

<meta charset="UTF-8">>:

This meta tag specifies the character encoding for the HTML document.

UTF-8 is a character encoding capable of encoding all possible characters (called code points) in Unicode. This includes characters from almost all written languages, mathematical symbols, and even emoji.

Using UTF-8 ensures that your web page can display text correctly across different languages and symbols without any issues.

<meta name="viewport" content="width=device-width, initial-scale=1.0">>:

This meta tag provides instructions on how browsers should control the page's dimensions and scaling.

width=device-width sets the width of the page to follow the screen-width of the device (i.e., it will be responsive to different screen sizes).

initial-scale=1.0 sets the initial zoom level when the page is first loaded by the browser. In this case, the page will be displayed at a 1:1 scale, meaning no zooming in or out.

Cascading style sheet (CSS), what is CSS, Why we use CSS, adding CSS to your web pages, Grouping styles-extensible, Introduction to XML, uses of XML, simple XML, XML key components, DTD and Schemas, Using XML with application. Transforming XML using XSL and XSLT. Dynamic HTML: Document object model (DCOM)-Accessing HTML & CSS through DCOM Dynamic content styles & positioning-Event bubbling-data binding.

by Pratishtha Gupta

What is CSS?

- CSS stands for Cascading Style Sheets.
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files.

Why Use CSS?

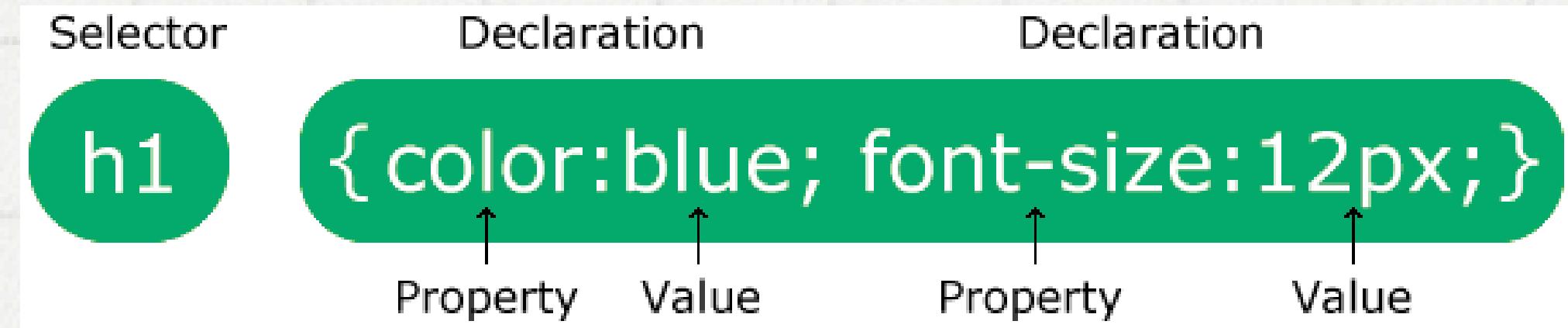
- CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.
- With an external stylesheet file, you can change the look of an entire website by changing just one file.
- The style definitions are normally saved in external .css files.

- HTML was NEVER intended to contain tags for formatting a web page!
- HTML was created to describe the content of a web page, like: <h1>This is a heading</h1>
- When tags like , and color attributes were added to the HTML 3.2 specification. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.
- To solve this problem, the World Wide Web Consortium (W3C) created CSS.
- CSS removed the style formatting from the HTML page.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: lightblue;
}
h1 {
    color: white;
    text-align: center;
}
p {
    font-family: verdana;
    font-size: 20px;
}
</style>
</head>
<body>
<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

CSS Syntax

- A CSS rule consists of a selector and a declaration block.



- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

Example

- p is a selector in CSS (it points to the HTML element you want to style: <p>).
- color is a property, and red is the property value.
- text-align is a property, and center is the property value.

```
p {  
    color: red;  
    text-align: center;  
}
```

Adding CSS to your Web Pages

- When a browser reads a style sheet, it will format the HTML document according to the information in the style sheet.

