Typography
Best Practices

T₃

Imprint

About This Book

Web design is not just about a flamboyant or simply beautiful website appearance. This book is about small — but crucial — details, in a specialty that is all about details. Whether you're interested in determining the right paragraph layout or typographic details, observing the correct typographic etiquette or making the other small decisions that will dramatically influence how your website is perceived, many answers will present themselves in this collection of articles.

TABLE OF CONTENTS

The Perfect Paragraph

BY HEYDON PICKERING ❧

In this chapter, I'd like to reacquaint you with the humble workhorse of communication that is the paragraph. Paragraphs are everywhere. In fact, at the high risk of stating the obvious, you are reading one now. Despite their ubiquity, we frequently neglect their presentation. This is a mistake. Here, we'll refer to some time-honored typesetting conventions, with an emphasis on readability, and offer guidance on adapting them effectively for devices and screens. We'll see that the ability to embed fonts with `@font-face` is not by itself a solution to all of our typographic challenges.

A Web Of Words

In 1992, Tim Berners-Lee circulated a document titled "HTML Tags[1]," which outlined just 20 tags, many of which are now obsolete or have taken other forms. The first surviving tag to be defined in the document, after the crucial anchor tag, is the paragraph tag. It wasn't until 1993 that a discussion emerged[2] on the proposed image tag.

Bursting with imagery, motion, interaction and distraction though it is, today's World Wide Web is still primarily a conduit for textual information. In HTML5, the focus on writing and authorship is more pronounced than ever. It's evident in the very way that new elements such as `article` and `aside` are named. HTML5 asks us to treat the HTML document more as… well, a document.

It's not just the specifications that are changing, either. Much has been made of permutations to Google's algorithms[3], which are beginning to favor better written, more authoritative content (and making work for the growing content strategy[4] industry). Google's bots are now charged with asking questions[5] like, "Was the article edited well, or does it appear sloppy or hastily produced?" and "Does this article provide a complete or comprehensive description of the topic?," the sorts of questions one might expect to be posed by an earnest college professor.

---

1. http://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/Tags.html
2. http://1997.webhistory.org/www.lists/www-talk.1993q1/0182.html
3. http://googlewebmastercentral.blogspot.com/2011/04/high-quality-sites-algorithm-goes.html
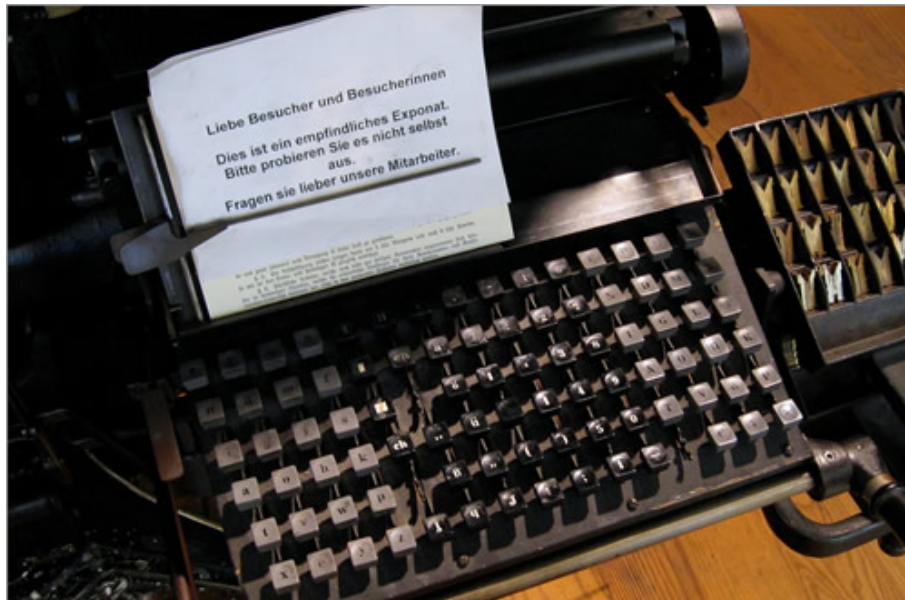4. http://www.alistapart.com/articles/thedisciplineofcontentstrategy/
5. http://googlewebmastercentral.blogspot.com/2011/05/more-guidance-on-building-high-quality.html

This increased support for quality writing, allied with the book-like convenience and tactility of smartphones and tablets, means there has never been a better time for reading online. The remaining task is to make the writing itself a joy to read.

What Is The Perfect Paragraph?

As designers, we are frequently and incorrectly reminded that our job is to "make things pretty." We are indeed designers — not artists — and there is no place for formalism[6] in good design. Web design has a function, and that function is to communicate the message[7] for which the Web page was conceived. The medium is not the message.

Never is this principle more pertinent than when dealing with type, the bread and butter of Web-borne communication. A well-set paragraph of text is not supposed to wow the reader; the wowing should be left to the idea or observation for which the paragraph is a vehicle. In fact, the perfect paragraph is unassuming to the point of near invisibility[8]. That is not to say that the appearance of your text should have no appeal at all. On the contrary: well-balanced, comfortably read typography is a thing of beauty; it's just not the *arresting* sort of beauty that might distract you from reading.

(Image: Marcin Wichary[9])

6. http://en.wikipedia.org/wiki/Formalism_%28art%29
7. http://www.lukew.com/ff/entry.asp?232
8. http://speakerdeck.com/u/smashingmag/p/the-invisible-side-of-design
9. http://www.flickr.com/photos/mwichary/2251302479/sizes/l/in/photostream/

As a young industry that champions innovation and rates its practitioners based on their ability to apprehend (sorry, "grok") the continual emergence of new technologies, frameworks, protocols and data models, we are not particularly familiar with tradition. However, the practice of arranging type for optimal pleasure and comfort is a centuries-old discipline. As long ago as 1927, the noted typographer Jan Tschichold[10] spoke of the typesetting "methods and rules upon which it is impossible to improve" — a set of rules it would be foolish to ignore.

So, please put your `canvas` element and data visualization API to one side just for a short while. We are about to spend a little time brushing up on our typesetting skills. It's called "hyper*text*," after all.

Setting Your Paragraphs

THE TYPEFACE

Your choice of font is important, but the *kind* of "family" you choose is project-specific, and we won't discuss it here except to make one point: the conventional wisdom among Web designers that only sans-serif[11] fonts are suitable for body text is just a rule of thumb. Although serif fonts, with their greater complexity, may tend to be less effective at small sizes, there are many other factors to consider. A diminutive x-height[12], for example, could impair the readability of a font from either camp. Some serif fonts are highly legible and attractive for paragraph text if they are set properly. Matthew Carter's[13] screen-sympathetic Georgia[14] is a case in point.

---

10. http://en.wikipedia.org/wiki/Jan_Tschichold
11. http://en.wikipedia.org/wiki/Sans-serif
12. http://en.wikipedia.org/wiki/X-height
13. http://en.wikipedia.org/wiki/Matthew_Carter
14. http://www.will-harris.com/verdana-georgia.htm

X-height is the distance between the baseline and midline — a measure of lowercase character height. (Image: adactio[15])

Having dispensed with the subject of preference, let's cover some important technical issues relating to one's choice of typeface.

The first thing to consider when choosing a Web font (read: `@font-face` font) is the breadth of the family. Does the font include all of the necessary bold, italic (or even better, semi-bold and bold-italic) styles? One style is fine for headings, but paragraphs need greater variety. Without these variations at your disposal, not only will your text look insipid, but the lack of proper emphasis will make your writing difficult to follow.
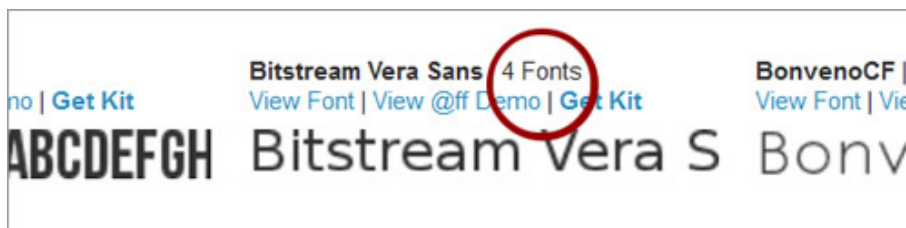
I personally don't like Bitstream[16], but it is fully functional for paragraph text

With the full gamut of stylistic variations at your disposal, you will not have to rely on the unsatisfactory "faux" styles that are applied to a regular font when `font-style: italic` or `font-weight: bold` is called. Typefaces are not designed to be contorted in this way. Using the prop-

---

15. http://www.flickr.com/photos/adactio/5739318101/sizes/l/in/photostream/
16. http://www.fontsquirrel.com/fonts/Bitstream-Vera-Sans

er styles provided by a family like Bitstream (above) will make your typography not only more attractive but more accessible: dedicated italic glyphs have a much clearer intent than text that is simply "leaned over a bit."

The trick is to make sure that the declaration of, for example, `font-style: italic` requests the italic resource rather than triggers the faux style. It should be as effortless as using a system font family such as Georgia. This is probably best explained, like so many things, in commented code. For brevity, we'll set up just a regular font and an italic (not bold) style variation.

```css
@font-face {
    /* Change this to whatever you like. */
    font-family: 'MyWebfont';
    /* The "regular" font resource. */
    src: url('mywebfont-regular.ttf') format('truetype');
    /* Associates values of "normal" with this resource. */
    font-style: normal;
    /* As above for weight. */
    font-weight: normal;
}


@font-face {
    /* The same as in the above block; the family name. */
    font-family: 'MyWebfont';
    src: url('mywebfont-italic.ttf') format('truetype');
    /* Associates values of "italic" with this resource. */
    font-style: italic;
    /* ... It's not a bold-italic font style. */
    font-weight: normal;
}


body {
    /* Provides a system font fallback. */
    font-family:'MyWebfont', georgia, serif;
}


em {
/* If @font-face is supported, the italic Web font is used. If
not, the italic Georgia style is lifted from the user's computer.
Either way, a faux style is not allowed to creep in. */
    font-style: italic;
}
```

Our second typeface consideration relates to rendering. Some fonts, replete with beautiful glyphs and exceptional kerning[17] as they may be, simply don't render very well at small sizes. You will have noticed that embedded fonts are often reserved for headings, while system fonts (such as Verdana[18] here) are relied on for body text.

One of the advantages of Verdana is that it is a "well-hinted" font. Delta hinting[19] is the provision of information within a font that specifically enhances the way it renders at small sizes on screen. The smaller the font, the fewer the pixels that make up individual glyphs, requiring intelligent reconfiguration to keep the font legible. It's an art that should be familiar to any Web designer who's ever tried to make tiny icons comprehensible.

Hinting is a tricky and time-consuming process, and not many Web fonts are hinted comprehensively. Note the congealed upper portion of the bowl[20] in the lowercase "b" in the otherwise impressive Crimson[21] font, for instance. This small unfortunate glitch is distracting and slightly detracts from a comfortable reading experience. The effect is illustrated below and can be seen in context as a demo[22].

Slightly unsatisfactory hinting for the Crimson Roman style. I love Crimson all the same.

The good news is that, as font embedding becomes more commonplace, font designers are increasingly taking care of rendering and are supplying ever better hinting instructions. Typekit[23] itself has even intervened by manually re-hinting popular fonts such as Museo[24]. Your best bet is to view on-page demonstrations of the fonts you are considering, to see how well they turn out. Save time by avoiding, sight unseen, any fonts with the words "thin" or "narrow" in their names.

---

17. http://type.method.ac/
18. http://en.wikipedia.org/wiki/Verdana
19. http://www.fontshop.com/glossary.php?def=Delta%20hinting
20. http://www.fontshop.com/glossary.php?ltr=b
21. http://www.fontsquirrel.com/fontfacedemo/Crimson
22. http://www.fontsquirrel.com/fontfacedemo/Crimson
23. http://blog.typekit.com/2011/01/20/new-and-updated-fonts-from-exljbris/
24. http://new.myfonts.com/fonts/exljbris/museo/

FONT SIZE AND MEASURE

As a recent Smashing Magazine article[25] compellingly attests, you put serious pressure on readability by venturing below a 16-pixel font size for paragraph text. All popular browsers render text at 16 pixels by default. This is a good enough indication (given the notorious tendency among browser makers to disagree) that 16 pixels is a clear standard. What's more, the standard is given credence by an equivalent convention in print typography, as the article points out.

We often express `16px` as `100%` in the declaration block[26] for the `body` in our CSS reset style sheets. This makes perfect sense, because it is like saying, "100% the same as the browser would have chosen for you." If you want the paragraph text to be bigger than 16 pixels, just edit this value in the `body` block using a percentage value that equates to a "whole pixel." Why whole pixels? Two reasons. First, whole numbers are less ungainly and are easier to use as multipliers in style sheets. Secondly, browsers tend to round "sub-pixel" values differently, giving inconsistent results[27]. An 18-pixel font size expressed as a percentage is `112.5%` (1.125 × 16).

Normalizing the size of default text (or "paragraph text," if you're being good and semantic) in such a way is extremely important because it sets us up to use ems[28] as a multiplier for the size of surrounding headings and other textual elements. For instance, to render an `h3` heading at 1.5 times the font size of the paragraph, we should give it the value of `1.5em`. Because ems (pronounced as in "Emma," not E.M. Forster) are relative units, they change according to the default font size. This makes it much easier to maintain style sheets and, more pertinently, ensures that the perceived importance of headings is not increased or diminished by adjusting the size of the paragraph text.

The "measure" is the number of characters in a line of text. Choosing a comfortable measure[29] is important for usability, because if lines are

25. http://www.smashingmagazine.com/2011/10/07/16-pixels-body-copy-anything-less-costly-mistake/
26. http://www.blooberry.com/indexdot/css/syntax/declaration.htm
27. http://ejohn.org/blog/sub-pixel-problems-in-css/
28. http://en.wikipedia.org/wiki/Em_%28typography%29
29. http://webtypography.net/Rhythm_and_Proportion/Horizontal_Motion/2.1.2/

too long, then scanning back to find the start of the next line can be awkward. Without conscious effort, the reader might miss or reread lines. In *The Elements of Typographic Style*[30], Robert Bringhurst puts a good measure at somewhere between 45 and 75 characters. It is the main reason why we use the `max-width` property when designing elastic layouts.

Whatever your page's ideal maximum width, it is likely much narrower than what you are used to seeing. According to an in-depth study[31] of typographic design patterns published on Smashing Magazine, the average website exhibits a measure of 88.74 characters, far exceeding the optimal range.

LEADING AND VERTICAL RHYTHM

(Image: andrechinn[32])

Leading[33] (pronounced "ledding") is the spacing between consecutive lines of text. Leading has a similar impact on readability as "measure," because it helps to define and demarcate the rows of glyphs that one must traverse from left to right and back again. The trick with leading is to avoid adding too much: text with lines that are too far apart ap-

---

30. http://www.amazon.com/Elements-Typographic-Style-Robert-Bringhurst/dp/0881792063/ref=sr_1_1?ie=UTF8&qid=1317888735&sr=8-1
31. http://www.smashingmagazine.com/2009/08/20/typographic-design-survey-best-practices-from-the-best-blogs/
32. http://www.flickr.com/photos/andrec/6067541623
33. http://en.wikipedia.org/wiki/Leading

pears fragmented, and the intent of a judicious use of leading is undone by a negative result.

In mechanical typesetting, leading was set by inserting strips of lead metal (hence the pronunciation) between lines. In CSS, the `line-height` property is the tool we use, and exposure to it is much less likely to make you go mad.

Instead of accounting for space between lines, as with leading, `line-height` is a vertical measure of the whole line — including the text itself and any spacing to follow. There are three ways to set it: the wrong way, the redundant way and the right way.

The wrong way to set `line-height` is in pixels. By introducing an absolute value, we would undo all of the good work from the previous section. As the `font-size` increases (either in the style sheet or the user's browser settings), the `line-height` would persist. This produces an interesting effect:
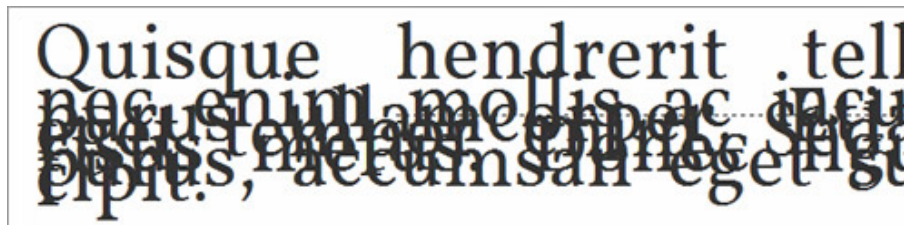
Absolute and relative values do not mix.

So, should we use the same em multipliers that we saw in the last section? This is the redundant way. Although the method would work, providing an em unit is not necessary. Rather, a value of `1.5` for the `line-height` that is 1.5 times that of the font size will suffice. The `line-height` property belongs to an exclusive club of CSS properties that accept unit-less numeric values.

```
p {
    font-size:; /* Silence is golden; implicity 1em. */
    line-height: 1.5;
}
```

The attentive among you will have noticed that so far I have only mentioned font sizes that are even numbers. The reason is that I favor a line height of `1.5`. So, a font size of 18 pixels means lines with a height of 27 pixels or, if you prefer, lead strips that are 9 pixels thick. Using even numbers is another bid to maintain whole pixel values — I know that any even number multiplied by 1.5 will result in a whole number. A `line-height` stated in whole pixels is particularly important, because it is the key value used to achieve "vertical rhythm."

All components on the page should have a height divisible by the height of one line of paragraph text.

Maintaining vertical rhythm (or composing to a baseline grid[34]) is the practice of making sure that the height of each textual element on the page (including lists, headings and block quotes) is divisible by a common number. This common number (the single beat in a series of musical bars, if you will) is typically derived from the height of one paragraph line. You should be able to see by now why an impossible value like 26.5 pixels would be a mistake for such an integral measure.

The benefits of vertical rhythm to readability are much subtler than those of hinting, measure or leading, but they are still important. Vertical rhythm gives the page decorum. Because we have made sure so far that all of our measurements are co-dependent and relative, altering the font size for the body (all the way up at the top of the cascade[35]) will not damage the page's vertical rhythm.

It is worth noting that, although a line height of 1.5 is fairly dependable, not all fonts are made equal. Fonts with a tall x-height or long descenders[36] might benefit from more generous, separative leading. When basic readability is at stake, adopting a more complex vertical rhythm algorithm is worth it.
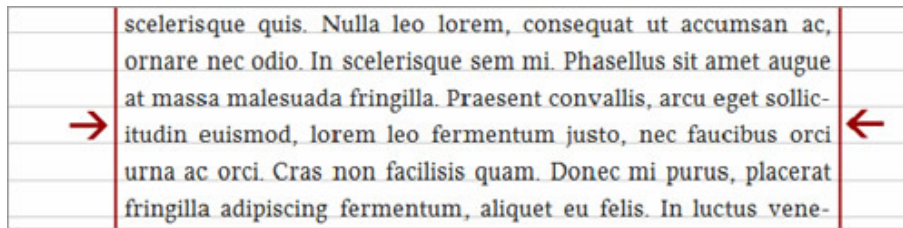
WORD SPACING AND JUSTIFICATION

Without intervention, paragraph text on Web pages is set "ragged right" (`text-align: left` in CSS): the start of each line is flush with the left margin, but the lengths of the lines vary, giving an uneven "ragged"

34. http://www.alistapart.com/articles/settingtypeontheweb
35. http://reference.sitepoint.com/css/cascade
36. http://www.fontshop.com/glossary.php?ltr=d

effect on the right side. If you're like me, you prefer the tidiness of full justification[37] (illustrated below). But implementing justification without impairing readability is not as straightforward in HTML as it is by using desktop-publishing software.

> scelerisque quis. Nulla leo lorem, consequat ut accumsan ac, ornare nec odio. In scelerisque sem mi. Phasellus sit amet augue at massa malesuada fringilla. Praesent convallis, arcu eget sollicitudin euismod, lorem leo fermentum justo, nec faucibus orci urna ac orci. Cras non facilisis quam. Donec mi purus, placerat fringilla adipiscing fermentum, aliquet eu felis. In luctus vene-

Fully justified text necessitates, arguably, a narrower measure than text set ragged right.

The problem with justified text in HTML (`text-align: justify`) is word spacing. In print media (such as newspapers), hyphenation is used to break up long words. This results in more components (words or part words) per line, thus improving distribution and curbing aggressive word spacing. Browsers do not hyphenate automatically, and the soft hyphen[38] (`­`) is implemented inconsistently. Besides, imagine having to manually insert `­` all the way through your copy. CSS3's `text-justify`[39] property, which aims to give us more control over `text-align: justify`, could ease the problem by enabling inter-character distribution. Bizarrely, it is currently available only with Internet Explorer.
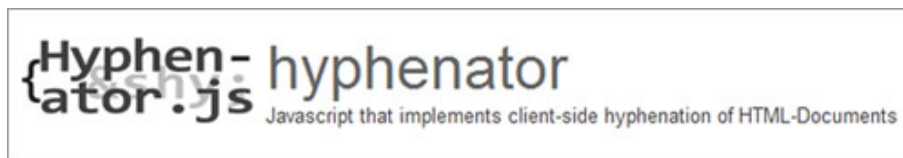
Hyphenator.js[40]

Thankfully, Firefox and Safari now support[41] the CSS3 property `hyphens`, which can automatically insert hyphens much as you would manually with `­`. In fact, you can revert to manual hyphenation in a document set to `hyphens: auto` by using the `hyphens: manual` over-

---

37. http://en.wikipedia.org/wiki/Justification_%28typesetting%29
38. http://en.wikipedia.org/wiki/Soft_hyphen
39. http://www.css3.com/css-text-justify/
40. http://code.google.com/p/hyphenator/
41. http://blog.fontdeck.com/post/9037028497/hyphens

ride. Browser prefixes are required, but Lea Verou can help[42] you with that.

Until the other browsers catch up, a consistent cross-browser solution is currently possible only with JavaScript. Hyphenator.js[43] is a powerful tool that takes a library of syllabic patterns specific to each language and uses them to dynamically insert soft hyphens in the correct places. It is a bit pricey (two scripts at a total of 72 KB uncompressed just for the English implementation), but it does work. Its effect is shown in the first screenshot for this section.

