Recap: Styling elements

Overview

In this reading, you will revisit some of the various different CSS concepts that will assist you with styling elements in your Little Lemon web app project.

Default CSS styling

Recall that CSS is a design language that allows you to affect the way that your HTML elements are displayed in the browser. Even if you don't have any CSS styles declared in your web page (or web app), the browser will still display the HTML. This means that the browser comes with some pre-built styling. In other words, the browser has some preconceptions of how every single element on the page should look. For example, there are block elements, such as `<h1>` to `<h6>`, `<p>` and `<div>`, which, by default, come with the `display: block` property set. In fact, the browser comes with pre-built styles for every single HTML tag. No matter how minimal an HTML element looks on the page, it's sure to have some CSS styling applied. Likewise, it would not be possible to show an element on the page if it did not have some CSS styling applied.

Practical CSS styling

You can think of your own CSS code as code that overrides the default, built-in styles. For example, there is not a single element in HTML that a browser styles as `display: grid` by default. This allows you, as discussed previously, to switch on the CSS grid as needed. Perhaps that's all that it takes. You get to keep all the browser's styles for all the HTML elements, with the default display property overridden to be grid.

This brings us to the topic of **CSS selectors**. You don't have to use CSS element styles like the following:

```css
div {
    display: grid;
}

```

Adding the above code snippet to your CSS styles would likely be bad because you'd be setting every single `<div>` element on the page to display as `grid`. Luckily, CSS selectors are very versatile, so by virtue of simply adding a single CSS class to a `<div>` tag in your style sheet, you can target only those `<div>` tags that have a given class in your HTML, and thus your CSS selector becomes a much more fine-tuned tool, with pin-point precision:

```css
div.main-grid {
    display: grid;
    /* ... your styles ... */
}
```

Of course, to make it work, you'd need to update your HTML as well:

```html
<div class="main-grid">
    <!-- Your content -->
</div>
```

Conclusion

In this reading, you revisited the basics of styling elements using CSS. To explore the topics covered in this lesson in greater detail, please visit the following lesson items from the **Introduction to Front-end development** course:

- Selecting and styling
- Alignment basics
- Box model introduction

By completing this overview, you're ready to tackle your next exercise, which involves styling elements.