



UIS Training Services

Web Authoring: CSS for Beginners

Practical Notes

```
@import url('https://fonts.googleapis.com/css?family=Noto+Serif|Roboto');
```

```
body { font-family: 'Noto Serif', serif, 'Roboto',  
sans-serif; font-size: 16px; font-weight: 200;  
color: #555; margin: 0; padding: 0; }
```

```
html { width: 100%; height: 100%; }  
page { max-width: 1200px; margin: 0 auto; position: relative; top: -8px; }
```

```
h1 { margin: 0 0 1em 0; font-family: 'Noto Ser;
```


Contents

Part 1

Getting Started	01
Open website and familiarise yourself with the content	01
Create a CSS file using Notepad++	02

Part 2

Importing and setting the font attributes	03
Adding the font attributes	03
What is an em?	04

Part 3

Linking the CSS file to the html pages.....	05
Link the header section of the html pages to the CSS file	05

Part 4

Adding a CSS style to the button.....	06
Style the buttons using CSS	06

Part 5

Styling the <header> section	07
Add a CSS style to the <header> section	07
Insert the Cambridge Shield SVG graphic	08
Style and position the <header> h1 and h3 text.....	09

Part 6

Creating the Navigation Bar	10
Style and position the elements of the Navigation bar	10

Part 7

The Person Biography section	13
Style and position the elements of the Person Biography section	13
Resize the person image using pixlr.com online photo editor	13
Style and width and height of the personbio section	17
Insert the person image and target the h2, hr and p attributes	18

Part 8

The Institutional Info section.....	20
Add CSS styling to the Institutional Info section.....	20
Style the Width, Height and Colour of the Institutional Info section	20
Style the elements within the Institutional Info section.....	21

Part 9

The Footer	24
Add CSS styling to the Footer	24

Part 10

The Curriculum Vitae CV section	26
Add CSS Styling to the CV section	26

Part 11

The My Interests section	30
Add CSS Styling to the My Interests section and table	30
Style the interests section	31
Position the table	31
Style the table title	31
Style the properties of the table	32
Style the table data and table header cells	32
Add a colour to alternating table rows	32
Add a hover effect to colour the table rows	33
Style the table header	33

Part 12

The Contact Details section	34
Add CSS Styling to the Contact Details section	34
Style the contact section	35
Position the text container box	35
Style the h2, hr and p elements	36

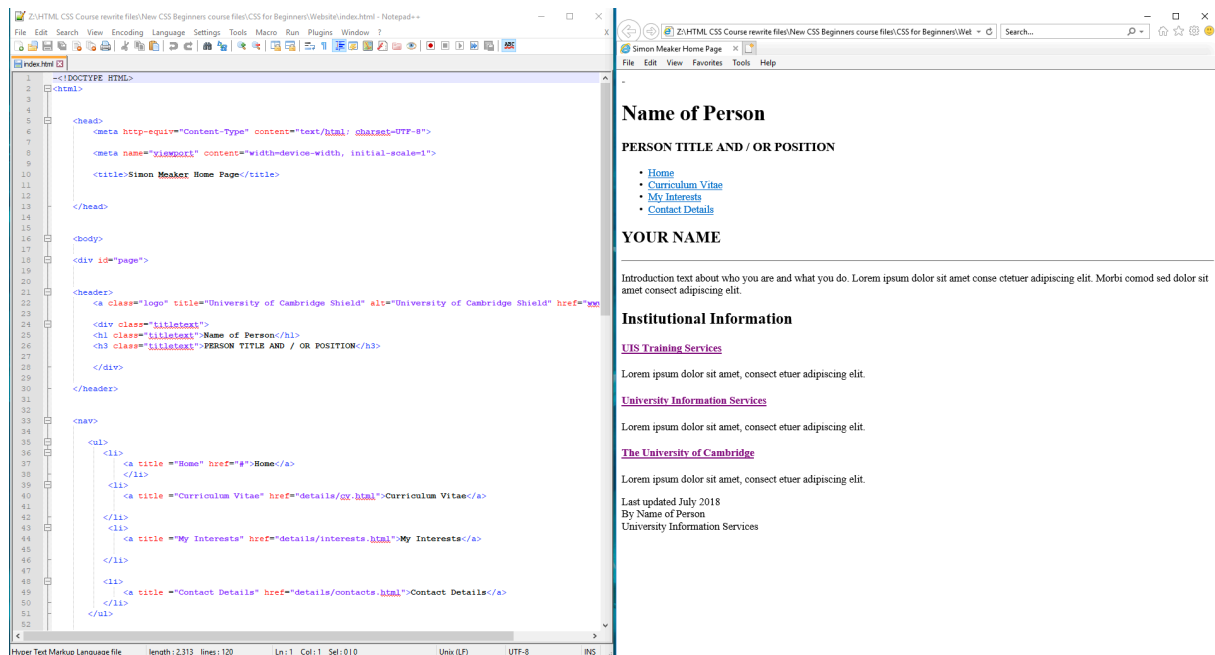
Part 13

The finished article	37
Your completed website	37

Part 14

Publish your website to the world	38
How to publish your website	38
Course Feedback Link	39
Please complete the course feedback before you leave	39

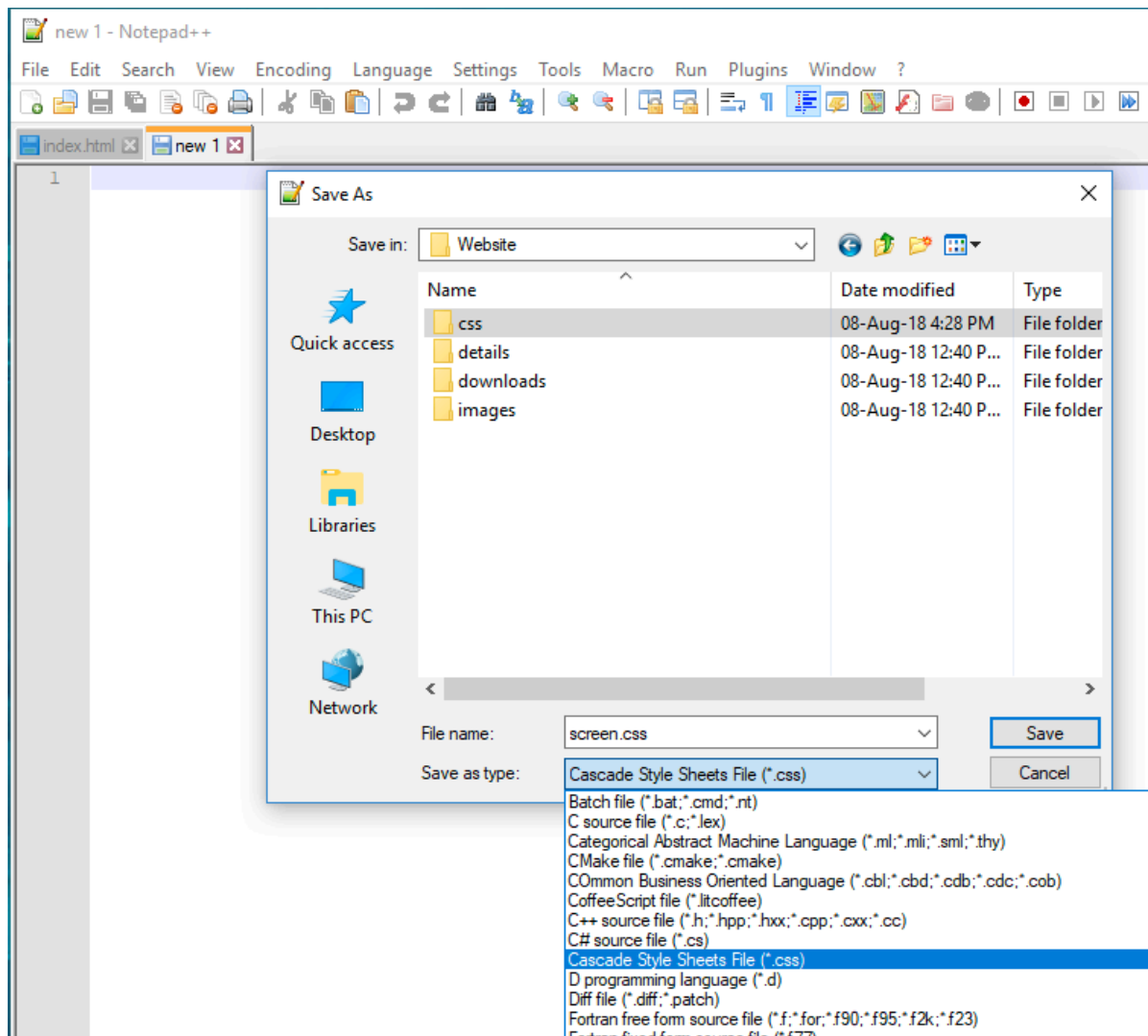
Open website and familiarise yourself with the content



Instructions

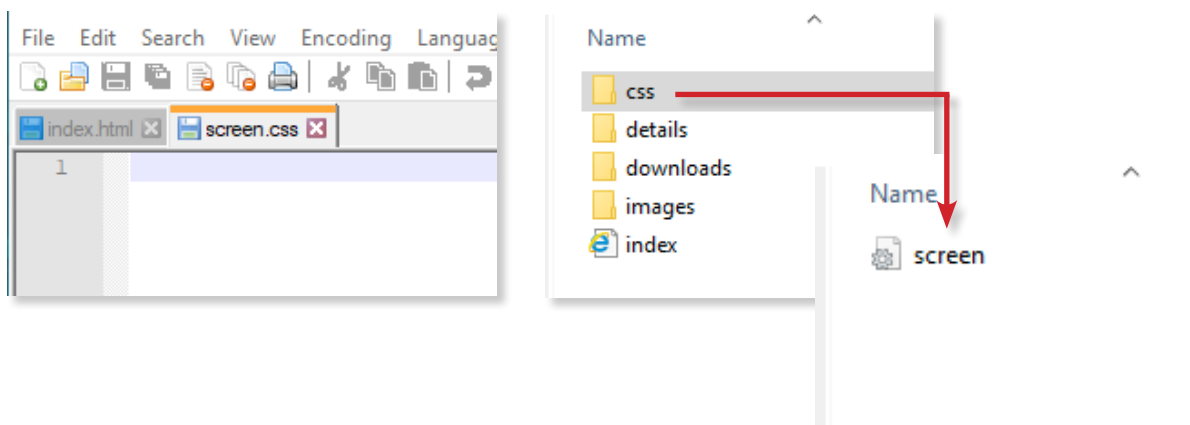
Using Notepad++ and Chrome web browser open the file *index.html* which can be located inside the website folder. To do this, first right click on the *index.html* file and choose to open it in Notepad++, or browse to it through Notepad++. To open the same file in Chrome right click and choose 'Open with' from the sub-menu and choose Chrome or open Chrome first and then browse to the file. Rearrange the two windows so they are side by side. Familiarise yourself with the basic website provided which was the finishing point of the 'HTML for Beginners course'. Your screen should resemble the screen-shot above.

Create a CSS file using Notepad++



Instructions

Open a new document in Notepad++ and save it as *screen.css* in a new folder called *css* within the *website* folder, as shown above and below.



Adding the font attributes

Now that you have your *index.html* and *screen.css* files open in Notepad++ we are now going to import the fonts and target the attributes of the fonts using CSS. To help with this the code has already been made for you.

Instructions

Using Notepad++ click **File** and **Open** and navigate to your course files folder, then into the **Scripts** folder and open the CSS file called **fonts.css**. Copy all the text and paste it into your *screen.css* file in Notepad++.

The *screen.css* file tab in Notepad++ should now be red, which is a warning to you that the page needs to be saved. Click the Save button on the Notepad++ tool bar. Close the **fonts.css** file.

Explanation

Study the code you have pasted into your *screen.css* file.

The top line which starts `@import url` imports the Google fonts when the web-page loads, as mentioned during the presentation of this course. This must be the first line of our CSS script.

The next line reads `body { font-family: 'Noto Serif', serif, 'Roboto', sans-serif; font-size: 16px;`

This sets the fonts which will be used for the body section. 16px is set as the base font size and this is the only time we will use pixels as the unit to state the size of our font. From here on in we will use '**em**'s. More about em's in a moment. The base font weight, colour, margin and padding have all been set here.

The next line down reads as follows:

```
#page {max-width: 1200px; margin: 0 auto; position: relative; top: -8px; }
```

This targets the *page div* and sets the maximum page width to 1200px.

Lastly we have 6 lines of code which target the h1 to h4 size fonts, the paragraph font and the anchor tag, parts of which are shown below:

```
h1 {margin: 0 0 1em 0; font-family: 'Noto Serif'; font-size: 2.5em;
h2 {margin: 0 0 .5em 0; font-family: Roboto; font-size: 2em;
h3 {margin: 0 0 .5em 0; font-family: Roboto; font-size: 1.3em;
h4 {margin: 0 0 1.5em 0; font-family: Roboto; font-size: 1.1em;
p {margin: 0 0 1em 0; font-family: 'Noto Serif'; font-size: 1.1em;
a {font-family: Roboto; font-size: 1.2em;}
```

Lets look more closely at what is incorporated into these lines of code, but firstly lets answer the question of what is an em.

What is an em?

