Notes

CSS variables

CSS variables, also known as custom properties, allow you to store values that can be reused throughout a stylesheet. This makes managing and updating styles more efficient, especially for large projects.

Syntax

Declaring a Variable:

```
:root {
  --primary-color: #3498db;
  --font-size: 16px;
}
```

Using a Variable:

```
body {
  color: var(--primary-color);
  font-size: var(--font-size);
}
```

Scope

Global Scope:
Defined in the `:root` pseudo-class, making them available throughout the document.
Example:

```
:root {
  --main-bg-color: coral;
}

body {
  background-color: var(--main-bg-color);
}
```

Local Scope:
Defined within a specific selector, making them available only within that selector.
Example:

```css
.container {
  --container-padding: 20px;
}

.container {
  padding: var(--container-padding);
}
```

Fallback Values

You can provide a fallback value to be used if the variable is not defined.

```css
p {
  color: var(--text-color, black);
}
```

Dynamic Updating

Variables can be updated dynamically using JavaScript.

```javascript
document.documentElement.style.setProperty('--primary-color',
'#2ecc71');
```

Advantages

- Reusability: Write once, use multiple times.
- Maintainability: Easier to manage and update styles, especially for large projects.
- Theming: Simplifies the implementation of themes.

Use Cases

1. Theming: Create light and dark themes with ease.

```css
:root {
  --background-color: white;
  --text-color: black;
}
```

```css
[data-theme="dark"] {
  --background-color: black;
  --text-color: white;
}

body {
  background-color: var(--background-color);
  color: var(--text-color);
}
```

2. Consistent Design: Ensure uniform spacing, colors, and typography across the site.

```css
:root {
  --main-spacing: 16px;
  --main-font: 'Arial, sans-serif';
}

.header {
  padding: var(--main-spacing);
  font-family: var(--main-font);
}

.footer {
  padding: var(--main-spacing);
  font-family: var(--main-font);
}
```

CSS !important

The !important declaration in CSS is used to increase the specificity of a style rule, ensuring that it overrides any other declarations of the same property, regardless of specificity.

Syntax

Basic Syntax:

```css
selector {
  property: value !important;
}
```

Example:

```
.example {
  color: red !important;
}
```

Usage

1. Override Inline Styles:

HTML:

```
<p style="color: blue;">This text will be red due to the
!important rule.</p>
```

CSS:

```
p {
  color: red !important;
}
```

2. Override Styles in Specific Contexts:

```
.button {
  background-color: green;
}

.special-button {
  background-color: red !important;
}
```

Browser Engines

Browser engines are the core components of web browsers responsible for rendering web pages, executing JavaScript, and handling interactions. Different browsers use different engines, each with unique characteristics and optimizations.

Major Browser Engines

1. **WebKit**

Used By:

- Safari (Apple)
- Older versions of Chrome (Google) and other WebKit-based browsers.

Features:

- Known for speed and efficiency.
- Excellent performance on macOS and iOS devices.

Key Technologies:

- WebCore: The HTML and CSS rendering engine.
- JavaScriptCore: The JavaScript engine.

History:

- Originally developed by Apple based on the KHTML engine from the KDE project.

2. **Blink**

Used By:

- Chrome (Google)
- Edge (Microsoft)
- Opera
- Brave

Features:

- Fork of WebKit, with significant modifications.
- Focus on performance, security, and extensibility.

Key Technologies:

- V8: The high-performance JavaScript engine.

History:

- Created by Google in 2013 by forking WebKit to better meet its needs for Chrome.

3. **Gecko**

Used By:

- Firefox (Mozilla)
- Thunderbird

Features:

- Strong support for web standards.
- High flexibility and customizability.

Key Technologies:

- SpiderMonkey: The JavaScript engine.

History:

- Developed by Mozilla and has been the backbone of Firefox since its inception.

