eduonix

The Live FULLSTACK DEVELOPMENT Certification Program

Agenda of the Session

- Introduction to CSS
- Inline, Internal & External CSS
- Selectors in CSS
- Using Chrome Developer Tools
- Fonts In CSS
- Colors In CSS
- Borders and Backgrounds
- Visibility & z-index Explained

CSS - Introduction

- CSS is the language we use to style a Web page.
- 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

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

How to add CSS

- There are three ways of inserting a style sheet:

External CSS

Internal CSS

Inline CSS

How to add CSS (Examples)

```html
<!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>
```

```html
<!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>
```

```html
<!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>
```

CSS Selectors

- A CSS selector selects the HTML element(s) 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)

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:

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

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
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.

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

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.

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

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

- **Note:** A class name cannot start with a number!

CSS Universal & grouping selector

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

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

- The grouping selector selects all the HTML elements with the same style definitions.

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

Fonts in CSS

- Choosing the right font has a huge impact on how the readers experience a website.
- The right font can create a strong identity for your brand.
- Using a font that is easy to read is important. The font adds value to your text. It is also important to choose the correct color and text size for the font.

Generic font families

- In CSS there are five generic font families:

Serif fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.

Sans-serif fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.

Monospace fonts - here all the letters have the same fixed width. They create a mechanical look.

Cursive fonts imitate human handwriting.

Fantasy fonts are decorative/playful fonts.

font-family example

```css
.p1 {
  font-family: "Times New Roman", Times, serif;
}

.p2 {
  font-family: Arial, Helvetica, sans-serif;
}

.p3 {
  font-family: "Lucida Console", "Courier New", monospace;
}
```

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;
}

p.italic {
  font-style: italic;
}

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

font-weight

- It specifies the weight of a font.

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

p.thick {
  font-weight: bold;
}
```

font-size

- font-size property sets the size of the text.
- Being able to manage the text size is important in web design. *However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.*
- Absolute size:

Sets the text to a specified size

Does not allow a user to change the text size in all browsers (bad for accessibility reasons)

Absolute size is useful when the physical size of the output is known

- Relative size:

Sets the size relative to surrounding elements

Allows a user to change the text size in browsers

font-size example

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

h2 {
  font-size: 30px;
}

p {
  font-size: 14px;
}
```

set font-size using em

- To allow users to resize the text (in the browser menu), many developers use em instead of pixels.
- 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 */
}

h2 {
  font-size: 1.875em; /* 30px/16=1.875em */
}

p {
  font-size: 0.875em; /* 14px/16=0.875em */
}
```

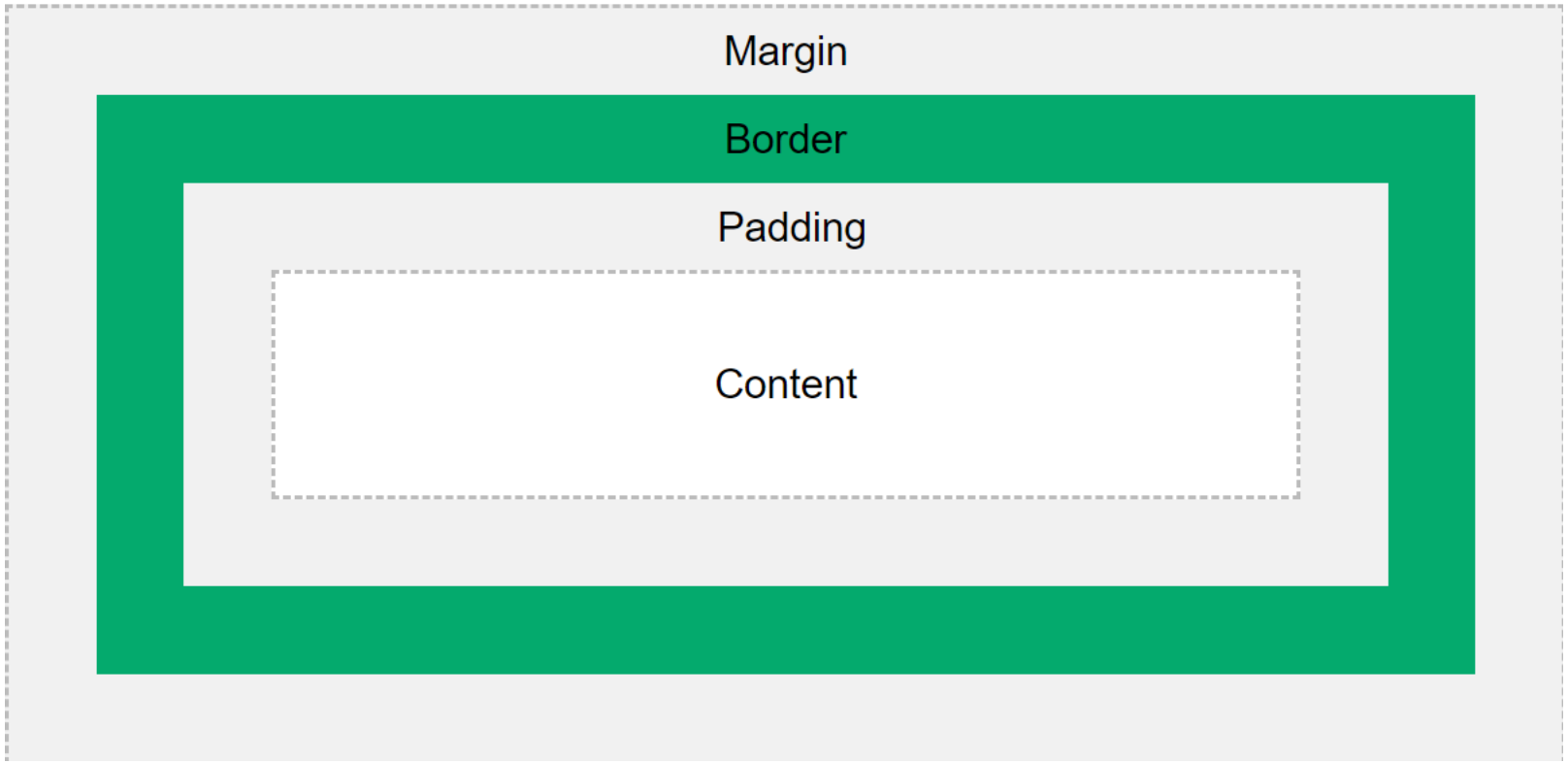
colors in CSS

- Colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.
- 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 Tomato;">Hello World</h1>

CSS Box model

- In CSS, the term "box model" is used when talking about design and layout.
- The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:

Box Model



Box Model

- Explanation of the different parts:

Content - The content of the box, where text and images appear

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

CSS border

- 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

The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).

border-style (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;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
```

