



Pemrograman Web

3. Cascading Style Sheets (CSS)

M. Udin Harun Al Rasyid, S.Kom, Ph.D
<http://lecturer.eepis-its.edu/~udinharun>
udinharun@eepis-its.edu

Table of Contents

- Introduction to CSS
 - Styles Solved a Big Problem
 - CSS Saves a Lot of Work
 - Inline Styles
 - Internal Stylesheet
 - External Stylesheet
 - CSS Classes
 - CSS ID
 - CSS Divisions
 - CSS Spans
 - CSS Margins
 - CSS Padding
 - CSS Text Properties
 - CSS Font Properties
 - CSS Anchors and Links
 - CSS Borders
 - CSS Lists
 - CSS Table
-

Introduction to CSS

- CSS stands for Cascading Style Sheets
 - A style sheet is made up of style rules that tell a browser how to present a document.
 - External Style Sheets are stored in CSS files.
 - This element is placed in the document HEAD, and it contains the style rules for the page.
-

Styles Solved a Big Problem

- HTML was intended to define the content of a document, like:
 - `<h1>This is a heading</h1>`
 - `<p>This is a paragraph.</p>`
 - When tags like ``, and color attributes were added large web sites, 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 Saves a Lot of Work

- CSS defines HOW HTML elements are to be displayed.
 - Styles are normally saved in external .css files. External style sheets enable you to change the appearance and layout of all the pages in a Web site, just by editing one single file.
-

-
- A CSS (cascading style sheet) file allows you to separate your web sites (X)HTML content from it's style. As always you use your (X)HTML file to arrange the content, but all of the presentation (fonts, colors, background, borders, text formatting, link effects & so on...) are accomplished within a CSS.
 - How to use the CSS:
 - Inline Style (inside (X)HTML element).
 - Internal Style Sheet (inside the <head> tag)
 - External Style Sheet.
-

Inline Styles

- Inline styles are defined right in the (X)HTML file along side the element you want to style. See example below.

```
<p style="color: #ff0000;">Some red text</p>
```

Some red text

Internal Stylesheet

- This way you are simply placing the CSS code within the <head></head> tags of each (X)HTML file you want to style with the CSS.

```
<head>  
<title><title>  
<style type="text/css">  
CSS Content Goes Here  
</style>  
</head>  
<body>
```

- With this method each (X)HTML file contains the CSS code needed to style the page. Meaning that any changes you want to make to one page, will have to be made to all.
 - This method can be good if you need to style only one page, or if you want different pages to have varying styles.
-

External Stylesheet

- A CSS file contains no (X)HTML, only CSS. You simply save it with the .css file extension.
- You can link to the file externally by placing one of the following links in the head section of every (X)HTML file you want to style with the CSS file.

```
<link rel="stylesheet" type="text/css" href="Path To  
stylesheet.css" />
```

- Or you can also use the @import method:

```
<style type="text/css">@import url(Path To stylesheet.css)  
</style>
```

- Either of these methods are achieved by placing one or the other in the head section as shown in example below.

```
<head>  
<title><title>  
<link rel="stylesheet" type="text/css"href="style.css" />  
</head>  
<body>
```

Or

```
<head>  
<title><title>  
<style type="text/css"> @import url(Path To stylesheet.css) </style>  
</head>  
<body>
```

-
- By using an external style sheet, all of your (X)HTML files link to one CSS file in order to style the pages.

 - This means, that if you need to alter the design of all your pages, you only need to edit one .css file to make global changes to your entire website.

 - Here are a few reasons this is better.
 - Easier Maintenance
 - Reduced File Size
 - Reduced Bandwidth
 - Improved Flexibility
-

CSS Syntax

- Consists of only 3 parts.

```
selector { property: value }
```

- The selector is the (X)HTML element that you want to style. The property is the actual property title, and the value is the style you apply to that property.
-

- Each selector can have multiple properties, and each property within that selector can have independent values.
- The property and value are separated with a colon and contained within curly brackets.
- Multiple properties are separated by a semi colon.
- Multiple values within a property are separated by commas, and if an individual value contains more than one word you surround it with quotation marks.

```
body {  
  background: #eeeeee;  
  font-family: "Trebuchet MS", Verdana, Arial, serif;  
}
```

Inheritance

- When you nest one element inside another, the nested element will inherit the properties assigned to the containing element. Unless you modify the inner elements values independently.

```
body {font-family: Verdana, serif;}
```

- If you wanted to style certain text with another font, like an h1 or a paragraph then you could do the following.

```
h1 {font-family: Georgia, sans-serif;}
```

```
p {font-family: Tahoma, serif;}
```

■ Combining Selectors

```
h1, h2, h3, h4, h5, h6 {  
  color: #009900;  
  font-family: Georgia, sans-serif;  
}
```

■ Comment tags

```
/* This is a comment */
```

CSS Classes

- The class selector allows you to style items within the same (X)HTML element differently.

