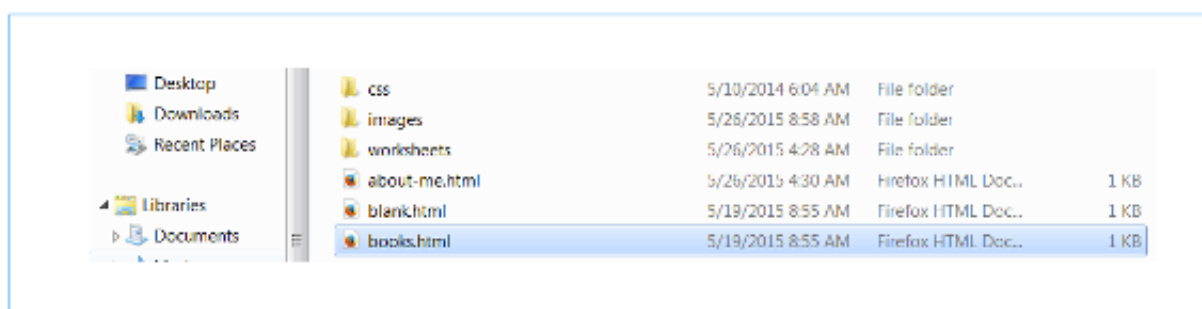
Now you have a copy of the **blank.html** file in the same folder. It is called **blank - Copy.html**

You'll make this copied file into a new page which will list your favourite books.

- 3 Click the file to select it and press **F2**. The **blank - Copy** part of the filename is highlighted
- 4 Change the name to **books.html**

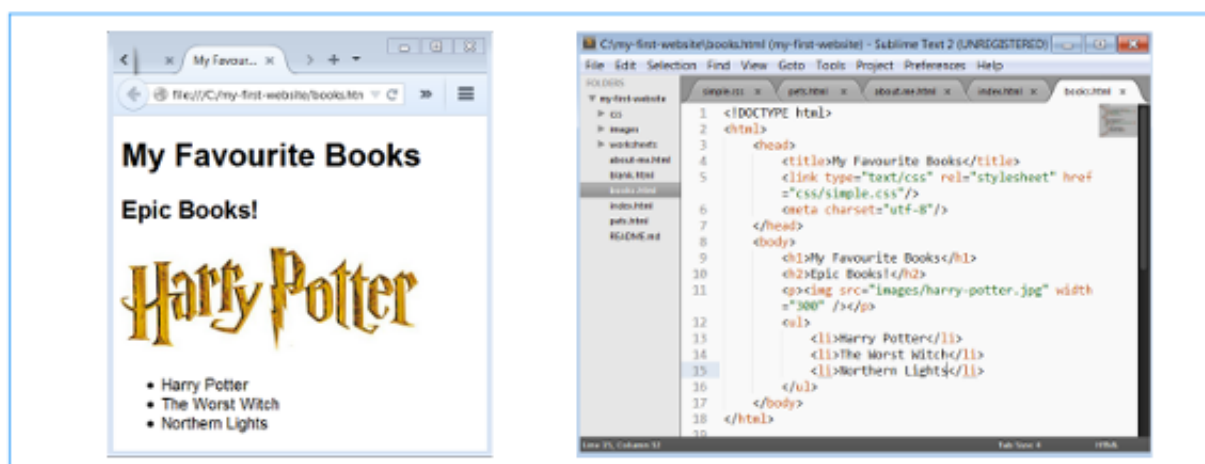


When you have renamed the file it will look like this:




You now have a web page named **books.html**. At the moment it is the same as the **blank.html** page.

- You'll need to open it and edit the text so that the page is now about your favourite books. Use the example below for ideas. Can you figure out how to download a picture (like the Harry Potter logo) for your top website and display it in the page?



Save the file every so often as you're working on it. To save press the **CTRL** and **S** keys at the same time.

That way you can keep your work even if your laptop battery gives out suddenly! Every time you save the file, refresh  the web page to see the effect of your most recent changes.

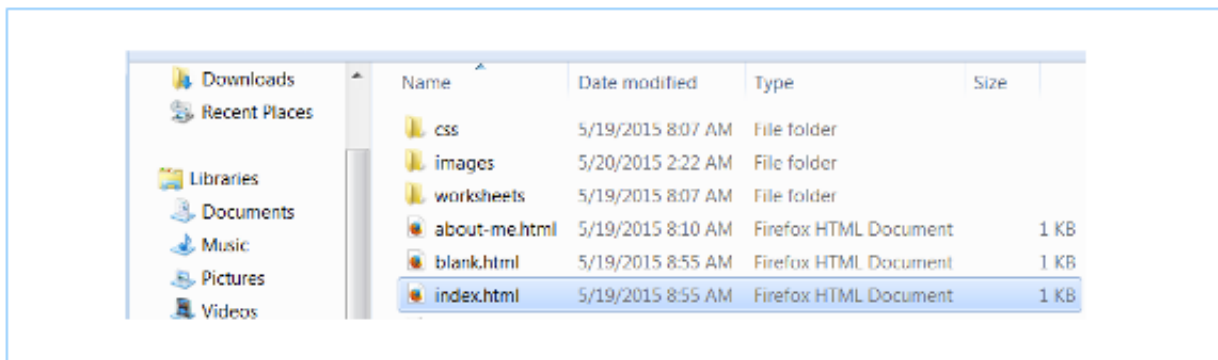
Can you see that we've added a list to the page?
The list has several **li** tags inside a **ul** element

What happens if you put the **li** elements inside an **ol** instead of a **ul** element?

