2 Linking styles for printing

As explained in the CSS media and alternative style sheets tutorial, there are three ways for defining alternative styles:

(1) Use @media rules in a stylesheet

@media print body {

font-family: "Palatino Linotype", "Book Antiqua", Palatino, serif;

font-size: 1em;

color: #333333;

margin-top: 2cm;

margin-right: 2cm;

margin-bottom: 1.5cm;

margin-left: 2cm

}

(2) Define an alternative stylesheet in the HTML

Most web designs probably prefer this solution.

<link rel="stylesheet" href="print-style.css" type="text/css" media="print" />

(3) Define an alternative stylesheet in the CSS

The @import at-rule allows to specify a media type, i.e. you may use this strategy to load various CSS variants from a single CSS file or within a HTML script section.

Inside a CSS file:

@import url(print-style.css) print;

Inside an HTML file

<style type="text/css">

@import url(print-style.css) print;

</style>

3 Adapt styles for printing (minimal solution)

Most often, you will have to do the following:

(1) Remove unwanted items, e.g.

#navigation, .do-not-print, #menu {display:none}

Of course, you will have to adapt this to your CSS. #navigation is an example of an element ID and .do-not-print an example of a class.

(2) Contents should not float and width should be set to 100% or auto

Remove :floats

Set width's back to 100%

(3) Contrasts should be optimized for a printer.

E.g. text should be black and the background should be white, or at least print ok on a printer with grey levels.

Simple minimalistic example

Taken from CSS Design: Going to Print by Eric Meyer, May 2002:

body { background: white; }

#menu { display: none; }

#wrapper, #content {

width: auto;

border: 0;

margin: 0 5%;

padding: 0;

float: none !important;

}

In addition, you could print out URLs like this:

a[href^="http://"]:after, a[href^="ftp://"]:after {

content: " (" attr(href) ")";

color: blue;

font-size: small;

}

4 Using CSS 2.1 pagination

4.1 @Page rule

Instead of defining margins, you can use the @page directive. **(Paged media, CSS 2.1 Specification]**

**The @page rule takes an optional pseudo-selector for first, left, and right pages. You then can define margins. However, you can't use px and pt's since these are useless units for a printer. Examples:**

**/\* Default left, right, top, bottom margin is 2cm \*/**

**@page { margin: 2cm }**

**/\* First page, 10 cm margin on top \*/**

**@page :first {**

**margin-top: 10cm;**

**}**

**/\* Left pages, a wider margin on the left \*/**

**@page :left {**

**margin-left: 3cm;**

**margin-right: 2cm;**

**}**

**@page :right {**

**margin-left: 2cm;**

**margin-right: 3cm;**

**}**

**4.2 Page breaks**

**You should specify when page breaks must occur and when they should not occur. Modern browsers implement three CSS constructs: page-break-before, page-break-after, page-break-inside, orphans and widows.**

**page-break-before and page-break-after take the following values: auto, always, avoid, left, right and inherit**

**page-break-inside can be avoid, auto or inherit**

**orphans and widows, is a number (e.g. 2)**

**Note that these are recommendations, i.e. a browser generating the print document should but does not need to use these heuristics:**

**Break as few times as possible.**

**Make all pages that do not end with a forced break appear to have about the same height.**

**Avoid breaking inside a replaced element.**

**Below are some typical use cases.**

**page-break-before**

**defines if page breaks should occur before the tag, typically titles**

**Example**

**section {page-break-before: always;}**

**h1 {page-break-before: always;}**

**page-break-after**

**defines if page breaks should occur after a tag**

**section {page-break-after: always;}**

**h1 {page-break-after: avoid;}**

**page-break-inside**

**defines if page breaks should occur inside an element.**

**p {page-break-inside: avoid;}**

**orphans and widows**

**CSS 2.1 also allows to specify what happens with orphans and widows. (from the specs**

**Example**

**p {orphans:3; widows:2;}**

**5 CSS3 Multiple columns**

**While multiple columns can make sense on web page styles that aim at larger screens - in the CSS media and alternative style sheets tutorial we explain how to target wide screen areas- it makes a lot of sense when printing on A4 or US letter paper. CSS3, in the CSS Multi-column Layout Module offers the possibility to define multiple columns in a quite flexible way.**

**Details need to be written, search for a tutorial on the web or see the specification ...**

**Example code:**

**body {**

**column-count: 2;**

**column-gap: 2em;**

**column-rule: thin solid black;**

**}**

**/\* This would create an h1 title that spans across the page \*/**

**h1 {**

**column-span: all**

**break-before: column;**

**break-inside: avoid-column;**

**break-after: avoid-column;**

**}**

**Support for this module, is so far quite bad. However, experimental proprietary extensions do work as of spring 2013, e.g.**

**-moz-column-count:3; /\* Firefox \*/**

**-webkit-column-count:3; /\* Safari and Chrome \*/**

**6 Simple example**

**Print this or better just "preview print" in order to see the stylesheet effect.**

**http://tecfa.unige.ch/guides/css/ex/just-so-stories-print.html**

**http://tecfa.unige.ch/guides/css/ex/just-so-stories-print.css**

**Highlights from the HTML code:**

**<link rel="stylesheet" href="just-so-stories.css" type="text/css">**

**<link rel="stylesheet" href="just-so-stories-print.css" type="text/css" media="print">**

**.........**

**<p class="noprint"><a href="#copyright">Copyright information</a></p>**

**.........**

**<h2 class="noprint">Copyright Information</h2>**

**"Highlights" from the print CSS:**

**/\* ------- Pagination \*/**

**h1, h2 {**

**page-break-after: avoid;**

**page-break-before: always;**

**}**

**p {**

**orphans:3;**

**widows:3;**

**}**

**.noprint, pre.copyright {**

**display:none;**

**}**