

Styling the New Web

Web Usability with Style Sheets

A CHI 2003 Tutorial

Steven Pemberton

CWI and W3C
Kruislaan 413
1098 SJ Amsterdam
The Netherlands

Steven.Pemberton@cwi.nl
www.cwi.nl/~steven

This document is an extract from the full notes; it is written in XHTML and uses CSS.

It should go without saying that to properly appreciate this document, you have to view it with a browser with good CSS support.

Such browsers include [IE](#) 5 or higher, [Netscape](#) 5 or higher, [Mozilla](#) or its derivatives, and Opera.

Extracts are divided by thick lines that look like this:

Agenda

The day is split into four blocks, each of 90 minutes.

Each block consists of about 45 minutes lecture, followed by 45 minutes practical.

The breaks between blocks are 30 minutes, with 90 minutes for lunch.

Block 1

Introduction, basic CSS: selectors, fonts, colours.

Break

Block 2

Intermediate level: advanced selectors, inheritance, margins, borders, padding.

Lunch

Block 3

Advanced: text properties, background, positioning, cascading.

Break

Block 4

The Future: XML and XHTML.

About the Instructor

Steven Pemberton is a researcher at the CWI, The Centre for Mathematics and Computer Science, a nationally-funded research centre in Amsterdam, The Netherlands, the first non-military Internet site in Europe.

Steven's research is in interaction, and how the underlying software architecture can support the user. At the end of the 80's he built a style-sheet based hypertext system called Views.

Steven has been involved with the World Wide Web since the beginning. He organised two workshops at the first World Wide Web Conference in 1994, chaired the first W3C Style Sheets workshop, and the first W3C Internationalisation workshop. He was a member of the CSS Working Group from its start, and is a long-time member (now chair) of the HTML Working Group, and co-chair of the XForms Working Group. He is co-author of (amongst other things) HTML 4, CSS, XHTML and XForms.

Steven is also Editor-in-Chief of ACM/interactions.

Objectives

HTML has for too long, and incorrectly, been seen as a purely presentation language. It was originally designed as a structure description language, but extra elements were added later by browser manufacturers in order to influence the presentation. This has had the effect of limiting Web site usability by introducing presentation elements that slow down Web access, reduce or prevent accessibility to the sight-impaired, limit the sorts of devices that may be used to view websites, and reduce the end-user's options when viewing a document.

The World Wide Web Consortium (W3C) started the Style Sheet activity in 1995 in order to get HTML back to its pure form. The result of this was Cascading Style Sheets (CSS), which allows the separation of content and presentation in Web sites.

Using style sheets has many benefits, including:

- Separation of content and presentation means that Web pages are easier to write.
- Since images are no longer needed to represent styled text, Web pages download significantly faster.
- By separating out the presentation elements, blind and other sight-impaired users are able to access the Web much more easily, especially since CSS explicitly supports aural browsers.
- By allowing style sheets to specify sizes in relation to other sizes, rather than as absolute sizes, people with reduced sight can scale pages up and still see them as they were intended.
- By not coding device-dependent presentations in the HTML pages are viewable on a wider range of devices.
- You can now design the look of your site in one place, so that if you change your house style, you only need to change one file to update your entire site.

Even if the Web remained based on HTML, these would be enough reasons to use style sheets. However, the Web is now going in a new direction: XML, and XML has no inherent presentation semantics at all. To use XML you *have* to use a style sheet to make your site visible.

As a part of the movement to XML, a new version of HTML, called XHTML, is being developed. Since all presentation-oriented elements are being dropped, style sheets will become essential there too.

So the objectives of this course are to give an advanced introduction to the use of CSS to style HTML and XML documents, showing in passing how this can improve usability, and to give an introduction to the use of XML and XHTML.

CSS level 1 will be presented, since this is the version currently widely implemented, with pointers to what is in CSS 2, and what can be expected in CSS 3.

