

HTML & CSS Cheat Sheet

from Typographic Web Design 3 by Laura Franz

Web safe fonts vs web fonts

You can expect these web safe fonts to work across most platforms and browsers without using @font-face or a web font provider.

Georgia	Verdana	Arial	Times New Roman
Trebuchet MS	Impact	Courier	Comic Sans MS

Starting an HTML document

Whenever you start a new HTML document, you need to start somewhere!

```
<!DOCTYPE html>
<html>
<head>
<title>Title of the document</title>
<meta charset="UTF-8">
<link href="name-of-file.css" rel="stylesheet" type="text/css">
</head>
<body>

</body>
</html>
```

Basic Type Tags

Wrap text in tags to tell the browser what kind of element it is.

You have six headings to work with (h1-h6) but only one p!

h1, h2, h3, h4, h5, h6, p

Describe type elements in CSS like so (optional values are in parenthesis):

```
h1{
  font-family: Georgia;
  font-weight: normal; (bold)
  font-style: italic; (normal)
  font-size: 60px; (can measure in px, em, rem, %)
  line-height: 60px; (can measure in px, em, %)
  text-transform: uppercase; (lowercase, capitalize)
  letter-spacing: 5px; (can measure in px, em, %)
  text-align: center; (right, left, justify)
  color: #ffffff; (#cccccc, #999999, #666666, #333333, #000000)
}
```

Using Divs

Add a div with an ID in the HTML like so:

```
<div id="name-of-div">  
</div>
```

Describe divs in CSS like so (this is a 400px wide div centered left-right in the browser, with a 1px black border and a white background):

```
#name-of-div{  
  width: 400px;  
  border-width: 1px;  
  border-style: solid;  
  margin-top: 20px;  
  margin-right: auto;  
  margin-bottom: 0px;  
  margin-left: auto;  
  background-color: #ffffff;  
}
```

Giving the page a background color

Style the background of the whole page in CSS by describing the body:

```
body{  
  background-color: #999999;  
}
```

Validating Your Files

Validate your HTML and CSS at:

```
validator.w3.org  
jigsaw.w3.org/css-validator
```

Adjusting Space Around Elements

You can adjust the space around elements (including text elements) using margin and padding. To control spacing, set text elements to:

```
margin:0;  
padding:0;
```

You can then add space back in where needed, as needed:

```
margin-top:0;  
margin-right:0;  
margin-bottom:5px;  
margin-left:0;  
padding:0;
```

You can indent paragraphs by using text-indent :

```
text-indent: 22px;  
(use whatever measurement works best  
for you!)
```

You can also "outdent" text by adding a positive margin to the text, then move the first line out with a negative amount.

```
margin:0px;  
margin-left:22px;  
text-indent:-22px;
```

Adding emphasis without using an h1-6 tag: em, strong, class

You can emphasize words using:

```
<em></em>
```

By default, text emphasized with will be italic. But you can always style it in your CSS (just like other elements), like so:

```
em{  
}
```

You can emphasize words more strongly with:

```
<strong></strong>
```

By default, text will be bold. Again, you can style the strong element like so:

```
strong{  
}
```

You can style an element differently on the same page (for example, you can make some paragraphs look different from other paragraphs) with a class. Unlike divs, classes can be used multiple times on the same page:

```
<p class="intro">Paragraph of text goes  
here.</p>
```

And again, you can style the class in the CSS:

```
.class{  
}
```

Curly Quotes!

Use character entities so your quotation marks are curly:

```
&ldquo; (left) &rdquo; (right)
```

Using FontSquirrel for @font-face Web Fonts

You can get @font-face fonts from other sources, but FontSquirrel.com is a great place to start. It's free, and it provides you with the files you need.