- 1 Go into the [my-first-website](#) folder, copy the [blank.html](#) file (again!) and this time rename it to [index.html](#). [index.html](#) will be the home page of your site. Home pages are often given a file name of [index.html](#)

See card 2 for a reminder of how to copy and paste a file!

After you have pasted and renamed the file, you'll see this:



- 2 Edit [index.html](#) so that it's like the one in the screenshot below (except it has your name in it instead of Ling Ling):

```
<!DOCTYPE html>
<html>
  <head>
    <title>Home</title>
    <link type="text/css" rel="stylesheet" href="css/simple.css"/>
    <meta charset="utf-8"/>
  </head>
  <body>
    <h1>Ling Ling's website</h1>
    <p>This is my website</p>
    <p>Click on a link to visit a page:</p>
    <ul>
      <li><a href="about-me.html">About Me</a></li>
      <li><a href="books.html">My Favourite Books</a></li>
      <li><a href="family.html">My Family</a></li>
      <li><a href="pets.html">My Pets</a></li>
    </ul>
  </body>
</html>
```

The index.html page has two links on it. If you click one of the links you are taken to one of the pages you made earlier.

A link has two main parts:

```
<a href="books.html">My Favourite Books</a>
```

↑ The name of the file you want to link to

```
<a href="books.html">My Favourite Books</a>
```

The text that appears on the web page ↑

Watch out when typing links - links need all the different pieces to appear so the browser can understand how to display them.

They always have the same pattern:

start and end <code>a</code> tags:	<code>Link text</code>
<code>href</code> attribute:	<code>Link text</code>
filename:	<code>Link text</code>
link text:	<code>Link text</code>

Notice how the `href` attribute puts quote marks " " around the filename and how it comes inside the `a` tag.

Your challenge...

Make three more pages and add the links from the home page.

Some suggestions (your own ideas welcome!):

Title	File Name
My Family	family.html
My Pets	pets.html
My Top Songs	songs.html
My Movies	movies.html
My Hobbies	hobbies.html
My Heroes	heroes.html

The index.html home page looks a bit plain. You can change how it looks with CSS!

Ling Ling's website

This is my website.

Click on a link to visit a page.

- [About Me](#)
- [My Favourite Websites](#)
- [My Family](#)
- [My Pets](#)

Ling Ling's website

This is my website.

Click on a link to visit a page.

- [About Me](#)
- [My Favourite Websites](#)
- [My Family](#)
- [My Pets](#)



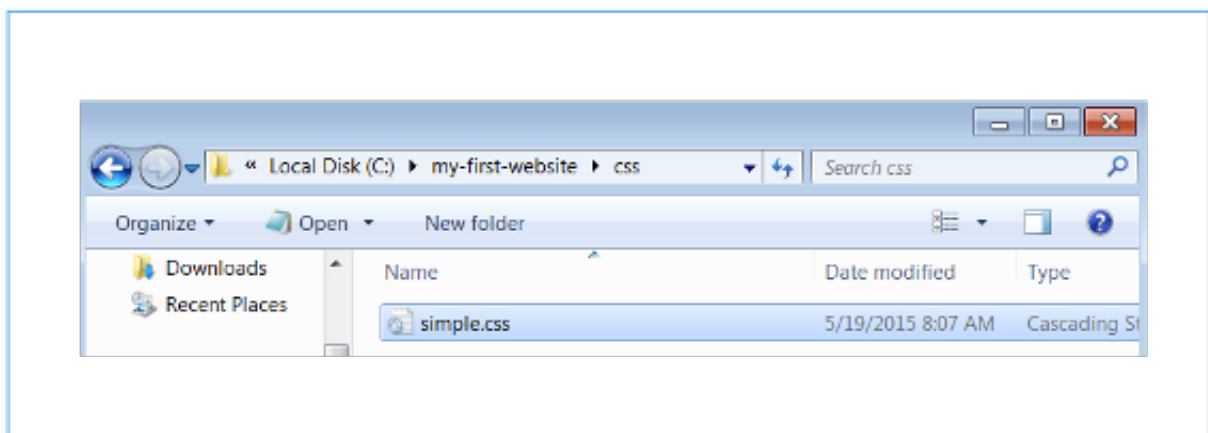
From this... To this... Using CSS!

A CSS file contains the recipe for how the web page appears.

It's a separate file from the HTML file and it's linked to the HTML file like this:

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

- 1 Open the [index.html](#) file in your text editor and locate the link (it's the fifth line down). What this link is telling you is that the appearance of the page is controlled by a stylesheet named [simple.css](#) which is located in the [css](#) folder.
- 2 Go into the [css](#) folder and locate the file:



- 3 Open [index.html](#) in your browser and open [simple.css](#) alongside it in your text editor, so that you can see them both side-by-side.

The [simple.css](#) file contains just one statement:

```
body {  
  font-family: sans-serif;  
}
```

The statement is called a rule and it tells the browser how to format everything inside the body element of the HTML file.

What this selector is saying is "set everything in the body element to a sans-serif font".

- 4 Modify the [body](#) tag by adding a declaration to include a background image, and add a new rule which says how to format the [h1](#) element. So the finished [home.css](#) file will look like this:

```
body {  
  font-family: sans-serif;  
  background-image: url('../images/ling-ling.png');  
}  
h1 {  
  padding: 12px;  
  background-color: black;  
  color: white;  
  border-radius: 10px;  
}
```

- 5 Add the declarations one at a time, saving as you go. Click refresh in your browser each time you add a new declaration so that you understand the difference the new rule makes.

Notice how the selectors contain a number of individual formatting rules. Each rule is ended by a semi-colon (;) and consists of a property (such as [color](#)) followed by a colon (:) and then a value (such as [white](#)). Notice, also, that rules are enclosed in curly braces ({ and }).

