



UIS Training Services

Web Authoring: CSS Responsive Web Design

Practical Notes

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<title>Home Page</title>
```

```
</head>
```


Contents

Part 1

Getting Started	01
Open the website and familiarise yourself with the content	01

Part 2

Create the first Media Query	02
Add a Media Query to target screens larger than 1200px wide	02

Part 3

Targeting a break point at 650px wide	04
Redesigning the Navigation for a vertical layout with menu button	04
Assign a class name to the <nav> section	04
Add JavaScript to our web design	04
Add a CSS media query for screens less than 650px wide	08
Use CSS to Style the Nav bar for screens less than 650px wide	09
Redesign the Person Bio section for vertical layout	13
Redesign the Institutional Information section for vertical layout	16
Redesign the CV section for a vertical layout	20
Redesign the My Interests section for a vertical layout	24
Redesign the Contact section for a vertical layout	25

Part 4

Targeting a break point at 414px wide	28
Add a Media Query to target large mobile phones	28
Change the <body> background to black	29
Adjust the font size in the “personbio” section	29
Adjust font size and margins of the “Institutional Info” section	30
Adjust h2 and p font sizes in the “Curriculum Vitae” section	30
Adjust h2 font size in the “Interests” section	31
Adjust h2 font size in the “Contact” section	31

Part 5

Target a break point at 320px wide	32
Add Media Query to target small mobile phones	32
Swap the header image for a smaller one	33
Adjust the font size within the header section	33
Adjust the position of the menu icon	34
Adjust the position of the responsive drop down menu	35
Adjust the position of the responsive menu icon	36
Adjust the font position for the Personbio section	37
Adjust the font size for the Personbio section	38
Adjust font size in the Institutional Information section	39
Adjust .svg graphic size in the Institutional Information section	39
Adjust .svg graphic padding in the Institutional Information section	40
Adjust the font size in the Footer section	41
Adjust the font size in the Curriculum Vitae section	41
Adjust the font size in the My Interests page	42
Adjust the font size in the Contact Information page	43

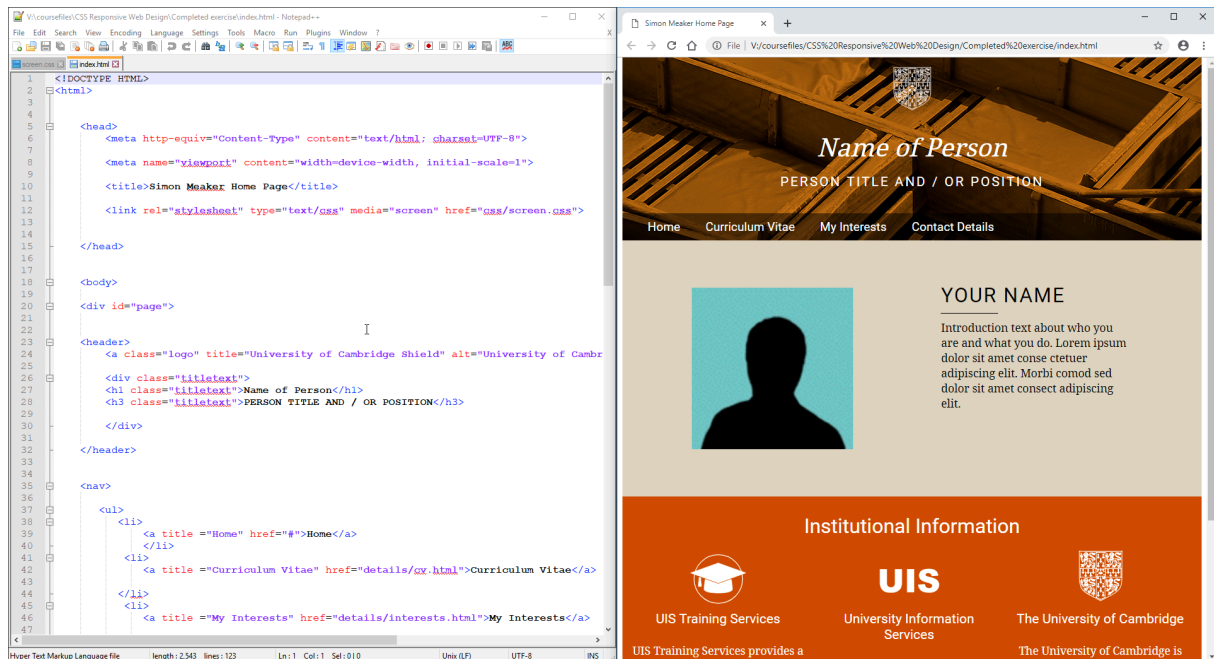
Part 6

Conclusion.....	44
Upload your website and view using your phone and PC.....	44

Course Feedback Link

Before you go.....	45
--------------------	----

Open the website and familiarise yourself with the content



Instructions

Using Notepad++ open the files *index.html* and *screen.css* which can be located inside the website folder. Then open the file *index.html* in Chrome Web Browser. Rearrange the two windows so they are side by side. Familiarise yourself with the basic website which was the finishing point of the 'CSS for Beginners course'. Your screen should resemble the screen-shot above.

Add a Media Query to target screens larger than 1200px wide

Instructions

In Notepad++ scroll down to the bottom of the screen.css document.
Add the following two comment lines as shown below.

The Code

```
/* Media Queries */  
  
/* Add a background image for screens larger than 1200px wide  
*/
```

Explanation

This is a note to ourselves and marks the place on the CSS document where the media query code starts.

Instructions

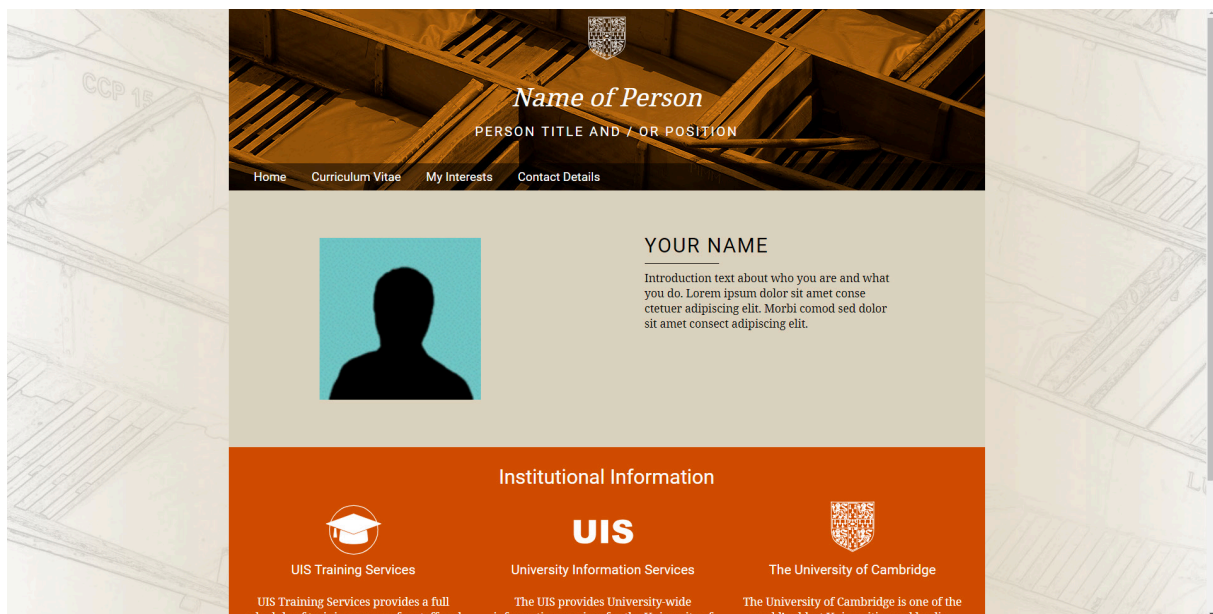
Now we are ready to write our first media query. Underneath the second comment line type in the code below.

The Code

```
@media screen and (min-width:1201px){  
  
body { background-image: url(../images/background.jpg);  
        background-repeat: no-repeat;  
        background-position: center top;  
        background-size: 100% 100%;}  
}
```

Explanation

The line of code starting `@media screen and (min-width:1201px)` targets the Web Browser window when it exceeds a width of 1200px. In essence when this Web Site is viewed using large screens, for example a 27" iMac, instead of having a white background for our Web Site we can fill it with an image for more interest. We then target the body section followed by `background-image: url(../images/background.jpg);` which sets the background image and specifies the location of the image. The `background-repeat: no-repeat;` line makes sure we only have one instance of the image showing in the background. The `background-position: center top;` line simply positions the image at the top of the screen and centers it. The `background-size: 100% 100%;` sets the image to cover 100% of the background so the image will keep resizing as the Web browser window is extended. Save the changes to the `css.screen` document and refresh your web page. Then maximize the window to fit full screen. You should see the background image appear as shown below.



Redesigning the Navigation for a vertical layout with menu button

Using Chrome press Ctrl+Shift+i to open the Chrome Developer Tool. Don't worry too much about the panel on the right. Now grab the left side of the Web Page and drag it inwards. The width and height in Pixels should show up at the top right hand corner of the screen. Keep going until the page gets to a width of 641px wide, at which point the 'Contact Details' button will drop down a line. Keep making the Web Page narrower until the 'Your Name' text moves on top of the 'Person Picture'. These are the problems we need to target with our responsive design. We are going to make a break point at 650px wide, before these problems show up, and change our design to a vertical layout ready for further changes where we will target some phone sizes. We will also introduce a 'burger' menu for use with mobile devices and this will rely on some Java script which will also be incorporated. Before we can introduce the Java script there are a few changes we need to make to our *html* pages.

Assign a class name to the <nav> section

Instructions

With the *index.html* page open in Notepad++ scroll to the `<nav>` section. We are going to give this section a class name so we can target it with CSS and Javascript to make it work with the burger menu button we are going to incorporate. Change the `<nav>` tag to `<nav class="topnav">` as per the code below.

The Code

```
<nav class="topnav">
```

Add JavaScript to our web design

Next we need to insert a place for our menu burger icon at the end of the unordered list of menu items. At the bottom of the `<nav>` section in between the `` and `</nav>` tags insert the following code.

```
<a class="icon" href="javascript:void(0);"
onclick="myFunction()"></a>
```

Your `<nav>` section should now resemble the screenshot at the top of the next page. If it does so repeat the last couple of steps to add the above lines of code into the

same place within the `<nav>` sections in your *contacts.html*, *cv.html* and *interests.html* webpages.