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:

```
p.one {
  border-style: solid;
  border-width: 5px;
}

p.two {
  border-style: solid;
  border-width: medium;
}

p.three {
  border-style: dotted;
  border-width: 2px;
}

p.four {
  border-style: dotted;
  border-width: thick;
}
```

```
p.one {
  border-style: solid;
  border-width: 5px 20px; /* 5px top and bottom, 20px on the sides */
}

p.two {
  border-style: solid;
  border-width: 20px 5px; /* 20px top and bottom, 5px on the sides */
}

p.three {
  border-style: solid;
  border-width: 25px 10px 4px 35px; /* 25px top, 10px right, 4px bottom and 35px left */
}
```

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)“

```
p.one {
  border-style: solid;
  border-color: red;
}

p.two {
  border-style: solid;
  border-color: green;
}

p.three {
  border-style: dotted;
  border-color: blue;
}
```

CSS Margins

- The CSS margin properties are used to create space around elements, outside of any defined borders.
- With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

CSS Margins

- CSS has properties for specifying the margin for each side of an element:

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

```
p {
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
}
```

All the margin properties can have the following values:

- 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
- Tip: Negative values are allowed.

CSS Margins shorthand

- To shorten the code, it is possible to specify all the margin properties in one property.
- margin: 25px 50px 75px 100px;

top margin is 25px

right margin is 50px

bottom margin is 75px

left margin is 100px

```
p {
  margin: 25px 50px 75px 100px;
}
```

CSS Margin shorthand

- If the margin property has three values:

margin: 25px 50px 75px;

top margin is 25px

right and left margins are 50px

bottom margin is 75px

CSS margin shorthand

- If the margin property has two values:
- margin: 25px 50px;

top and bottom margins are 25px

right and left margins are 50px

CSS Padding

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

```
div {
  padding-top: 50px;
  padding-right: 30px;
  padding-bottom: 50px;
  padding-left: 80px;
}
```

CSS height and width

- The CSS height and width properties are used to set the height and width of an element.
- The CSS max-width property is used to set the maximum width of an element.

```
div {
  height: 200px;
  width: 50%;
  background-color: powderblue;
}
```

CSS max-width

- The problem with the above occurs when the browser window is smaller than the width of the element (500px). The browser then adds a horizontal scrollbar to the page.
- Using max-width instead, in this situation, will improve the browser's handling of small windows.

```
div {
  max-width: 500px;
  height: 100px;
  background-color: powderblue;
}
```

Git and Github links

- Git - Downloads (git-scm.com)
- GitHub

Session Summary

THANK YOU

for attending the session