Now we're going to add links to all the pages we've created. This will allow us to navigate from one page to another no matter where we are in the site. Right now, the only links we have are on the [index.html](#) Home page.

To do this we'll open up the [index.html](#) file, copy the HTML code for the links and then paste it into the other files ([about-me.html](#), [books.html](#), [family.html](#) and so on). We'll also add a link back to the Home page.

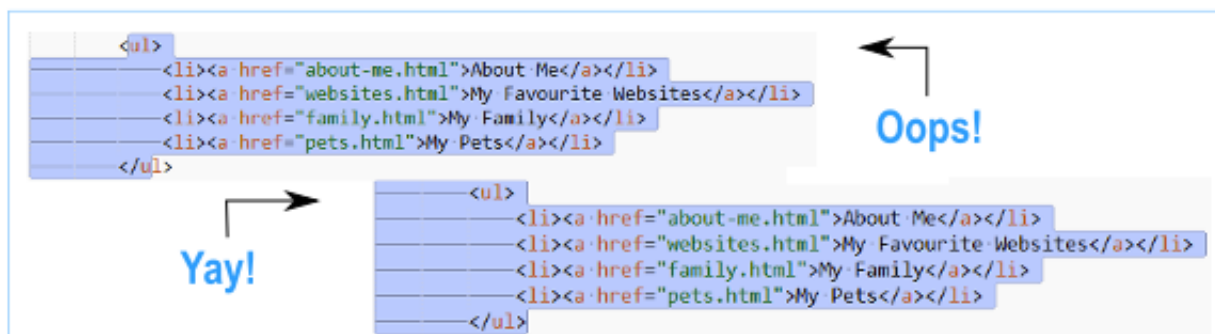
- 1 Open [index.html](#) in your text editor and select the HTML code for the links.

```
<p>Click on a link to visit a page:</p>
<ul>
  <li><a href="about-me.html">About Me</a></li>
  <li><a href="websites.html">My Favourite Websites</a></li>
  <li><a href="family.html">My Family</a></li>
  <li><a href="pets.html">My Pets</a></li>
</ul>
```

To select the code for the links:

- Click just before the **ul** opening tag
- Keep the mouse or touchpad button pressed down and then drag down, highlighting the text as you go
- Take your finger off the button when you get to the **ul** closing tag

- 2 When the links are selected, check that you've included *all* of the start and end tags for the **ul** element. Spot the difference below



```
<ul>
  <li><a href="about-me.html">About Me</a></li>
  <li><a href="websites.html">My Favourite Websites</a></li>
  <li><a href="family.html">My Family</a></li>
  <li><a href="pets.html">My Pets</a></li>
</ul>
```

Oops!

```
<ul>
  <li><a href="about-me.html">About Me</a></li>
  <li><a href="websites.html">My Favourite Websites</a></li>
  <li><a href="family.html">My Family</a></li>
  <li><a href="pets.html">My Pets</a></li>
</ul>
```

Yay!

Copy the links into the Clipboard memory by pressing **CTRL** and **C**

4 Open one of the web pages you made earlier in your text editor (in the example below, we'll use the About Me page ([about-me.html](#))).

5 Click just before the **h1** opening tag and paste the links by pressing **CTRL** **V**

```

<body>
  <h1>About me</h1>
  <p>My name is Ling Ling.</p>
  <p>I live in Bray, Ireland.</p>
  <p>I am ten years old.</p>
  <p>This is my photo:</p>
  
```

Click here →

You will end up with this:

```

<!DOCTYPE html>
<html>
<head>
  <title>About Me</title>
  <link type="text/css" rel="stylesheet" href="css/simple.css"/>
  <meta charset="utf-8"/>
</head>
<body>
  <ul>
    <li><a href="about-me.html">About Me</a></li>
    <li><a href="books.html">My Favourite Books</a></li>
    <li><a href="family.html">My Family</a></li>
    <li><a href="pets.html">My Pets</a></li>
  </ul>
  <h1>About me</h1>
  <p>My name is Ling Ling.</p>
  <p>I live in Bray, Ireland.</p>
  <p>I am ten years old.</p>
  <p>This is my photo:</p>
  <p></p>
</body>
</html>
  
```

Can you see where the links have been pasted?

If you want to, you can press the return key...



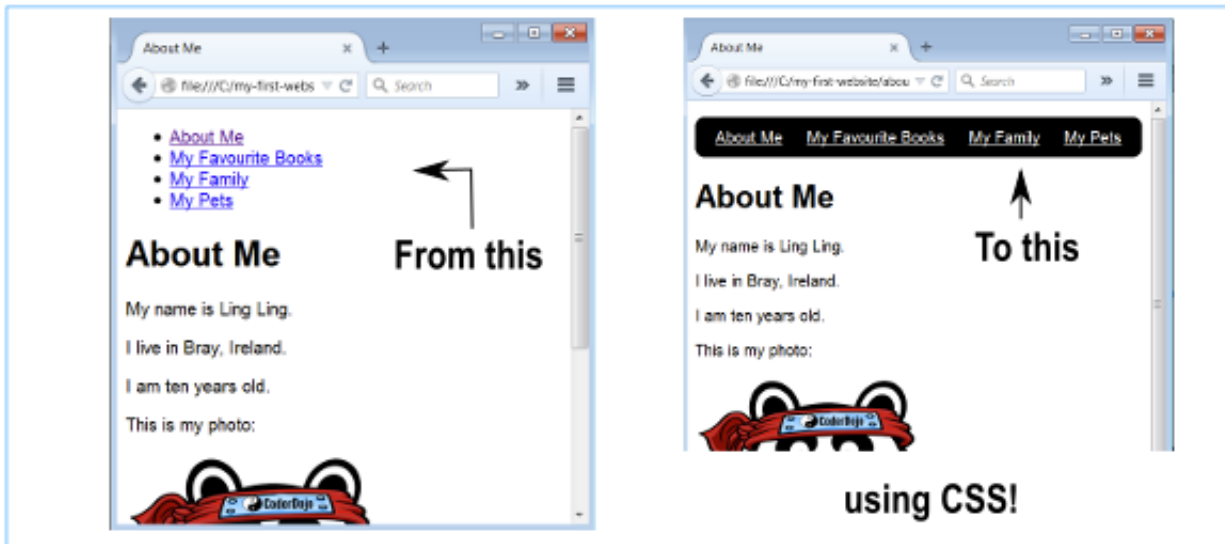
...to push the h1 element down to the next line and make your code look a little neater (although the browser doesn't care!)