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External CSS
- Internal CSS
- Inline CSS

External CSS: Linked via <link> tag (assumes you have a .css file).

Internal CSS: Defined within the <style> tag in the <head>.

Inline CSS: Style applied directly to an HTML element using the style attribute.

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet"
      href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

External CSS

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: linen;
}
h1 {
    color: maroon;
    margin-left: 40px;
}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

Internal CSS

```
<!DOCTYPE html>
<html>
<body>

<h1
style="color:blue;text-align:center;">This
is a heading</h1>
<p
style="color:red;">This
is a paragraph.</p>

</body>
</html>
```

Inline CSS

External CSS:

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

Internal CSS:

```
<head>
<style>
body{ background-color: pink;}
h1{margin-left: 40px;}
</style>
</head>
```

Inline CSS:

```
<body>
<h1 style="color:red;">Heading Statement</h1>
```

External CSS

- With an external style sheet, you can change the look of an entire website by changing just one file.
- Each HTML page must include a reference to the external style sheet file inside the `<link>` element, inside the head section.

Example

- An external style sheet can be written in any text editor, and must be saved with a .css extension.
- The external .css file should not contain any HTML tags.

Internal CSS

- An internal style sheet may be used if one single HTML page has a unique style.
- The internal style is defined inside the `<style>` element, inside the head section.

Inline CSS

- An inline style may be used to apply a unique style for a single element.
- To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

Multiple Style Sheets

- If some properties have been defined for the same selector (element) in different style sheets, the value from the last read style sheet will be used.

Example

- Assume that an external style sheet has the following style for the <h1> element.
- Then, assume that an internal style sheet also has the following style for the <h1> element:

```
h1 {  
    color: navy;  
}
```

```
h1 {  
    color: orange;  
}
```

- a. If the internal style is defined after the link to the external style sheet, the `<h1>` elements will be "orange".
- b. However, if the internal style is defined before the link to the external style sheet, the `<h1>` elements will be "navy".

```
<head>
<link rel="stylesheet"
type="text/css"
href="mystyle.css">
<style>
h1 {
  color: orange;
}
</style>
</head>
```

a.

```
<head>
<style>
h1 {
  color: orange;
}
</style>
<link rel="stylesheet"
type="text/css"
href="mystyle.css">
</head>
```

b.

Cascading Order

- What style will be used when there is more than one style specified for an HTML element?
- All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:
- Inline style (inside an HTML element)
- External and internal style sheets (in the head section)
- Browser default
- So, an inline style has the highest priority, and will override external and internal styles and browser defaults.

CSS Selectors

- A CSS selector selects the HTML element(s) you want to style.
- CSS selectors are used to "find" (or select) the HTML elements you want to style.
- We can divide CSS selectors into five categories:
- **Simple selectors** (select elements based on name, id, class)
- **Combinator selectors** (select elements based on a specific relationship between them)
- **Pseudo-class selectors** (select elements based on a certain state such as visited and unvisited links, element in focus, mouse hovers over the element.)
- **Pseudo-elements selectors** (select and style a part of an element)
- **Attribute selectors** (select elements based on an attribute or attribute value)

CSS element Selector

- The element selector selects HTML elements based on the element name.

Example

- Here, all `<p>` elements on the page will be center-aligned, with a red text color:

CSS id Selector

- The id selector uses the `id` attribute of an HTML element to select a specific element.
- The id of an element is unique within a page, so the id `selector` is used to select one unique element.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<p>Every paragraph will be affected
by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>

</body>
</html>
```

Uniqueness: The ID selector is used to style a single, unique element. Each ID must be unique within the HTML document.

Syntax: Uses the hash symbol (#) followed by the element's ID.

Specificity: Higher specificity compared to class selectors. ID selectors are more specific and will override class selectors if both apply to the same element.

- To select an element with a specific id, write a hash (#) character, followed by the id of the element. Only specified elements will be submitted to the change.

Example

- The CSS rule below will be applied to the HTML element with id="para1".

CSS class Selector

- The class selector selects HTML elements with a specific class attribute.
- To select elements with a specific class, write a period (.) character, followed by the class name.

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<p id="para1">Hello World!</p>
<p>This paragraph is not
affected by the style.</p>

</body>
</html>
```

Reusability: The class selector is used to style multiple elements. Multiple elements can share the same class.

