PROJECT 2
BUILDING BASIC WEB PAGES

□	■	□	□	□	□	□
HOW HTML WORKS	BUILDING BASIC WEBPAGES	MAKING YOUR FIRST WEBSITE	STYLING WEBPAGES	STYLING AND FORMATTING TEXT	POSITIONING ELEMENTS USING CSS	DESIGN & BUILD YOUR OWN WEBSITE

Track your progess by ticking off the boxes below:

Introduction: You know a few HTML tags, so it's time to make your first page! Let's get started right away.

Step 1: Open the starting document

1. Open a **text editor**. □
2. Open the **`about_me.html`** file. It contains only a little bit of HTML code to get you started, but you will write the rest yourself. □

Step 2: Make a page about yourself

About mistakes

Mistakes often happen. It's very easy to make them in HTML because you have to remember to close each tag, and the opening and closing tag are slightly different. Let's try making some mistakes to see how browser tries to make sense of our code even if we haven't written it perfectly.

1. Let's pick the list of things we like as an example. One of the mistakes that happen often is forgetting the **closing tag**, so let's remove the **</ul>** to see how it affects the page. **Save** the file and **refresh** it in the browser. □

What happened? Some things below the list moved a little bit to the right. If you inspect the page with **X-Ray Goggles** you can see that things which followed the list now nest inside it, which is why they have moved to the right. After we removed the closing tag the browser simply doesn't know the list of items has ended.

Add the closing **</ul>** tag back in and **save** it. Now when you refresh the page the rest of tags aren't inside **<ul>** anymore. □

2. Tags need to be spelt correctly for browser to understand them. What would happen if we misspelt something?

   Find the **<h1>** tag. Let's change it to **<d1>**. Save the document and refresh it in the browser. □

What happened? Since the browser doesn't know what you mean by this tag it can no longer tell that it's a heading and so it doesn't use a larger text to show how important this piece of text is.

Track your progess by ticking off the boxes below:

3. Find one of the **<img>** tags. We've just tried misspelling the tag name and the browser wasn't sure what to do with it. But what if we misspell the attribute? □

   Inside **<img>** tag we have the **href** and **alt** attributes:

```
<img href="kitten.jpg" alt="This is a kitten." />
```

   Try changing **href** to something else. □

Now SAVE your file and VIEW it in your browser □

Oh no! The kitten is gone! Suddenly, the browser no longer knows where to look for the picture to display - it is looking for the file name inside the `href` attribute, which is no longer there.

   Change it back to **href** so we can keep looking at the kitten. □

4. Now remove the second quote (**"**) from the **alt** attribute of this image: the one after the text, so you end up with this: □

```
<img href="kitten.jpg" alt="This is a kitten. />
```

Now SAVE your file and VIEW it in your browser □

The next tag disappeared. Why? The browser will think that everything after **alt="** and before the next quote (**"**) is the additional text for this image, including the end of the image tag and the next opening tag.

5. Fix it again by adding a quote after the **'alt'** text. □

We've made some common mistakes together, and sometimes a simple error might make the browser struggle to understand what we mean. But most of the time it will try to show us something anyway, so when we've changed the header tag to something else it didn't understand this piece of text was a heading, but it still showed us the text. So it's a little bit understanding, but some mistakes can make it very confused.

Track your progess by ticking off the boxes below:

Now SAVE your file and VIEW it in your browser □

Step 3: Create another page and link to it

1. Let's create another page. Open **about_me_page_2.html**. It has a little bit less code than the last page you were working with, but I'm sure you can figure out how to add new tags by now. □

Some hints and ideas:

- Add a heading that will serve as the title of this page.
- You could make this page about your pet, your favourite hobby or your friends and their hobbies.
- Add a list of things your pet likes, if your page is about the pet.

2. Are you done? Great! Let's now link the two pages you have created together. When we've been linking to parts of the same page, we could just point links to a specific **'id'** within a page, like this: □

```
<a href="#kitten">Click to see a kitten</a>
```

Which then took you to something like this:

```
<div id="kitten">
   <img src="kitten.jpg" alt="This is a kitten." />
</div>
```

To link to another page, we don't need to include the hash symbol (**#**), but instead we need to say which file we would like the link to take us to.

So to link from **about_me_page_2.html** to **about_me.html** write it like this: □

```
<a href="about_me.html">Go to About Me page</a>
```

Track your progess by ticking off the boxes below:

You can change the link text to something else, like the page title if you have changed it. To link back from **about_me.html** to **about_me_page_2.html** you would have to write it like so:

```
<a href="about_me_page_2.html">Go to my second page</a>
```

You can change the link text to something else, like the page title if you have changed it. To link back from **about_me.html** to **about_me_page_2.html** you would have to write it like so:

```
<a href="about_me_page_2.html">Go to my second page</a>
```

Congratulations! You have made your own website.

Now SAVE your file and VIEW it in your browser □

Step 4: Putting your website on the web (extra activity)

Now you have made your own site, you want to show it off, am I right?

If you simply copied the address of the web page from your browser and then sent it to someone, they wouldn't see it. That's because this address describes a place on your computer, and your friends don't have access to it. Even of they did, what if they wanted to look at it when your computer wasn't turned on?

Remember **servers** from the first session? **Servers** are computers that are always on and connected to the internet, and they are set up so people can visit websites that live on those computers.

To do that we will use **Cyberduck** - it's a program for moving files from your computer to a server.

1. Click **Open Connection**. □
2. Add the **server** name, **user** name and **password** as instructed by the CodeClub volunteer. □

Track your progess by ticking off the boxes below:

3. Click **Connect**. You will then see all the folders and files on the server - most likely the server will be empty, as you haven't added your files yet. □
4. Drag your website files from your computer into your server window. The uploading will begin. □
5. Once uploaded, you can visit your website at the address given to you by the CodeClub volunteer. □

Now SAVE your file and VIEW it in your browser □

Things to try

A). How could you link to another page on the web? (**Hint:** try adding **http://** and then the address of the site you want to link to) □

B). Similar to suggestion above, how would you add a picture from somewhere on the web instead of from your computer? (**Hint:** again, try adding **http://** and the address of the picture) □