



Accelerate your cloud-native dev with CloudBees. Seamless, secure, and now FREE to try!

ads via Carbon

@adamfschwartz 

Code on GitHub

THE MAGIC OF



CHAPTER 5

Typography

Typography is hard

First off, give yourself a pat on the back. You've made it this far. You're reading about typography!

Now take a moment to realize that *most of the things you look at every day are type*. So before discussing typography, let's look at it.

Take a moment and just fly through some of these beautiful typographically-based design collections:

- Awwwards: Typography in Web Design
- Tutsplus: 10 Examples of Inspirational Typography on the Web
- Web Design Ledger: 21 Beautiful Examples of Typography in Web Design
- Smashing Magazine: A Journey Through Beautiful Typography In Web Design
- Web3Canvas: 20 beautifully designed Typography Website Inspiration 2013

Seeing

As with color, designing well with typography requires using your eyes. Really see the shape of each glyph. Notice the negative space between glyphs. You have more control over these things than you might think. And over the course of the next few chapters, I'm going to show you how.

Basic CSS Typography Tools

The basic tools in your toolkit consist of the following:

- Font properties
 - `font-family` — inherit, serif, monospace
 - `font-size` — inherit, 18px, 10px
 - `font-weight` — inherit, **bold**, 100
 - `font-style` — inherit, *italic*
 - `font-variant` — inherit, SMALL-CAPS
- Text properties
 - `text-align` — inherit, center
 - `text-decoration` — inherit, underline, overline, ~~line-through~~
 - `text-indent` — inherit, 50px
 - `text-shadow` — inherit, 0 1px 3px red
 - `text-transform` — inherit, UPPERCASE
- Miscellaneous
 - `letter-spacing` — inherit, 5 p x
 - `white-space` — inherit, pre pre
 - `line-height` — inherit, .75em
 - `word-spacing` — inherit, 100px 100px

Now that you have a basic idea of what is possible, let's look at a few combinations of these properties.

Examples

Example 1: Light, uppercase, with spacing

Light text colors on dark backgrounds tend to feel as if they have a heavier weight. So with this example of white on purple , we went with a `font-weight: 300` .

Using all uppercased letters can be a powerful effect, but you'll want to use it sparingly. One thing that can happen with uppercased words is THE LETTERS CAN FEEL JAMMED TOGETHER. This is because the default kerning and letter-spacing is meant for mostly-lowercased words. To compensate, and also to add a little more gravitas, we added a generous letter-spacing: .4em .

As can be seen in the letter spacing potion, letter-spacing and text-align don't play so nicely together because the spacing is added to the right of each letter. To compensate for this, when using these two properties together, we add a padding-left to match the chosen letter-spacing (in this case .4em).

```
.typography-example-1 {  
  background-image: linear-gradient(135deg, #723362, #9d223c);  
  background-color: #9d223c;  
  color: #fff;  
  padding: 1em 0;  
  font-weight: 300;  
  font-size: 1.8em;  
  text-transform: uppercase;  
  text-align: center;  
  letter-spacing: .4em;  
  padding-left: .4em  
}
```

EXAMPLE ONE

Example 2: Contrasting weights and styles

In this next example, we show how two lines of text interact with each other.

The first line is given a similar treatment as Example 1, but with a heavier weight of 700 . To contrast this, the second line is given a thinner weight, 300 , an italic font style, and a lighter color, #888 .

```
.typography-example-2 {  
  background: #fff;  
  color: #000;  
  border: .5em solid;
```

```
font-size: 1.5em;
padding: 1em 0;
text-align: center
}

.typography-example-2 .title {
  font-weight: 700;
  text-transform: uppercase;
  letter-spacing: .4em;
  padding-left: .4em
}

.typography-example-2 .author {
  color: #888;
  font-size: .7em;
  font-weight: 300;
  font-style: italic;
  letter-spacing: .12em;
  padding-left: .12em
}
```



Example 3: Individual letter treatment

Sometimes you want to *have a little fun* with type. Lettering.js is a great little tool for doing just that. But if you're willing to type out a bunch of `span` wrappers for every letter yourself, you can do the same sort of thing without javascript!

```
.typography-example-3 {
  border: .2em solid #ff4136;
  font-size: 3em;
  text-align: center;
  padding: .2em;
  color: #fff
}

.typography-example-3 span {
  display: inline-block;
```

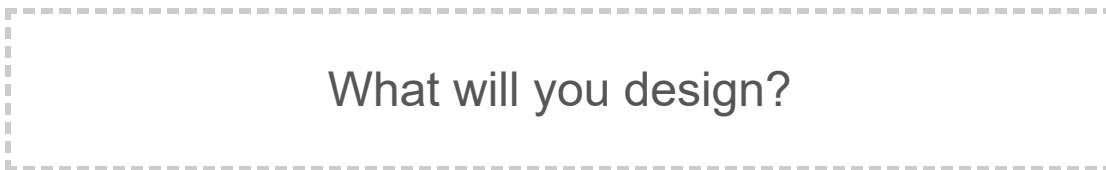
```
width: 20%;  
padding: .8125rem  
}
```

```
.typography-example-3 span:nth-child(1) { background: #ff4136 }  
.typography-example-3 span:nth-child(2) { background: #ff851b }  
.typography-example-3 span:nth-child(3) { background: #2ecc40 }  
.typography-example-3 span:nth-child(4) { background: #0074d9 }  
.typography-example-3 span:nth-child(5) { background: #b10dc9 }
```



Typography lives everywhere

Hopefully these few examples have given you an idea of what's possible with the various typographical CSS properties. Typography is such a crucial part of design that a lot of typographical thinking and concepts are embedded in the other chapters. We'll continue to explore typography as we go.



Butterick's Practical Typography

For more fun with typography, check out the amazing resources below. In particular, glance over Butterick's Practical Typography^[1]. It's a must-read for anyone with an interest in typography.

Further reading

- Butterick's Practical Typography
- CSS Typography Cheat Sheet
- Type Scale
- Codedrops: Make a Statement with Type

Citations

1. Butterick's Practical Typography



Chapter 4: Color

Chapter 6: Transitions

