CSS

What is CSS?

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

Why use CSS?

- 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.

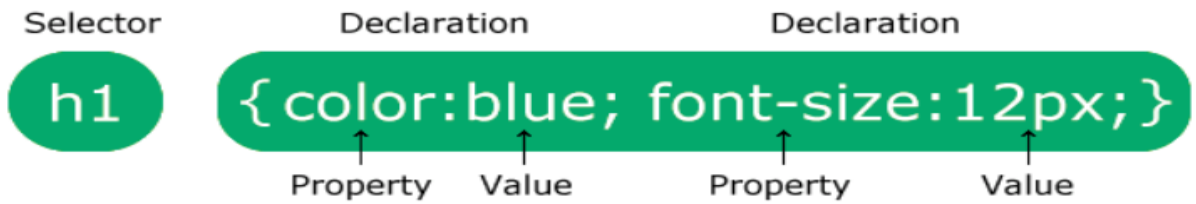
- The style definitions are normally saved in external .css files.

- Example:

```
<head>
<style>
body {
  background-color: yellow;
}
h1 {
  color: red;
  text-align: left;
}
p {
  font-family: times new roman;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>This is CSS Example</h1>
<p>Let's try this</p>
</body>
```

CSS Syntax

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



- selector: the HTML element you want to style.
- 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:

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

<p>Today we will learn how to use CSS.</p>
<p>This is a paragraph.</p>

</body>
</html>
```

- 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

Adding CSS to your webpage

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

- 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.

Example:

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

<h1>This is a heading</h1>

</body>
</html>
```

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.

Here is how the "mystyle.css" file looks:

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

h1 {
  color: navy;
}
```

Internal CSS

An internal style sheet may be used if one single HTML page has a unique style.

Internal styles are defined within the <style> element, inside the <head> section of an HTML page.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: pink;
}
h1 {
  color: purple;
}
</style>
</head>
<body>
<h1>This is a heading</h1>
</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.

Inline styles are defined within the "style" attribute of the relevant element

Example:

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

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

</body>
</html>
```

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:

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

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

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

- If the internal style is defined after the link to the external style sheet, the <h1> elements will be "orange":

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

1. external css

2. internal css

2. internal css will be prioritized over 1. external css

- If the internal style is defined before the link to the external style sheet, the <h1> elements will be "navy":

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

1. internal css

2. external css

2. external css will be prioritized over 1. internal css

Whichever css (external or internal) is defined at the end will be implemented.

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:

1)Inline style (inside an HTML element)

2)External and internal style sheets (in the head section)

3)Browser default

So, an inline style has the highest priority, and will override external and internal styles and browser defaults.

- Example :

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
body {background-color: green;}
</style>
</head>
<body style="background-color: lavender">
<h1>Cascading order</h1>
<p>All the styles in a page will "cascade" into a new "virtual" style sheet by the rules.</p>
</body>
```

EXTERNAL CSS: Added using an external .css link. Added in the <head> element.

INTERNAL CSS: Added in the <style> element.

INLINE CSS: Added along the attribute in the <body> element Has the highest priority..

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:

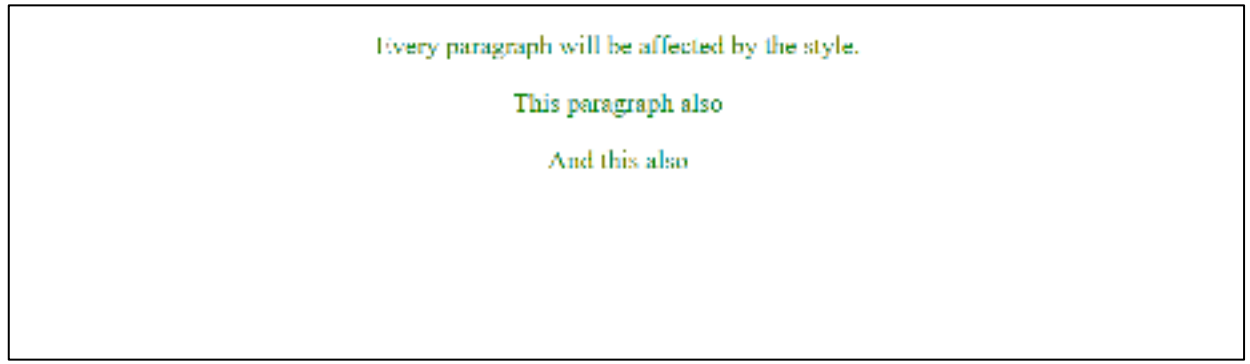
- **Simple selectors** (select elements based on name, id, class)
- **Combinator selectors** (select elements based on a specific relationship between them)
- **Pseudo-class selectors** (select elements based on a certain state)
- **Pseudo-elements selectors** (select and style a part of an element)
- **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:

All <p> elements on the page will be center-aligned, with a green text color

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  text-align: center;
  color: green;
}
</style>
</head>
<body>
<p>Every paragraph will be affected by the style.</p>
<p id="para1">This paragraph also</p>
<p>And this also</p>
</body>
</html>
```

