ISTA 230

Print CSS

CSS

Screen vs. Print

When we think about web pages, we typically picture them as viewed on an electronic screen, through some device with which we can interact with the page. However, we must also consider the experience users have when printing our web pages. Here are just a few of the differences between these two mediums:

Screen

- Endless canvas
- Geared towards interaction
- Unlimited resources

Print

- Fixed canvas size
- Geared towards readability
- Finite resources

While these differences are notable, it is your job as a designer to make sure that users have a pleasant experience accessing your website content, regardless of whether its on a screen or a piece of paper.

CSS

Printed Website

Traditionally, there have been three approaches when considering your printed website:

- **Do nothing.**
  Sadly, this is the approach most designers take. The printed version of their page isn't even considered, typically leaving users with a minimally readable version of the page's content, lots of wasted ink, and lots of wasted paper. While this is the easiest approach for the designer, it is rarely a good experience for the user.

- **Create a separate "printer-friendly" version of the site.**
  Some sites will actually create a second copy of their web content, linking to it from their main site with a button that says something like 'Print Version'. While this does provide a clear separation of print/screen content, it often requires content duplication*, increased HTTP requests & bandwidth usage, and usually requires user action to be effective. In my opinion, this is not nearly as effective as the third approach...

- **Use print-specific CSS rules!**
  CSS 2.1 provides designers with the option of specifying a set of CSS rules that should only be applied when printing the web page. This doesn't require any content duplication, and though it minimally increases bandwith usage and may require an additional HTTP request, it doesn't require any additional actions on the user's part. It just works!

* While content duplication is a concern, many content management systems will allow you to view the same page content using different style sheets. While this addresses the issue of maintaining multiple copies of your content, it doesn't address the issue of having to request it twice from the server. Additionally, this approach could have negative impact on your search engine rankings.

CSS

Print CSS

We've already covered how to link to a CSS file from your web page. Using <link> elements, you can link to one or more CSS files. Including a print-specific CSS file is exactly the same with the only difference being that you will specify a "media" attribute with the value of "print". Here's an example:

HTML

```
<link rel="stylesheet" type="text/css" href="common.css" />
<link rel="stylesheet" type="text/css" href="specific.css" />
<link rel="stylesheet" type="text/css" href="print.css" media="print" />
```

If not defined, the browser assumes that all <link> elements have a media attribute value of 'all', meaning that they should be applied in every context. If the designer so desires, they can set the media attribute to 'screen' (specific to desktop monitors, laptops, tablets, most mobile devices) or 'print' (specific to the printed version of the webpage). There are other values that are technically valid media attribute values. That said, they're not very well supported and, therefore, not recommended.

In the example below, the media attribute has been set for every <link> element. The designer has specified that 'common.css' and 'specific.css' should be applied when viewing the web page on a screen but **NOT** any other time. Conversely, 'print.css' will only be applied when the page is printed. While this provides a clear distinction between what styles are applied to the screen vs. the printed page, it might not be ideal since there may be CSS rules in 'common.css' or 'specific.css' (typography rules, etc.) that you'd want applied when printing the page as well.

HTML

```
<link rel="stylesheet" type="text/css" href="common.css" media="screen" />
<link rel="stylesheet" type="text/css" href="specific.css" media="screen" />
<link rel="stylesheet" type="text/css" href="print.css" media="print" />
```

CSS

Print CSS - Alternative syntax

Most of the time, designers will use a separate style sheet for their print styles. However, you can also use a CSS at-rule to include your print-specific CSS rules. Using the syntax seen below, you can specify print-specific styles inside of CSS file that is otherwise being applied to 'all' media types.

CSS - main.css

```
@media print
{
   body
   {
      background-color:white;
      color:black;
   }
}
```

This is more common when you only have a small set of print-specific CSS rules. As the size of your print-specific CSS rule set grows, a separate file often proves to be more manageable than trying to keep everything in a single file.

CSS

Printed Website - Things to consider

When thinking about your printed page, there are three questions that can help guide your thinking:

- What elements don't apply in a printed format?
- What elements should be re-formatted for print?
- What elements don't make sense in a printed format?

CSS

Hiding Elements

When thinking about what elements don't apply in a printed format, it may help to think about elements that simply don't make any sense when printed on paper. For example, site navigation links rarely make any sense when printed out. They can't be clicked on. Users can't hover their mouse over them to see where they link to. And most of the time, users aren't printing out your page because they like your navigation design.

