

CSS3 PART2

Juha Söderqvist

HTML5, CSS3, JS COMMENTS

`/*.....this is a comment in style sheet.....*/`

`<!-- html comment -->`

`// in javascript`

`<head>`

`<!-- html comment -->`

`<script> // in javascript </script>`

`<style>`

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

`/* CSS by Juha */`

`p {`

`color: #111111;`

`}`

`/* This is`

`a multi-line`

`comment */`

`</style>`

CSS3 SELECTOR

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts

1. Selector – A selector is an HTML tag at which a style will be applied. This could be any tag like `<h1>` or `<table>` etc.
2. Property - A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border etc.
3. Value - Values are assigned to properties. For example, color property can have value either red or `#F1F1F1` etc.

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

CSS3 TYPE SELECTOR

selector { property: value }

```
table{ border :1px solid #C00000; }
```

Here table is a selector and border is a property and given value 1px solid #C00 is the value of that property.

example to give a color to all level 1 headings

```
h1 {  
  color: #36CFFF;  
}
```

CSS3 UNIVERSAL SELECTOR

Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type –

```
* {  
  color: #000000;  
}
```

This rule renders the content of every element in our document in black.

CSS3 DESCENDANT SELECTOR

Suppose you want to apply a style rule to a particular element only when it lies inside a particular element. As given in the following example, style rule will apply to `` element only when it lies inside `` tag.

```
ul em {  
  color: #000000;  
}
```

CSS3 CLASS SELECTOR

You can define style rules based on the class attribute of the elements. All the elements having that class will be formatted according to the defined rule.

```
.black {  
  color: #000000;  
}
```

This rule renders the content in black for every element with class attribute set to black in our document. You can make it a bit more particular. For example:

```
h1.black {  
  color: #000000;  
}
```

This rule renders the content in black for only `<h1>` elements with class attribute set to black.

You can apply more than one class selectors to given element. Consider the following example:

```
<p class="center bold">
```

This para will be styled by the classes center and bold.

```
</p>
```

CSS3 ID SELECTOR

You can define style rules based on the id attribute of the elements. All the elements having that id will be formatted according to the defined rule.

```
#black {  
  color: #000000;  
}
```

This rule renders the content in black for every element with id attribute set to black in our document. You can make it a bit more particular. For example –

```
h1#black {  
  color: #000000;  
}
```

This rule renders the content in black for only <h1> elements with id attribute set to black.

The true power of id selectors is when they are used as the foundation for descendant selectors, For example:

```
#black h2 {  
  color: #000000;  
}
```

CSS3 CHILD SELECTOR

You have seen the descendant selectors. There is one more type of selector, which is very similar to descendants but have different functionality. Consider the following example –

```
body > p {  
  color: #000000;  
}
```

This rule will render all the paragraphs in black if they are direct child of <body> element. Other paragraphs put inside other elements like <div> or <td> would not have any effect of this rule.

CSS3 ATTRIBUTE SELECTOR

You can also apply styles to HTML elements with particular attributes. The style rule below will match all the input elements having a type attribute with a value of text –

```
input[type = "text"]{  
    color: #000000;  
}
```

The advantage to this method is that the `<input type = "submit" />` element is unaffected, and the color applied only to the desired text fields.

There are following rules applied to attribute selector.

`p[lang]` - Selects all paragraph elements with a lang attribute.

`p[lang="fr"]` - Selects all paragraph elements whose lang attribute has a value of exactly "fr".

`p[lang~="fr"]` - Selects all paragraph elements whose lang attribute contains the word "fr".

`p[lang|= "en"]` - Selects all paragraph elements whose lang attribute contains values that are exactly "en", or begin with "en-".

CSS3 MULTIPLE STYLE RULES

define multiple style rules for a single element. You can define these rules to combine multiple properties and corresponding values into a single block as defined in the following example –

```
h1 {  
  color: #36C;  
  font-weight: normal;  
  letter-spacing: .4em;  
  margin-bottom: 1em;  
  text-transform: lowercase;  
}
```

Here all the property and value pairs are separated by a semi colon (;). You can keep them in a single line or multiple lines. For better readability we keep them into separate lines.

CSS3 GROUPING SELECTORS

You can apply a style to many selectors if you like. Just separate the selectors with a comma, as given in the following example –

```
h1, h2, h3 {  
  color: #36C;  
  font-weight: normal;  
  letter-spacing: .4em;  
  margin-bottom: 1em;  
  text-transform: lowercase;  
}
```

This define style rule will be applicable to h1, h2 and h3 element as well. The order of the list is irrelevant. All the elements in the selector will have the corresponding declarations applied to them.

You can combine the various class selectors together as shown below –

```
#content, #footer, #supplement {  
  position: absolute;  
  left: 510px;  
  width: 200px;  
}
```

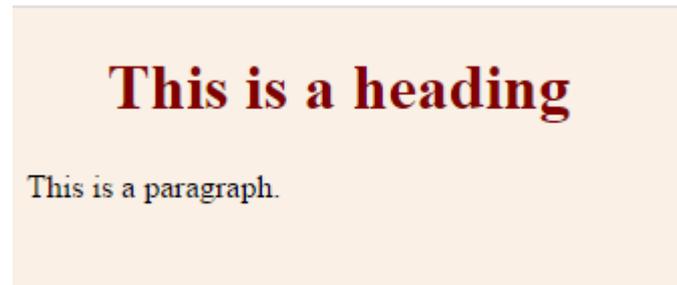
CSS3 INCLUSION

There are four ways to associate styles with your HTML document. Most commonly used methods are **inline** CSS and **External** CSS.

