

http://css-discuss.incutio.com/wiki/Print_Stylesheets Go JUL AUG OCT
183 captures 14 Mar 2010 - 11 May 2019 2017 2018 2019 About this capture

Print Stylesheets

From CSS Discuss

Contents

- 1 Printing Web documents and CSS
 - 1.1 Print stylesheets
 - 1.2 Print headers/footers and print margins
 - 1.3 Page breaks
 - 1.4 Suggested strategy for page breaks
 - 1.5 Backgrounds
 - 1.6 The @page rule and forcing Landscape orientation
 - 1.7 Printing Quirks
 - 1.8 High-quality printing
 - 1.9 Other sources
 - 1.10 About this page

Printing Web documents and CSS

This document describes some of the issues concerning the use of CSS to reformat Web documents for printing (using the media type "print"). We also discuss those aspects that CSS is not able to control or even influence. We assume a good knowledge of CSS and concentrate on practical issues, given the current deficiencies in browsers in implementing print-related CSS. For more basic tutorials and theory see "Other Sources" below. See also Media Stylesheets for practical media stylesheet strategies.

Two of the most common problems with printing occur when positioning other than static is used (e.g. position: absolute) or when there are floats. See the section below "Suggested strategy for page breaks".

Print stylesheets

At present, few sites specify print stylesheets. Some sites provide an alternative HTML file with a link entitled "Printer-friendly version" or "Print this document". The downside, of course, is in the extra maintenance required. However, some clients insist on such a solution, believing that users have come to expect such links. In his design for Devedge, Eric Meyer simply invoked a small piece of Javascript which displays a pop-up box saying "This page is already printer-friendly!". Another solution would be to link to the "Site Help" page (you do have one, don't you?), explaining that all of the site's pages will "automatically" print nicely, i.e. by design.

Here are some ideas for what to put in your print stylesheet:-

- Removing navigation aids such as menus, breadcrumbs and search boxes
- Suppressing background colours [colors] and background images
- Changing font (e.g. to a serifed font)
- Changing text line-height (leading)
- Exposing link URLs through generated content (advanced browsers only, e.g. Opera, Firefox)
- Removing link underscores (although one school of thought is that it's still useful to see, on paper, what the document links to).

A thread in css-d in September 2003 suggested that some users have come to expect the printed version of a

page to look exactly the same as what they see on screen. So it's a good idea not to make the printed version unnecessarily different and perhaps to outline, on your Site Help page, what the differences are and the reasons for them.

Be warned though, that for most users, the printed page will not look like what they see on screen in respect of backgrounds and perhaps borders and text colour too - and there's nothing you, as author, can do about it. See below under "Backgrounds".

Print headers/footers and print margins

When printing Web documents, margins are set in the browser's Page Setup (or Print Setup) dialog box. These margin settings, although set within the browser, are controlled at the operating system/printer driver level and are not controllable at the HTML/CSS/DOM level. (For CSS-controlled printed page headers and footers see Printing Headers.)

The settings must be big enough to encompass the printer's physical non-printing areas. Further, they must be big enough to encompass the header and footer that the browser is usually configured to print (typically the page title, page number, URL and date). Note that these headers and footers, although specified by the browser and usually configurable through user preferences, are not part of the Web page itself and therefore are not controllable by CSS. In CSS terms, they fall outside the Page Box CSS2.1 Section 13.2 (<https://web.archive.org/web/20180830121800/http://www.w3.org/TR/CSS21/page.html#page-box>).

The values you enter (in inches, cm or mm) are not an accurate reflection of the actual margins. The discrepancy varies between browsers and between printers, so set the values by trial and error. To calibrate your own browser/printer combination, use a test Web document with a visible border on `<body>` and a CSS margin of zero. Set these properties on `<html>` as well for good measure:-

```
html { margin: 0; border: 1px solid red; }
```

```
body { margin: 0; border: 1px solid green; }
```

I've created a calibration page on my public Web site, which also permits calibration of font size (yes, that dreaded subject). Please remember that this public page was written for general Web users, not Web designers:-

<http://www.syntacticweb.co.uk/calib.htm>

When subsequently printing pages, you may find that the top or bottom line of text is "cut off", or find even worse page-break mangling (see "Page Breaks" below). If so, slightly increase the top or bottom margin respectively. Experiments with my own sites suggest that a generous bottom margin will reduce page-break problems (it forms a kind of overflow area). However, adjusting margins is under the control of browser users rather than developers and you can place no reliance on its having been done at all, let alone well.

In your site's screen stylesheet, you may have chosen to set margins and padding on the `<body>` element (browsers set their own default values). If you have a print stylesheet, you will probably want to set the values for margin and padding to zero since the user's "print margins", as described above, must be assumed to be sufficient (and what the user wants).

