CSS FAQ

|  |
| --- |
| 1. [What is CSS?](http://www.hwg.org/resources/faqs/cssFAQ.html#css) 2. [What are Style Sheets?](http://www.hwg.org/resources/faqs/cssFAQ.html#ssheets) 3. [What is external Style Sheet? How to link?](http://www.hwg.org/resources/faqs/cssFAQ.html#external) 4. [What is embedded style? How to link?](http://www.hwg.org/resources/faqs/cssFAQ.html#embedded) 5. [What is inline style? How to link?](http://www.hwg.org/resources/faqs/cssFAQ.html#inline) 6. [What is imported Style Sheet? How to link?](http://www.hwg.org/resources/faqs/cssFAQ.html#import) 7. [What is alternate Style Sheet? How to link?](http://www.hwg.org/resources/faqs/cssFAQ.html#alternate) 8. [What is persistent style?](http://www.hwg.org/resources/faqs/cssFAQ.html#persistent) 9. [What is preferred style?](http://www.hwg.org/resources/faqs/cssFAQ.html#preferred) 10. [How do I combine multiple sheets into one?](http://www.hwg.org/resources/faqs/cssFAQ.html#combine) 11. [What is CSS rule 'ruleset'?](http://www.hwg.org/resources/faqs/cssFAQ.html#ruleset) 12. [What is CSS rule 'at-rule'?](http://www.hwg.org/resources/faqs/cssFAQ.html#atrule) 13. [What is selector?](http://www.hwg.org/resources/faqs/cssFAQ.html#selector) 14. [What is CLASS selector?](http://www.hwg.org/resources/faqs/cssFAQ.html#clselector) 15. [What is ID selector?](http://www.hwg.org/resources/faqs/cssFAQ.html#idselector) 16. [What is contextual selector?](http://www.hwg.org/resources/faqs/cssFAQ.html#conselector) 17. [What is attribute selector?](http://www.hwg.org/resources/faqs/cssFAQ.html#attselector) [CSS2] 18. [What is parent-child selector?](http://www.hwg.org/resources/faqs/cssFAQ.html#parentchild) [CSS2] 19. [What is CSS declaration?](http://www.hwg.org/resources/faqs/cssFAQ.html#declaration) 20. [What is 'important' declaration?](http://www.hwg.org/resources/faqs/cssFAQ.html#important) 21. [What is property?](http://www.hwg.org/resources/faqs/cssFAQ.html#property) 22. [What is shorthand property?](http://www.hwg.org/resources/faqs/cssFAQ.html#shproperty) 23. [What is value?](http://www.hwg.org/resources/faqs/cssFAQ.html#value) 24. [What is initial value?](http://www.hwg.org/resources/faqs/cssFAQ.html#initialvalue) 25. [Can I attach more than one declaration to a selector?](http://www.hwg.org/resources/faqs/cssFAQ.html#mult_decl) 26. [What is class?](http://www.hwg.org/resources/faqs/cssFAQ.html#class) 27. [What is grouping?](http://www.hwg.org/resources/faqs/cssFAQ.html#group) 28. [What are pseudo-elements?](http://www.hwg.org/resources/faqs/cssFAQ.html#pseudoel) 29. [What are pseudo-classes?](http://www.hwg.org/resources/faqs/cssFAQ.html#pseudoclass) 30. [What is cascade?](http://www.hwg.org/resources/faqs/cssFAQ.html#cascade) 31. [What is cascading order?](http://www.hwg.org/resources/faqs/cssFAQ.html#cascorder) 32. [What are inline, block, parent, children, replaced and floating elements?](http://www.hwg.org/resources/faqs/cssFAQ.html#elements) 33. [How does inheritance work?](http://www.hwg.org/resources/faqs/cssFAQ.html#inheritance) 34. [Are Style Sheets case sensitive?](http://www.hwg.org/resources/faqs/cssFAQ.html#case) 35. [Which characters can CSS-names contain?](http://www.hwg.org/resources/faqs/cssFAQ.html#charincl) 36. [Can I include comments in my Style Sheet?](http://www.hwg.org/resources/faqs/cssFAQ.html#comment) 37. [Can Style Sheets and HTML stylistic elements be used in the same document?](http://www.hwg.org/resources/faqs/cssFAQ.html#combine2) 38. [Which set of definitions, HTML attributes or CSS properties, take precedence?](http://www.hwg.org/resources/faqs/cssFAQ.html#precedence) 39. [Can CSS be used with other than HTML documents?](http://www.hwg.org/resources/faqs/cssFAQ.html#otherdoc) |

