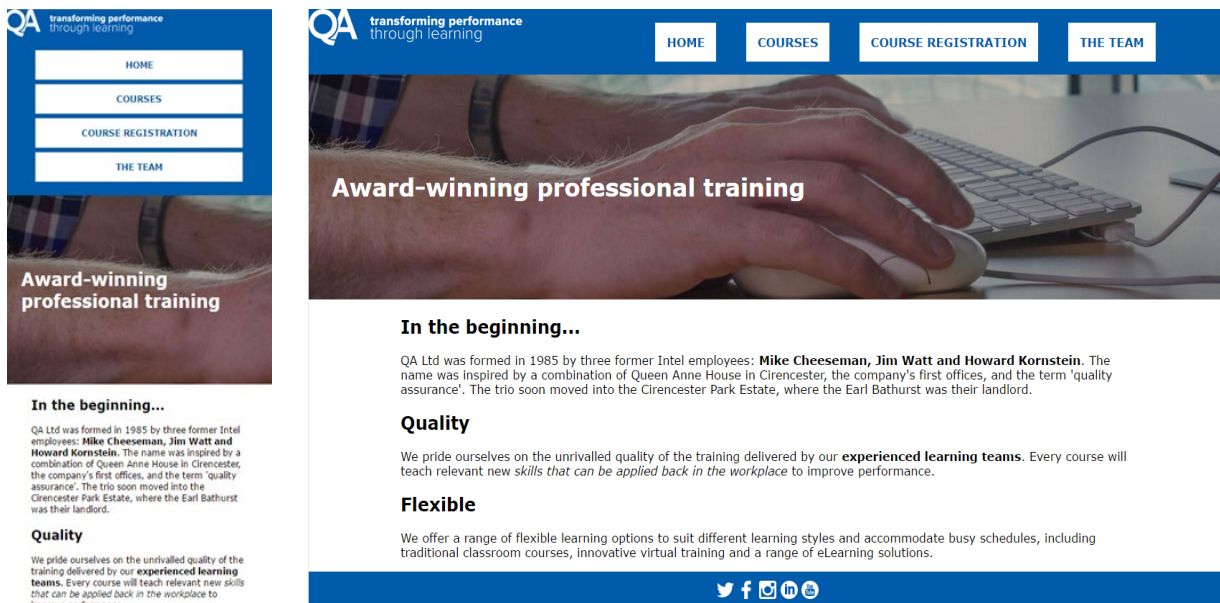


Exercise 10 – Mobile First Design

Objective

In the first part of this exercise, you will create a navigation menu which is optimised for mobile devices. Then in part two of the exercise, you will introduce breakpoints through the use of media queries which will turn a static website into a responsive one.



Exercise Instructions

Part 1 – Mobile First Navigation

Open index.html from the starter folder. We have a fairly simple page, which is a good starting point for our mobile first design – a logical structure consisting of three key areas:

- A header: consisting of our navigation, heading and logo
- Main area: consisting of the main content of this page
- A footer: consisting of social media icons

In this section of the exercise, we are going to make some changes to this page to ensure that it is optimised for mobile use.

1. Open index.html in Chrome and hit F12. There you will find an icon that looks not dissimilar to the below.



2. When you click this, Chrome will render the page in the device you select from the emulation. At the moment, the site likely looks like a zoomed out version of the desktop view.
3. Add the following line of code to the head of the index page:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

This ensures that the viewport is at parity with the device width – refresh the browser and see the difference. It's not great but it's a start.

4. We want to make our menu mobile friendly, so let's start by fixing the structure. At the moment, the menu and logo are fixed – this takes up a lot of real estate on a mobile device, so let's undo that. Find the styles that are applied to those two elements and remove the fixed positioning.
5. Refresh the page in the mobile view in chrome and see how it's already looking better!
6. Time for a little clean-up. Remove the float and z-index from the logo and move the background-color from the nav to the header.
7. Our text links are fine but buttons would provide a better user experience. The following rules applied to the link styles will sort that out:

color	#005BAB
display	block
background-color	white
padding	1em
font-weight	bold

8. Remove the padding from the nav element and set its left and right margins to auto, to centre it.
9. If you place the meta tag from step 3 into the other pages, you will be able to enjoy the fruits of your labour across the whole site!

Part 2 – Media Queries

Our mobile first navigation is all well and good, except it looks dreadful on a desktop machine. Let's use media queries to optimise the view on different screen sizes. We're going to take our mobile first design and apply new style rules for each larger size we wish to consider.

1. Let's create a new file in our CSS folder called `mediaQueries.css` – bear in mind that in deployment it is best practice to combine our CSS into as few a number of files as possible. Reference the new file in `index.html`.
2. Create a media query for our first 'break point' in the new stylesheet for any device that has a width over 768px.
3. Select the list items in our navigation element and change their display property to `inline-block` and their width to 45% this will bring them into two columns rather than one, making the most of our added width.
4. Test out your new media query by resizing the browser window and observe the layout change when you hit the break point.
5. Link our new stylesheet to the other pages of the website and confirm it works as expected across each.
6. Create another media query to represent a break point for even larger screens. Those of at least 1026px.
7. Within this media query, we need to do a number of things:
 - i) Set the width of the list items within the navigation element to be `auto`
 - ii) Set the padding of the list in the navigation element to be 0
 - iii) Add a float to the right to the navigation element and restrict its width to 70%
 - iv) Set the image within the header element to display `inline-block` and restrict its maximum width to 30%
8. Load the page in the browser. You may like the slightly overlaid nature of the navigation buttons caused by the floating of the navigation element, but we wish them to be within the blue bar and not over the image. How can we achieve that? Ask your instructor if you are unsure.
9. In our original `style.css` stylesheet, you will find the following code:

```
main > article.grid {  
  background-color: #f1f1f1;  
  float: left;  
  margin: 1%;  
  width: 48%;  
  height: 300px;  
  overflow-y: scroll;  
}
```

This code creates the 2x2 grid on the courses page. This is ok on a large screen but not so on a small, so move it into the largest of the break points we created.

10. View the website and again, observe the layout changes when you adjust the width of the courses page.