6 Press **CTRL** **S** to save the file and then open it in your browser. You can see that the links appear above the h1 element. It would be better if they ran across the top in a row like the top menu bar you see on most websites. We'll do that on the next card.

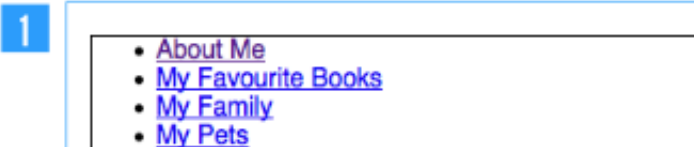
Can you do two things first?

- Add a link to the home page above the About Me link. Then copy the full set of links to all the other pages (except the home page as it already has them).
- Put the links in the same place before the h1 tag.



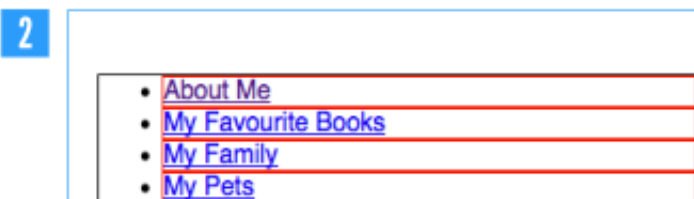


Do it one step at a time to see how the CSS works!



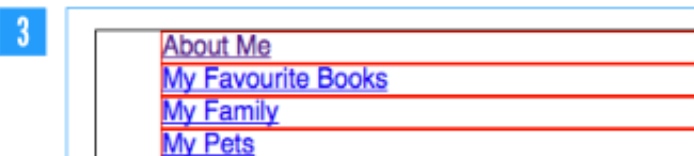
Add a solid border 1 pixel in width to the whole list

```
body {
    font-family: sans-serif;
}
ul {
    border-style: solid;
    border-width: 1px;
}
```



Add a solid red border 1 pixel in width to each of the list items

```
ul {
    border-style: solid;
    border-width: 1px;
}
ul li {
    border-style: solid;
    border-width: 1px;
    border-color: red;
}
```



Remove the bullet points from the list (lists have bullet points by default)

```
ul li {
    border-style: solid;
    border-width: 1px;
    border-color: red;
    list-style-type: none;
}
```

4

[About Me](#) [My Favourite Books](#) [My Family](#) [My Pets](#)

Make the list items appear alongside each other.

List items are block level elements

(appear on separate lines) by default.

```
ul li {
  border-style: solid;
  border-width: 1px;
  border-color: red;
  list-style-type: none;
  display: inline;
}
```

5

[About Me](#) [My Favourite Books](#) [My Family](#) [My Pets](#)

Put spacing of 10 pixels in width and height around the entire list.

```
ul {
  border-style: solid;
  border-width: 1px;
  padding: 10px;
}
```

6

[About Me](#) [My Favourite Books](#) [My Family](#) [My Pets](#)

Put a margin of 10 pixels on the left and 10 pixels on the right of each list item.

This leaves some space in between each list item.

```
ul li {
  border-style: solid;
  border-width: 1px;
  border-color: red;
  list-style-type: none;
  display: inline;
  margin-right: 10px;
  margin-left: 10px;
}
```

7

[About Me](#) [My Favourite Books](#) [My Family](#) [My Pets](#)

Make the border around the entire list have rounded corners with border-radius.

Give the whole list a black background.

```
ul {
  border-style: solid;
  border-width: 1px;
  padding: 10px;
  border-radius: 10px;
  background-color: black;
}
```

8

[About Me](#) [My Favourite Books](#) [My Family](#) [My Pets](#)

About me

My name is Ling Ling.

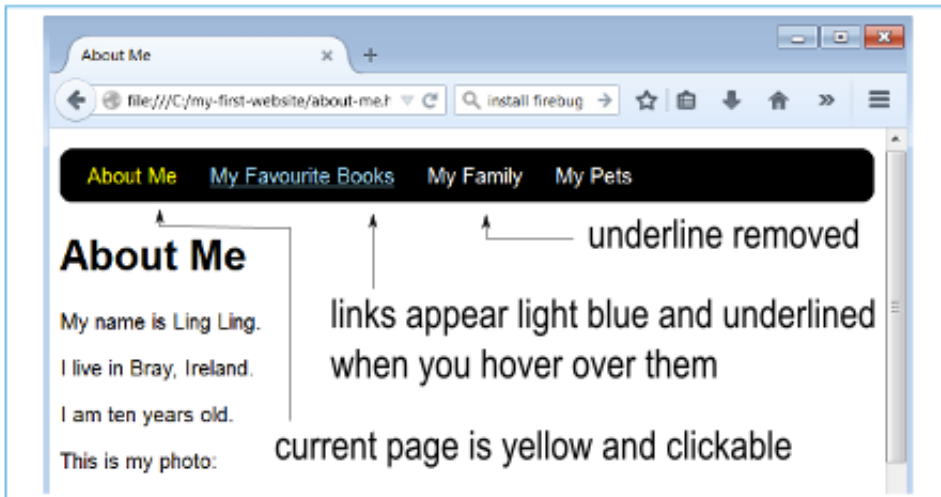
I live in Bray, Ireland.