Embedded CSS - The <style> Element

You can put your CSS rules into an HTML document using the <style> element. This tag is placed inside <head>...</head> tags. Rules defined using this syntax will be applied to all the elements available in the document.

```
<!DOCTYPE html>
<html>
  <head>
    <style type = "text/css" media="all">
      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>
```



CSS3 TYPE MEDIA DEVICES

`type` `media_type` Specifies the media type of the linked document

`media` `media_query` Specifies on what device the linked document will be displayed

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

```
<link rel="stylesheet" type="text/css" href="demo_print.css" media="print">
```

If you print this page, or open it in print preview, you will see that it is styled with the `media="print"` stylesheet. The "print" stylesheet contains black text on white background.

Value	Description
-------	-------------

<code>all</code>	Default. Suitable for all devices
------------------	-----------------------------------

<code>aural</code>	Speech synthesizers
--------------------	---------------------

<code>braille</code>	Braille feedback devices
----------------------	--------------------------

<code>handheld</code>	Handheld devices (small screen, limited bandwidth)
-----------------------	--

<code>projection</code>	Projectors
-------------------------	------------

<code>print</code>	Print preview mode/printed pages
--------------------	----------------------------------

<code>screen</code>	Computer screens
---------------------	------------------

<code>tty</code>	Teletypes and similar media using a fixed-pitch character grid
------------------	--

<code>tv</code>	Television type devices (low resolution, limited scroll ability)
-----------------	--

CSS3 ATTRIBUTES STYLE

Attributes associated with <style> elements are:

1. Attribute
2. Value
3. Description

Attribute: type

Values: text/css

Description: Specifies the style sheet language as a content-type (MIME type). This is required attribute.

Attribute: media

Values:

Screen

Tty

tv

projection

handheld

print

braille

aural

All

Description: Specifies the device the document will be displayed on. Default value is all. This is an optional attribute.

CSS3 INLINE CSS

```
<html>  
  <head>  
  </head>  
  <body>  
    <h1 style = "color:#36C;"> This is inline CSS </h1>  
  </body>  
</html>
```

CSS3 EXTERNAL CSS

The <link> element can be used to include an external stylesheet file in your HTML document.

An external style sheet is a separate text file with .css extension. You define all the Style rules within this text file and then you can include this file in any HTML document using <link> element.

Examples external CSS file

```
<head>
```

```
<link rel="stylesheet" href="css/w3.css">
```

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

```
</head>
```

```
-----
```

CSS3 MULTIPLE CSS

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.

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>
```

CSS3 CASCADING ORDER

What style will be used when there is more than one style specified for an HTML element?

Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

Inline style (inside an HTML element)

External and internal style sheets (in the head section)

Browser default

So, an inline style (inside a specific HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or a browser default value.

```
<link rel="stylesheet" type="text/css" href="mystyle.css"> a. external CSS
```

```
<style>
```

```
body {background-color: linen;} b. Internal CSS
```

```
</style>
```

```
</head>
```

```
<body style="background-color: lightcyan"> c. Inline CSS
```

CSS3 IMPORTED CSS

@import is used to import an external stylesheet in a manner similar to the <link> element. Here is the generic syntax of @import rule.

url

Is a <string> or a <uri> representing the location of the resource to import. The URL may be absolute or relative.

```
<head>  
  @import "mystyle.css";  
</head>
```

```
@import url("fingerprint.css") print;  
@import url("bluish.css") projection, tv;  
@import 'custom.css';  
@import url("chrome://communicator/skin/");  
@import "common.css" screen, projection;
```

CSS3 COMPLEXITY CSS

```
@import url(http://css1.style.org/example.css);
```

```
BODY {background-image: url(hatch.gif);}
```

NOTE: relative URLs are relative to the style sheet

1. External CSS

```
<LINK REL="stylesheet" TYPE="text/css"  
  HREF="http://www.waffles.org/styles/basic.css">
```

Inside the file `basic.css` is an `@import` statement referring to another style sheet:

```
@import url(special/toppings.css);
```

This `@import` will load the file `http://www.waffles.org/styles/special/toppings.css`, not `http://www.waffles.org/toppings/special/toppings.css`. If you have a style sheet at the latter location, then the `@import` in `basic.css` should read:

```
@import url(http://www.waffles.org/toppings/special/toppings.css);
```

In other words, use an absolute URL

CSS3 VS SASS AND LESS

SCSS is a special type of file for SASS, a program written in Ruby that assembles CSS style sheets for a browser. SASS adds lots of additional functionality to CSS like variables, nesting and more which can make writing CSS easier and faster

Sass is a scripting language that is interpreted into Cascading Style Sheets (CSS)

Less is a CSS pre-processor, meaning that it extends the CSS language, adding features that allow variables, mixins, functions and many other techniques that allow you to make CSS that is more maintainable, themable and extendable.

Less (sometimes stylized as LESS) is a dynamic style sheet language that can be compiled into Cascading Style Sheets (CSS) and run on the client side or server side