**General Questions**

1. **What is CSS?**

CSS stands for Cascading Style Sheets and is a simple styling language which allows attaching style to HTML elements. Every element type as well as every occurance of a specific element within that type can be declared an unique style, e.g. margins, positioning, color or size.

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What are Style Sheets?**

Style Sheets are templates, very similar to templates in desktop publishing applications, containing a collection of rules declared to various selectors (elements).

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is external Style Sheet? How to link?**

External Style Sheet is a template/document/file containing style information which can be linked with any number of HTML documents. This is a very convenient way of formatting the entire site as well as restyling it by editing just one file.

The file is linked with HTML documents via the LINK element inside the HEAD element. Files containing style information must have extension .css, e.g. style.css.

<HEAD>  
<LINK REL=STYLESHEET HREF="style.css" TYPE="text/css">  
</HEAD>

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is embedded style? How to link?**

Embedded style is the style attached to one specific document. The style information is specified as a content of the STYLE element inside the HEAD element and will apply to the entire document.

<HEAD>  
<STYLE TYPE="text/css">  
<!--  
P {text-indent: 10pt}  
-->  
</STYLE>  
</HEAD>

Note: The styling rules are written as a HTML comment, that is, between <!-- and --> to hide the content in browsers without CSS support which would otherwise be displayed.

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is inline style? How to link?**

Inline style is the style attached to one specific element. The style is specified directly in the start tag as a value of the STYLE attribute and will apply  exclusively to this specific element occurance.

<P STYLE="text-indent: 10pt">Indented paragraph</P>

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is imported Style Sheet? How to link?**

Imported Style Sheet is a sheet that can be imported to (combined with) another sheet. This allows creating one main sheet containing declarations that apply to the whole site and partial sheets containing declarations that apply to specific elements (or documents) that may require additional styling. By importing partial sheets to the main sheet a number of sources can be combined into one.

To import a style sheet or style sheets include the @import notation or notations in the STYLE element. The @import notations must come before any other declaration. If more than one sheet is imported they will cascade in order they are imported - the last imported sheet will override the next last; the next last will override the second last, and so on. If the imported style is in conflict with the rules declared in the main sheet then it will be overridden.

<LINK REL=STYLESHEET HREF="main.css" TYPE="text/css">

<STYLE TYPE="text=css">

<!--

@import url(http://www.and.so.on.partial1.css);

@import url(http://www.and.so.on.partial2.css);

.... other statements

-->

</STYLE>

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is alternate Style Sheet? How to link?**

Alternate Style Sheet is a sheet defining an *alternate* style to be used in place of style(s) declared as *persistent and/or preferred* .

Persistent style is a default style that applies when style sheets are enabled but can disabled in favor of an alternate style, e.g.:

<LINK REL=Stylesheet HREF="style.css" TYPE="text/css">

Preferred style is a default style that applies automatically and is declared by setting the TITLE attribute to the LINK element. There can only be one preferred style, e.g.:

<LINK REL=Stylesheet HREF="style2.css" TYPE="text/css" TITLE="appropriate style description">

Alternate style gives an user the choice of selecting an alternative style - a very convenient way of specifying a media dependent style. Note: Each group of alternate styles must have unique TITLE, e.g.:

<LINK REL="Alternate Stylesheet" HREF="style3.css" TYPE="text/css" TITLE="appropriate style description" MEDIA=screen>  
<LINK REL="Alternate Stylesheet" HREF="style4.css" TYPE="text/css" TITLE="appropriate style description" MEDIA=print>

Alternate stylesheets are not yet supported.