FINISHING TOUCHES (STYLING CONTEXTUALLY)

Now that we have dealt with the important business of sizing, setting and distributing our paragraphs compellingly, you may wish to apply a few small enhancements and decorations for the purpose of signposting the document. These nuances concern only certain paragraphs, and choosing which paragraphs to set off is a question of context[44]. With the help of special selectors and combinators, we are able to target specific paragraphs depending on where they appear on the page, making sure that the difference in their design is consistent with their intended role and meaning.

Paragraphs separated with a margin (such as `margin: 0 0 1.5em;` *) do not require indentation. With paragraphs, margins and indentation serve the same purpose.*

Let's use indentation[45] as an introductory example. Although less common in Web typography than in print, indenting the first line of each paragraph is a conventional method of grouping paragraphs into chunks of information. In terms of rhythm, it is also a sort of punctuation: the reader is invited to pause briefly before each paragraph. Because no indentation is required for the first paragraph — why pause before we've even started? — we should exclude this paragraph from

42. http://coding.smashingmagazine.com/2011/10/12/prefixfree-break-free-from-css-prefix-hell/
43. http://code.google.com/p/hyphenator/
44. http://robertwalkeronline.com/2008/09/22/importance-of-context/
45. http://en.wikipedia.org/wiki/Paragraph#Paragraph_breaks

our CSS rule. Using the adjacent sibling combinator[46], we are able to target only paragraphs with a preceding paragraph, and so the convention that has been familiar to book typography over the centuries is ably reproduced.

```css
p + p {
    text-indent: 1.5em  /* I like to keep the indentation length
equal to the line height. */
}
```

In the next example, I have combined the adjacent sibling combinator with the `:first-letter` pseudo-class[47] to create a pronounced introductory glyph or "elevated cap":

```css
h1 + p:first-letter {
    font-size: 2em;
    line-height: 0;
    /* The line-height must be adjusted to compensate for the
    increased font size, otherwise the leading for the overall line
    is disrupted. I find that any values below 0.4 work. */
}
```


The beauty of adding these refinements (many more of which are demonstrated by Jon Tan[48], including "drop caps" and "outdents") contextually is that they are activated only when semantically meaningful and appropriate. For instance, our `:first-letter` style above is appropriate only for introductory copy. Because the introductory paragraph is always (in this particular schema) preceded by an `h1` heading, we have a way to bind its style to its particular role in the document's flow. In other words, we can honor its meaning through its design.

As long as we rigorously adhere to semantic HTML[49], we can employ many nuances that are impervious to both the rearrangement of the page itself and the introduction of dynamic content.

---

46. http://eriwen.com/css/css-adjacent-sibling-selectors/
47. http://css-tricks.com/5762-pseudo-class-selectors/
48. http://jontangerine.com/silo/typography/p/#continuous
49. http://en.wikipedia.org/wiki/Semantic_HTML

Conclusion

(Image: primatage[50])

Walking down an aisle in a library, I no more than glance at the vast majority of books shelved on either side of me. Only a madman would suggest that my disregard of these books should sanction their pages being torn out. Nonetheless, because research has shown[51] that visitors don't read the average Web page in full, and because the "success" of a page is more easily measured by user action than cognition, we are often encouraged to marginalize our writing in favor of visual signifiers or action cues.

Sure, most people will "bounce" your content, but if you really have something to say, don't alienate the people who are willing to give your writing a chance. Good typography does justice to your words, and good wording does justice to your ideas. If readers are comfortable reading your type, then they will more likely be comfortable with what you are writing about. ❧

---

50. http://www.flickr.com/photos/primatage/4590087621/sizes/l/in/photostream/
51. http://www.useit.com/alertbox/percent-text-read.html

Mind Your En And Em Dashes: Typographic Etiquette

BY DAVID KADAVY ❧

An understanding of typographic etiquette separates the master designers from the novices. A well-trained designer can tell within moments of viewing a design whether its creator knows how to work with typography. Typographic details aren't just inside jokes among designers. They have been built up from thousands of years of written language, and applying them holds in place long-established principles that enable typography to communicate with efficiency and beauty.

Handling these typographic details on the Web brings new challenges and restrictions that need to be considered. Below are a few rules of thumb that will have you using typography more lucidly than ever before.

Setting Body Copy

Good typography comes down to communicating information, and the basis of information is good old-fashioned body copy – simple blocks of text. Here are a few ways to make your blocks of text nice and clean.

INDENTATION OR SPACE AFTER A PARAGRAPH?

When signalling the end of a paragraph and the beginning of another, you can generally either indent or insert a space between the paragraphs. Doing both is redundant and creates awkward, irregular chunks of white space.

INDENT   SPACE AFTER PARAGRAPH
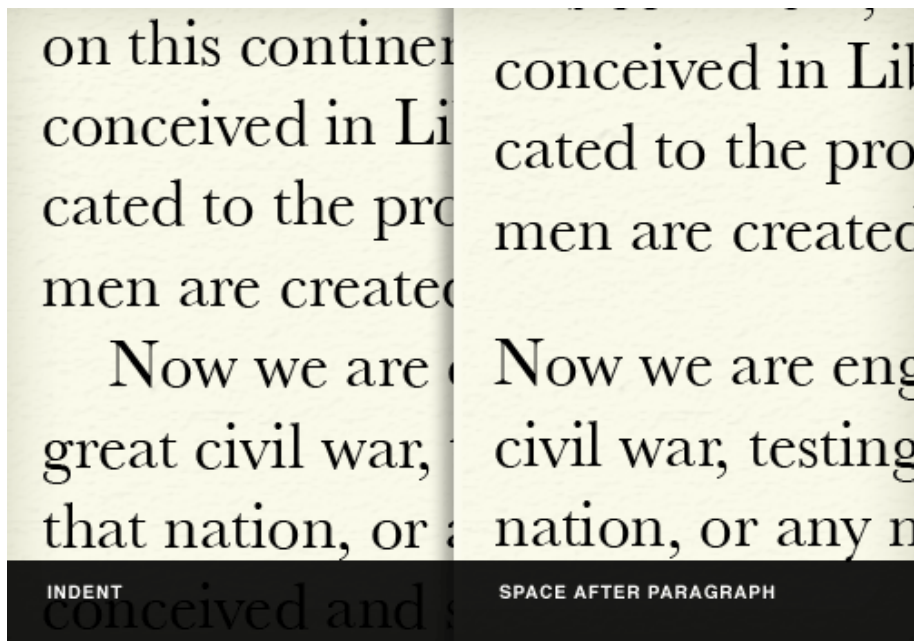
Indentation

Indent the first line of a new paragraph about 1 em (if your font size is 12px, then that would amount to 12 pixels). Indenting the opening paragraph of a new section would be redundant, because that paragraph would be the first in that page or section anyway.

Space after paragraph

A full line break of 1 em (like when you hit the "Return" key twice) is generally more than enough to signal a new paragraph. About 0.8 ems is sufficient.
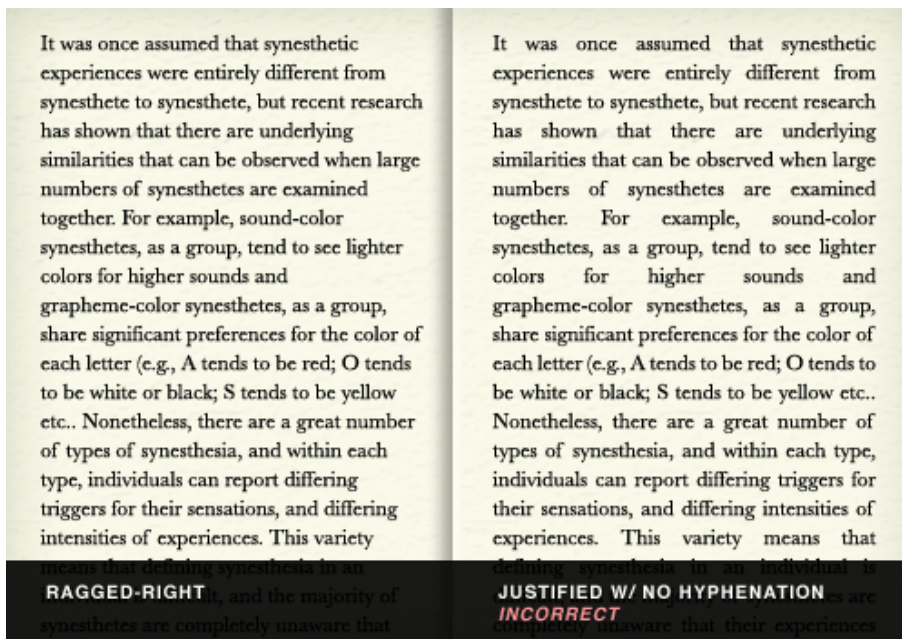
Text set flush left (or ragged-right) creates an even texture and is easier to read. Because the Web allows for less typographic control, justified type usually creates rivers and big gaps in blocks of text.

Flush left (ragged-right)

In text set flush left, the lines will break naturally, resulting in a "ragged" right side of the column. This is generally considered easier to read because the eye can better distinguish one line from the next, because of the different line lengths. Additionally, studies have shown that readers score higher in comprehension and recall when text is flush left. Hyphenation should not be used with text that is flush left, because that would defeat the whole point of a ragged right.

Avoid justified type on the Web

Justified text is spaced out so that each line is of the same length. Because all lines have to be set at the same length, some variation in letter spacing (i.e. the space between letters), and word spacing (the space between words) is done to lengthen and shorten lines. This changes the overall texture of the body copy, making it less even.

High-end desktop publishing applications (such as Adobe InDesign) have sophisticated systems for justifying type so that it doesn't produce "rivers" (large gaps in the text from too much word spacing). They use a combination of letter spacing, word spacing, hyphenation and glyph scaling (i.e. very slightly adjusting the width of individual letters) to produce an even texture in the body copy.

But there is no hyphenation control in CSS. Beyond JavaScript libraries that handle hyphenation[52] and CSS3's `text-justify` property, browsers don't have very sophisticated measures for maintaining an

even texture in blocks of text. This makes justified text on the Web highly susceptible to big holes and rivers in the body text.

Avoiding justified type on the Web is generally a good idea. At the optimal line length for reading (about 50 to 80 characters or 8 to 15 words), making lines of text all the same length, without hyphenation or subtle letter spacing, nearly always results in rivers.

One Space After a Period. Not Two.

Very few typographic debates match the intensity of whether there should be one or two spaces after a sentence.

This practice is a holdover from typographers of the Victorian era. The practice began when typewriters were introduced. Because early typewriters used monospaced fonts (meaning that the spaces between letters were always the same length), typists used two spaces to mimic the slightly wider spaces that typesetters were using after periods. Now that computers (and the fonts on them) are equipped with proportionally spaced (rather than monospaced) fonts, this practice is outdated, and it creates awkward breaks in the body text.
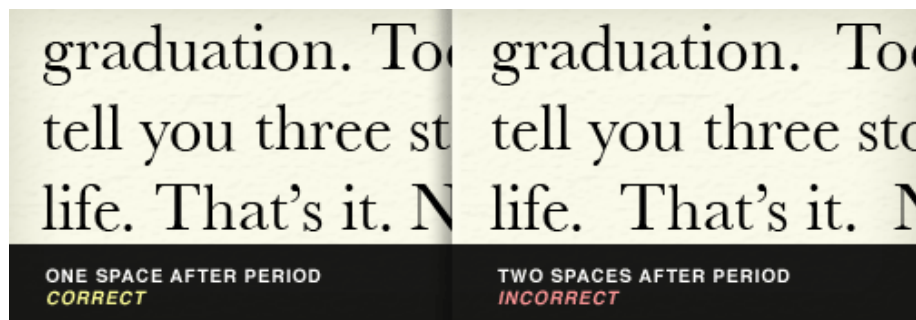
One space after a period is plenty to signal the end of a sentence. It also keeps the text block even.

For those who are still skeptical, consider this: the beginning of a new sentence is already indicated by a period, a full space and a capital letter. Adding yet another space merely pokes a hole in the body text and interrupts the flow of reading.

The typist may wish to continue using two spaces after a period, but the typographer should not.

---

52. http://carlos.bueno.org/2010/04/sweet-justice.html

Use Dashes Dashingly

Most fonts are equipped with at least two dashes: an en dash (–, `–`, which is the width of a lowercase "n") and an em dash (—, `—`, which is the width of a lowercase "m"). Don't confuse these with the hyphen (-), which isn't a dash at all but a punctuation mark.

THE HYPHEN

Hyphens and dashes are confused to the point that they are now used almost interchangeably by some. Some fonts, such as Adobe Garamond Pro, retain hyphens in their original form; those hyphens look more like the diagonal stroke of a calligrapher's pen than a straight horizontal line. You'll also often see hyphens used as a replacement for a minus sign; however, a longer character is available in some fonts for this purpose.

Although the hyphen does look quite a bit like a dash or minus sign, it is a punctuation mark. It should be used primarily to hyphenate words in justified type. On the Web, this isn't much of a concern because, as mentioned, there is no standard hyphenation control in browsers. The hyphen should also be used in compound modifiers (such as "fine-tuned"), to join digits in phone numbers, and in a few other rare cases (covered in detail on Wikipedia[53]).

THE EN DASH AND THE EM DASH

In *The Elements of Typographic Style* – which is the unofficial bible of the modern typographer – Robert Bringhurst recommends that dashes in text should be the en dash flanked by two spaces. This is much less visually disruptive than using the em dash with no space — which is recommended in editorial style books such as *The Chicago Manual of Style* — because there is less tension between the dash and the characters on either side of it.

53. http://en.wikipedia.org/wiki/Hyphen

Why go against *The Chicago Manual of Style* in this case? The reason is that style manuals are concerned mostly with punctuation, not typography. An en dash surrounded by spaces achieves the same effect as an em dash with no spaces, but typographically it is less disruptive. This was a big debate between my editor and me when I was writing my book[54].

The practice of using two hyphens for a dash is a holdover from the days of typewriters. Besides being visually disruptive to smooth blocks of text, it is now unnecessary with the richer character sets that are available to typographers.

The en dash is also used to indicate ranges of numbers (such as "7–10 days"), although it isn't flanked by spaces in this case.

Use Smart Quotes, Dummy

The quotation mark that is produced when you type " on your computer is not a true quotation mark (unless your word processing or page layout program has automatic formatting). True quotes are sometimes called "smart quotes," and the right one must be used according to whether you are opening or closing a quotation. To open a quotation, use " (`“`), and to close a quotation, use " (`”`). For opening single quotes, use ' (`‘`), and for closing single quotes and apostrophes, use ' (`’`). CMS' such as WordPress usually either convert dumb quotes to smart quotes automatically or have plug-ins for the job.
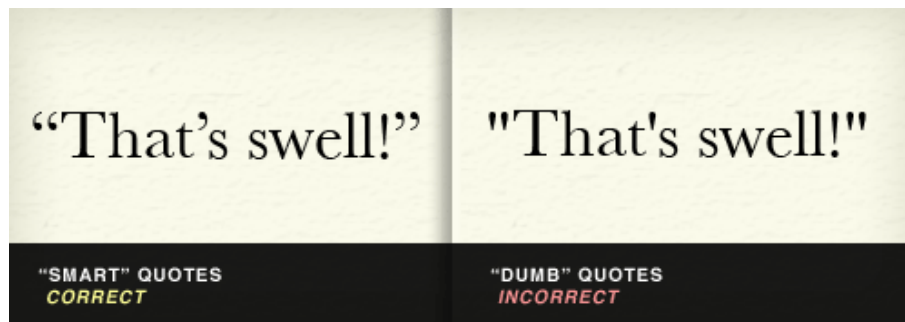
SMART QUOTES IN ACTION

CNN uses dumb quotes, which look clunky and lifeless, while The New York Times uses smart quotes, which look clean and sophisticated:

---

54. http://www.amazon.com/Design-Hackers-Reverse-Engineering-Beauty/dp/1119998956
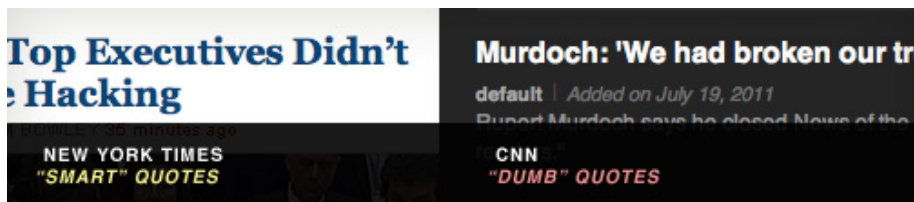
Smart quotes are so sexy that their sinuous forms can be used as ornamentation, such as in this pull quote on Jason Santa Maria's blog[55], which features smart quotes only in shadow form:

Quotation Marks Don't Measure Up

The ″ and ' on your keyboard are also often used to designate feet and inches (Steve Jobs is 6′ 2″). This, too, is technically incorrect. For feet, you should use ′ (`′`), and for inches, ″ (`″`). (Steve Jobs is, in fact, 6′ 2″.)

Note that many desktop publishing programs are so advanced that they automatically (and incorrectly) insert "smart" apostrophes and quotation marks in all of these situations. This can be changed in the program's settings, but the quickest way around this is simply to copy and paste the correct mark from a "less sophisticated" text editor.

---

55. http://jasonsantamaria.com/articles/on-good/

+1 For Proper Math Symbols

Sometimes in our haste, we use the wrong marks for math symbols. For example, as we learned above, a hyphen is not a minus sign (−, `−`). Additionally (no pun intended), neither an x nor an asterisk (*) is a multiplication symbol (×, `×`).

Proper fractions not only add elegance to typography, but are more readily recognizable. Using a slash in your fractions (1/3) is not ½ (`½`) as cool as using the proper vulgar fraction (as it's called).

Certain other symbols are harder to substitute with common characters, such as when you write 98¢ (`¢`) and 104° F (`°`). You might be tempted to just type these out as "cents" and "degrees."

There's Always More With Ellipses

Sometimes, there's more than meets the eye. Most of us type the ellipsis as three periods (...), but most fonts come equipped with their own character (…, `…`) that is slightly more spaced out. *The Chicago Manual of Style* would have you think that the proper ellipsis consists of periods separated by spaces (. . .), but any typographer knows that is too disruptive to body text. Some typographers prefer to set their own ellipses by using thin spaces (` `) between periods (. . .). But your judgment is best; pick the ellipsis treatment that makes your text block look most consistent (which will never be with full spaces between periods.)

Accent Marks And Other Diacritics Aren't Passé

Sure, in the US you can work on your resume while sitting in a cafe, and it wouldn't be much different than if you were working on your résumé in a café in France. No matter how naïve you are, sometimes using the proper diacritical mark is… critical. A diacritic is a mark placed near (usually above or below) a letter to change its sound. The most common diacritic in English is the acute (´) accent, which is used in a number of words borrowed from other languages and which is sometimes very important, such as for the sake of declaring your passion for saké.

ACUTE ACCENT   GRAVE ACCENT

The acute accent is often confused with the grave accent (`), which leans to the left and is readily available on US keyboards below the tilde (~). The grave accent is actually much less common in English than the acute accent. If you ever see one again, you might think you were having a déjà vu.

To insert an acute accent on Mac OS X, hold down Option while typing "e," and then type the letter that you want to appear below the accent (so, if you want "é," you'd be typing "e" twice). Play around with Option + u, i and ~, and a whole world of diacritics will open for you.

Ligatures Bring Letters Together

In the early days of printing, when type was set in lead, the lead slugs on which characters were set made it difficult to set certain character pairs close enough. For example, in the letter combination "fi," the top terminal of the "f" stuck out so far that the letter couldn't be set close enough to the "i" because of the dot on the "i." Thus, many fonts (usually the classic ones, such as Adobe Garamond) have ligatures for certain pairings that actually meld the letterforms together. Some modern fonts, such as Helvetica, also have ligatures, but their effect is negligible.

Below, you can see that ligatures are needed in some fonts more than others. Notice how an "fi" ligature is more critical to clean typog-

raphy in Adobe Garamond than it is in Helvetica. Because of this, different fonts come with different ligatures.
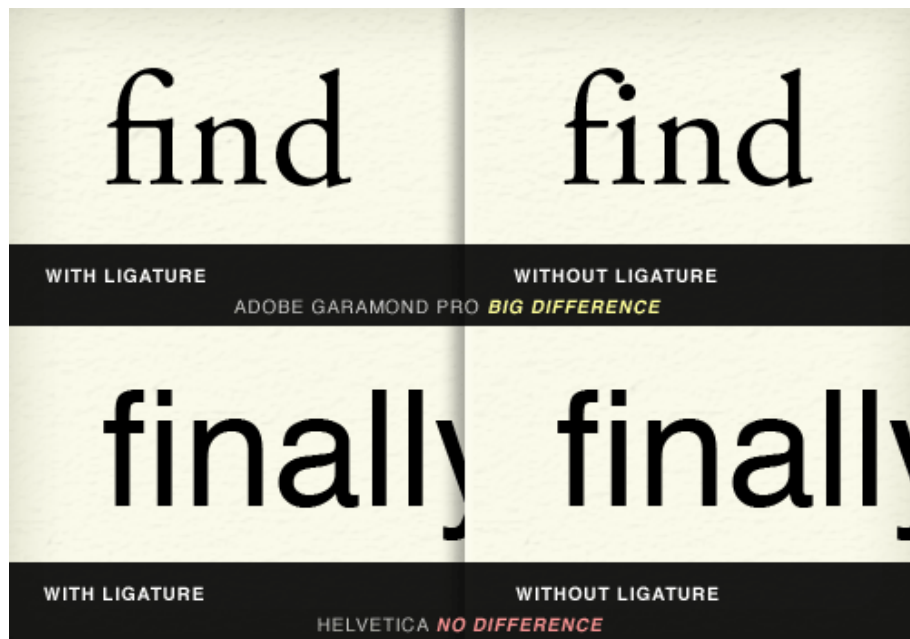
In addition to "fi," ligatures are most commonly needed for the groupings "fl," "ff," "ffi" and "ffl." Below you can see that Apple makes beautiful use of the "ffl" ligature in the type treatment of "iPod shuffle."
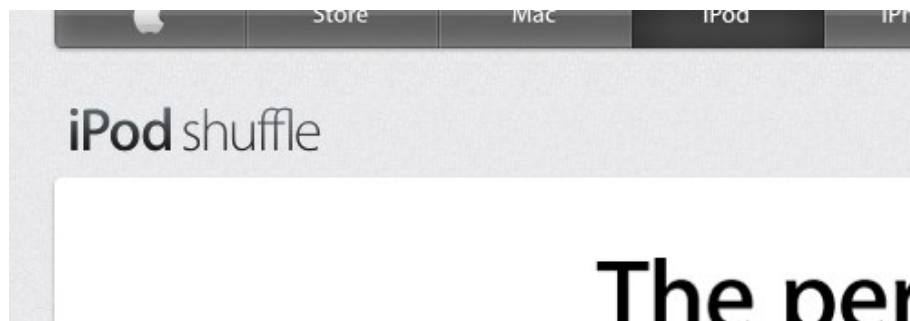
USING LIGATURES PRACTICALLY

Obviously, typing in ligatures every time you need them is impractical. Even if you did, your content might not ultimately be displayed in a font that is equipped with the appropriate characters. Fortunately, plugins such as jQuery's Ligature.js will automatically insert the most commonly supported ligatures where appropriate.