An **em** is short for the letter **M**. A letter **M** is the biggest sized letter in a font so by using **em's** we can scale our font accordingly relative to the base font, which in our case is 16px. So a font size of 1em would give us 16px, whereas a font size of 2em would scale the font up to match a size of 32px or .5em would scale down to give us 8px. This is a more modern way of scaling fonts and adjusting space around fonts and works better with modern browsers, especially when it come to responsive web design.

So back to our code. Looking at the code in Notepad++ lets look at the h1 line in detail. Firstly we have the **margin** settings. There are four settings here which are 0 0 1em and 0. It's important to know that these run in a clockwise motion starting from the top or the 12 O'Clock position. So we have a margin of 0 above the text, a margin of 0 to the right of the text, a margin of 1em below the text to create some space and finally a margin of 0 to the left of the text.

After this we have **font-family** and we are using '**Noto Serif**'. We need the single quotes around this name only because there is a space between the names of **Noto** and **Serif**, otherwise the browser will get confused.

Next we have **font-size** and for the **h1** heading we are using **2.5em** which should scale up to give us a font size of 40px. $16 \times 2.5 = 40$.

Then we have **font-weight**. Font weight determines how thick or bold the font is. The higher the number the thicker the font. Our base font was 200, so 500 should come out nice and bold.

Lastly we have **line-height**. Line height is the space between rows, like single or double spacing. We are using a line height of **1.1em** which incorporated with our margin should give us some space around our **h1** headings.

Challenge

Have a look at the other lines of code for h2 onwards and see if you can figure out what is happening here. Any questions please ask.

Link the header section of the html pages to the CSS file

Instructions

In Notepad++ open your index.html web-page if not already open. Within the <head> section enter the code below:

The Code

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

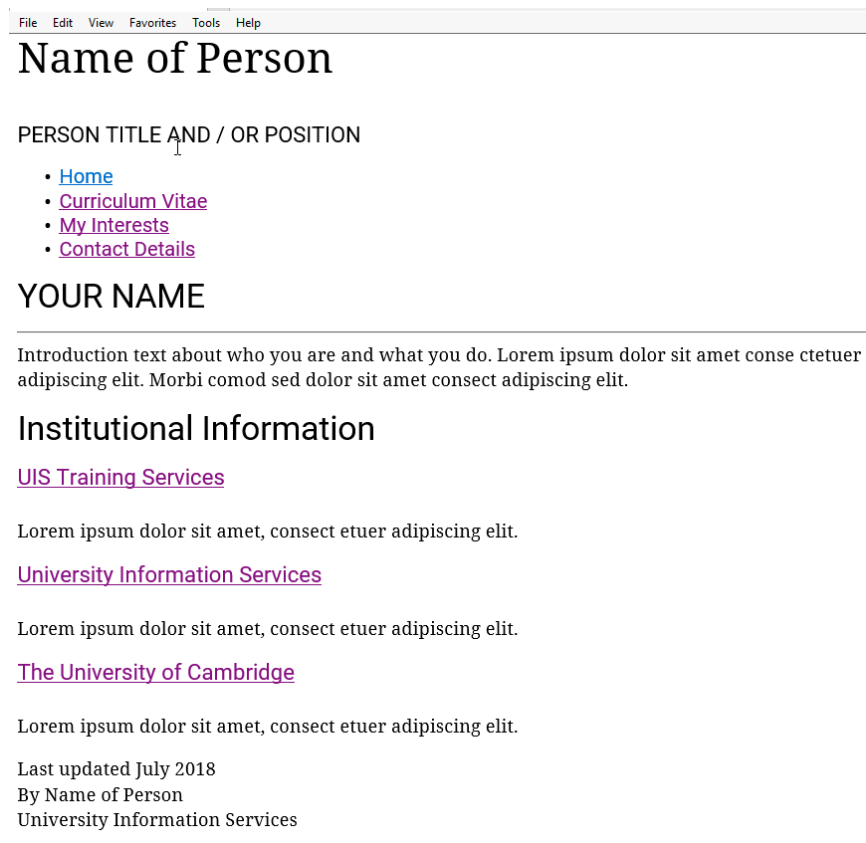
Save the index.html file and refresh the Home page. Do you understand the result?

Challenge

Add the same code to the head section for the remaining pages in the *details* folder remembering to change the path to the style sheet as needed using `../`
View the results for each of the four pages within the website.

Ask for help if you need it!

The font style of your web-page should now resemble the screen-shot below.



Style the buttons using CSS

Our *Curriculum Vitae* and *Contact Details* web-pages will each have a button which can be clicked to either produce an email or download a file. During the 'HTML for Beginners' course practical a HTML piece of code was inserted within each of these pages with a class of *btn* (short for button), as shown below:

```
<a class="btn" title="Download my CV" href="../downloads/cv.pdf">Download my CV</a>
```

In Notepad++ on our CSS style sheet we will now target the attributes of the button. Enter the code below on your screen.css document below the font attributes:

The Code

```
a.btn {font-size: 1.2em;
      text-decoration: none;
      color: #351a02;
      border: 1px solid #ca6a15;
      padding: 4px 15px;
      transition: background-color .5s; }

a.btn:hover {background-color: rgba(0,0,0,0.3)}
```

Note!

You can enter the `a.btn` code as separate lines as written above or all on one line. Spaces don't matter to CSS. I have written it above as separate lines to make it easier for you to read.

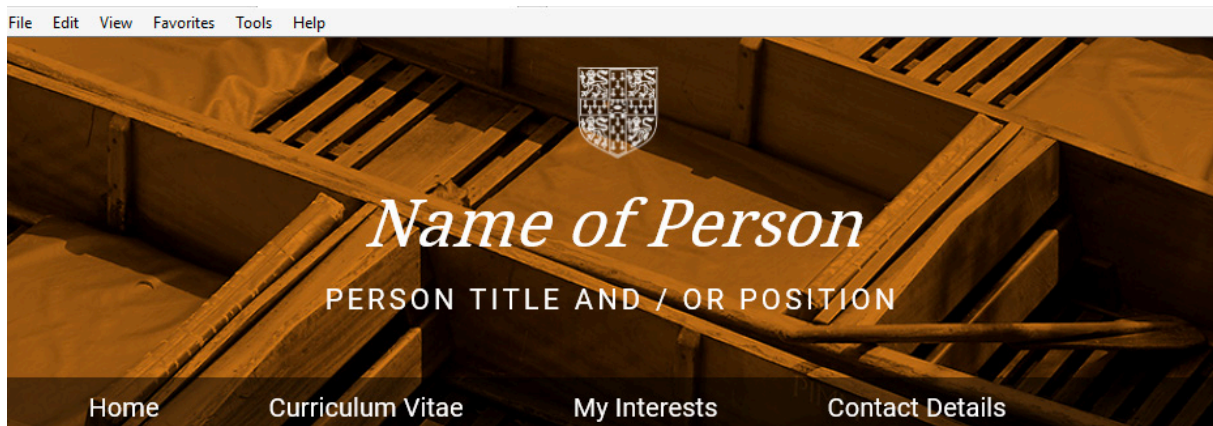
Explanation

The code we have added targets the font size of our buttons which will be (1.2em), the colour of the text inside the button, a brown colour in our case (#351a02), an orange border (#ca6a15) 1px in width, some padding 4px top and bottom and 15px left and right of the text and finally a transition effect. This makes the button more interesting as the user hovers the mouse over it. It will take .5 of a second to make the hover effect which is styled using the lower piece of code.

The mouse over effect is created by using the `a.btn:hover` command and by using the `rgba` attribute. The is short for Red, Green, Blue and Alpha. The alpha channel controls the transparency of the object. This enables us to darken the button when the mouse hovers over it.

The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque). We are using 0.3 which will darken the button a bit.

Add a CSS style to the <header> section



As shown above the Header section of our web-page will hold a background image of some Cambridge Punts, an SVG graphic of the University of Cambridge Shield which will also hold a hyper-link to www.cam.ac.uk, the h1 heading for your name and lastly the h3 heading for your title or position. If you wish to customise this for yourself you will need to modify the HTML <header> tags in all four pages to enter your own name and title / position if you haven't already done so in the 'HTML for Beginners course'. Lastly the navigation bar will be positioned over the top of the <header> at the bottom.

Instructions

First of all we are going to target the header section itself. We will set the height of the section and insert the background image. On your *screen.css* page open in Notepad++ create a couple of lines of space after the last line of code which started [a.btn:hover](#). Enter the code below either on one line or two separate lines:

The Code

```
header {height: 290px;  
background: url(../images/bannerpuntsbrown1200_290.png); }
```

Save the *screen.css* file and refresh your browser. The image of the punts should appear. If it doesn't ask for help before you continue as this needs to be correct before we proceed with styling the other attributes of the header. That's next!

Insert the Cambridge Shield SVG graphic

Instructions

Within your *images* folder there is a file called *camshield.svg*. We will now insert this over the top of our *punts* image and position it at the top of the *header* section. Enter the code below:

The Code

```
header a.logo {  
    position: absolute;  
    top: 15px; left: 50%; margin-left: -30px;  
    width: 60px; height: 69px;  
    background: url(../images/camshield.svg) no-repeat 0 0;  
    background-size: contain;  
}
```

Explanation

The first line targets the `<header>` and `<a>` tags with a class of *logo* which can be seen in the `<header>` section of our HTML page. We then target the position with an attribute of *absolute*. This means the element is positioned relative to its first positioned ancestor element which in our case is the `<header>` section filled with the image of the punt. Essentially this is a *div* within a *div*. We can then position the *camshield* *svg* graphic 15px from the top of the `<header>` section (the top of the web-page) and 50% from the left hand side of the web-page. It's important to use a percentage here so that the graphic remains centered even if the web-page is reduced in size width wise. To help it stay perfectly centered horizontally we use a trick here and add a *margin-left* attribute of -30px. 30px being half the width of our graphic which is 60px. We then enter the width and height of our graphic using the *width* and *height* attributes. The *background* attribute targets the location and name of the graphic and the additional *no-repeat* attribute makes sure the background-image is not repeated across the screen and will only be shown once. Lastly the *background-size* attribute of *contain* makes sure the image is fully visible on the web-page.

Save the *screen.css* file and refresh your web-page. The shield should appear.

Style and position the <header> h1 and h3 text

Now we are ready to target the h1 and h3 headings of our <header> section. Enter the code below into your screen.css page underneath the last line of code.

The Code

```
div.titletext {
    position: relative;
    width: 100%;
    top: 120px;
    text-align: center; }

h1.titletext {
    margin: 0 0 20px 0;
    font-style: italic; color: #fff; }

h3.titletext {
    color: #fff; letter-spacing: 3px;}
```

Explanation

The `div.titletext` attributes target the position of the container which will hold the h1 and h3 headings. If you want to see what is going on here you can add the following line of code `border: 1px solid white` in between `center;` and `}` like so:

```
text-align: center; border: 1px solid white }
```

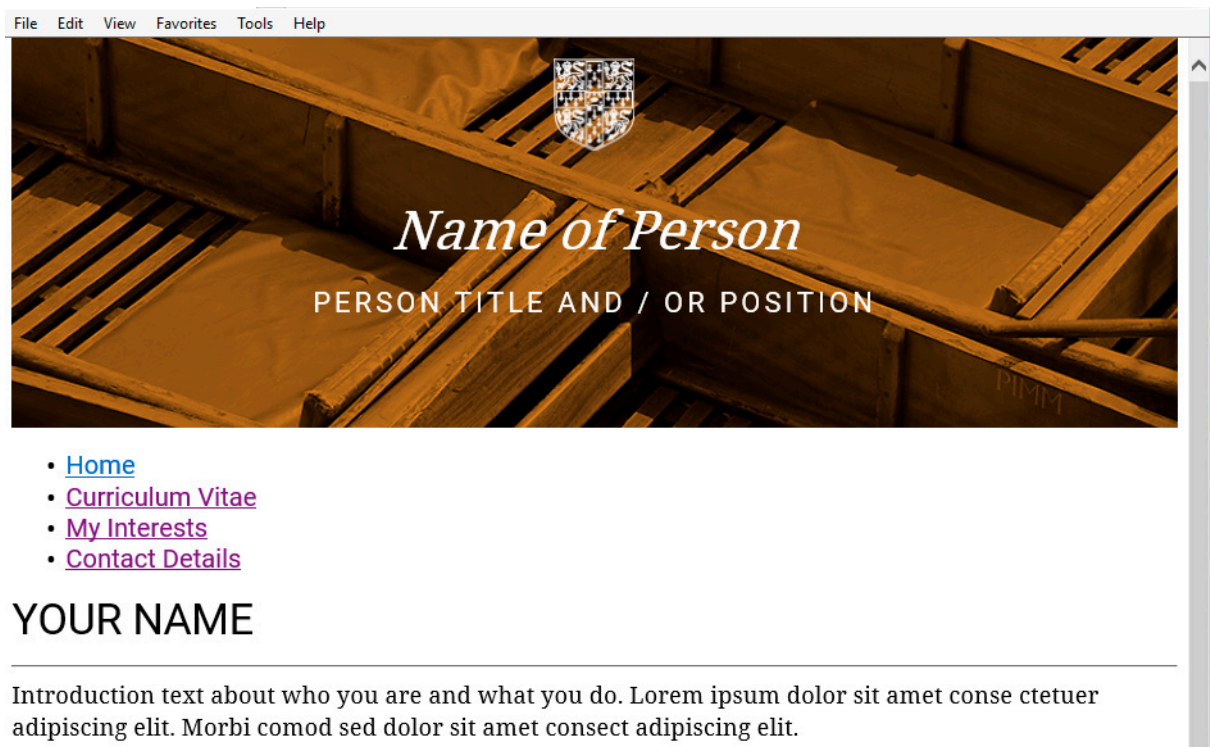
Adding a temporary border around an element is sometimes handy as it allows you to see what is happening to an element. The border can then be removed by deleting the extra line once you are happy the section is complete.