```

38 <nav class="topnav">
39
40 <ul>
41 <li>
42 <a title ="Home" href="#">Home</a>
43 </li>
44 <li>
45 <a title ="Curriculum Vitae" href="details/cv.html">Curriculum Vitae</a>
46
47 </li>
48 <li>
49 <a title ="My Interests" href="details/interests.html">My Interests</a>
50
51 </li>
52
53 <li>
54 <a title ="Contact Details" href="details/contacts.html">Contact Details</a>
55 </li>
56 </ul>
57
58 <a class="icon" href="javascript:void(0);" onclick="myFunction()"></a>
59
60 </nav>
61

```

Now we can introduce the Java script into our *html* pages.

Instructions

With the *index.html* page open in Notepad++ scroll up to the `<head>` section. Make some room under the line of code which starts `<link rel="stylesheet"` and enter the two lines of code below.

The Code

```

<script type="text/javascript" src="js/jquery-3.3.1.min.js"></script>
<script type="text/javascript" src="js/design.js"></script>

```

Now enter the same code, with a minor `../` adjustment, as shown below into the same place within the `<head>` section of your *contacts.html*, *cv.html* and *interests.html* webpages.

```

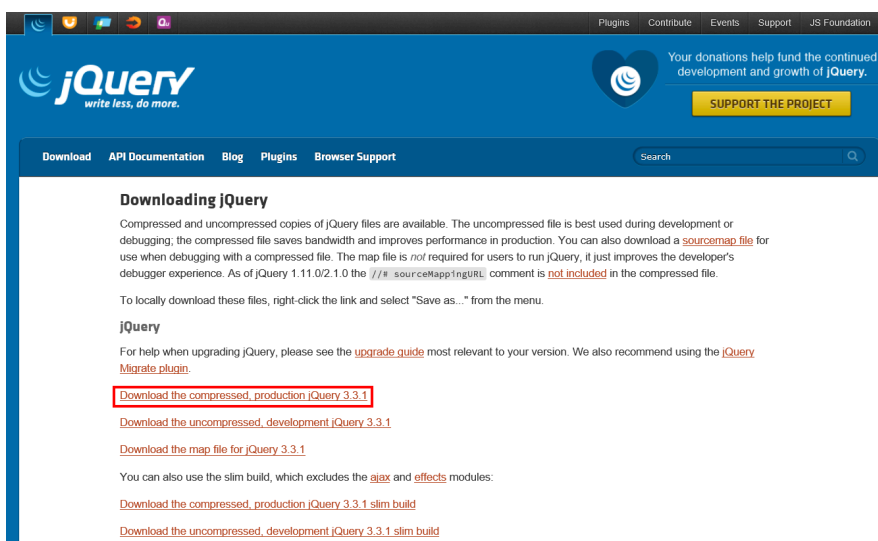
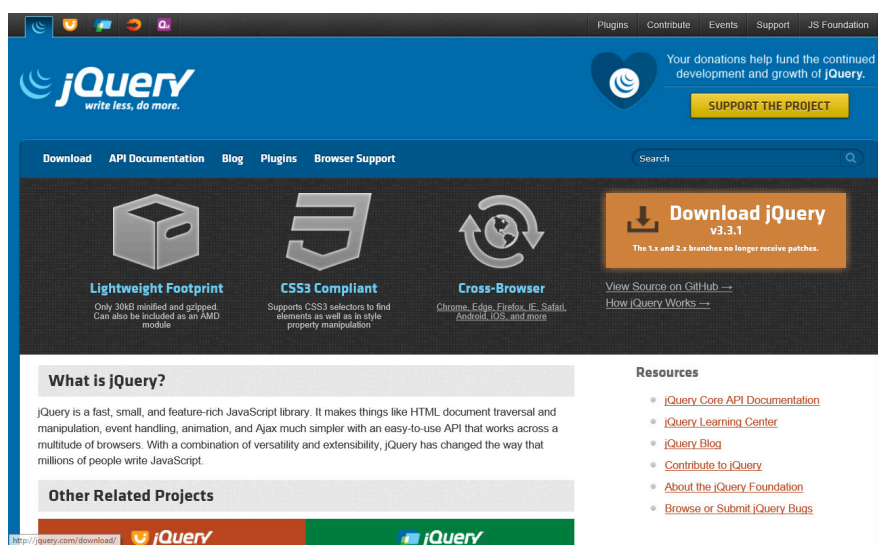
<script type="text/javascript" src="../js/jquery-3.3.1.min.js"></script>
<script type="text/javascript" src="../js/design.js"></script>

```

Explanation

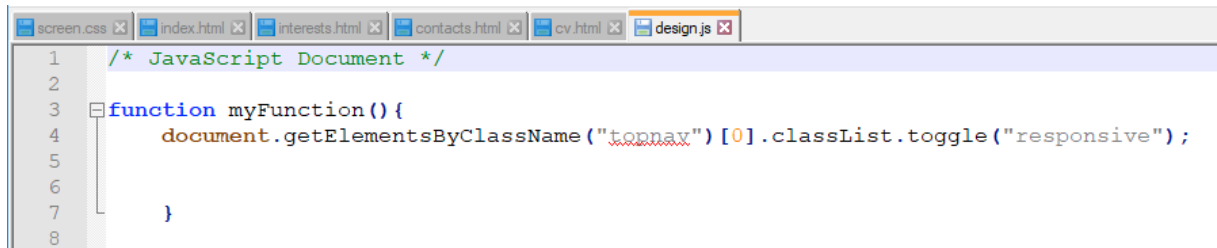
Our responsive website will have a menu button which will appear when the screen size is less than 650px wide. We want to be able to click on the menu button to show the navigation button links and then click the menu button again to hide the menu. Unfortunately this can't be done with html and css alone. We need some help from a Javascript jQuery. Your website folder already contains an additional folder called js which contains two files. We have just linked our four html pages to these two jQuery files.

If you need to use jQuery within your website it can be downloaded from <http://jquery.com/>. If you navigate to <http://jquery.com/> you can see how it is done. On the home page (shown below) click the orange **Download jQuery v3.3.1** button. On the next page (shown below) click on the '**Download the compressed, production jQuery 3.3.1**' link to save this to your downloads folder. The next step (**You don't need to do this!**) would be to move the file from downloads into your js folder within your website course files folder. If you look in your js folder you will see this has already been done for you.



Instructions

Now open the *design.js* file in Notepad++. It will look like the screen shot below.



```

1  /* JavaScript Document */
2
3  function myFunction() {
4      document.getElementsByClassName("topnav")[0].classList.toggle("responsive");
5
6
7  }
8

```

The JavaScript code works in conjunction with the code we inserted in our html `<nav>` section and the CSS targeting the responsive nav bar, which are both shown below. An explanation of how this works is given on the following page.



```

38  <nav class="topnav">
39
40      <ul>
41          <li>
42              <a title ="Home" href="#">Home</a>
43          </li>
44          <li>
45              <a title ="Curriculum Vitae" href="details/cv.html">Curriculum Vitae</a>
46          </li>
47          <li>
48              <a title ="My Interests" href="details/interests.html">My Interests</a>
49          </li>
50          <li>
51              <a title ="Contact Details" href="details/contacts.html">Contact Details</a>
52          </li>
53      </ul>
54
55      <a class="icon" href="javascript:void(0);" onclick="myFunction()"></a>
56
57  </nav>
58
59
60
61

```

```

@media screen and (max-width:650px) {

nav.topnav ul li {display: none; background-color: none;}
nav.topnav .icon {display: block; position: relative; float: right;
nav.topnav { background-color: rgba(0,0,0,.0); }

nav.topnav.responsive { z-index: 1; position: absolute; top: 247px;
nav.topnav.responsive .icon { position: relative; top: -263px; }
nav.topnav.responsive ul li {float: none; display: inline; }
nav.topnav.responsive ul {list-style: none; margin: auto; padding: 0; }
nav.topnav.responsive ul li a {display: block; background-color: #fff;
nav.topnav.responsive ul li a:hover { background-color: #ff6600; color: #fff; }

```

Explanation

We inserted this line of code `` into our `<nav>` section. Importantly we inserted it after the last link which is our *Contact Details* button. This is because we want our menu button to be at the right hand end of the *nav* bar after the row of buttons. This would be the natural position on a webpage without any styling attributes. The `` part allows us to toggle between adding and removing a `show` class which is used to show and hide drop down content. This connects with the following JavaScript code in our *design.js* document, shown below.

```
function myFunction(){
    document.getElementsByClassName("topnav")[0].classList.
toggle("responsive");
```

The first part of the code `'function myFunction(){'` simply declares the function which is then saved for future use. In our case it will be executed later when the menu button is clicked.

The `document.getElementsByClassName` part of the code returns a collection of all elements in the document. In our case this is "topnav" and "responsive" dependent on whether the menu button is clicked or not. In our case it is the CSS styling targeting how the `<nav>` section will look before we press the menu button "topnav" or after we press the menu button "responsive".

The following CSS attributes are used to target each state.

`nav.topnav .icon` targets the styling of the menu button

`nav.topnav` targets the styling of the menu bar

`nav.topnav.responsive` targets the styling of the menu bar after the menu button is clicked. This brings us to our next task, that of adding the CSS code to style the new menu and following sections if the page width is less than 650px.

Add a CSS media query for screens less than 650px wide

Instructions

On your *screen.css* document make a couple of lines of space below your last media query and type the following code shown at the top of the next page. This is the media query which will look to see if the screen is less than 650px wide and if so will activate the CSS styling attributes we are about to code below the media query line of text you are about to enter.

The Code

```
/* Break point for screens less than 650px wide. Hide menu
items and display the menu icon */

@media screen and (max-width:650px) {

    }
```

Use CSS to Style the Nav bar for screens less than 650px wide

In between the two { } enter the code below a section at a time. An explanation is shown underneath each part of the code.

```
nav.topnav ul li { display: none; background-color: none;}
nav.topnav .icon {
    display: block;
    position: relative;
    float: right;
    margin-right: 10px;
    margin-top: 5px;
    background: url(../images/menu.svg) no-repeat 0 0;
    background-size: contain;
    height: 40px; width: 40px; }
```

```
nav.topnav {background-color: rgba(0,0,0,.0); }
```

Explanation

Save the *screen.css* document and refresh the web page. The menu burger button should appear.

The first line `nav.topnav ul li { display: none; background-color: none;}` hides the original horizontal button menu.

The section which starts `nav.topnav .icon` targets the position and size of the menu button and locates the *menu.svg* file in the *images* folder.

The last line starting with `nav.topnav` simply turns off the semi opaque transparent bar which the menu buttons sit on when the screen size is over 650px wide.

