

CSE 106

Lecture 6 – CSS

Acknowledgement: [w3schools.com](https://www.w3schools.com)

CSS

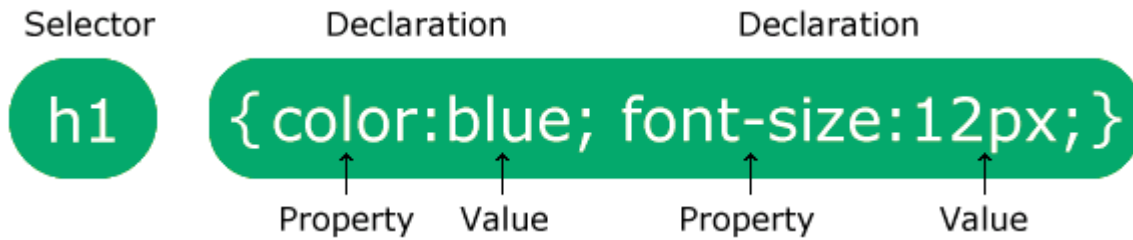
- CSS stands for **C**ascading **S**tyle **S**heets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves work, controlling the layout of multiple web pages at once
- External stylesheets are stored in CSS files

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
- Tags like and color attributes were added to the HTML 3.2
 - Fonts and color information added to every single page
 - Became unmanageable for large web sites
- To solve this problem, the W3C created CSS
- CSS removed the style formatting from the HTML page

CSS Syntax

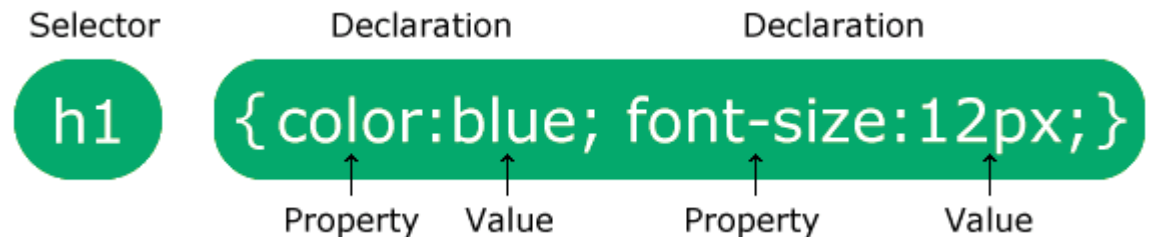
- A CSS rule consists of a selector and a declaration block:



CSS Syntax

- **p** is a selector (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;  
}
```



CSS Selectors – Element Selector

- The element selector selects HTML elements based on element name
- Here, all <p> elements on the page will be center-aligned, with a red text color:

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

CSS Selectors – 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
- Make an id selector with a hash (#) character then the element id
- The CSS is applied to the HTML element with id="greeting":

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

```
<p id="greeting">Hello Everyone!</p>
```

CSS Selectors – class Selector

- Selects HTML elements with a specific class attribute
- Make an id selector with a period (.) character then the class name
- The CSS is applied to all HTML elements with class="center":

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

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


CSS Selectors – class Selector

- Specify specific HTML elements affected by a class
- The CSS is applied to only <p> elements with class="centerme":

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

```
<h1 class="centerme">This heading will not be affected</h1>  
<p class="centerme">Red and center-aligned paragraph.</p>
```

CSS Selectors – class Selector

- HTML elements can also refer to **more than one class**
- The <p> element will be styled according to class="center" and to class="large":

```
p.center {  
    text-align: center;  
    color: red;  
}  
  
p.large {  
    font-size: 300%;  
}
```

```
<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>
```

CSS Selectors – Universal Selector

- The universal selector (*) selects all HTML elements on the page
- The CSS rule below will affect every HTML element on the page:

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

CSS Selectors – Grouping Selector

- Selects all the HTML elements with the same style definitions
- To group selectors, separate each selector with a comma:

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

External CSS

- Can change the look of an entire website by changing just a file (recommended)
- HTML pages include a reference to the CSS file inside <link> element

