

Theory

HTML (HyperText Markup Language) is the standard markup language used to create the structure of a webpage. It defines elements such as headings, paragraphs, images, links, forms, and sections of a page. HTML provides the skeleton of a webpage, giving content its structure, but it does not control the visual style or layout. Most HTML elements use opening and closing tags, and it can be written in inline, internal, or external form.

CSS (Cascading Style Sheets) is used to style and design HTML elements, controlling colors, fonts, spacing, layout, and overall appearance. CSS separates content from design, making webpages visually attractive and easier to maintain. It can be applied inline, internally within a `<style>` tag, or externally via a separate `.css` file, which is the most common approach.

Flexbox is a one-dimensional layout system in CSS that arranges elements in a row or column and provides flexible alignment, spacing, and distribution of items. It is ideal for simpler layouts like navigation bars, buttons, or cards in a row. Important properties of Flexbox include `flex-direction`, `justify-content`, `align-items`, and `flex-wrap`, which allow precise control over the alignment and spacing of elements.

CSS Grid is a two-dimensional layout system that allows placement of elements into rows and columns simultaneously, making it ideal for creating complex webpage layouts such as dashboards or portfolio websites. Key properties include `grid-template-rows`, `grid-template-columns`, `grid-row`, `grid-column`, and `gap`, which control the placement and spacing of grid items. Grid provides more control for page-wide layouts compared to Flexbox, which is primarily for one-dimensional alignment.

Both Flexbox and Grid can be combined with media queries to make webpages responsive, meaning they adapt to different screen sizes and devices. Media queries allow designers to apply different CSS rules based on conditions like screen width, height, or orientation. This ensures that the webpage is usable and visually appealing on desktops, tablets, and mobile devices, providing a better user experience across all devices.

Discussion and Conclusion

In this lab, we created a simple portfolio webpage using HTML and CSS. The layout was designed using CSS Grid to organize sections such as the header, about, skills, photo, and contact areas. We applied media queries to make the webpage responsive, allowing the layout to adapt for smaller screens by stacking the sections vertically. This demonstrated how CSS Grid can efficiently manage complex layouts, and how media queries ensure usability across different devices.

The lab helped in understanding the structure and styling of webpages using HTML and CSS. Using Grid, we were able to create a clear and organized layout, and with media queries, the webpage became responsive. This exercise illustrated the importance of combining modern CSS techniques to create visually appealing and flexible web designs that work well on both desktop and mobile devices.