```
nav.topnav.responsive { z-index: 1;
                        position: absolute;
                        top: 247px;
                        height: 275px;
                        background-color: #d9d9d9;
                        box-shadow: 0 12px 6px -6px #606060; }
```

Explanation

Save the `screen.css` document and refresh the web page.

This section introduces the responsive nav bar and makes the menu button work.

The `z-index: 1;` line makes the nav bar drop down over the content beneath it, opposed to pushing all the content down the page. We use a position of absolute to set it exactly where we want which is 247px from the top of the page and with a height of 275px. The background is coloured gray using `background-color: #d9d9d9;` and a box shadow is added for extra effect.

```
nav.topnav.responsive .icon { position: relative;
                              top: -263px; }
```

Explanation

Save the `screen.css` document and refresh the web page.

This line of code targets the menu button when the menu is in its responsive state. Although this has now moved the burger menu icon out of position it will be correctly positioned once the other list items have been added and when the responsive list is active.

```
nav.topnav.responsive ul li { float: none; display: block; }
```

Explanation

Save the `screen.css` document and refresh the web page. The menu items should appear as does the orange hover effect. This targeted the list items within the unordered list when the drop menu is active. The float left setting we had from the original menu bar is removed and the list menu items run vertically down the page.

```
nav.topnav.responsive ul { list-style: none;
                           margin: auto;
                           padding: 60px 10px 15px 10px; }
```

Explanation

Save the screen.css document and refresh the web page.

This line of code targets the unordered list menu items and sets their position. The padding settings of 60px at the top moves the menu items down the gray drop menu. The padding settings of 10px right and left target the orange hover effect boxes which are now positioned correctly.

```
nav.topnav.responsive ul li a {  
    display: block;  
    background-color: #fff;  
    color: #000;  
    text-align: center;  
    padding: 10px 20px;  
    text-decoration: none;  
    border: 2px solid #d9d9d9; }
```

Explanation

Save the screen.css document and refresh the web page.

This targets the individual anchor tags within the list items within the unordered list. The background color is set to white using `background-color: #fff;` which together with the padding settings of `padding: 10px 20px;` and the border settings of `border: 2px solid #d9d9d9; }` will make the anchor links look like buttons. The anchor link text is set to black using `color: #000;`. The anchor link text is center aligned using `text-align: center;` and finally the automatic anchor link underline is turned off using `text-decoration: none;`.

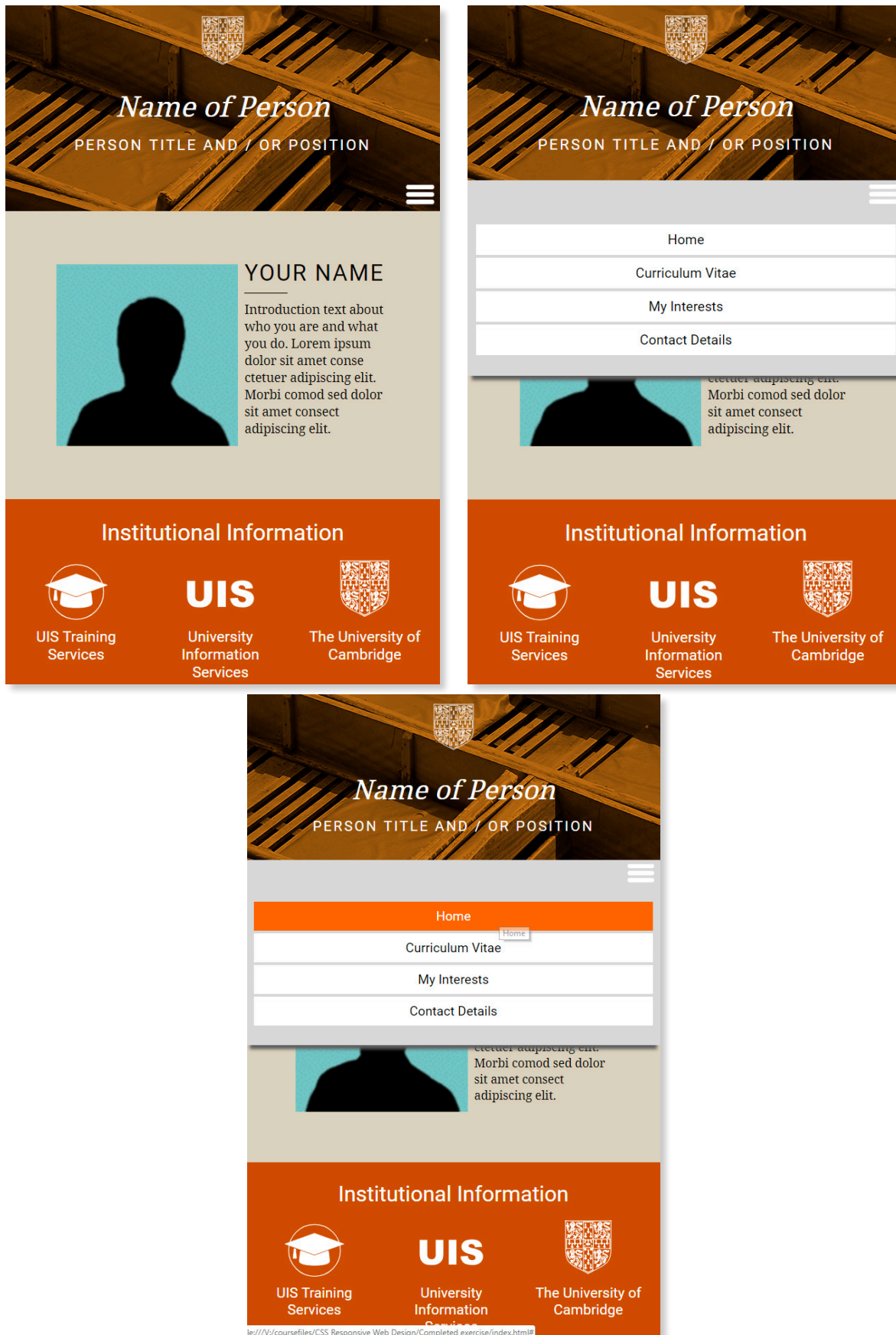
```
nav.topnav.responsive ul li a:hover {  
    background-color: #ff6600;  
    color: #fff;  
    transition: 0.4s; }
```

Explanation

Save the screen.css document and refresh the web page.

This last part targets the menu button hover effect. The white menu button changes to orange using `background-color: #ff6600;` and the button anchor link text changes to white using `color: #fff;`. Finally the `transition: 0.4s;` is used for added effect.

Save the screen.css document and refresh the web page.
Your menu should now resemble the screen shots below.



Redesign the Person Bio section for vertical layout

Now that we are changing the style of our website to a vertical layout which will suit mobile devices better we now need to redesign the *personbio* section. We are going to center the person image and then have the person name and introduction text centrally positioned under the image.

Instructions

In Notepad++ scroll down to the bottom of the screen.css document. Make some room after the last line of code and enter the following code a portion at a time, saving and refreshing as you go. An explanation follows each piece of code.

The Code

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

Explanation

Save the screen.css document and refresh the web page. You will notice that not much has changed. In actual fact because we are not changing the width, height and background colour at this point its arguable that we don't really need this line as the settings for this would be governed by the main styling code at the top of the document before the media query section. However adding it helps reaffirm what we want to do at 650px wide and having it in place now makes things easier if we choose to change the background colour or height in the future for this width of screen.

```
div.personimage {  
position: relative; top: 20px; left: 50%; margin-left: -100px;  
width: 200px; height: 200px;  
background: url(../images/person200x200.jpg) no-repeat 0 0;  
background-size: auto; }
```

Explanation

Save the screen.css document and refresh the web page.

The person image should now be in position. We positioned it relative to it's normal position, which would have been at top left corner of the *personbio* section. 20px from the top of the section and 50% across from the left hand side. The `margin-left: -100px;` piece of code keeps the image centered on the page, 100px being half the width of the image. We specified the space that the image would take up on the page using `width: 200px; height: 200px;` and gave the location of the image using `background: url(../images/person200x200.jpg) no-repeat 0 0;` and made sure the image was only shown once with no repeats. Notice that we have swapped the original image for a smaller one as this will look better on smaller screens. Lastly the `background-size: auto;` was used. This is the default value for this property and makes sure the background image is displayed in its original size.

```
div.persontext {  
    position: absolute; margin-top: 170px; left: 5%;  
    width: 90%;  
    text-align: center; }
```

Explanation

Save the screen.css document and refresh the web page.

The text should now be centered underneath the image. We first used a position of absolute. This goes back to the PowerPoint presentation in the first CSS course when I was talking about the box model and a div within a div. That's what we have here, a box in a box (a div in a div). The first box is actually the main *personbio* section. We then have a box to hold the image and that was positioned using the property of *relative*. The next box then has to use a property of *absolute*. Using two properties of *relative* or two properties of *absolute* just won't work. If you add the following code `border: 1px solid blue;` into the last line of text in the last three sections of code you have just added and you will see what I mean by a box within a box. Remove the border code once you are done. The `margin-top: 170px;` positions the box holding the text 170px down from the top of the section. If we had used the selector of `top` then it would have been positioned 170px down from the top of the web page and therefore in the wrong position. The `left: 5%; width: 90%;` part of the code simply specifies that the text box will take up 90% of the page width and will be positioned 5% in from the left, which will automatically give 5% on the right and therefore some space either side of the text box. Lastly `text-align: center;` centers the text.

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

Explanation

Save the screen.css document and refresh the web page.

Here we are targeting the h2 heading within the div of persontext. The letter spacing property of 2px spreads the letters out a bit for effect. The colour is set to black and the line height property makes some room underneath the h2 heading before the horizontal rule. The way this works is as follows. If the line height is specified using a number, as in `line-height: .75em`; then the total space under the text is the value of the line height number multiplied with the current font-size. Try changing `.75em` to `1.75em` and see what happens.