We use a position of relative here as we are going to position this div relative to its normal position on the page, which by default is the top left hand corner. We have given the container box a width of 100% to make it stretch across the page and also helps us center the text using the `text-align` attribute. It is positioned 120px from the top of the page.

The `h1` heading with a class of `titletext` is given a 20px margin underneath to make some room before the h3 heading. The font style will be *italic* and the hexadecimal colour code of `#fff` makes the text white. The h3 heading is also coloured white and the space between the letters is extended to add some additional styling to this attribute. The rest of the font styling is picked up from the text attributes we copied into our css document at the start of the practical. Remove the border if you added one and save and refresh to see the change.

Style and position the elements of the Navigation bar

Your web-page should now resemble the image below. If so, you can continue on with creating the Navigation bar, if not please ask for help.



Instructions

We are now going to create our Navigation bar from the unordered list of hyper-links within our HTML `<nav>` section. Enter the code below into your `screen.css` document. For this section I have given you the code one piece at a time followed by an explanation of what it does. Save and refresh as you go to view the changes.

The Code

```
nav {
    background-color: rgba(0,0,0,.50);
    width: 100%;
    position: absolute; top: 247px;
}
nav::after { content:''; display: block; clear: both; }
```

The `background-color: rgba(0,0,0,.50);` attribute makes the bar transparent but darker than the background image it will sit upon.

The `width: 100%` attribute sets the width to cover the entire page width and is then positioned using the `absolute` attribute 247px from the top of the page.

The `nav::after` attribute makes sure the bar carries on to the edge of the page, opposed to stopping after the last link word which in our case will be **‘Details’**.

```
nav ul {list-style: none; margin: auto; padding: 0 0 0 20px;}
```

The `nav ul list-style` attribute removes the bullets from the original list and the `padding:` attribute nudges the `nav` bar in a bit from the left edge of the page by 20px.

```
nav ul {list-style-type: none; }
```

The `nav ul list-style-type:` attribute also targets the unordered list and removes any margins and padding created from having a list in the first place.

```
nav ul li { float: left; }
```

The `nav ul li { float: left; }` attribute rearranges the list from a vertical line into a horizontal line and makes the block elements float next to each other.

```
nav ul li a {display: block; color: #fff; text-align: center; padding: 10px 20px; text-decoration: none; }
```

The `nav ul li a` CSS attribute targets the text of all four of the `` `<a>` HTML hyper-link tags in one go. The text is coloured white using the hexadecimal value of `#fff` and center aligned within each block. 10px of padding is added around the top and bottom of the text and 20px of padding left and right. To finish off text decoration is set to none.

Note! To see the effect of this more clearly try adding the following line of code in between the `none;` and `}` attributes:

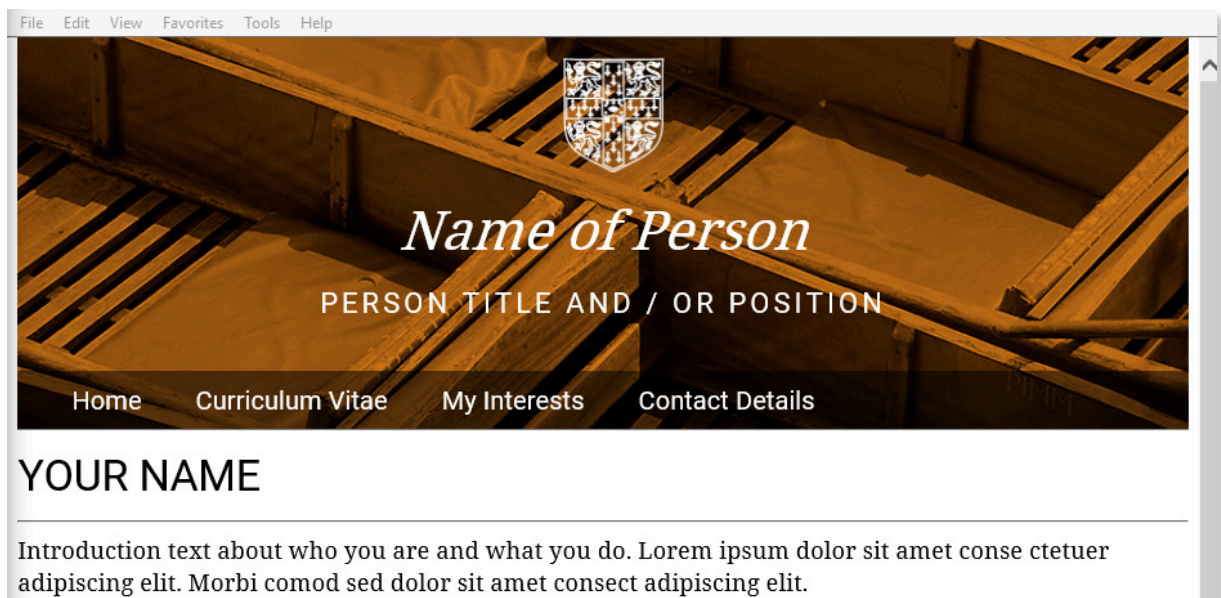
```
border: 1px solid white;
```

Remove the border afterwards.

```
nav ul li a:hover {  
    background-color: #CB4F00;  
    color: #fff;  
    transition: 0.4s;}
```

The `nav ul li a:hover` attribute targets the navigation button when the mouse hovers over it. The `background-color: #CB4F00;` will change the background of the button to an orange but the colour of the text is kept white by using the `color: #fff;` attribute. Lastly the `transition: 0.4s;` attribute allows us to state the length of time we want the transition to take. 0.4 of a second in our case gives a slight fade in effect.

Save your `screen.css` document and refresh your browser to see the finished Navigation Bar. It should resemble the screen-shot below. If it does move onto the next section, if it doesn't please ask for help.



Style and position the elements of the Person Biography section

Our finished *index.html* **Home** page will have a Person Biography section as shown below. This will include an image of yourself, your name and a brief introduction about who you are and what you do. This section will fill the same space on our other three pages although the content on those will change.

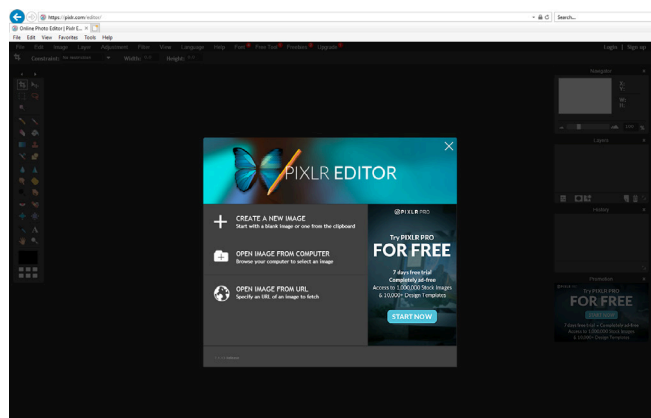


Resize the person image using pixlr.com online photo editor

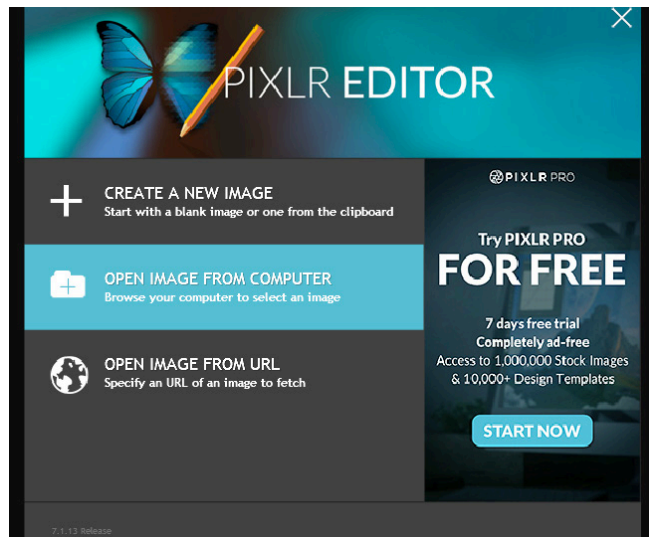
The image size of the person image on our Person Biography section needs to be 256px x 256px in size. We will use a free online Photo Editor called Pixlr to resize our person image from a width and height of 128px to 256px. There are other tools capable of doing this, Adobe Photoshop being the industry standard plus other free online tools which are available. In the absence of Adobe Photoshop, Pixlr works very well.

Instructions

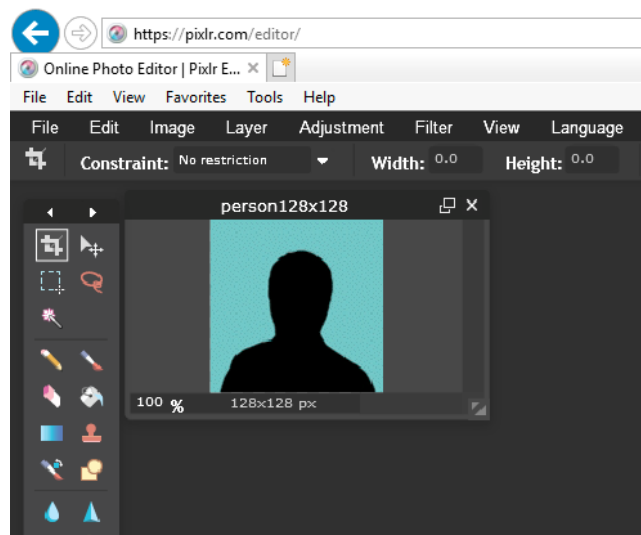
Using Edge web-browser navigate to:
<https://pixlr.com/editor/>



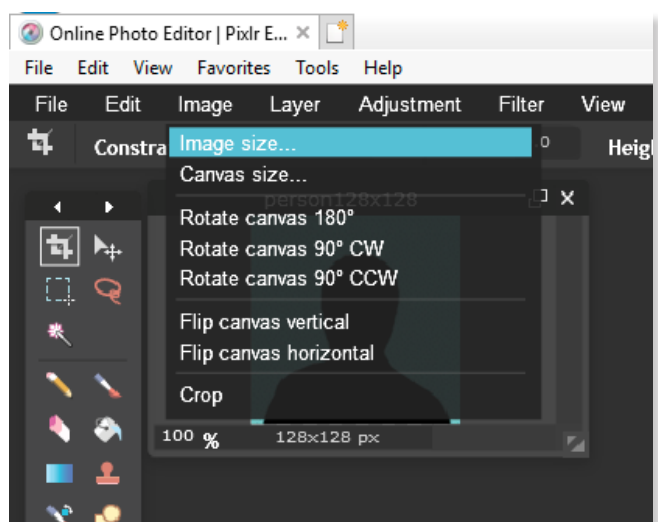
Choose 'Open Image From Computer'



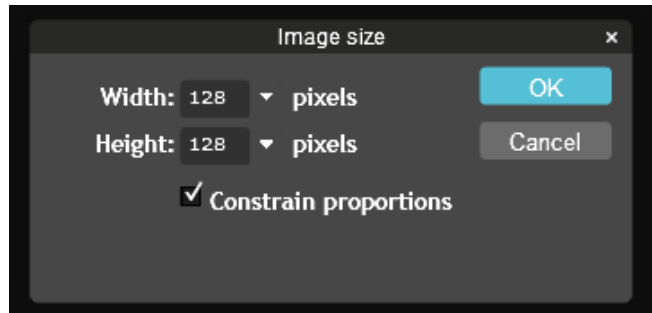
Navigate to your *images* folder and open the *person128x128* image. The image will open up in the Pixlr workspace.



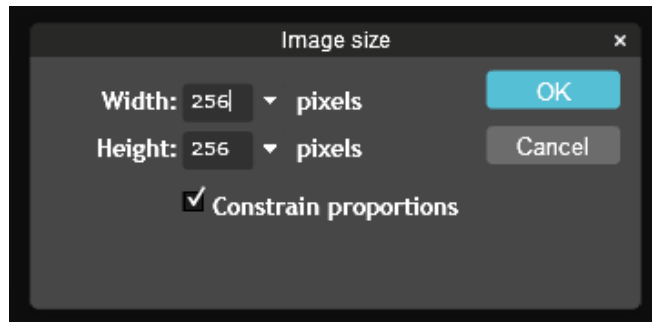
From the **Image** menu choose **Image size**.



