Lecture No: 5
Topic: CSS3

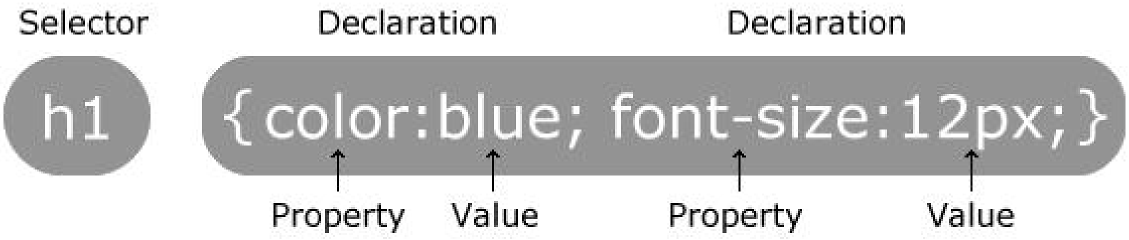
Introduction to CSS3

- The principle of Cascading Style Sheets (CSS) has roots in Standardized Generalized Markup Language (SGML) from the **1980s**. Its goals are to create a consistent look across many webpages and to separate structure from presentation so you can provide different style sheets for printing, browsing, or other scenarios.
- The World Wide Web Consortium (W3C) published CSS Level 1 recommendations in **December 1996**, and then started working on CSS Level 2. (The word recommendation means a formal release of the publication.) In **May 1998**, CSS Level 2 was published, and the W3C started working on various modules of CSS Level 3 (CSS3). Rather than creating one large publication for CSS3, the W3C separated CSS3 into modules that could be published independently.
- **In 2005**, the W3C became stricter with enforcement of the requirements for standards, and already-published standards such as CSS Level 2 Revision 1 (CSS 2.1), CSS3 Selectors, and CSS3 Text were pulled back from Candidate Recommendation to Working Draft level. It wasn't until **June 2011** that the CSS 2.1 recommendation was published. Because CSS3 is modular, the stability of each module differs, and its status differs. More than 50 modules are published from the W3C's CSS Working Group.
- CSS 2.1 is included in CSS3 because CSS3 is being designed to be backward compatible with the CSS 2.1 recommendation. This chapter introduces you to CSS3 by covering many aspects of CSS 2.1. Cascading style sheets provide a means to apply a presentation to an HTML structure by defining how HTML elements are displayed. By using CSS, you can set background and foreground colors, margins, borders, fonts, positions, and much more. You have creative control of the HTML elements, so you can decide what the elements look like and where they display on the screen.

CSS Solved a Big Problem

- HTML was **NEVER** intended to contain tags for formatting a web page.
- When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. 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.

CSS Syntax and Selectors



- A CSS rule-set 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**.
- A CSS declaration always ends with a **semicolon**, and declaration blocks are surrounded by **curly braces**.

The element Selector

- The element selector selects elements based on the element name.
- You can select a
- ll <p> elements on a page like this (in this case, all <p> elements will be center-aligned, with a red text color):
- **p {**

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

The id Selector

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element should be unique within a page, so the id selector is used to select one unique element!
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.
- The style rule below will be applied to the HTML element with id="para1":

```
#para1 {
  text-align: center;
  color: red;
}
```

The class Selector

- The class selector selects elements with a specific class attribute.
- To select elements with a specific class, write a period (.) character, followed by the name of the class.
- In the example below, all HTML elements with class="center" will be red and center-aligned:

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

Grouping Selectors

- If you have elements with the same style definitions, like this:

```
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 starts with /* and ends with */. Comments can also span multiple lines:

```
p {
  color: red;
  /* This is a single-line comment */
  text-align: center;
}
/* This is
a multi-line
comment */
```

Three Ways to Insert CSS

- There are three ways of inserting a style sheet:

- External style sheet
- Internal style sheet
- Inline style

External Style Sheet

- With an external style sheet, you can change the look of an entire website by changing just one file!
- Each page must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the <head> section:
- **<head>**
  <link rel="stylesheet" type="text/css" href="mystyle.css">
  </head>

mystyle.css