I am ten years old.

This is my photo:

Remove the red borders and add white text.

```
ul li {
  border-style: solid;
  border-width: 1px;
  border-color: red;
  list-style-type: none;
  display: inline;
  margin-right: 10px;
  margin-left: 10px;
}
ul li a {
  color: white;
}
```



To style to your links like those above you'll need to edit the HTML files and the CSS file.

- In each of your HTML files remove the one link from the menu that leads to same page. For example, in the [about-me.html](#) page you will remove the `` link, in the [books.html](#) page you will remove the `` link.

File	Menu code before editing	Menu code after editing
about-me.html	<pre> About me My Favourite Books My Family My pets </pre>	<pre> About me My Favourite Books My FamilyMy pets </pre>
books.html	<pre>My Favourite Books</pre>	<pre>My Favourite Books </pre>

Do this for all the files in your site (for example [family.html](#) and [pets.html](#)). Notice that after you make this change the menu entry is no longer clickable.

2 Add a `class` attribute with a value of `selected` to the same `li` tags:

File
about-me.html

Menu code before editing

```
<ul>
<li>About Me</li>
<li><a
href="books.html">My
Favourite Books</a></
li>
<li><a
href="family.html">My
Family</a></li>
<li><a
href="pets.html">My
Pets</a></li>
</ul>
```

Menu code after editing

```
<ul>
<li class="selected">About
Me</li>
<li><a
href="books.html">My
Favourite
Books</a></li>
<li><a
href="family.html">My
Family</a></li>
<li><a href="pets.html">My
Pets</a></li>
</ul>
```

This doesn't do anything on its own (refresh in the browser and the menu still looks the same). *But*, adding the class attribute allows you to pick out the link in the CSS file and colour it yellow. Here's how:

File
simple.css

Menu code before editing

```
ul li a {
    color: white;
}
```

Menu code after editing

```
ul li a {
    color: white;
    text-decoration: none;
}
```

3 Refresh and you'll see the selected link change to yellow. What you've done is added a new rule which applies only to `li` elements with a class attribute of `selected`. Notice the syntax in the CSS file - instead of specifying `li` in the rule, you specify `li.selected`

4 Remove the underline from the menu links:

File
simple.css

Menu code before editing

```
ul li a {
    color: white;
}
```

Menu code after editing

```
ul li a {
    color: white;
    text-decoration: none;
}
```

5 Make the links light-blue and show the underline when you hover over them by adding a new rule to the CSS. This rule defines what will happen when you hover over an `a` element.

```
ul li a:hover {
    color: #99DDFF;
    text-decoration:underline;
}
```

Note that `#99DDFF` is the code for the light blue colour (in hexadecimal).

If we have a second unordered list within our page it will be formatted in the same way as the top menu - unless we can distinguish between them.

The web page below shows what it looks like when you add a second unordered list to the page.

Rules applying to ul

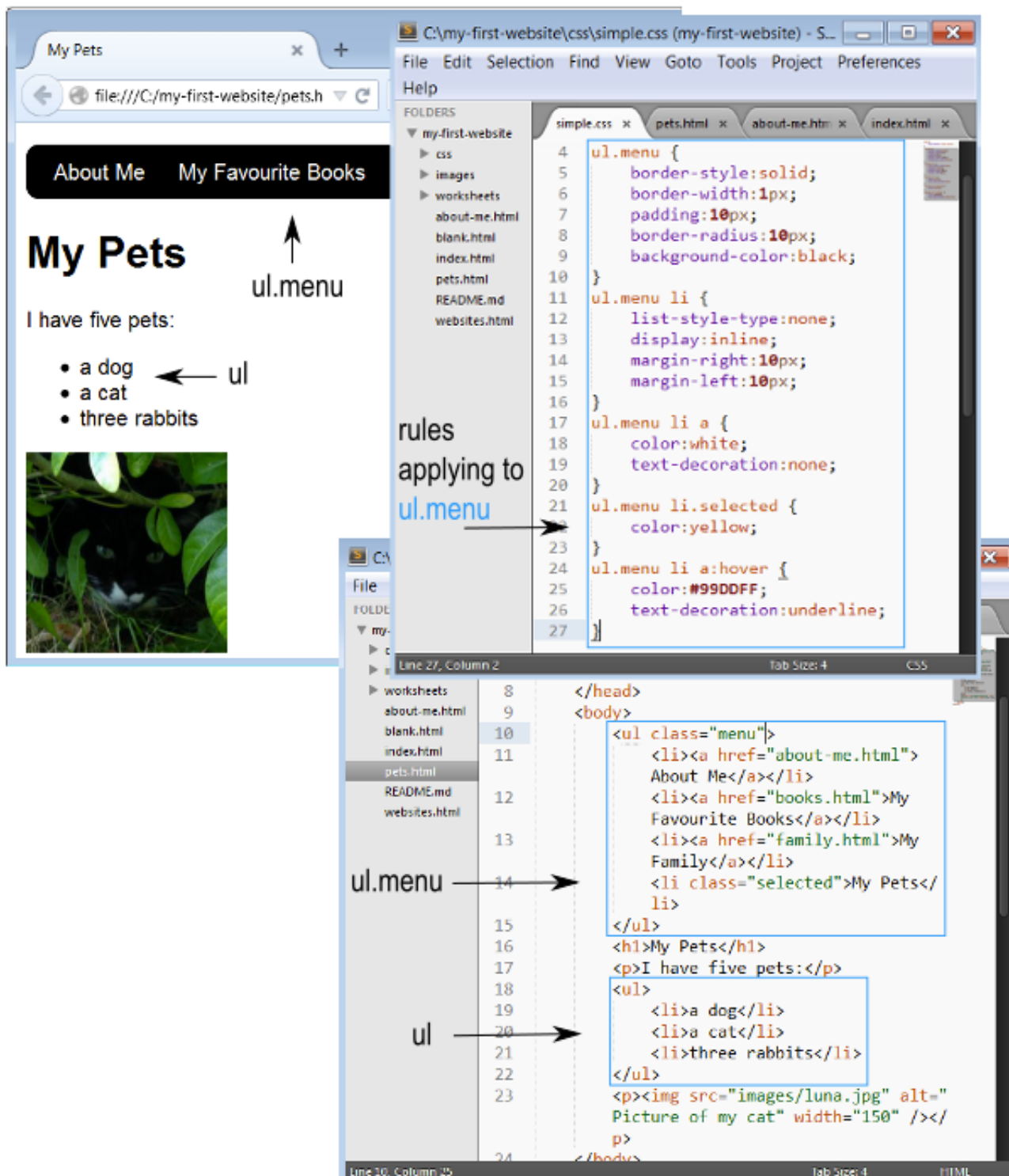
```
4 ul {
5   border-style:solid;
6   border-width:1px;
7   padding:10px;
8   border-radius:10px;
9   background-color:black;
10 }
11 ul li {
12   list-style-type:none;
13   display:inline;
14   margin-right:10px;
15   margin-left:10px;
16 }
17 ul li a {
18   color:white;
19   text-decoration:none;
20 }
21 ul li.selected {
22   color:yellow;
23 }
24 ul li a:hover {
25   color:#99DDFF;
26   text-decoration:underline;
27 }
```

```
<body>
<ul>
<li><a href="about-me.html">About Me</a></li>
<li><a href="books.html">My Favourite Books</a></li>
<li><a href="family.html">My Family</a></li>
<li class="selected">My Pets</li>
</ul>
<h1>My Pets</h1>
<p>I have five pets:</p>
<ul>
<li>a dog</li>
<li>a cat</li>
<li>three rabbits</li>
</ul>
<p></p>
</body>
</html>
```