1. Always testdrive a font using the word/ words you'll be using (helps you catch problems early).
2. Get the Webfont Kit (go to Webfont Kit tab).
3. If the Webfont Kit is available, you can select all four font formats, and click the "download @font-face kit" button.
4. If the Webfont Kit is NOT available, download the OTF and use the webfont generator (use optimal setting) to create a Webfont Kit.
5. View the html file with the word demo in its name in a browser (if the webfont shows up correctly, everything downloaded correctly)
6. Open the stylesheet.css file provided in the Webfont Kit, copy the @font-face syntax, and paste it into the top of your css file.
7. Copy the font-family name (including the quotation marks) from the @font-face syntax, and paste it into the element (e.g., h1) where you want to use the font. *Font family names must match exactly* (in the @font-face syntax and in the element where you are using it)!
8. Put the four or five font files (eot, woff, ttf, svg – sometimes one of the files is split into two), in the *exact same folder as the css file*. Do not put the font files into their own folder inside the folder. They must be at the same level as the css.
9. Make a note in your CSS about where you got the font from, who the designer is, etc.

Making Notes in CSS with Comments

CSS comments don't affect the HTML file.

```
/* Use slashes and asterisks like  
this to comment in the CSS. */
```

Building a Font Stack

Font stacks provide a "fallback font" in case something goes wrong with the web font. Separate fonts with a comma like this:

```
h1{  
    font-family: `rochester`, Georgia;  
}
```

Using Google Web Fonts

You can get web fonts from other services, but fonts.google.com is a great place to start. It's free, and it provides you with the files you need.

The directions on the site are pretty straight forward. Just remember: don't use more weights and styles than you need (it will slow your page down). And use the directions for the "Standard" Embed.

RESPONSIVE LAYOUTS

Mobile First

Always build mobile first (styling bigger versions with media queries).

Breakpoints / Media Queries

At the point where a layout "breaks" and you need to restyle elements, use a media query.

Example:

```
@media (min-width: 590px) {  
}
```

Universal Selector (*)

Is used to style all elements in the CSS file. For example, to set the margin and padding of all elements to 0, type:

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

max-width

To keep everything in place, all elements should live *inside* the main_container, which is set with a max-width. For example:

```
#main_container{  
    max-width:1400px;  
}
```

Responsive Widths

The divs inside the main_container are made responsive by setting width in %, not pixels.

Put Columns Side-by-Side

Using the float property. For example:

```
#intro_column{
  float:left;
}
```

Total Width Must be Less Than 100%

The left margin, left padding, width, right padding, and right margin of all elements in a row must equal 100% or less. Otherwise, they won't fit, and the last element will fall to the next row. *Recommendation: don't set right margin.* Let the "left over" space act as the right margin.

Main_container{overflow:hidden;}

Otherwise, when internal (aka "child") elements float, the main_container won't recognize them, and will look too short.

Columns Equal Height

Add a large amount of padding-bottom (e.g., 999px), then a negative amount of margin-bottom (-999px). Note: This only works if the main_container containing the columns is set to overflow:hidden;

vw Units

If you need a vertical space (height) to be responsive and similar to a horizontal space (width), try using vw units for the height. 1 vw is equal to 1/100 of the viewport width. Thus it is equal to 1% of the width of the viewport.

Note: once the viewport is bigger than the main_container, 1vw (1/100 of the viewport) is *no longer equal* to 1% (1/100 of the main_container).

Make Mobile = Responsive

Tell mobile device to use a viewport equal to their device width by including this line of syntax in the <head></head>

```
<meta name="viewport"
content="width=device-width">
```

Use respond-min.js for IE6-8

Help early Internet Explorer browsers read the max-width or min-width media queries by putting a copy of respond-min.js in a scripts file and putting this syntax immediately following the css line of syntax:

```
<!--[if lt IE 9]>

<script type="text/javascript"
  src="scripts/respond-min.js">
</script>
<![endif]-->
```

ADDING IMAGES

Content Images

Put an image in your "images" folder and use the following syntax in your HTML:

```

```

Make Content Images Pseudo Responsive

Use the following syntax in your CSS:

```
img{
  width:100%;
}
```

Background Images

Can be used in the background of any element. Put an image in your "images" folder and use the following syntax in your CSS (this example puts the background image in the body):

```
body{
  background-image:url(images/filename.png)
}
```

GENERAL TYPOGRAPHY

Break tag

Sometimes, you need lines of text to break in a specific way, but stay in the same paragraph. (Example, lines of poetry). Use the break tag. But use it sparingly! Never use it to fix ragged edges.

```
<br>
```