But, be careful: even if the proper characters are available, you might make it difficult for users to search for text on the page. Currently, Internet Explorer and WebKit-based browsers such as Chrome and Safari match ligatures to their corresponding letter pairs when the search command is used (for example, when a user searches for "fig" on the page), whereas Firefox does not.

CSS3's `font-variant-ligatures` property displays the proper ligatures without interfering with the actual HTML code of the page, thus preserving full functionality of the search command in all browsers. In this case, while Internet Explorer, Chrome and Safari do not currently support this property, Firefox does.

So, if you apply ligatures at the content level, then your Firefox users won't be able to use the search command for them. And if you use `font-variant-ligatures`, only your Firefox users will see the ligatures. Because of the spotty browser support for ligatures, ignoring ligatures altogether in the body copy would not be unreasonable. A lack of ligatures in big headlines and headings might be more obvious, though, so picking your preferred method of support there might be worthwhile.

Being Reasonable

Obviously, keeping track of and implementing some of these typographic details might seem pretty tedious. Fortunately, most Web frameworks and CMS' have plug-ins that take care of them for you, and some of CSS3's typography controls are at least beginning to be supported in browsers.

But as with all matters of design and production, economy comes into play. You may decide that implementing some of these details just isn't worth the trouble. What's important is at least *knowing* typographic etiquette, so that you can rely on it when it matters and make wise decisions that are appropriate for your project.

Other Resources

You may be interested in the following articles and related resources:

- How to Type[56]
  Learn how to actually type these characters on your computer.

- *The Elements of Typographic Style*[57], Robert Bringhurst
  The Amazon ordering page for the unofficial bible of modern typographers. This book is full of great advice on any typographic detail you can imagine.

---

56. http://www.howtotype.net/
57. http://www.amazon.com/Elements-Typographic-Style-Robert-Bringhurst/dp/0881791326

- The Elements of Typographic Style Applied to the Web[58]
An adaptation of Robert Bringhurst's classic book for the Web.

This chapter is based upon a chapter from David's book, Design for Hackers:
Reverse-Engineering Beauty[59]. ❧

---

58. http://webtypography.net
59. http://www.amazon.com/gp/product/1119998956

How To Choose The Right Face For A Beautiful Body

BY DAN REYNOLDS ☙

What is it that makes a typeface into a text font, instead of a font for larger sizes? The answer differs slightly, depending on whether one aims for print or Web-based environments.

Nevertheless, there are certain features that most good text faces have in common. Familiarity with these helps to select the right fonts for a given project. This chapter presents a few criteria to help the process along.

Some of today's most successful typefaces were designed to excel in very specific areas of use: Frutiger grew out of airport signage, Georgia and Verdana were among the first mass-market fonts created for on screen reading, FF Meta was conceived as a telephone book face, and even the Stalwart Times New Roman was tailored for the pages of the London Newspaper *The Times*. Many typefaces are also often fine-tuned for using in certain sizes.

It should be noted that in this chapter, when "text" is mentioned, it is in discussion of *body* text, or *running* text (in other words, text at a similar size to what you are probably reading right now, rather than much larger sized words).

Features Of A Good Text Typeface

The features outlined in this chapter are those that *type designers* keep in mind while developing new typefaces. It's important to realize that these aspects of typeface design are different from the text treatment a graphic designer employs while laying out a book page or website — no matter what a typeface's inherent rhythm and niceties are, setting a text is still something that must be done with great care in respect to readability. There are problems that good fonts themselves cannot solve — whether or not a text sings on the page or screen depends on factors like the width of the column, the amount of space between each line, the contrast between the foreground/background and a number of other factors.

Above, Bembo over the years: this typeface was a favorite of many book designers throughout the 20th century. At the top of the image is a scan of the original Bembo type-face, printed with letterpress. The digital version of the typeface — Bembo[60], seen in the middle, is too light for ideal text in print. A newer digitization was published in 2002 — Bembo Book[61], seen at the bottom. This font is much darker, and is a better repre-sentation of the original Bembo idea. However, the middle version is still very elegant, and may still be used well in sub-headlines.

Every typeface has its own inherent rhythm, created by the designer who made the font. With typefaces that are intended for use in body text, it is primarily *this* rhythm that will make the typeface readable. But there are additional factors that go into the making of a good text face: the space between the letters, the degree of contrast in the letters' strokes, as well as the x-height and relative size of the whitespace inside of the letters. Not every typeface that works well in text will apply all of these factors in the same way, but all good ones will have many of these features in common.

---

60. http://www.fonts.com/findfonts/detail.htm?productid=215078
61. http://www.fonts.com/findfonts/detail.htm?productid=215077
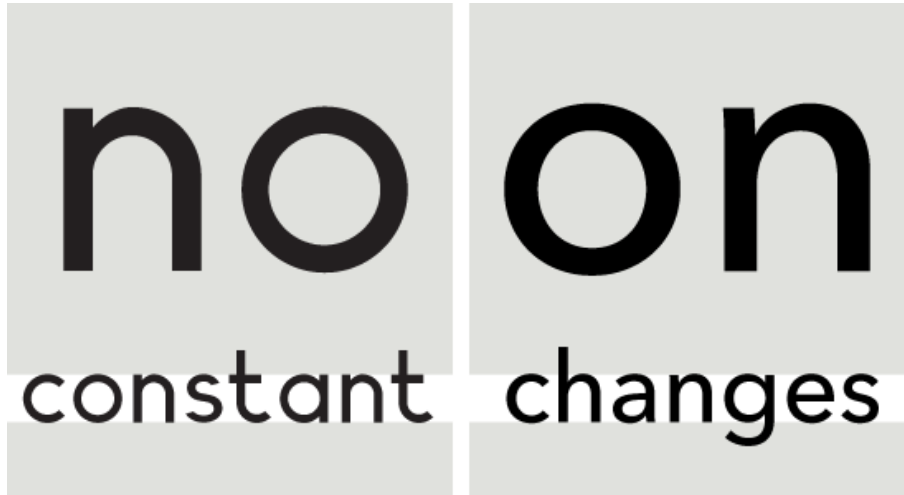
1. STROKE CONTRAST

When it comes to typefaces, the term "monolinear" is used to describe letters that appear to be designed with a consistent stroke thickness. Monolinear typefaces are low-contrast typefaces. Stroke contrast can be a helpful feature in small text sizes, but it is not paramount that a text face appears to be monolinear. Indeed, many newspapers employ high-contrast fonts; the question that must be considered is just *how thick* the thin strokes in high-contrast typefaces are.
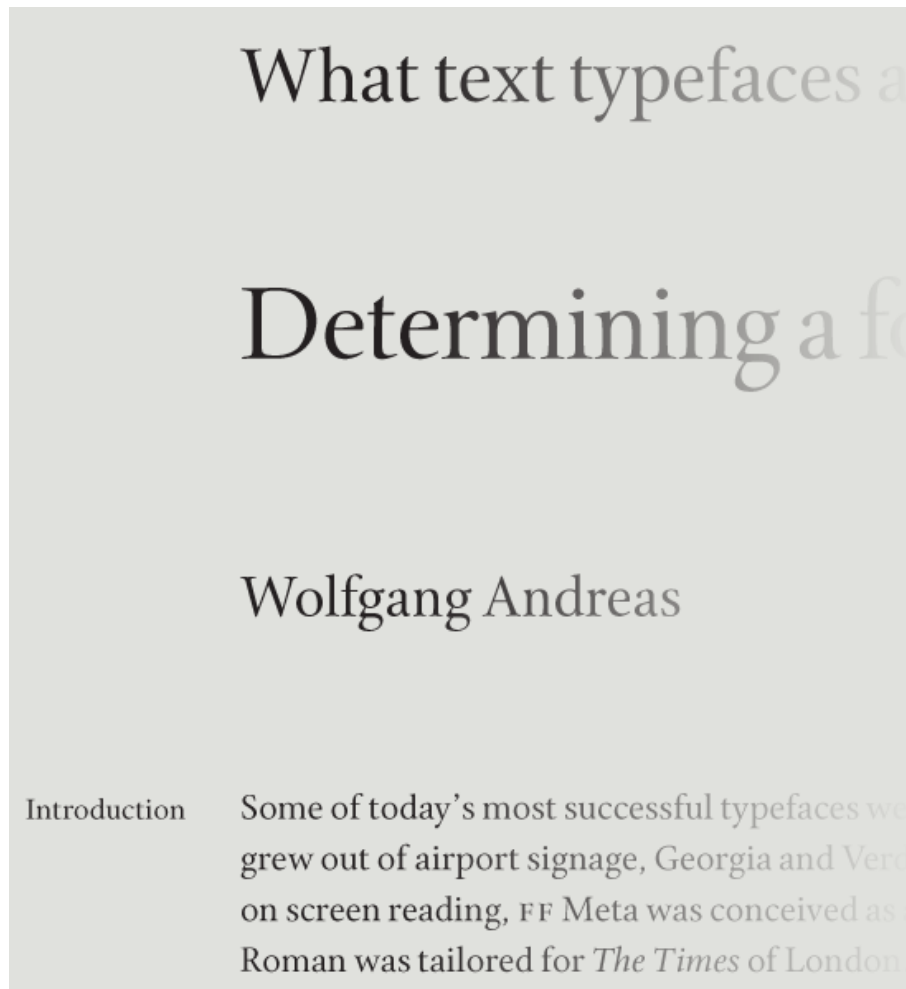
The images in this section show different ends of the contrast spectrum: the Cycles[62] types shown above are serifed, with a good deal of contrast. Sumner Stone's Cycles typeface is an excellent choice for book design as its letter forms combine clarity with a rather high degree of stroke contrast and an almost timeless appearance. Five separate "versions" of Cycles are used in the above image; each block of text is set in its own optically-sized font.

---

62. http://www.stonetypefoundry.com/cyclesoverview.html

Below, Avenir Next — also a great text face — is from another style of letter, and has very little contrast. I wouldn't split good typefaces up into *good contrast* and *bad contrast* groups. Rather, some typefaces have a degree of contrast — be it too high or too low — that makes them less suitable for use in text. There is no definite rule on how much or how little contrast impacts a text face's legibility. However, it is clear that both *no contrast* and *excessive contrast* can have adverse effects.

Geometric sans serif typefaces often appear to be monolinear stokes; their letters seem not to have any stroke contrast. In order to achieve this effect to the max, type designers have always made slight optical corrections. To look monolinear, a geometric sans needs some degree of thinning. In the image above, Planeta (left) is compared with Avenir Next (right). Both typefaces are more recent additions to the geometric sans category than stalwart faces (like Futura), or classic display designs (like ITC Avant Garde Gothic). Planeta has no visible stroke contrast, which must be a conscious decision on the part of its designer. While this does give it a unique style, it makes the face less suitable for text than Avenir Next, which is actually not as monolinear as it appears at first glance.

2. OPTICAL SIZES

The Garamond Premier[63] typeface family features different versions of each font. These variants are tailored for use in a certain size range. Above, the Display font (left) is compared with the Caption font (right). The Display font is optimized for texts that will appear in very large point sizes, while the Caption font has been optimized for very small text.

In her book *Thinking with Type*, Ellen Lupton writes:

> "A type family with optical sizes has different styles for different sizes of output. The graphic designer selects a style based on context. Optical sizes designed for headlines or display tend to have delicate, lyrical forms, while styles created for text and captions are built with heavier strokes."

The intended size of a text should be considered when selecting the typeface: is the typeface you want to use appropriate for the size in which you need to set it? Does the family include optical sizes (that is, different versions of the typeface that are tailored specifically for use at different sizes)? As with each of the factors mentioned in this chapter, the size at which a font is set can make or break your text.

In many ways, it is easiest to see the qualities necessary for good text faces by comparing potential selections with "display" faces. Like the term "text," "display" refers to the size at which a specific font may best be used. In print media, as well as in many screen and mobile-based applications, the term "display" is often analogous with "headlines." If a typeface that you are considering looks more like something

---

63. http://store1.adobe.com/cfusion/store/html/index.cfm?store=OLS-US&event=displayFontPackage&code=1738

that you might like to use for a headline, it won't be the best choice for body text.

In the comparison image below, the Garamond Premier Display font has a tighter rhythm than the Caption font — not as much space is necessary between letters when they are set in large point sizes. Why should one consider type families with optical sizes, anyway? Well, as users bump up the point size of digital fonts, the space between letters increases in equal proportion. This inter-letter space slowly becomes too large, and makes a text feel like it is breaking apart. When a proper text font is set large, it may require some tighter tracking. Typeface families that offer optically-sized variants of their styles play a helpful role here.

Stanley Morison
Good-looking, as he remained; dressed in black, as he was to the end of his life. His voice low-toned
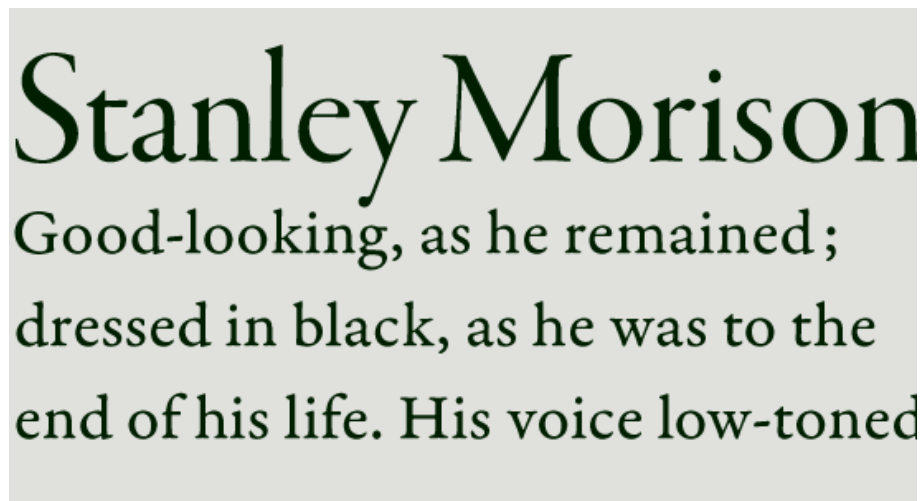
In the image above, the first line of text — "Stanley Morison" — is set in the Garamond Premier Display font, while the lines of text underneath it are set in Garamond Premier Caption. Each font is balanced for its size, and they also harmonize well with one another. In another image (below), these fonts have been switched: the headline is now set in the Garamond Premier Caption font, and the text in the Garamond Premier Display. The letters in the Caption face look too clumsy when they are set so large, while the Display fonts' letters appear uncomfortably thin in a "text" setting.

The amount of stroke contrast visible in caption-sized fonts is much lower than in display-sized fonts. If the Garamond Premier Display font (from the above image) was rendered in a smaller point size, its thin strokes would begin to break apart, making the text unreadable. But this would not occur with the Caption version.

Garamond Premier Caption can robustly set real text, even in poor printing conditions. How well a font will render in small sizes *on screen* depends on the operating system and applications in question. Font formats themselves also play a role; in certain environments, TrueType fonts with "hinting" information may vastly improve on screen display (see the "Hinting" section at the end of this chapter).

3. X-HEIGHT

Garamond Premier's Display face (above left) is shown next to the Caption face (above right). Both fonts are set at the same point size. The Caption face features a much higher x-height than the Display font.

Many successful text faces feature high x-heights; this means that the ratio of the central vertical area of lowercase letters — the height of the letter x, for instance — is large when compared to the length of the ascenders and descenders. Depending on its design, a text face may have a low x-height and still be quite legible. But the benefit of incorpo-

rating a large x-height in a design is that it maximizes the area of primary activity.

A high x-height may also prevent some letters, like the a or the s, from appearing to become too dark; these two letters have three horizontal strokes inside the x-height space, which is a very small area in text sizes. In order for letters to maintain clarity and understandability, they must have a consistent rhythm, as well as include large, open forms.

4. THE SPACES INSIDE OF LETTERS

The images below illustrate just a few of the intra-letter spacing elements that should be understood and considered when choosing which typeface to choose for your body text. In order for the white spaces inside of letters to remain visible in small sizes, it is necessary for their counterforms to have a certain minimum mass, proportionally.

Counters

The image above shows text set in two members of the ITC Bodoni[64] family: ITC Bodoni Seventytwo and ITC Bodoni Six typefaces. In the first line, "Randgloves" is set in a size mastered for 72pt display (ITC Bodoni Seventytwo), while "and jam" is in the Caption size (ITC Bodoni Six). These words are reversed in the second line. Note how the enclosed white space in the top portion of the e changes between the display and text optical sizes.

---

64. http://www.fonts.com/findfonts/detail.htm?productid=662864

Apertures

"Apertures" are the gateways that whitespaces use to move in and out of the counterforms of a typeface's letter. The above image highlights the wide apertures in four letters from Erik Spiekermann's FF Meta[65] typeface. These allow for the typeface's letterforms to feel more open. In certain sizes and settings, wide apertures — and the large counterforms that are their result — will make a text more readable.
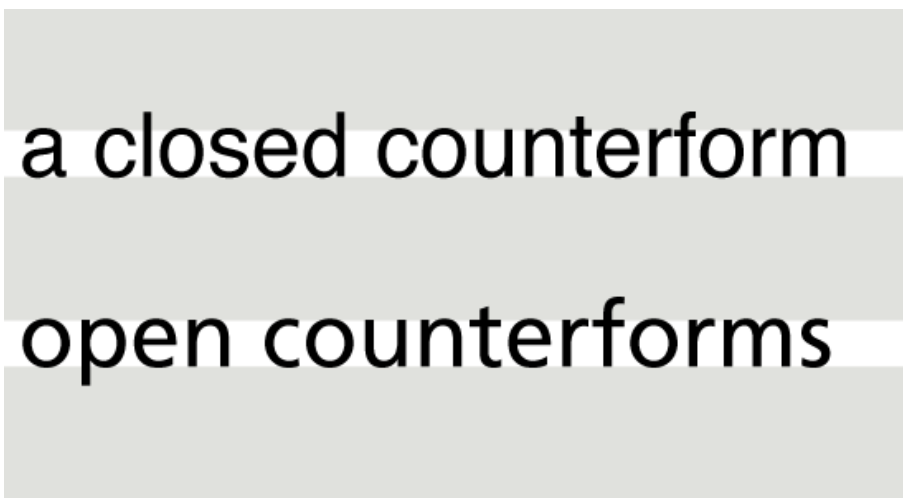
The top line of the image above is set in Helvetica[66], and the bottom line in Frutiger[67]. While the counterforms inside the letters of these two typefaces are similar in size, Helvetica's apertures are much smaller. Because of this, white spaces inside of Helvetica's letters and between Helvetica's letters are much more closed off from each other than in a typeface with more open counters — like Frutiger.

---

65. https://www.fontfont.com/fonts/meta
66. http://www.linotype.com/526/Helvetica-family.html
67. http://www.linotype.com/2534/frutiger.html

Other counterforms and problematic letters worth remembering include the c; if the apertures of a, e, s are very open, the c should follow this same route. Then there are lowercase letters like a, e, g, s that often have rather complex shapes — specifically, they each feature several horizontal strokes inside a small amount of vertical space. How do their forms relate to one another? How large is the typeface's x-height? Do the ascenders and descenders have enough room, particularly f and g? Do the counterforms inside of roundish letters (e.g., b, d, p, q, o) have the same optical size and color as those inside of straight-sided letters like, h, n, m, and u? How different from one another are the forms of the capital I, the lowercase i and l, and the figure 1? Can the 3 and the 8 be quickly differentiated from each other? How about the 5 and the 6?
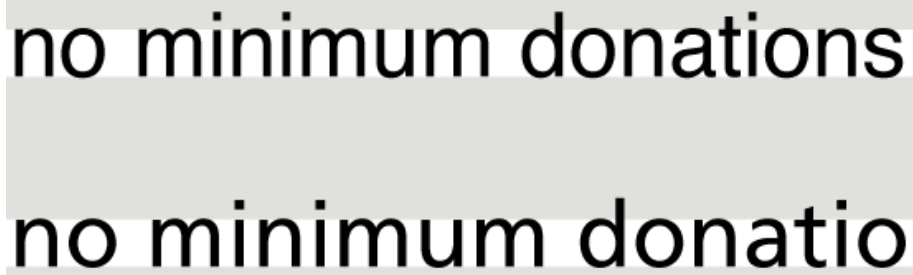
5. KERNING
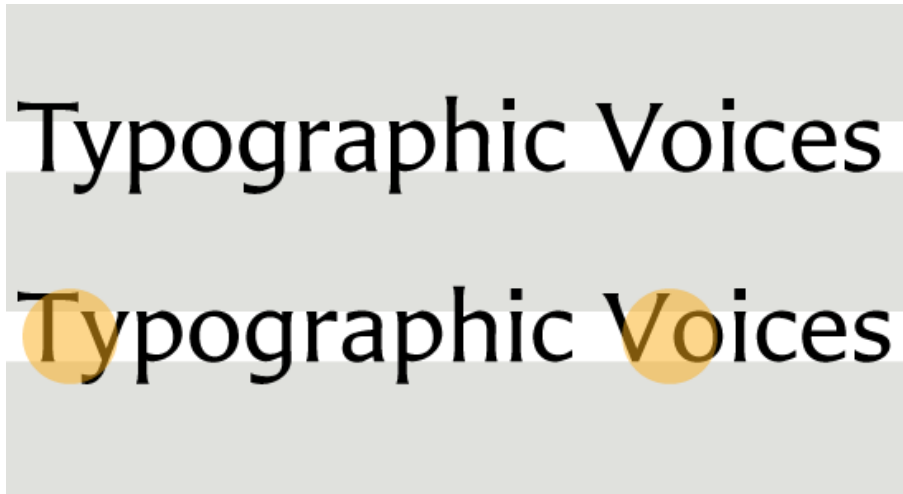
In the sample above, kerning has been deactivated for the second line. The gaps between the letters Ty and Vo are too large when compared with the amount of space between the other letters in the text. The typeface used in the image is Carter Sans[68].