These notes have been produced entirely in XHTML and CSS, using different stylesheets for printing, screen use, and presentation. In particular, there are only three images used (to demonstrate the use of images): a star, and two versions of a flower. All other things that look like images are in fact just styled with CSS.

It should go without saying that to properly appreciate this document, you have to view it with a browser with good CSS support.

Course Plan

9:00-10:30

Introduction, basic CSS: selectors, fonts, colours; Practical

11:00-12:30

Intermediate-level stuff: advanced selectors, inheritance, margins, borders, padding; Practical

2:00-3:30

Advanced stuff: text properties, background, positioning; Practical

4:00-5:30

The Future: XML and XHTML; Practical

CSS

- CSS is a language that allows you to specify how a document, or set of documents, should look.
- Advantages:
 - Separates content from presentation
 - Makes HTML a structure language again
 - Makes HTML easier to write (and read)
 - All HTML styles (and more) are possible
 - You can define a house style in one file
 - Accessible for the sight-impaired
 - You can still see the page on non-CSS browsers
- CSS is also an enabling technology for XML

Separating Content and Presentation: Author Advantages

- Easier to write your documents
- Easier to change your documents
- Easy to change the look of your documents
- Access to professional designs
- Your documents are smaller
- Visible on more devices
- Visible to more people



Separating Content and Presentation: Webmaster Advantages

- Separation of concerns
- Simpler HTML, less training
- Cheaper to produce, easier to manage
- Easy to change house style
- Documents are smaller: less bandwidth
- Reach more people
- Search engines find your stuff easier
- Visible on more devices

Separating Content and Presentation: Reader Advantages

- Faster download (one of the top 4 reasons for liking a site)
- Easier to find information
- You can actually read the information if you are sight-impaired
- Information more accessible
- You can use more devices

Separating Content and Presentation: Implementor Advantages

- Improves the implementation (separation of concerns)
- Can produce smaller browsers

Levels

- CSS has been designed with upwards and downwards compatibility in mind.
 - CSS1: basic formatting, fonts, colours, layout; quick and easy to implement
 - CSS2: more advanced formatting; aural style sheets
 - CSS3: printing, multi-column, ...
- In general a valid CSS1 style sheet is also a valid CSS2 style sheet.
- In general a CSS2 style sheet can be read and used by a CSS1-supporting browser.

Check your log files!

- More than 95% of surfers now use a CSS1-compatible browser:
 - Microsoft IE 3, 4, 5, 6
 - Netscape 4, 6, 7 and its Mozilla-based relatives
 - Opera 3.5, 4, 5, 6, 7
- While the quality of the support for CSS on these browsers varies, though is continually improving, you never need to use the tag again!
- Today we'll be largely talking about CSS1, since it is widely implemented

Why is Usability Important for Websites?

Forrester did research among 8000+ users on why they chose one website above another equivalent one. Reasons were:

- Content 75%
- Usability 66%
- Speed 58%
- Freshness 54%

All other reasons were 14% or lower.

CSS can't help you with Content or Freshness, but it can with the other two!

Why is CSS good for usability?

- Presentation is not hard-wired in the HTML
- Users can make their own choices (font size, colours, etc), and override the documents
- Pages load faster
- Pages become more accessible for the sight-impaired (who can use speech browsers)
- Pages are viewable on a wider range of platform types

Warning about old browsers

CSS implementations are now quite good, but older browsers had a variety of mistakes. Unfortunately, some browser manufacturers want to offer backwards compatibility with those buggy old browsers. So they have two modes: compliant mode, and legacy mode.

To decide which mode to use they look at the document.

