SIT120 Pass Task 1.2P

GitHub repository link:

https://github.com/mattshep36/SIT120-Task-1.2P

Video link:

https://youtu.be/rZIn2EPbDpc

Responsive Web Design

What is it and why is it important?:

When it comes to web development, it's important to ensure that the content of a web page is displayed in a format that is readable and pleasing to the eye regardless of the device that is used to interface with the page.

This is essentially the principle of responsive web design, 'responsive' being the key word. For a website to be considered 'responsive', elements of the site must resize, shift location, enlarge, shrink, or become hidden to suit differing viewport aspect ratios and screen sizes. 'Break points' can be defined which govern at which point content will transform or move to accommodate a changing viewport resolution.

If responsive design strategies are not adopted when creating web pages, the resulting page may become unreadable, unsightly, and difficult / impossible to navigate. Applying responsive web design principles during the web development process will ensure the user experience is of a consistent quality.

How to implement responsive web design with HTML and CSS:

Organising HTML element into various 'divs' that can be nested within each other can assist greatly in improving a page's responsiveness. CSS can be applied to these 'div' HTML elements to make pages behave dynamically.

There are many CSS properties generally associated with responsive web design. For example, minimum and maximum unit values (e.g., max-width, min-height) can be used to prevent a web page's content from overflowing its container or appearing too small.

Additionally, the use of relative units when dictating sizing can ensure that a page scales appropriately and proportionally as resolution increases / decreases thus maintaining a consistent appearance. Examples of relative units include 'em' and 'rem', used in responsive typography, as well as 'vw' and 'vh' which are used to size HTML elements relative to the viewport they're displayed within.

Responsive layout techniques can also be implemented to improve a site's responsiveness, for example, the parameter 'column-count' will split content into a specified number of columns that will adapt to the screen size of the viewing device. As well as this, CSS grid and flexbox layouts can be utilized that allow the position, size, and proportion of page elements to be structured and organised more dynamically relative to one another.

Reflection:

This task provided an opportunity to explore using JavaScript for the first time. I've utilized many JS methods and implemented them into functions to add interactivity to the HTML page I created for this task.

I've also made use of a fair amount of CSS as I worked on this task. I divided various elements of the page into nested 'divs' to apply CSS to multiple elements at a time within defined containers. I've also made use of responsive web design principles in my code, these include the use if the 'flex' layout in the main body section of my web page, as well as CSS properties such as 'min-width' coupled with relative values for the sake of scalability.

- Matthew Sheppard (Student ID 215162718)

References

1. Interaction Design Foundation. (n.d.). Responsive Design: Let the Device Do the Work. Available at: https://www.interaction-design.org/literature/article/responsive-design-let-the-device-do-the-work (Accessed 27 August 2023).
2. w3schools.com. (n.d.). HTML Responsive Web Design. Available at: https://www.w3schools.com/html/html_responsive.asp (Accessed 27 Aug 2023).
3. Mozilla Developer Network. (n.d.). CSS Flexbox. Available at: https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox (Accessed 28 Aug 2023).