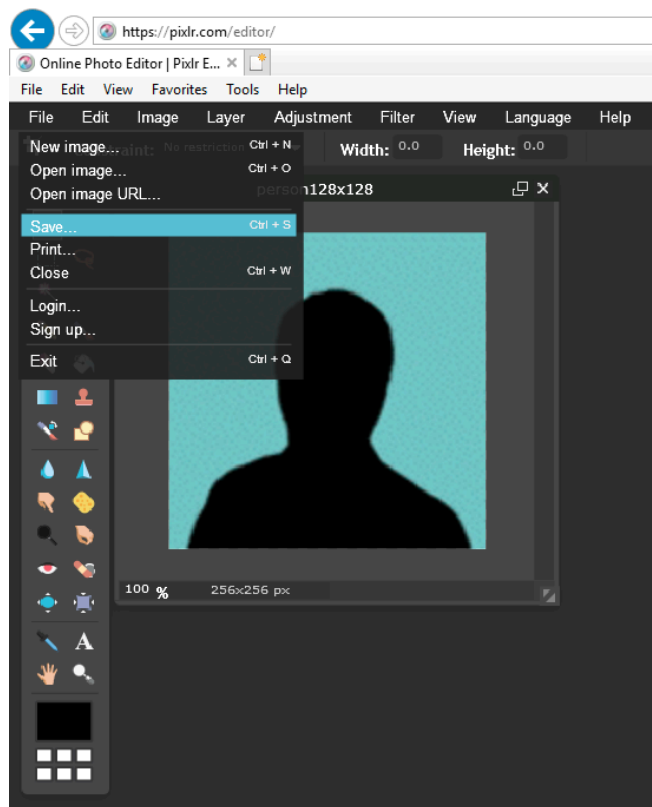
An 'Image size' dialogue box will appear showing the original size of the image in pixels, Width 128 and Height 128 in our case.

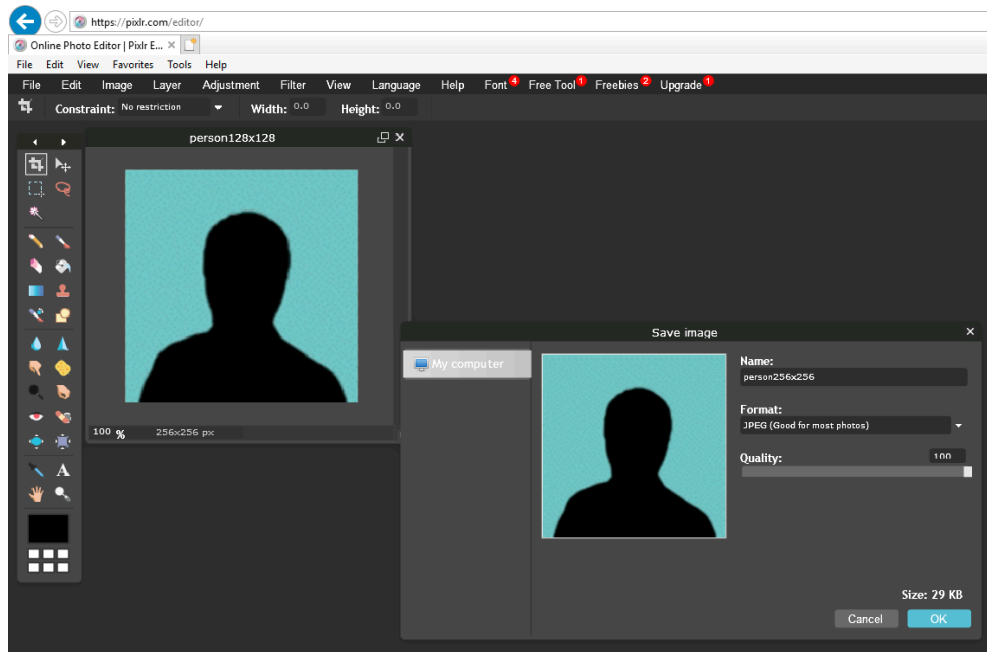


Enter 256 in both boxes making sure the 'Constrain proportions' tick box is ticked. Click **OK**.



The Person image will now appear larger on the Pixlr workspace. From the **File** menu choose **Save**.





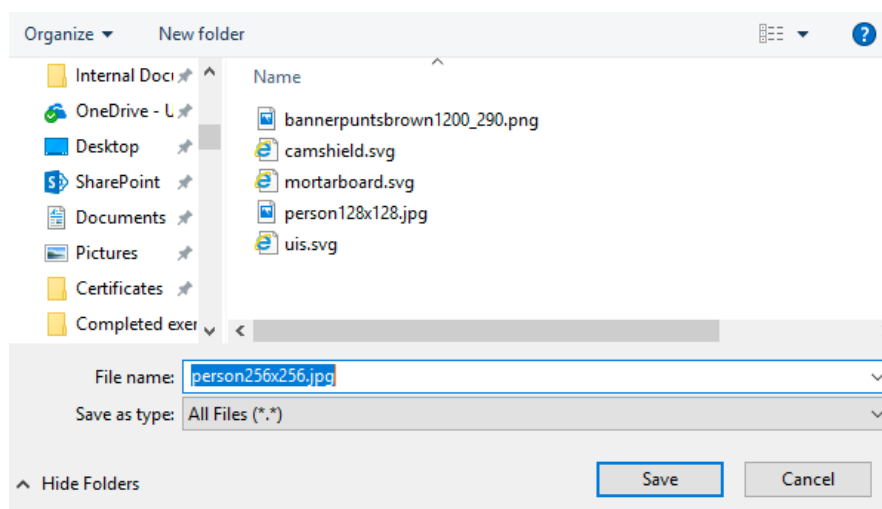
A 'Save Image' dialogue box will appear as shown above.

Enter **person256x256** in the **Name Field**.

Make sure the **Format** is set to JPEG

Quality is set to **100**.

Press **OK**



A 'Save Dialogue Box' will appear on your computer.

Save the file as **person256x256.jpg** into your **images** folder.

With our image resized we are now ready to style the elements in our *personbio* section. These will include the width and height of the section, the background colour, the person image and the h2, hr and p attributes.

Style the width and height of the personbio section

Instructions

Back on our *screen.css* page open in Notepad++ create a few lines of space after the last lines of code. Enter the code below:

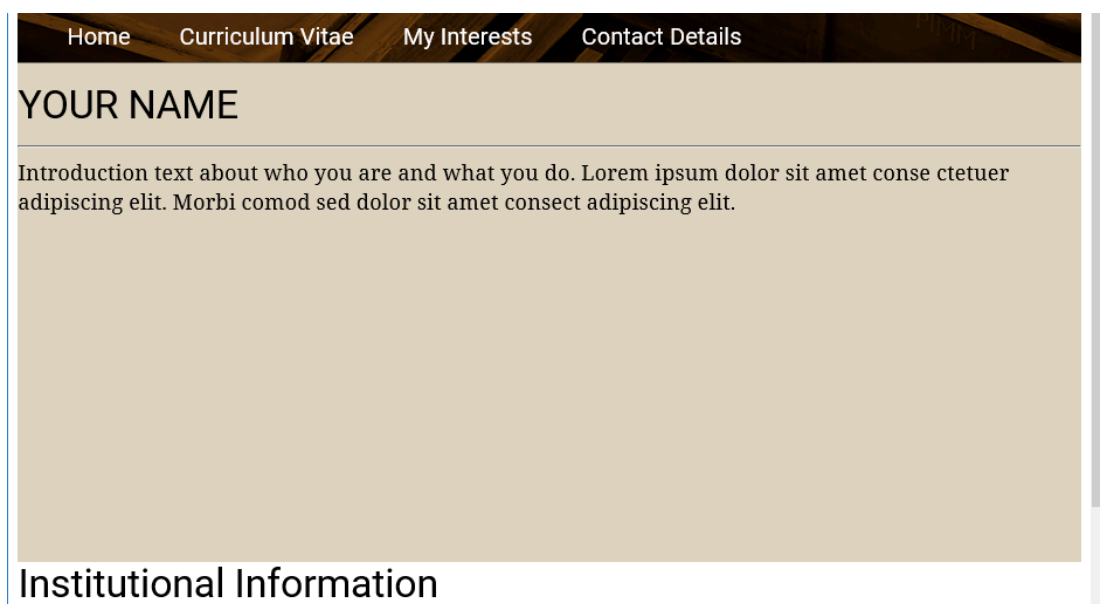
The Code

```
section.personbio {  
    width: 100%;  
    height: 406px;  
    background-color: #dcd3be;}
```

Explanation

First of all we target the section itself using the attribute `section.personbio`. The width is set to 100% so the section spans the whole width of the page. The intention is to have 75px of space above and below the person image. The height of the person image is 256px so working from top to bottom we have $75\text{px} + 256\text{px} + 75\text{px} = 406\text{px}$. Therefore the height of the *personbio* section has been set to 406px. Some working out on paper is sometimes required when designing a website together with some drawings of the finished design. This can cut down on time when coding the web-page as all the measurements have been worked out before hand. Lastly the `background-color:` attribute and hexadecimal value of `#dcd3be` sets the cream beige background colour.

Save you *screen.css* document and refresh the web-page. Your *personbio* section should now look like the screen-shot below.



Insert the person image and target the h2, hr and p attributes

Instructions

We are now ready to insert the person image and target the h2 heading, the horizontal rule line and the paragraph text attributes. Enter the code below.

The Code

```
div.personimage {
  position: relative; top: 75px; left: 12%;
  width: 256px; height: 256px;
  background: url(../images/person256x256.jpg) no-repeat 0 0;
  background-size: auto;}

div.persontext {
  position: absolute;
  top: 365px; left: 55%; width: 33%; margin: auto;
  text-align: left;}

div.persontext h2 {
  letter-spacing: 2px;
  color: #000;
  line-height: .75em;}

div.persontext hr {
  width: 30%;
  text-align: left;
  margin-left: 0;
  margin-bottom: 10px;
  color: #000;
  height: 1px;
  background-color: #000;
  border: 0 none; }

div.persontext p {color: #000; }
```

Explanation

The `div.personimage` CSS attribute targets the `class="personimage"` HTML tag. We use a position of relative as we are placing the image 75px from the top and 12% from the left relative to it's normal position, which would be the top left corner. We need to specify the width and height of the image, ours being 256px for width and height. The `background: url` attribute specifies the location of the image and ends with a `no-repeat 0 0;` attribute which prevents the image from repeating across or down the web-page. Lastly the `background-size: auto;`

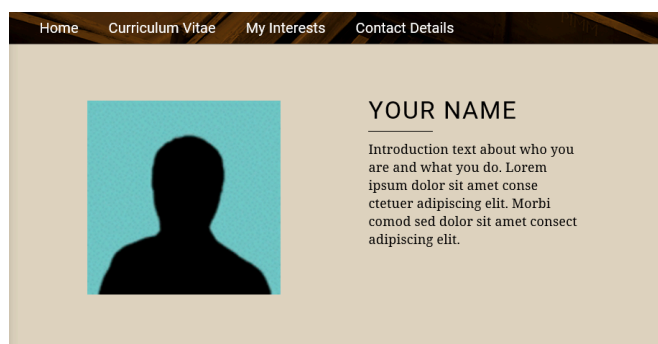
setting is the default value and makes sure the background image is displayed in its original size.

The `div.persontext` CSS attribute targets the `<div class="persontext">` HTML tag which is essentially a container or box which holds the h2 text, the horizontal rule and the paragraph text. If you add a solid border around this container you will see it in action more clearly. Remove the border once you have finished looking. We are using a position setting of absolute here. To put it another way we want to position this container absolutely where we want it on the page. Basically we want the top of the container level with the top of the person image. We know that there is 75px of space between the top of the person image and the top of the personbio section. We also know the height of the header section which is 290px. $75\text{px} + 290\text{px} = 365\text{px}$. Therefore the line beginning `top: 365px;` will position the container 365px from the top of the web-page, 55% from the left hand side and will give the container a width of 33% of the total page width. The `margin: auto;` setting will allow the container to resize as more or less text is added into the HTML document.

The `div.persontext h2` attribute styles the h2 text. Although most of the attributes are taken from the original text styling settings at the top of the screen. css document we are adding a couple of additional settings. Firstly the `letter-spacing: 2px;` setting increases the gap between the letters to style the YOUR NAME text further. Secondly the `color: #000;` setting will make the text colour black. Lastly the third setting of `line-height: .75em;` overrides the original setting forcing the h2 text to line up at the top of the container and level with the top of the person image.

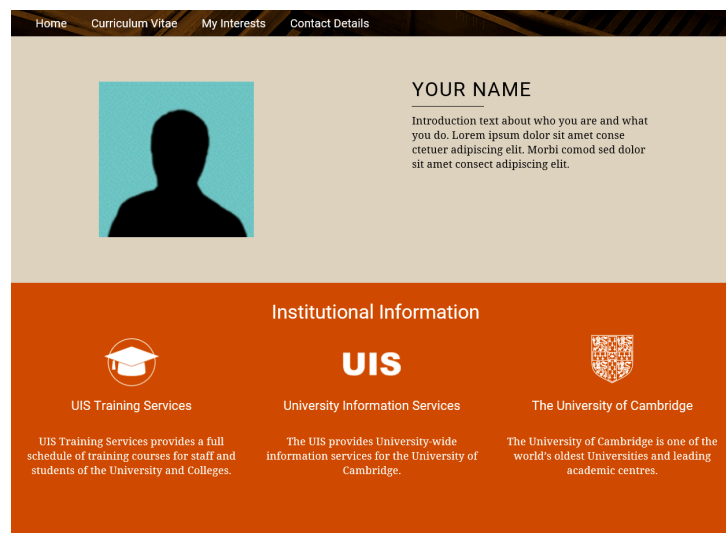
The `div.persontext hr` attribute styles the horizontal rule. The width is set to 30% of the width of the *persontext* container and aligned to the left of it. The `margin-left: 0;` setting makes it stick to the left side of the container and the `margin-bottom: 10px;` setting makes some room underneath it before the paragraph text starts. Colour is set to black using the hexadecimal code of `#000`. The `height: 1px;` setting controls the thickness of the line. Try adjusting this to 2px to see what it does. Lastly we have another colour setting of `background-color: #000;` and `border: 0 none;` again making the line black and setting the border attributes. These last two settings take into account different makes of web-browser which interpret the code differently and may display the hr line differently to how we want it.