```
div.persontext hr {  
  width: 30%;  
  text-align: center;  
  margin-left: 35%;  
  margin-bottom: 10px;  
  color: #000;  
  height: 1px;  
  background-color: #000;  
  border: 0 none; }
```

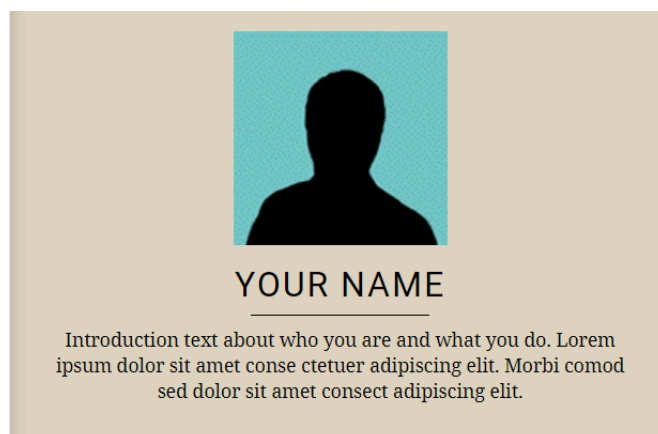
Explanation

Save the screen.css document and refresh the web page.

Here we are targeting the horizontal rule within the div of persontext.

The width is set to 30% and the hr is centered on the page. It's given a margin of 35% on the left which will automatically give 35% on the right and therefore help keep the hr centered. $30\% + 35\% + 35\% = 100\%$. The `margin-bottom: 10px`; simply makes some room under the hr before the paragraph text starts. The rest of the code targets the color and thickness of the line and makes sure it works across different browsers as some look at the color and height properties of the hr while others are effected by the background-color and border properties. At this point we don't need to target the paragraph text under the heading as the styling for that has already been set at the top of the page and at the moment the text size looks fine.

Your *personbio* section should now resemble the screen shot below.



Redesign the Institutional Information section for vertical layout

We are now ready to change the *institutionalinfo* section to a vertical layout.

In Notepad++ scroll down to the bottom of the screen.css document.

Make some room after the last line of code and enter the following code a portion at a time, saving and refreshing as you go. An explanation follows each piece of code.

The Code

```
/* Changes to Institutional Information section */  
  
section.institutionalinfo { width: 100%; height: 740px;  
                           background-color: #cb4f00; }
```

Explanation

Save the screen.css document and refresh the web page.

Again we start by adding a comment so we know what this section of code refers to. Then we target the section itself. The width is kept the same at 100% so it stretches across the whole web page and the orange background colour is kept the same. The main thing which has changed is the height setting. We have increased it to 740px to accommodate the boxes containing the graphic, web link and text which will now run vertically down the page.

```
section.institutionalinfo h2 {  
    position: relative;  
    top: 30px;  
    width: 100%;  
    text-align: center;  
    color: #fff;  
    margin-bottom: 35px; }
```

Explanation

Save the screen.css document and refresh the web page.

Here we are targeting the **h2** heading inside the *institutionalinfo* section. We position it **30px** from the top **relative** to its normal position. We don't have to worry about left or right position settings as using a width of 100% makes this container holding the **h2** text run the full width of the web page. Add **border: 1px solid white;**

after the `margin-bottom: 35px;` line of code and you will see what I mean. Remove the border line of code once you are done. `text-align: center;` and `color: #fff;` simply centers the text within it's container and maintains the white text colour from before. The `margin-bottom: 35px;` property makes some space under the heading to help push down the first graphic which will be the mortar board for UIS Training Services.

```
section.institutionalinfo aside {  
    width: 100%; text-align: center; }
```

Explanation

Save the screen.css document and refresh the web page.

The three aside sections should now be placed underneath each other vertically down the page. The bottom one runs over the footer at the bottom, that will be fixed after the next section of code. If you look back to our original styling for this section you will see we had the following settings:

```
width: 33%; float: left; text-align: center;
```

This made the three aside boxes line up horizontally across the page with each one taking up a third of the page width. Simply by removing the `float` property and assigning a width of `100%` the boxes now line up vertically instead of horizontally. An easy change to make and far less complicated than you may think. To see these three aside boxes in action add a temporary border after the last line of code.

```
section.institutionalinfo aside .content {  
    margin: 5px 10px 0px 10px;  
    background: no-repeat center top; }
```

Explanation

Save the screen.css document and refresh the web page.

This section of code targets the content inside the container. If you add a temporary border you will see the `content` section more clearly. Remove the border afterwards. The content is actually the `.svg` graphics. A margin of 5px at the top and 10px left and right is added by using the `margin: 5px 10px 0px 10px;` line to make some room around each box. The `background: no-repeat center top;` line of code targets the position of all three `.svg` graphics in one go and positions them inside, centrally and at the top, of each box and makes sure we only see one graphic and not lots of them repeating.

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

Explanation

Save the screen.css document and refresh the web page.

Here we are targeting the `h4` heading inside the `aside` which is inside the `institutionalinfo` section. Arguably we don't need the `color: #fff;` property but its handy to include it should we want to change the text colour in the future. The `text-decoration: none;` property turns off the automatic underline which is the default for any web page text with a link. We then turn it on when the mouse moves over it using the following line:

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

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

Explanation

Save the screen.css document and refresh the web page.

Nothing will have changed here as this line of code is not really needed, however there is a good reason for including it now. If we didn't have it nothing would change as the paragraph text size attributes would be styled using the properties at the very top of our `screen.css` document. The text color would be styled white anyway as that is targeted in the original `institutionalinfo` section for large screens. The reason for adding it will become clear later but in short when we restyle our code for even smaller screens we will simply copy and paste this code further down the page. We can then target the paragraph text size for this section to make it smaller. Having this line included now will save time later.

```
section.institutionalinfo aside .content.trainingservices {  
    background-size: 90px 90px; }  
  
section.institutionalinfo aside .content.uis {  
    background-size: 95px 95px; }  
  
section.institutionalinfo aside .content.uniofcam {  
    background-size: 80px 80px; }
```

Explanation

Save the screen.css document and refresh the web page.

Here we are targeting the size of each of the .svg graphics. These are inserted as a background image within the three aside content containers, `.trainingservices`, `.uis` and `.uniofcam`. The size of each graphic is actually staying the same as per the original styling for this section for large screens further up the page. The reason for adding it now is so we can copy and paste it further down the page to make changes later for even smaller screens. There is no need to state the location of each graphic as that is still being targeted by the original lines of code in the styling for large screens, as shown below:

```
100
101 section.institutionalinfo aside .content.trainingservices {background-image: url(../images/mortarboard.svg); background-size: 95px 95px; background-repeat: no-repeat; background-position: center; background-color: #f4a460;}
102 section.institutionalinfo aside .content.uis {background-image: url(../images/uis.svg); background-size: 95px 95px; background-repeat: no-repeat; background-position: center; background-color: #f4a460;}
103 section.institutionalinfo aside .content.uniofcam {background-image: url(../images/camshield.svg); background-size: 95px 95px; background-repeat: no-repeat; background-position: center; background-color: #f4a460;}
104
```

Your *institutionalinfo* section should now resemble the screen shot below.



Redesign the CV section for a vertical layout

We are now ready to style the *cv* section on the Curriculum Vitae page. However there is a slight change we have to make first to our *cv.html* document. Both the *cv.html* page and the *contacts.html* page have buttons in them. In the original CSS coding for large screens we applied the button styling to both buttons in one go using a `class` of `btn`, as shown below:

```
<a class="btn"
```

However our two buttons are different lengths due to the amount of text in each. Therefore it will be impossible to centrally align both in one go when we change to a vertical layout. One button will be aligned in the center of the page, the other will not. Therefore we need to treat the positioning of each button separately. To do this we will modify the class name for each button.

Instructions

In Notepad++ open the *cv.html* document.

Find the section which starts `<section class="cv">`.

Within this section locate the line which starts `<a class="btn"`.

Change the class name to `"btncv"` to it looks like `<a class="btncv"`.

Save the *cv.html* document.

Repeat the previous steps above for the *contacts.html* document.

Locate the section of code starting `<section class="contact">`.

Change the button class name to `<a class="btncontact"`.

Save the *contacts.html* document.

Still in Notepad++ return to the *screen.css* document and scroll up to the original styling code for large screens until you find the section which starts `section.cv`. At the bottom of this section change the `.btn` `{` line of code to `.btncv` `{`.

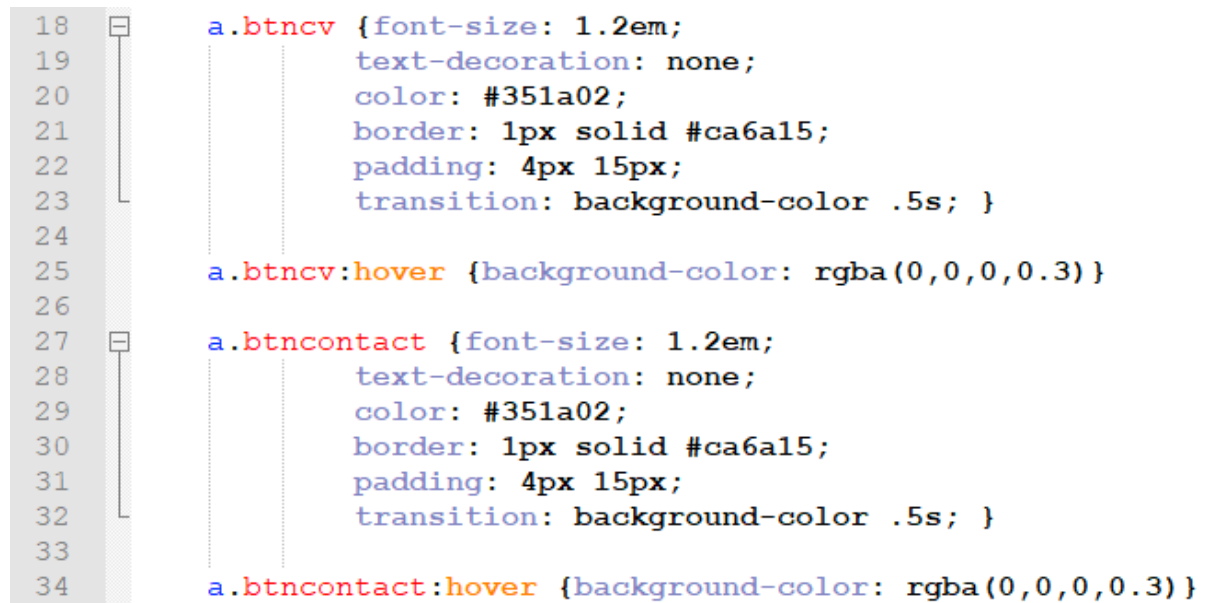
Select and copy this line of text. Scroll down the page a bit until you come to the section which starts `section.contact`. At the bottom of this section underneath the line `div.contacttext p {color: #000; }` paste the button line of code into the document. Rename the start of the line from `.btncv` to `.btncontact`.

Still on the *screen.css* document scroll right to the top of the document and locate the following section of code as shown below:

