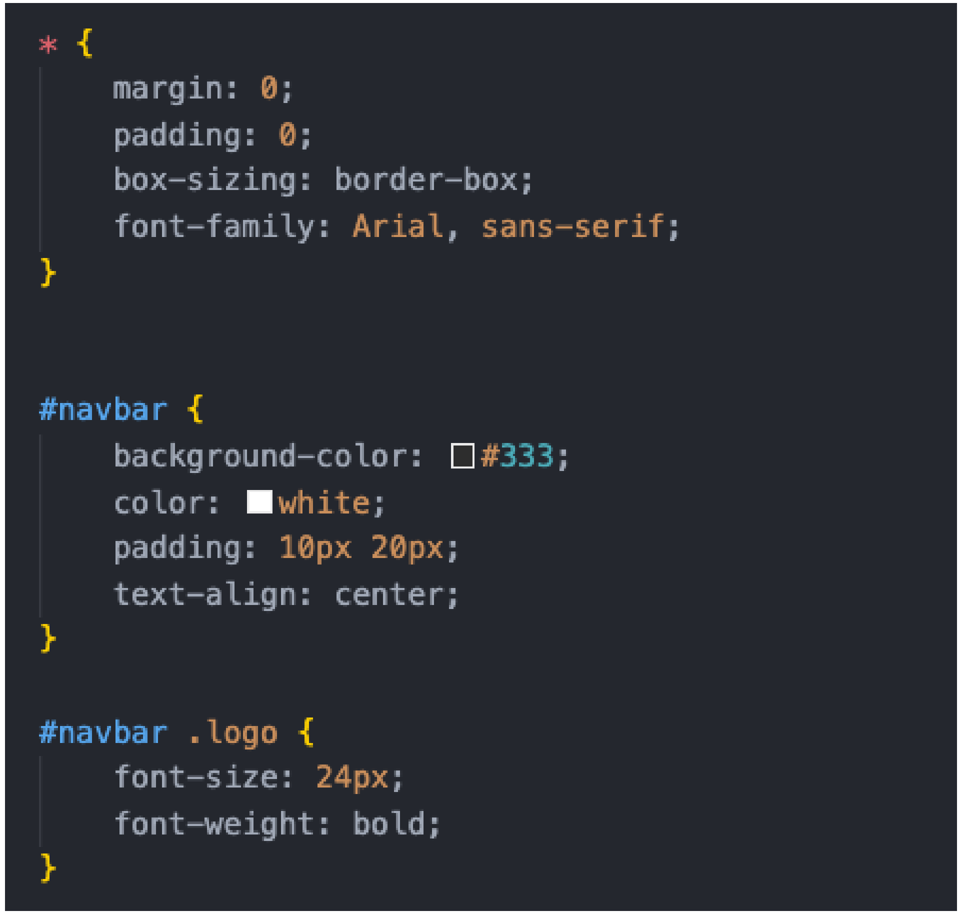
INTRODUCING CSS

FULL STACK SKILLS BOOTCAMP

INTRODUCING CSS

- **Lesson Overview:**
- In this lesson, we will be introduced to CSS (Cascading Style Sheets) and learn how to style HTML elements using different methods and techniques. The lesson will cover inline and external CSS, different CSS selectors, how to define colors and fonts, and an introduction to display modes. By the end of the lesson, students will understand how to apply CSS to HTML to enhance the look and feel of web pages.



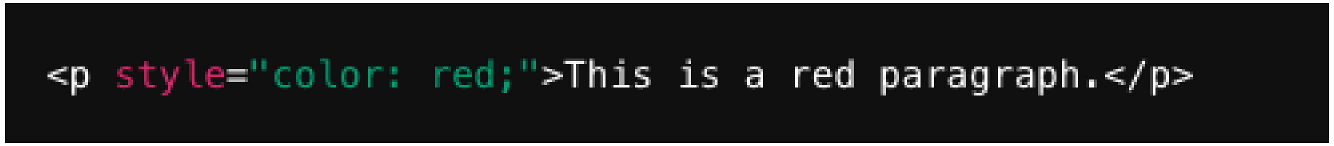
```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: Arial, sans-serif;
}


#navbar {
    background-color: □#333;
    color: ■white;
    padding: 10px 20px;
    text-align: center;
}

#navbar .logo {
    font-size: 24px;
    font-weight: bold;
}
```

INTRODUCTION TO CSS

- Definition: CSS is a language used to style the layout and presentation of HTML documents.
- Key Point: CSS allows you to separate content (HTML) from presentation (CSS).