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is persistent style?**

Se [Alternate style](http://www.hwg.org/resources/faqs/cssFAQ.html#alternate)

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is preferred style?**

Se [Alternate style](http://www.hwg.org/resources/faqs/cssFAQ.html#alternate)

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **How do I combine multiple sheets into one?**

To combine multiple/partial style sheets into one set the TITLE attribute taking one and the same value to the LINK element. The combined style will apply as a preferred style, e.g.:

<LINK REL=Stylesheet HREF="default.css" TITLE="combined">  
<LINK REL=Stylesheet HREF="fonts.css" TITLE="combined">  
<LINK REL=Stylesheet HREF="tables.css" TITLE="combined">

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is CSS rule 'ruleset'?**

There are two types of CSS rules: ruleset and at-rule. Ruleset identifies selector or selectors and declares style which is to be attached to that selector or selectors. For example P {text-indent: 10pt} is a CSS rule. CSS rulesets consist of two parts: *selector*, e.g. P and *declaration*, e.g. {text-indent: 10pt}.

P {text-indent: 10pt} - CSS rule (ruleset)  
{text-indent: 10pt} - CSS declaration  
text-indent - CSS property  
10pt - CSS value

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is CSS rule 'at-rule'?**

There are two types of CSS rules: ruleset and at-rule. At-rule is a rule that applies to the whole style sheet and not to a specific selector only (like in ruleset). They all begin with the @ symbol followed by a keyword made up of letters a-z, A-Z, digits 0-9, dashes and escaped characters, e.g. @import or @font-face.

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is selector?**

CSS selector is equivalent of HTML element(s). It is a string identifying to which element(s) the corresponding declaration(s) will apply and as such the link between the HTML document and the style sheet.

For example in P {text-indent: 10pt} the selector is P and is called type selector as it matches all instances of this element type in the document.

in P, UL {text-indent: 10pt} the selector is P and UL (see grouping); in .class {text-indent: 10pt} the selector is .class (see class selector).

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is CLASS selector?**

Class selector is a "stand alone" class to which a specific style is declared.  Using the CLASS attribute the declared style can then be associated with any HTML element. The class selectors are created by a period followed by the class's name. The name can contain characters a-z, A-Z, digits 0-9, period, hyphen, escaped characters, Unicode characters 161-255, as well as any Unicode character as a numeric code, however, they cannot start with a dash or a digit. (Note: in HTML the value of the CLASS attribute can contain more characters).It is a good practice to name classes according to their function than their appearance.

.footnote {font: 70%}  /\* class as selector \*/

<ADDRESS CLASS=footnote/>This element is associated with the CLASS footnote</ADDRESS>  
<P CLASS=footnote>And so is this</P>

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is ID selector?**

ID selector is an individually identified (named) selector  to which a specific style is declared.  Using the ID attribute the declared style can then be associated with one and only one HTML element per document as to differentiate it from all other elements. ID selectors are created by a character # followed by the selector's name. The name can contain characters a-z, A-Z, digits 0-9, period, hyphen, escaped characters, Unicode characters 161-255, as well as any Unicode character as a numeric code, however, they cannot start with a dash or a digit.

#abc123 {color: red; background: black}

<P ID=abc123>This and only this element can be identified as abc123</P>

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is contextual selector?**

Contextual selector is a selector that addresses specific occurrence of an element. It is a string of individual selectors separated by white space, a search pattern, where only the last element in the pattern is addressed providing it matches the specified context.

TD P CODE {color: red}

The element CODE will be displayed in red but only if it occurs in the context of the element P which must occur in the context of the element TD.