Syntax: Uses the dot symbol (.) followed by the class name.

Specificity: Lower specificity compared to ID selectors. Useful for applying general styles to multiple elements.

Example

- In this example all HTML elements with class="center" will be red and center-aligned.
- You can also specify that only specific HTML elements should be affected by a class.

Example

- p.center { text-align: center; color:red;}
- In this example only <p> elements with class="center" will be red and center-aligned.
- HTML elements can also refer to more than one class.

Example

- <p class="center large">This paragraph refers to two classes.</p>

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<h1 class="center">Red and center-
aligned heading</h1>
<p class="center">Red and center-
aligned paragraph.</p>

</body>
</html>
```

CSS Universal Selector

- The universal selector (*) selects all HTML elements on the page.

```
* {  
    text-align: center;  
    color: blue;  
}
```

Example

- The CSS rule below will affect every HTML element on the page:

CSS Grouping Selector

- The grouping selector selects all the HTML elements with the same style definitions. Elements are separated using commas

```
h1, h2, p {  
    text-align: center;  
    color: red;  
}
```

CSS Comments

- Comments are used to explain the code, and may help when you edit the source code at a later date.
- Comments are ignored by browsers.
- A CSS comment is placed inside the `<style>` element, and starts with `/*` and ends with `*/`.

Single Line Comment

- `/* This is a single-line comment */`

Multi-Line Comment

- Comments can also span multiple lines:

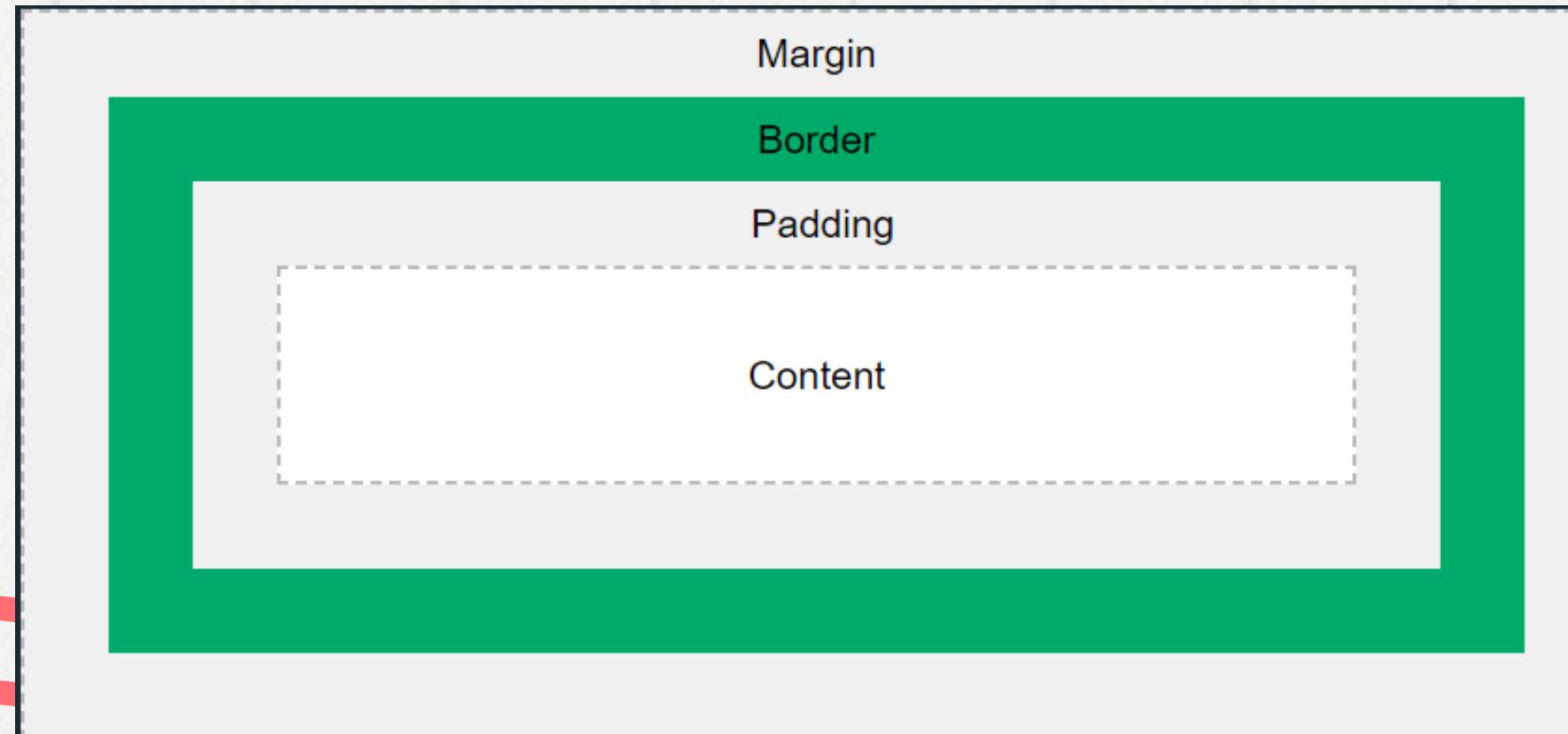
```
/* This is  
a multi-line  
comment */
```

CSS Commonly used Properties

- **CSS Color:** In CSS, you can use a color name to specify colors. Eg: Orange, Violet, Gray
- To add **color** to the text, use **color** property. Eg: `<h1 style="color:Gray;">Hello World</h1>`