INLINE CSS

- Inline CSS applies styles directly to HTML elements using the style attribute.

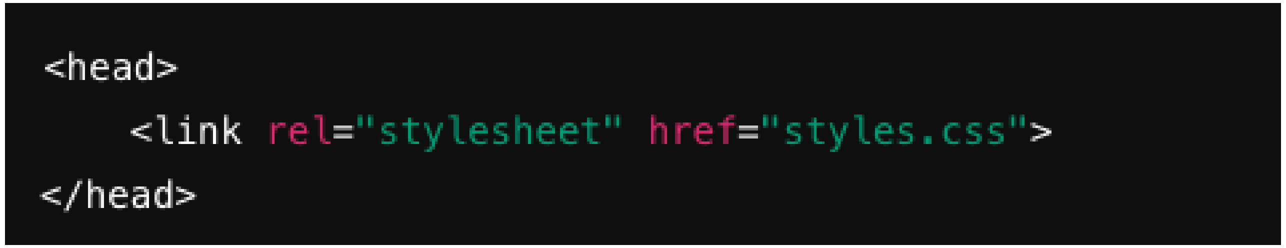


```
<p style="color: red;">This is a red paragraph.</p>
```

- Use Case: Quick, one-off changes.
- Drawback: Harder to manage styles for large projects.

EXTERNAL CSS

- External CSS uses a separate .css file, which is linked to the HTML document using the <link> element.



```
<head>
    <link rel="stylesheet" href="styles.css">
</head>
```

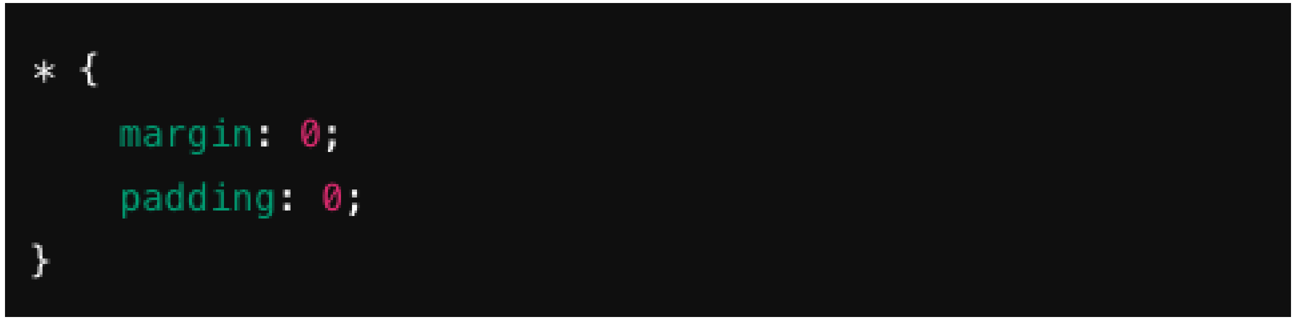
- CSS file (style.css)
- Use Case: Recommended for larger projects or when the same styles need to be applied across multiple pages.

CSS SELECTORS: UNIVERSAL, ELEMENT, CLASS, AND ID

- We will learn different types of CSS selectors and how to target elements effectively.

UNIVERSAL SELECTOR (*)

- Targets all elements on the page.

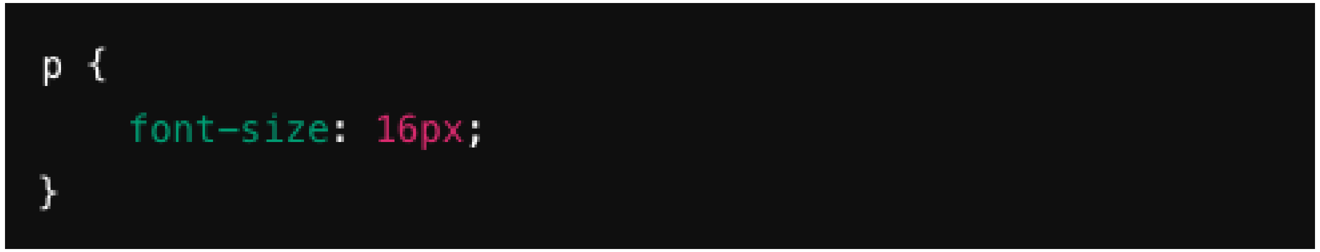


```
* {
    margin: 0;
    padding: 0;
}
```

- Use Case: Applying general styles across the entire page.