TD P CODE, H1 EM {color: red}

The element CODE will be displayed in red as described above AND the element EM will also be red but only if it occurs in the context of H1

P .footnote {color: red}

Any element with CLASS footnote will be red but only if it occurs in the context of P

P .footnote [lang]{color: red}

Any element with attribute LANG will be red but only if it is classed as "footnote" and occurs in the context of P

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is attribute selector? [CSS2]**

Attribute selector is a selector defined by 1) the attribute set to element(s), 2) the attribute and value(s),  3) the attribute and value parts:

1a) A[title] {text-decoration: underline}  
All A elements containing the TITLE attribute will be underlined

1b) A[class=name] {text-decoration: underline}  
The A elements classed as 'name' will be underlined

2) A[title="attribute element"] {text-decoration: underline}  
The A elements containing the TITLE attribute with a value that is an exact match of the specified  value, which in this example is 'attribute element', will be underlined

3) A[title~="attribute"] {text-decoration: underline}  
The A elements containing the TITLE attribute with a value containing the specified word, which in this example is 'attribute', will be underlined

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is parent-child selector? [CSS2]**

Parent-child selector is a selector representing the direct descendent of a parent element. Parent-child selectors are created by listing two or more tilde (~) separated selectors.

BODY ~ P {background: red; color: white}  
The P element will be declared the specified style only if it directly descends from the BODY element:  
<BODY><P>Red and white paragraph </P></BODY>

BODY ~ P ~ EM {background: red; color: white}  
The EM element will be declared the specified style only if it directly descends from the P element which in its turn directly descends from the BODY element:  
<BODY><P><EM>Red and white EM</EM></P></BODY>

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is CSS declaration?**

CSS declaration is style attached to a specific selector. It consists of two parts; property which is equivalent of HTML attribute, e.g. text-indent: and value which is equivalent of HTML value, e.g.10pt. NOTE: properties are always ended with a colon.

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is 'important' declaration?**

Important declaration is a declaration with increased weight. Declaration with increased weight will override declarations with normal weight. If both reader's and author's style sheet contain statements with important declarations the author's declaration will override the reader's.

BODY {background: white ! important; color: black}

In the example above the background property has increased weight while the color property has normal.

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is property?**

Property is a stylistic parameter (attribute) that can be influenced through CSS, e.g. FONT or WIDTH. There must always be a corresponing value or values set to each property, e.g. font: bold orfont: bold san-serif.

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is shorthand property?**

Shorthand property is a property made up of individual properties that have a common "addressee". For example properties: font-weight, font-style, font-variant, font-size, font-family, refer to the font. To reduce the size of style sheets and also save some keystrokes as well as bandwidth they can all be specified as one shorthand property **font**, e.g.:

H1 {font-weight: bold;

        font-style: italic;

        font-variant: small-caps;

        font-size: 160%;

        font-family: serif}

can be all shorthanded to a space separated list:

H1 {font: bold italic small-caps 160% serif}

Note: To make things even simpler the line-height property can be specified together with the font-size property:

H1 {font: bold italic small-caps 160%/170% serif}

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is value?**

Value is a 'physical' characteristic of the property. Property declares **what** should be formatted, e.g. FONT  while value suggests **how** the property should be formatted, e.g. 12pt. By setting the value 12pt to the property FONT it is suggested that the formatted text be displayed in a 12 point font. There must always be a corresponding property to each value or set of values.

H1 {font: bold 180%}

In the example above the H1 selector is declared the FONT property which in its turn is declared the values BOLD and 180%.

The values suggesting alternatives are specified in a comma separated list, e.g.

H1 {font-family: font1, font2}

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is initial value?**

Initial value is a default value of the property, that is the value given to the root element of the document tree. All properties have an initial value. If no specific value is set and/or if a property is not inherited the initial value is used. For example the background property is not inherited, however, the background of the parent element shines through because the initial value of background property is transparent.

<P style="background: red">Hello <strong>World</strong></P>  
Content of the element P will also have red background

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **Can I attach more than one declaration to a selector?**

Yes. If more than one declaration is attached to a selector they must appear in a semi colon separated list, e.g.;

Selector {declaration1; declaration2}  
P {background: white; color: black}

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is class?**

Class is a group of 1) instances of the same element to which an unique style can be attached or 2) instances of different elements to which the same style can be attached.