Despite the popular misuse of the term in graphic design circles, "kerning" does not refer to the spacing values to the left and right of the letters in a font. Rather, fonts contain a list of *kerning* pairs to improve the spacing between the most troubling lettering combinations. The importance of kerning in a font is the role it may play in maintaining an optimal rhythm. Just as kerning describes something much more specific than a typeface's overall spacing — or the tracking that a graphic designer might apply to a text — kerning is not the rhythm of a type-

---

68. http://www.linotype.com/759464/CarterSans-family.html

face itself, but an element that may strengthen a typeface's already existing rhythm. Not every typeface design requires kerning, and there are typefaces on the market that indeed may have too many kerning pairs — a sign that the basic letter spacing in the font could have been too faulty in the beginning.

6. CONSISTENT RHYTHM ALONG THE LINE

no minimum donations

no minimum donatio

In the image above, compare the spaces between the letters of the Helvetica typeface (first row) with Frutiger's (second row). Frutiger is a more humanist design, featuring a slight diagonal axis in its letters; many of them look similar to Helvetica's, at least at face value. However, the space between Helvetica's letters is much tighter.

While most of the images in this chapter feature typeface families that include Optical Size variants, many commonly used typefaces on the market today do not offer these options. This is why it is helpful to be able to identify text typefaces based on their features, rather than just on their names in the font menu. As mentioned earlier, it is primarily the typeface's rhythm that dictates the readability of a block of text.

Take Frutiger and Helvetica, which are both commonly used in text, especially for corporate communication — Neue Helvetica is even the UI typeface in iOS and MacOS X 10.7. Yet, despite its popularity, Helvetica is not very effective as a text typeface; its rhythm is too tight. By rhythm, I'm not referring to tracking — or any other feature that a designer can employ when typesetting — but the natural flow of space between letters, and within them as well. Frutiger is a much more open typeface — the spaces between its letters are closer in size to the white spaces inside of the letters than in the case of Helvetica. Like all good text typefaces, Frutiger has an even rhythm — space weaves in and out of the letters easily.

7. CAVEAT: SIGNAGE FACES

To round off my discussion on text typefaces, I'd like to briefly mention some fonts that are often shown in rather large sizes: fonts for signage. Interestingly, many signage typefaces have design features very similar to typefaces created for very small applications. The Frutiger typeface, based on letters that Adrian Frutiger originally developed for the Roissy airport in Paris (now named after Charles De Gaulle), is quite legible in small sizes precisely because it is a good signage typeface. Despite their size, signage fonts serve a rather different purpose than Display fonts.

Additional Elements To Consider
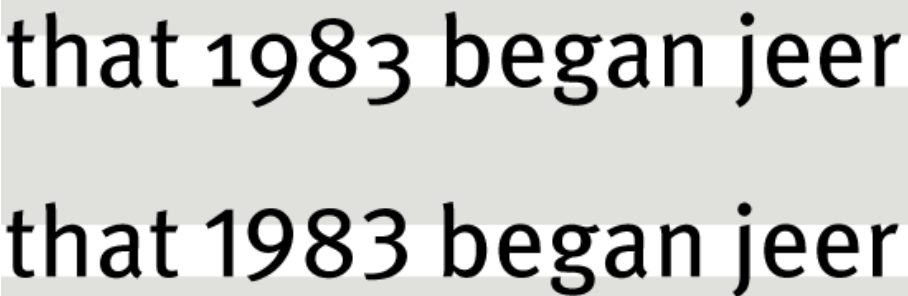
After considering the criteria mentioned above, the next question that often comes up is, "does this font have oldstyle figures, or small caps and ligatures, etc.?" A font's letters might look really great in text, but if they do not include additional elements and features, their use is somewhat minimized. I avoid using fonts with small character and feature sets where I can, because I feel that the lack of these "extras" may break the kind of rhythm I aim to achieve.

1. OPENTYPE FEATURES

Once you've established a consistent rhythm by setting your text according to the correct size and application, it would be a pity to inadvertently break that flow. Large blocks of tall figures or capital letter combinations do just that.

Even in languages like German, where capital letters appear at the start of many words, the majority of letters in a text planned for immersive reading will be lowercase letters. Every language has its own frequency concerning the ratio of "simple" lowercase letters like a c e m n o

r s u v w x z to lowercase letters with ascenders or descenders — b d f g h j k l p q y. In international communication, language support is a key consideration when choosing a font, and other character set considerations may especially play a role.

that 1983 began jeer

that 1983 began jeer

Traditionally, the style of figures used in running text also have ascenders and descenders. These figures — often called *oldstyle figures* or *text figures* — harmonize better with text than the "uppercase" lining figures. These so-called *lining figures* either align with the height of a typeface's capital letters, or are slightly shorter. It is no surprise that, when shipping the Georgia fonts for use onscreen and online, Matthew Carter and Microsoft made the figures take the oldstyle form. Many other typefaces that have long been popular with graphic designers, like FF Meta (seen above), also use oldstyle figures as the default style. In my opinion, lining figures are best relegated to text set in all-caps.

Long all-caps acronyms — like NAFTA, NATO, or USSR — also create an uncomfortable block in the line for the reader. Setting these letter-strings in small caps helps reestablish a specific typeface's natural rhythm in reading sizes, as may be seen in the first line of the image below (set in Erik Spiekermann's FF Meta).

Along with common ligatures like `fi ff fl`, small caps and the many figure options are the most common OpenType features found in quality text fonts. Aside from having both lining and oldstyle figures, OpenType-functionality can enable a font to include both tabular and proportionally-spaced figures, numerators and denominators for fractions, as well as superior and inferior figures for academic setting. Additional OpenType features (such as contextual alternates or discretionary ligatures), are more powerfully noticed in display sizes, and in some cases can even be distracting in text.

2. HINTING

The display of text on screen, particularly on computers running a version of the Windows operating system, may be fine-tuned and improved with the help of size-specific instructions inside of the font file. These instructions are commonly referred to as "hints." A TrueType font (or a TrueType-flavored OpenType font), is capable of including hinting. However, not every font manufacturer goes to the effort of optimizing the onscreen appearance of its fonts for Windows — even those fonts specially created for use in text sizes.

andere, höhere   andere, höhere   andere, höhere

All of the text in the above image is shown in the same font: Prensa[69], set at 18 pixels. The lowest row shows this at actual size in three different onscreen rendering environments. In the enlargements, the top row shows a close-up of rendering in Safari on MacOS X, which ignores the hinting data in fonts. The second row shows rendering in Internet Explorer/WindowsXP (Grayscale only, for this sample). The third row is from a ClearType environment — in this case, from Firefox on Windows7. Prensa is a typeface designed by Cyrus Highsmith at the Font Bureau; the Web font[70] is served by the Webtype service.

Recommended Typefaces For Readability

Aside from the typefaces already mentioned in this chapter and its images, here is a small selection of faces that I personally enjoy at the moment. Even though lists of "favorite" typefaces are about as useful as lists of favorite songs or favorite colors, I am happy to pass my subjective recommendations along. No doubt that as new projects arise, my list of favorites is likely to change, too. I do think that these typefaces serve as great starting places. Some are also just from cool friends whose work I dig. Alongside each selection, I mention whether this choice is currently available for print only, or if there is a Web font ver-

69. http://www.fontbureau.com/fonts/Prensa/
70. http://www.webtype.com/font/prensa-complete-family/

sion, as well. Don't forget: the typefaces that you pick in the end should depend on your projects, their audience, and the content at hand.

Arnhem

Arnhem[71] is a no-nonsense high-contrast oldstyle-serif face. It is a contemporary classic for newspaper and book setting, designed by Fred Smeijers and distributed via OurType. Available for print and Web[72].

Benton Sans

Benton Sans[73] is a Tobias Frere-Jones performance of Morris Fuller Benton's News Gothic genre. Designed for Font Bureau, it is not only a great typeface for small print in newspapers, but one of the best-rendering text faces for the Web as well. Available for print and Web[74].

Ibis

Ibis[75] is another Font Bureau typeface, designed by Cyrus Highsmith. This square serif family is also no stranger to cross-media text-setting. Ibis works just as well whether you use it in print or on screen. Available for print and Web[76].

Ingeborg

Ingeborg[77] is modern serif family from the Viennese type and lettering powerhouse, the Typejockeys. Like any proper family should, Ingeborg has optically-sized variants for text and display settings. The display versions of the typeface can get pretty far out, too! Designer Michael

---

71. https://ourtype.com/#/try/pro-fonts/arnhem/
72. http://www.extensis.com/en/WebINK/fonts/font.jsp?family=Arnhem%20Pro
73. http://www.fontbureau.com/fonts/BentonSans/
74. http://www.webtype.com/font/bentonsansre-family/
75. http://www.fontbureau.com/fonts/IbisText/styles/
76. http://www.webtype.com/font/ibisre-family/
77. http://www.typejockeys.com/fonts/Ingeborg

Hochleitner named this typeface after his mother. Available for print and Web[78].

Ludwig

Fred Smeijer's work in contempory type design is so significant that he gets *two* shout-outs in my list. His Ludwig[79] type family takes a nod from 19th century grotesques, but he does not try to sanitize their quirky forms, as so many type designers had tried to do before him. Available for print and Web[80].

Malabar

This is one of the typefaces that I've designed. I'm somewhat partial to Malabar[81]. Available for print and Web[82].

Scala Sans

Martin Majoor's FF Scala Sans[83] has been my top go-to typeface for almost 15 years. It mixes well with the serif FF Scala type, but it's also really great on its own. Available for print and Web.

URW Grotesk

Of all the typefaces designed by Hermann Zapf over his long career, URW Grotesk[84] is clearly the best. Unfortunately, it has been a little overlooked. URW Grotesk is a geometric sans, with a humanist twist that brings much more life into the letters than this genre usually allows for. Plus, the family is super big. Available for print and Web.

---

78. http://www.typejockeys.com/specimen/ingeborg.html
79. https://ourtype.com/#/try/pro-fonts/ludwig/
80. http://www.extensis.com/en/WebINK/fonts/font.jsp?family=Ludwig%20Pro
81. http://www.linotype.com/521506/Malabar-family.html
82. http://webfonts.fonts.com/en-US/Project/Choose-Fonts?fontQuery=malabar#keyword%3Dmalabar%26page%3D1
83. https://www.fontfont.com/fonts/scala-sans
84. http://www.myfonts.com/fonts/urw/grotesk/

Weiß-Antiqua

Are you a DIY-fan? Do you like to print with letter press, whether you set your own type by hand, or have polymer plates made? Then check out the typefaces of Emil Rudolf Weiß! His Weiß-Antiqua[85] is an eternal classic. Weiß may have passed away 70 years ago, but his work is still relevant. He was German, so his last name is sort of pronounced like *Vice,* as in *Miami Vice.* Available for print and Web.

Conclusion

There are many factors that play a role in typeface selection. Aside from just browsing through the available fonts that they have, or fonts that could be newly licensed for a project, designers regularly spend considerable effort determining *the right typeface* to complement a project's content, or the message at hand. Understanding some of the thoughts that go into the making of text typeface — including how a typeface's letters are fitted to each other to determine a text's default underlying rhythm — helps lead to better informed decisions regarding what types are indeed apt, and which faces are better suited for other sorts of jobs. After having read this chapter, I hope you feel more comfortable with this kind of decision making, and that you will know what to look for with a font in the future.

Other Resources

For more information about choosing the right text fonts, you may be interested in the following books and Web resources: ❧

1. WEBSITES

- Better Web Typography With OpenType Features[86]

- Figuring Out Numerals[87]

- Figuring Out Numerals — The Sequel[88]

---

85. http://www.myfonts.com/fonts/urw/weiss-antiqua/
86. http://www.opentype.info/blog/2010/08/14/better-web-typography-with-opentype-features/
87. http://fontfeed.com/archives/figuring-out-numerals/
88. http://fontfeed.com/archives/figuring-out-numerals-the-sequel/

- The `@font-face` Rule And Useful Web Font Tricks[89]

- The Making Of FF Tundra[90]

- What Should I Look For In A UI Typeface?[91]

- Making Geometric Type Work[92]

2. BOOKS

- Counterpunch: Making Typefaces In The Sixteenth Century, Designing Typefaces Now[93]

- Size-Specific Adjustments To Type Designs[94]

- Thinking With Type[95]

89. http://www.smashingmagazine.com/2011/03/02/the-font-face-rule-and-useful-tricks
90. http://www.ilovetypography.com/2011/10/05/the-making-of-ff-tundra
91. http://www.design-by-izo.com/2011/10/18/what-should-i-look-for-in-a-ui-typeface/
92. http://typographica.org/2010/on-typography/making-geometric-type-work/
93. http://www.hyphenpress.co.uk/books/978-0-907259-42-8
94. http://www.markbattypublisher.com/books/typography-monographs-volume-2-size-specific-adjustments-to-type-designs-an-investigation-of-the-principles-guiding-the-design-of-optical-sizes/
95. http://www.thinkingwithtype.com/

Why Subtle Typographic Choices Make All The Difference

BY CAROLYN KNIGHT & JESSICA GLASER ❧

A strong understanding of how designers control meaning is essential for anyone interested in graphic design or typography. In this chapter, we'll look at the reasons *why* subtle typographic changes can create considerable effect. We'll refer to one or two linguistic and semiotic examples, as well as design case studies, to get to grips with why subtle changes can make all the difference.

Let's consider a couple of simple sentences: "The boy walks a dog" and "The boy walks the dog." The meanings are significantly different simply by the change of one small word. "A dog" is any old dog, while "the dog" is one we know and recognize. Similarly, small changes in typography can fundamentally alter impact and interpretation. For example, type size can be increased, the weight or font can be changed, and positioning within a frame altered; with each alteration, the meaning also changes. An authoritative, urgent, big, bold "STOP" suddenly becomes more lighthearted and less weighty and might even come across as teasing when rendered as "Oh, stop, stop it! I like it!"

Two visualizations of the same word but with typographic treatments that have entirely different emphases and meaning. (Credit: Bright Pink Communication Design)

The last couple of examples stem from the ideas of one of the fathers of 20th-century linguistics, Ferdinand de Saussure, who felt that each word (or "sign") had one specific meaning (signification). Applying this concept to typography, de Saussure's principles imply that each typeface and design choice carries a predetermined meaning. Just as in the two sentences about "the" and "a" dog and the two visuals showing different takes on "Stop," each has a significantly different meaning. (To read about de Saussure's ideas in depth, see *Course in General Linguistics*[96].)

However, meaning comes not simply from comparing one visual interpretation to another. As de Saussure suggests, meaning is also established by the context in which visuals are set. This idea, from 20th-century French philosopher Roland Barthes, is known as "secondary signification." Barthes notes that a visual interpretation (sign) does not have just one meaning, as might be deduced from de Saussure's work, but that a second socially and culturally specific meaning can be gleaned

96. http://www.amazon.com/Course-General-Linguistics-Ferdinand-Saussure/dp/0231157274

from the context in which the visual treatment appears. In Western society, you can find an example of this in the typographic styling of fashion brand Juicy Couture[97].

Juicy Couture

The name style of an iconic fashion brand.

The gothic letterforms are seen by the brand's target audience as being the height of fashion and desirability. To some, however, the same typeface in a different context would imply tradition, heritage, reputation and possibly even political persuasion. A more traditional use of this style of font can be found in the masthead of the UK newspaper The Telegraph[98].

The Telegraph

The online masthead of one of the UK's premier newspapers.

Type undoubtedly plays a role in the creation of meaning, but connotation does, too. For example, the distinct stylings of handwritten text can be deliberately used to bring a range of meanings to a message. On the page from Ben and Jerry's website[99] shown below, we can discern economy, personal concern and friendliness, but look at how the casual font gets you to drop your guard for just a moment. This allows Ben and Jerry's to talk about something as serious as citizenship, when you expect it to be talking about imaginative ice-cream flavors!

97. http://www.juicycouture.com/
98. http://www.telegraph.co.uk/
99. http://www.benandjerrysfoundation.org/

A page from the website of Ben and Jerry's charitable foundation.

In a 2010 UK Volkswagen ad campaign[100] designed by DDB UK[101], what appears to be slightly innocent, carefully hand-lettered text reinforces the message that the operating costs of a Volkswagen Golf car are 18% cheaper than the (unspecified) competition. This typographic approach also communicates the secondary yet equally important message that Volkswagen cars are affordable to purchase and that the company doesn't produce extravagant, expensive ads — all to keep prices low for customers.

---

100. http://www.flickr.com/photos/ddblondon2009/5146075159/in/set-72157625312654226/
101. http://www.ddbuk.com/

Ad for the Volkswagen Golf

Typography That Reinforces Meaning And Context

In their book *Reading Images: The Grammar of Visual Design*[102], Gunther Kress and Theo van Leeuwen confirm the role of the visual aspect of typography.

> *"The visual component of text is an independently organized and structured message, connected with the verbal text, but in no way dependent on it and similarly the other way round."*

Amazingly, we are conditioned to relate certain typographic styles, colors, shapes and patterns to certain products and situations. For example, you are not likely willing to clean your teeth with paste that comes out of a pack covered in brown, lightweight serif type. When aligned with our expectations, visual language can make a stronger impact than verbal meaning. Returning to our first example, the word "STOP" could now say "LOVE" and still be seen as an authoritative, urgent, big, bold warning sign.

---

102. http://www.amazon.com/Reading-Images-Grammar-Visual-Design/dp/0415319153/

A typographic treatment that asserts "LOVE" as a powerful, consuming emotion. (Credit: Bright Pink Communication Design)

The impact that can be achieved by designers being involved in the writing or editing of copy is dramatic. If we combine the eloquence of the copywriter with the knowledge and skill of the designer or typographer, we are able to achieve surprising yet totally cohesive solutions to design problems. Words carefully selected not only for their literal meaning but also for their sound, length and shape can help to "sculpt" a particular message, thus enhancing its value.

An example of this level of collaboration can be found in the promotional brochure for UK homebuilders Wimpey. As with most consumables, all new homes could be said to share many similar characteristics, and most homebuilders could be said to employ similar promotions, language and terminology. So, vendors need to be distinctive in order to create that must-have feeling.

The synergy of type and language in this UK home-buying brochure builds atmosphere and maximizes desirability. (Image: Bright Pink Communication Design)

This double-page spread features an extract from the poem "Spring" by Christina Rossetti. Its descriptive tone not only conveys the development site's proximity to the countryside and nature, but is also used as a pleasing metaphor for putting down roots and starting a new life.

The typographic treatment of the Rossetti poem emphasizes certain words both for their meaning and for their sound and shape. The highlighted words function as a group on their own, in addition to being integral to the poem as a whole. The poem allows the designer to be deliberate with the line lengths, with the words sitting comfortably together as a staggered group, complementing the other compositional elements on the page.

The result of collaboration between writer and designer is demonstrated in many of the promotional products of Ben and Jerry's. Ice cream flavors are given humorous names and descriptions, accompanied by energetic design and typographic exuberance. This careful approach reinforces the message that Ben and Jerry's is a speciality product — even handmade — produced with high-quality ingredients by caring individuals with a sense of humor and eye to all things fun.

There is a full range of Ben and Jerry's highly memorable ads in its online gallery (currently unavailable).

The ads, which are done tongue in cheek, focus on the ingredients in the ice cream and suggest that competing products might not have the same quantity and quality of ingredients that consumers expect. The illustrations capture the hand-drawn style mentioned earlier and also heighten the customer's anticipation — the cherries are drawn extra large and juicy; the nuts are varied and plentiful. The typography has been carefully selected to drive home these characteristics and get your taste buds moving and convince you that Ben and Jerry's is the only ice cream worth eating.

Of course, endless examples can be found of how subtle typographic choices make all the difference. Whether designers and writers work together or independently, fine-tuning their words and designs can lead to clever and effective results. And yet there is no doubt that understanding the implications of typographic details and the use of verbal language enables us to maximize communication. Effective design begets effective communication, and while major design decisions are obviously important, the little things often make all the difference.

A big "thank you" to our typography editor, Alexander Charchar[103], for preparing this chapter. ❧

103. http://retinart.net/

The Creative Way To Maximize Design Ideas With Type

BY CAROLYN KNIGHT & JESSICA GLASER ❧

As with most designers, being sure that we explore and select the most successful, memorable and stimulating designs is a vital aspect that underpins every project we undertake. For us, the beginning of a new challenge has never been as simple as asking ourselves what might be the best avenue to take and then sitting down at a computer and attempting to fulfill that idea.