ELEMENT SELECTOR

- Targets specific HTML elements by their tag name.

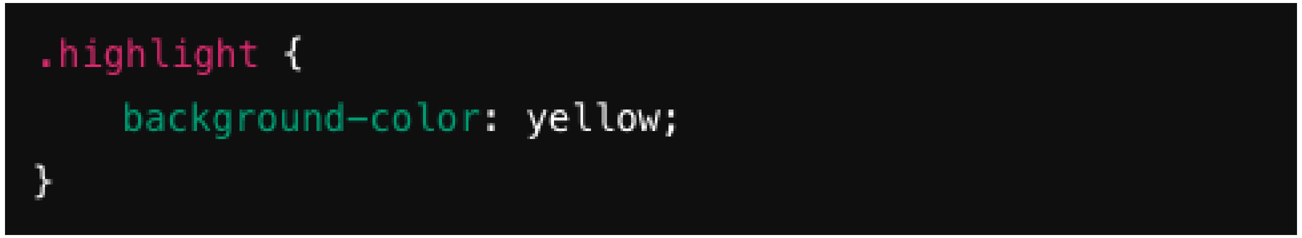


```
p {
    font-size: 16px;
}
```

- Use Case: Styling all instances of a particular HTML element.

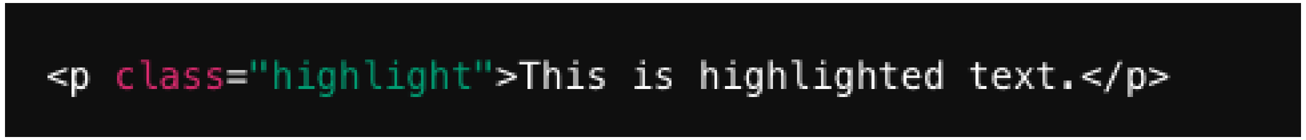
CLASS SELECTOR (.)

- Targets elements with a specific class attribute. Classes can be reused.



```
.highlight {
    background-color: yellow;
}
```

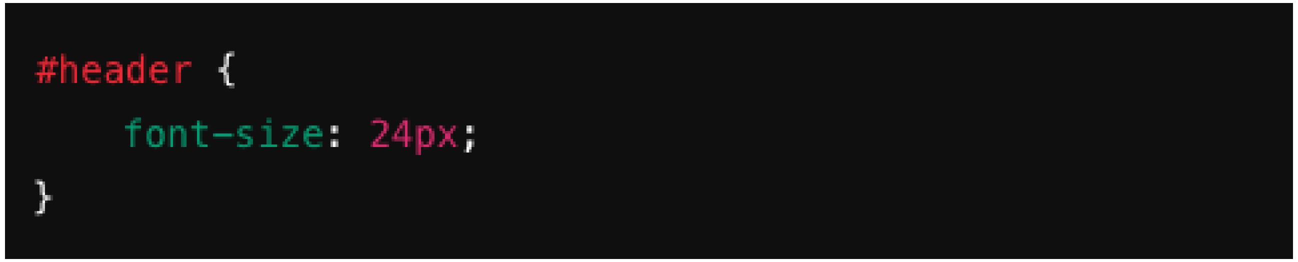
- HTML example:



```
<p class="highlight">This is highlighted text.</p>
```

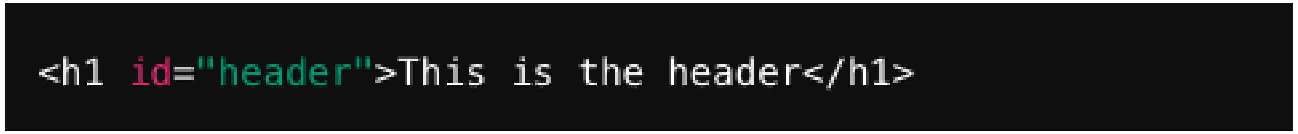
ID SELECTOR (#)

