

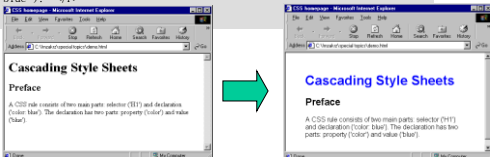
## Introduction to CSS (CSS2)

### Cascading Style Sheets

- A **style sheet** is a collection of formatting rules that can be applied to multiple HTML documents
- it acts as a template, allows the same look for each occurrence of a particular element;
- it can be contained in a separate file, referenced with a <LINK REL="stylesheet"> tag, or embedded within an HTML document by using a <STYLE> tag

### CSS Example

```
<HTML>
<HEAD>
<TITLE>CSS homepage</TITLE>
<STYLE type="text/css">
H1 { color: blue; }
BODY { font-family: "Gill Sans", sans-serif;
font-size: 12pt;
margin: 5em }
</STYLE>
</HEAD>
<BODY>
<H1>Cascading Style Sheets</H1>
<H2>Preface</H2>
<P> A CSS rule consists of two main parts: selector
('H1') and declaration ('color: blue'). The
declaration has two parts: property ('color') and
value ('blue'). </P>
</BODY>
</HTML>
```



The image shows a code editor window with the HTML and CSS code above. Below the code, there are two browser screenshots. The left screenshot shows the rendered page with the title 'Cascading Style Sheets' in blue and a paragraph of text. A green arrow points from the code editor to the browser window.

### CSS Rules

- A CSS rule consists of two main parts: **selector** (e.g. 'H1') and **declaration** (e.g. 'color: blue'); the declaration has two parts: property (e.g. 'color') and value (e.g. 'blue')
- A **selector** is a chain of one or more simple selectors separated by combinators; **combinators** are: whitespace, ">", and "+"; whitespace may appear between a combinator and the simple selectors around it.
- If all conditions in the selector are true for a certain element, the selector *matches* the element

## Selector Grouping

- When several selectors share the same declarations, they may be **grouped** into a comma-separated list; example:
  - we condense three rules with identical declarations into one:

```
H1 { font-family: sans-serif }
H2 { font-family: sans-serif }
H3 { font-family: sans-serif }
```
  - is equivalent to:

```
H1, H2, H3 { font-family: sans-serif }
```

## Descendant Selector

- To match an element that is the descendant of another element in the document tree; made up of two or more selectors separated by whitespace; a descendant selector of the form "A B" matches when an element B is an arbitrary descendant of some ancestor element A; e.g.:

*consider the following rules:*

```
H1 { color: red }
EM { color: red }
```

*although the intention of these rules is to add emphasis to text by changing its color, the effect will be lost in a case such as:*

```
<H1>This headline is <EM>very</EM> important</H1>
```

*we address this case by supplementing the previous rules with a rule that sets the text color to blue whenever an EM occurs anywhere within an H1:*

```
H1 { color: red }
EM { color: red }
H1 EM { color: blue }
```

## Universal Selector and Type Selector

- The **universal selector**, written "\*", matches the name of any element type; it matches any single element in the document tree
- If the universal selector is not the only component of a simple selector, the "\*" may be omitted; for example:
  - \*[LANG=fr] and [LANG=fr] are equivalent
  - \*.warning and .warning are equivalent
  - \*#myid and #myid are equivalent
- A **type selector** matches the name of a document language element type; a type selector matches every instance of the element type in the document tree; e.g. the following rule matches all H1 elements in the document tree:

```
H1 { font-family: sans-serif }
```

## Child Selector and Adjacent Sibling Selector

- A **child selector** matches when an element is the child of some element; a child selector is made up of two or more selectors separated by ">"; e.g. the following rule sets the style of all P elements that are children of BODY:

```
BODY > P { line-height: 1.3 }
```
- **Adjacent sibling selectors** have the syntax: E1 + E2, where E2 is the subject of the selector; matches if E1 and E2 share the same parent in the document tree and E1 immediately precedes E2, e.g. the following rule states that when a P element immediately follows a MATH element, it should not be indented:

```
MATH + P { text-indent: 0 }
```

## Attribute Selectors

- Attribute selectors allows authors to specify rules that match attributes defined in the source document; may match in four ways:
  - **[att]** match when the element sets the "att" attribute, whatever the value of the attribute
  - **[att=val]** match when the element's "att" attribute value is exactly "val"
  - **[att~val]** match when the element's "att" attribute value is a space-separated list of "words", one of which is exactly "val"
  - **[att|=val]** match when the element's "att" attribute value is a hyphen-separated list of "words", beginning with "val"

