Cascading Style Sheets
CSS

Asst. Prof. Dr. Özgü Can

CSS

- Language for formatting and designing information on the web, including ASP.NET pages.
- Change the appearance of pages.
- CSS code → to style pages
- CSS was created to address HTML's styling shortcomings.

Problem's of HTML Formatting

- Limited set of options to style pages.
  - Not rich enough to create attractive web pages.
- HTML forces you to embed your formatting in your HTML document, making it harder to reuse or change the design later.

```
<p><font face="Arial" color="red" size="+1">
   This is red text in an Arial type face and slightly larger than the default text.
</font></p>
```

- Additional Markup → increase in the size of the page → slower download & display and harder to maintain pages

CSS

- Define all formatting information in external files.
- With separation;
  - HTML → what you want to display
  - CSS → how you want to display
- Style sheets don't change with each request
  - *Cached copy* → Decrease in bandwidth
  - CTRL+F5 → Get a fresh copy!

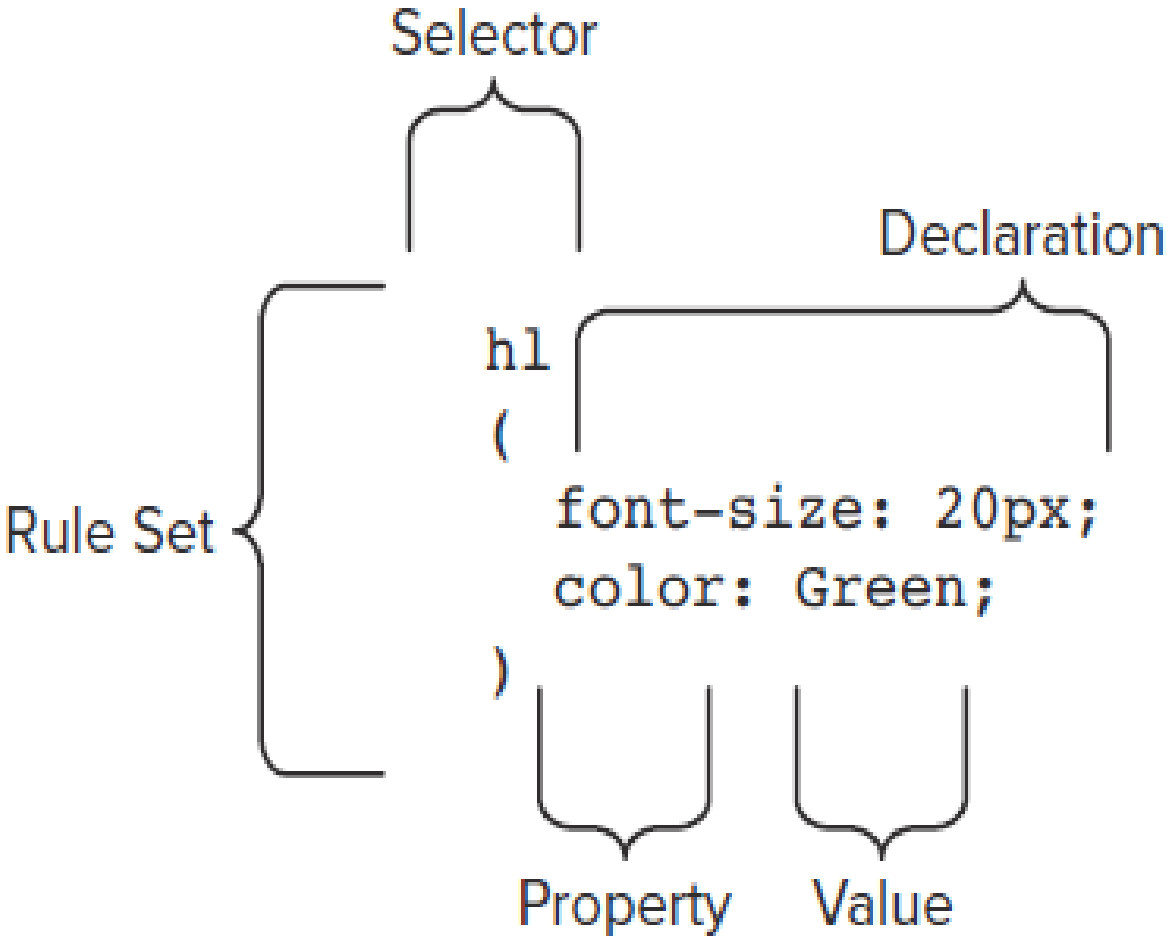
CSS

- Maintained by World Wide Web Consortium (W3C)