- Targets a single element by its unique ID. IDs should be used only once per page.



```
#header {
    font-size: 24px;
}
```

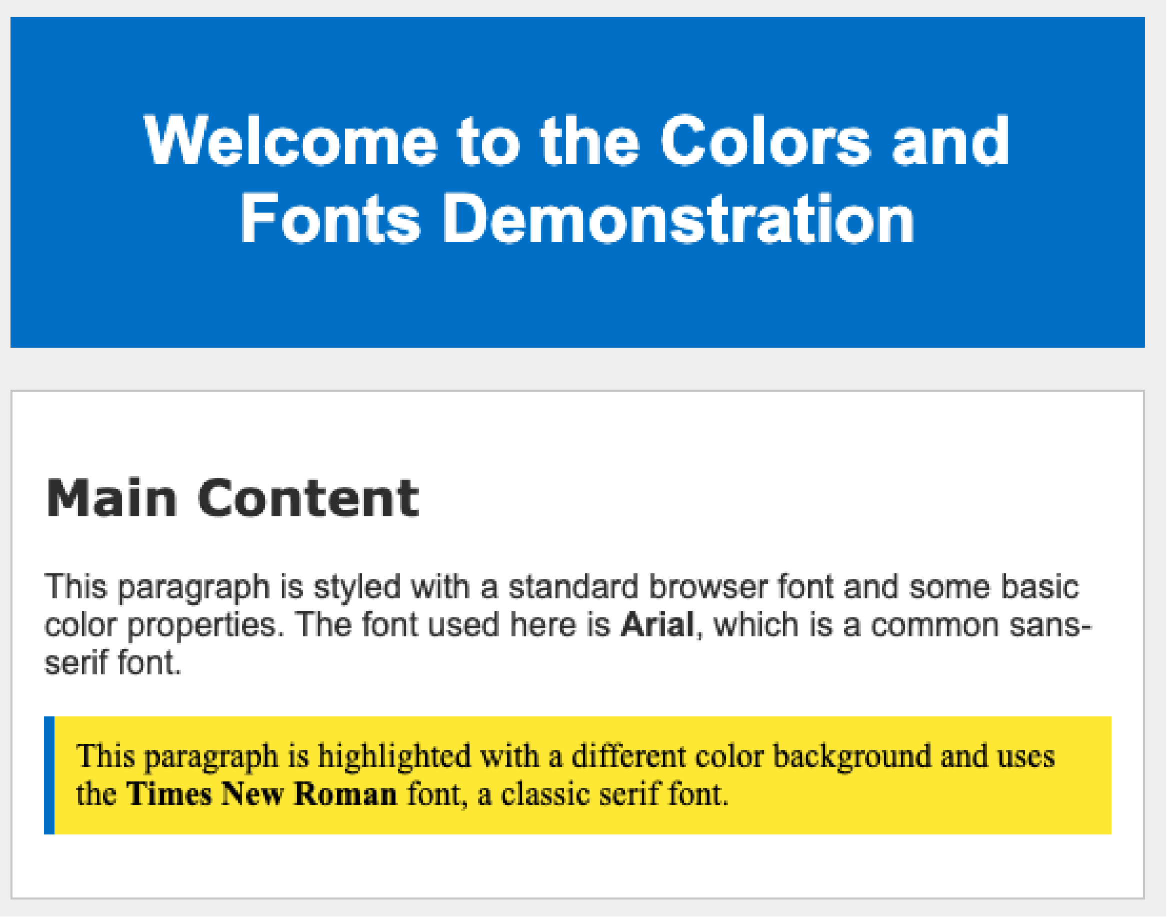
- HTML example:



```
<h1 id="header">This is the header</h1>
```

DEFINING COLORS AND STANDARD BROWSER FONTS

- We will learn how to define colors in CSS using various methods and how to apply standard browser fonts.



Welcome to the Colors and Fonts Demonstration

Main Content

This paragraph is styled with a standard browser font and some basic color properties. The font used here is **Arial**, which is a common sans-serif font.

This paragraph is highlighted with a different color background and uses the **Times New Roman** font, a classic serif font.