Page breaks

Since the "print margins" are not under your control as Web author, you can't guarantee how much of a Web document will be printed on each page and hence where the page breaks will be. It will vary according to the user's browser, browser settings and printer. And the user's base font size and stationery size are also outside your control. You will want to use CSS to influence (not control) the position of page breaks. CSS2.1 Chapter 13 Paged Media (<https://web.archive.org/web/20180830121800/http://www.w3.org/TR/CSS21/page.html>) is essential reading. Note that, in this Chapter, CSS2.1 drops some features in CSS2 that browsers have never implemented.

Support for the page-break properties (such as `page-break-inside`) is poor, even in the latest browsers, the major exception being Opera (Opera 5 onwards) whose support is very good:-

http://www.westciv.com/style_master/academy/browser_support/printing.html

<http://www.ncdesign.org/html/s060oth.htm#pagebreak>

Worse, browsers are liable to break pages in clearly unacceptable places (see also next paragraph). For example, I have found cases where browsers will break an image across pages. Testing on my various sites suggests that Opera's page-break handling, whilst not perfect, is distinctly more robust than that of Firefox or IE.

Eric's article (see below) mentions a bug in Gecko (Mozilla) -based browsers regarding the printing of floated elements. This bug apparently has yet to be fixed [at July 2009] result in floated elements being chopped at the end of the printed page and sometimes in only one page of a document being printed [Note 2]. I've found a similar float-chopping problem with older versions of Opera (e.g. O7.21). A different bug has been reported in IE where, if the top of a floated element happens to coincide with a page break, the floated element doesn't print at all [acknowledgement: Rennan Rieke].

For printing long tables across pages see [Printing Tables](#) .

Suggested strategy for page breaks

The best policy (certainly, until browsers improve) is to give them a break (ha!) by allowing them as much discretion as you can in their choice of break points. They have particular problems with side-by-side elements, such as positioned

s; floats; and tables with two or more columns. Suitable strategies therefore include making positioned s static; unfloating floats; and breaking up tables used for layout into a series of shorter tables [Note 1]. Even though the page-break properties are poorly supported, don't over-constrain the browsers: use the "avoid" value with as light a touch as you can.

Browsers really don't like floats when it comes to printing (see previous section). The workaround is to unfloat floated elements (float: none and perhaps width: auto as well). If you do risk having a floated element when printing, it may print across a page break (the top of a floated element is anchored to the current line). Depending on the nature of the floated material, this may be acceptable behaviour. If not, only Opera will enable you to suppress such a split.

If you must have some content printed on only one page, e.g. a form, your best chance is to make it significantly less than one page long, so that you have a good margin (sic) of error. Regarding stationery size, you might choose to assume the international (ISO) standard A4 size (for an international audience) or US Letter size (for an exclusively US audience) although there's no guarantee of what size users are using. But ask yourself how you would respond if you had produced a one-page form using a word processor and a user who was vision-impaired asked for a large-print version. As always with Web design and CSS, life will be much simpler if you don't try to fight with the user by insisting on pixel perfection or guaranteed page breaks. Fluid design applies on paper as well as on screen.

Backgrounds

Most browsers default to having background colours and images switched off when printing. Nevertheless, you'll probably want to suppress the background for the body element in your print stylesheet. If the user's browser is set to print backgrounds, your action will save on ink cartridges and make printing faster, and your users will thank you for these things.

My testing shows that most browsers (e.g. Opera, Firefox and IE) not only suppress the body background by default but also the backgrounds of div, table, td and p elements (and presumably all other elements). The suppression, as part of the document's rendering, is controlled by the browser (albeit as a user preference). However, it can't be overridden through an author stylesheet. Instead, it is equivalent to the following rule in a user stylesheet (see [User Stylesheets](#)):- @media print {

```
* {
background-color: white !important;
background-image: none !important;
}
```

} This drastic suppression doesn't take account of the design of an individual Web document and can play

havoc with its readability and general presentation. Since most users will have this browser setting (probably without knowing) you may wish to take account of it in your Print Stylesheet and try to mitigate any damage.

Interestingly, Opera (up to Opera 7, I think) used also to suppress borders set on div, table, td and p elements (and presumably on all other elements).

I have also found that all the above browsers, in addition to suppressing backgrounds, may change text colours to black to try to ensure readability (for example, if a document has white text on a black background). Thus, your document may be printed very differently from how you intended.

The @page rule and forcing Landscape orientation

The @page rule has been cut down in scope from CSS2 to CSS2.1. The full CSS2 @page rule was reportedly implemented only in Opera (and buggily even then). My own testing shows that IE and Firefox don't support @page at all. According to the now-obsolete CSS2 spec section 13.2.2

(<https://web.archive.org/web/20180830121800/http://www.w3.org/TR/CSS2/page.html#page-size-prop>) it is possible to override the user's setting of orientation and (for example) force printing in Landscape but the relevant "size" property has been dropped from CSS2.1

(<https://web.archive.org/web/20180830121800/http://www.w3.org/TR/CSS21/page.html#page-box>), consistent with the fact that no current browser supports it. It has been reinstated in the CSS3 Paged Media (<https://web.archive.org/web/20180830121800/http://www.w3.org/TR/css3-page/#page-size-prop>) module but note that this is only a Working Draft (as at July 2009).