*Example, the selector matches all P elements whose "align" attribute has exactly the value "center":*

```
P[align=center] { color: blue }
```

## ID Selectors

- For style sheets used with HTML, authors may use the (#) notation as an alternative to the "=" notation when matching on the "id" attribute; e.g. the following ID selector matches the H1 element whose ID attribute has the value "chapter1":
  - H1#chapter1 { text-align: center }**
  - the following ID selector matches the **any** element whose ID attribute has the value "chapter1":
    - \*#chapter1 { text-align: center }**

## Class Selectors

- For style sheets used with HTML, authors may use the dot (.) notation as an alternative to the "~=" notation when matching on the "class" attribute; e.g. the following assigns style to H1 elements with class~="pastoral":

```
H1.pastoral { color: green }
```

- given these rules, the first H1 instance below would not have green text, while the second would:

```
<H1>Not green</H1>  
<H1 class="pastoral">Very green</H1>
```

- to match a subset of "class" values, each value must be preceded by a ".", in any order; e.g. the following rule matches any P element whose "class" attribute has been assigned a list of space-separated values that includes "pastoral" and "marine":

```
P.pastoral.marine { color: green }
```

(This rule matches when class="pastoral blue aqua marine" but does not match for class="pastoral blue")

## Pseudo Class Selectors - :first-child

- The **:first-child** pseudo-class matches an element that is the first child of some other element; e.g. in the following example, the selector matches any P element that is the first child of a DIV element (indentation for the first paragraph of a DIV):

```
DIV > P:first-child { text-indent: 0 }
```

*this selector would match the P inside the DIV of the following fragment:*

```
<P> The last P before the note.  
<DIV class="note">  
<P> The first P inside the note.  
</DIV>
```

*but would not match the second P in the following fragment:*

```
<P> The last P before the note.  
<DIV class="note">  
<H2>Note</H2>  
<P> The first P inside the note.  
</DIV>
```

## Pseudo Class Selectors - :link and :visited

- The **:link** pseudo-class applies for links that have not yet been visited
- The **:visited** pseudo-class applies once the link has been visited by the user

– The document language determines which elements are hyperlink source anchors; e.g. in HTML 4.0, the link pseudo-classes apply to A elements with an "href" attribute; thus, the following two CSS2 declarations have similar effect:

```
A:link { color: red }  
:link { color: red }
```

E.g. if the following link:

```
<A class="external" href="http://out.side/">external link</A>
```

has been visited, this rule:

```
A.external:visited { color: blue }
```

will cause it to be blue

## Pseudo Class Selectors - :hover, :active, :focus

- The **:hover** pseudo-class applies while the mouse pointer hovers over a box generated by the element
- The **:active** pseudo-class applies while an element is being activated by the user; e.g. between the times the user presses the mouse button and releases it
- The **:focus** pseudo-class applies while an element has the focus
- Examples:

```
A:hover { color: yellow }
```

```
A:active { color: lime }
```

```
A:focus { background: yellow }
```

```
A:focus:hover { background: white }
```

## Pseudo Element Selectors - :first-line and :first-letter

- The **:first-line** pseudo-element applies special styles to the first formatted line of a paragraph; e.g.:

```
P:first-line { text-transform: uppercase }
```

- The **:first-letter** pseudo-element may be used for "initial caps" and "drop caps", which are common typographical effects; e.g.:

```
P:first-letter { font-size: 200%; font-weight: bold; float: left }
```

**T**HE FIRST few words of an article in the Economist

## Pseudo Element Selectors - :before and :after

- The **:before** and **:after** pseudo-elements can be used to insert generated content before or after an element's content; e.g.:
- When the **:first-letter** and **:first-line** pseudo-elements are combined with **:before** and **:after**, they apply to the first letter or line of the element including the inserted text; e.g.:

```
P.special:before {content: "Special!" }
```

```
P.special:first-letter {color: #ffd800 }
```

*(This will render the "S" of "Special!" in gold)*

*Other examples:*

```
– H1:before { content: "Chapter " counter(chapter) " ";  
  counter-increment: chapter; /* Add 1 to chapter */  
  counter-reset: section; /* Set section to 0 */ }
```

```
– P.note:before { content: "Note: " }
```

## Selector Matching Summary

- \* matches any element
- E matches any E element (i.e., an element of type E)
- E F matches any F element that is a descendant of an E element
- E > F matches any F element that is a child of an element E
- E:first-child matches element E when E is the first child of its parent
- E:link and E:visited matches element E if E is the source anchor of a hyperlink of which the target is not yet visited (:link) or already visited (:visited)
- E:active, E:hover, and E:focus matches E during certain user actions
- E:lang(c) matches element of type E if it is in (human) language c (the document language specifies how language is determined)
- E + F matches any F element immediately preceded by an element E
- E[foo] matches any E element with the "foo" attribute set (whatever the value)
- E[foo="warning"] matches any E element whose "foo" attribute value is exactly equal to "warning"
- E[foo~="warning"] matches any E element whose "foo" attribute value is a list of space-separated values, one of which is exactly equal to "warning"
- E[lang]="en" matches any E element whose "lang" attribute has a hyphen-separated list of values beginning (from the left) with "en".
- DIV.warning is the same as DIV[class~="warning"].
- E#myid matches any E element ID equal to "myid".