After researching the subject matter, we will almost always begin with a sheet of paper and pencil and draw out a variety of design options to help bring together and develop the breadth of ideas that are maturing in our minds. In this chapter, we will explore the use of drawing and mark-making as an integral part of the creative process.
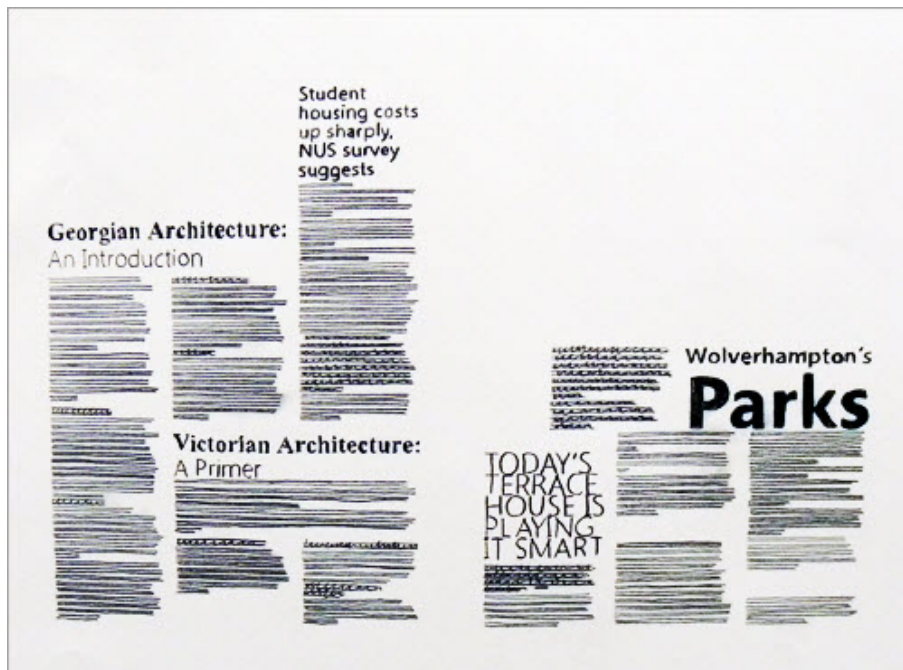
An example of mark-making that helps to formulate design ideas for working with type and image. Note the changes in mark-making that indicate different levels of type.

We have found that exploring design options on paper using drawing and mark-making is a great way to ensure that we are moving in the right direction with a project; plus, we don't think this working process

can be beaten for stimulating unexpected solutions that would otherwise have been very unlikely to see the daylight. We'll focus on different types of drawing and mark-making as problem-solving tools and skills; they form a vital part of visualizing and exploring design alternatives that involve quantities of type, with or without images.

Why Textural And Tonal Qualities Of Type Should Be Addressed In Drawing

Letterforms, lines of type and words come together with different tonal values as well as varying characteristics of patterning; depending on the darkness of tone generated, together with the scale and nature of texture, a viewer is attracted to a greater or lesser degree. Some great examples of this can be found by looking at the newspaper and magazine designs of Jacek Utko. Looking at all of the sample pages below, we are struck by the number of dynamic levels of text created by different typefaces, point sizes, weights and measures, as well as the imagery. The changing tonal values especially tempt and guide the reader through the pages in a particular sequence.

Working with art direction from Liudas Parulskis and Vilmas Narecionis, Jacek Utko has designed some captivating pages for the weekend section of Lithuanias Verslo Zinios. The layouts make dramatic use of textural and tonal diversity and also contrast of scale.

The textural and tonal qualities in layouts are used as much to help guide the audience in a particular order as for aesthetics. These qualities should be effectively captured through drawing and mark-making if the visual of a concept is to be sufficiently realistic to enable adequate design judgments to be made. This can be fairly easily achieved by using a relatively speedy design shorthand practiced by and familiar to many designers. Larger type is lettered in, capturing the stylistic essence, weight and proportions of the desired letterforms; text can be

lined, or "greeked," in using a mix of mark-making techniques, pens, pencils and/or varied pressure to indicate textural and tonal differences.

The examples featured in this chapter go only some of the way to demonstrating and capturing the infinite rhythms and varieties of typographic alternatives and combinations, but they do still demonstrate that even during early-stage drawing of visuals, capturing subtleties and changes of pace is essential. Drawing and mark-making styles can be developed to indicate and capture the textural and tonal difference that are present when working, for example, with all-caps sans-serif letterforms, as opposed to the very different visual "beat" that comes from uppercase and lowercase characters.

Finding The Right Marking Tools And Paper

In highlighting the refinement of this type of design drawing, we should also briefly comment on the tools that can be used to express these subtle typographic nuances. We work with a smooth lightweight paper that in the UK is called layout paper; the semi-transparent properties of this inexpensive material are great for tracing through from one sheet to another, making for speedy refinement of drawings. Many of the designers we speak to use a mix of mark-making tools, depending on the characteristics they wish to create; some work with marker pens, others prefer fiber-tipped fine-line pens, and some draw with soft pencils. The one aspect that these choices seem to have in common is that they enable the designer to vary the quality of the mark simply by varying the pressure: press hard to create a thicker, darker mark, and press more gently to create a lighter, finer tone.

Developing Your Own Design Shorthand

Returning to the description of this style of visualizing as being "relatively speedy," in reality, this process can be time-consuming, and while the results are not necessarily great in detail, this is thoughtful work that we certainly find to be the most time-efficient and creative way to work, particularly when confronted by completely new design challenges.

The drawing and mark-making in visuals that we are discussing here function on a number of levels. They capture alternatives of the textural and tonal details of layout; they are also a great way to explore different compositional alternatives; and they can be really helpful when used as templates or patterns to help streamline the process from marking to final design.

Typographic texture and tone will make different facets of a design more or less prominent. Choices of face, color, type size, tint, weight, inter-character spacing, line spacing and overall spatial distribution will affect the density of type and, consequently, the lightness or dark of the work. All of these aspects can be captured well with drawings and mark-making, affecting not only tonal values, but also the subtle textural qualities of type. Too often, visuals capture only the scale and position of type, without showing more detailed characteristics.
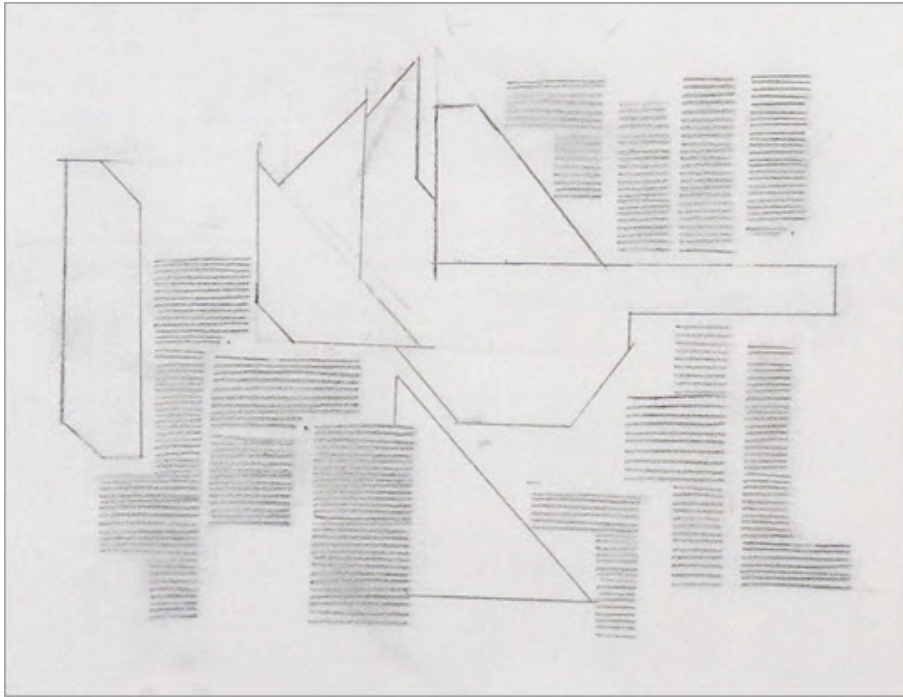
Aleksandrs Golubovs' layout deals with composition and does not show hierarchy or changes in typeface, point size, weight or leading. Minimal additional work with more varied mark-making would easily capture all of these missing elements.

The examples below demonstrate that we are not suggesting hugely time-consuming mark-making, but rather merely sufficient variety to convey a realistic and satisfying impression.
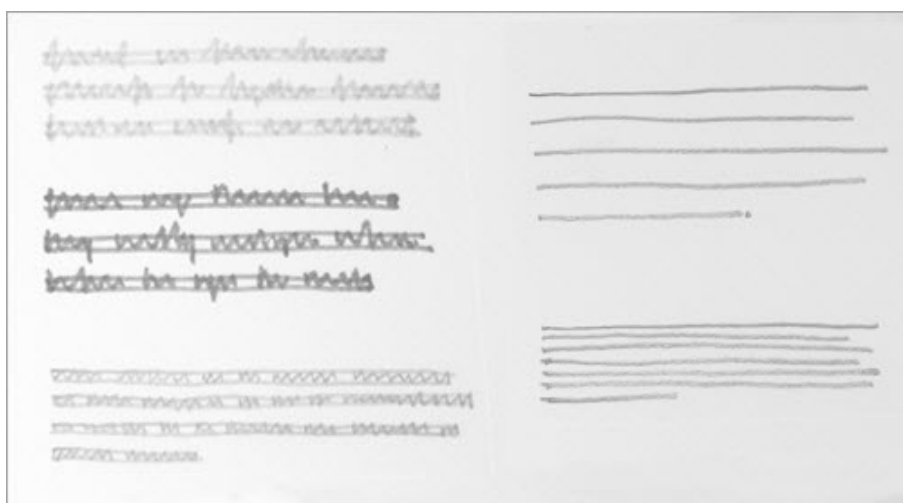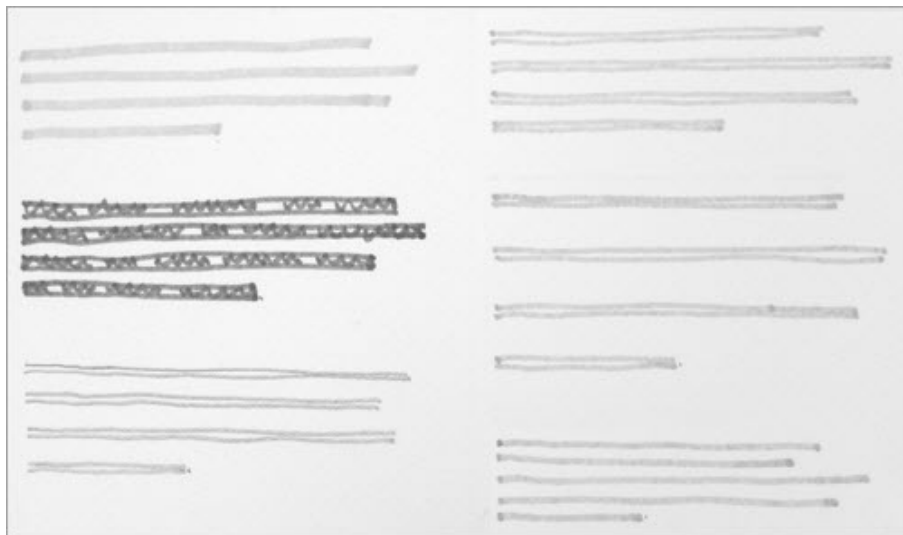
The visuals above are examples of different mark-making styles that demonstrate changes in weight, scale, leading and case. A mixture of marker, pen and pencil have been used here, but any tools may be used; ideally, though, tools that create marks of different quality in response to varied pressure are best.

Why Is Visualizing So Helpful To Designing With Type?

We take the view that using drawing and mark-making to produce visuals is an excellent way to develop ideas and extend design in order to explore creative possibilities. Drawing often pushes us into unexpected and exciting avenues of design that we might not have otherwise considered. An amazing link is forged between the brain and hand, effectively enabling the translation and visualization of even the most subtle design ideas, including textural and tonal variations. The act of drawing is an expression of this link. There are plenty of well-known quotes about the connections between the head, the heart and the hand. John Ruskin, the British artist and writer, said in the late-19th century, "The

education of a young artist should always be a matter of the head and heart and the hand." Art and design, Ruskin said, "must be produced by the subtlest of all machines, which is the human hand."

In 1950, Richard Guyatt, the British designer and academic, described the three interrelated elements of design as the combined action of head, heart and hand. The head provides logic; the heart, emotional stimuli; and the skill that gives form to design concepts is executed by the hand.

As part of our work with students and in our own design practice, we often use two levels of drawing to visualize. These two forms we describe as micro and macro. Micro-visualizing within more complex design challenges explores the alternatives of detail "close in"; for example, how a heading, subheading and paragraph of text come together or how an image and caption unite.

When the most effective combination is decided upon, the relationships and combinations can then be developed and applied across other micro aspects of the design challenge. Taking a number of these effectively resolved micro-visuals, we would progress to our second level of visualization: macro-visualizing. By using the semi-transparent properties of layout paper, we would speedily trace through and bring together a group of micro-visuals to form an entire "page" grouping. It is at this stage that we would draw in the page's parameters and try out different relationships of scale and varied compositional possibilities.
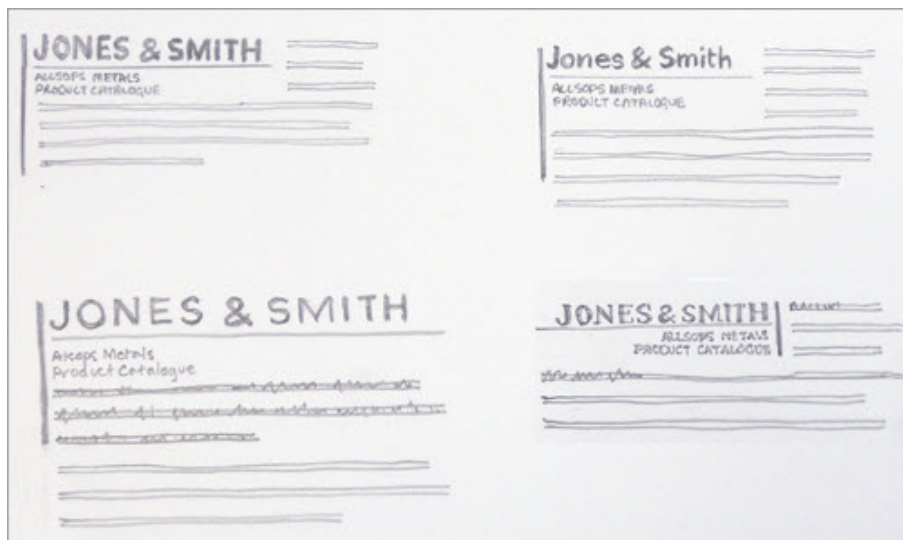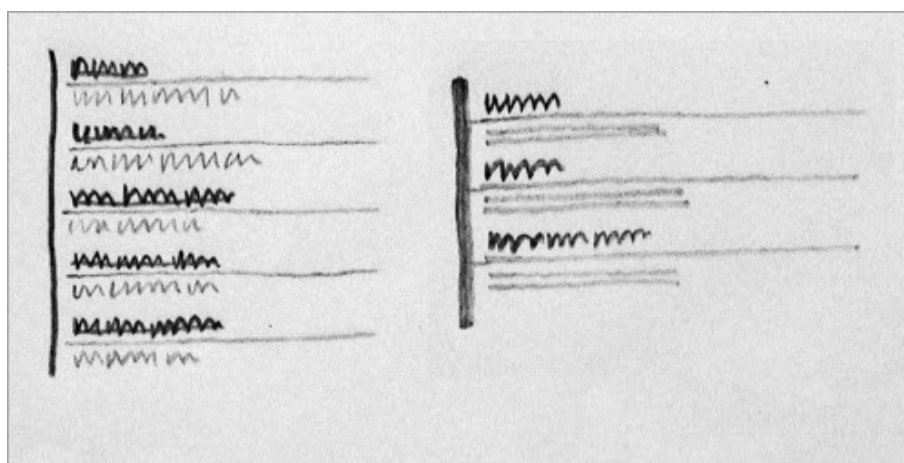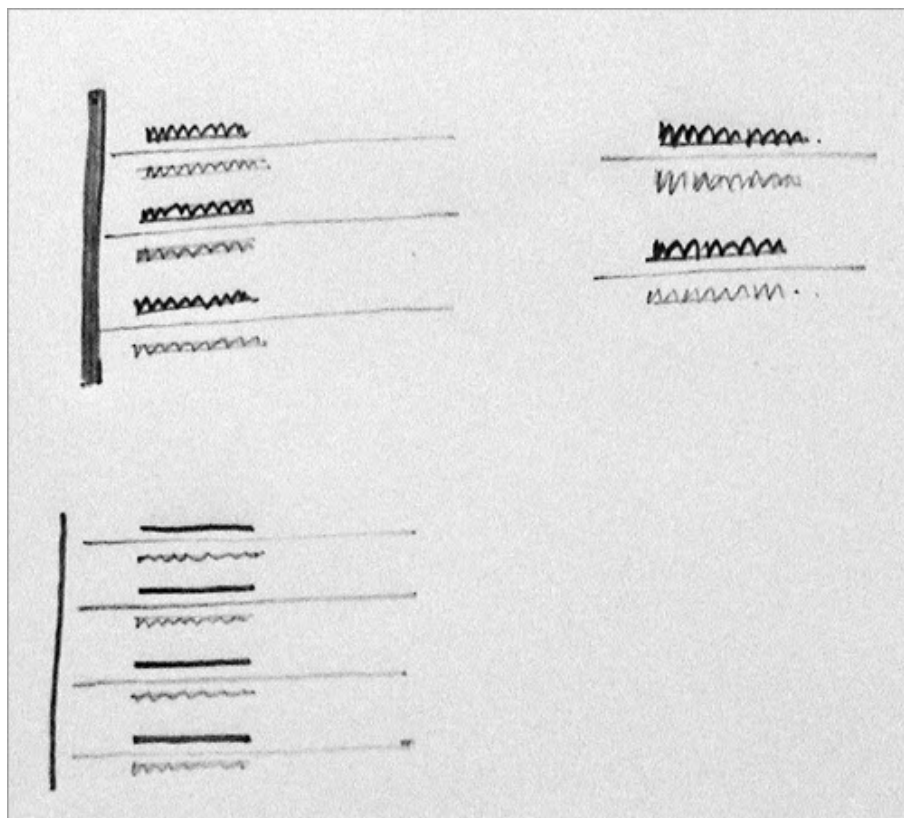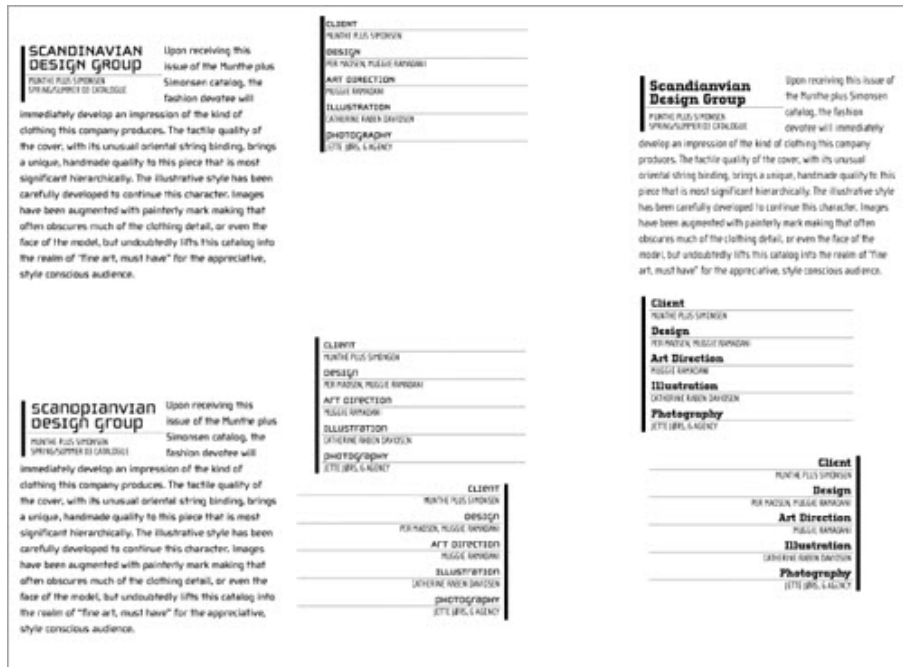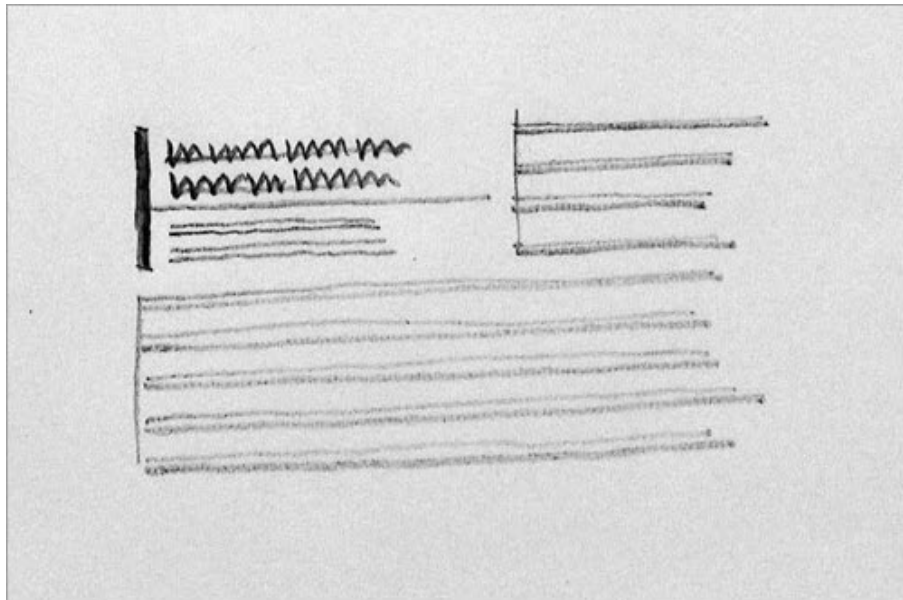
These images show some of the micro-thumbnails for our book design for "Sticky Graph-ics." Given that the book is about memory devices, we wanted to be sure to create a last-ing impression with type. In our design drawings, shown above, we indicate possible textural and tonal variations and hierarchy for the design credits and captioning; also, we have noted the compositional aspects needed for these micro-elements of the design. The final image shows some primary digital interpretations of these drawings.
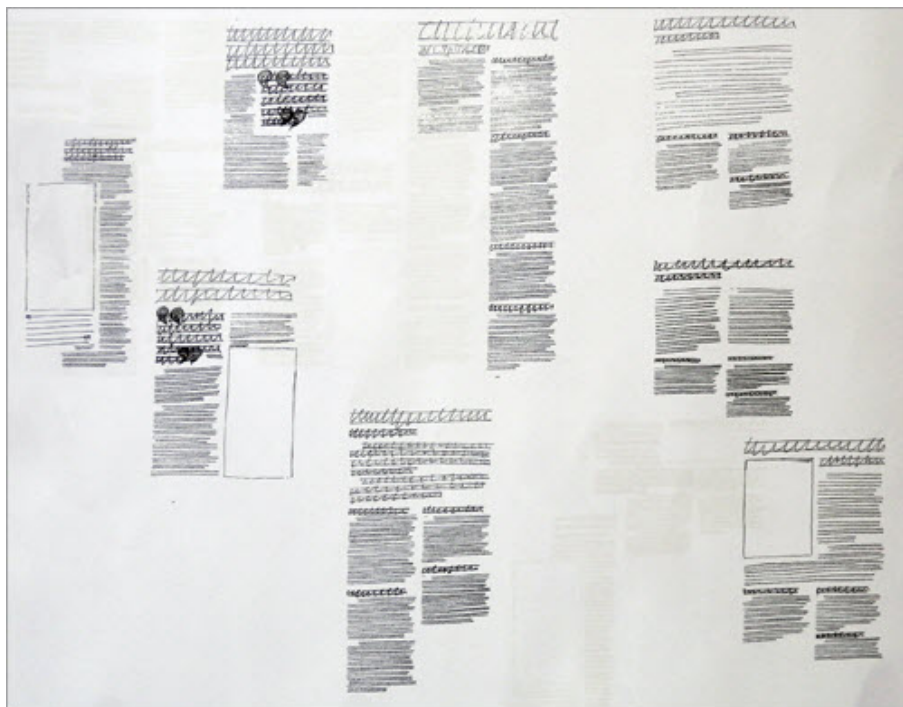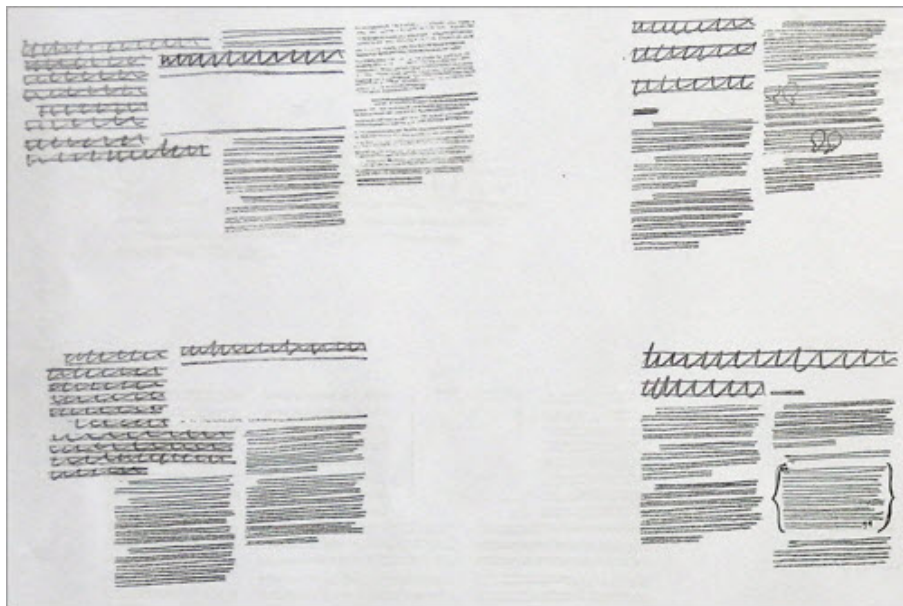
These three images are by graphic designer Abigail Urwin, who has kindly shared some of her macro design drawing and visuals. In these examples, Abigail is developing designs with type for a news spread in a magazine. These pages will ultimately contain five separate smaller articles that come together under a common theme. In the first two images, she is exploring alternative ways to design these articles. In the third image, Abigail has started to show how she might bring these five articles together in the one spread.
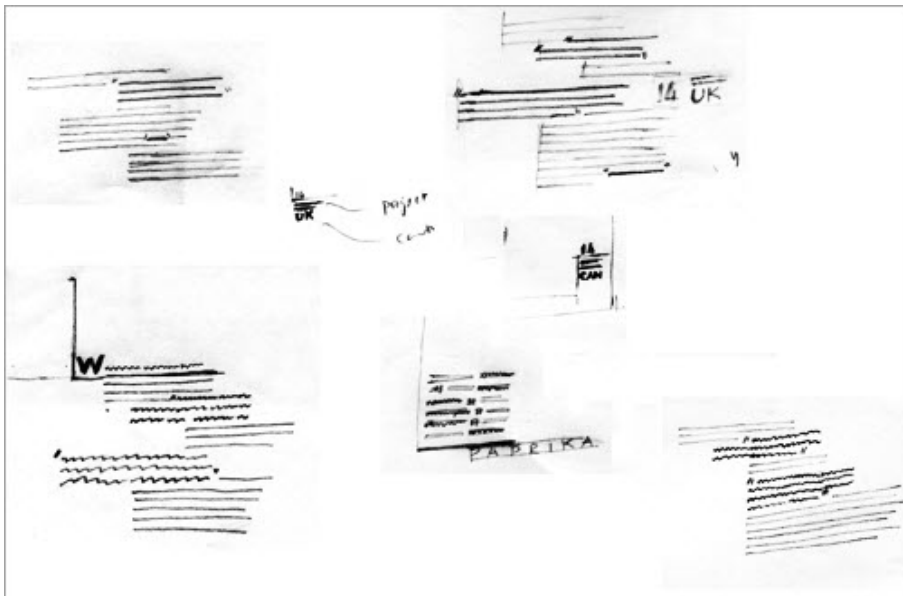
These examples show some of the initial micro-visuals for our first book, "Layout: Making It Fit." The drawings explore textural, tonal, hierarchical and compositional options, and the two double-page spreads show some of our digital interpretations of aspects of these design drawings.