Lastly the `div.persontext p` attribute targets the `<p>` HTML tag. The `{color: #000; }` setting targets the paragraph text and makes it black. That's all we need for this as the other settings for the paragraph font have already been set at the beginning of this practical. Your personbio section should now resemble the screen-shot shown on the right. If not please ask for help.



Add CSS styling to the Institutional Info section

As shown below our web-page will have an 'Institutional Information' section. This will have an orange background, an h2 heading and three boxes or containers created by using the HTML5 Semantic tag of `<aside>`. These three containers will each hold a .svg graphic, h4 text incorporating a hyper-link and some paragraph text with a short summary about each institution.



Style the Width, Height and Colour of the Institutional Info section

Instructions

We will start this section by targeting the `<section class="institutionalinfo">` HTML tag and setting the CSS styling for Width, Height and Colour. The colour for this section is the Pantone 718C Orange taken from the University of Cambridge core colour palette of recommended colours. This can be found at <https://www.cam.ac.uk/brand-resources/guidelines/typography-and-colour/colour-palette>. Back on our *screen.css* page open in Notepad++ create a few lines of space after the last lines of code. Enter the code below:

The Code

```
section.institutionalinfo {
    width: 100%; height: 425px; background-color: #CB4F00; }
```

Challenge

You should by now be able to work out what this line of code does. If you are unsure please ask.

Style the elements within the Institutional Info section

We are now ready to target the individual elements of the Institutional Information section.

Instructions

I have given you each bit of code separately with an explanation underneath. If you Save and refresh after each stage you will be able to see how this section is built up. You can enter the code below as shown or on one single line.

The Code

```
section.institutionalinfo h2 {  
    position: relative;  
    width: 100%;  
    text-align: center;  
    top: 30px;  
    color: #fff; }
```

`section.institutionalinfo h2 {` targets the `<h2>Institutional Information</h2>` HTML tag which will read 'Institutional Information' in white letters. Using CSS we first target the section followed by a full stop then the class name of the section followed by a space and the h2 attribute.

We use the `position: relative;` setting as we are positioning the heading relative to it's normal position which would be the top left corner of the section. The width is set to 100% so it stretches across the whole page which then makes it easier to center the text using the `text-align: center;` setting. The position will be 30px from the top of the section and the font colour is set to white using the hexadecimal value of `#fff;`.

To help gain an idea of how these sections and divs sit on the page try adding this line of code `border: 1px solid white` to the end of each piece of code I have given you. It slots in just before the last `}` and can be removed afterwards.

```
section.institutionalinfo aside {  
    width: 33%;  
    float: left;  
    text-align: center;}
```

The `section.institutionalinfo aside {` targets all three `<aside>` HTML tags on one go. `width: 33%;` sets the width of each aside container to 33% of the total page width. The `float: left;` setting essentially makes each one float to the left of the other, thus lining them up in a straight line horizontally. The `text-align: center;` setting simply centers the text within each container.

```
section.institutionalinfo aside .content {  
    margin: 35px 15px 15px 15px;  
    background: no-repeat center top;  
    padding-top: 95px; }
```

The first line of code which reads `section.institutionalinfo aside .content` targets the appropriate section, that of the `<section class="institutionalinfo">` HTML tag, followed by the three `<aside>` HTML tags and lastly the HTML Div's with a class of `content` as per the first one shown here `<div class="content trainingservices">` for the *trainingservices* aside container. You can see here that by using a `class` it allows us to specifically target a number of items in one go.

The `margin: 35px 15px 15px 15px;` line sets the margins for the space the logo will fill.

The image will be inserted as a background image hence the line `background: no-repeat center top;` which sets the logo to display in the center and top position of the container and stops the image from repeating so we only see it once. Lastly `padding-top: 95px; }` pushes the `h4` heading and paragraph text down to make room for the logo's.

```
section.institutionalinfo aside h4 a {  
    color: #fff;  
    text-decoration: none; }
```

The first line which reads `section.institutionalinfo aside h4 a {` targets the three `<h4>` `<a>` anchor hyper-links within each `<aside>` element of our HTML code. The `color: #fff;` line sets the font colour to white and any text decoration is turned off, for the moment, by using `text-decoration: none; }`. The rest of the settings for our `<h4>` text are pulled from our original font specifications at the top of the page.

```
section.institutionalinfo aside h4 a:hover {  
    text-decoration: underline;}
```

The above line of code targets the same `<h4>` `<a>` anchor hyper-links as above with the added setting of `a:hover`. Setting the `text-decoration:` to `underline;` will produce a line under the `h4` heading when the link is moused over thus making it a bit more obvious there is a hyper-link to click on.

```
section.institutionalinfo aside p {color: #fff; }
```

The line of code above sets the paragraph font colour to white. Again that's all we need here as the paragraph font properties have already been set.

```

section.institutionalinfo aside .content.trainingservices {
    background-image: url(../images/mortarboard.svg);
    background-size: 90px 90px;}

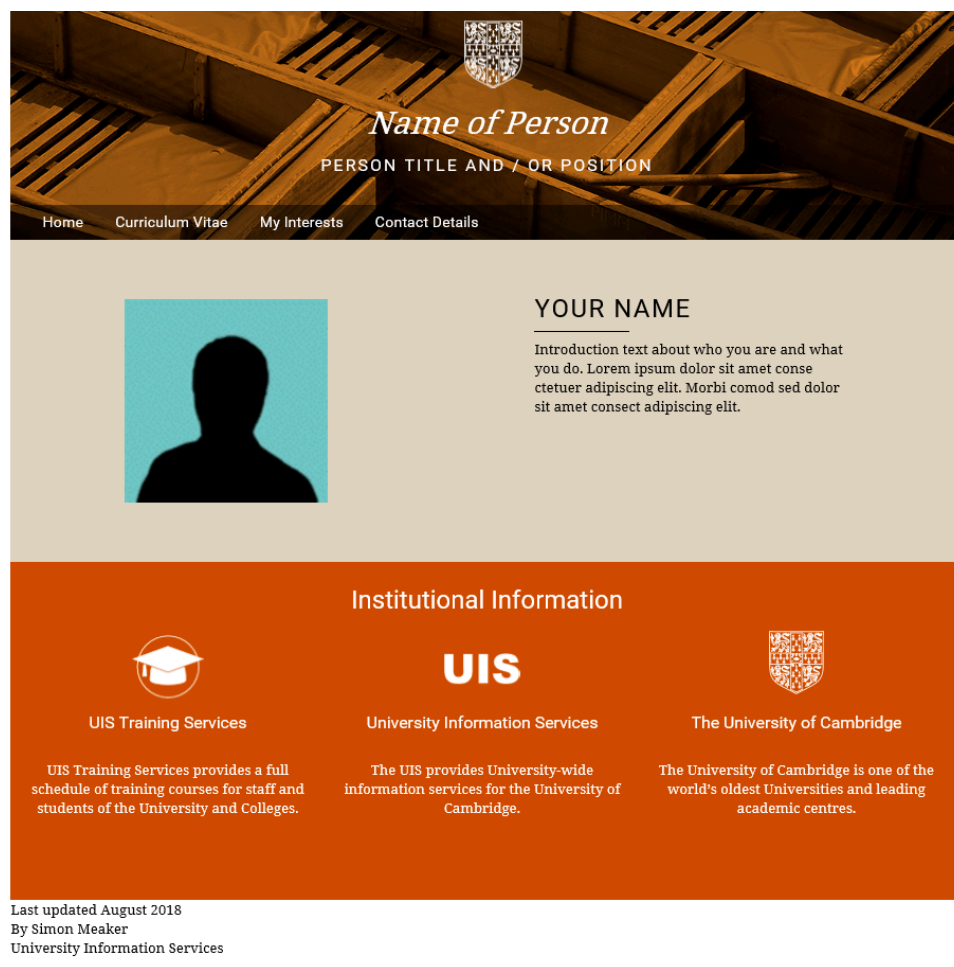
section.institutionalinfo aside .content.uis {
    background-image: url(../images/uis.svg);
    background-size: 95px 95px;}

section.institutionalinfo aside .content.uniofcam {
    background-image: url(../images/camshield.svg);
    background-size: 80px 80px; }

```

The last piece of code above targets each HTML `<aside>` `<div>` with a `class="content"` attribute and inserts each .svg logo as a background image. The `background-size:` setting adjusts the size of each logo.

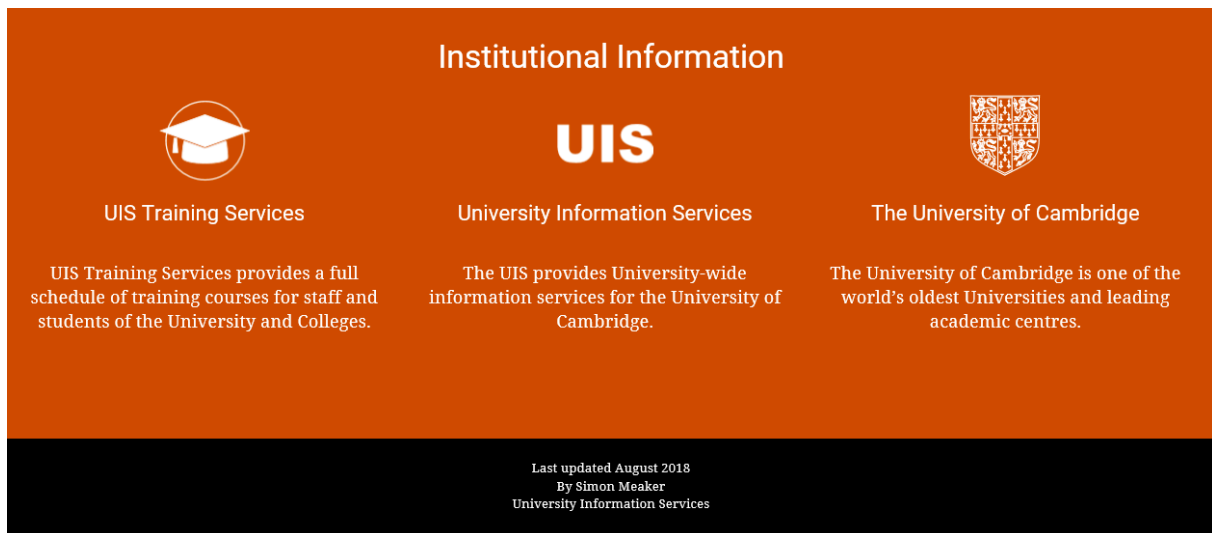
Your web-page should now resemble the screen-shot below.



If it doesn't please ask for help.
Check the Institutional Information hyper-links work and move onto the next stage where we will style the `<footer>` section.

Add CSS styling to the Footer

Our web-page is finished off at the bottom with a simple `<footer>` as shown below. This will include the month and year your web-page was last updated, your name and your department or college.



Instructions

Enter the code below. At this stage you may want to go back to your four html pages and edit the month and year to the current date, enter your own name and department or college.

The Code

```
footer {  
    width: 100%;  
    height: 100px;  
    background-color: #000;  
    text-align: center;}  
  
footer p {  
    font-size: 0.8em;  
    color: #fff;  
    padding-top: 20px;}
```

Explanation

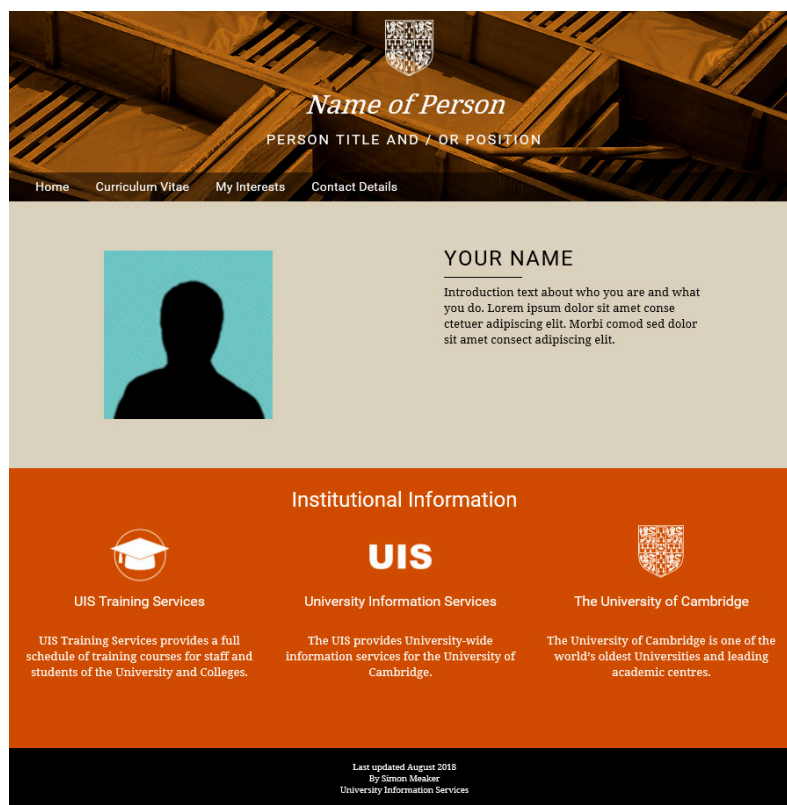
The `footer {` line targets the `<footer>` HTML tag. The width, height and background colour properties are set and the text is centered within the container.