Other common elements that don't usually have any place on a printed page are:

- Search Boxes
- Paid Advertisements
- Audio Elements
- Video Elements
- Possibly images

For images, the context of the image often determines whether it should be printed or not. If the image is decorative, it probably can be hidden when printing. However, if the image is closely related to the content of the page, then it probably should be printed. You'll need to use your judgement as a designer (and optimally client input) to determine whether images should be printed.

Because every case is different, there are likely individual elements that you'll find on your pages that you don't want to print. When you come across those, it is often useful to have a utility class (e.g., 'noPrint') that you can apply to an element to prevent it from being printed. See below for an example of how to hide specific elements when printing, as well as the implementation of the utility class described above.

CSS - print.css

```
.noPrint,
#nav,
#search,
.ad_space
{
   display:none;
}
```

CSS

Showing Print-Only Elements

On the other hand, you may want elements that **ONLY** show up when printing your webpage. For these elements, another set of utility classes often prove helpful.

CSS - main.css

```
.inlinePrintOnly,
.blockPrintOnly
{
   display:none;
}
```

CSS - print.css

```
.inlinePrintOnly { display:inline; }
.blockPrintOnly  { display:block;  }
```

Note that you'll need to determine whether you want your element to show up as an inline element or block element when applying your class names to them.

CSS

Remove white space

When printing, another item to consider is the excess whitespace that may appear when printing. For example, if you've specified that your <body> element should have left and right padding of 20%, the printed version of the page will include that padding. This will result in 20% of your user's paper going to waste! It's a good idea to remove padding and margins from your body element when printing. That said, adding in a small amount of margin around your <body> element may improve readability. *Note that adding padding to your <body> element can have mixed results in different browsers.*

CSS - print.css

```
body
{
   margin:0;
   padding:0;
}
```

Other elements may also have unnecessary padding that should be removed when printing. For example, if you've added some padding to an element to allow room for it's background image, that should be removed. Background images, by default, don't show up when a webpage is printed.

CSS

Reformatting Elements

Another area that deserves consideration is reformatting elements for print. In particular, typography should be considered a prime candidate for reformatting. By default, most browsers set the default font size to 16 pixels. However, 'pixel' is a unit of measurement that is specific to screens. If I asked multiple architects to build a house that was 1200 pixels wide, I would end up with a variety of sizes. Similarly, if I ask a user to print text at a font size of 16 pixels, the result will vary from browser to browser and printer to printer.

To address this issue, CSS has provided us with print-specific units of measurement.

- Size units (for print)
    - pt - points
    - in - inches
    - cm - centimeters
    - mm - millimeters

A 'point' is 1/72 of an inch. Most word processing software uses 12 points as the default font size. Similarly, most designers use points for their print unit of measurement, though you're free to use whatever you're most comfortable with.

While the task of reformatting your font size may sound daunting, it can be done quite easily if you use best practices: Use relative font sizes in your default CSS file and simply reset the font size of the body element to a print unit of measurement in your print-specific CSS file. The relative values will propogate using the print unit of measurement!

CSS - main.css

```
body  { font-size:62.5%;  }
h1    { font-size:150%;   }
h2    { font-size:130%;   }
h3    { font-size:115%;   }
#main { font-size:100%;   }
```

CSS - print.css

```
body  { font-size:12pt; }
```

Additionally, it's recommended that you reset any elements using decorative fonts, as they typically use more ink and are often less readable in print.

CSS - main.css

```
body  { font-family: 'WackyFont', helvetica, arial, sans-serif; }
```

CSS - print.css

```
body  { font-family: helvetica, arial, sans-serif; }
```

CSS

Reformatting Elements

Another important aspect of reformatting your page for print is removing any background images and/or colors. As previously mentioned, most browsers disable the printing of background images/colors by default. If users have enabled this, they may have done so unwittingly or for another website during the same browsing session.

Regardless of whether users have background image printing enabled, it is your job as the designer to make sure that your printed page is formatted for readability and minimal ink usage.

CSS - main.css

```
body
{
   background-color: #036;
   background-image: url('../images/tile.png');
   color: #fff;
}
```

CSS - print.css

```
body
{
   background-color: transparent;
   background-image: none;
   color: #000;
}
```

In the above example, you'll notice that we not only remove the background color and image for our page but we also adjust the font color to ensure that it is readable. Had we left the font color as 'white', we would have ended up with white text on a white background. Some browsers will notice this and automatically invert the color of the text. Others may change it to a light gray color so that it is visible but still close to the original CSS. And others will do nothing at all, resulting in a virtually blank page being printed. Make sure that you adjust font colors to a darker hue when printing.

It is also important to 'undo' any elements that have image replacements. For example, if you've used an image replacement to display the logo of your website, you'll want to make sure that users actually see something when they print your website.