-
  ```
  body {
    background-color: lightblue;
  }

  h1 {
    color: navy;
    margin-left: 20px;
  }
  ```

Internal Style Sheet

- An internal style sheet may be used if one single page has a unique style.
- Internal styles are defined within the <style> element, inside the <head> section of an HTML page:

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

Inline Styles

- 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.
- The example below shows how to change the color and the left margin of a <h1> element:
- **<h1 style="color:blue;margin-left:30px;">This is a heading</h1>**

Cascading Order

- What style will be used when there is more than one style specified for an HTML element?

- Generally speaking, we can say that all the styles 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 (inside a specific HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or a browser default value.

CSS COLORS

- Background Color
  - **<h1 style="background-color:DodgerBlue;">Hello World</h1>**
- Text Color
  - **<h1 style="color:Tomato;">Hello World</h1>**
- Border Color
  - **<h1 style="border:2px solid DodgerBlue;">Hello World</h1>**

Color Values

- In HTML, colors can also be specified using
  - RGB values **rgb(*red, green*, *blue*)**,
  - HEX values ***#rrggbb***,
  - HSL values **hsl(*hue*, *saturation*, *lightness*)**,
  - RGBA values **rgba(*red, green*, *blue, alpha*)**, and
  - HSLA values **hsla(*hue, saturation*, *lightness*, *alpha*)**

CSS Backgrounds

- The CSS background properties are used to define the background effects for elements.
- CSS background properties:
  - background-color
  - background-image
  - background-repeat
  - background-attachment
  - background-position

Example:

-
  ```
  body {
    background-image: url("img_tree.png");
    background-repeat: no-repeat;
    background-position: right top;
    background-attachment: fixed;
  }
  ```

Background - Shorthand property

- To shorten the code, it is also possible to specify all the background properties in one single property. This is called a **shorthand property**.
- The shorthand property for background is background:

-
  ```
  body {
    background: #ffffff url("img_tree.png") no-repeat right top;
  }
  ```

CSS Borders

- The CSS border properties allow you to specify the style, width, and color of an element's border.
- The **border-style** property specifies what kind of border to display.

The following values are allowed:

  - **dotted** - Defines a dotted border
  - **dashed** - Defines a dashed border
  - **solid -** Defines a solid border
  - **double** - Defines a double border
  - **groove** - Defines a 3D grooved border. The effect depends on the border-color value
  - **ridge** - Defines a 3D ridged border. The effect depends on the border-color value
  - **inset** - Defines a 3D inset border. The effect depends on the border-color value
  - **outset -** Defines a 3D outset border. The effect depends on the border-color value
  - **none -** Defines no border
  - **hidden -** Defines a hidden border

Example:

-
  ```
  p.dotted {border-style: dotted;}
  p.dashed {border-style: dashed;}
  p.solid {border-style: solid;}
  p.double {border-style: double;}
  p.groove {border-style: groove;}
  ```

Border Width

- The **border-width** property specifies the width of the four borders.
- The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick.
- The border-width property can have from one to four values (for the top border, right border, bottom border, and the left border).

Border Color

- The **border-color** property is used to set the color of the four borders.
- The color can be set by:
  - name - specify a color name, like "red"
  - Hex - specify a hex value, like "#ff0000"
  - RGB - specify a RGB value, like "rgb(255,0,0)"
  - transparent
- The border-color property can have from one to four values (for the top border, right border, bottom border, and the left border).

Border - Individual Sides

- In CSS, there are also properties for specifying each of the borders (top, right, bottom, and left):

-
  ```
  p {
    border-top-style: dotted;
    border-right-style: solid;
    border-bottom-style: dotted;
    border-left-style: solid;
  }
  ```

Border - Shorthand Property

- The border property is a shorthand property for the following individual border properties:

  - border-width
  - border-style (required)
  - border-color

- **p {**
  border: 5px solid red;
  }

Left Border

  - **p {**
    border-left: 6px solid red;
    background-color: lightgrey;
    }

Bottom Border

  - **p {**
    border-bottom: 6px solid red;
    background-color: lightgrey;
    }

Rounded Borders

The **border-radius** property is used to add rounded borders to an element:

```
p {
  border: 2px solid red;
  border-radius: 5px;
}
```

CSS Height and Width

- The height and width properties are used to set the height and width of an element.
- The height and width can be set to auto (this is default. Means that the browser calculates the height and width), or be specified in length values, like px, cm, etc., or in percent (%) of the containing block.

Setting max-width

- The **max-width** property is used to set the maximum width of an element.
- The max-width can be specified in length values, like px, cm, etc., or in percent (%) of the containing block, or set to none (this is default. Means that there is no maximum width).

CSS Border Images

- The CSS border-image property allows you to specify an image to be used instead of the normal border around an element.
- The property has three parts:

1. The image to use as the border
2. Where to slice the image
3. Define whether the middle sections should be repeated or stretched

```
#borderimg {
  border: 10px solid transparent;
  padding: 15px;
  border-image: url(border.png) 30 round;
}
```

CSS Styling Images

- **Rounded Images**
- Use the border-radius property to create rounded images:

  -
    ```
    img {
       border-radius: 8px;
    }
    ```
- **OR**

  -
    ```
    img {
       border-radius: 50%;
    }
    ```

Thumbnail Images

- Use the border property to create thumbnail images.

-
  ```
  img {
     border: 1px solid #ddd;
     border-radius: 4px;
     padding: 5px;
     width: 150px;
  }
  <img src="paris.jpg" alt="Paris">
  ```

Thumbnail Image as Link:

-
  ```
  img {
     border: 1px solid #ddd;
     border-radius: 4px;
     padding: 5px;
     width: 150px;
  }
  img:hover {
     background-color: #f5f5f5;}
  }
  <a href="paris.jpg">
   <img src="paris.jpg" alt="Paris">
  </a>
  ```

Responsive Images

- Responsive images will automatically adjust to fit the size of the screen.
- Resize the browser window to see the effect:

-
  ```
  img {
     max-width: 100%;
     height: auto;
  }
  ```

Transparent Image

- The opacity property can take a value from 0.0 - 1.0. The lower value, the more transparent:

-
  ```
  img {
     opacity: 0.5;
  ```

```
  filter: alpha(opacity=50); /* For IE8 and earlier */
}
```

- **Note:** IE8 and earlier use filter:alpha(opacity=x). The x can take a value from 0 - 100. A lower value makes the element more transparent.

CSS Tables

- **Table Borders**

  -
    ```
    table, th, td {
      border: 1px solid black;
    }
    ```
- **Collapse Table Borders**
  - The border-collapse property sets whether the table borders should be collapsed into a single border

-
  ```
  table {
    border-collapse: collapse;
  }
  table, th, td {
    border: 1px solid black;
  }
  ```

- **Table Width and Height**

  -
    ```
    table {
      width: 100%;
    }
    th {
      height: 50px;
    }
    ```
- **Horizontal Alignment**
  - The text-align property sets the horizontal alignment (like **left, right**, or **cente**r) of the content in <th> or <td>.
- **Vertical Alignment**
  - The vertical-align property sets the vertical alignment (like **top, bottom**, or **middle**) of the content in <th> or <td>.
- **Table Padding**
  - To control the space between the border and the content in a table, use the padding property on <td> and <th> elements

  -
    ```
    th, td {
      padding: 15px;
      text-align: left;
    }
    ```
- **Hoverable Table**
  - Use the :**hover** selector on <tr> to highlight table rows on mouse over:
  - **tr:hover {background-color: #f5f5f5;}**
- **Table Color**

- The example below specifies the background color and text color of <th> elements:

-
  ```
  th {
    background-color: #4CAF50;
    color: white;
  }
  ```

CSS Text Color

The **color** property is used to set the color of the text. The color is specified by:

- a color name - like "red"
- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)"

Text Alignment

- The **text-align** property is used to set the horizontal alignment of a text.
- A text can be left or right aligned, centered, or justified.

-
  ```
  h1 {
    text-align: center;
  }
  ```

Text Decoration

- The **text-decoration** property is used to set or remove decorations from text.

-
  ```
  h1 {
    text-decoration: overline;
  }
  h2 {
    text-decoration: line-through;
  }
  h3 {
    text-decoration: underline;
  }
  ```

Text Tranformation

- The text-transform property is used to specify uppercase and lowercase letters in a text.

```
p.uppercase {
  text-transform: uppercase;
}
p.lowercase {
  text-transform: lowercase;
}
p.capitalize {
  text-transform: capitalize;
}
```