1) The rule P {color: red} will display red text in all paragraphs. By classifying the selector P different style can be attached to each class allowing the display of some paragraphs in one style and some other paragraphs in another style.

2) A class can also be specified without associating a specific element to it and then attached to any element which is to be styled in accordance with it's declaration. All elements to which a specific class is attached will have the same style.

To classify an element add a period to the selector followed by an unique name. The name can contain characters a-z, A-Z, digits 0-9, period, hyphen, escaped characters, Unicode characters 161-255, as well as any Unicode character as a numeric code, however, they cannot start with a dash or a digit. (Note: in HTML the value of the CLASS attribute can contain more characters). (Note: text between /\* and \*/ are my comments).

CSS  
P.name1 {color: red} /\* one class of P selector \*/  
P.name2 {color: blue} /\* another class of P selector \*/  
.name3 {color: green} /\* can be attached to any element \*/

HTML  
<P class=name1>This paragraph will be red</P>  
<P class=name2>This paragraph will be blue</P>  
<P class=name3>This paragraph will be green</P>  
<LI class=name3>This list item will be green</LI>

It is a good practice to name classes according to their function than their appearance; e.g. P.fotnote and not P.green. In CSS1 only one class can be attached to a selector. CSS2 allows attaching more classes, e.g.:

P.name1.name2.name3 {declaration} <P class="name1 name2 name2">This paragraph has three classes attached</P>

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is grouping**

Grouping is gathering (1) into a comma separated list two or more selectors that share the same style or (2) into a semicolon separated list two or more declarations that are attached to the same selector (2).

1. The selectors LI, P with class name .first and class .footnote share the same style, e.g.:

LI {font-style: italic}  
P.first {font-style: italic}  
.footnote {font-style: italic}

To reduce the size of style sheets and also save some typing time they can all be grouped in one list.

LI, P.first, .footnote {font-style: italic}

2. The declarations {font-style: italic} and {color: red} can be attached to one selector, e.g.:

H2 {font-style: italic}  
H2 {color: red}

and can also be grouped into one list:

H2 {font-style: italic; color: red}

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What are pseudo-elements?**

Pseudo-elements are fictional elements that do not exist in HTML. They address the element's sub-part (non-existent in HTML) and not the element itself. In CSS1 there are two pseudo-elements: 'first-line pseudo-element' and 'first-letter pseudo-element'. They can be attached to block-level elements (e.g. paragraphs or headings) to allow typographical styling of their sub-parts. Pseudo-element is created by a colon followed by pseudo-element's name, e.g:

P:first-line

H1:first-letter

and can be combined with normal classes; e.g:

P.initial:first-line

**First-line pseudo-element** allows sub-parting the element's first line and attaching specific style exclusively to this sub-part; e.g.:

P.initial:first-line {text-transform: uppercase}

<P class=initial>The first line of this paragraph will be displayed in uppercase letters</P>

**First-letter pseudo-element** allows sub-parting the element's first letter and attaching specific style exclusively to this sub-part; e.g.:

P.initial:first-letter { font-size: 200%; color: red}

<P class=initial>The first letter of this paragraph will be displayed in red and twice as large as the remaining letters</P>

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What are pseudo-classes?**

Pseudo-classes are fictional element types that do not exist in HTML. In CSS1 there is only one element type which can be classed this way, namely the A element (anchor).  By creating three fictional types of the A element individual style can be attached to each class. These three fictional element types are: A as unvisited link, A as active link and A as visited link. Pseudo-classes are created by a colon followed by pseudo-class's name. They can also be combined with normal classes, e.g.:

A:link {background: black; color: white}

A:active {background: black; color: red}

A:visited {background: transparent; color: black}

<A HREF....>This anchor (or rather these anchors) will be displayed as declared above</A>

A.foot:link {background: black; color: white}

A.foft:active {background; black: color: red}

A.foot:visited {background: transparent; color: black}

<A CLASS=foot HREF....>This anchor and all other anchors with CLASS foot will be displayed as declared above</A>

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is cascade?**

Cascade is a method of defining the weight (importance) of individual styling rules thus allowing conflicting rules to be sorted out should such rules apply to the same selector.

