
Font and text properties + media types

Last edited: Fri, 16 Mar 2001 21:55:22 GMT

Topics

- **Common**
- **Basic definitions**
- **Printing, media types and groups**
- **The list of CSS2 at-rules, which affect to the page presentation**
- **User interface**
- **Internationalization**
- **CSS3**

Common

As I have explained in the page **The basic structure of HTML and XML documents**, defining font with CSS-properties are problematic concerning compatibility. **The basis view is, that how more documents have easily and flexible exchangeable CSS-properties, the lesser documents are compatible towards HTML 3.2 concerning the presentation.**

In order to get maximum compatibility, it is necessary to use the element 'FONT' even if CSS-specification mention it in passing :

```
"<FONT class="font-style1" face="Arial"  
color="#3f3f3f">"
```

If you define the class 'font-style1', you can afterward define properties to the element using CSS. Elder browsers use instead alternative HTML 3.2 level attributes. **Don't define font-size with HTML 3.2 style** like this:

```
"<FONT class="font-style1" face="Arial" size="3"  
color="#3f3f3f">"
```

IE bypass the 'font-size' attribute if other value is given with CSS, but in Opera 3.x and Netscape 4.x don't (in Opera 4.x the bypassing works). Because the bypassing doesn't work in all CSS-capable browsers, it would be better to avoid it. If you want to exchange afterward the size of the font, enclose the element with the element 'BIG' according to the following example (you can define also 'BIG' with CSS):

```
"<BIG><FONT class="font-style1" face="Arial"  
color="#3f3f3f"></FONT></BIG>"
```

This kind of method is however toil and HTML 3.2 level attributes are quite modest compared to CSS-properties. You can never get very big compatibility. In a way some level

compatibility fulfils the media aims of CSS to offer different presentation to different devices.

But the situation would be quite different, if browsers could support attribute selectors. Existing HTML 3.2 level attributes could be used as matching criterion. There would be not needed to use special CSS class or id selectors at all. This doesn't work in Opera 4.02 even if Opera 4.02 supports generally attribute selectors. Some working examples to Netscape 6.0 :

```
"font[color="#660033"]{color:blue} /* element FONT, which
attribute 'color="#660033"', get the color 'blue' - remember the space rule; If
you put a space after the element type selector, the rule says: element FONT,
which have a descendant element, which have attribute 'color="#660033"' get the
property 'color:blue' */
*[color="#660033"]{color:blue} /* color-attribute can give also as
universal; This rule says: any element, which have exactly the attribute
'color="#660033"', get the color 'blue' */"
```

Basic definitions

To the element 'BODY' is reasonable give basic 'font-size' and 'font-family' properties:

```
"body
{font-family: Algerian, "Times New Roman", serif; /* note
quotes if the name of font have spaces - without quotes the browser doesn't
necessary find the right font-face */ "
font-size: small; } " /* it is recommended to use in the base font size
relative units */"
```

The browser goes font-families in order. First two are installed font-faces. The third ('serif') is so called as a **generic font family**, which are 'serif, sans-serif, cursive, fantasy, monospace'. Browsers can decide with certain criterion, which font they use and they can offer different font faces.

```
W3C:
CSS2: <http://www.w3.org/TR/REC-CSS2/fonts.html#generic-font-families>
(15 Fonts, 15.2.6 Generic font families) .
```

In headers is reasonable to use rich properties. I have collected to the following example in theory possible properties. I have remarked properties, which belongs only to the CSS2 standard, and I tell what properties don't work in any browser:

```
"h2
{color: #660033; /* the color of the text as keyword value */
font-family: Algerian, "Times New Roman", serif; "
font-weight: bold; /* how thick font is used - in this case as keyword
value */"
```

```

"font-size: 18pt; " /* absolute font unit */
"font-style: oblique; " /* almost like italic - the computer slant the
normal font (the default value is normal) */
"font-variant: small-caps; " /* all letters are big, but small letter are
lower; don't work correctly in MS IE 5.0 */
"text-transform: capitalize; " /* quite like the previous definition -
the first letter of the word is in upper case (other values are 'lowercase' and
'uppercase'); Note, that 'lowercase' cancels the previous definition! */
"text-decoration: overline; " /* other possible values: none,
underline, line-through, blink - the last works only in Netscape browsers and
Opera 5.x+; This value is not automatic inherited */

/* Following properties doesn't work at all and I don't recommend to use them:
*/
"font-stretch: wider; " /* CSS2 definition; It should make the font
wider, but I don't find any difference, when I tested it in IE 5.0 and Mozilla
Gecko */
"text-shadow: 5px 5px red; " /* CSS2 definition in order to shadow
text - doesn't work in any browser, but I hope that at least Netscape 6.0 would
support it */ " } "