- To add **background-color**, use **background-color** property. Eg: `<h1 style="background-color:DodgerBlue;">Hello World</h1>`
- To add color to **border**, use **border: name of color**. Eg: `<h1 style="border: 2px solid DodgerBlue;">Hello World</h1>`
- Colors can be specified using 3 methods: RGB values, HEX values, HSL values
 - `<h1 style="background-color:rgb(255, 99, 71);">rgb(255, 99, 71)</h1>`
 - `<h1 style="background-color:#ff6347;">#ff6347</h1>`
 - `<h1 style="background-color:hsl(9, 100%, 64%);">hsl(9, 100%, 64%)</h1>`

- **CSS backgrounds**

- a. background-color
- b. background-image
- c. background-repeat
- d. background-attachment
- e. background



- **CSS Box Model**

- The CSS box model is essentially a box that wraps around every HTML element. It consists of: content, padding, borders and margins.
- **Padding** - Clears an area around the content. The padding is transparent.
- **Border** - A border that goes around the padding and content.
- **Margin** - Clears an area outside the border. The margin is transparent.

border-style: "dotted, dashed, solid, double, groove, ridge, inset, outset, none, hidden;"

margin-top
margin-right
margin-bottom
margin-left

auto - the browser calculates the margin
length - specifies a margin in px, pt, cm, etc.
% - specifies a margin in % of the width of the containing element
inherit - specifies that the margin should be inherited from the parent element

- **CSS borders**

- a. border-style
- b. border-width
- c. border-color
- d. border-sides
- e. rounded border

- **CSS Padding**

- Padding is used to **create space around an element's content, inside of any defined borders.**

- **CSS Margin**

- a. Margins are used to **create space around elements, outside of any defined borders.**
- b. CSS has properties for specifying the margin for each side of an element:
 - i. margin-top
 - ii. margin-right
 - iii. margin-bottom
 - iv. margin-left
- c. To shorten the code, it is possible to specify all the margin properties in one property using margin property.