



# Creating a new print stylesheet

I don't hear about this very often but once in a while I will hear about building print stylesheets so that web pages will look exactly the same in print as they do on screen.

While i think the idea is laughable, it made me think about what it would take to make web content printable. This is different than using all of CSS Paged Media (as documented in [Creating printable content from the web](#)), we might test part of the [Paged Media](#) and [Generated Content](#) specifications but I'm not confident that they'll work in all browsers.

What we want to do:

1. Put consistent margins around the text so it doesn't print right on the border of the page
2. Where possible make the text 80 characters wide and adjust the line height of the text accordingly
3. Change the background color to white and the text color to black
4. Add the link text to all URLs
5. Remove the image and replace it with the content of the figcaption element if it exists

Something to make clear from the beginning: the ideas in this post will not create a printed document that looks exactly the same as the page on the screen. Quite the opposite; this is meant to create something that is easier to read and doesn't require a full laser printer toner cartridge to print a page because of the images in it.

## Getting started

HTML has had a way to link stylesheets for different media formats since the early days. The syntax is simple, something like:

```
<link href="print.css"
      rel="stylesheet"
```

```
media="print">
```

In the print stylesheet we can add print-specific items that will enhance the printed page without changing the way the page looks on screens.

To make sure that the styles in the print stylesheet don't conflict with our regular styles, we need to wrap them in a [@media print](#) block. **Print styles must be the last declaration on a page or they will be overridden by later declarations.**

```
@media print {  
  
}
```

## What can we do

Here are some ideas of what we can do with CSS to create a better reading experience.

### Setting page margins

By default printed web pages have no margins unless you assign them to it. From Experience, reading things that are hard to read. To account for this we set equal margins for all sides of the page.

```
body {  
  margin: 2em;  
}
```

If we expect users to bind the printed pages, then we can use either `margin-left` and define each margin individually, or the four-value version of `margin`.

```
/* These two declarations are equivalent */  
body {  
  margin-top: 1em;
```

```
margin-right: 1em;
margin-bottom: 1em;
margin-left: 2em;
}

body {
  /* Clockwise: top, right, bottom, left */
  margin: 1em 1em 1em 2em;
}
```

## Adding the URL to links

There is nothing more frustrating than seeing a link and knowing it is a link but not knowing what the URL is and not being able to copy/paste it to the browser. We can fix this by appending the value of the href attribute to the link it references.

```
a[href^="http"]:after {
  content: " (" attr(href) ")";
}
```

## Hiding images and adding alt text

While it would be nice keep images in the printed output there are issues with excessive toner/ink consumption that can make it so you can't print the entire document.

To handle that, we'll remove images and captions inside figures, and insert the image's alt attribute where the image would go.

```
figure > img {
  display: none;
}

figure > figcaption {
  display: none;
}
```

```
}

/* We might not always want this */
figure > img:after {
  content: "Image: " attr(alt) ;
}
"Image: "
```

## Handling typography

Typography on screen is different than typography in print. I personally like to do the following:

1. Replace the font with one of Roboto, Verdana or Helvetica
2. Tweak the line height so it'll be comfortable to read
3. Set the top, right, and bottom margins to one inch and the left margin to two inches
  1. This assumes left-hand binding. As far as I know there's no way to do alternate page binding with CSS that doesn't include paged media extensions

```
@font-face { /* 1 */
  font-family: "Roboto";
  src: "Roboto" url("/fonts/Roboto-Regular-webfont.woff2")"woff2");
}

body {
  "woff2"body: "Roboto", sans-serif; /* 1 */
  line-"Roboto"/* 1 */
  line-height: 1.25; /* 2 */
  margin: 1in 1in 1in 2in; /* 3 */
}
```