4. Quantum

Used By:

- Firefox (Mozilla)

Features:

- Successor to Gecko, part of the Quantum project aimed at improving performance and reducing memory usage.
- Utilizes Rust programming language for better safety and concurrency.

Key Technologies:

- Servo: A new browser engine being developed by Mozilla to eventually replace Gecko/Quantum.

History:

- Launched in 2017 as part of a major Firefox overhaul.

5. Trident

Used By:

- Internet Explorer (Microsoft)

Features:

- Proprietary engine with various compatibility quirks.
- Legacy support for older web technologies.

Key Technologies:

- Direct integration with Windows.

History:

- Dominated the browser market during the early 2000s, now largely obsolete.

6. EdgeHTML

Used By:

- Legacy versions of Edge (Microsoft)

Features:

- Fork of Trident with modern standards compliance.
- Focus on performance and interoperability.

Key Technologies:

- Improved rendering pipeline and Chakra JavaScript engine.

History:

- Replaced by the Chromium-based Edge using Blink engine.

7. Presto

Used By:

- Older versions of Opera (before switching to Blink)

Features:

- Highly customizable and lightweight.
- Rapid development cycle with frequent updates.

Key Technologies:

- Integrated with various proprietary technologies for enhanced performance.

History:

- Developed by Opera Software, now discontinued in favour of Blink.

WebKit Vendor Prefix

The -webkit- prefix is used to enable CSS properties that are specific to WebKit-based browsers (like Safari and older versions of Chrome). Below is a detailed list of various -webkit- prefixed properties and their uses.

WebKit Scrollbar Styling

Customizing scrollbars with -webkit- properties can enhance the user interface design.

1. ::-webkit-scrollbar

Targets the entire scrollbar.

Example:

```
::-webkit-scrollbar {
  width: 12px; /* Width of the vertical scrollbar */
  height: 12px; /* Height of the horizontal scrollbar */
}
```

2. ::-webkit-scrollbar-track

Targets the track (background) of the scrollbar.

Example:

```
::-webkit-scrollbar-track {
  background: #f1f1f1;
}
```

3. ::-webkit-scrollbar-thumb

Targets the draggable part of the scrollbar.

Example:

```
::-webkit-scrollbar-thumb {
  background: #888;
}
```

4. ::-webkit-scrollbar-thumb

Targets the scrollbar thumb when hovered.

Example:

```
::-webkit-scrollbar-thumb:hover {
  background: #555;
}
```

5. ::-webkit-scrollbar-button

Targets the buttons at the ends of the scrollbar (not commonly used).

Example:

```
::-webkit-scrollbar-button {
  display: none; /* Hides the scrollbar buttons */
}
```

6. ::-webkit-scrollbar-corner

Targets the bottom corner of the scrollbar, where horizontal and vertical scrollbars meet.

Example:

```
::-webkit-scrollbar-corner {
  background: #999;
}
```

WebKit Text Properties

1. -webkit-text-fill-color

Sets the fill color of text.

Example:

```
.text-fill {
  -webkit-text-fill-color: red;
}
```

2. -webkit-text-stroke

   Adds a stroke to the text.

   Example:

```
.text-stroke {
  -webkit-text-stroke: 1px black;
}
```

3. -webkit-text-size-adjust

   Controls the text size adjustment on mobile devices.

   Example:

```
body {
  -webkit-text-size-adjust: 100%; /* Prevents automatic text
resizing */
}
```

Other Vendor Prefixes

1. -moz-

Used By:

- Firefox

Example:

```
.box {
  -moz-transition: all 0.3s ease-in-out;
}
```

Features:

- Prefixed properties that Mozilla implemented in Firefox before they became standard.

2. -ms-

Used By:

- Internet Explorer
- Edge (legacy versions)

Example:

```
.box {
  -ms-transition: all 0.3s ease-in-out;
}
```

Features:

- Specific to Microsoft's implementations, including grid layout and transform properties.