Conclusion: forget about @page for the present. If you feel your document needs to be printed in Landscape orientation, ask yourself if you can instead make your design more fluid. If you really can't (perhaps because the document contains data tables with many columns, for example), you will need to advise the user to set the orientation to Landscape and perhaps outline how to do it in the most common browsers.

Of course, some browsers have a print fit-to-width (shrink-to-fit) feature (e.g. Opera, Firefox, IE7) but it's inadvisable to rely on users having this facility or having it switched on.

[Unidentified contributor] If landscape is vital to your presentation, you can always consider PDF. There are free PDF generators for Windows, and Mac OS X has PDF generation natively. If creating a static document won't work for you (dynamic, database generated pages, for example) take a look at server side PDF generation. PHP has functions for this and more information can be found at <http://www.php.net/pdf>

Printing Quirks

(Authored by Kevin Venkiteswaran) Part of browsers' poor support for printing includes butchering content that is laid out with anything but position: static. This includes "simple" positioning such as relative and absolute. For example, Win/IE5.0 and Win/IE5.5 would show only the top-most onscreen portion of a that was positioned absolute and had overflow: auto (a CSS setup to emulate HTML frames). Similarly, Mozilla 1.6 had similar problems. The solution was to change all position properties from fixed, absolute, etc. back to static.

High-quality printing

If you've read this far, you'll see that current CSS and browser support in no way match the sophisticated results possible with PDF and the rest of the "proper" printing industry.

If you need to produce a high-quality printed version of a Web document, with sophisticated features like printed page headers and footers and with sensible page breaks, then you can't do it with current browsers. But you can do so using Prince (<https://web.archive.org/web/20180830121800/http://www.princexml.com/>) by Yeslogic (<https://web.archive.org/web/20180830121800/http://www.yeslogic.com/>). Prince can convert any well-formed HTML or XHTML document into a PDF document that can then be printed. In doing this, Prince has excellent support of CSS2.1 and of aspects of CSS3 including the CSS3 Paged Media (<https://web.archive.org/web/20180830121800/http://www.w3.org/TR/css3-page/>) module (currently a Working Draft as at July 2009). It supplements this with support for its own, proprietary CSS rules and properties. It will respect your own print stylesheet. You can specify an additional stylesheet that can make use of Prince's CSS extensions.

Using Prince, you can do advanced stuff like:-

- printed page headers and footers
- rounded corners
- newspaper-style columns with hyphenation
- counters
- tables of contents
- cross-references
- footnotes.

Prince is free for personal use. I have found it very easy to use but, of course, you have to know your CSS well to be able to produce suitable print styling rules.

[Unidentified contributor] An even better option might be the open source product CSSToXSLFO (<https://web.archive.org/web/20180830121800/http://www.re.be/css2xslfo/>), which is a utility which can convert an XML document, together with a CSS2 style sheet, into an XSLFO document, which can then be converted into PDF. It has special support for the XHTML vocabulary, because that is the most obvious language it would be used for. I have tried it, and it is fast, streaming (low footprint) and reliable.

Other sources

Eric Meyer has written an excellent basic tutorial for A List Apart called Going to Print (<https://web.archive.org/web/20180830121800/http://www.alistapart.com/articles/goingtoprint/>), giving a working example of some of the things you might want to do.

Zeldman describes his own site's print stylesheet at <http://www.zeldman.com/essentials/print/>. I have the temerity to disagree with Zeldman's use of points for print, for the accessibility reasons given above.

Jens Meiert describes the fundamentals and minimal requirements for a print stylesheet in his post: Print style sheets: The basics (<https://web.archive.org/web/20180830121800/http://meiert.com/en/blog/20070221/print-style-sheets-the-basics-for-no-excuses/>)

On the Boxes and Arrows site James Kalbach, in his article Printing the Web (https://web.archive.org/web/20180830121800/http://www.boxesandarrows.com/archives/printing_the_web.php), argues the importance of designing for print from a usability perspective and lists the candidate features to include.

About this page

This page was created by Jim Wilkinson in June 2003, who welcomes more research; questions; comments; and corrections.

Note 1: similar research is needed for multiple-column layouts implemented using <div>s and positioning.

Note 2: http://bugzilla.mozilla.org/show_bug.cgi?id=45788

Retrieved from "http://css-discuss.incutio.com/wiki/Print_Stylesheets"

- This page was last modified on 22 March 2010, at 01:50.