```

In the previous definitions it is remarkable 'font-weight' value. In theory a font family can have nine font-weight values, which are marked 100-900 like 'font-weight: 600' (I don't know any, which have so many). The value of normal is '400' and bold is '700'. Browsers use certain algorithms to fill holes. In order to temporary exchange font-weight is possible to use keywords 'lighter' and 'bolder', which I don't recommend as basic values.

Font properties can be defined also as shorthand properties, which includes also the 'line-height' property:

```

"font: normal small-caps 120%/120% fantasy; " /* shorthand
definition to fonts: the first %-value is for relative font-size and the latter
concerns the line-height (this value can be expressed exceptionally:
1.2=120%=1.2em); Values can be any possible units or keywords and
combinations of them; In my mind it is however in this case easier to control
separate values; This example has also properties, which I explain later and I use
this example an another time */ " } "

```

Some properties, which are used with text don't work just any browser. In my mind it is not sensible to use these even if in future Netscape 6.0 understand them: 'word-spacing', 'letter-spacing', 'white-space', 'word-wrap'. 'Letter-spacing' is most useful of them, because many browsers support it.

Printing, media types and groups

CSS2 gives the possibility to download fonts using '@font-face' rule. Today special fonts are used by creating images, which is extremely slow method. It seems, that

downloadable fonts don't work properly (look an example from my error page)). Here is some working examples from the pages of W3C and my own definition:

```
@font-face {
font-family: "Ribbon 131 Bold BT"; /* the name of the download
able font-face - remember that all font faces doesn't include Scandinavian
special characters! */
src: url("TT028M_.TTF") /* the relative of absolute address of the
download able font */
@font-face {
src: local("Helvetica Medium"), /* the browser tries to find this
font from installed fonts */
url(http://www.fonts.org/sans/Helvetica_family)
format("truedoc"); /* if it can't find it, it tries to download it; If it is
doesn't succeed to download it, it use alternative fonts */
font-family: "Helvetica";
font-style: normal}
h2 {font-family: "Ribbon 131 Bold BT", sans-serif } /*
the browser use the downloaded font-face as ordinary way */
h3 {font-family: "Helvetica Medium", sans-serif }"
```

W3C:

CSS2: <<http://www.w3.org/TR/REC-CSS2/fonts.html#font-descriptions>> (15 Fonts, 15.3.1 Font Descriptions and @font-face).

CSS2 brings new *font matching algorithms*, which gave better result using alternative fonts ('font-size-adjust').

W3C:

CSS2: <<http://www.w3.org/TR/REC-CSS2/fonts.html#font-size-props>> (15.2.4 Font size), <<http://www.w3.org/TR/REC-CSS2/fonts.html#algorithm>> (15.5 Font matching algorithm).

In CSS2 it is possible to use '@media' to make different presentation to the screen and printing or into some other devices:

```
@media print { /* this works relative well with MS IE 5.0 and Mozilla
Gecko */
body { font-size: 10pt }
}
@media screen { /* Opera 3.6x supports only @media screen and all but
doing so, this rules doesn't make any impassable problems to it */
body { font-size: 12pt }
}
@media screen, print { /* other possible media types are: aural (speech
synthesizer), braille, embossed, handheld, projection, tty (devices with fixed
size characters), tv - note, that the device must identify itself and certain devices
```

```

need special browsers and it is impossible to test these features before there are
necessary browsers! */
body { line-height: 1.2 }
} /* note, that round brackets surrounds ordinary definitions */"
    
```

W3C:
 CSS2: <<http://www.w3.org/TR/REC-CSS2/media.html#at-media-rule>> (7
 Media types; 7.2.1 The @media rule).