```
<html>
  <head>
    <link rel="stylesheet" href="mystyle.css">
  </head>
  <body>
    <h1>This is a heading</h1>
  </body>
</html>
```

External CSS

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

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

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

Internal CSS

- May be used if a single HTML page has a unique style
- The internal style is defined inside the <style> element

```
<html>
  <head>
    <style>
      body {
        background-color: linen;
      }
    </style>
  </head>
  <body>
    <h1>This is a heading</h1>
  </body>
</html>
```

Inline CSS

- May be used to apply a unique style for a single element
- Add the style attribute to the relevant element:

```
<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>
```


Cascading Order

- If properties are defined for the same selector (element) in different areas, the value from the last read will be used
- All the styles in a page will cascade into a new virtual style sheet by the following rules:
 - Inline style (inside an HTML element)
 - External and internal style sheets (in the head section)
 - Browser default
- Inline style has the highest priority and overrides external and internal styles and browser defaults

Wake up!

- <https://youtu.be/oY2nVQNIUB8>

CSS Comments

- A CSS comment is placed inside the <style> element, and starts with /* and ends with */:

```
/* This is a single-line comment */  
h1 {  
    color: blue;  
}  
p {  
    color: red;  /* Set text color to red */  
}
```

CSS Colors

- Colors are applied to background, text, borders, etc
- Colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values

```
<h1 style="color:rgb(255, 99, 71);">Hello World</h1>
```

```
<h1 style="color:#ff6347;">Hello World</h1>
```

```
<h1 style="color:hsl(9, 100%, 64%);">Hello World</h1>
```

```
<h1 style="color:rgba(255, 99, 71, 0.5);">...</h1>
```

```
<h1 style="color:hsla(9, 100%, 64%, 0.5);">...</h1>
```

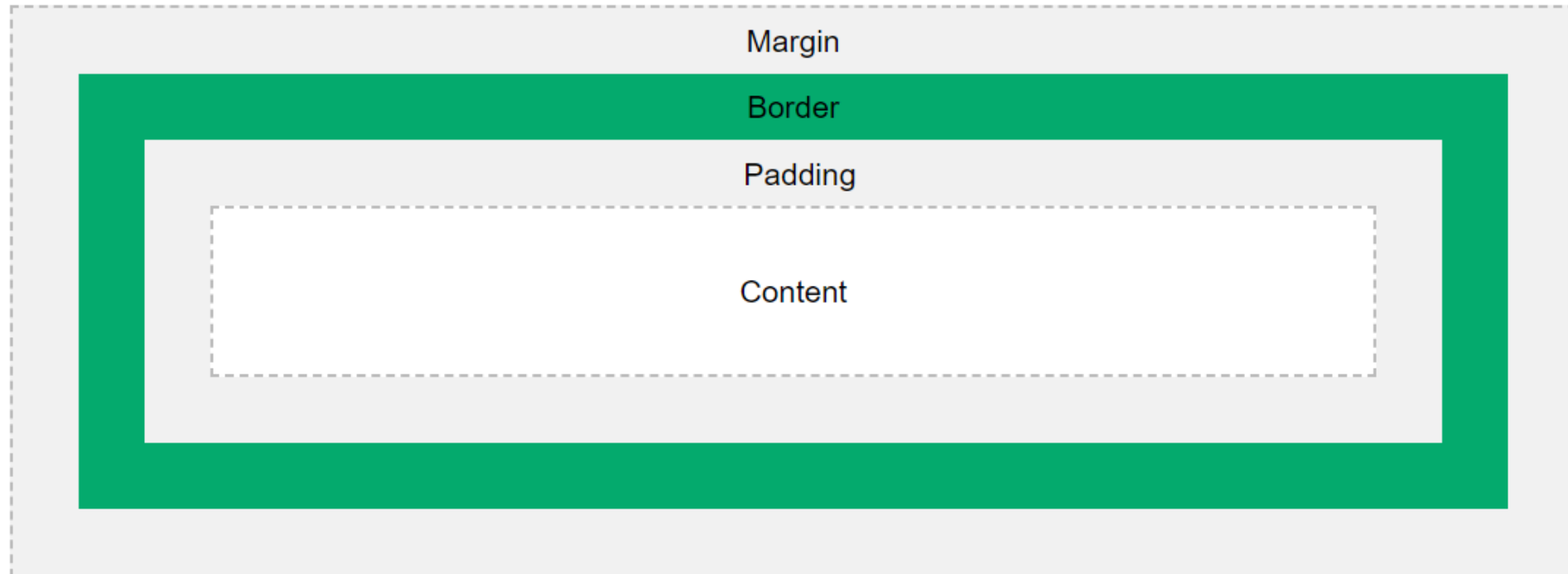
```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
```