The `footer p {` line specifically targets the `<p>` HTML tag for the paragraph text.

The `font-size: 0.8em;` line overrides the original setting for the paragraph font and reduces the size of this just for the footer section.

The colour of the font is set to white using the following line `color: #fff;` and some room is created between the top of the first line and the top of the footer section by using the following line `padding-top: 20px;}`.

Your final web-page should resemble the screen-shot below. Please ask for assistance if it doesn't.

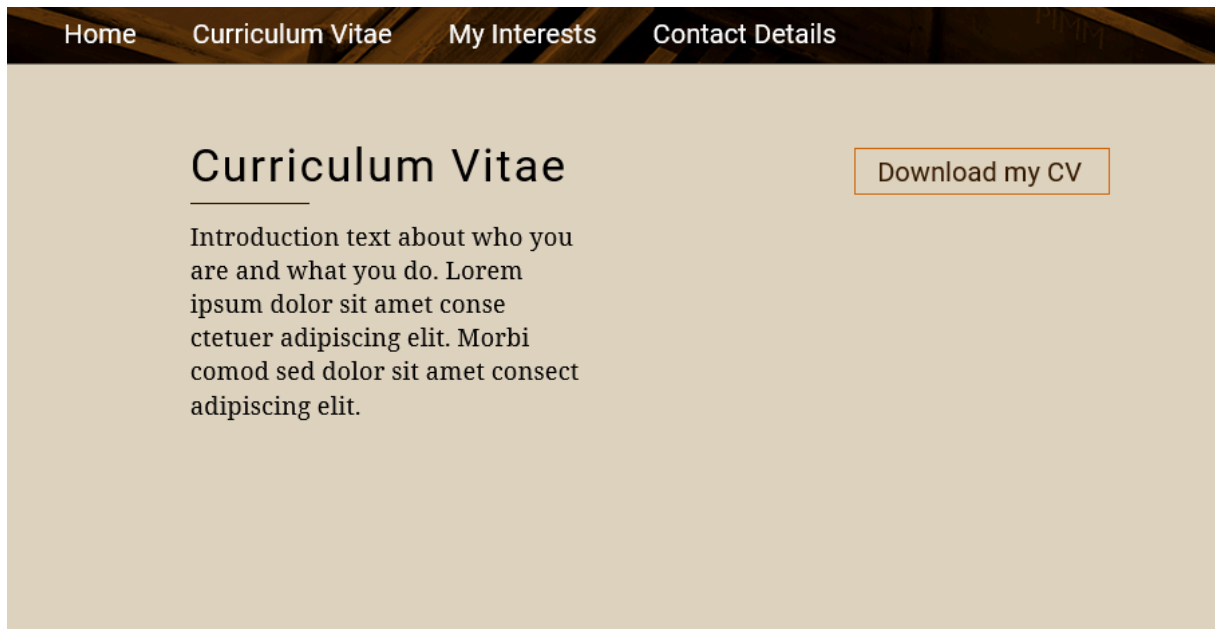


Try clicking the Navigation links to the *Curriculum Vitae*, *My Interests* and *Contact Details* web-pages. These pages should have the same styling elements as the Home page, other than the sections which replace the *personbio* section on the Home page. They should still have the default no-style applied.

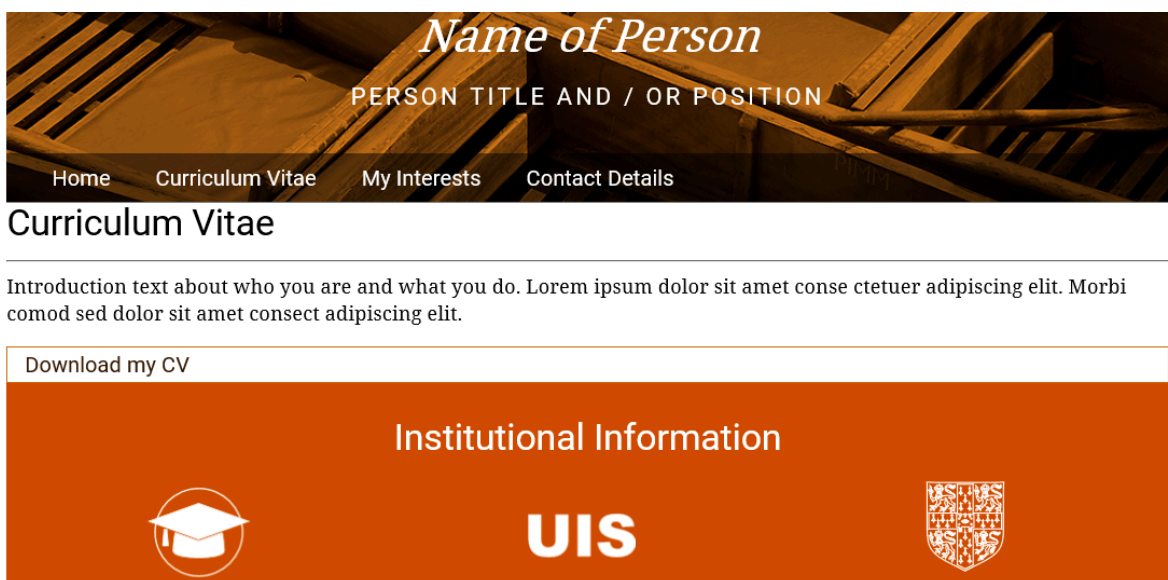
Our next challenge is too add styling to those and we are done!

Add CSS Styling to the CV section

As shown below our Curriculum Vitae cv section will contain some introduction text about you and a button to enable the viewer to download a pdf document.



At the moment though the section looks like the screen-shot below as it hasn't been styled yet. Lets rectify that now.



Instructions

Make a few lines of space on your `screen.css` document after the previous lines of code. Enter the code shown on the next page, an explanation follows each part.

The Code

```
section.cv {  
    width: 100%; height: 406px; background-color: #dcd3be; }
```

`section.cv {` targets the `<section class="cv">` HTML tag. The width is set to 100% which means it will span across the whole page. The height is set to 406px, the same as the *personbio* section on the Home page for continuity. Finally the background colour is set to the same beige colour as the *personbio* section.

Save and refresh as you go to see how this section is built up.

```
div.cvtext {  
    position: absolute;  
    left: 130px;  
    width: 33%;  
    margin: 60px 0 0 0;  
    text-align: left; }
```

`div.cvtext {` targets the `<div class="cvtext">` HTML tag which is essentially the container which will hold the text.

It is positioned using the `position: absolute;` setting which enables us to position it exactly where we want within this section. We are placing it 130px from the left hand side with a width of 33% of the total page width. Notice there is no `top` setting here. Instead we are using the `margin: 60px 0 0 0;` setting which gives 60px of room in-between the top of the text box and the top of the section. If we were to replace the margin setting with `top: 60px` this would position the text box 60px from the top of the web-page resulting in our text box being placed over the top of the punt image in the `<header>` section. Not what we want.

Finally we use the `text-align: left; }` setting to make sure the text is left aligned in the box.

If you Save your *screen.css* file and refresh your web-browser you may notice that the text has moved from it's default position of the top left corner of this section to where we have chosen to position it. As soon as that happened this had a counter effect on the 'Download my CV' button, which has now jumped from beneath the text which was it's default no style position, to take up a new position in the top left corner. This actually makes it easier to position the button later on using relative positioning, but more on that later. First we need to style the text.

```
div.cvtext h2 {  
    letter-spacing: 2px; color: #000; margin: 0 0 15px 0; }
```

`div.cvtext h2 {` targets the `<h2>Curriculum vitae</h2>` HTML tag within the div class of “cvtext.”

We use the `letter-spacing: 2px;` setting to spread out the characters a bit to add some extra styling. `color: #000;` sets the font colour to black and `margin: 0 0 15px 0;` makes 15px of room in-between the h2 heading and the horizontal rule below.

```
div.cvtext hr {
  width: 30%;
  text-align: left;
  margin-left: 0;
  margin-bottom: 10px;
  color: #000;
  height: 1px;
  background-color: #000;
  border: 0 none; }
```

The code above is the styling for the horizontal rule line and has been copied from the *personbio* section for continuity. An explanation is on page 19.

```
div.cvtext p {color: #000; }
```

The code above targets the paragraph font within the div class of “cvtext” and sets the colour to black.

```
.btn {
  position: relative;
  top: 60px;
  left: 70%;
  width: 150px; text-align: center; }
```

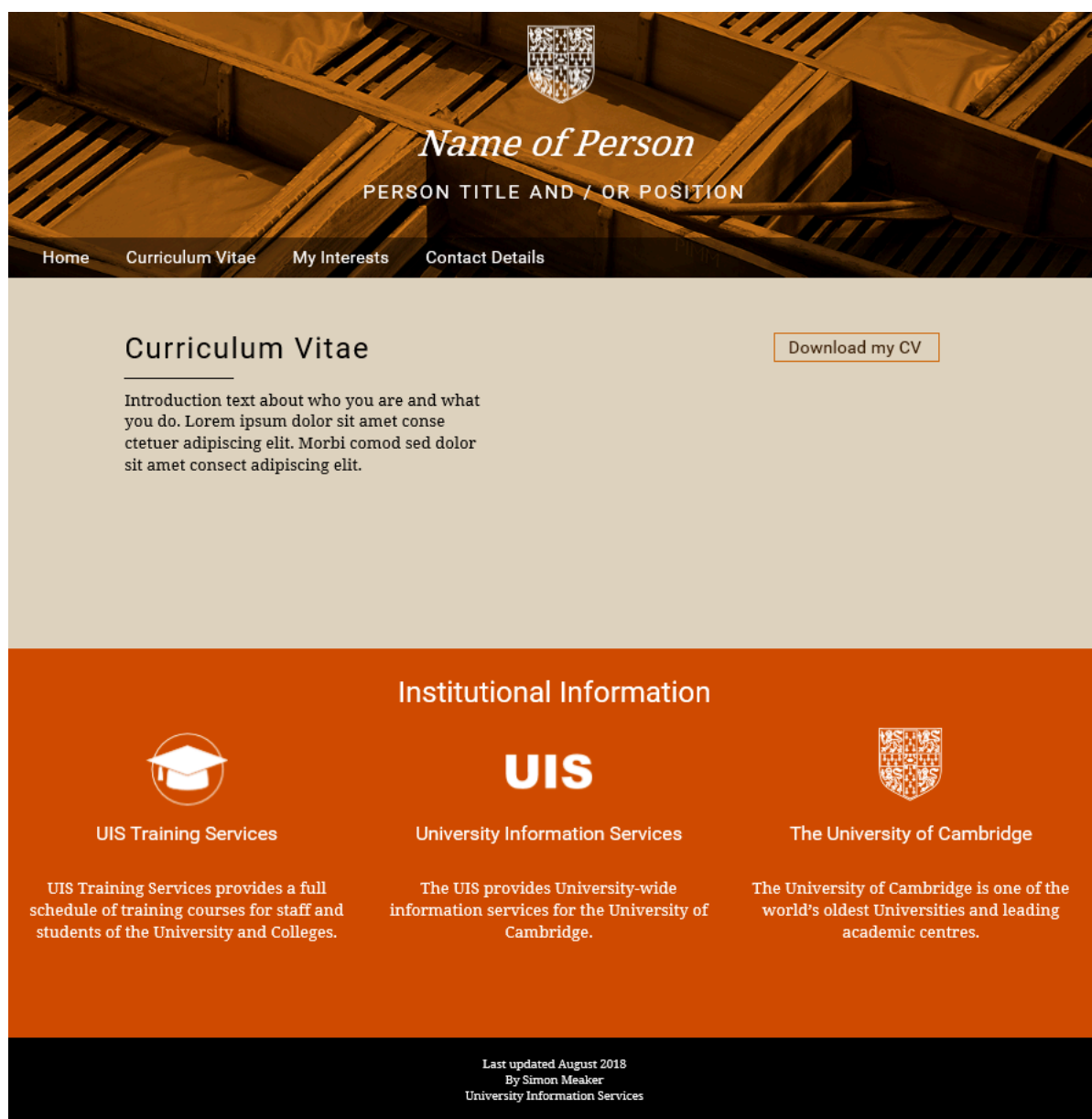
Lastly we come to our *download button* positioning and this is where some explanation is required concerning the `position:` settings. The problem comes from having to position two elements within the same section. Referring to our HTML code in our *cv.html* document the main section is called `<section class="cv">` which we target with CSS using `section.cv {`. Within that we have our text box element called `<div class="cvtext">` which we target with CSS using `div.cvtext`. Finally we have our button element which is called `` and that is targeted with simply `.btn`. So we need to position two elements within the same section. To make this work one has to have a position of absolute and the other has a position of relative. When we positioned our text box earlier using `div.cvtext { position: absolute;` the *Download my CV button* took up a new position in the top left corner. As this is now its new default position we can position it relative to its default position, hence the code above. `top: 60px;` moved it 60px down from the top of the section.

left: 70%; moved it 70% across the page. Again we use a percentage here so it moves in relation to the width of the web-browser.

width: 150px; sets the width of the button to 150px.