Declarations with increased weight take precedence over declaration with normal weight:

P {color: white ! important}  /\* increased weight \*/  
P (color: black} /\* normal weight \*/

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What is cascading order?**

Cascading order is a sorting system consisting of rules by which declarations are sorted out so that there are not conflicts as to which declaration is to influence the presentation. The sorting begins with rule no 1. If a match is found the search is over. If there is no match under rule no 1 the search continues under rule no 2 and so on.

* 1. Find all declarations that apply to a specific selector/property  
     and  
     Declare the specified style if the selector matches the element  
     if there isn't any  
     Let the element inherit its parent property  
     if there isn't any  
     Use initial value
  2. Sort by weight (! important)  
     Increased weight take precedence over normal weight
  3. Sort by origin  
     Rules with normal weight declared in author's style sheet will override rules with normal weight declared in user's personal style sheets  
     Rules with increased weight declared in user's personal style sheet will override rules with normal weight declared in author's style sheet  
     Rules with increased weight declared in author's style sheet will override rules with increased weight declared in user's personal style sheets  
     Author's and user's rules will override UA's default style sheet.
  4. Sort by selector's specificity  
     More specific selector will override less specific one:

ID-selector (most specific), followed by  
Classified contextual selectors (TABLE P **EM.fot**)  
Class selectors (EM.fot)  
Contextual selectors - the "lower down" the more weight, (TABLE P **EM**), (TABLE P EM **STRONG**) - STRONG has more weight than EM.

* 1. Sort by order specified  
     If two rules have the same weight, the latter specified overrides ones specified earlier. Style sheets are sorted out as follows:

The STYLE attribute (inline style) overrides all other styles  
The Style element (embedded style) overrides linked and imported sheets  
The LINK element (external style) overrides imported style  
The @import statement - imported style sheets also cascade with each other in the same order as they are imported

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **What are inline, block, parent, children, replaced and floating elements?**

Inline

elements which do not have line breaks. Can occur in block elements or other inline elements, cannot contain block elements.

Inline elements in HTML 3.2; EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, A, IMG, APPLET, FONT, BASEFONT, BR, SCRIPT, MAP, INPUT, SELECT, TEXTAREA.

Inline elements in HTML 4.0; EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, ABBR, ACRONYM, TT, I, B, BIG, SMALL, SUB, SUP, A, IMG, OBJECT, BR, SCRIPT, MAP, Q, SPAN, BDO, INPUT, SELECT, TEXTAREA, LABEL, BUTTON, (INS, DEL).

Inline elements in HTML 4.0 Transitional; EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, ABBR, ACRONYM, TT, I, B, U, S, STRIKE, BIG, SMALL, SUB, SUP, A, IMG, APPLET, OBJECT, FONT, BASEFONT, BR, SCRIPT, MAP, Q, SPAN, BDO, IFRAME, INPUT, SELECT, TEXTAREA, LABEL, BUTTON, (INS, DEL).

Block

elements which do have line breaks. May occur in other block elements, cannot occur in inline elements, may contain both block and inline elements.

Block elements in HTML 3.2; H1, H2, H3, H4, H5, H6, ADDRESS, P, DL, DT, DD, UL, OL, DIR, MENU, LI, DIV, CENTER, BLOCKQUOTE, PRE, HR, ISINDEX, TABLE, FORM.

Block elements in HTML 4.0; P, H1, H2, H3, H4, H5, H6, UL, OL, PRE, DL, DIV, NOSCRIPT, BLOCKQUOTE, FORM, HR, TABLE, FIELDSET, ADDRESS, (INS, DEL).

Block elements in HTML 4.0 Transitional; P, H1, H2, H3, H4, H5, H6, UL, OL, DIR, MENU, PRE, DL, DIV, CENTER, NOSCRIPT, NOFRAMES, BLOCKQUOTE, FORM, ISINDEX, HR, TABLE, FIELDSET, ADDRESS, (INS, DEL).

