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)
  }
```

Note: See http://www.december.com/html/spec/colorhues.html for more colors!

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 on your web page, use the Universal Selector (*) like so:

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

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

```
margin-bottom:5px;
```

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-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 test a font using the word(s) 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. */
```

Making Notes in HTML with Comments
HTML comments don't show up in the browser.

```
<!-- use angle brackets, hyphens, and
spaces to comment -->
```

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). This means the CSS at the top is for Mobile view!

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, and all type to Verdana, type:

```
*{margin:0;
  padding:0;
  font-family: verdana;
  }
```

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.

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 EI6–8
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:

```
<figure>
<img src="images/yourimage.png">
</figure>
```

Add Alternative Text
For the browser to display if the image doesn't load (will also be used by text readers for people with vision disabilities.).

```
<figure>
<img src="images/yourimage.png"
alt="descriptive words about image">
</figure>
```

Add the caption
For images that need captions, add them inside the figure tags.

```
<figure>
<img src="images/yourimage.png"
alt="descriptive words about image">
<figcaption>
Caption about image
</figcaption>
</figure>
```

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)
   }
```

NOTE: a background image will repeatedly tile by default. You can change this using:

```
background-repeat: no-repeat
```

You can repeat an image along the x-axis:

```
background-repeat: repeat-x;
```

You can repeat an image along the y-axis:

```
background-repeat: repeat y;
```

You can place an image in a specific location. There are multiple ways to do this, so see

https://css-tricks.com/almanac
/properties/b/background-position/

MORE 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>
```

Unordered Lists and List Items

An unordered list (will have bullets) with two list items inside it uses this syntax:

```
<ul>
  <li>peaches</li>
  <li>apples</li>
</ul>
```

Ordered Lists and List Items

An ordered list (will have numbered items) with two list items inside it uses this syntax:

```
<ul>
  <li>peaches</li>
  <li>apples</li>
</ul>
```

Remove Bullets from Unordered Lists

Uses this CSS:

```
li{
  list-style-type: none;
  }
```

Use Images as Bullets in Unordered Lists

Remove the bullets, then bring in the image:

```
li{
  list-style-type: none;
  list-style-image:
     url(images/bullet.png);
  }
```

Collapsing Margins

The best way to set vertical space above and below paragraphs and headings is to use *margin-top* and/or *margin-bottom*.

Why? Because the browser will collapse a margin-bottom and margin-top that bump up against each other. The browser will show the larger of the two spaces (instead of adding them together and giving you a space that is too big).

Adding a Rule Line

A rule line is just a border, but using only setting one "side" of the element. For example, you can add a border above an h2 like so:

```
h2{
  border-top:1px solid #666666;
  }
```

The syntax above uses some short-hand. It lists the width, kind of line, and color all at once… separating the three values with a space. NOTE: if you do not tell the border what kind of line to use, it will not show up!

Applying Multiple Classes

You can apply multiple classes to an element. Simply separate the classes with a space like so:

```
<h2 class="class1 class2">
A Heading of Sorts</h2>
```

Em dashes and En dashes

```
&mdash;     &ndash;
```

More Character Entities

Whether you need an accented letter or a copyright symbol, try this resource for character entities:

```
http://digitalmediaminute.com/
reference/entity/index.php
```

Fix Fake Bold and Italic (Google Fonts) in IE8

IE8 and Google Fonts don't play well together. To fix this, create a conditional comment that tells IE8 to access font weights and styles separately. For example:

```
<!--[if lte IE8]>
<link
href='https://fonts.googleapis.com/css?f
amily=Crimson+Text:400'
rel='stylesheet' type='text/css'>
<link
href='https://fonts.googleapis.com/css?f
amily=Crimson+Text:400italic'
rel='stylesheet' type='text/css'>
<![endif]-->
```

Show or Hide Using Display Property

You can hide an element at one browser size, then show it in another using the display property. The display property makes the element disappear and NOT leave a blank space To show/hide an element, apply a class to it. For example:

```
<img src="images/clown.jpg"
class="show-hide">
```

In the CSS, in the media query (or mobile view) where you want the element hidden, use the following syntax:

```
.show-hide{
  display:none;
```

```
}
```

In the media query (or mobile view) where you want the element to show up, use the following syntax:

```
.show-hide{
  display:block;
  }
```

NOTE: display:block; makes the element a block-level element… meaning it will have a break before and after it. If the element should be inline (in the same line as the elements before and after it), use this syntax instead:

```
.show-hide{
  display:inline;
  }
```

ADVANCED CSS
(INSTEAD OF APPLYING A CLASS)

Combine Two Selectors
Sometimes, you want an h3 that comes after an h2 to have different spacing than the other h3s. You can target this situation like so:

```
h2+h3{
  margin-top:15px;
  }
```

The above syntax means: any h3 that immediately follows an h2 should do the following…

Of course, you can combine two selectors anytime you want or need to target an element that comes after another element.

:last-child
You can also target the last kind-of-element inside another element. For example, if you want to style the last paragraph inside a div called "experience_column" you'd write:

```
#experience_column p:last-child{
  }
```

Target Elements Inside a Div
Sometimes, you want to style an element (e.g., a list) in one column differently than the same kind of element in another column.

You can target elements based on what divs they are in. Type in the ID for the div, a space, then the element you are styling. For example, these two lists would be different colors:

```
#hot_column ul{
  color:red;
  }
#cold_column ul{
  color:blue;
  }
```

NAVIGATION / LINKS
Navigation bars and lists are set using lists.
Use the unordered list (ul) and list item (li) tags to create a list in the HTML:

```
<ul>
    <li>link one</li>
    <li>link two</li>
    … (do all the items)
</ul>
```

In the CSS, style the list items by indicating which ul and li you are styling. For example, styling the following ul and li would affect only the ul and li in the div with the id="header":

```
#header ul{
    margin-bottom:60px;
    }
#header ul li{
    list-style:none;
    margin-bottom:7px;
    }
```

Creating a Navigation Bar
The CSS below describes a horizontal list of links in your header, with no bullets, and some space between the links:

```
#header ul li{
  list-style:none;
  display:inline;
  padding-left:2%;
  padding-right:2%;
  }
```

Make the Links Work
To link to another page, use the following syntax:

```
<a href="nameofthepage.html"></a>
```

You can wrap that syntax around a word, a list item, an image, even an entire div.

For the purposes of your navigation, you'll wrap your list items in a similar manner:

```
<li><a
href="nameofpage.html">Link</a></li
>
```

Trouble-shooting: If Your Links Don't Work

1. make sure there are no spaces in the names of your files or your a href;
2. make sure the link syntax and the names of the folders are a perfect match (including case);

Style Your Links

By default, unvisited links are bright blue and underlined. Visited links are bright purple and underlined. Active (clicked) links are bright red and underlined. Hover (rollover) links do not have a unique state unless you describe one.

The syntax for describing a link in CSS is

```
a:link { }
a:visited { }
a:hover { }
a:active { }
```

If the link colors and styling don't work, make sure they are in the order shown above. Hover must come third, and active must come last in the syntax for these two states to work! (A mnemonic to help you remember the order: LoVeHAte).

Describe the links like any other element or selector in CSS. For example, unvisited links set in black, and no underline:

```
a:link {
   color:#000000;
   text-decoration:none;
   }
```

If your links have a background color make sure it is styled in the link syntax, not the li syntax. That way, you can change the background color in a:hover... and create a rollover effect!