We need to identify the menu by giving it a class attribute.

```
<ul class="menu"> ... </ul>
```

Once we have done this we can modify the CSS stylesheet so that the menu formatting rules are only applied to the `ul` elements which have a `class` selector with a value of `menu`. Here's how:



The screenshot shows a web browser on the left and two code editors on the right. The browser displays a page titled "My Pets" with a navigation menu at the top containing "About Me" and "My Favourite Books". Below the menu is a section titled "My Pets" with the text "I have five pets:" followed by a list: "a dog", "a cat", and "three rabbits". An image of a cat is shown below the list. Annotations with arrows point from the browser to the code editors.

The top code editor is a CSS file named `simple.css`. It contains the following CSS rules:

```

4  ul.menu {
5      border-style:solid;
6      border-width:1px;
7      padding:10px;
8      border-radius:10px;
9      background-color:black;
10 }
11 ul.menu li {
12     list-style-type:none;
13     display:inline;
14     margin-right:10px;
15     margin-left:10px;
16 }
17 ul.menu li a {
18     color:white;
19     text-decoration:none;
20 }
21 ul.menu li.selected {
22     color:yellow;
23 }
24 ul.menu li a:hover {
25     color:#99DDFF;
26     text-decoration:underline;
27 }

```

The bottom code editor is an HTML file. It shows the following HTML structure:

```

8  </head>
9  <body>
10 <ul class="menu">
11 <li><a href="about-me.html">
12 About Me</a></li>
13 <li><a href="books.html">My
14 Favourite Books</a></li>
15 <li><a href="family.html">My
16 Family</a></li>
17 <li class="selected">My Pets</
18 li>
19 </ul>
20 <h1>My Pets</h1>
21 <p>I have five pets:</p>
22 <ul>
23 <li>a dog</li>
24 <li>a cat</li>
25 <li>three rabbits</li>
26 </ul>
27 <p></
29 p>
30 </body>

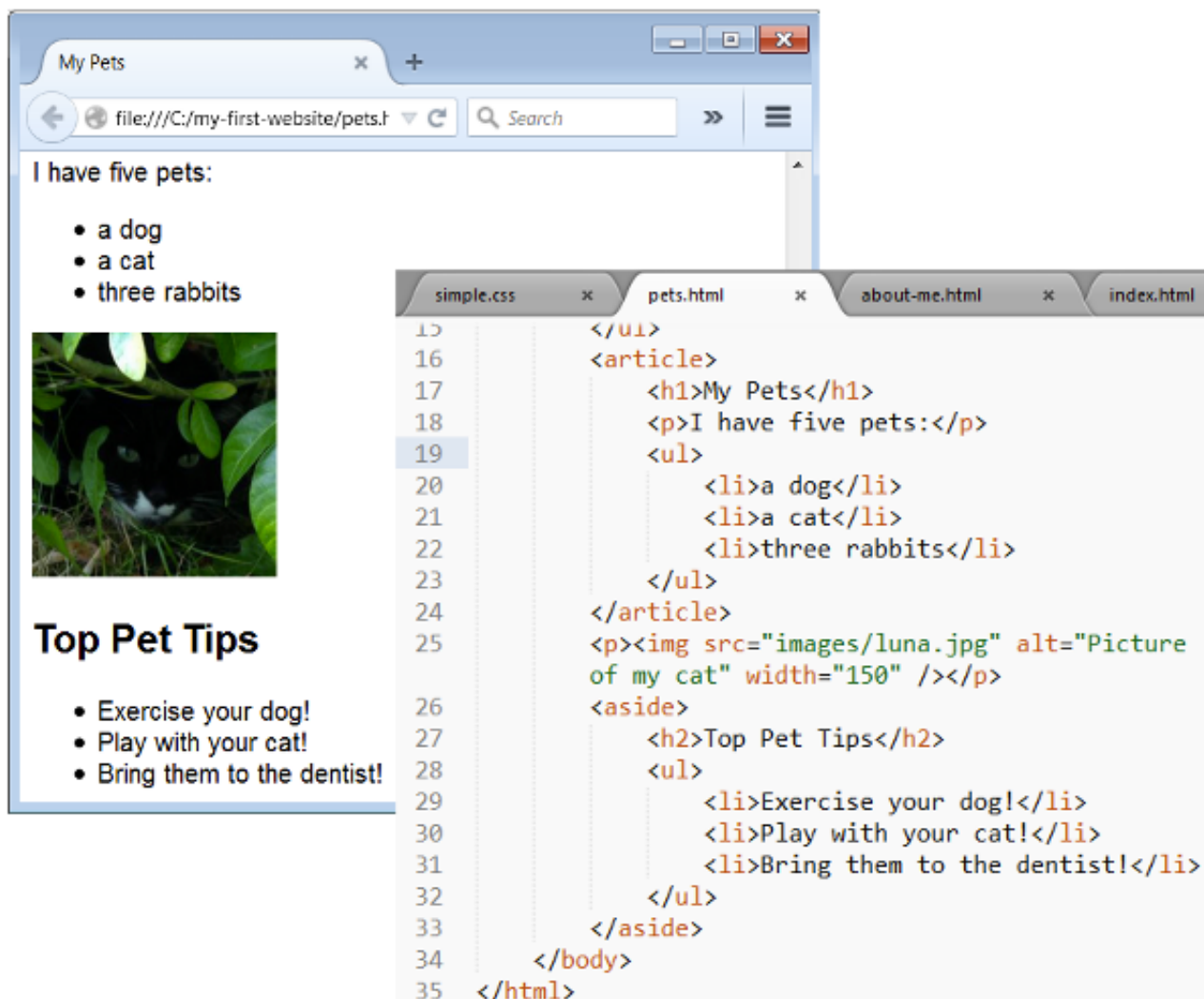
```

Annotations in the browser and code editors identify the `ul.menu` class and the `ul` element.