3. -o-

Used By:

- Opera (before switching to Blink engine)

Example:

```
.box {
  -o-transition: all 0.3s ease-in-out;
}
```

Features:

- Opera's proprietary prefix before adopting WebKit/Blink.

Navbar with Hamburger Menu

Bootstrap 5 provides a responsive navbar component with a collapsible hamburger menu for smaller screens. Here's how to implement it:

Basic Structure:

- Use the navbar class for the main container.
- Use navbar-expand-{breakpoint} to specify when the navbar should expand (e.g., navbar-expand-lg for large screens).
- Use navbar-toggler and navbar-toggler-icon for the hamburger menu button.
- Use collapse and navbar-collapse for the collapsible part of the navbar.

Example:

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button"
data-bs-toggle="collapse" data-bs-target="#navbarNav"
aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item">
```

```
          <a class="nav-link active" aria-current="page"
href="#">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Features</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Pricing</a>
        </li>
        <li class="nav-item">
          <a class="nav-link disabled">Disabled</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

Alternative Approach: Dropdown Outside Navbar

HTML Structure:

- Add a dropdown menu outside the navbar and use JavaScript to toggle its visibility when clicking the hamburger button.
- Use Bootstrap's dropdown components to style the dropdown menu.

Example:

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button"
id="navbarToggler" aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
  </div>
</nav>
<div class="dropdown-menu" id="dropdownMenu"
aria-labelledby="navbarToggler">
  <a class="dropdown-item" href="#">Home</a>
  <a class="dropdown-item" href="#">Features</a>
  <a class="dropdown-item" href="#">Pricing</a>
  <a class="dropdown-item disabled" href="#">Disabled</a>
</div>
```

Carousel

Bootstrap 5 also provides a responsive and flexible carousel component for creating slideshows.

Basic Structure:

- Use the carousel class for the main container.
- Use carousel-inner for the slides container.
- Use carousel item for each slide.
- Use data-bs-ride="carousel" to enable automatic cycling.

Example:

```
<div id="carouselExample" class="carousel slide"
data-bs-ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      <img src="slide1.jpg" class="d-block w-100" alt="Slide 1">
    </div>
    <div class="carousel-item">
      <img src="slide2.jpg" class="d-block w-100" alt="Slide 2">
    </div>
    <div class="carousel-item">
      <img src="slide3.jpg" class="d-block w-100" alt="Slide 3">
    </div>
  </div>
  <button class="carousel-control-prev" type="button"
data-bs-target="#carouselExample" data-bs-slide="prev">
    <span class="carousel-control-prev-icon"
aria-hidden="true"></span>
    <span class="visually-hidden">Previous</span>
  </button>
  <button class="carousel-control-next" type="button"
data-bs-target="#carouselExample" data-bs-slide="next">
    <span class="carousel-control-next-icon"
aria-hidden="true"></span>
    <span class="visually-hidden">Next</span>
  </button>
</div>
```

Indicators and Controls:

- Add indicators using the carousel-indicators class.

- Add previous and next controls using carousel-control-prev and carousel-control-next.

Example with Indicators:

