

Web development

lesson 8

Oct 20, 2024

CSS Syntax Structure 2

- **Selectors**

- Selectors define which HTML elements the style will apply to.
- Can target elements by:
 - **Type** (e.g., `p` , `h1`): Selects all elements of that type.
 - **Class** (e.g., `.example`): Selects all elements with that class.
 - **ID** (e.g., `#unique`): Selects the element with that specific ID.
 - **Attribute** (e.g., `[type="text"]`): Selects elements with specific attributes.
 - **Combinations**: Can combine selectors to increase specificity or target multiple elements.
- Example:

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

CSS Syntax Structure 3

- **Declaration Block**

- Enclosed in curly braces `{ }`, containing one or more declarations.
- Each **declaration** has:
 - A **property** (e.g., `color`, `font-size`, `margin`): Defines the aspect of the element to style.
 - A **value** (e.g., `blue`, `16px`, `auto`): Specifies the style applied to that property.
- Properties and values are separated by a colon (`:`) and end with a semicolon (`;`).
- Multiple declarations can be added within one declaration block, separated by semicolons.

CSS Syntax Structure 4

- **Importance of Semicolons and Formatting**

- Semicolons (;) are crucial to separate multiple declarations within a block.
- Correct syntax ensures styles are applied as expected.
- Formatting with proper indentation and line breaks improves readability and maintainability.

CSS Syntax Structure 5

- **Example of a Complex CSS Rule**

- Example:

```
#header .nav-item:hover {  
  background-color: #333;  
  color: #fff;  
  border-bottom: 2px solid orange;  
}
```

- In this example:

- Selector `#header .nav-item:hover` targets `.nav-item` elements within an element with the ID `header` when hovered.
- The declarations set background and text color and add a border effect on hover.

CSS Colors: Basics

- **What Are CSS Colors?**

- CSS colors define the appearance of elements by specifying color values for properties like `color`, `background-color`, `border-color`, etc.
- Colors enhance visual appeal and are essential for readability, branding, and design aesthetics.

- **Common CSS Color Properties:**

- `color` : Changes text color.
- `background-color` : Sets the background color of an element.
- `border-color` : Defines border color.

- **Example:**

```
p {  
  color: darkgreen;  
  background-color: lightyellow;  
}
```

CSS Colors: Named Colors

- **What Are Named Colors?**

- CSS provides 140 named colors (e.g., `red` , `blue` , `lightgray`) that can be directly used without specifying values.
- Named colors are simple to remember and use, making them convenient for quick styling.

- **Examples of Named Colors:**

- Primary colors: `red` , `green` , `blue`
- Neutral colors: `black` , `white` , `gray`
- Other examples: `cyan` , `magenta` , `orange` , `violet`

- **Usage Example:**

```
h1 {  
  color: tomato;  
}
```

CSS Colors: Hexadecimal Notation

- **What Is Hexadecimal Notation?**

- Hex codes represent colors using a six-digit code (e.g., `#RRGGBB`), where `RR` , `GG` , and `BB` stand for red, green, and blue values.
- Each pair ranges from `00` to `FF` , with `00` being no color and `FF` being full color.

- **Examples of Hex Codes:**

- Black: `#000000`
- White: `#FFFFFF`
- Red: `#FF0000`
- Example with shorthand: `#F00` (same as `#FF0000`)

```
div {  
  background-color: #3498db;  
}
```


CSS Colors: RGB and RGBA

- **What Are RGB and RGBA?**

- RGB stands for Red, Green, Blue. Colors are specified with values between 0 and 255 for each component.
- RGBA adds an alpha channel for transparency, ranging from 0 (fully transparent) to 1 (fully opaque).

- **Syntax and Examples:**

- RGB: `rgb(255, 0, 0)` (red)
- RGBA: `rgba(255, 0, 0, 0.5)` (semi-transparent red)

- **Usage Example:**

```
.overlay {  
  background-color: rgba(0, 0, 0, 0.7);  
}
```

CSS Colors: HSL and HSLA

- **What Are HSL and HSLA?**

- HSL stands for Hue, Saturation, Lightness. Hue is given as a degree (0–360), Saturation and Lightness are percentages.
- HSLA adds an alpha channel for transparency, similar to RGBA.

- **Syntax and Examples:**

- HSL: `hsl(240, 100%, 50%)` (blue)
- HSLA: `hsla(240, 100%, 50%, 0.3)` (semi-transparent blue)

- **Usage Example:**

```
.highlight {  
  color: hsl(120, 60%, 70%);  
}
```

CSS Colors: Advanced Techniques

- **Opacity and Transparency**

- Use `opacity` to set transparency for an entire element. Ranges from 0 (transparent) to 1 (opaque).
- RGBA/HSLA for specific color transparency.

- **CSS Gradients**

- Create transitions between multiple colors.
- **Linear Gradient:** `background: linear-gradient(to right, red, yellow);`
- **Radial Gradient:** `background: radial-gradient(circle, red, yellow, blue);`

CSS Colors: Advanced Techniques 2

- **Blend Modes and Filters**

- **Mix-Blend-Mode:** Blends an element with the background (e.g., `multiply`, `screen`).
- **Filter:** Applies effects like `grayscale()`, `sepia()`, `brightness()`.

- **Example:**

```
.blend {  
  background: url('image.jpg');  
  mix-blend-mode: overlay;  
}
```