Responsive websites are sites which are viewable on all types of devices - desktop PCs, laptops, tablets, mobile phones. Responsive websites do this by changing the size of the elements on a web page to suit the amount of space on the screen which is available.

In order to do this, responsive sites use the CSS `float` property. We'll use the `float` property in this challenge.

First we'll add a new block of text on our pets page which contains our top tips. We'll put these tips inside an `aside` tag. We'll also put the information about our pets inside an `article` tag.



The screenshot shows a web browser window titled 'My Pets' displaying the rendered page. The page content includes:

- A heading: **I have five pets:**
- A list of items:
 - a dog
 - a cat
 - three rabbits
- An image of a white cat (Luna) sitting in green foliage.
- A heading: **Top Pet Tips**
- A list of tips:
 - Exercise your dog!
 - Play with your cat!
 - Bring them to the dentist!

Below the browser window, a code editor shows the HTML code for the 'pets.html' file, with line numbers 15 to 35:

```

15     </ul>
16     <article>
17         <h1>My Pets</h1>
18         <p>I have five pets:</p>
19         <ul>
20             <li>a dog</li>
21             <li>a cat</li>
22             <li>three rabbits</li>
23         </ul>
24     </article>
25     <p></p>
26     <aside>
27         <h2>Top Pet Tips</h2>
28         <ul>
29             <li>Exercise your dog!</li>
30             <li>Play with your cat!</li>
31             <li>Bring them to the dentist!</li>
32         </ul>
33     </aside>
34 </body>
35 </html>

```

Then add the following selectors to your CSS file. Note the `float: left` property!

```

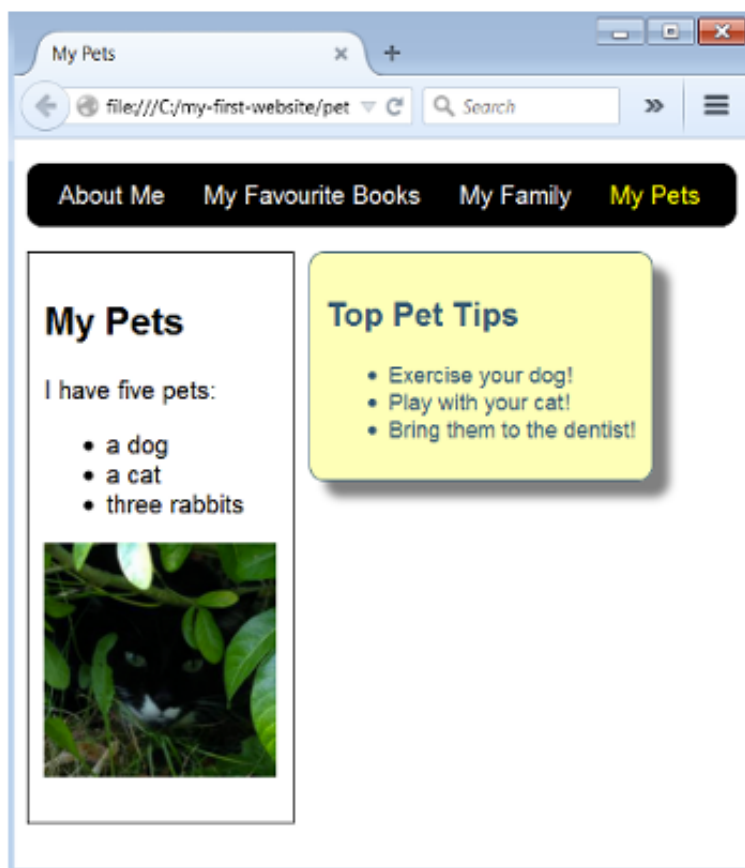
article {
  float: left;
  border-style: solid;
  border-width: 1px;
  margin-right: 1em;
  margin-bottom: 1em;
  padding: 1em;
}
aside {
  float: left;
  border-style: solid;
  border-width: 1px;
  padding: 1em;
  border-radius: 1em;
  background-color: #FFFFC0;
}

```



Challenge!

