BY: **CHRIS COYIER** | LAST UPDATED ON: **JANUARY 26, 2015**

Link to CSS files

You essentially hotlink directly to CSS files on Google.com. Through URL parameters, you specificity which fonts you want, and what variations of those fonts.

HTML

```
<head>

   ...

   <link rel="stylesheet" type="text/css" href="http://fonts.googleapis.com/css?family=Ta
</head>
```

> **Idea:** You can avoid an extra network request by opening that stylesheet and copy-and-pasting the @font-face stuff into your main stylesheet. But beware: Google does some User Agent sniffing stuff to sometimes serve different things to different devices as needed. You won't benefit from that if done this way.

CSS

In your CSS you can then specify these new fonts by name on any element you wish to use them.

CSS

```
body {
  font-family: 'Tangerine', 'Inconsolata', 'Droid Sans', serif;
  font-size: 48px;
}
```

FontLoader

You can use the FontLoader JavaScript instead of linking to the CSS. It has advantages, like controlling the Flash of Unstyled Text (FOUT), and also disadvantages, like the fact that the fonts won't load for users with JavaScript off.

HTML

```
<script src="http://ajax.googleapis.com/ajax/libs/webfont/1/webfont.js"></script>
<script type="text/javascript">
  WebFont.load({
    google: {
      families: ['Cantarell']
    }
  });
</script>
<style type="text/css">
```

```
  .wf-loading h1 { visibility: hidden; }
  .wf-active h1, .wf-inactive h1 {
    visibility: visible; font-family: 'Cantarell';
  }
</style>
```

Those class names, e.g `.wf-loading` get applied to the `<html>` element. So notice when the fonts are "loading", you can use that class name to hide the text. Then when they are shown, make them visible again. That is how FOUT is controlled.

Reference URL