# Specification

The specification page is where the technical details of the product are held. As seen in the designs, the page has a heading, then three sub headings and then the text/image content underneath that.

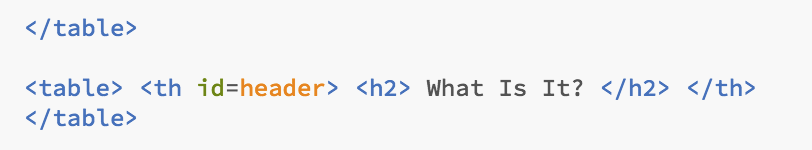
For the heading, I used ‘<table>’ so that it could go underneath the navigation bar. I created a new ID for this header in CSS as it was different to other elements on the page, this I declared in <th> (the header for the table).

For the sub headings, I also followed through with a table, however as this background was not grey, I had to create another ID for it within CSS. As described in the design, these buttons would jump to the necessary pages for that information. However, to make it more practical, I thought it would be better if the buttons would jump to part of one page that the information was on. This would then reduce the clicks the user needs to make to get to the information they want. Thus, these buttons are directed to a place on the ‘Specification’ rather than another page. As the information on the page is not the information that would be on there if the website went live, I was unable to implement this feature into the current product.

Underneath these buttons is the text. I placed these elements once again into a table as I had trouble using a grid. In the design the image is to the side and the text next to it. However as I had trouble adjusting the image size within the grid, the image is currently placed underneath the text. Realistically this would have been fixed.

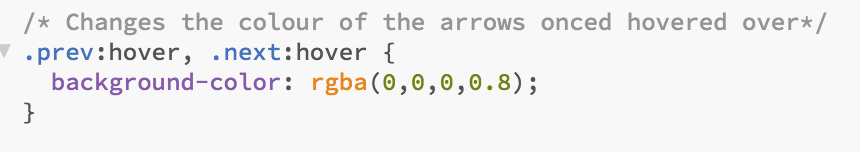
# What Is It?

Similarly to Specification, this page also has a header. As I have used the same header from the page before, I used the same CSS ID from Specification.



The feature on this page, if the slideshow. This element was created using HTML, CSS and JavaScript for the functionality. I started off by creating a class in CSS called ‘Slideshow’ to determine the position and size of the slideshow.

The slideshow consisted of many components; the container, the slides, the fade, the arrows and the dots.

For each component, there is a different CSS ID as they all do different things. In the CSS is where I also included changing of colours when the pointer is hovered over elements.

I decided to use JavaScript for the slideshow so that it could animate and change the slides. It also allowed the slide image to automatically go back to the first image when the last ‘next’ button is clicked.

Within the HTML code, is calls the JavaScript function ‘plus slides’ so that when the arrow is clicked it will go to the appropriate slide.

The for the dots the function ‘current slide’ is called so that depending on what ‘dot’ is clicked, the slide displayed will mirror that.