Can you modify the styling of the page so that it looks like this?



Hints:

```

box-shadow: 10px 10px 5px gray;
color: #2F5670;
list-style-type: square;
font-size: 14px;

```

Experiment with these CSS properties!

```

opacity: 0.5;
transform: rotate(30deg);

```

```

aside:hover {
  transition:
    1s ease-in-out;
}

```

```

border-style:dotted;

```

We're going to change our My Favourite Books page so that it shows a table listing our top five books and showing marks out of ten for each of the books.

Here's how it will look:



- 1 Start by adding the following selectors to your [simple.css](#) file

```
table, th, td {
    border: 1px solid white;
    border-collapse:
    collapse;
}
tr {
    background-color:silver;
}
th, td {
    vertical-align: top;
    padding: 0.5em;
    text-align: left;
}
```

- 2 Add the table to your books.html page.

Here is some code to get you started:

```
<h2>Top Three Books</h2>
<table>
  <tr>
    <th>Position</th>
    <th>Name</th>
    <th>Rating</th>
  </tr>
  <tr>
    <td>1</td>
    <td>Harry Potter</td>
    <td>9/10</td>
  </tr>
```

[you'll need to add the rest of the rows!]

```
</table>
```

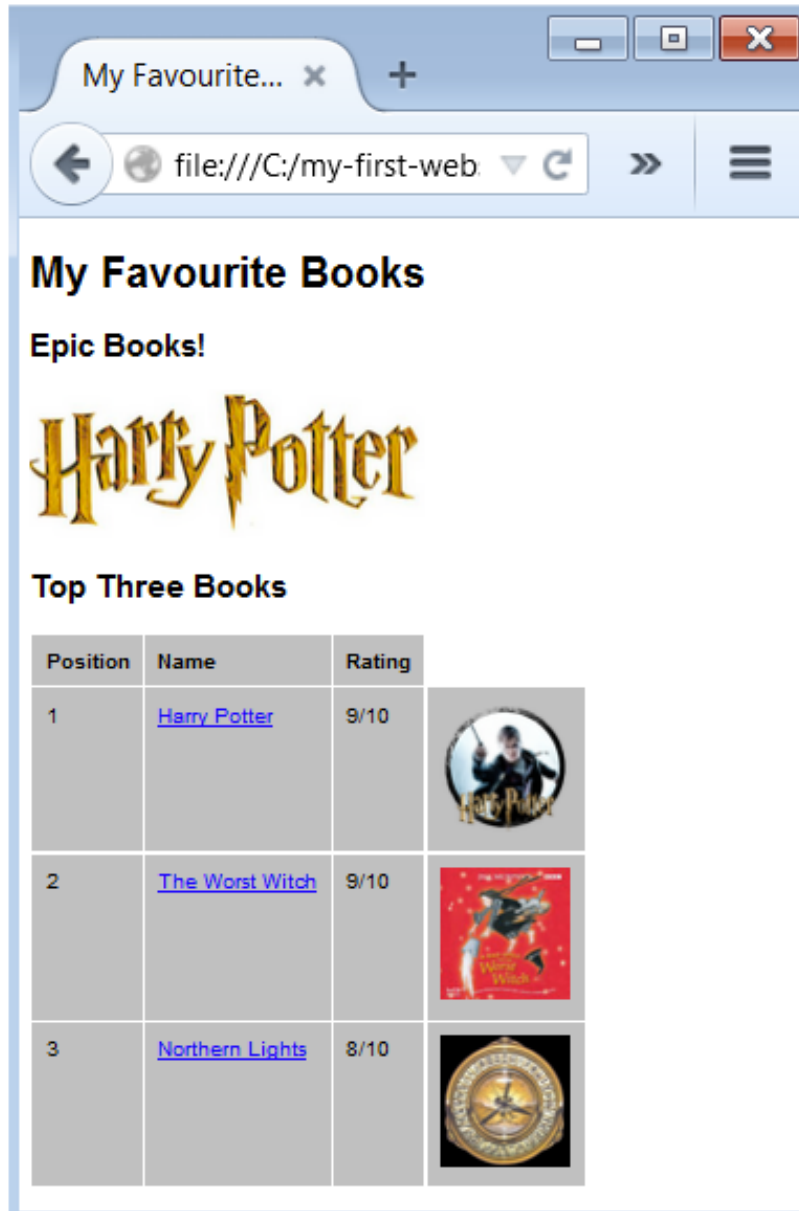
Can you do the rest? Add more than three if you like. Make them your own favourite books.

Did you know?

tr stands for table row
th stands for table heading
 and
td stands for table data

Challenges!

- 1 Can you add a column showing icons for each of the games?
Tip - you can use this website to find images: www.googlejunior.com
- 2 Can you make the name of the book into a link to buy it?



One more tag!

Try using the marquee tag to make text move across the screen like this:

```
<marquee>Epic Books!</marquee>
```

Congratulations! You've created your first website!