Starting a new design in this way, we would always begin by searching out the most complex aspect of the project. Figuring out a design system to tackle the most difficult scenario first makes it easier to apply the resulting design systems and relationships to simpler aspects of the project. There are no hard and fast rules to sequencing the various aspects of visualization. We have found that the order in which we draw by hand and work on the computer varies, and the two methods are sometimes merged. However done, using these two methods of development in partnership is an excellent way to maximize design ideas.

If You're An Experienced Designer, Is Visualizing Really Necessary?

In a completely new design project, the methodology outlined above helps to ensure that our result is not in any way restricted by the starting point on the computer or by the relatively simple manipulations that are often tempting to pursue but not necessarily the most appropriate or stimulating. Drawing helps to put our most effective design concepts squarely at center stage and forces us to then find the best way to achieve the desired result. If we start with a blank sheet of layout paper and pencil and, of course, undertake research to get inspiration, then design-wise, the sky's the limit.

Were ever inclined to simply reuse a palette of typographic styles that worked in a previous project, simply for ease and speed? We know we were, but taking the time and effort to try out alternatives and to develop completely different options would have improved the result and, for that matter, increased our satisfaction. We had our eyes on the clock and produced solutions that were satisfactory but definitely not the most effective, stimulating or satisfying.

Translating Drawing Into Design

A vital aspect of this process is being able to translate the nuances of mark-making into the final work. This requires making a detailed and precise evaluation of the drawings. The subtle contrasts of tone and texture in your visuals should help to build a valuable picture of the type and imagery and, for that matter, every other aspect of the design. Look carefully at the subtleties in the visuals, and use them as a starting point in selecting the nuances of typeface, weight, tracking, kerning and even leading.

Making the transition from smaller paper visuals to Web or print design can be difficult, and we discuss this issue often with our students. One system that seems to work well for now — albeit, one that relies totally on the visuals being proportional in size to the final design — is to scan or carefully photograph the visuals and then drop them into the background of a digital file to be used as a template for the final artwork. When the design is sufficiently rendered, the template can be deleted.

The three images here show the process described in the preceding paragraph. The first step shows a simple drawing of the front page of a newsletter. The next two images show stages leading to the finished version. The digital rendering, including of the type, is shown in magenta to distinguish between the layers for the purpose of this chapter.

Another Reason To Draw: As Inspiration For Working With Type And Image

Another reason to do mark-making is to develop ideas for type, which we have done very effectively and included as an exercise in our book *Create Impact with Type Image and Color*. The method involves finding samples of type and then cutting and pasting them to form new lay-outs. The copy does not have to make sense, but the abstract results can

be a surprising and inspiring way into a design project that involves type.

This exercise has been very popular with our students, who have been inspired by the design options captured by their abstract layouts. In the examples shown here, students have brought together found samples of type to create textural and tonal contrast and dynamic composition, without the pressure of designing for sense.

The ease and instant gratification of rapidly generating a design right on the computer screen is very tempting. Acceptable concepts can be and are generated this way. But if a designer aspires to a wider spectrum of interesting and usable visual concepts, then achieving this by working solely on the computer is unlikely.

In no way are we belittling the computer; the asset is invaluable to the designer. But we firmly believe that it should not replace the intuitive pathways between the head, heart and hand. Drawing by hand first and then by computer is not set in stone; either may come first if we use them in tandem to explore and develop the most effective solutions for audiences and designers alike.

Paper and pencil concepts must be a part of the design process if ideas are to be maximized. There are no rules or precise styles that designers need to follow, although sufficient accuracy and detail is needed in order to be able to make informed judgments. Our cut-and-paste way of working with found samples of type and putting together unusual combinations can also be an inspiring starting point. While there are no set do's and don'ts in the design process, an appropriate amount of drawing, mark-making and experimentation is bound to improve the final result.

This chapter has been a wonderful opportunity to examine the role that drawing and mark-making can play when generating ideas with type. Previously, we had only briefly touched on this topic when writ-

ing more extensively about the textural and tonal qualities of designing with type and imagery in a number of our books, including *Create Impact with Type Image and Color* and *The Graphic Design Exercise Book*. Both of these practical guides are useful reminders of the fundamentals of graphics and of designing with type. ❧

Useful Resources

- *Graphic: Inside the Sketchbooks of the World's Great Graphic Designers*[104], Steven Heller and Lita Talarico

- *Typography Sketchbooks*[105], Steven Heller and Lita Talarico

- *Create Impact with Type Image and Color*[106], Jessica Glaser and Carolyn Knight

- *The Graphic Design Exercise Book*[107], Jessica Glaser and Carolyn Knight

- Jacek Utko[108]

- "Jacek Utko Designs to Save Newspapers[109]," TED

- Portfolio[110], Vilmas Narecionis

- "Chapter 5: Head, Heart and Hand," *Basics Illustration: Global Contexts*[111], Mark Wigan

---

104. http://www.amazon.com/Graphic-Inside-Sketchbooks-Worlds-Designers/dp/1580932975
105. http://www.amazon.com/Typography-Sketchbooks-Steven-Heller/dp/1616890371
106. http://www.amazon.com/Create-Impact-Type-Image-Color/dp/2940361770
107. http://www.amazon.com/The-Graphic-Design-Exercise-Book/dp/1600614639
108. http://www.utko.com/
109. http://www.ted.com/talks/jacek_utko_asks_can_design_save_the_newspaper.html
110. http://newspagedesigner.org/profile/VilmasNarecionis
111. http://smashed.by/basics-illustration

Applying Macrotypography For A More Readable Web Page

BY NATHAN FORD ❧

Any application of typography can be divided into two arenas: *micro* and *macro*. Understanding the difference between the two is especially useful when crafting a reading experience, because it allows the designer to know when to focus on legibility and when to focus on readability.

This chapter focuses mostly on a few simple macrotypographic techniques — with a dash of micro — and on how to combine them all to build a more harmonious, adaptable and, most importantly, *readable* Web page.

First, some definitions. *Microtypography* has to do with the details; setting the right glyph, getting the appropriate kerning and tracking, and making stylistic choices such as when to use small-caps. Micro techniques have received a lot of attention recently[112], as browser makers adopt new CSS attributes that allow for finer control over Web type[113]. Microtypography deals mainly with legibility and can be thought of as the design of *letters* and *words*.

Macrotypography focuses on how type is arranged on the page. Most macro techniques have been achievable through CSS for quite some time, but because our understanding of the Web page is changing, the way we use these techniques must adapt. Macrotypography deals mainly with readability and can be thought of as the design of *paragraphs* and the *page*.

The Importance Of Readability

For the designer's purpose, readability refers to the ease with which a body of text can be consumed, and it correlates closely to the reader's eye strain[114]. This should not be confused with legibility[115], which refers

---

112. http://coding.smashingmagazine.com/2012/01/30/the-future-of-screen-typography-is-in-your-hands/
113. http://ie.microsoft.com/testdrive/Graphics/opentype/Default.html
114. http://www.eyemagazine.com/opinion.php?id=157
115. http://www.sitepoint.com/typography-readability-and-legibility-part-1/

to the degree to which individual glyphs in text can be discerned. The techniques for creating a great reading experience are complementary to those for creating a great user experience (UX), and vice versa. They also both share the same symptoms of failure. Poor readability on a website can lead to confusion, frustration and ultimately abandonment, while a great reading experience is invisible, supportive and engaging.

As with UX design, every website design would benefit from some measure of concern for readability. For example, text-heavy websites — such as blogs, newspapers and magazines — should uphold readability as a priority, while websites for events and e-commerce might just need to tweak line heights and font sizes. Whatever level of importance you place on readability, you should undertake a continual process of refinement towards an effortless reading experience.

Techniques For Improving Readability

The foundation of great reading experiences on the Web lies in the study of book design. After all, books are where readable typography was honed. Personally, I hold *The Form of the Book* by Jan Tschichold as the ultimate resource for good taste in book design, and I am certainly[116] not alone[117].

Many of the techniques we'll cover here have been adapted for the Web page from lessons introduced in this book. Sadly, the book has been out of print for about 20 years (at least in the US), and a copy can cost around $150 on Amazon's marketplace[118]. I have created a digest of it[119], but if you want to read the full text, you could always try your local library or university (which is how I finally got my hands on it).

Now, let's look at the various macro techniques — and a few micro techniques — to make your website's content more readable. I have chosen an article that is typical of the kinds of reading experiences users encounter. I have removed the header and some branding elements, but it remains mostly as I found it.

In our example, important content (navigation, advertising, related links) lies on either side of the reading area. For optimum readability, a less obtrusive placement of these elements would be best, but this is not always possible. We will, therefore, not rearrange the layout, but work within it. Here is what we are starting with:

116. http://jasonsantamaria.com/reading/the-form-of-the-book
117. http://aworkinglibrary.com/library/archives/the_form_of_the_book/
118. http://www.amazon.com/Form-Book-Morality-Classic-Typography/dp/0881791164/ref=sr_1_1?ie=UTF8&qid=1330421526&sr=8-1
119. http://artequalswork.com/posts/form-of-the-book.php

As we learn about each technique, we will apply it to our example. But keep in mind that this exercise is to improve only the reading experience, not the overall design.

Command Your Margins

Margins give the eye room to maneuver. They provide a buffer between the main content and ancillary elements — such as related links and ads — allowing the reader to focus on the text. Beyond this purely functional purpose, margins can also bring deeper harmony to the layout.

Margins comprise negative space; they afford the designer an opportunity to build a relationship between a body of text (the figure) and its surroundings (the ground). As Tschichold tells us, "Harmony between page size and the type area is achieved when both have the same proportions." Now, the proportions of a page in a book are much different than those of most digital displays (especially ones in landscape orientation), so to adapt this concept to the Web, we can work towards a harmony between our text and its immediate visual container.

IN PRACTICE

On our example page, the margins are not very generous. Also, the main content is crammed in between two very loud columns. First, we can add more space to the right of the text, giving the reader room to go from the end of one line to the beginning of the next without being dis-

tracted by the secondary content on the right. And adding more margin to the left of the text allows the reader to easily find the start of the next line and to scan the article for topics they are interested in.

Margins can be set intuitively by increasing the amount on each side until the content feels comfortable. Applying this to our `article` element is easily achieved by adding `padding` in our CSS (rather than `margin`, in this case). For now, we will just double the `padding` on the left and right:


```
article {
    padding-left: 20px;
    padding-right: 40px;
}
```

In our adjustment of the margins, we can create still greater harmony between the copy and its surroundings, but first we must visualize an invisible container around the content. This will be our "page" with which to build harmony in the reading area:

The way to create harmonious proportion between text and its container is to give them the same shape. The content should have the same proportions — only smaller — as its containing element. Tschichold surveyed a mountain of book proportions and concluded that the most harmonious proportions for margins are roughly 2:3:4:6 (top:right:bottom:left) for the left-facing page (recto) of a book. Given that we do not have facing pages on the Web, we can make the margins symmetrical and adjust the ratio to 2:3:4:3 by shaving off a bit of the left margin. Web text does not need as much side margins as book text because Web pages do not need to accommodate the reader's thumbs.

Though they may seem the obvious unit of measure, percentages for `padding` will only measure in relation to the *width* of our `article` element's container, skewing our top, bottom and side proportions to an inappropriate degree. Therefore it's best work with `padding` in ems or pixels until the reading area has the same proportions as its container. To keep things simple, let's start with 2em for the top padding in our example. After applying our adjusted ratio from above, our `article`'s `padding` can be written as `2em 3em 4em 3em` or `2em 3em 4em` in CSS shorthand. Given the fluid nature of content on the Web, this is just an ap-

proximation of Tschichold's proportions. For a typical body of text on the web — which is taller than it is wide — the margin should be generally less on top, even on the sides, and most at the bottom:
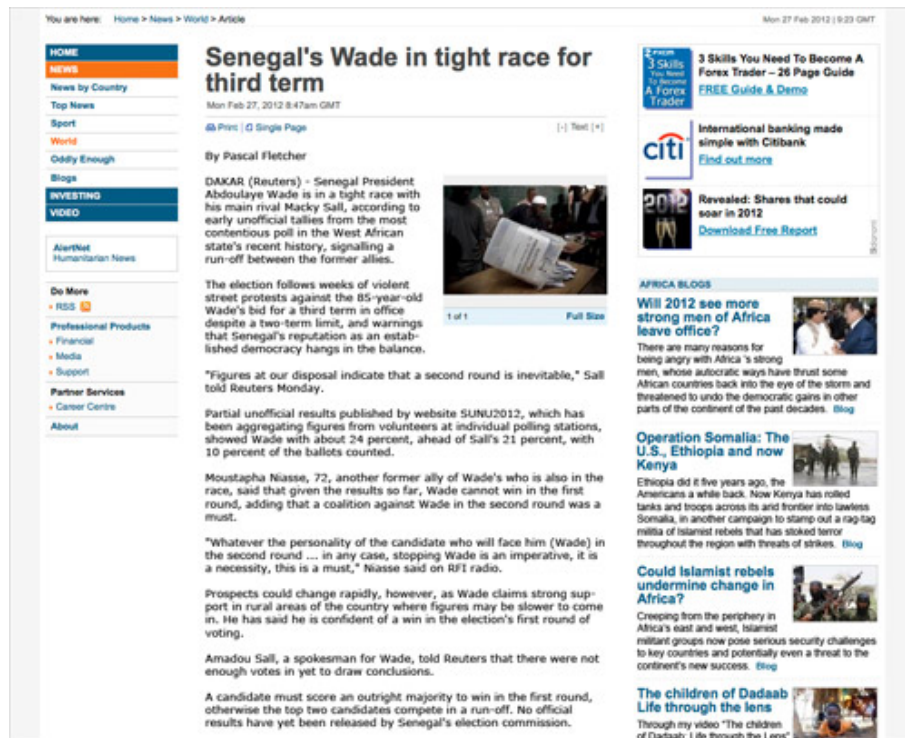

```
article {
    padding: 2em 3em 4em;
}
```

We can also move the lead image to the right. This allows the body copy to hold its shape better and allows for even easier scanning of the article. We can break this principle to draw attention to images and figures, of course, but for our example the image is too distracting on the left when placed early in the article.

If we want, we can bring the text forward on the z axis, putting even more focus on our copy and de-emphasizing the ancillary content by creating a visible container for our text. This is a tactic we can easily use in Web design that books do not need:

```css
body {
    background: #fcfcfc;
}


article {
    background: #fff;
    border: 1px solid #eee;
    padding: 2em 3em 4em;
}
```

Our page already feels more balanced and less intimidating, but we can use more techniques in the body of the text to further enhance readability.

Choose Readable Fonts

Font selection is a micro concern, but it has a tremendous impact on the macro appearance. In *Detail in Typography*[120], Jost Hochuli best outlines this interdependence: "In typography, details can never be considered in isolation."

The font for the body copy should be chosen for its on-screen readability, before any concern for style. The headings and pull quotes are perfect opportunities to flex your typographic creativity, but leave the

---

120. http://www.amazon.com/Detail-Typography-Jost-Hochuli/dp/0907259340

long runs of copy to dependably readable workhorses such as Georgia, Arial and Myriad, which were all designed for optimal reading on a back-lit screen.

Fonts that are more readable on digital screens typically exhibit the following attributes:

- Tall x-height;

- Slightly wider em width, but not condensed or extended;

- Mostly devoid of style;

- Serifs for larger type, and sans-serif for smaller type.