```
17
18  a.btn {font-size: 1.2em;
19      text-decoration: none;
20      color: #351a02;
21      border: 1px solid #ca6a15;
22      padding: 4px 15px;
23      transition: background-color .5s; }
24
25  a.btn:hover {background-color: rgba(0,0,0,0.3) }
```

Rename `a.btn` to `a.btncv` on both lines of code.

Copy and paste this section below and rename `a.btncv` to `a.btncontact` on the pasted text. The two sections of text should now look like the screen shot below:



```
18  a.btncv {font-size: 1.2em;
19      text-decoration: none;
20      color: #351a02;
21      border: 1px solid #ca6a15;
22      padding: 4px 15px;
23      transition: background-color .5s; }
24
25  a.btncv:hover {background-color: rgba(0,0,0,0.3) }
26
27  a.btncontact {font-size: 1.2em;
28      text-decoration: none;
29      color: #351a02;
30      border: 1px solid #ca6a15;
31      padding: 4px 15px;
32      transition: background-color .5s; }
33
34  a.btncontact:hover {background-color: rgba(0,0,0,0.3) }
```

Scroll back down to the bottom of the `screen.css` document and make some room below the last line of code. Enter the following code a portion at a time, saving and refreshing as you go. An explanation follows each piece of code.

The Code

```
/* Changes to CV section */

div.cvtext { position: absolute;
left: 0px;
width: 100%;
margin: 60px 0 0 0;
text-align: center; }
```

Explanation

Save the `screen.css` document and refresh the web page.

We start by targeting the container `div.cvtext` which holds the text. It is given an `absolute` position and aligned to the left hand side of the container using the `left: 0px;` property. The width of the container is made to stretch the full width of the screen by using the `width: 100%;` property. A margin of 60px at the top is applied to push the text box down and to create some room at the top of the section.

Finally the text is centered using the `text-align: center;` property.

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

Explanation

Save the `screen.css` document and refresh the web page.

Next we target the heading 2 `h2` text title of 'Curriculum Vitae'. Just for emphasis and effect we increase the space between the letters using `letter-spacing: 2px;`. The text colour is set to black and a margin of 15px underneath the text is used to create some room before the horizontal bar.

```
div.cvtext hr {  
    width: 30%;  
    text-align: center;  
    margin-left: 35%;  
    margin-bottom: 10px;  
    color: #000;  
    height: 1px;  
    background-color: #000;  
    border: 0 none; }
```

Explanation

Save the `screen.css` document and refresh the web page.

Here we are targeting the horizontal bar in exactly the same way as per the *personbio section*. See page 15 for a full explanation.

```
div.cvtext p { color: #000; margin: 30px 30px 0 30px; }
```

Explanation

Save the `screen.css` document and refresh the web page.

Now we target the paragraph text. The text colour is set to black and a margin of 30px at the top and 30px to the left and right is applied using the property of `margin: 30px 30px 0 30px;`. This creates some room between the horizontal rule and the start of the paragraph text and creates some space either side of the text so it looks more centered.

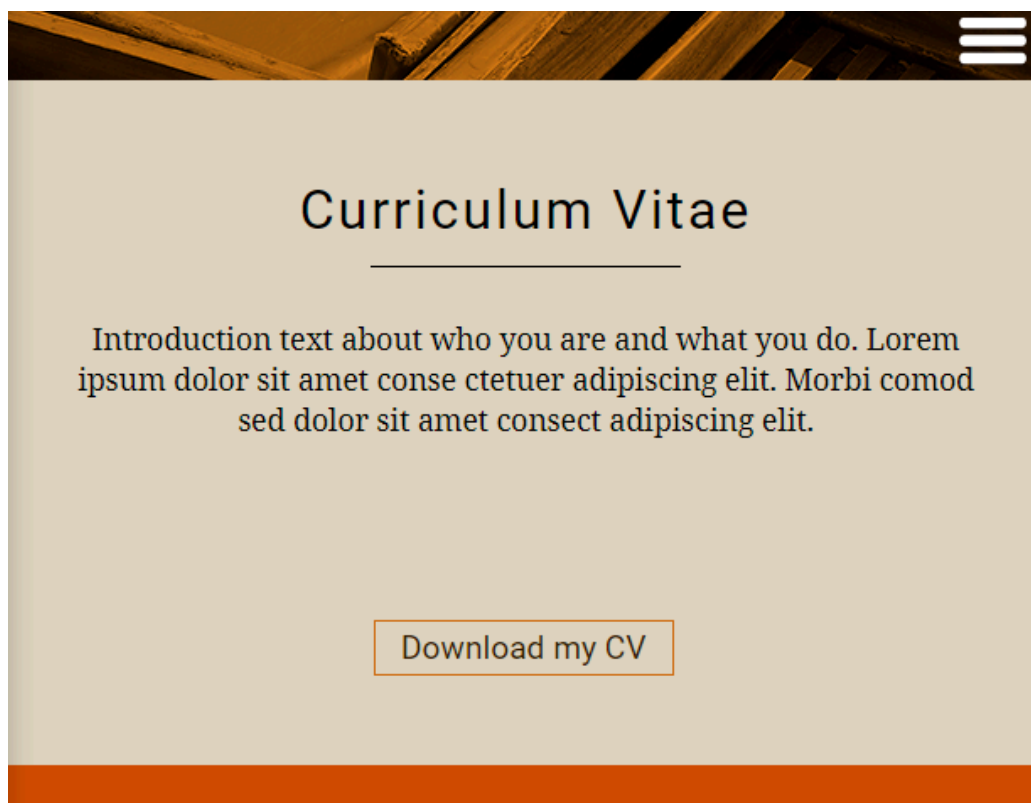
```
a.btncv { position: relative;
         top: 325px;
         left: 50%;
         margin-left: -90px;
         text-align: center; }
```

Explanation

Save the *screen.css* document and refresh the web page.

Finally we can reposition the button. We use a position of relative as we have already used the position of absolute for the text box. At the moment the button is maintaining the position it was given in the styling for a large screen size. The button is pushed further down the page by using the `top: 325px;` property. We want the button positioned dead center on the page so we use a combination of the `left: 50%;` property and the `margin-left: -90px;` property. To work out what the length of the button is, remember we have not specified the length as the button automatically sizes, depending on the amount of text in the box. So a screen shot was taken and opened in Photoshop. It was then cropped and the resulting image size of 180px was the length of the button in pixels. The `margin-left:` setting requires half the length of the button to be inserted as a minus number to make sure the button stays aligned to the center of the page. `text-align: center;` is just to make sure the text inside the button remains centered.

Your cv section should now resemble the screen shot below.



Redesign the My Interests section for a vertical layout

We are now ready to change the *interests* section to a vertical layout.

In Notepad++ scroll down to the bottom of the screen.css document. Make some room after the last line of code and enter the following code a portion at a time, saving and refreshing as you go. An explanation follows each piece of code.

The Code

```
/* Changes to My Interests section */  
div.intereststable h2 { text-align: center; }
```

Explanation

Save the screen.css document and refresh the web page.

This is actually a very quick change. The table is still resizing automatically and still looks good at this size. The only thing we are going to do here is center the 'My Interest' heading.

Your *interests* section should now resemble the screen shot below.



Redesign the Contact section for a vertical layout

We are now ready to change the *contact* section to a vertical layout.

In Notepad++ scroll down to the bottom of the screen.css document.

Make some room after the last line of code and enter the following code a portion at a time, saving and refreshing as you go. An explanation follows each piece of code.

The Code

```
/* Changes to Contact section */  
  
div.contacttext {  
    position: absolute;  
    left: 0px;  
    width: 100%;  
    margin: 60px 0 0 0;  
    text-align: center; }
```

Explanation

Save the screen.css document and refresh the web page.

Here we are targeting the container which holds the text in this section which is called `div.contacttext`. The styling for this section is exactly the same as that used for the `cvtext` section as we want the heading of 'Contact Information' to be positioned in the same place as per the 'Curriculum Vitae' section for continuity. It is given an `absolute` position and aligned to the left hand side of the container using the `left: 0px;` property. The width of the container is made to stretch the full width of the screen by using the `width: 100%;` property. A margin of 60px at the top is applied by using the `margin: 60px 0 0 0;` property to push the text box down and to create some room at the top of the section. Finally the text is centered using the `text-align: center;` property. If you want to see this container in action then add a border to the last line of code. `border: 1px solid white;`

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

Explanation

Save the *screen.css* document and refresh the web page.

Next we target the heading 2 `h2` text title of 'Contact Information' and this is treated in exactly the same way and with the same code as per the `cvtext` section. Again we increase the space between the letters using `letter-spacing: 2px;`. The text colour is set to black and a margin of 15px underneath the text is use to create some room before the horizontal bar.

```
div.contacttext hr {  
    width: 30%;  
    text-align: center;  
    margin-left: 35%;  
    margin-bottom: 10px;  
    color: #000;  
    height: 1px;  
    background-color: #000;  
    border: 0 none; }
```

Explanation

Save the *screen.css* document and refresh the web page.

Here we are targeting the horizontal bar in exactly the same way as per the *personbio* section. See page 15 for a full explanation.