Parents and children

elements which either contain (parents) or are in the content of (children) other elements, e.g. <P>text<STRONG>text</STRONG>text</P>. P is a parent of STRONG. STRONG is a child of P. If not specified otherwise, children will inherit parent's properties. NOTE: not all properties are inherited. For more information, see INHERITANCE.

Replaced

elements which content is replaced. For example content of the IMG element is replaced with an image, content of the INPUT element is replace with a field.

Floating

elements which follow the flow of a parent - inline elements.

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **How does inheritance work?**

HTML documents are structured hierarchically. There is an ancestor, the top level element, the HTML element, from which all other elements (children) are descended. As in any other family also children of the HTML family can inherit their parents, e.g. color or size.

By letting the children inherit their parents a default style can be created for top level elements and their children. (Note: not all properties can be inherited).  The inheritance starts at the oldest ancestor and is passed on to its children and then their children and the children's children and so on.

Inherited style can be overridden by declaring specific style to child element.  For example if the EM element is not to inherit its parent P then own style must be declared to it. For example:

BODY {font-size: 10pt}   
All text will be displayed in a 10 point font

BODY {font-size: 10pt}  
H1 {font-size: 14pt} or H1 {font-size: 180%}

All text except for the level 1 headings will be displayed in a 10 point font. H1 will be displayed in a 14 point font (or in a font that is 80% larger than the one set to BODY). If the element H1 contains other elements, e.g. EM then the EM element will also be displayed in a 14 point font (or 180%) it will inherit the property of the parent H1. If the EM element is to be displayed in some other font then own font properties must be declared to it, e.g:

BODY {font-size: 10pt}  
H1 {font-size: 14pt} or H1 {font-size: 180%}  
EM {font-size: 15pt} or EM {font-size: 110%}

The EM element will be displayed in a 15 point font or will be 10% larger than H1. NOTE: EM is, in this example, inside H1 therefore will inherit H1's properties and not BODY's.

The above declaration will display **all** EM elements in 15 point font or font that is 10% larger than font declared to the parent element. If this specific font is to apply to EM elements but only if they are inside H1 and not every occurrence of EM then EM must take a form of a contextual selector.

H1 EM {font-size: 15pt} or H1 EM {font-size: 110%}

In the example above EM is a contextual selector. It will be displayed in specified font only if it will be found in the context of H1.

Not all properties are inherited. One such property is background. However, since it's initial value is transparent the background of the parent element will shine through by default unless it is explicitly set.

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **Are Style Sheets case sensitive?**

No. Style sheets are case insensitive. Whatever is case insensitive in HTML is also case insensitive in CSS. However, parts that are not under control of CSS like font family names and URLs can be case sensitive - IMAGE.gif and image.gif is not the same file.

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **Which characters can CSS-names contain?**

The CSS-names; names of selectors, classes and IDs can contain characters a-z, A-Z, digits 0-9, period, hyphen, escaped characters, Unicode characters 161-255, as well as any Unicode character as a numeric code. The names cannot start with a dash or a digit. (Note: in HTML the value of the CLASS attribute can contain more characters).

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **Can I include comments in my Style Sheet?**

Yes. Comments can be written anywhere where whitespace is allowed and are treated as white space themselves. Anything written between /\* and \*/ is treated as a comment (white space). NOTE: Comments cannot be nested.

/\* This is a CSS-comment \*/

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **Can Style Sheets and HTML stylistic elements be used in the same document?**

Yes. Style Sheets will be ignored in browsers without CSS-support and HTML stylistic elements used.

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **Which set of definitions, HTML attributes or CSS properties, take precedence?**

CSS properties take precedence over HTML attributes. If both are specified, HTML attributes will be displayed in browsers without CSS support but won't have any effect in browsers with CSS support.

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)

1. **Can CSS be used with other than HTML documents?**

Yes. CSS can be used with any ny structured document format. e.g. XML, however, the method of linking CSS with other document types has not been decided yet (Feb 98).

[Table of Contents](http://www.hwg.org/resources/faqs/cssFAQ.html#toc)