

- CSS is the language we use to style an HTML document.
- CSS describes how HTML elements should be displayed.

CSS Solved a Big Problem

- 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>`
- `<p>This is a paragraph.</p>`
- When tags like ``, 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 stands for Cascading Style Sheets.

- Cascading Style Sheets (CSS) is used to format the layout of a webpage.
- With CSS, you can control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes, and much more!

Using CSS

- CSS can be added to HTML documents in 3 ways:
- **Inline** - by using the `style` attribute inside HTML elements
- **Internal** - by using a `<style>` element in the `<head>` section
- **External** - by using a `<link>` element to link to an external CSS file
- The most common way to add CSS, is to keep the styles in external CSS files..

Inline CSS

- An inline CSS is used to apply a unique style to a **single HTML element**.
- An inline CSS uses the style attribute of an HTML element.
- The following example sets the text color of the <h1> element to blue, and the text color of the <p> element to red:

Internal CSS

- An internal CSS is used to define a style for a **single HTML page**.
- An internal CSS is defined in the <head> section of an HTML page, within a <style> element.
- The following example sets the text color of ALL the <h1> elements (on that page) to blue, and the text color of ALL the <p> elements to red.
- In addition, the page will be displayed with a "powderblue" background color:

External CSS

- An external style sheet is used to define the **style for many HTML pages.**
- To use an external style sheet, add a link to it in the <head> section of each HTML page:

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

Selector

h1

Declaration

Declaration

{color:blue; font-size:12px;}

Property

Value

Property

Value

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

# CSS Selectors

- 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)
- [Pseudo-elements selectors](#) (select and style a part of an element)
- [Attribute selectors](#) (select elements based on an attribute or attribute value)

Simple selectors (select elements  
based on name, id, class)

- The **universal selector (\*)** selects all HTML elements on the page.
- Example
- The CSS rule below will affect every HTML element on the page:
- \*
- {  
text-align: center;  
color: blue;  
}
-

# All CSS Simple Selectors

- *#id* #firstname
- Selects the element with id="firstname"
- *.class* .intro
- Selects all elements with class="intro"
- *element.class* p.intro
- Selects only <p> elements with class="intro"
- *\** \*Selects all elements
- *Element* p Selects all <p> elements
- *element,element,..* div, p Selects all <div> elements and all <p> elements

# Three Ways to Insert CSS

- There are three ways of inserting a style sheet:
- External CSS
- Internal CSS
- Inline CSS
-

# 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.**



- `<!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>`

# 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.

- ```
<!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>
```

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.

- <!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 comments

- CSS comments are not displayed in the browser, but they can help document your source code.
- 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 `*/`

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

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

HTML and CSS Comments

- ```
<!DOCTYPE html>
<html>
<head>
<style>
p {
 color: red; /* Set text color to red ...CSS comment*/
}
</style>
</head>
<body>

<h2>My Heading</h2>

<!-- These paragraphs will be red – HTML comment>
<p>Hello World!</p>
<p>This paragraph is styled with CSS.</p>
<p>CSS comments are not shown in the output.</p>

</body>
</html>
```



# CSS Colors

Colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.

- 1. CSS/HTML support [140 standard color names](#).
- Ex: `<h1 style="background-color:DodgerBlue;">Hello World</h1>`
- 2. colors can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values:
- Same as color name "Tomato":
- `<h1 style="background-color:rgb(255, 99, 71);">`
- [https://www.w3schools.com/css/css\\_colors.asp](https://www.w3schools.com/css/css_colors.asp)

- a color is most often specified by:
- a valid color name - like "red"
- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)"

## CSS Backgrounds

The CSS background properties are used to add background effects for elements.

- background-color
- background-image
- background-repeat
- background-attachment
- background-position
- background (shorthand property)

# Opacity / Transparency

## Opacity / Transparency

The `opacity` property specifies the opacity/transparency of an element. It can take a value from 0.0 - 1.0. The lower value, the more transparent:



### Example

```
div {
 background-color: green;
 opacity: 0.3;
}
```

## Transparency using RGBA

If you do not want to apply opacity to child elements, like in our example above, use **RGBA** color values. The following example sets the opacity for the background color and not the text:



You learned from our [CSS Colors Chapter](#), that you can use RGB as a color value. In addition to RGB, you can use an RGB color value with an **alpha** channel (**RGBA**) - which specifies the opacity for a color.

An RGBA color value is specified with: `rgba(red, green, blue, alpha)`. The *alpha* parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque).

- `<div style="opacity:0.1;">` `<!-- parent tag→`
- `<h1>10% opacity</h1>` `<!-- child tag→`
- `</div>`

# CSS background-image

# CSS border

- properties allow you to specify the style, width, and color of an element's border.
- 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;  
}
```


he border-width property can have from one to four values (for the top border, right border, bottom border, and the left border):

- ```
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 */
}
```

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)"

HSL - specify a HSL value, like "hsl(0, 100%, 50%)"

transparent

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

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

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

```
} Note: If border-color is not set, it inherits the color of the element.
```

Specific Side Colors

- The border-color property can have from one to four values (for the top border, right border, bottom border, and the left border).
- ```
p.one {
 border-style: solid;
 border-color: red green blue yellow;
 /* red top, green right, blue bottom and yellow left */
}
```

[https://www.w3schools.com/css/css\\_border\\_color.asp](https://www.w3schools.com/css/css_border_color.asp)

# CSS Margins

- Margins are used to create space around elements, outside of any defined borders.
- CSS has properties for specifying the margin for each side of an element:
  - margin-top
  - margin-right
  - margin-bottom
  - margin-left
- 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.

- ```
p {  
  margin-top: 100px;  
  margin-bottom: 100px;  
  margin-right: 150px;  
  margin-left: 80px;  
}
```
- Use the margin shorthand property with four values:
- ```
p {
 margin: 25px 50px 75px 100px;
}
```

# The auto Value

- You can set the margin property to auto to horizontally center the element within its container.
- The element will then take up the specified width, and the remaining space will be split equally between the left and right margins.
- Example
- Use margin: auto:
- ```
div {  
  width: 300px;  
  margin: auto;  
  border: 1px solid red;  
}
```

CSS Padding

- The CSS padding properties are used to generate space around an element's content, inside of any defined borders.
- With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).
- Padding - Individual Sides
- CSS has properties for specifying the padding for each side of an element:
 - padding-top
 - padding-right
 - padding-bottom
 - padding-left
- All the padding properties can have the following values:
 - *length* - specifies a padding in px, pt, cm, etc.
 - % - specifies a padding in % of the width of the containing element
 - inherit - specifies that the padding should be inherited from the parent element
- **Note:** Negative values are not allowed.

Padding - Shorthand Property

- `div {
padding: 25px 50px 75px 100px;
}`

- 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.
- CSS Setting height and width
- The height and width properties are used to set the height and width of an element.
- The height and width properties do not include padding, borders, or margins. It sets the height/width of the area inside the padding, border, and margin of the element.
- CSS height and width Values
- The height and width properties may have the following values:
 - auto - This is default. The browser calculates the height and width
 - length - Defines the height/width in px, cm, etc.
 - % - Defines the height/width in percent of the containing block
 - initial - Sets the height/width to its default value
 - inherit - The height/width will be inherited from its parent value
 -

- Set the height and width of a <div> element:
- ```
div {
 height: 200px;
 width: 50%;
 background-color: powderblue;
}
```
-