All of these rules have exceptions, but they should be your guiding principles when choosing a font for the body copy. Here are some font stacks that I find give some flavor of style, provide appropriate fallbacks and are all highly readable:

- `"Proxima Nova Regular`[121]`", "Helvetica Neue", Arial, Helvetica, sans-serif;`
  (As used on Smashing Magazine)

- `"Myriad Pro`[122]`", Arial, Helvetica, sans-serif;`
  (As used on my website[123])

- `"Fanwood Text`[124]`"", Georgia, Times, "Times New Roman", serif;`

- `"PT Sans`[125]`", "Trebuchet MS", Arial, sans-serif;`

IN PRACTICE

Let's apply Myriad Pro to the body text on our page:

121. https://typekit.com/fonts/proxima-nova
122. https://typekit.com/fonts/myriad-pro
123. http://artequalswork.com
124. http://www.google.com/webfonts/specimen/Fanwood+Text
125. http://www.google.com/webfonts/specimen/PT+Sans
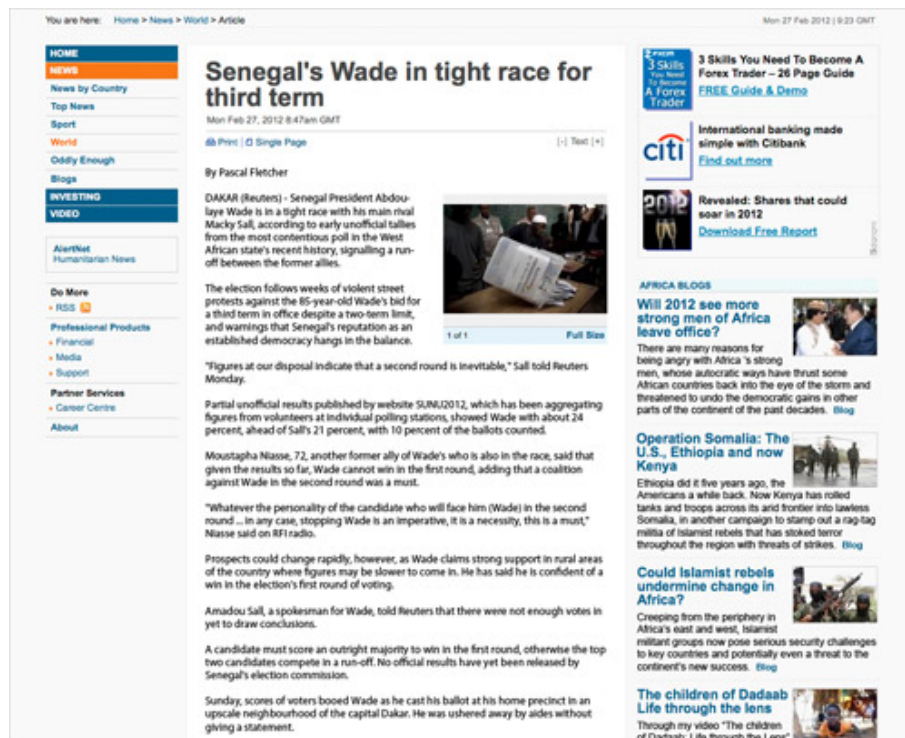
```css
article {
    background: #fff;
    border: 1px solid #eee;
    font-family: "Myriad Pro", Arial, Helvetica, sans-serif;
    padding: 2em 3em 4em;
}
```

Keep It Measured

In setting any block of text, we must consider its measure. Measure is defined by either the number of characters per line or the number of words. I use words because they are easier to count, and I try to follow Tshichold's suggestion of 8 to 12 words per line. If you base your measure on characters, then 45 to 85 characters per line is ideal. Once the margins and widths have been set, proper measure can be achieved through two attributes in the CSS, `font-size` and `word-spacing`.

When deciding on a size, strike a balance between making the font small enough that the reader is not too distracted when jumping to the next line, but big enough that they do not have to lean in to read the text on the screen. For most fonts, 16 pixels works well. Other factors might lead you to making it larger or smaller, but 16 pixels is a great place to begin. As for word spacing, most browsers do a decent job of setting this for you, but if you are having trouble getting an appropriate measure, cheating this attribute slightly either way can be handy.

On our page, let's add a 16-pixel font size, and cheat the word spacing in just a tiny bit to achieve a proper measure (`word-spacing` is supported in all major browsers[126]). You might instead want to use ems or rems[127] here so that the layout remains flexible whatever the font size set by the user as their default.

Until we set a new line height, our page will look like a jumbled mess, so let's just look at the code at this point:

```css
article {
    background: #fff;
    border: 1px solid #eee;
    font-family: "Myriad Pro", Arial, Helvetica, sans-serif;
    font-size: 16px;
    padding: 2em 3em 4em;
    word-spacing: -0.05em;
}
```

Set An Appropriate Line Height

Once the font size is set, you can determine the appropriate line height. On the Web, we work in terms of line height, which by default is an equal amount of space above and below text on a line. Not to be confused with leading in print design, which generally refers to the amount of space below a line of text. The governing rule for line height (and leading) is, the longer the line length, the taller the line height should be. And vice versa: the shorter the line length, the shorter the line height.

Find an appropriate line height by first determining the point at which the ascenders and descenders of the lines of text do not touch, yet the lines are close enough that the reader requires no effort to find the next line. Then adjust until the height feels balanced with the line length. Some may leave the `line-height` attribute to the browser's default, while some may set a global `line-height` on the `body` element. Both approaches make sense because the line height would then stay proportional to the element's font size; but both also assume that the line width of the content will stay consistent, which could lead to situations that violate our governing rule.

---

126. http://reference.sitepoint.com/css/word-spacing
127. http://snook.ca/archives/html_and_css/font-size-with-rem

IN PRACTICE

Let's add a line height of 1.3 ems to our example, using ems so that our line height stays proportional to the font size, and see how the font size and line height work together:


```css
article {
    background: #fff;
    border: 1px solid #eee;
    font-family: "Myriad Pro", Arial, Helvetica, sans-serif;
    font-size: 16px;
    line-height: 1.3em;
    padding: 2em 3em 4em;
    word-spacing: -0.05em;
}
```

It is important to note that readable line heights can be especially tricky to keep consistent in responsive layouts, as line lengths will vary based on device widths. To solve this issue, Tim Brown has proposed an idea he calls "Molten Leading[128]," which would allow line heights to increase proportionally based on the screen width. His post links through to a couple of Javascript implementations of this idea. In lieu of Javascript intervention, you can also manually find the screen widths at which the

---

128. http://nicewebtype.com/notes/2012/02/03/molten-leading-or-fluid-line-height/

line heights become uncomfortable, use a media query to target that width, and set a more readable `line-height` in the CSS.

Find The Proper Paragraph Styles

We need to figure out which paragraph style best fits the content. Jon Tan has done a fantastic job of outlining various styles[129] and how to craft them with CSS. The appropriate style for a piece of content varies based on the flavor of the content and the rhythm of the paragraphs. I have written about my preference for using indents[130], rather than line breaks, when setting long-form text. This helps to keep the flow between ideas, but it can be distracting when the paragraphs are short or the line length is long. Deciding what constitutes the perfect paragraph[131] for your content is up to you.

IN PRACTICE

Our page is a news article, where the flow between paragraphs is dictated more by chronology than by ideas, so line breaks are still appropriate. We could easily apply indents, if appropriate, to the paragraphs with one simple CSS rule:

```css
article p + p {
    text-indent: 2em;
}
```

We specify `p + p` rather than just applying the rule to all `p` tags because we want to indent only those paragraphs that follow other paragraphs. Ones that follow headings, images and so on should not be indented.

Instead of indenting, though, we just want to shrink the line breaks a bit so that each paragraph is not so disconnected from the last. For our page, let's use half of the line height:

129. http://jontangerine.com/silo/typography/p/
130. http://artequalswork.com/posts/islands-of-thought.php
131. http://uxdesign.smashingmagazine.com/2011/11/29/the-perfect-paragraph/
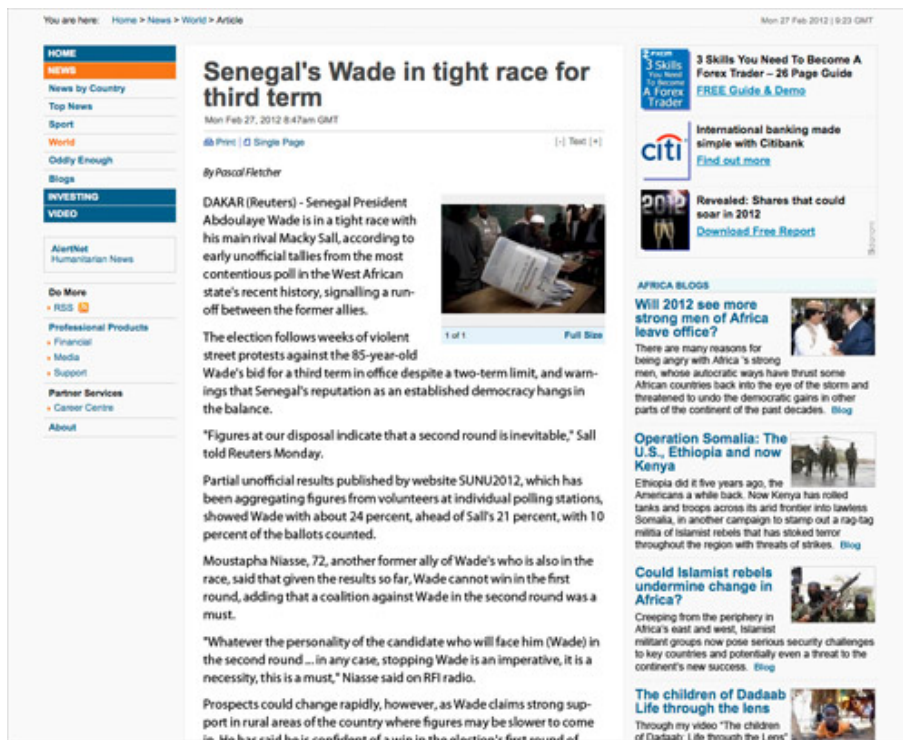
```
article p {
    margin-bottom: 0.5em;
}
```

Balance The Text's Contrast

One final consideration for content is text color. Contrast is a major contributing factor in eye strain[132] and so greatly impacts readability. Low contrast between text and background causes more squinting and blinking among readers[133], a sure sign of strain. Black on yellow has the highest contrast, but we have been conditioned to view this as a sign of warning or alarm, thus increasing anxiety among readers. Black on white is high in contrast, too, but too harsh for extended reading on back-lit screens. For long-form text, I have found dark-gray text (around `#333`) on a white or light-gray background (no darker than `#EEE`) to be optimal. This is a gross simplification of color theory to suit the purposes of this chapter. To learn more about color, Mark Boulton[134] has written a great primer on color theory for the Web[135]; you can also find many great examples in Smashing Magazine's series on color[136].

132. http://www.writer2001.com/colwebcontrast.htm
133. http://www.eyemagazine.com/opinion.php?id=157
134. http://www.markboulton.co.uk/
135. http://designingfortheweb.co.uk/book/part4/part4_chapter18.php
136. http://www.smashingmagazine.com/2010/01/28/color-theory-for-designers-part-1-the-meaning-of-color/

Our article already has a white background (serving as a boundary for the margins), set against a wider light-gray background. We should probably keep the white, and lessen the darkness of the text to #444. We can then use #000 on the headings to give them slightly more emphasis:

```css
article p {
    margin-bottom: 0.5em;
    color: #444;
}


article h1 {
    color: #000;
}
```

The Result
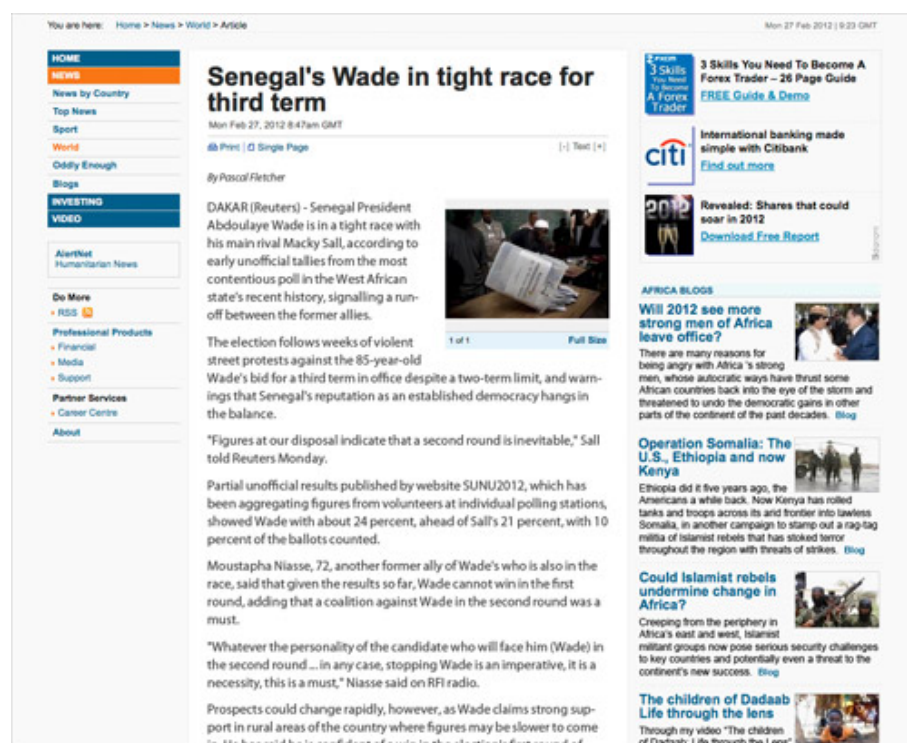
We now have a much more readable page that invites users into the content. We could employ many more techniques across the entire website, but we have focused here on the main content block. Harry Roberts[137] has written a great overview of these techniques[138] and more

---

137. http://coding.smashingmagazine.com/author/harry-roberts/

for Smashing Magazine, which will give you a deeper understanding of everything covered here.

With a clean reading experience, people will better absorb the ideas being presented and will undoubtedly come back for more — that is, if your content is worth reading… but I can't help you there.

Excellent Reading Experiences On The Web

Readability is not a new concept, of course. If you are just discovering what makes for a good reading experience, then congratulations, and welcome to all the discomforts of recognizing cramped and neglected type on the Web. It's not all pain, though. Plenty of well-considered blocks of content are to be found. Let's look at a few great ones and a couple that could be great with slight tweaks.

Please note: In the interest of showcasing only the reading experience, we have cropped each page to a scrolled view of the main content.

24 Ways[139]

The reading experience on 24 Ways is quite nice. The text contrast is well balanced, the measure is not too long, and the font size is generous. At all responsive breakpoints, the design is a perfect example of a page with sufficient and balanced margins around the main reading area.

---

138. http://coding.smashingmagazine.com/2011/03/14/technical-web-typography-guidelines-and-techniques/
139. http://24ways.org/2011/crafting-the-front-end
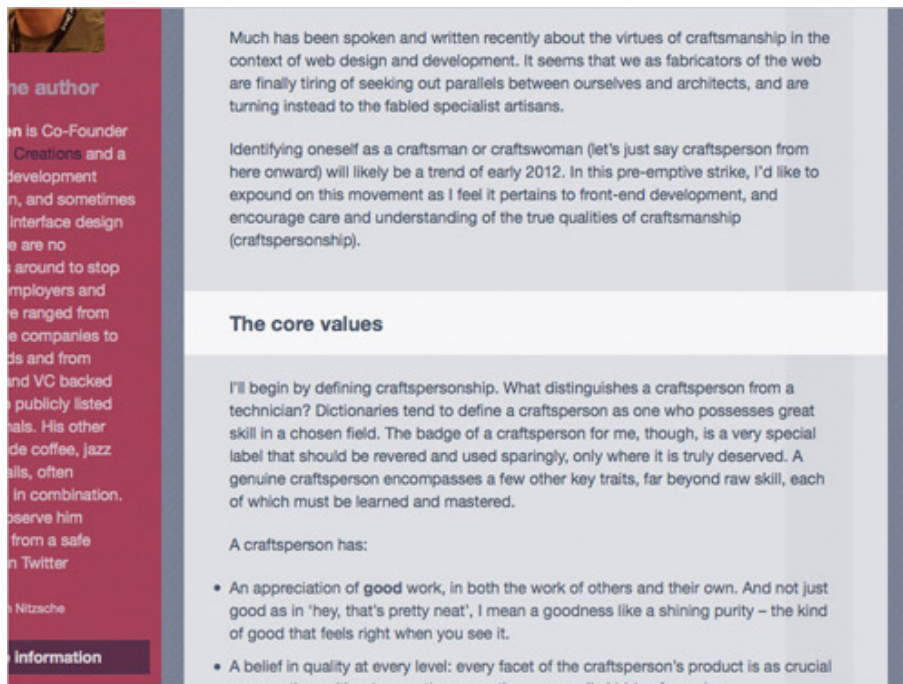
CNN[140]

Long-form articles on CNN are good examples of how readability can work well on news websites. The layout does not show a visible container for the article — which in this case might have been distracting on a page already laden with so much content — but the margins are generous. Also, the line breaks for the paragraph styles are completely appropriate, because most online news stories are collated and updated from many sources and are not linear ideas. The font size (currently 14 pixels for the body copy) could stand to be a bit bigger, though.
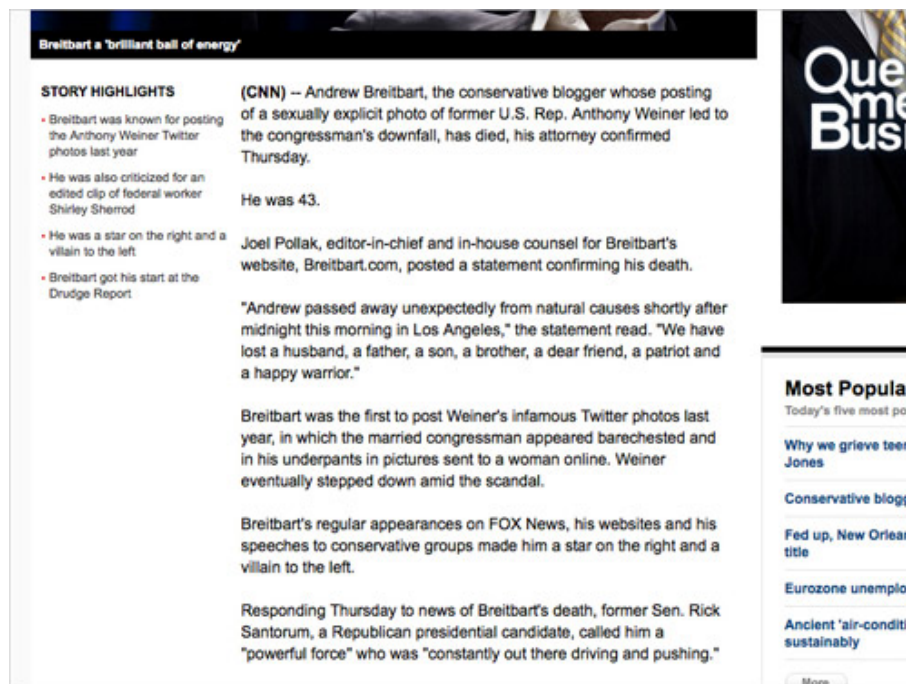
Contents[141]

The tablet view of Contents magazine is a wonderful experience all around. The measure is perfect, the line height and font size play together nicely, and the paragraph styles are perfectly suited to the content. The measure does get too long at desktop sizes, but with all of the other factors working so well, the effect on overall readability is negligible.

---

140. http://edition.cnn.com/
141. http://contentsmagazine.com/articles/space-to-breathe/
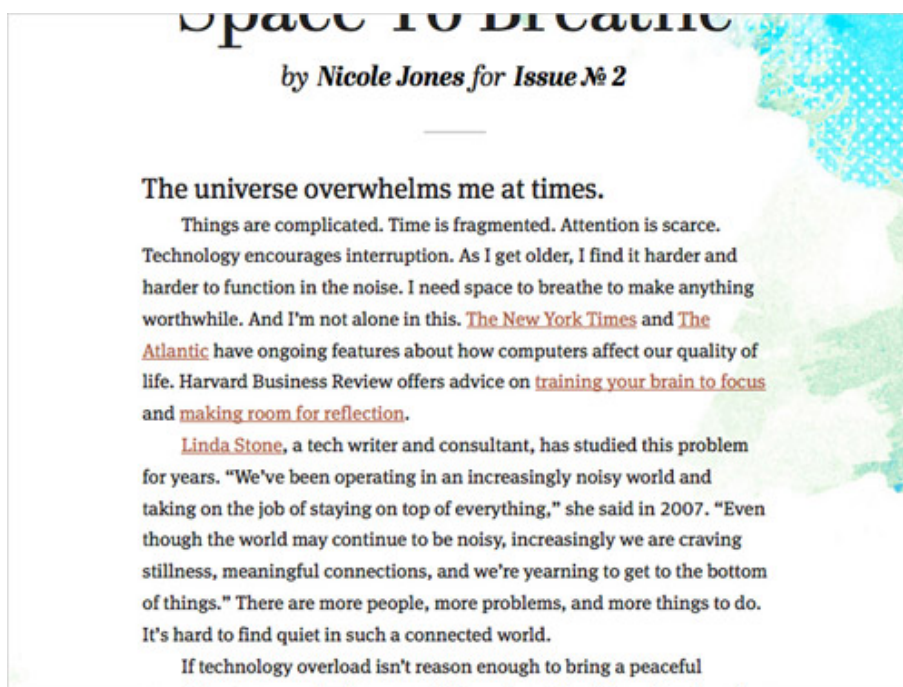
Tablet view
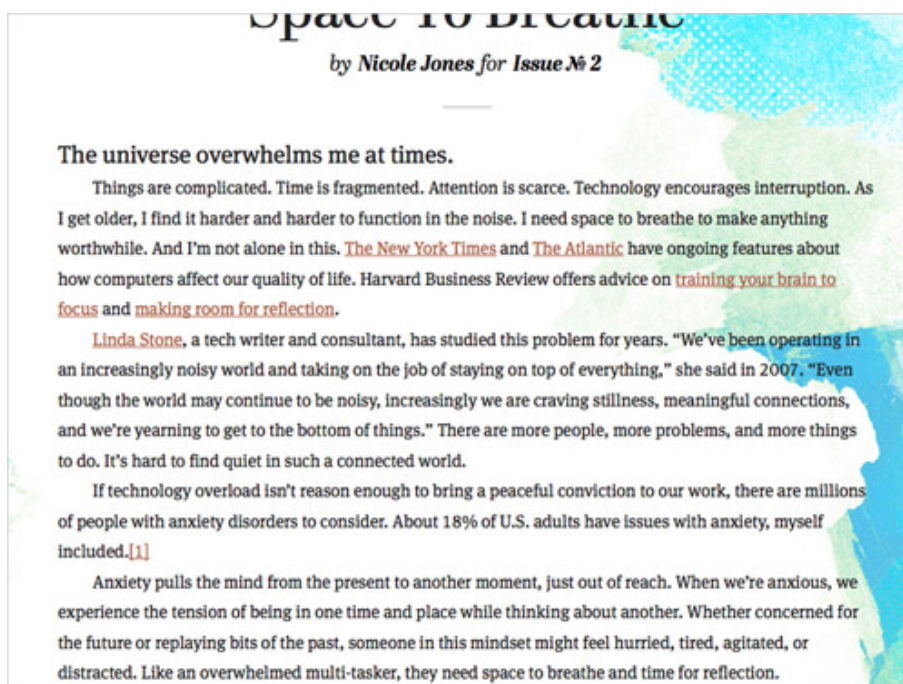
Desktop view