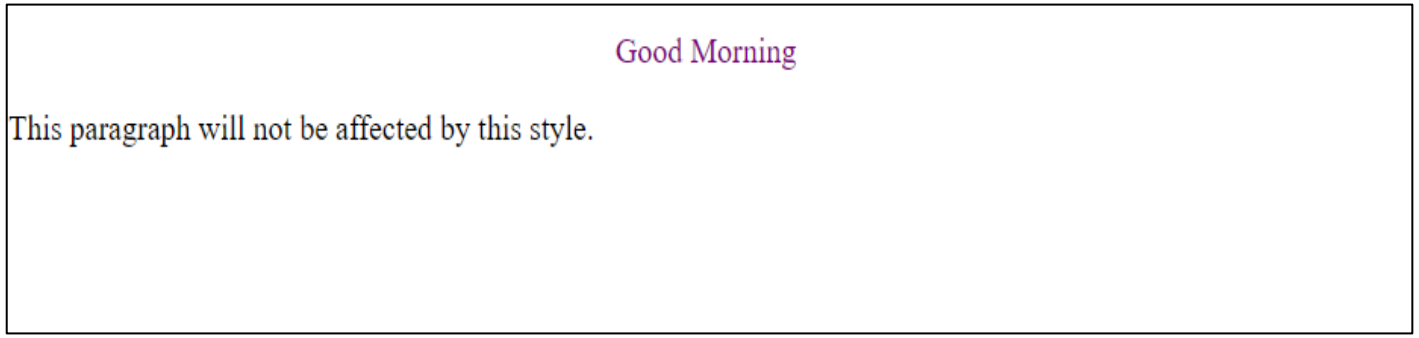


The CSS id Selector

- 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.
- Note: An id name cannot start with a number.
- Example:

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

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
  text-align: center;
  color: purple;
}
</style>
</head>
<body>
<p id="para1">Good Morning</p>
<p>This paragraph will not be affected by this style.</p>
</body>
</html>
```



The 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.
- Note: A class name cannot start with a number.
- Example

In this example, all HTML elements with class="center" will be green and center-aligned:

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
  text-align: center;
  color: green;
}
</style>
</head>
<body>
<h1 class="center">This is a heading</h1>
<p class="center">This is a paragraph</p>
</body>
</html>
```

- You can also specify that only specific HTML elements should be affected by a class.
- Example

In this example only <p> elements with class="center" will be pink and right-aligned:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {
  text-align: right;
  color: pink;
}
</style>
</head>
<body>
<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be pink and right-aligned.</p>
</body>
</html>
```

- HTML elements can also refer to more than one class.
- Example

In this example the <p> element will be styled according to class="center" and to class="large":

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {
  text-align: center;
  color: red;
}
p.large {
  font-size: 200%;
}
</style>
</head>
<body>
<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-aligned.</p>
<p class="center large">This paragraph will be red, center-aligned, and in a large font-size.</p>
</body>
</html>
```

The CSS Universal Selector

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

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

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  text-align: right;
  color: orange;
}
</style>
</head>
<body>
<h1>CSS selector</h1>
<p>Every element on the page will be affected by the style.</p>
<p id="para1">This paragraph will also be affected</p>
<p>This also</p>
</body>
</html>
```

The CSS Grouping Selector

- The grouping selector selects all the HTML elements with the same style definitions.
- It will be better to group the selectors, to minimize the code.
- To group selectors, separate each selector with a comma.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1, h2, p {
  text-align: center;
  color: blue;
}
</style>
</head>
<body>
<h1>Good Morning</h1>
<h2>Heading</h2>
<p>Paragraph.</p>
</body>
</html>
```

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

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

- **CSS text color**

```
<h3 style="color:Tomato;">Good Evening</h3>
```

- **CSS Background color**

```
<h1 style="background-color:DodgerBlue;">Good Evening</h1>
```

- **CSS Border color**

```
<h1 style="border: 2px solid Violet;">Good Evening</h1>
```

- **CSS Color Values**

In CSS, colors can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values

Example:

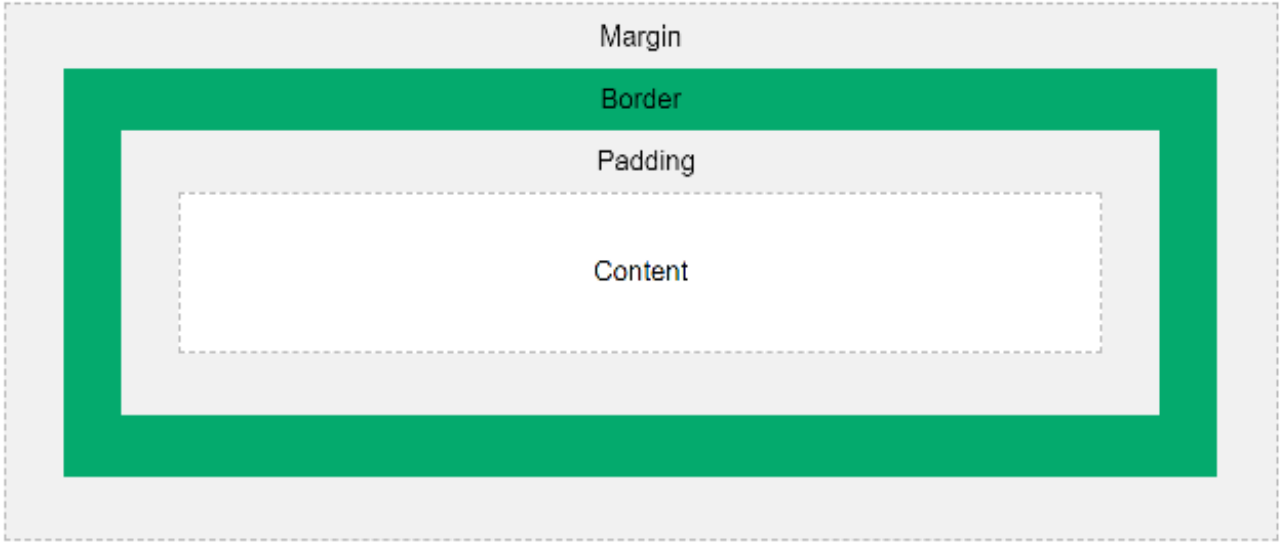
```
<!DOCTYPE html>
<html>
<body>

<p>Same as color name "Tomato":</p>

<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>

<p>Same as color name "Tomato", but 50% transparent:</p>
<h1 style="background-color:rgba(255, 99, 71, 0.5);">rgba(255, 99, 71, 0.5)</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">hsla(9, 100%, 64%, 0.5)</h1>

<p>In addition to the predefined color names, colors can be specified using RGB, HEX, HSL, or even transparent colors using RGBA or HSLA color values.</p>

</body>
</html>
```

CSS Backgrounds

Example:

```
body {
  background-image: url("flower.jpg");
}
```

- background-color
- background-image
- background-repeat
- background-attachment
- background-position
- Background



CSS Box Model

- In CSS, the term "box model" is used when talking about design and layout.
- The box model allows us to add a border around elements, and to define space between elements.
- The CSS box model is essentially a box that wraps around every HTML element. It consists of: content, padding, borders and margins. The image below illustrates the box above:

- Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
</style>
</head>
<body>
<div>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
</div>
</body>
</html>
```

CSS Borders

- border-style
- border-width
- border-color
- border-sides
- rounded border

CSS Margins

- The CSS margin properties 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

- To shorten the code, it is possible to specify all the padding properties in one property.

Example :

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

Example:

```
<html>
<head>
<style>
div {
  border: 1px solid black;
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
  background-color: lightblue;
}
</style>
</head>
<body>
<div>This div element has a top margin of 100px, a right margin of 150px, a bottom margin of 100px, and a left margin of 80px.</
div>
</body>
</html>
```

Follow this for further reference:

https://www.w3schools.com/css/css_margin.asp

CSS Padding

- The CSS padding properties are used to generate space around an element's content, inside of any defined borders.
- CSS has properties for specifying the padding for each side of an element:
  - padding-top
  - padding-right
  - padding-bottom
  - padding-left

- If the padding property has three values:

padding: 25px 50px 75px;

top padding is 25px

right and left paddings are 50px

bottom padding is 75px

- Example

Use the padding shorthand property with three values:

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

Example:

```
<head>
<style>
div {
  border: 1px solid black;
  background-color: lightblue;
  padding-top: 50px;
  padding-right: 30px;
  padding-bottom: 50px;
  padding-left: 80px;
}
</style>
</head>
```

Follow this for further reference:

https://www.w3schools.com/css/css_padding.asp