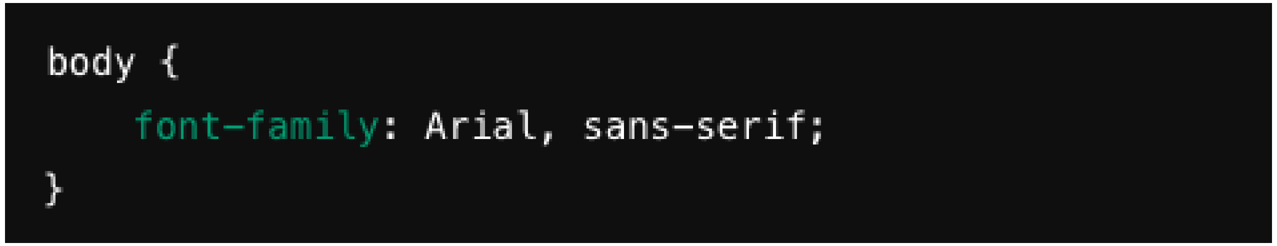
DEFINING COLORS IN CSS

- **Color by Name**: Predefined color names in CSS.
  Example: color: red;
- **Hexadecimal**: Specifies color using a hex code.
  Example: color: #ff0000;
- **RGB**: Specifies color using RGB values.
  Example: color: rgb(255, 0, 0);
- **HSL**: Specifies color using hue, saturation, and lightness.
  Example: color: hsl(0, 100%, 50%);

- Key Point: Colors can be defined in multiple ways, depending on the project needs.

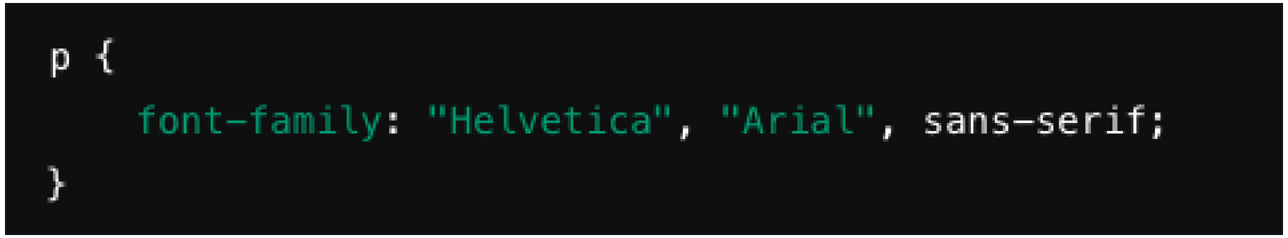
DEFINING FONTS

- **Standard Browser Fonts**:
  Common fonts available across most browsers: Arial, Times New Roman, Verdana, etc.
- **Applying fonts:**



```
body {
    font-family: Arial, sans-serif;
}
```

- **Fallback Fonts**: Providing multiple font options in case the primary font is unavailable.
  Example:



```
p {
    font-family: "Helvetica", "Arial", sans-serif;
}
```

DISPLAY MODES

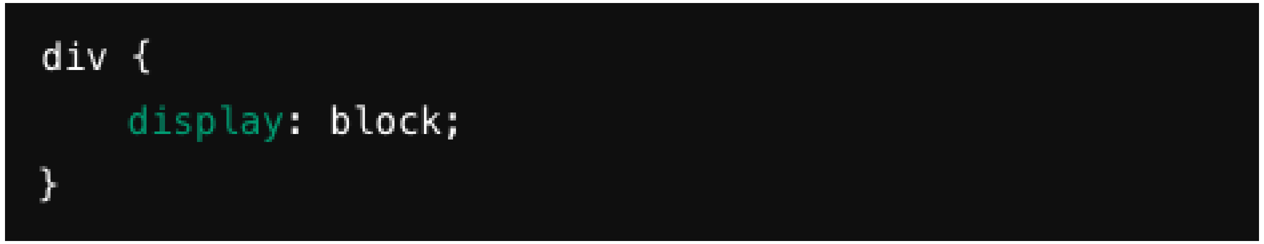
Learning Objective:

We will understand the different CSS display modes and how they control element layout on the page.

DISPLAY MODE: BLOCK

- Block-level elements take up the full width of the container and start on a new line.

  Examples of block elements: <div>, <p>, <h1>.