If you put the media at-rule inside a single style sheet, the following conceptual scheme describes definition levels. I have used this kind of chart already, but I have added an extra level (the media rule block can have several rules like the declaration-block can have several declarations):

media rule block							
		rule					
			declaration-block				
				declaration			
	selector		property		value		
'@media print {'	'body'	'{'	'font-family'	':'	'Verdana, Arial, sans-serif;'	'}'	'}'

The media type information can connect also to the 'STYLE' or 'LINK'-elements ('<LINK rel="stylesheet" type="text/css" media="screen, print">') or '@import'-rule like '@import url(print.css) print;' (this definition is used only in printing the document).

These work at least with newest browsers. MS IE 5.x reads media-attribute values 'print' and 'screen' and Opera 4.0x in addition 'all'. Corresponding values work also with the '@media'-rule. I have found however following problems:

- '@import (...) print;' etc. has not worked any browser, which I have tested
- MS IE 4.01 and Opera 3.x read '@media print {...}' also to the screen version. '@media screen {...}' must write last and the author must be careful and define all necessary CSS another time.
- In MS IE 5.5 'media="all"' doesn't work (this concerns a beta version - Microsoft might fix this in the final release).

In theory it is also possible to define page brake points, adjust page size or orientation using '@page' rule. I tested them with IE 5.0, but it seems that they don't work (according to an e-mail they work in the newest MS IE Mac-version).

The at rule '@page' works relative well in Opera 5.x, where works also the property to prevent page breaks inside elements ('page-break-inside') - it is one of the reasons,

why this site is best printed with newest Opera. Below is some examples of using them:

```
"@page {size: 210mm 297mm;}

@page :first {size: 210mm 210mm; margin-left: 4cm;
margin-right: 4cm;}

@page :right {margin-left: 3cm; margin-right: 4cm;}

@page :left {margin-left: 4cm; margin-right: 3cm;}

@media print {
blockquote {page-break-inside:avoid} /* avoid dividing this
element into two pages */
...
}"
```

W3C:
CSS2: <<http://www.w3.org/TR/REC-CSS2/page.html>> (13 Paged media) .

Note. I handle one useful property for printing in the page Lists.

Paged media is one of the *media groups* of CSS2. CSS2 formatting features are combination of media types, media groups and *formatting models*. Formatting models and media groups are following:

- *Visual formatting model*
- *Aural rendering model*
- *Tactile media*
- *Paged media - continuous media* (this matter concerns if pages are broken using page brake rules or not)
- *Interactive - static* (can the user exchange the apperarence of the page or not)
- *Grid - bitmap* (in the previous situation the screen use *fixed-pitch character grid*; The latter media group can use also propotional true type fonts)

The relationship them to media groups and illustrate this table, which I have taken from the CSS2 specification (*N/A* means that this possibility is not applicable in certain circumstances):

Media Types	Media Groups			
	continuous/ paged	visual/aural/ tactile	grid/bitmap	interactive/ static
aural	continuous	aural	N/A	both
braille	continuous	tactile	grid	both
emboss	paged	tactile	grid	both
handheld	both	visual	both	both
print	paged	visual	bitmap	static

projection	paged	visual	bitmap	static
screen	continuous	visual	bitmap	both
tty	continuous	visual	grid	both
tv	both	visual, aural	bitmap	both

The problem of this table is in the media type 'handheld'. New WAP mobile phones can handle quite limited way graphics. So-called palm pilot and Windows CE using devices have more capabilities. Which should be the dominant device? How the WAP servers should handle information to different handheld devices? The media type 'tty' is text-only devices and it can't be used to them. Should there be a special group 'mobile_phones'? (Look at also an appendix page about this matter).