Example:

```
p {  
  font-size: small;  
  color: #333333  
}
```

- I wanted to change the word “sentence” to green bold text, while leaving the rest of the sentence untouched.

```
<p>  
To put it more simply, this <span  
class="greenboldtext">sentence</span> you are reading is  
styled in my CSS file by the following.  
</p>
```

-
- Then in my CSS file you would add this style selector:

```
.greenboldtext{  
  font-size: small;  
  color: #008080;  
  font-weight: bold;  
}
```

- The final result would look like the following:
“To put it more simply, this **sentence** you are reading is styled in my CSS file by the following.”
 - Please note that a class selector begins with a (.) period.
-

CSS ID

- IDs are similar to classes, except once a specific id has been declared it cannot be used again within the same (X)HTML file.
 - I generally use IDs to style the layout elements of a page that will only be needed once, whereas I use classes to style text and such that may be declared multiple times.
-

- In CSS file:

```
#container{  
  width: 80%;  
  margin: auto;  
  padding: 20px;  
  border: 1px solid #666;  
  background: #ffffff;  
}
```

- In the main of page

```
<div id="container">  
Everything within my document is inside this division.  
</div>
```

- An id selector begins with a (#) sign, while the class selector begins with a (.) sign.
-

CSS Divisions

- Divisions are a block level (X)HTML element used to define sections of an (X)HTML file. A division can contain all the parts that make up your website. Including additional divisions, spans, images, text and so on.
- You define a division within an (X)HTML file by placing the following between the <body></body> tags:

```
<div>
```

```
Site contents go here
```

```
</div>
```

```
<div id="container">
```

```
Site contents go here
```

```
</div>
```

-
- The CSS file contains:

```
#container{  
  width: 70%;  
  margin: auto;  
  padding: 20px;  
  border: 1px solid #666;  
  background: #ffffff;  
}
```

- You can use both classes and IDs with a division tag to style sections of your website.
-

CSS Spans

- Spans are very similar to divisions except they are an inline element versus a block level element. No linebreak is created when a span is declared.
- You can use the span tag to style certain areas of text, as shown in the following:

```
<span class="italic">This text is italic</span>
```

- In CSS file:

```
.italic{  
  font-style: italic;  
}
```



CSS Margins

- The margin property declares the margin between an (X)HTML element and the elements around it.
- The margin property can be set for the top, left, right and bottom of an element.

```
margin-top: length percentage or auto;  
margin-left: length percentage or auto;  
margin-right: length percentage or auto;  
margin-bottom: length percentage or auto;
```

```
margin: 10px 10px 10px 10px;
```

```
margin: 10px 10px; /* 2 values */  
margin: 10px 10px 10px; /* 3 values */
```

```
body{  
  margin: 20px;  
  background: #eeeeee;  
  font-size: small;  
  font-family: Tahoma, Arial, "Trebuchet MS", Helvetica,  
  sans-serif;  
  text-align: left;  
}
```

CSS Padding

- Padding is the distance between the border of an (X)HTML element and the content within it.

```
padding-top: length percentage;  
padding-left: length percentage;  
padding-right: length percentage;  
padding-bottom: length percentage;
```

```
<html>  
<head>  
<style type="text/css">  
p  
{  
background-color:yellow;  
}  
p.padding  
{  
padding-top:25px;  
padding-bottom:25px;  
padding-right:50px;  
padding-left:50px;  
}  
</style>  
</head>  
  
<body>  
<p>This is a paragraph with no specified padding.</p>  
<p class="padding">This is a paragraph with specified paddings.</p>  
</body>  
</html>
```

```
#container{  
width: 70%;  
margin: auto;  
padding: 30px;  
border: 1px solid #666;  
background: #ffffff;  
}
```

```
padding: 10px 10px 10px 10px;
```

CSS Text Properties

Color

- You can set the color of text with the following:

```
color: value;
```

- Possible values are
 - color name – example:(red, black...)
 - hexadecimal number – example:(#ff0000, #000000)
 - RGB color code – example:(rgb(255, 0, 0), rgb(0, 0, 0))
-

- **Letter Spacing.**

You can adjust the space between letters in the following manner. Setting the value to 0, prevents the text from justifying.

```
letter-spacing: value;
```

Possible values are: normal, length.

- **Text Align**

```
text-align: value;
```

Possible values are left, right, center, justify.

- **Text Decoration**

```
text-decoration: value;
```

Possible values are: none, underline, overline, line through, blink.

- **Text Transform**

`text-transform: value;`

Possible values are: none, capitalize, lowercase, uppercase.

- **Word Spacing**

You can adjust the space between words in the following manner.

`word-spacing: value;`

Possible values are: normal, length.

CSS Font Properties

- font: italic bold normal small/1.4em Verdana, sans-serif;
 - font-family: Verdana, sans-serif;
 - font-size: value;
Possible values: large, medium, small, etc.
 - font-style: value;
Possible values: normal, italic, oblique.
 - font-variant: value;
Possible values: normal, small-caps.
 - font-weight: value;
Possible values: lighter, normal, 100, 200, etc.
-

CSS Anchors and Links

```
a:link {color: #009900;}  
a:visited {color: #999999;}  
a:hover {color: #333333;}  
a:focus {color: #333333;}  
a:active {color: #009900;}
```

CSS Borders

- You can set the color, style and width of the borders around an element in one declaration by using the border property.
 - Example:
 - `border: 1px solid #333333;`
 - `border-color: value;`
 - `border-style: value; [dashed, dotted, etc].`
 - `border-width: value; [Length , thin, medium, etc].`
 - `border-bottom: 1px solid #333333;`
 - `border-left: 1px solid #333333;`
 - `border-left-color: value;`
 - Etc.
-

CSS Lists

- You can control the appearance of ordered and unordered lists in one declaration with the list-style property.
 - list-style: value value;
Values: image, position, type.
 - list-style-image: url(path_to_image.gif, jpg or png);
 - list-style-position: value;
Values: inside, outside.
 - list-style-type: value;
Values: disc, circle, square.
-

CSS Table

- To specify table borders in CSS, use the border property.

- Example:

```
table, th, td
```

```
{
```

```
border: 1px solid black;
```

```
}
```

Finish