```
<h1 style="color:Tomato;">Hello World</h1>
```

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
```

CSS Box Model

- A box that wraps around every HTML element including margins, borders, padding, and the content



CSS Box Model

```
<style>
  div {
    background-color: lightgrey;
    width: 300px;
    border: 5px solid green;
    padding: 40px;
    margin: 20px;
  }
</style>
</head>
<body>
  <div>This text is the content of the box which has border, padding and margin</div>
</body>
```

CSS Width and Height

- Sets the width and height of an element
- Does not include padding, borders, or margins
 - Size of the area inside the padding, border, and margin of the element
- May have the following values:
 - auto - The browser calculates the height and width (default)
 - length - Defines the height/width (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

CSS Width and Height

```
<head>
  <style>
    div {
      height: 200px;
      width: 50%;
      background-color: powderblue;
    }
  </style>
</head>
<body>
  <div>This div element has a height of 200px and a width of 50%.</div>
</body>
```


CSS Width and Height

- max-height/width, min-height/width sets min and maxes with changes screen size

```
<style>
  div {
    max-width: 500px;
    height: 100px;
    background-color: powderblue;
  }
</style>
</head>
<body>
  <div>This div element has a height of 100px and a max-width of 500px.</div>
</body>
```

CSS Layout - The display Property

- The most important CSS property for controlling layout
- Specifies if/how an element is displayed
- **display: none;** is commonly used with JavaScript to hide and show elements – this displays as if the element is not there
- **visibility: hidden;** also hides an element, but it still takes up space

Override The Default Display Value

- You can override an element's default display value
- A common example is making inline elements for horizontal menus

```
<style>

  li {

    display: inline;

  }

</style>

<p>Display a list of links as a horizontal menu:</p>

<ul>

  <li><a href="/html/default.asp" target="_blank">HTML</a></li>

  <li><a href="/css/default.asp" target="_blank">CSS</a></li>

  <li><a href="/js/default.asp" target="_blank">JavaScript</a></li>

</ul>
```

CSS Layout - The position Property

- Specifies the positioning method used for an element
 - **static** - positioned according to the normal flow of the page (default)
 - **relative** - positioned relative to its normal position
 - **fixed** - positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled
 - **absolute** - positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed)
 - **sticky** - positioned based on the user's scroll position

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

CSS Float

- Used for positioning and formatting content
- In its simplest use, the float property can be used to wrap text around images
- The float property can have one of the following values:
 - **left** - The element floats to the left of its container
 - **right** - The element floats to the right of its container
 - **none** - The element does not float (displayed where it occurs in the text). Default.
 - **inherit** - The element inherits the float value of its parent

CSS Float

```
<style>
```

```
  img {
```

```
    float: right;
```

```
  }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
  <p>
```

In this example, the image will float to the right in the paragraph, and the text in the paragraph will wrap around the image.</p>

```
</body>
```

CSS Align

- To horizontally center a block element, use `margin: auto;`
- Setting the width of the element will prevent it from stretching out to the edges of its container
- The element will then take up the specified width, and the remaining space will be split equally between the two margins:

```
.center {  
    margin: auto;  
    width: 50%;  
    border: 3px solid green;  
    padding: 10px;  
}
```

CSS Align Text

- To center the text inside an element, use **text-align: center;**

```
.center {  
    text-align: center;  
    border: 3px solid green;  
}
```

```
<div class="center">  
    <p>This text is centered.</p>  
</div>
```