The list of CSS2 at-rules, which affect to the page presentation

It is possible to use different media groups by using at rules. One at-rule, which I have not mentioned is the charset at-rule. This exchange the used character set. The rule can be used only in external style sheets and it might look like this:

```
@charset "ISO-8859-1";
```

There is two kinds of at-rules, simple **line statements** and **statement-blocks** (or in other words block statements) and at-rules are following (I put them to the order, which they are used in style sheets):

- **Simple line statements:**
 - '@charset'
 - '@import' (before the end of the import rule must put the media type like '@import () print;')
- **Statement-blocks:**
 - '@page'
 - '@font-face'
 - '@media'

Into CSS3 is proposed new at-rules. New Mozilla/Netscape supports '@namespace', which I explain in the pages CSS notes 1 and Selectors. It is a line statement and it should be at the beginning of the style sheet like the '@charset' at-rule.

W3C:

CSS2-CSS3: <<http://www.w3.org/TR/REC-CSS2/syndata.html#q23>> (4 CSS2 syntax and basic data types, 4.4 CSS document representation), <<http://www.w3.org/TR/css3-namespace>> (CSS Namespace Enhancements) (a proposal for CSS3).

Other sites:

John Allsop:

<http://www.westciv.com/style_master/academy/css_tutorial/statements.html> (Statements).

User interface

At the same main principle it is possible to take account user settings in font and background properties.

I have tested a property ('outline'), which has worked in some preview-versions of Netscape 6.0. This paragraph should be rendered with the foreground and background colors of the desktop. Works in Mozilla Gecko and MS IE 4.01 or newer version. In those browsers work also partially the cursor like in this paragraph 'cursor:wait' (all values don't work).

In this connection in the CSS2 specification (not in CSS1) is also mentioned the possibility to use system fonts ('caption | icon | menu | message-box | small-caption | status-bar') in a part of shorthand definition like 'p {font: status-bar ...}', which work at least in MS IE 5.5 Windows. I don't recommend to use but in exceptional case like making in future pages for handheld devices.

W3C:
CSS2: <<http://www.w3.org/TR/REC-CSS2/ui.html>> (18 User interface).

Internationalization

The user is also taking account by using special language dependent properties. This aim is called as *internationalization* (it is shortened commonly to the format *i18n*). It is possible to use at the same row from left and right written text (like Hebrew) by using the 'direction' property, which is handled in the chapter 9. This is however very seldom needed possibility and I have not tested it.

Other i18n features are handled in the CSS2 specification chapter 12. It handles language depending quotations by using element 'Q' (= 'quote'; works with Mozilla Gecko and Opera 4.x+). It is also possible to use special list styles types (I handle lists in an own page).

W3C:
CSS2: <<http://www.w3.org/TR/REC-CSS2/visuren.html#direction>> (9 Visual formatting model, 9.10 Text direction: the 'direction' and 'unicode-bidi' properties), <<http://www.w3.org/TR/REC-CSS2/generate.html>> (12 Generated content, automatic numbering, and lists).

CSS3

CSS3 goes further concerning the internationalization. It will have for example the 'layout-grid' property. MS IE 5.5 supports it like some other properties, which belongs to this group.

I referred to a new at-rule. CSS3 adds also some new user interface properties. Especially paged media is developed further. The *multi-column layout* proposal of *Håkon Wium Lie* could be quite useful. I handle some CSS3 matters in the last page .

W3C:

*<<http://www.w3.org/Style/CSS/current-work>> (Working drafts) of CSS3:
<<http://www.w3.org/TR/css3-namespace>> (CSS Namespace Enhancements)
(W3C Working Draft 25 June 1999), <<http://www.w3.org/TR/i18n-format>>
(International Layout) (W3C Working Draft 10-September-1999),
<<http://www.w3.org/TR/css3-multicol>> (Multi-column layout in CSS) (W3C
Working Draft 22 June 1999), <<http://www.w3.org/TR/css3-page>> (Paged
Media Properties for CSS3) (W3C Working Draft 28 September 1999),
<<http://www.w3.org/TR/css3-userint>> (User Interface for CSS3) (W3C
Working Draft 16 Feb 2000).*

Next however something about anchors and links, because they use commonly changing font styles in different states.