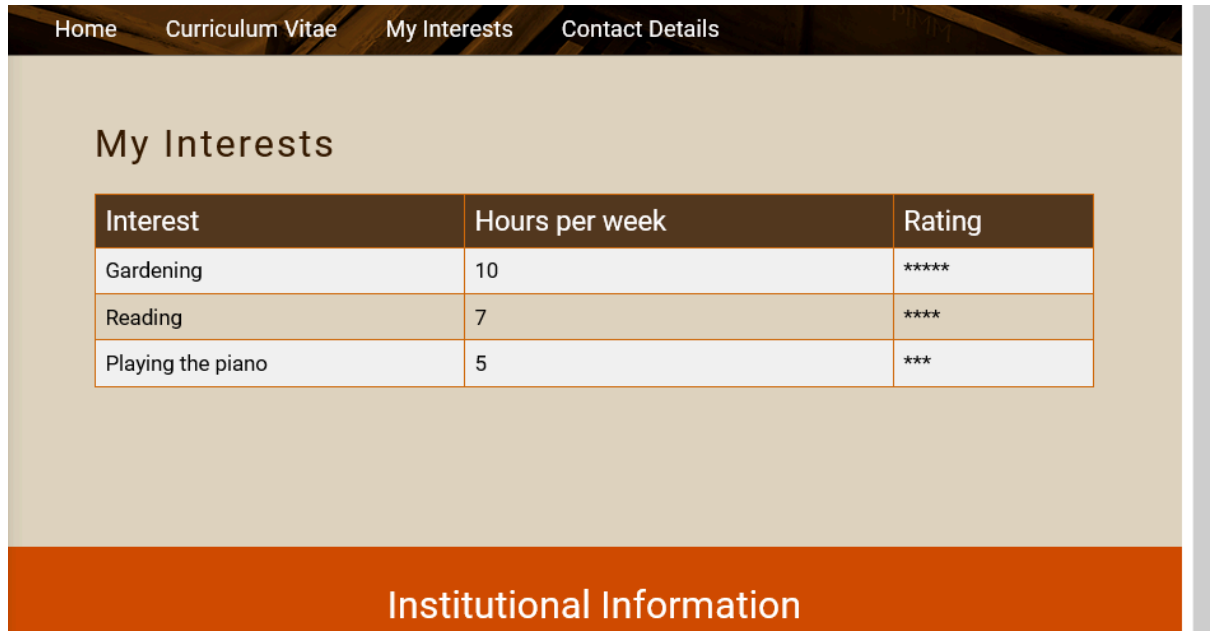
Using the same position setting for both the text box and the button, whether that's a position of relative for both or absolute for both is problematic and will cause problems. Here we used a position of absolute first for the text box and a position of relative second for the button. In our *personbio* section for the Home page we did it the other way round and used a position of relative first for the person image and a position of absolute second for the text box. www.w3schools.com has a page about CSS Positioning and although there are some general rules to follow there is room for maneuver and at the end of the day you do what works best.

Your finished Curriculum Vitae page should now resemble the screen-shot below. Check the *Download my CV button* works, you should be presented with a PDF document.

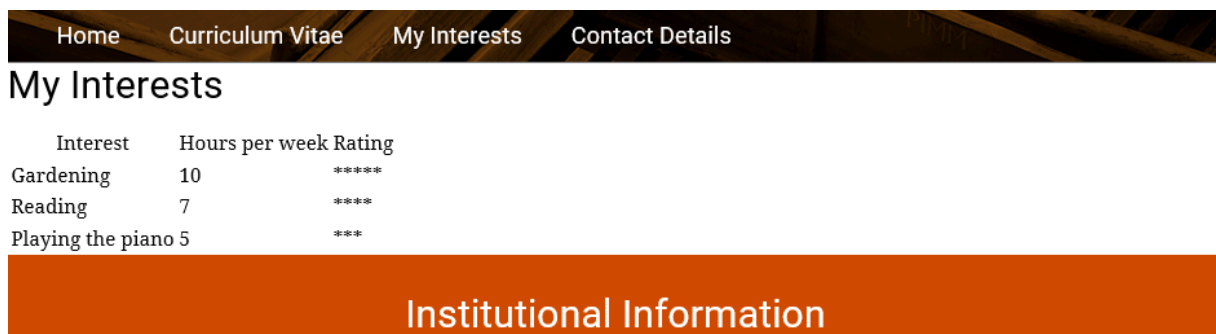


Add CSS Styling to the My Interests section and table

As shown below our My Interests section contains a table which we will now style, adding colour to the table heading, cells and border. We will also add a mouse roll over effect.



At the moment though your web-page should resemble the screen-shot below. If you haven't done so already you may want to edit the table in your *interests.html* document and add in your own interests.



Instructions

Make a few lines of space on your *screen.css* document after the previous lines of code. Enter the code shown on the next page, an explanation follows each part.

Style the interests section

```
section.interests {  
    width: 100%; height: 406px; background-color: #dcd3be; }
```

As before with our *personbio* and *cv* section this sets the width, height and colour of this section.

Position the table

```
div.intereststable {  
    position: relative;  
    top: 60px; left: 7.5%; width: 85%;  
    text-align: left; }
```

The `div.intereststable {` line targets the `<div class="intereststable">` HTML tag.

At the moment the default position of the table is the top left hand corner of the section as shown on the lower screen-shot on the previous page. Therefore we are positioning it relative to it's default position using the `position: relative;` setting.

`top: 60px;` places it 60px down from the top of the section.

`width: 85%;` will make the table take up 85% of the width of the web-page and will automatically adjust the size of the table should the web-browser be resized.

Obviously we still want the table to be centered horizontally within the section which is where the `left: 7.5%;` line comes into play which will position the table 7.5% in from the left hand side of the section. $100\% - 85\% = 15\% / 2 = 7.5\%$.

Finally we use the `text-align: left;` setting to align the text to the left of the table.

Style the table title

```
div.intereststable h2 {  
    letter-spacing: 2px;  
    color: #351a02; }
```

The `div.intereststable h2` line targets the `<h2>My Interests</h2>` HTML tag inside the division with a class of 'intereststable' `<div class="intereststable">`.

`letter-spacing: 2px;` creates some space in-between the heading 2 characters to add some additional styling to the table title of 'My Interests'.

`color: #351a02;` changes the 'My Interests' table title to brown.

Style the properties of the table

```
table {  
  font-family: Roboto, sans-serif;  
  font-size: 1.1em;  
  font-weight: normal;  
  border-collapse: collapse;  
  width: 100%;  
  margin: 30px 0 0 0 ; }
```

`table {` targets the `<table>` HTML tag and sets the properties for the table itself.

`font-family: Roboto, sans-serif;` sets the font for the text inside the table. If Roboto is not available then sans-serif will be used instead.

`font-size: 1.1em;` adjusts the size of the font in our table.

`font-weight: normal;` makes sure the default font weights for the table are maintained.

`border-collapse: collapse;` The border-collapse property sets whether the table borders should be collapsed into a single border. The outside line would normally be a double line border, this setting collapses it to a single line.

`width: 100%;` ensures the table extends to 100% of the width we have given over to the table. In our case we set it to 85% of the page width.

`margin: 30px 0 0 0 ; }` creates some room between the top of the table and the table title, just to spread things out a bit.

Style the table data and table header cells

```
table td, table th {  
  border: 1px solid #ca6a15;  
  padding: 8px; }
```

`table td, table th {` targets both the table data and table header cells in one go.

`border: 1px solid #ca6a15;` applies an orange border around the cells.

`padding: 8px; }` creates some space around the text in each cell.

Add a colour to alternating table rows

One way to make large tables readable is to give alternate rows a different colour.

```
table tr:nth-child(even){  
  background-color: #f2f2f2;}
```

`table tr:nth-child(even){` targets every other table row starting with row 2, as 2 is an even number. An off white background colour is then added to those rows using the `background-color: #f2f2f2;` property. The odd rows, that's rows 1 and 3, will not receive any colour and so will remain transparent thus taking on the beige background colour. Although if we wanted we could colour the odd rows as well using the `table tr:nth-child(odd){` setting.

Add a hover effect to colour the table rows

```
table tr:hover {  
    background-color: #cec3a8;}
```

`table tr:hover {` is used to target the table rows to give a mouse over hover effect.

`background-color: #cec3a8;}` will change any row that the mouse hovers over to a pale brown colour.

Style the table header

```
table th {  
    text-align: left;  
    background-color: #503720;  
    color: #fff;  
    font-size: 1.3em;  
    font-weight: normal; }
```

`table th {` specifically targets the table header HTML tags.

`text-align: left;` aligns the table header text to the left.

`background-color: #503720;` applies a dark brown background colour to the cells in the table header.

`color: #fff;` applies a white colour to the font.

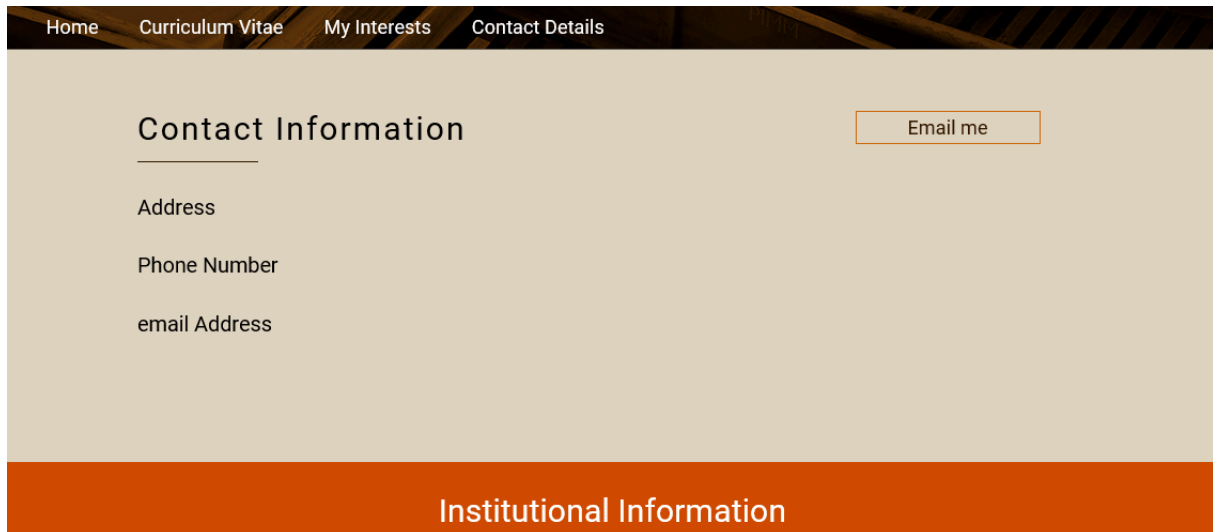
`font-size: 1.3em;` increases the size of the table header font to make it a bit larger than the table data font.

`font-weight: normal;` maintains a normal look to the table header font as opposed to Bold.

Your My Interests section and table should now resemble the screen-shot at the top of page 30. If it doesn't please ask for help, otherwise move on to the last part where we will style the Contact Details section.

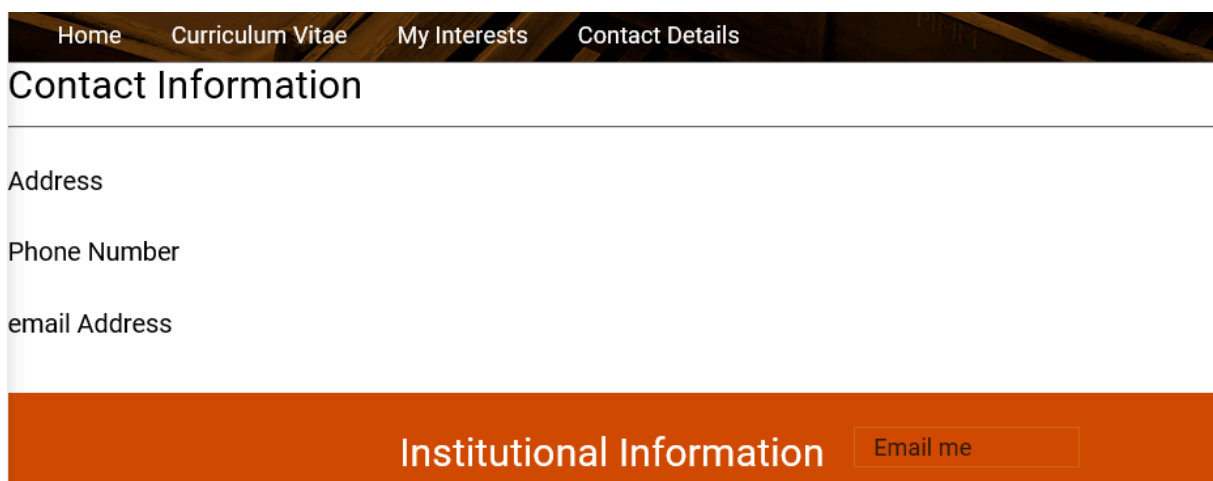
Add CSS Styling to the Contact Details section

As shown below our Contact Details section contains three lines of text where your University Address, University Phone number and University email address can be entered. You will have to modify the *contacts.html* web-page if you wish to enter your own information. This section also includes a button which if clicked will open an email to your email address.