While there are ways to display the background image (using CSS-generated content), this is typically more effort than is merited. Simply displaying the text of the logo is sufficient for most clients, though you should make efforts to make it look as close to the screen version as you can.

CSS - main.css

```
h1
{
   text-indent: -9999px;
   height: 100px;
   width: 200px;
   background-image: url('../images/logo.png');
}
```

CSS - print.css

```
h1
{
   text-indent: 0;
   height: auto;
   width: auto;
   background-image: none;
}
```

CSS

Reformatting Elements

When reformatting your website for print, it is also important to consider any elements that you may have floated or positioned non-statically. Historically, browsers have not done a great job printing these types of elements. Modern browsers tend to do a better job with elements floated to the left than they do with those floated to the right. Absolute- and fixed-positioned elements can cause print layout issues and are, in my experience, not worth dealing with.

CSS - main.css

```
img.rightImg  { float: right; }
```

CSS - print.css

```
img.rightImg
{
   float: none;
   display: block;     /* Center-align   */
   margin: 9pt auto;   /* printed images */
}
```

In the example above, we removed the float property for our images but also added in some additional styles to make them display centered in the middle of the page with a small margin on the top and bottom. While this may work for some cases, it is not a set rule! Use your best judgement.

CSS

Adding Content

While we often want to hide elements on our page when printing, there are other elements that may need some additional content in order to make sense on a printed page. For example, consider the following hyperlink:

HTML

```
<a href='http://www.answers.com'>Click for the answer!</a>
```

Because of the interactive nature of a webpage when viewed online, it isn't uncommon to come across cryptically titled links like this. However, when the page is printed, users won't have a clue where to look for the answer. While there's a chance the URL for your website might be listed elsewhere on the printed page, we don't want to rely on that if the user's interested in finding out where your link goes.

Instead, we can use CSS-generated content to print out the 'href' attribute of our hyperlink.

CSS - print.css

```
a:after
{
   content: "[" attr(href) "]"
   font-style: italic;
   color: #333;
}
```

While this isn't a must-have feature, it is often a nice surprise for users.

```
Click for the answer [http://www.answers.com]
```

CSS

Page Breaks

CSS also provides us with two properties specific page breaks before/after an element. 'page-break-before' allows us to specify how the browser should handle page breaks before an element. Similarly, 'page-break-after' allows us to specify how the browser should handle page breaks after an element. Both of these properties can take any of the following values:

- auto - Default, page breaks as necessary
- always - Always insert a page break before/after
- left - Force one or two page breaks, so that the next page is a left page.
- right - Force one or two page breaks, so that the next page is a right page.
- avoid - Always avoid a page break before/after *if possible*

CSS - print.css

```
#comments
{
   /* Force comments onto a new page */
   page-break-before:always;
}
```

CSS - print.css

```
h1
{
   /* Start new chapters on a new page */
   page-break-before: left; /* May be interpreted as 'always'... */
}
h2, h3, h4
{
   /* Avoid page breaks after sub-headings */
   page-break-after: avoid;
}
```

Similarly, CSS also provides us with 'page-break-inside' which allows us to specify whether the browser should allow page breaks inside of an element. The only valid options for this property are 'auto' (which is the default) and 'avoid'.

CSS - print.css

```
table
{
   page-break-inside: avoid;
}
```

It's worth noting that page break properties cannot be used on empty or absolutely positioned elements. Additionally, you may get mixed results if you try to get too fancy with these, as some browser implementations are buggy!

CSS

Print CSS - Testing

When trying to develop your print-specific CSS, testing can be challenging. There are a few different approaches for testing, listed below in order from least effective to most effective.

- **Apply your print CSS to the screen**
  You can view your print-specific CSS on the screen simply by changing/removing the media attribute in your <link> element. While this will give you a rough idea of how your page will print, issues with background images, non-print friendly units, etc. will not be evident.

- **Chrome Developer Tools - Emulation**
  Google Chrome has a built-in tool for emulating print views. In my experience, it is pretty reliable for displaying a good view of what you'll see when printing, though there are some gotchas due to the size of your screen vs. the size of a printed page.

- **Print Preview (if available)**
  If your browser supports it, 'Print Preview' will give you a better idea of how your page will print. That said, some browsers erroneously use 'screen' stylesheets for their print preview. Trial and error with your browser of choice recommended.

- **Print to PDF**
  Most modern browsers will allow you to print to a PDF file. This is an easy way to see exactly what your page will look like without using lots of paper in the process.

It is worth noting that for this course, we'll be using Google Chrome to view all of your print-specific CSS work and will be printing to PDF to view the most accurate rendering of your printed page.