- For more information :

  http://www.w3.org/Style/CSS/

CSS


Selector
Declaration
Rule Set
h1
(
font-size: 20px;
color: Green;
)
Property
Value

CSS

- To be able to style an element on a page, a browser has to know three things:
  1. What element of the page must be styled?
     Selectors
  2. What part of that element must be styled?
     Properties
  3. How do you want that part of the selected element to look?

     Values

Selectors

- ***What element of the page must be styled?***
  - The Universal Selector
  - The Type Selector
  - The ID Selector
  - The Class Selector

The Universal Selector

- Indicated by an asterisk (*)
- Applies to all elements in your page
- Can be used to set global settings
  - Font family
- Changing the font for all elements in the page to Arial:

```
*
{
  font-family: Arial;
}
```

The Type Selector

- Points to an HTML element of a specific type

```
h1
{
   color: Green;
}
```

- Not case sensitive!
  - H1 = h1

The ID Selector

- Always prefixed by a hash symbol (#)
- Enables to refer to a single element in the page.
- With the ID selector, you can change the behavior for a unique single element

```
#IntroText
{
  font-style: italic;
}
```

The ID Selector

- Reuse the ID across multiple pages in the site

```
<p id="IntroText">I am italic because I have the right ID.</p>
<p id="BodyText">I am NOT italic because I have a different ID.</p>
```

- ID Selectors are case sensitive!

The Class Selector

- Enables to style multiple HTML elements through the `class` attribute.
- Handy when you want to give the same type of formatting to a number of unrelated HTML elements.

```
.Highlight
{
  font-weight: bold;
  color: Red;
}
```

- Uses a period (.) in its name!

The Class Selector

- Do not use period (.) is when referring to it!

```
This is normal text but <span class="Highlight">this is Red and Bold.</span>
This is also normal text but
   <a href="CssDemo.aspx" class="Highlight">this link is Red and Bold as well.</a>
```

Grouping Selectors

- Enables you to *group* multiple selectors by separating them with a comma.

```
h1, h2, h3, h4, h5, h6
{
  color: Red;
}
```

Combining Selectors

- Enables to target to a specific element in a page.
- Separate the selectors with a space

```
#MainContent p
{
  font-size: 18px;
}
```

Combining Selectors

- Also use it with the other selectors.

```
#MainContent p.Attention
{
  font-weight: bold;
}
```

```
<div id="MainContent">
  <p class="Attention">My class is Attention, so my text is bold.</p>
  <p>My text is not bold, as it lacks the Attention class.</p>
</div>
<p class="Attention">I am NOT bold because I don't fall within MainContent.</p>
```

Properties

- ***What part of that element must be styled?***
- The element that you want to change with your style sheet.
- You don’t have to remember them all.

Values

- ***How do you want that part of the selected element to look?***
- The available values depend on the property.

```
h1
{
  color: Red;
}
```

```
h1
{
   color: #FF0000;
 }
```

```
h1
{
  color: rgb(100%, 0%, 0%);
}
```