```
<div id="carouselExampleIndicators" class="carousel slide"
data-bs-ride="carousel">
  <div class="carousel-indicators">
    <button type="button"
data-bs-target="#carouselExampleIndicators" data-bs-slide-to="0"
class="active" aria-current="true" aria-label="Slide 1"></button>
    <button type="button"
data-bs-target="#carouselExampleIndicators" data-bs-slide-to="1"
aria-label="Slide 2"></button>
    <button type="button"
data-bs-target="#carouselExampleIndicators" data-bs-slide-to="2"
aria-label="Slide 3"></button>
  </div>
  <div class="carousel-inner">
    <div class="carousel-item active">
      <img src="slide1.jpg" class="d-block w-100" alt="Slide 1">
    </div>
    <div class="carousel-item">
      <img src="slide2.jpg" class="d-block w-100" alt="Slide 2">
    </div>
    <div class="carousel-item">
      <img src="slide3.jpg" class="d-block w-100" alt="Slide 3">
    </div>
  </div>
  <button class="carousel-control-prev" type="button"
data-bs-target="#carouselExampleIndicators" data-bs-slide="prev">
    <span class="carousel-control-prev-icon"
aria-hidden="true"></span>
    <span class="visually-hidden">Previous</span>
  </button>
  <button class="carousel-control-next" type="button"
data-bs-target="#carouselExampleIndicators" data-bs-slide="next">
    <span class="carousel-control-next-icon"
aria-hidden="true"></span>
    <span class="visually-hidden">Next</span>
  </button>
</div>
```

Customisation:

- Adjust the transition time by using the data-bs-interval attribute on each carousel item.
- Control the pause on hover behaviour with the data-bs-pause attribute.
- Add captions with carousel captions inside each carousel item.

Example with Captions:

```
<div id="carouselExampleCaptions" class="carousel slide"
data-bs-ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      <img src="slide1.jpg" class="d-block w-100" alt="Slide 1">
      <div class="carousel-caption d-none d-md-block">
        <h5>First Slide</h5>
        <p>Some representative placeholder content for the first
slide.</p>
      </div>
    </div>
    <div class="carousel-item">
      <img src="slide2.jpg" class="d-block w-100" alt="Slide 2">
      <div class="carousel-caption d-none d-md-block">
        <h5>Second Slide</h5>
        <p>Some representative placeholder content for the second
slide.</p>
      </div>
    </div>
    <div class="carousel-item">
      <img src="slide3.jpg" class="d-block w-100" alt="Slide 3">
      <div class="carousel-caption d-none d-md-block">
        <h5>Third Slide</h5>
        <p>Some representative placeholder content for the third
slide.</p>
      </div>
    </div>
  </div>
  <button class="carousel-control-prev" type="button"
data-bs-target="#carouselExampleCaptions" data-bs-slide="prev">
    <span class="carousel-control-prev-icon"
aria-hidden="true"></span>
    <span class="visually-hidden">Previous</span>
  </button>
  <button class="carousel-control-next" type="button"
```

```
data-bs-target="#carouselExampleCaptions" data-bs-slide="next">
    <span class="carousel-control-next-icon"
aria-hidden="true"></span>
    <span class="visually-hidden">Next</span>
  </button>
</div>
```

Conclusion

Understanding and utilizing the various tools and components of web development is crucial for creating responsive, user-friendly, and visually appealing web applications. CSS variables and the !important declaration offer powerful ways to manage styles and ensure that critical styles are applied as needed. Familiarity with browser engines like WebKit and vendor prefixes such as -webkit- helps ensure compatibility across different browsers, enabling developers to create a consistent user experience.

Bootstrap 5 enhances web development by providing a rich set of components, including the responsive navbar with a collapsible hamburger menu and customizable carousels. These components simplify the implementation of common web features, making it easier to build modern, responsive websites. Additionally, the alternative approach for a navbar with an external dropdown menu showcases the flexibility and customization options available in Bootstrap 5, allowing developers to create unique and intuitive navigation experiences. By combining Bootstrap's predefined classes with custom CSS and JavaScript, developers can achieve a high degree of control over their web designs, catering to specific project requirements and user interactions.

References

- https://developer.mozilla.org/en-US/docs/Web/CSS/--*
- https://developer.mozilla.org/en-US/docs/Web/CSS/important
- https://en.wikipedia.org/wiki/Browser_engine
- https://developer.mozilla.org/en-US/docs/Glossary/Vendor_Prefix
- https://developer.mozilla.org/en-US/docs/Web/CSS/WebKit_Extensions
- https://getbootstrap.com/docs/5.3/components/navbar/
- https://getbootstrap.com/docs/5.3/components/carousel/