The screenshot shows a web page with a dark navigation bar at the top containing links: Home, Curriculum Vitae, My Interests, and Contact Details. The main content area has a light beige background. On the left, under the heading 'Contact Information', there are three text input fields labeled 'Address', 'Phone Number', and 'email Address'. To the right of these fields is a rectangular button labeled 'Email me'. At the bottom of the page is a solid orange footer bar with the text 'Institutional Information' in white.

At the moment though your web-page should resemble the screen-shot below. The text is positioned in it's default position and you may notice the button is taking up position next to the words 'Institutional Information'. Lets sort all of this out.



This screenshot shows the same web page as the previous one, but with a different layout. The 'Email me' button is now positioned to the right of the 'Institutional Information' text in the orange footer bar, instead of being above it. The rest of the page structure, including the navigation bar and the 'Contact Information' section with its input fields, remains the same.

Instructions

If you haven't done so already you may want to edit the *contacts.html* web-page and add in your own University address, phone number and email address. The button

can also be edited to include your own University @cam email address. Simply edit the “btn” HTML tag replacing the part which reads `YOUR CRSid@cam.ac.uk` with your own email address.

Once you have finished editing the *contacts.html* web-page move back to your *screen.css* file open in Notepad++ and make a few lines of space after the previous lines of code. Enter the code below, an explanation follows each part.

Style the contact section

```
section.contact {  
    width: 100%; height: 406px; background-color: #dcd3be; }
```

As before with our *personbio*, *cv* and *interests* sections these properties set the width, height and colour of this *contact* section.

Position the text container box

Our text element is held within a div container box with a class of “contacttext” as per the `<div class="contacttext">` HTML tag on our *contacts.html* web-page. We are going to follow the same formula as per the *Curriculum Vitae cvtext* section, whereby we will use absolute positioning to place the text box and relative positioning for the button.

```
div.contacttext {  
    position: absolute;  
    left: 130px;  
    width: 33%;  
    margin: 60px 0 0 0;  
    text-align: left; }
```

`div.contacttext {` targets the `<div class="contacttext">` HTML tag which is essentially the container holding the text.

`position: absolute;` enables us to position it exactly where we want within this section.

`left: 130px;` sets it in place 130px from the left hand side of the web-page.

`width: 33%;` sets the width to 33% of the total page width.

`margin: 60px 0 0 0;` gives 60px of room in-between the top of the text box and the top of the section. We are not using the `top` setting here as this would position the text box 60px from the top of the web-page resulting in our text box being placed over the top of the punt image in the `<header>` section. Not what we want.

`text-align: left; }` makes sure the text is left aligned in the box.

Style the h2, hr and p elements

```
div.contacttext h2 {  
    letter-spacing: 2px;  
    color: #000;  
    margin: 0 0 15px 0; }
```

`div.contacttext h2 {` targets the `<h2>Contact Information</h2>` HTML tag within the div class of "contact."

`letter-spacing: 2px;` spreads out the characters a bit to add some extra styling.

`color: #000;` sets the font colour to black.

`margin: 0 0 15px 0;` makes some room in-between the h2 heading and the horizontal rule.

```
div.contacttext hr {  
    width: 30%;  
    text-align: left;  
    margin-left: 0;  
    margin-bottom: 10px;  
    color: #000;  
    height: 1px;  
    background-color: #000;  
    border: 0 none; }
```

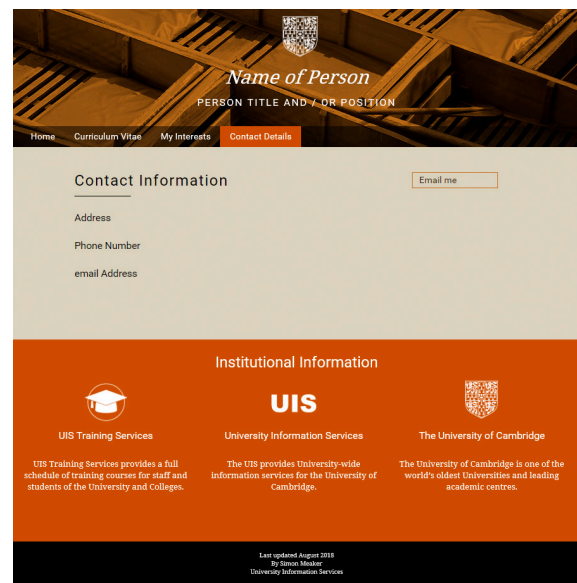
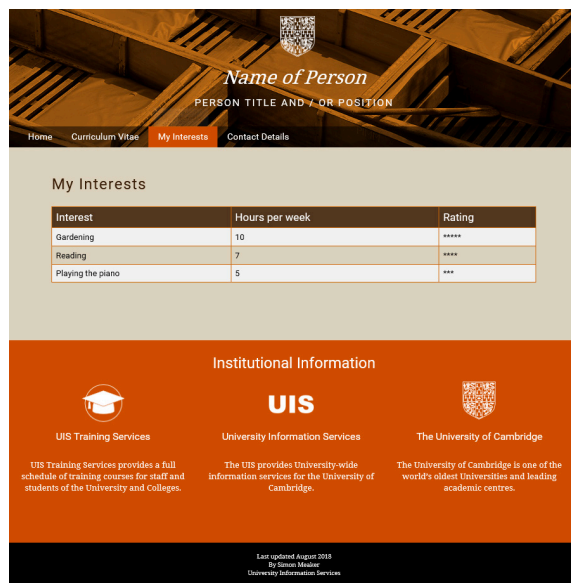
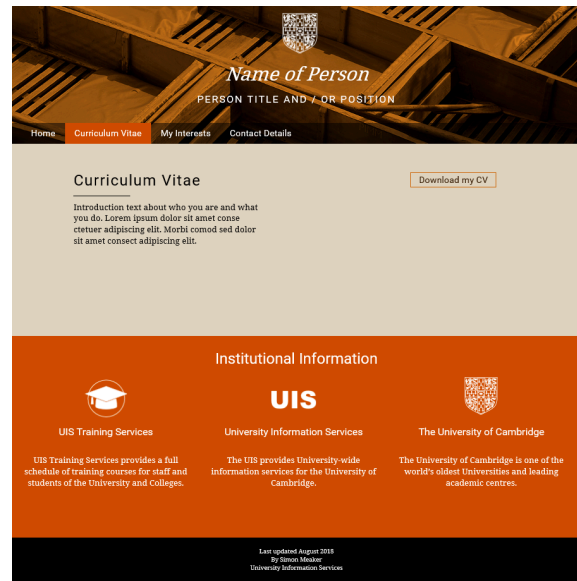
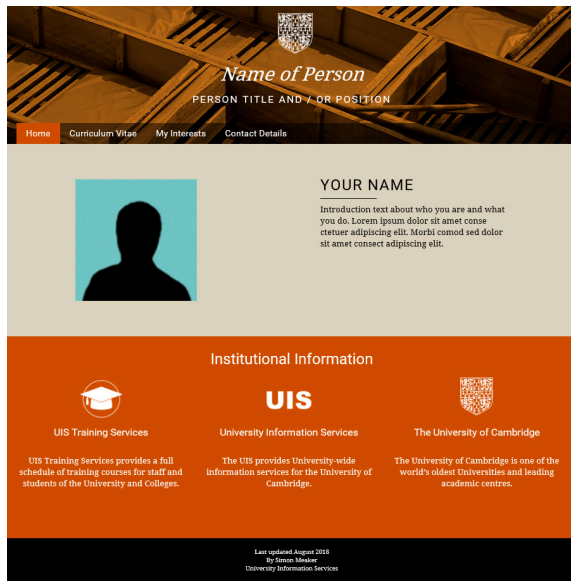
The code above is the styling for the horizontal rule line and has been copied from the *personbio* section for continuity. An explanation is on page 19.

That's all we need for this section.

Any idea why we don't need to add any code to position the button?

Please ask if you are unsure.

Your completed website



Congratulations!

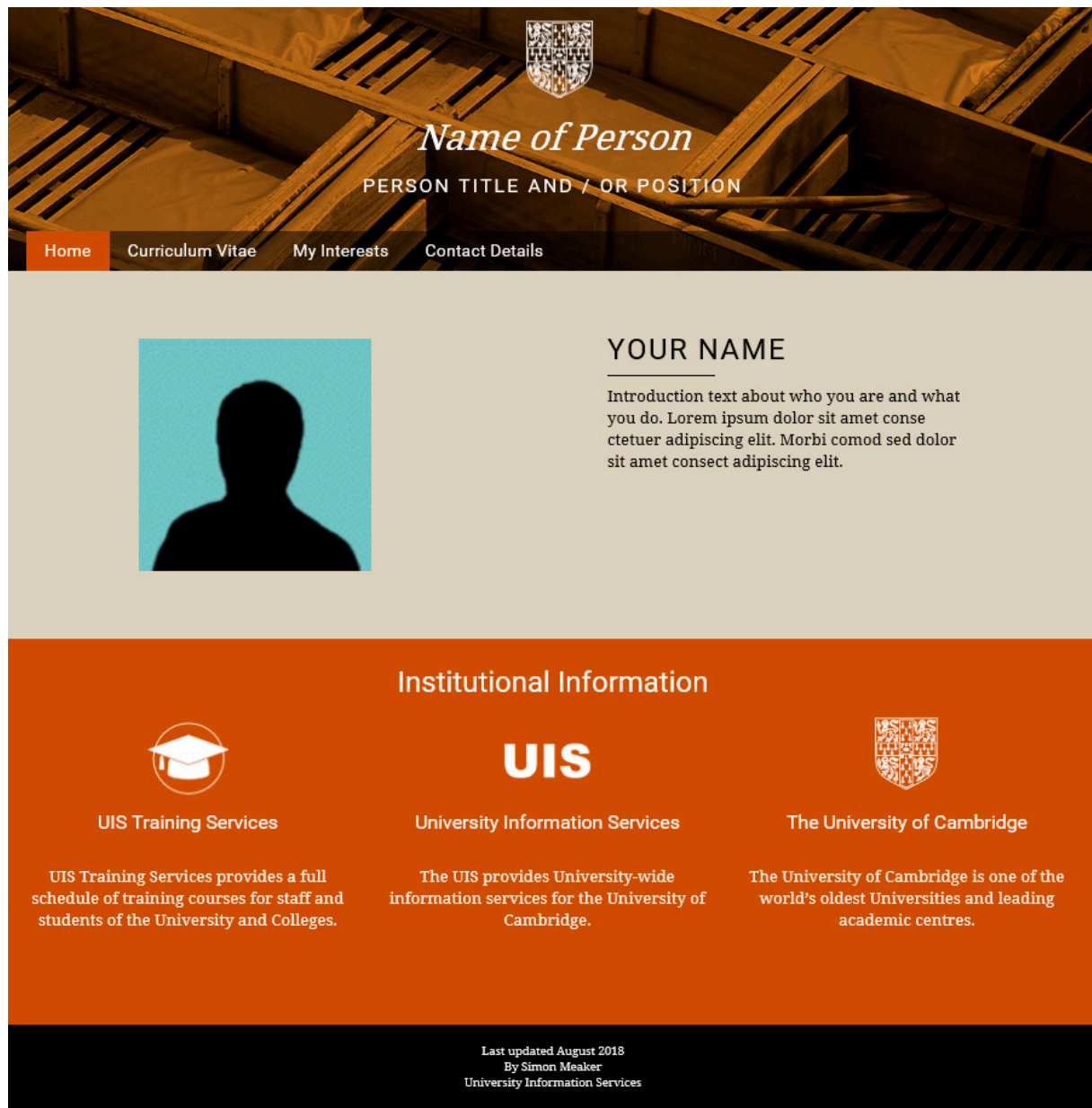
You have successfully styled your website using CSS and it should resemble the screen-shots above. Hopefully you now have a bit more than a basic understanding of how CSS works but there is still plenty more to learn. Check out the resources below if you want to take things further.

www.w3schools.com

<http://www.htmldog.com>

www.linkedin.com/learning

How to publish your website



Copy the contents of the *website* folder onto a memory stick and then copy it to the *public_html* folder of your own MCS Desktop Account. To view it in a web browser navigate to <http://people.ds.cam.ac.uk/CRSid/> replacing **CRSid** with your own.

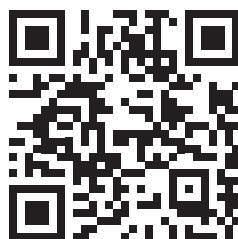
Note! You will need to remove the default files from this location first, but it's best to copy them onto the Desktop for safe keeping should you need to put them back later.

Your website is now published for anyone to view. If you don't want to use your own account you can use the training account you are currently using. Ask the trainer for the correct CRSid to add to the end of the <http://people.ds.cam.ac.uk/> URL. If you have used this method the published website will be automatically deleted shortly after the course has finished when the accounts are cleared.

Before you go....

Please complete the course feedback before you leave

<http://feedback.training.cam.ac.uk/uis>



Thank you

Notes