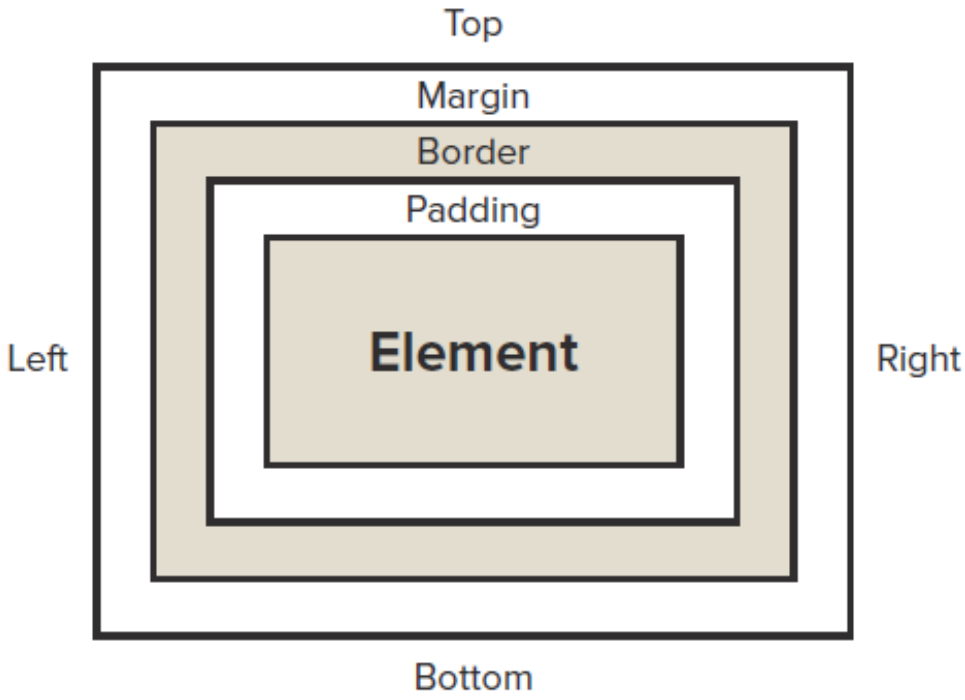
Shorthand

- The border size will be 1px, the style will be solid and the border color will be set to Black.

```
border: 1px solid Black;
```

```
border-top-width: 1px;
border-top-style: solid;
border-top-color: Black;
border-right-width: 1px;
border-right-style: solid;
border-right-color: Black;
border-bottom-width: 1px;
border-bottom-style: solid;
border-bottom-color: Black;
border-left-width: 1px;
border-left-style: solid;
border-left-color: Black;
```

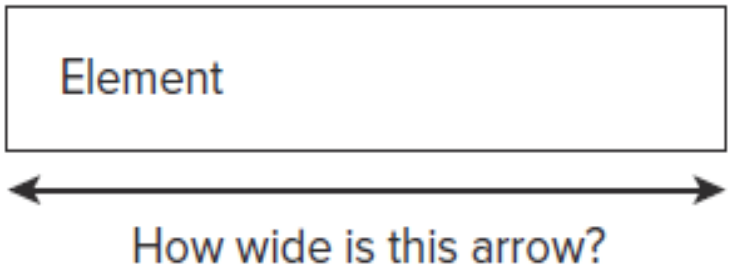
The CSS Box Model



The CSS Box Model

```
.MyDiv
{
  width: 200px;
  padding: 10px;
  border: 2px solid black;
}

...
<div class="MyDiv">Element</div>
```



Adding CSS to Page

- External

```
<link href="StyleSheet.css" rel="Stylesheet" type="text/css" media="screen" />
```

- Embedded

```
<head runat="server">
  <title></title>
  <style type="text/css">
    h1
    {
      color: Blue;
    }
  </style>
</head>
```

- Inline

```
<span style="color: White; background-color: Black;">
    This is white text on a black background.
</span>
```

Overriding

- External over Embedded over Inline

```
h1
{
   color: Green;
}
```

Styles.css

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

Embedded overrides *External*

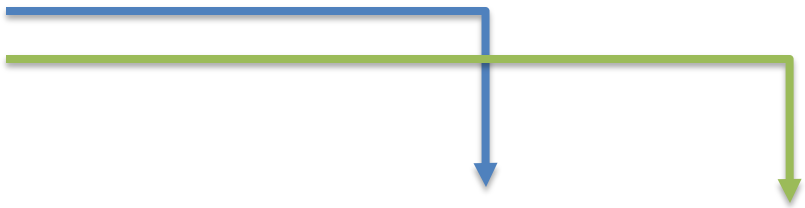
Embedded

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

External overrides *Embedded*

CSS overrules attributes on HTML elements

```
img
{
  height: 100px;
  width: 100px;
}
...
<img src="SomeImage.jpg" width="200px" height"200px" />
```



100px 100px

NOTE

- Try to create smaller and reusable rule sets that you can combine if required, rather than creating large, monolithic rules that can only be used on a single UI element.

```
.ImportantHeading
{
  font-size: 20px;
  font-weight: bold;
  color: red;
}
```

```
h1
{
  font-size: 20px;
}

.Attention
{
  font-weight: bold;
  color: red;
}
```

```
<h1 class="Attention">
```