## Calculating a Selector's Specificity

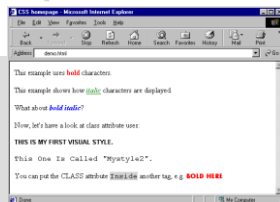
- When many rules match the same element, a selector's specificity is calculated as follows:
  - count the number of ID attributes in the selector (= a)
  - count the number of other attributes and pseudo-classes in the selector (= b)
  - count the number of element names in the selector (= c)
  - ignore pseudo-elements.

concatenating the three numbers a-b-c gives the specificity; the rule with maximal specificity wins; examples:

```
* {} /* a=0 b=0 c=0 -> specificity = 0 */
LI {} /* a=0 b=0 c=1 -> specificity = 1 */
UL LI {} /* a=0 b=0 c=2 -> specificity = 2 */
UL OL+LI {} /* a=0 b=0 c=3 -> specificity = 3 */
H1 + *[REL=up] {} /* a=0 b=1 c=1 -> specificity = 11 */
UL OL LI.red {} /* a=0 b=1 c=3 -> specificity = 13 */
LI.red.level {} /* a=0 b=2 c=1 -> specificity = 21 */
#x34y {} /* a=1 b=0 c=0 -> specificity = 100 */
```

## CSS Example

```
<HTML>
<HEAD>
<TITLE>CSS homepage</TITLE>
<LINK REL="stylesheet" HREF="demo.css">
</HEAD>
<BODY>
<P>This example uses <B>bold</B> characters.</P>
<P>This example shows how <I>italic</I>
characters are displayed.</P>
<P>What about <B><I>bold italic</I></B>?</P>
<P>Now, let's have a look at class attribute uses:</P>
<P CLASS="mystyle1">This is my first visual style.</P>
<P CLASS="mystyle2">This one is called "mystyle2".</P>
<P>You can put the CLASS attribute <SPAN CLASS="mystyle2">inside</SPAN> another tag, e.g.
<B CLASS="mystyle1">bold here</B></P>
</BODY>
</HTML>
```



DEMO.CSS file

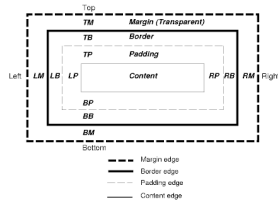
```
B {color: red;}
I {text-decoration: underline; color: green;}
B I {text-decoration: none; color: blue;}
*.mystyle1 {font-family: sans-serif; font-variant: small-caps; font-weight: bolder;}
*.mystyle2 {font-family: monospace; text-transform: capitalize;}
SPAN.mystyle2 {background: lightgrey;}
```

## CSS Descriptors

- Box model
- Visual formatting model
- Visual effects
- Generated content, automatic numbering, lists
- Colors and backgrounds
- Fonts
- Text
- Tables
- User interface

## CSS Box Model

- The CSS **box model** describes the rectangular boxes that are generated for elements in the document tree and laid out according to the **visual formatting model**
- Each box has a **content area** (e.g., text, an image, etc.) and optional surrounding **padding, border, and margin areas**



## CSS Box Model: Margins

- Margin properties: **margin-top, margin-right, margin-bottom, margin-left, margin;** example:

```
BODY { margin: 2em } /* all margins set to 2em */
BODY { margin: 1em 2em } /* top & bottom = 1em, right & left = 2em */
BODY { margin: 1em 2em 3em } /* top=1em, right=2em, bottom=3em, left=2em */
```

the last rule of the example above is equivalent to the example below:

```
BODY {
  margin-top: 1em;
  margin-right: 2em;
  margin-bottom: 3em;
  margin-left: 2em;
}
```

**Note:** '1em' is equal to the size of the font in use

## CSS Box Model: Padding

- Padding properties: **padding-top, padding-right, padding-bottom, padding-left, and padding;** example:

```
H1 {
  background: white;
  padding: 1em 2em;
}
```

The example above specifies a '1em' vertical padding (padding-top and padding-bottom) and a '2em' horizontal padding (padding-right and padding-left)

