



Which is the best method?

You might wonder which of these three methods is the best? Each of them has their own advantages and disadvantages. For example, the `link` method means that you can serve separate CSS files for various media. However, using the `@media` method means you can include all of your CSS in one CSS file. So you could have a chunk of CSS for print, and a chunk for screen, and this means less server requests. Instead of serving up two different CSS files, you're only serving one.

Tips for simple print CSS

You might think that setting styles for print would be very simple, but believe it or not, creating robust print CSS is not easy at all. The reason is because there are three levels of complexity. You have to deal with different printers that all offer it in different ways. You have to deal with browsers talking to printers, and each of the browsers offer it in a different way. And on top of that, there are different operating systems.

Because of all these variables, it's best to keep your print CSS as simple as possible. Here are a range of quick tips:

- Avoid background colors and background images, as many browsers will ignore the background property entirely.
- Avoid relying on color, as many users don't use color printers anyway.
- Avoid large areas of strong color, especially black, as this can use up people's toner, and they're probably not going to be very happy if they print out one of your pages and then lose all of their toner
- Avoid floats and positioning for large areas of the content. Some browsers will cut off loaded or positioned items at the end of the first page and they just don't print the rest of the content.
- Choose to hide sections of the layout if they're not needed for print. Classic examples might be rich footers, primary navigation, etc.

Exercise

What we're going to do is simplify styles for print. To start off we need to use an `@media` rule. Let's set the type to screen for now, so that we can see how our page looks in a browser before printing it. At the very end we can change this back to print.

```
@media screen {  
}
```

Our first rule is going to be the universal selector, which will style every single element on the page.

```
@media screen {  
  *  
  {  
    background: transparent !important;  
  }  
}
```

Our first declaration is `background: transparent`. Many browsers don't support the `background` property, including background images and color. So what this does is force them all to be transparent.

```
@media screen {  
  *  
  {  
    background: transparent !important;  
    color: #000 !important;  
    box-shadow: none !important;  
    text-shadow: none !important;  
  }  
}
```

`box-shadow` is a CSS3 property that applies a bit of shadow to any boxes. `text-shadow` applies drop shadows to the text elements. These are not good when printing, so both of these have also been set to `none` with `!important`.

```
@media screen {  
  img { max-width: 100% !important; }  
}
```

What we're trying to do here is to force images to their natural sizes.

The next thing you want to do is to quickly remove the navigational block:

```
@media screen {  
  .nav { display: none; }  
}
```

In many cases, some of this extra information is not required for printing.

When you are done, change the media type back to `print`:

```
@media print {  
}
```