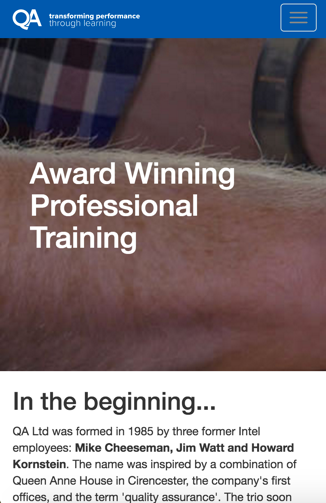
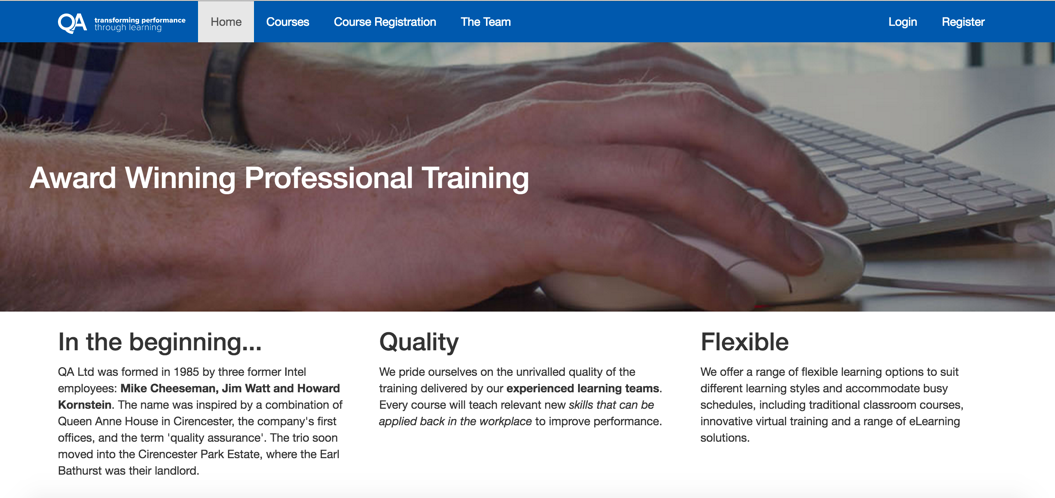
Exercise 16 – Introduction to Bootstrap

Objective

In this exercise we are going to use Bootstrap to build a responsive website.



Exercise Instructions

In the starter folder for this exercise you will find the files as they are downloaded from the Bootstrap website (plus an images folder, a starter index.html and a content.txt we’ve added for you to use during this exercise).

1. Open up the Bootstrap.css file and have a look around, this is the bones of Bootstrap. Pay attention to the groups of selectors within their appropriate media queries which define the width of the columns in percentage terms along with a left float. This is the core of many grid systems.
2. Open up index.html file and remove the h1 element. This leaves us to build our page the way we want.
3. Let’s start at the top: add a nav and give it the classes “navbar” and “navbar-default”.
4. Within this nav create a div with a class of navbar-header. Within this div place an anchor with a class of “navbar-brand” and then place the QA logo within the anchor. Give the QA logo a height property of 25.
5. Within the nav create 2 additional children, unordered lists with the classes “nav” and “navbar-nav” and give the 2nd one a class of “navbar-right”.
6. In the first unordered list add 4 list items containing links (as detailed below).

|  |  |
| --- | --- |
| Text | Link |
| Home | index.html |
| Courses | courses.html |
| Course Registration | registration.html |
| The Team | team.html |

1. In the second list create similar links for “Login” and “Register” – point these at #.
2. View the page in a browser and see how it responds to different screen sizes.
3. Within the navbar-header create a button and give it classes of “navbar-toggle” and “collapsed”, a data-toggle attribute with a value of “collapse” and a data-target attribute with a value of “#collapsableNav”
4. Wrap the 2 unordered lists you created within the nav in a div with an id that matches the data-target attribute from the previous step (collapsableNav) and give it the classes “collapse” and “navbar-collapse”
5. Test your page in a browser, you should find a burger icon which now shows the vertical menu on small screens, but on medium and above screens a horizontal menu is displayed.
6. We’re currently missing the heading for our page so let’s create that. Below the nav element add an h1 element with the text “Award Winning Professional Training”
7. It’s time to add our own style to the page. We’ll use the empty app.css for this purpose – open it now.
8. Create an appropriate selector to override default styles applied to the navbar we created and set its bottom margin to 0.
9. Add a h1 selector, set its color to white, its margin to 0 and add a padding of 4em to the top and bottom and 1em to the left and right.
10. Create a new class called keyboard and set its background-image to the keyboard.jpg file found in the images folder and set the background-size to cover.
11. Add the keyboard class to the h1 element in index.html and view the page in your browser.
12. We need to customise the built in Bootstrap styles to improve this further, set the background colour of the top-bar to #005BAB and set the border and border-radius to 0.
13. Create the following selector “.navbar .navbar-nav li > a” and set the color to white.
14. View the page in your browser.
15. We now need to create the content area of the website. Below the h1 create a div with a class of “container” and insert a main element with a class of ‘row’
16. We’re going to have three sections on this page which on a mobile will be one after the other in a vertical arrangement, so let’s do that first. Create three article elements with a class of “col-xs-12” this neatly defines that each of these articles will take up all 12 columns of the page, and being mobile first – this echoes through the larger grids also.
17. There’s some content in a file called content.txt which you can use to copy and paste into the three articles. Just grab one h2 and paragraph pair for each article.
18. For larger screens we might want the layout to make use of the available width. Add appropriate classes so that on the small breakpoint the first two articles take up half of the width each.
19. Finally add appropriate classes so that for the mediu breakpoint all three articles take up a third of the width each.