## CSS Box Model: Border

- Border width properties: **border-top-width, border-right-width, border-bottom-width, border-left-width, and border-width;** values: **thin, medium, thick,** or explicit thickness value; e.g. H1 {border-width: thin}
- Border color properties: **border-top-color, border-right-color, border-bottom-color, border-left-color, and border-color;** e.g. H1 {border-color: red}
- Border style properties: **border-top-style, border-right-style, border-bottom-style, border-left-style, and border-style;** values: **none, dotted, dashed, solid, double, groove, ridge, inset, outset;** e.g. H1 {border-style: solid dotted}

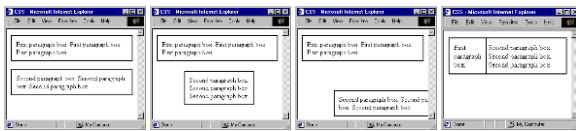
## Box Positioning Schemes

- **Normal flow**
- **Floats** - in the float model, a box is first laid out according to the normal flow, then taken out of the flow and shifted to the left or right as far as possible; content may flow along the side of a float.
- **Absolute positioning** - in the absolute positioning model, a box is removed from the normal flow entirely (it has no impact on later siblings) and assigned a position with respect to a containing block

## Choosing a Positioning Scheme

- Box positioning property: **position**; values: **relative**, **absolute**
- Box offset property: **top**, **right**, **bottom**, and **left**
- Float positioning property: **float**; values: **left**, **right**, and **none**
- Controlling flow next to floats: **clear** property; values: **left**, **right**, **both**, and **none**

## Positioning Schemes: Example



```

P {border-width: thin; border-style: solid; padding: 10}
P {border-width: thin; border-style: solid; padding: 10}
P:two
  {position: absolute; left: 75; top: 100}
P {border-width: thin; border-style: solid; padding: 10}
P:two
  {position: relative; left: 75; top: 50}
P {border-width: thin; border-style: solid; padding: 10}
P:one
  {float: left; width: 80 }
  
```

```

<P class=one> First paragraph box. First paragraph box. First
paragraph box </P>
<P class=two> Second paragraph box. Second paragraph box. Second
paragraph box </P>
  
```

## Font Specification

- **font-family** property; values: **serif**, **sans-serif**, **cursive**, **fantasy**, **monospace**, or any other font family name
- **font-style** property; values: **normal**, **italic**, **oblique**
- **font-variant** property; values: **normal**, **small-caps**
- **font-weight** property; values: **normal**, **bold**, **bolder**, **lighter**, **100**, **200**, **300**, **400**, **500**, **600**, **700**, **800**, **900**
- **font-stretch** property; values: **ultra-condensed**, **extra-condensed**, **condensed**, **semi-condensed**, **normal**, **semi-expanded**, **expanded**, **extra-expanded**, **ultra-expanded**
- **font-size** property; values: **xx-small**, **x-small**, **small**, **medium**, **large**, **x-large**, **xx-large**, **larger**, **smaller**, absolute font size, percentage (relative font size)

## Text Properties

- Indentation property: **text-indent**; e.g. `P {text-indent: 3em}`
- Alignment property: **text-align**; values: **left**, **right**, **center**, **justify**; e.g. `P {text-align: center}`
- Decoration property: **text-decoration**; values: **none**, **underline**, **overline**, **line-through**, **blink**; e.g. `P {text-decoration: line-through}`
- Text shadow property: **text-shadow**; e.g. `H1 {text-shadow: 0.2em 0.2em}`
- Letter and word spacing properties: **letter-spacing** and **word-spacing**; e.g. `H1 {letter-spacing: 0.1em; word-spacing: 1em}`
- Capitalization property: **text-transform**; values: **capitalize**, **uppercase**, **lowercase**, and **none**

## Colors and Backgrounds

- Foreground color property: **color**; e.g.:  
`EM { color: red } /* predefined color name */`  
`EM { color: rgb(255,0,0) } /* RGB range 0-255 */`
- Background properties: **background-color**, **background-image**, **background-repeat**, **background-attachment**, **background-position**, and **background**; e.g.:  
`H1 { background-color: #F00 }`  
`BODY { background-image: url("marble.gif") }`  
`BODY { background: white url("pendant.gif"); background-repeat: repeat-y; background-position: center; }`  
`BODY { background: red url("pendant.gif"); background-repeat: repeat-y; background-attachment: fixed; }`

## CSS - References

- [www.w3.org/TR/REC-CSS1-961217.html](http://www.w3.org/TR/REC-CSS1-961217.html)
- [www.w3.org/TR/REC-CSS2](http://www.w3.org/TR/REC-CSS2)