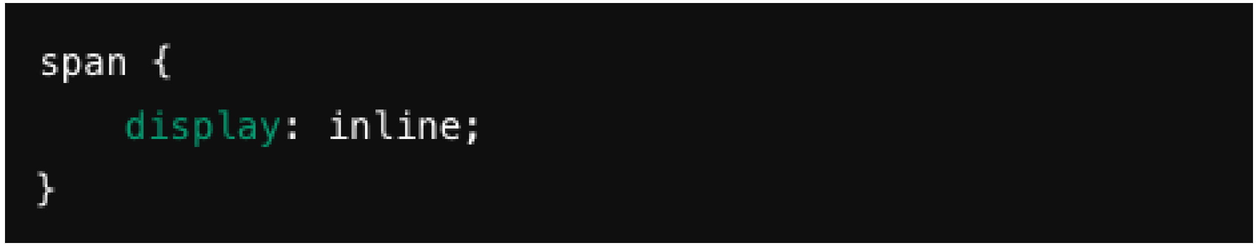


```
div {
    display: block;
}
```

DISPLAY MODE: INLINE

- Inline elements only take up as much width as necessary and do not force a new line.

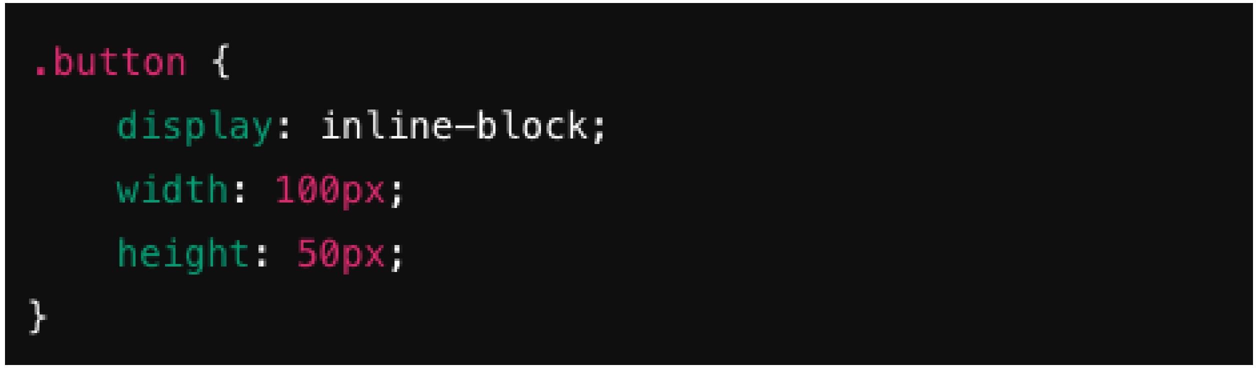
  Examples of inline elements: <span>, <a>, <img>.



```
span {
    display: inline;
}
```

DISPLAY MODE: INLINE-BLOCK

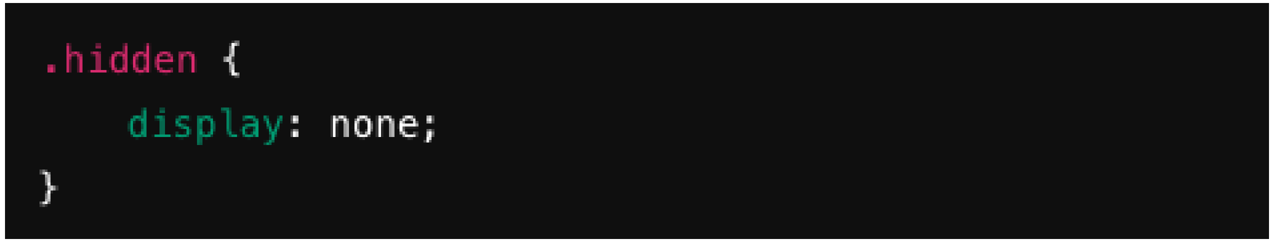
- Similar to inline elements, but can have a defined width and height, making them more flexible.



```css
.button {
    display: inline-block;
    width: 100px;
    height: 50px;
}
```

DISPLAY MODE: NONE

- Hides the element completely from the page (no space is taken).



```
.hidden {
    display: none;
}
```

RECAP

- Including CSS inline and through external files.
- Using various CSS selectors.
- Defining colors and fonts.
- Understanding CSS display modes.

- **W3Schools CSS Guide**: https://www.w3schools.com/css/

  MDN Web Docs: CSS: https://developer.mozilla.org/en-US/docs/Web/CSS

QUESTIONS?