```
a.btncontact {  
    position: relative;  
    top: 325px;  
    left: 50%;  
    margin-left: -56px;  
    text-align: center; }
```

Explanation

Save the *screen.css* document and refresh the web page.

Except for one small change, these are the same settings we used for the `btncv` button, therefore the explanation on page 23 for the `btncv` button is the same as for this `btncontact` button. The only change is the `margin-left: -56px;` property, simply because the `btncontact` button, which simply says 'Email me' is shorter than the 'Download my CV' `btncv` button. The length of the 'Email me' button is 112px long hence the `-56px` property. This will make the button line up centrally on the page and is also why we had to modify our code to treat each button separately at the start of this practical.

The 'Contact Information' section on your *contacts.html* page should now resemble

the screen shot below:

Contact Information

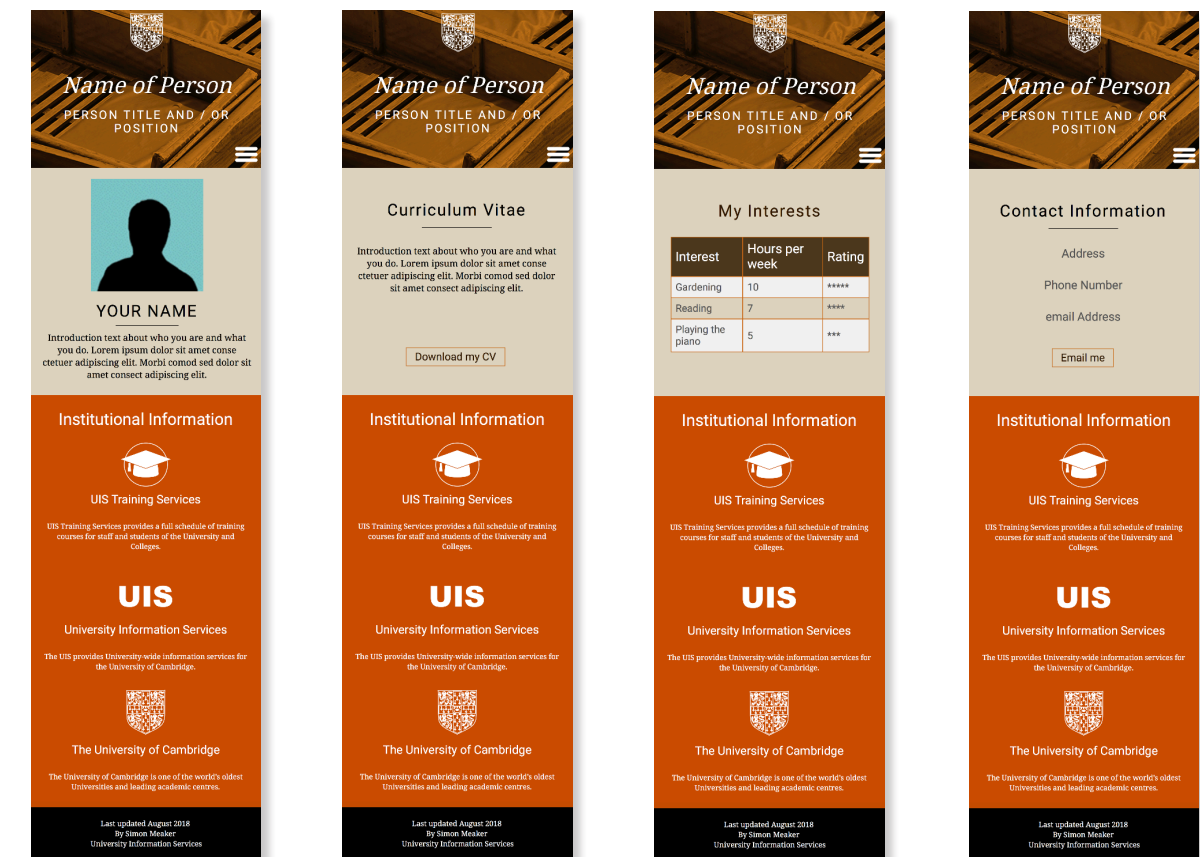
Address

Phone Number

email Address

Email me

Your completed 650px wide website should now resemble the images below.



With the majority of our CSS coding complete for our vertical redesign with burger menu, we are now going to target large mobile phones like the iPhone 8 and iPhone 8+. The iPhone 8+ has a device width of 414px and the slightly smaller iPhone 8 has a device width of 375px. There are so many different makes and models of mobile device on the market all with different device widths that trying to target every single one would be a never ending task. Instead we are going to target devices which have a maximum width of 414px. Anything above 414px will be targeted with the previous styling we have just finished for a maximum device width of 650px. Anything above that will be targeted by our original coding for large screens. At the end we will target smaller devices which have a device width of 320px or less.

Add a Media Query to target large mobile phones

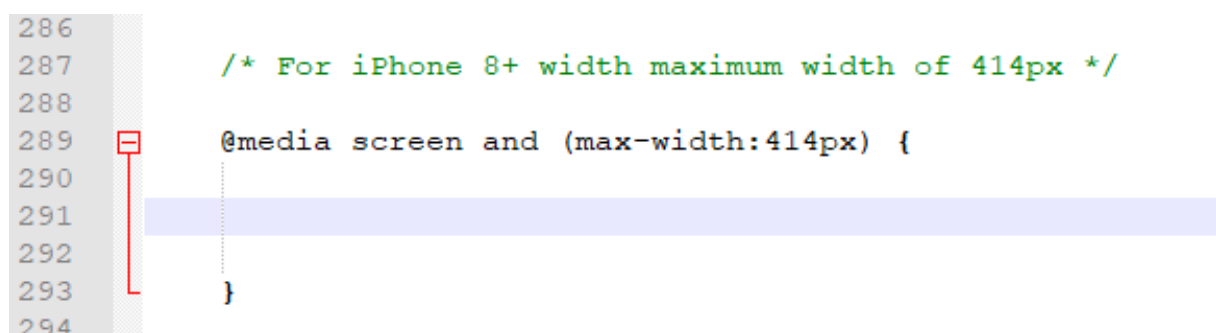
Instructions

With the `screen.css` page open in Notepad++ scroll down to the last line of code which starts `a.btncontact { position: relative;.....`. There should be an ending `}` at the end of the line. Underneath this line there is another `}`. This is the ending bracket for the media query `@media screen and (max-width:650px)` `{`. Our next media query needs to be inserted below all of this. So underneath the very last `{` enter the code below.

The Code

```
/* For iPhone 8+ maximum width of 414px */  
  
    @media screen and (max-width:414px) {  
  
    }
```

Put some line breaks in-between the `{ }` brackets as shown in the screen shot below. This is where the next piece of code will go, as indicated by the blue line on the screen shot.



```
286 /* For iPhone 8+ width maximum width of 414px */  
287  
288 @media screen and (max-width:414px) {  
289  
290  
291  
292  
293 }  
294
```

Change the <body> background to black

At the moment we have a white background, except for screens larger than 1200px wide where we have an image. The white background could show up on some mobile phone devices during page scrolling as some allow you to scroll slightly above and below the web page. This would then show as a white line. This looks a bit weird so we are going to make our background black to hide this effect a bit.

Instructions

In Notepad++ on our *screen.css* document click in-between the { } brackets as described on the previous page. As before enter the following code a portion at a time, saving and refreshing as you go.

The Code

```
/* Make background black */  
  
body { background-color: #000; }
```

Adjust the font size in the “personbio” section

Using the Chrome developer tools make the page width smaller until it gets to 413px wide. The paragraph text within the *personbio* section should not be overlapping the orange Institutional Information section below. We are going to adjust the text size within the *personbio* section to stop this from happening.

```
/* Make text smaller in PersonBio section */  
  
div.persontext h2 { font-size: 1.8em;}  
div.persontext p { font-size: 1.0em;}
```

Explanation

Save the *screen.css* document and refresh the web page.

Again we start with a comment. This is useful to do as we can keep track of the changes we are making. We start by targeting the h2 heading within the *div* of *persontext*, hence the line `div.persontext h2 { font-size: 1.8em;}`. The original text size for the h2 heading was set right at the beginning of the CSS For

Beginners course and currently resides at the very top of our *screen.css* document. Our default h2 font size was set to 2em. We are overriding this setting with a font size of 1.8em for this section. The second thing we are targeting is the size of the paragraph font. Again we have a default font size of 1.1em at the top of the page. For the *personbio* section we are overriding this and changing to a slightly smaller font size of 1.0em. These two settings will make both pieces of text smaller and prevent them from overlapping the orange Institutional Information section.

Adjust font size and margins of the “Institutional Info” section

Looking again at our web page at width of 413px using the Chrome developer tools, we can see the paragraph text within the *institutionalinfo* section looks a bit large for this screen size, and is overlapping the *footer* section at the bottom of the page. An adjustment to the font size is required.

```
/* Adjust font size and margins within Institutional Informa-
tion section */

section.institutionalinfo h2 {
    font-size: 1.8em; margin-bottom: 50px; }

section.institutionalinfo aside p {
    font-size: 0.8em; margin: 0 1em 1em 1em; }
```

Explanation

Save the *screen.css* document and refresh the web page.

We start by targeting the h2 heading within the *institutionalinfo* section. The font size is reduced down to 1.8em from the default 2em which was set at the top of the CSS document. The *margin-bottom*: property setting is increased from 35px to 50px to add a bit more room in-between the ‘Institutional Information’ heading and the first .svg graphic. The second line targets the paragraph font within the aside element within the *institutionalinfo* section. The font is reduced in size to 0.8em which tightens the text up nicely and a margin of 1em on the right, bottom and left of the paragraph text pulls the text in from the edges of the screen so it looks neater.

Adjust h2 and p font sizes in the “Curriculum Vitae” section

```
/* Adjust font size of h2 and p text in CV section */

div.cvtext h2{ font-size: 1.8em; }
div.cvtext p{ font-size: 1.0em; }
```

Explanation

Save the *screen.css* document and refresh the web page.

We start by targeting the h2 heading within the *cvtext* section. The font size is reduced down to 1.8em from the default 2em which was set at the top of the CSS document. This matches the font size in the *personbio* section on the home page for this screen size. The second line targets the paragraph font within the *cvtext* section. The font is reduced in size to 1.0em from the default 1.1em which looks a bit better for smaller screens.

Adjust h2 font size in the “Interests” section

```
/* Adjust font size of h2 heading in My Interests section */  
div.intereststable h2 { font-size: 1.8em; }
```

Explanation

Save the *screen.css* document and refresh the web page.

The only change we are targeting here is the h2 heading above the *intereststable* within the *interests* section. The font size is reduced down to 1.8em from the default 2em so it matches the font size in the *personbio* section and the *cvtext* section.

Adjust h2 font size in the “Contact” section

```
/* Adjust font size of h2 heading in contact section */  
div.contacttext h2 { font-size: 1.8em; }
```

Explanation

Save the *screen.css* document and refresh the web page.

Again the only change we are targeting here is the h2 heading of ‘Contact Information’ in the *contact section* on the *contacts.html* page. The font size is reduced down to 1.8em from the default 2em so it matches the font size on our other pages.

That's it, we are finished making our adjustments for screen sizes up to 414px wide. As the hard work was done when we changed the layout of our website for mobile devices the changes we have made in the section are fairly quick to do. We simply target the items we want to change. Anything we don't modify retains it's setting from further up the page.

Now in the final section of this practical we will make some further changes to accommodate smaller mobile phones and devices like the iPhone 5 and iPhone SE up to 320px wide.

Using the Chrome developer tools make the page width smaller until it gets to 319px wide. The header section is now too large for this size of screen so we will swap out the header image for a smaller one to reduce the height of the header section and will then adjust the font size. The paragraph text within the *personbio* section is again overlapping the orange Institutional Information section below. We will adjust the font size within the *personbio* section to stop this from happening and then make the same change to the other sections. The text at the bottom of the Institutional Information section is running down into the *footer* section. We will adjust the size of the .svg graphics and fonts in the Institutional Information section to combat this. The My Interests table also needs some adjustment.

Add Media Query to target small mobile phones

Instructions

With the *screen.css* page open in Notepad++ scroll down to beneath the last `}` which ends the previous media query. Make some room underneath and enter the code below for a new media query.

The Code

