

**WEB TYPE, PART I: RECAP**

**WEB-SAFE FONTS (FONTS ON SYSTEMS)** Fonts that are available to viewers’ browser because the fonts are already on their systems/devices. A short list of fonts meet the criteria of being available on numerous devices upon purchase. Easy to specify in CSS.

**NON WEB-SAFE FONTS** Fonts available through various CSS3 techniques that enable the specification and display of a much wider range of typefaces on web-based devices. Relatively easy to specify in CSS, along with sister HTML techniques, depending on the source/license permissions of the fonts.

**WEB TYPE, PART II**

*About that non web-safe font thing...* There are two main techniques for displaying typefaces on to computers that don’t have the fonts on them:

— **TECHNIQUE #1 (FONTS ON THIRD PARTY SERVER):** Asks the browser to “point to” a font housed on a server owned by a third party, like Google Fonts. In that example, Google Fonts is the host of the actual font files, and we write the code to ask the computer to connect to Google and render the font from there. Have to follow the source’s instructions for code placement.

— **TECHNIQUE #2 (@FONT-FACE EMBEDDING SERVICES):** Asks the browser to “push out” a font on your server or a third party server, and display it on other computers. Have to have license permission to do this. Depending on the source of the font files, you either reach to a font host’s server through a head tag element and css @font-face rule(s); or, you push fonts to your web server (include font files in a ‘fonts’ folder in your root folder), and call them up to display on the site with css @font-face rule(s). Can have high license costs: *Emigré* (\$), *H&Co* (\$), *Typotheque* (\$), *Adobe Fonts* (formerly *TypeKit*; free and \$), *FontSquirrel* (sometimes free), and many other foundries. Requires specific coding technique — the @font-face rule.



TypeKit, when it was a startup prior to Adobe ownership, 2011.

**@FONT-FACE SYNTAX**

Code varies based on source of license and font files. A simple example when you have a license to have the actual font files on your computer requires the font files be placed in a ‘fonts’ folder within your HTML root folder, and use of the following CSS:

```
@font-face {
  font-family: bttf_font;
  src: local(bttf_font), url('fonts/bttf.ttf') format('opentype');
}

h1 {
  font-family: bttf_font, Georgia, sans serif;
  font-size: 32px;
  font-weight: 400;
}
```

**CSS WEIGHTS & STYLES**

In CSS, font weights are styled using numeric equivalents:

- 100 = thin
- 200 = extra-light
- 300 = light
- 400 = normal, book
- 500 = medium
- 600 = demi-bold
- 700 = bold
- 800 = heavy
- 900 = black

### LISTS

In creating menus for site navigation, a good standard is to use html list tags. In HTML , there are ordered lists and unordered lists.

**ORDERED LISTS** Assume a need for a list of items that show up in a presented order, ordered lists are HTML elements that are designed to appear preceded by numbers or letters, as in:

- 1) Turn on your engine.
- 2) Pull out of your driveway.
- 3) Drive toward the interstate.

The html tag for an ordered list is: <ol> </ol>

**UNORDERED LISTS** Assume a need for a list of items that do not require alpha-/numeric prefixes as part of their presentation. They are designed to appear simply as content in list form, as in:

- This
- That
- The other thing

The html tag for an unordered list is: <ul> </ul>.

**LIST TAG** Individual items appearing in an ordered list or an unordered list need to be individually designated with the <li> </li> tag. Note that the unordered list defaults to showing a bulleted list. This attribute can be removed from <li> in the CSS styling (**list-style: none**). Unordered lists are good choices for navigation because menus are frequently comprised of 'lists' of links, and <li> items can be specified in css to display 'inline,' i.e., horizontally, as opposed to vertical (default).

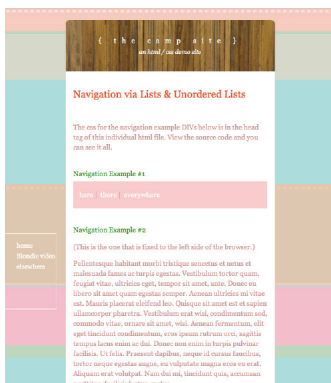
### NAVIGATION: PUTTING IT ALL TOGETHER

Here is an example of HTML code for a DIV that has been designated for navigation content (a menu of links) containing an unordered list and its corresponding individual list elements:

```
<div id="nav_example1">
  <ul>
    <li><a href="index.html"> Take me to your leader </a></li>
    <li><a href="http://www.youtube.com/watch?v=CqkwVTmLWy4"> Let me see her </a></li>
    <li><a href="http://www.google.com" target="_blank"> Get me outta here </a></li>
  </ul>
</div>
```

The corresponding CSS for these tags must include properties and values for the DIV, the unordered list relative to that DIV, list elements relative to the unordered list in that particular DIV, and corresponding link states in the DIVs unordered-list list elements, as below (see the nav demo page):

```
#nav_example1 { [properties and values here] }
#nav_example1 ul { [properties and values here] }
#nav_example1 li { [properties and values here] }
#nav_example1 li a { [properties and values here] }
#nav_example1 li a:link { [properties and values here] }
#nav_example1 li a:visited { [properties and values here] }
#nav_example1 li a:hover { [properties and values here] }
#nav_example1 li a:active { [properties and values here] }
```



CAMP SITE Demo page for controlling navigation with lists (link at end of Navigation page)