Text Indentation

```
p {
  text-indent: 50px;
}
```

Letter Spacing

```
h1 {
  letter-spacing: 3px;
}

h2 {
  letter-spacing: -3px;
}
```

Line Height

```
p.small {
  line-height: 0.8;
}

p.big {
  line-height: 1.8;
}
```

Text Direction

```
p {
  direction: rtl;
}
```

Word Spacing

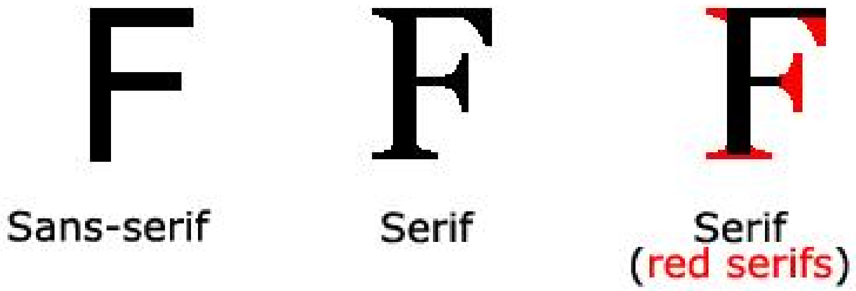
```
h1 {
  word-spacing: 10px;
}
h2 {
  word-spacing: -5px;
}
```

Text Shadow

```
h1 {
  text-shadow: 3px 2px red;
}
```

CSS Fonts

- Difference Between Serif and Sans-serif Fonts



CSS Font Families

- In CSS, there are two types of font family names:
  - **generic family** - a group of font families with a similar look (like "Serif" or "Monospace")
  - **font family** - a specific font family (like "Times New Roman" or "Arial")

Generic family	Font family	Description
Serif	Times New Roman Georgia	Serif fonts have small lines at the ends on some characters
Sans-serif	Arial Verdana	"Sans" means without - these fonts do not have the lines at the ends of characters
Monospace	Courier New Lucida Console	All monospace characters have the same width

Font Family

- The font family of a text is set with the **font-family** property.
- The font-family property should hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font, and so on.
- Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.
-
```
p {
  font-family: "Times New Roman", Times, serif;
}
```

Font Style

- The font-style property is mostly used to specify italic text.

This property has three values:

  - normal - The text is shown normally
  - italic - The text is shown in italics
  - oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

```
p.normal {
  font-style: normal;
}
```

Font Size

Set Font Size With Pixels

- h1 {
```
  font-size: 40px;
}
```

Set Font Size With Percent

- body {
```
  font-size: 100%;
}
```

Set Font Size With Em

- To allow users to resize the text (in the browser menu), many developers use em instead of pixels.
- The em size unit is recommended by the W3C.
- 1em is equal to the current font size. The default text size in browsers is 16px. So, the default size of 1em is 16px.
- The size can be calculated from pixels to em using this formula: *pixels*/16=*em*
- h1 {
```
  font-size: 2.5em; /* 40px/16=2.5em */
}
```

Font Weight

- The font-weight property specifies the weight of a font
- p.normal {
```
  font-weight: normal;
}
p.thick {
  font-weight: bold;
}
```

Font Variant

- The font-variant property specifies whether or not a text should be displayed in a small-caps font.

In a small-caps font, all lowercase letters are converted to uppercase letters. However, the converted uppercase letters appear in a smaller font size than the original uppercase letters in the text.

- p.normal {
```
  font-variant: normal;
}
p.small {
  font-variant: small-caps;
}
```

References:

- Carey. P (2018). *New perspectives on HTML5, CSS3, and Javascript*. Australia : Cengage Learning
- Jenkins, S. (2013). Web Design All-in-One for Dummies. John Wiley & Sons.
- Parker, J. (2021). HTML for Beginners: A Complete Beginners Guide to Learn Html in 1 Hour and Master Your Web Designing.
- W3Schools online web tutorials. (n.d.). https://www.w3schools.com/
- Sklar, J. (2012). *Web Design Principles*.
- Castro, E., & Hyslop, B. (2013). *HTML and CSS: Visual QuickStart Guide*. Peachpit Press.