```
/* Break point for small devices with screen width of 320px */  
  
@media screen and (max-width:320px) {  
  
}
```

As described on page 28 put some line breaks in-between the starting and ending `{ }` brackets for this media query. As before enter the following code a portion at a time in between the brackets, saving and refreshing as you go.

Swap the header image for a smaller one

```
/* Swap header image for smaller one, adjust header height to  
accommodate new image and adjust font size of header text*/  
  
header {  
    height: 190px;  
    background: url(../images/bannerpuntsbrown320_190.png)  
no-repeat; }
```

Explanation

Save the *screen.css* document and refresh the web page.

The header image of the Cambridge Punts should have changed. We started by reducing the height of the header down to 190px from the original 290px which has been the header size since the start. A height of 290px would have been too large for the small screen size we are now targeting. We then specify the location of the image including the **no-repeat**. This should make the image fill the new header size of 320px x 190px.

Adjust the font size within the header section

Looking at the header section of our web page, the text is now too large for this size of screen and is now overlapping the *personbio* section. An adjustment to the font size will pull the text back into place.

```
div.titletext {  
    position: relative;  
    width: 100%;  
    top: 90px;  
    text-align: center; }
```

Explanation

Save the *screen.css* document and refresh the web page.

We start by targeting the container which holds the header text, which is **div.titletext**. Up to this point the positioning of this container has been 120px from the top of the page because the height of the header has remained constant at 290px in height. Now that we have changed the height of the header from 290px to 190px the **titletext** container needs to be pulled back up the page to 90px from the top. The other settings remain the same.

```
h1.titletext {
    margin: 0 0 0.4em 0;
    font-style: italic;
    color: #fff;
    font-size: 1.6em; }

h3.titletext {
    color: #fff;
    letter-spacing: 3px;
    font-size: 0.7em; }
```

Explanation

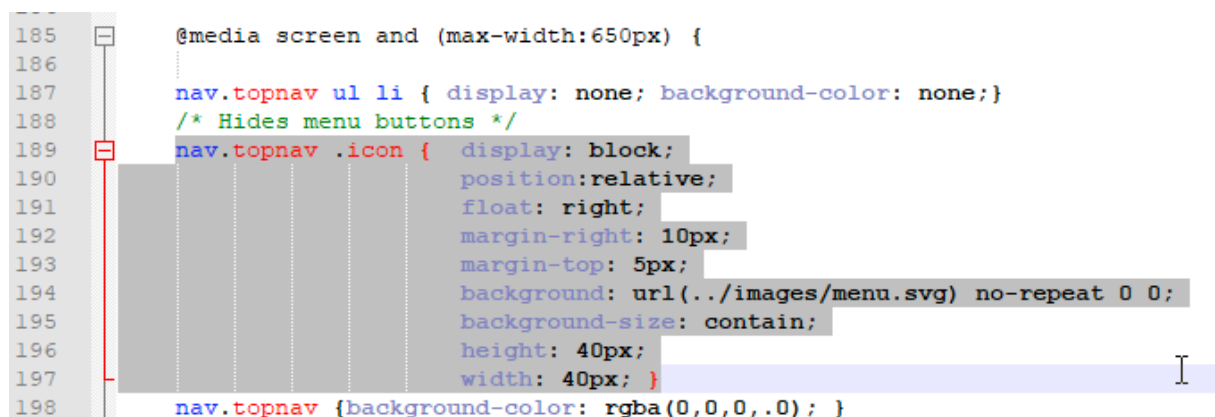
Save the *screen.css* document and refresh the web page.

The **h1** font size for the heading of 'Name of Person' has been reduced to 1.6em from it's default size of 2.5em which has been the case up until now. The space underneath this heading was also reduced from the default 1em to 0.4em using the **margin: 0 0 0.4em 0;** property.

The **h3** font size is also reduced down to 0.7em from it's default of 1.3em which was also originally set at the top of the *screen.css* document. The other settings remain the same

Adjust the position of the menu icon

In Notepad++ scroll up the *screen.css* document until you come to the **nav.topnav .icon** line just under the start of the media query **@media screen and (max-width:650px)** code. Select the code as shown in the screen shot below and copy it.



```
185 @media screen and (max-width:650px) {
186
187     nav.topnav ul li { display: none; background-color: none;}
188     /* Hides menu buttons */
189     nav.topnav .icon { display: block;
190                        position: relative;
191                        float: right;
192                        margin-right: 10px;
193                        margin-top: 5px;
194                        background: url(../images/menu.svg) no-repeat 0 0;
195                        background-size: contain;
196                        height: 40px;
197                        width: 40px; }
198     nav.topnav {background-color: rgba(0,0,0,.0); }
```

Scroll back down to the bottom of your *screen.css* document to the previous line of code which you have just entered. Add a new comment and paste in the code. It should look like the code section on the next page.

```
/* Adjust Navigation menu for small screens */  
nav.topnav .icon {  
    display: block;  
    position: relative;  
    float: right;  
    margin-right: 10px;  
    margin-top: 5px;  
    background: url(../images/menu.svg) no-repeat 0 0;  
    background-size: contain;  
    height: 40px;  
    width: 40px; }
```

Instructions

Now make the following changes to the code you have just copied:

```
margin-top: -86px;  
height: 30px;  
width: 30px;
```

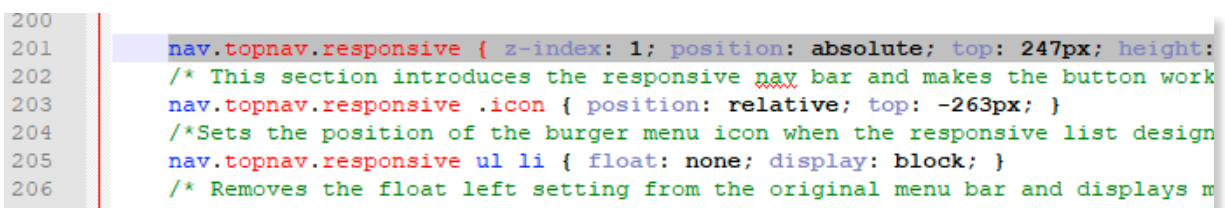
Explanation

Save the *screen.css* document and refresh the web page.

The menu icon should now be a bit smaller and should have moved up the screen back to its original position. Now if you click on the menu icon you will see that the responsive menu is now too far down the page due to the change we made to the header height. Our next job is to pull that back into place. Copying the original code from further up the *screen.css* document, pasting it in place, then making changes to it, is a quick way of modifying the code without typing everything out again.

Adjust the position of the responsive drop down menu

In Notepad++ scroll up the *screen.css* document until you come to the `nav.topnav.responsive` line within the media query `@media screen and (max-width: 650px)` code. Select the top line of code as shown in the screen shot below and copy it.



```
200  
201 nav.topnav.responsive { z-index: 1; position: absolute; top: 247px; height:  
202 /* This section introduces the responsive nav bar and makes the button work  
203 nav.topnav.responsive .icon { position: relative; top: -263px; }  
204 /*Sets the position of the burger menu icon when the responsive list design  
205 nav.topnav.responsive ul li { float: none; display: block; }  
206 /* Removes the float left setting from the original menu bar and displays m
```

Scroll back down to the bottom of your *screen.css* document and underneath the last line of code you entered, paste in the copied text. It should look like the section of code below.

```
nav.topnav.responsive {  
    z-index: 1;  
    position: absolute;  
    top: 247px;  
    height: 275px;  
    background-color: #d9d9d9;  
    box-shadow: 0 12px 6px -6px #606060; }
```

Instructions

Now make the following changes to the code you have just copied:

`top: 153px;`

Explanation

Save the *screen.css* document and refresh the web page.

All that was required here was a quick height adjustment to the `top` property setting. The responsive menu should now be in the correct place, however the menu button will have moved up. That's our next job.

Adjust the position of the responsive menu icon

In Notepad++ scroll up the *screen.css* document until you come to the `nav.topnav.responsive .icon` line within the media query `@media screen and (max-width:650px)` code. Select the line of code and copy it. Then paste the copied text in place at the bottom of your *screen.css* document. It should look like the section of code below.

```
nav.topnav.responsive .icon {  
    position: relative; top: -263px; }
```

Instructions

Now modify the `top` property so the code reads `top: -169px;`

Explanation

Save the *screen.css* document and refresh the web page.

Again all that was required here was a quick height adjustment to the `top` property setting to move the responsive menu icon back to it's correct location.

Adjust the font position for the Personbio section

As you can see the text in our *personbio* section is now too far down the page and is overlapping the 'Institutional Information section'. We will be pulling our text box back up into place and adjusting the font size of our text.

In Notepad++ scroll up the *screen.css* document until you come to the `div.persontext` line within the media query `@media screen and (max-width:650px)` code. Select the section of code and copy it. Move back down to the bottom of your *screen.css* document, add a comment line and paste in the code. It should look like the section of code below.