Legacy and compliant modes

- For XML it is always in compliant mode
- For HTML it is compliant mode if you include a DOCTYPE at the top of the document, for instance:

```
<!DOCTYPE HTML
PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

So if you want your CSS to work right, always include a DOCTYPE at the head of your document.

Warning about HTML: case

- HTML is case-insensitive. You can write

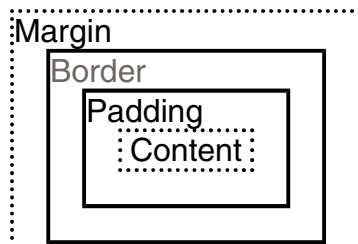
```
<H1>Title</H1>  
or  
<h1>Title</h1>
```

- Therefore, you can write selectors either as

```
H1 {font-weight: bold}  
or  
h1 {font-weight: bold}
```

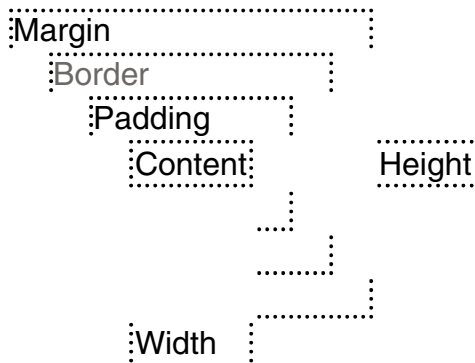
- **Newer versions of HTML are case sensitive, so specify your selectors in lower case**

Box model



- All elements have this box model.
- The margin's colour is transparent
- The border's colour is by default the same as the text of the content, but can be set separately
- The padding has the same colour as the background colour of the content

height and width properties



- The 'height' and 'width' properties of an element affect the height and width of the 'content' part
- So if you set width to 4em, padding to 3em, border to 2em, and margin to 1em, the whole box will be $(4 + 2 \times 3 + 2 \times 2 + 2 \times 1) = 15\text{em}$ wide. (But see note later on 'auto' values).

Use of margins

- Use margins for indenting:

Lorem ipsum dolor sit amet, consetetur
 sadipscing elitr, sed diam nonumy eirmod.

At vero eos et accusam et justo duo
 dolores et ea rebum. Stet clita kasd
 gubergren, no sea takimata sanctus est
 Lorem ipsum dolor sit.

Invidunt ut labore et dolore magna aliquyam
 erat, sed diam voluptua.

- extending (using negative margins):

Lorem ipsum dolor sit amet, consetetur
 sadipscing elitr, sed diam nonumy eirmod.

At vero eos et accusam et justo duo dolores
 et ea rebum. Stet clita kasd gubergren, no
 sea takimata sanctus est Lorem ipsum dolor
 sit.

Invidunt ut labore et dolore magna aliquyam
 erat, sed diam voluptua.

- adding space between paragraphs
- etc.

When vertical margins meet

When two margins meet vertically, only the larger is used (so the gap between a heading and the following paragraph is the larger of the heading's *margin-bottom* and the paragraph's *margin-top*)

An Example

When two margins meet vertically, only the larger is used (so the gap between a heading and the following paragraph is the larger of the heading's *margin-bottom* and the paragraph's *margin-top*)

An Example

When two margins meet vertically, only the larger is used (so the gap between a heading and the following paragraph is the larger of the heading's *margin-bottom* and the paragraph's *margin-top*)

Example of padding

```
blockquote
{ margin: 2em;
  background-color: yellow;
  padding: 2em
}
```

Lorem ipsum dolor sit amet, consetetur
saddipscing elitr, sed diam nonumy eirmod
tempor invidunt ut labore et dolore magna
aliquyam erat, sed diam voluptua.

Lorem ipsum dolor sit amet,
consetetur sadipscing elitr, sed
diam nonumy eirmod tempor
invidunt ut labore.

At vero eos et accusam et justo duo dolores et
ea rebum. Stet citta kasd gubergren, no sea
takimata sanctus est Lorem ipsum dolor sit
amet.

Usage of borders

Use borders for:

- Setting off text with a line each side
- Enclosing text in a box
- Putting a line under a paragraph
- Marking changed paragraphs with a line

A border will often be too close to the text: use padding to set it off from the text:

The End The End

text-decoration

- Values: *none*, or any combination of: *underline*, *overline*, *line-through*, *blink*
- Initial: *none*
- Not all browsers implement blink.
- Example:

```
a {text-decoration: underline}
```

underline, overline, ~~line-through~~, mixture

text-indent

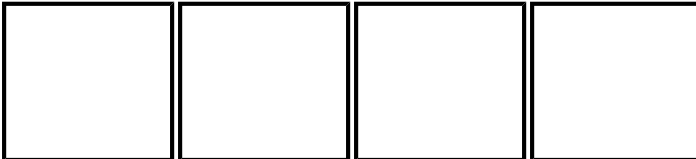
- This specifies the indentation of the first line of a block of text
- Example values: 0, 4em, 1%, ...
- Initial: 0
- Use negative values for extending a line.

Lorem ipsum dolor sit amet, consetetur
 sadipscing elitr, sed diam nonumy eirmod
 tempor invidunt ut labore et dolore magna
 aliquyam erat, sed diam voluptua. At vero eos
 et accusam et justo duo dolores et ea rebum.
 Stet clita kasd gubergren, no sea takimata
 sanctus est Lorem ipsum dolor sit amet.

m ipsum dolor sit amet, consetetur sadipscing
 elitr, sed diam nonumy eirmod tempor invidunt
 ut labore et dolore magna aliquyam erat, sed
 diam voluptua. At vero eos et accusam et justo
 duo dolores et ea rebum. Stet clita kasd
 gubergren, no sea takimata sanctus est Lorem
 ipsum dolor sit amet.

background-repeat

- Specifies how background image is to be displayed
- Values: repeat, no-repeat, repeat-x, repeat-y
- no-repeat: just once at start position
- repeat-x: repeat horizontally both sides of the start position
- repeat-y: repeat vertically above and below start position
- repeat: repeat in all directions (tile the element)
- Initial: repeat




Float

- Move elements relative to the parent
- Values: none, left, right
- Example. Logo always on the side of the window:


```
img.logo {float: right}
img.logo {float: left}
```

- Initial: none

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, no sea takimata sanctus est Lorem ipsum dolor sit amet. ***
 Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit.



Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, no sea takimata sanctus est Lorem ipsum dolor sit amet. ***
 Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit.



Use of float

- Float works on any type of element, even text blocks!
- It can replace the use of tables for some layout

```
p.menu {float: left;
        background-color: yellow;
        ....
```

```
<p class="menu">Menu items ...</p>
<p>Text ...</p>
```

```

line 1 Lorem ipsum dolor sit amet,
line 2 consetetur sadipscing elitr, sed
line 3 diam nonumy eirmod tempor
line 4 invidunt ut labore et dolore
line 5 magna aliquyam erat, sed diam
line 6 voluptua. At vero eos et accusam
line 7 et justo duo dolores et ea rebum.
Stet clita kasd gubergren, no sea takimata
sanctus est Lorem ipsum dolor sit amet.
```

Float for multi-columns

```
div.col {
  width: 31%;
  text-align: justify;
  padding-left: 1%;
  padding-right: 1%;
  border-left: solid 1px black;
  margin: 1em 0;
  float: left;
}
div.col.first {border-style: none}
```

An example of columns

Here is an example of columns using CSS:

This is a div of class = "first col". Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna.	This is a div of class = "col". Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna.	This is a div of class = "col". Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna.
--	--	--

And then back to single column format after the divs.

Implementation

Already available in:

- Microsoft IE 3, 4, 5, 6
- Netscape 4, 6, 7
- Opera 3, 4, 5, 6, 7
- X-Smiles
- NetClue
- OmniWeb
- Mozilla, Galeon, Konqueror, Phoenix, K-Meleon, Chimera, ...

Implementation

- Escape
- Icab
- Openwave for phones
- Nokia phones
- Ericsson phones
- Arachne
- Emacs-w3
- Amaya
- Athena
- Closure
- HP ChaiFarer
- ICE