```
/* Changes to personbio section */  
  
div.persontext {  
    position: absolute;  
    margin-top: 170px;  
    left: 5%;  
    width: 90%;  
    text-align: center; }
```

Instructions

Now modify the `margin-top` property so the code reads `margin-top: 60px;`

Explanation

Save the *screen.css* document and refresh the web page.

The `div.persontext` text box should now have moved up the screen back into position.

You should have noticed by now that for this media query for devices up to 320px wide we have been copying the original code from above, pasting it into position and modifying it to make the changes we need. Whereas for the previous media query where we were targeting devices up to 414px wide, we were simply adding just the

lines of code required to make the changes we needed. Either method is a valid way of working and there are pro's and con's for both. Copying and pasting the code is easier if you don't fully understand the code, but you will end up with copied code that doesn't require a change and is not actually needed, therefore taking up space. Just entering the code you need to make the changes is sometimes faster if you fully understand the code. You won't end up with lots of spare code you didn't really need and it keeps things neater, however reading through the changes you have made to the code may be confusing as you are not seeing all the code in context. You need to keep in mind the previous code from above which is also effecting the styling as a whole.

Adjust the font size for the Personbio section

We now need to make a couple of minor changes to the font size within the *persontext div* inside the *personbio section*.

Instructions

In Notepad++ on our *screen.css* document scroll down to the bottom of the page. Find your last section of code which started `div.persontext {`,. Underneath this code add the code shown below. As before enter the following code a portion at a time, saving and refreshing as you go.

The Code

```
div.persontext h2 { font-size: 1.6em; }  
div.persontext p { font-size: 0.9em; }
```

Explanation

Save the *screen.css* document and refresh the web page.

The `h2` and `p` font sizes within the `div.persontext` text box should now be a bit smaller. We adjusted the `h2` font size from `1.8em` to `1.6em` and the `p` font size from `1.0em` to `0.9em`.

Adjust font size in the Institutional Information section

The text at the bottom of the Institutional Information section is running down into the *footer* section. We will adjust the size of the .svg graphics and fonts in the Institutional Information section to combat this.

Instructions

In Notepad++ on our *screen.css* document scroll down to the bottom of the page. Find your last section of code which restyled the *personbio* section. Underneath that as before enter the following code below.

The Code

```
/* Changes to Institutionalinfo section */  
section.institutionalinfo h2 { font-size: 1.6em; }  
section.institutionalinfo aside h4 a { font-size: 1em; }
```

Explanation

Save the *screen.css* document and refresh the web page.

The **h2** and **h4** font sizes within the `section.institutionalinfo` text box should now be a bit smaller. We targeted the **h2** font inside the *institutionalinfo* section and adjusted the font size from **1.8em** to **1.6em**. We then targeted the **h4** anchor link font inside the *aside* which in turn is inside the *institutionalinfo* section, hence the fairly long `section.institutionalinfo aside h4 a` property, and changed the font size from **1.1em** down to **1em**. Up until this point the **h4** font has been taking its attributes from the default settings at the very top of the screen. This has worked well up until now, at which point we have had to adjust the font size for the smallest screens.

Adjust .svg graphic size in the Institutional Information section

Now we are going to adjust the size of the three .svg graphics inside the *institutionalinfo* section.

Instructions

In Notepad++ on our *screen.css* document scroll up to the section of code for large screens and find the three lines of code which start with:

```
section.institutionalinfo aside .content.trainingservices
section.institutionalinfo aside .content.uis
section.institutionalinfo aside .content.uniofcam
```

Select the three lines of code and copy them as shown below.



Scroll back down to the bottom of the *screen.css* document and underneath the previous code of the **h2** and **h4** font changes, paste in the code you have just copied.

Edit the code to change the background size of each graphic as follows:

- .trainingservices** *mortarboard.svg* change from 90px to 80px in size
- .uis** *uis.svg* change from 95px to 85px in size
- .uniofcam** *camshield.svg* change from 80px to 70px in size

Explanation

Save the *screen.css* document and refresh the web page.
The three *.svg* graphics should have reduced in size a bit.

Adjust *.svg* graphic padding in the Institutional Information section

The last thing we are going to adjust within the *institutionalinfo* section is the padding between the three *.svg* graphics.

The Code

```
section.institutionalinfo aside .content {
    padding-top: 90px; }
```

Save the *screen.css* document and refresh the web page.
The gap between the top of the graphics and the text above should have been reduced. This just tightens up the content within this section for smaller screens.

Adjust the font size in the Footer section

The text in the *footer* section at the bottom of the page is now the same size as the paragraph font in the *Institutional Information* section. We are going to reduce the font size in the *footer* section to match the styling we had for the larger screens.

Instructions

In Notepad++ on our *screen.css* document scroll down to the bottom of the page. Underneath your previous section of code add a new comment with the code shown below.

The Code

```
/* Change the footer section, decrease font size */  
footer p { font-size: 0.7em;}
```

Explanation

Save the *screen.css* document and refresh the web page.
The paragraph font inside the *footer* section should have decreased in size.

Adjust the font size in the Curriculum Vitae section

We just need to resize the fonts used in the *cv* section now so they match the size of those used in the other sections. Enter the code below underneath the previous changes you made to the *footer* section.

The Code

```
/* Changes to Curriculum Vitae page */  
  
div.cvtext h2 { font-size: 1.6em; }  
div.cvtext p { font-size: 0.9em; }
```

Explanation

Save the *screen.css* document and refresh the web page.

The font size for the 'Curriculum Vitae' heading and the paragraph text underneath the horizontal rule should have decreased in size. Nearly there, now we just have to match up the font sizes on our other two pages and we are finished.

Adjust the font size in the My Interests page

The table margins are currently unequal. We can correct this by reducing the font sizes used in the table.

Instructions

In Notepad++ on your *screen.css* document scroll down to the bottom of the page. Underneath the changes you made to your Curriculum Vitae page make a new comment and enter the code as shown below.

The Code

```
/* Changes to My Interests page */  
  
div.intereststable h2 { font-size: 1.6em; }  
  
table { font-size: 1em;}  
table th { font-size: 1.1em; }
```

Explanation

Save the *screen.css* document and refresh the web page.

The font size for the 'My Interests' heading should have reduced in size. This will now

match the font size for our other page headings. The table margins should now be equal as reducing the font size used in the table will have decreased the cell width and therefore the table as a whole.

Adjust the font size in the Contact Information page

The last styling change we need to make is to the *contacts.html* page. We just need to resize the font used for the 'Contact Information' heading to match the size of the other page headings and also adjust the h3 font size used for the address itself.

Instructions

In Notepad++ on our *screen.css* document scroll down to the bottom of the page. Find your last section of code which adjusted the styling for the My Interests page. As before enter the following code below, then save and refresh to see the change.

The Code

```
/* Changes to Contact Details page */  
  
div.contacttext h2 { font-size: 1.6em; }  
div.contacttext h3 { font-size: 1.2em; }
```

Explanation

Save the *screen.css* document and refresh the web page.

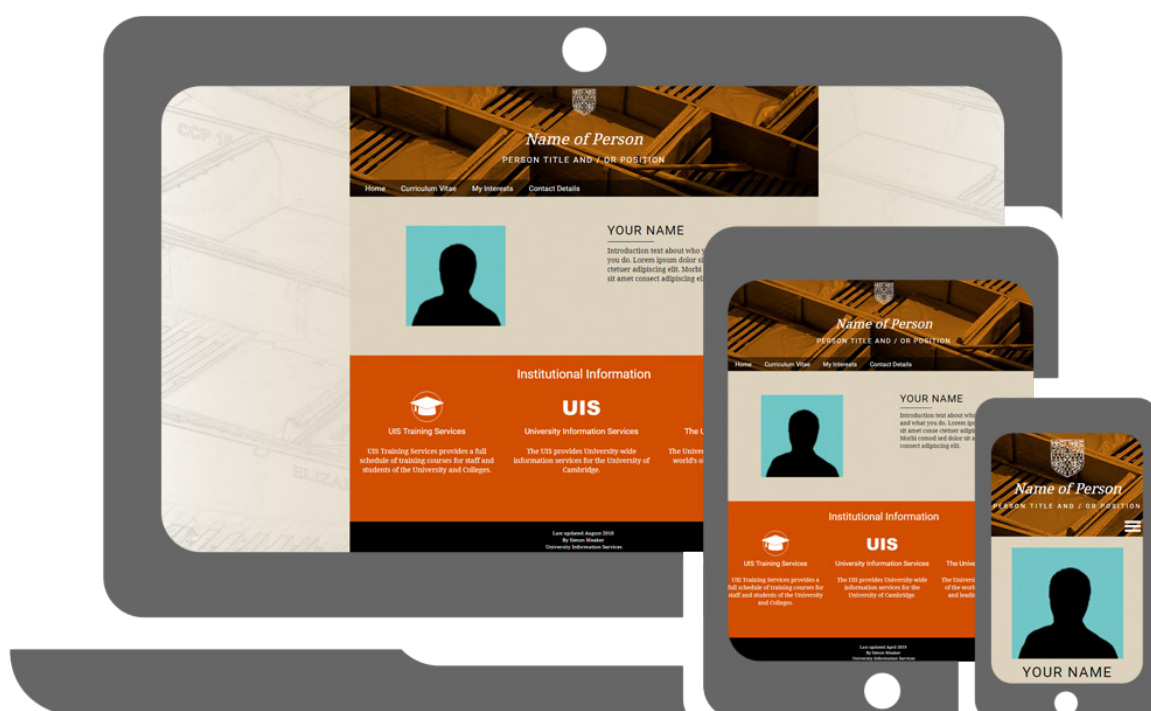
The **h2** and **h3** font sizes within the **contacttext div** text box should now be a bit smaller. We targeted the **h2** font inside the *contact* section and adjusted the font size from **1.8em** to **1.6em** to match the other pages. We then targeted the **h3** font which is used for the address. Up until this point the **h3** font has been taking it's attributes from the default settings at the very top of the screen. This has worked well so far but now we need to adjust the font size for the smallest screens to keep things in proportion.

Conclusion

Well done, you have finished! I think you will agree that was quite a lot of work. Sadly though there are not many quick fixes for making a web site responsive and we have only scratched the surface. Hopefully though the concepts you have learned during this course will give you the confidence to create your own responsive website and to put into practice the skills you have learned. Try adapting the website you have created during this course, change the colour scheme for certain screen sizes, or add in break points for the screen sizes we didn't cover in this course and personalise it to your taste. A huge amount of satisfaction can be had from being able to build your own web site. There is plenty of HTML and CSS still to learn and I would urge you to do so. <https://www.w3schools.com>, <https://htmldog.com> and <https://www.linkedin.com/learning> are all great resources and while you are at it why not learn more about JavaScript.

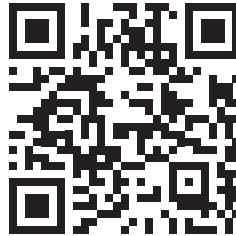
Upload your website and view using your phone and PC

If you wish to view your finished web site on your phone and PC in front of you then using **File Explorer** (right click on Windows 10 start button to find this) navigate to the **U:** folder and then into **public_html**. Inside that you will find an **index file** and a **folder called Style**. Copy these two out onto the desktop so they can be put back later. Copy all of your website folders and files into the public_html folder. Then using your PC or phone navigate to <http://people.ds.cam.ac.uk/x95?> ? being the number of the computer you are sitting at. Ask the trainer if you are unsure.



Please complete the course feedback before you leave

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



Thank you

Notes