Elliot Jay Stocks[142]

Elliot does quite a few things well on his website. The measure is right, the font (Skolar[143]) is very readable and set at a comfortable size (16 pix-

---

142. http://elliotjaystocks.com/blog/dog-days/
143. https://typekit.com/fonts/skolar-web

els), and the line height is just tall enough to accommodate the link style. Generous margins create harmony between the main content and its container, while the side margins are uneven, making the page look like the recto of a book and giving the layout a unique character.
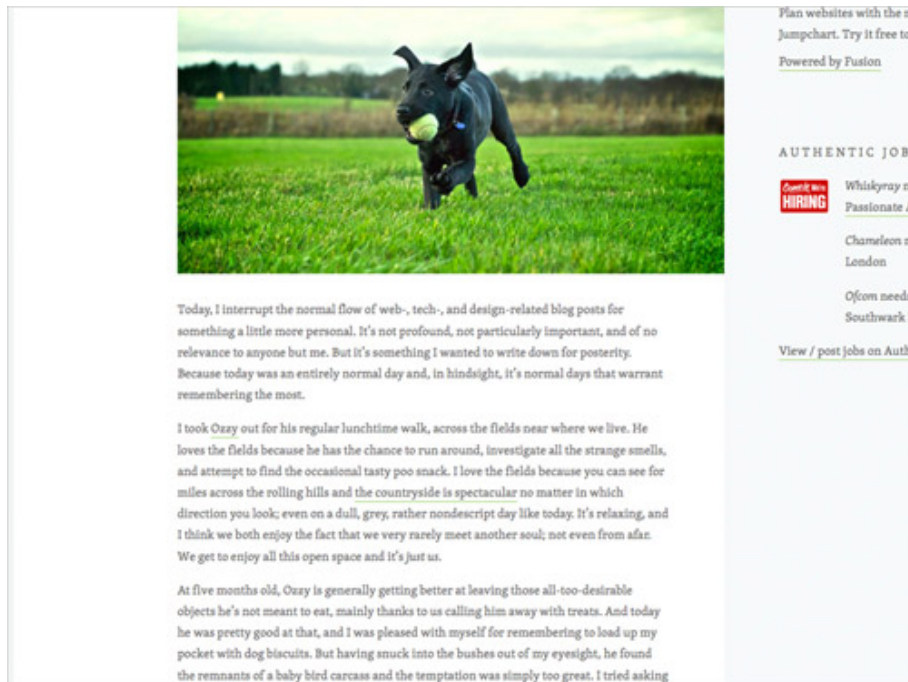
Esquire[144]

Most articles on Esquire are great, but the reading experience is merely good. The margins are ample, the font is readable, and the contrast is high. All of these go a long way towards establishing good readability, but a few simple tweaks would make it great. Increasing the right padding would shorten the measure, which is a bit too long as it is. The font size could also be increased by a couple of pixels. And given that most Esquire articles are a linear progression of ideas, I would suggest paragraph indents rather than line breaks.

---

144. http://www.esquire.com/blogs/mens-fashion/colorful-clothes-for-men-2012?click=mid

The Guardian[145]

The design team over at The Guardian pays attention to crafting great all-around experiences. Readability is no exception. Measure, contrast and paragraph styles all work together to create a focused and comfortable reading experience in the midst of what could be an overwhelming amount of content.
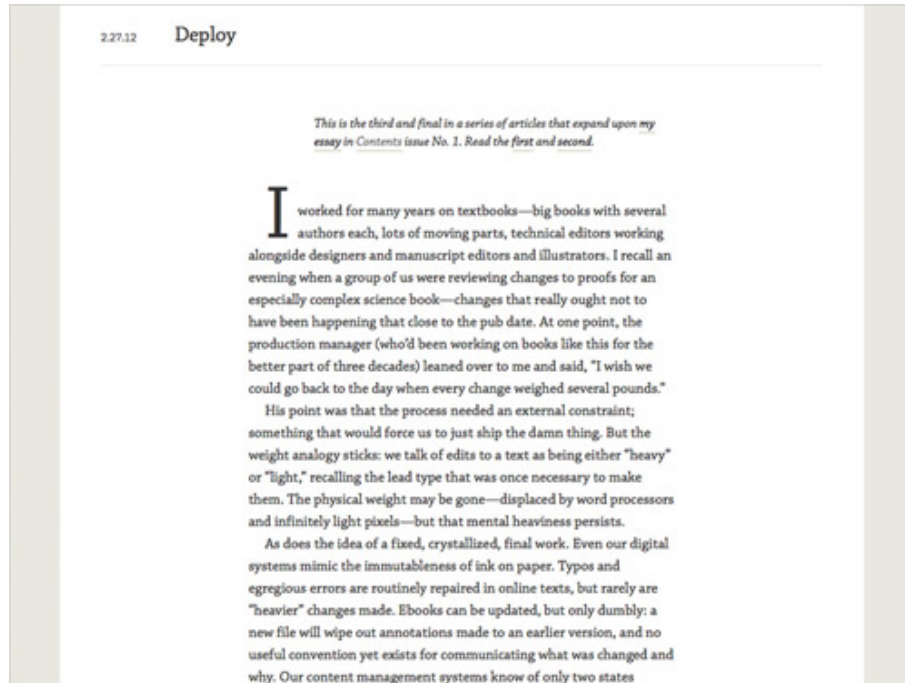

145. http://www.guardian.co.uk/business/2012/mar/04/pension-funds-infrastructure-os-borne

A Working Library[146]

A Working Library is one of the best reading experiences on the Web. Every aspect of readability discussed in this chapter has been well considered and executed. The harmony between text and its container is pitch perfect.

Refining Towards The Ideal

With the examples above, we have tried to show how readability can excel in a few different digital environments: blogs, news websites and online magazines. Some of these website do not have many of the constraints (such as ads and related content) of more commercial websites, so it could be argued that these designs exist in a vacuum, without pragmatic application or real-world pressures. We need these shining examples, though, to help us find the ideal reading experience for each project; and once we know that ideal, we should do our best to reach it.

In a recent talk on "What Is Reading For?[147]" the famous typographer and poet Robert Bringhurst[148] stated, "Books are and have to be utilitarian objects. They have to be *used*." The same could be said of Web pages. Ideal reading experiences create better user experiences. Our job as designers is to refine the aesthetic qualities of the Web's content in order to speed the process of consumption, thereby facilitating deeper understanding. Tired eyes all over the Web are counting on us. ❧

---

146. http://aworkinglibrary.com/library/archives/deploy/
147. http://futureofreading.cias.rit.edu/2010/video.php#RobertBringhurst
148. http://en.wikipedia.org/wiki/Robert_Bringhurst

Avoiding Faux Weights And Styles With Google Web Fonts

BY LAURA FRANZ ❧

If you're using Google Web Fonts on your websites, then there's a very good chance that 1 in 5 visitors are seeing faux bold and italic versions of your fonts — even if you correctly selected and used all of the weights and styles. That's because the implementation method recommended by Google Web Fonts doesn't work with Internet Explorer 7 or 8.

(As of 21 May 2012, StatCounter[149] reports that Internet Explorer 7 or 8 was used for 19.4% of the 45 billion page views collected in February, March and April 2012.)

As an experienced print and Web typographer, I embrace and use the term "font" when talking about Web fonts; it's the term used in CSS syntax and by a myriad of Web font providers.

Faux Bold And Italic Fonts Are Stretched And Slanted

Any designer who loves type will tell you that faux bold and italic aren't beautiful. When a browser can't find the true bold or italic version of a font, it will often "fake it" — creating faux bold and italic by stretching and slanting the original font.

FAUX BOLD

Faux bold is made by slightly stretching the vertical strokes of the original font. In the image below, I've used Droid Sans Bold, which has consistent strokes. Yet in the faux bold, the vertical strokes are a little thicker than the horizontal strokes. This is most noticeable in the letter "e"; the top of the letter, where the stroke is thinnest, looks pointy.

---

149. http://gs.statcounter.com

A faux bold (top) slightly stretches the vertical strokes of the original font. This creates odd shapes, like the pointy top of the letter "e." A true bold (bottom) is more consistent between horizontal and vertical strokes.

FAUX ITALIC

Faux italic is made by slanting the original font at an angle. In the image below, I've used Droid Serif italic. The faux italic is missing the tail on the lowercase "f," while the lowercase "a" continues to have the double-story shape. In a true italic font, the "f" and "a" look more calligraphic — or handwritten — especially in serif fonts. If you've chosen a serif font for an older, more traditional feel, then you'll probably want to preserve these true characteristics of italic.

Faux italic (top) is made by slanting the original font. True italic (bottom) often has traditional calligraphic characteristics, such as the extended stroke on the "f," the single-story "a" and the rounded "e."

FAUX BOLD ITALIC

Faux bold italic both stretches the vertical strokes and slants the letters at an angle. The resulting letters are clunky compared to the rhythm and texture of a true bold italic.

Faux bold italic (top) is both stretched and slanted. True bold italic (bottom) has a thoughtful rhythm and texture.

Faux bold and italic are not as beautiful as the real thing. But when it comes to text, even more important than beauty is readability.

Faux Bold And Italic Undermine Reading

Faux bold and italic are often less legible, which in turn undermines the readability of text. When letters are stretched and slanted, the strokes and spaces are no longer well balanced.

WELL-BALANCED STROKES AND SPACES IMPROVE READABILITY

If you've ever had to read a bad photocopy (or scan) for a class, conference or meeting, then you already appreciate how important strokes and spaces are for reading complex text.
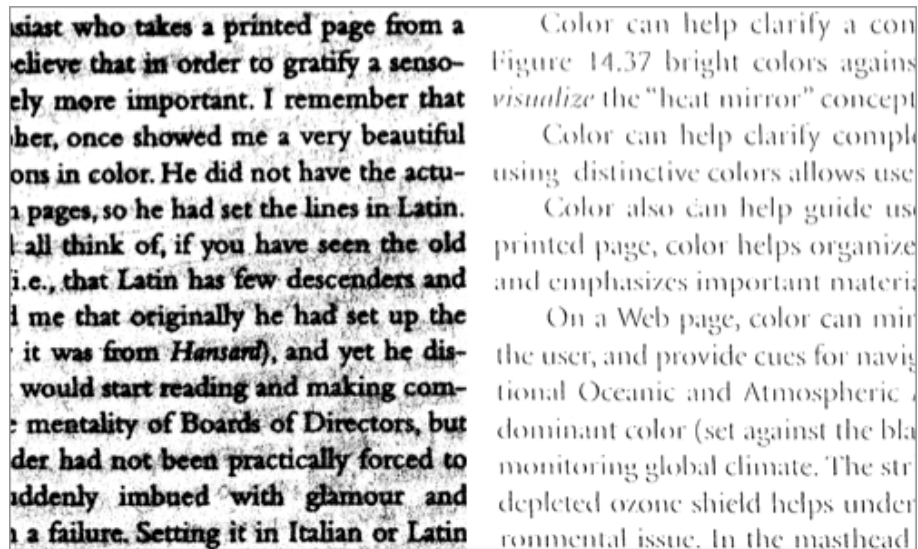
Reproductions that are too dark or too light are hard to read. Too dark and we lose the spaces in the letters (left); too light and we lose the strokes in the letters (right).

If a photocopy or scan is too dark, we lose the spaces in and around the letterforms. If a photocopy or scan is too light, we lose some of the strokes in the letterforms. In both cases, text is less legible. Reading speed slows as we try to identify the letters and words. We experience brain fatigue as we find ourselves rereading phrases to make out the words, leaving less energy for comprehension.

A good balance between strokes and spaces improves legibility and helps us read more quickly and easily.

FAUX BOLD, STROKES AND SPACES

Because faux bold is created by stretching the vertical strokes of letters, the top and bottom strokes on rounded forms are often too thin. This makes letters like "e," "c" and "s" start to break on the top and bottom curves. Meanwhile, letters with a diagonal stroke, such as "w" and "N," get too heavy and start to pop out of the rhythm of the text.

For decades, critics have predicted the end of the written word: "No one reads anymore! No one writes anymore!" Yet from birth certificates to gravestones, from T-shirts to text messaging, the written word—and thus reading—is woven into the fabric of our everyday lives.

For decades, critics have predicted the end of the written word: "No one reads anymore! No one writes anymore!" Yet from birth certificates to gravestones, from T-shirts to text messaging, the written word—and thus reading—is woven into the fabric of our everyday lives.

Droid Sans bold. The faux bold (top) is slightly less legible. The tops and bottoms of the rounded letters — like "e," "c" and "s" — tend to disappear. Diagonal letters, like "w" and "N," are too bold. True bold text (bottom) is more consistent.

FAUX ITALIC, STROKES AND SPACES

Because faux italic is created by slanting the original font at an angle, the spaces in the letters get condensed. This is a particular problem in the lowercase "a," which continues to have two counterforms. Ironically, while faux italic letters feel more squished and are more difficult to read, they often take up more room, and fewer characters fit on a line.

> *For decades, critics have predicted the end of the written word: "No one reads anymore! No one writes anymore!" Yet from birth certificates to gravestones, from T-shirts to text messaging, the written word—and thus reading—is woven into the fabric of our everyday lives.*
>
> ---
>
> *For decades, critics have predicted the end of the written word: "No one reads anymore! No one writes anymore!" Yet from birth certificates to gravestones, from T-shirts to text messaging, the written word—and thus reading—is woven into the fabric of our everyday lives.*

Droid Serif italic. The faux italic (top) is less legible. Spaces within the letters are more condensed. The text itself feels smaller, even though fewer characters fit on each line. True italic (bottom) is not just more visually pleasing, but more legible and, thus, easier to read.

Fixing Google Web Fonts Bold And Italic In IE 7 And 8

Because real bold and italic fonts are usually more beautiful and more readable than their faux counterparts, we should make them work on as many browsers as possible.

As is usually the case, figuring out how to fix the problem starts by understanding why the proper bold and italic fonts don't work in the first place.

In "Say No to Faux Bold[150]," recently published on A List Apart, Alan Stearns reminds us that a good start is to use fonts for which bold and italic styles are available — and to actually include the bold and italic styles that you need when choosing the fonts in the Google Web Fonts interface.

But choosing and using bold and italic styles aren't enough.

---

150. http://www.alistapart.com/articles/say-no-to-faux-bold/

MULTIPLE WEIGHTS AND STYLES DON'T WORK IN IE 7 AND 8

Google Web Fonts instructs us to implement its fonts by checking off all of the font weights and styles that we want to use, then copying the link it provides, and pasting it into the head of our HTML document.

For example, to use Droid Serif regular, italic, bold and bold italic, we would select all four weights and styles, like so:
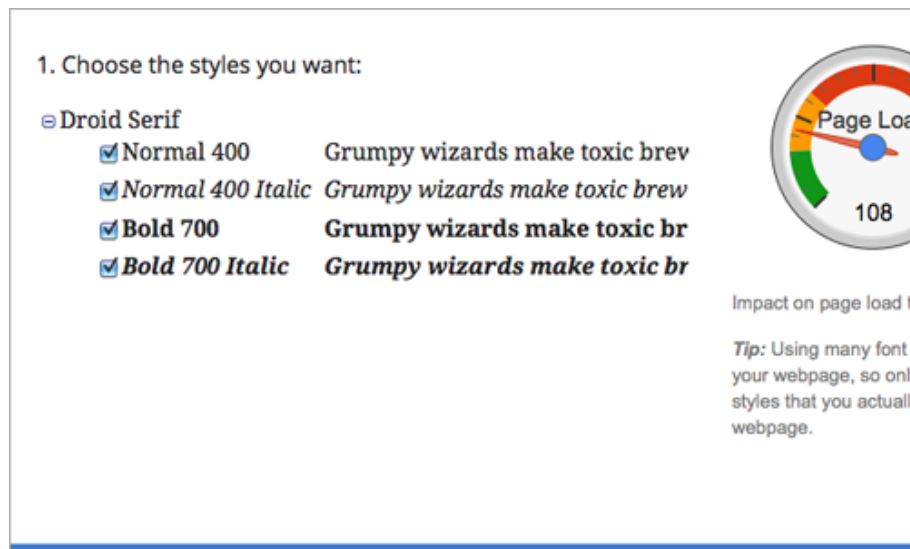
Droid Serif on Google Web Fonts

In response, Google Web Fonts will create a link to all four styles:

```
<link href="http://fonts.googleapis.com/
css?family=Droid+Serif:400,400italic,700,700italic"
rel="stylesheet" type="text/css">
```

This link points each browser to its own browser-specific source. For example, Firefox is taken to a document that returns the `@font-face` declarations below. Note that the declarations use the same font-family name each time. This would cause a problem in IE 7 and 8, which don't recognize multiple styles and weights that use the same font-family name.

```
@font-face {
font-family: 'Droid Serif';
font-style: normal;
font-weight: bold;
src: local('Droid Serif Bold'), local('DroidSerif-Bold'),
url('http://themes.googleusercontent.com/static/fonts/droidserif/
v3/QQt14e8dY39u-eYBZmppwTqR_3kx9_hJXbbyU8S6IN0.woff')
format('woff');
```

```css
}

@font-face {
font-family: 'Droid Serif';
font-style: italic;
font-weight: normal;
src: local('Droid Serif Italic'), local('DroidSerif-Italic'),
url('http://themes.googleusercontent.com/static/fonts/droidserif/
v3/cj2hUnSRBhwmSPr9kS5899kZXW4sYc4BjuAIFc1SXII.woff')
format('woff');
}

@font-face {
font-family: 'Droid Serif';
font-style: italic;
font-weight: bold;
src: local('Droid Serif Bold Italic'),
local('DroidSerif-BoldItalic'),
url('http://themes.googleusercontent.com/static/fonts/droidserif/
v3/c92rD_x0V1LslSFt3-QEpgRV2F9RPTaqyJ4QibDfkzM.woff')
format('woff');
}

@font-face {
font-family: 'Droid Serif';
font-style: normal;
font-weight: normal;
src: local('Droid Serif'), local('DroidSerif'),
url('http://themes.googleusercontent.com/static/fonts/droidserif/
v3/0AKsP294HTD-nvJgucYTaIbN6UDyHWBl620a-IRfuBk.woff')
format('woff');
}
```

One way around the "single font-family, multiple weights and styles" problem is to send IE 7 and 8 to a different source. Google Web Fonts does this, but unfortunately the @font-face declaration looks like this:

```css
@font-face {
font-family: 'Droid Serif';
font-style: normal;
font-weight: normal;
src: url('http://themes.googleusercontent.com/static/fonts/
droidserif/v3/0AKsP294HTD-nvJgucYTaGfQcKutQXcIrRfyR5jdjY8.eot');
src: local('Droid Serif'), local('DroidSerif'),
```

```
url('http://themes.googleusercontent.com/static/fonts/droidserif/
v3/0AKsP294HTD-nvJgucYTaGfQcKutQXcIrRfyR5jdjY8.eot')
format('embedded-opentype'),
url('http://themes.googleusercontent.com/static/fonts/droidserif/
v3/0AKsP294HTD-nvJgucYTaIbN6UDyHWBl620a-IRfuBk.woff')
format('woff');
}
```

This doesn't help us. The single declaration sets the `font-weight` to `normal` and `font-style` to `normal`, thus forcing IE 7 and 8 to "fake it" when requested to render bold and italic versions of the font.

The result? Faux bold and italic — even if we had carefully selected a font with bold and italic styles, and even if we had implemented all of the weights and styles as instructed.

Droid Serif. IE 8 delivers faux bold and italic (top) because it works only with a single `@font-face` *declaration and a single font-family name. Firefox delivers true bold and italic (bottom) because it can handle multiple weights and styles that are assigned to a single font-family.*

A COMMON FIX CAUSES PROBLEMS IN OPERA (AND AN-DROID)

A common "fix" for this problem is to insert separate links — one for each of the styles and weights used — into the head of the HTML document. That is, we select each weight and style of the font separately, taking the time to copy and paste each unique link into the head of the HTML document. The syntax for Droid Serif would look like this:

```
<link href="http://fonts.googleapis.com/css?family=Droid+Serif"
rel="stylesheet" type="text/css">
<link href="http://fonts.googleapis.com/
css?family=Droid+Serif:400italic" rel="stylesheet" type="text/
css">
<link href="http://fonts.googleapis.com/
css?family=Droid+Serif:700" rel="stylesheet" type="text/css">
<link href="http://fonts.googleapis.com/
css?family=Droid+Serif:700italic" rel="stylesheet" type="text/
css">
```

While this solves the problem in IE 7 and 8, referencing four CSS files for a single font family increases the number of requests that the client makes to the server and contributes to latency[151]. The fix also creates a new typographic problem in Opera (including Opera Mobile on Android). Opera renders text using the last weight and style served; so, if the last weight and style served is bold italic, then the font will come in as bold italic over the entire page.

---

151. https://developers.google.com/speed/docs/best-practices/rtt

Droid Serif. Using separate links to each weight and style fixes the problem in IE 7 and 8 (top), but it causes problems in Opera (bottom). Opera renders text using the last weight and style served.

USING A CONDITIONAL COMMENT WORKS ACROSS BROWSERS

There's a better solution: adding a conditional comment. IE 7 and 8 are the only browsers that need the fonts to be served separately. And because conditional comments work only in IE, the solution is solid. The new syntax looks like this:

```
<link href="http://fonts.googleapis.com/
css?family=Droid+Serif:400,400italic,700,700italic"
rel="stylesheet" type="text/css">
<!--[if IE]>
<link href="http://fonts.googleapis.com/css?family=Droid+Serif"
rel="stylesheet" type="text/css">
<link href="http://fonts.googleapis.com/
css?family=Droid+Serif:400italic" rel="stylesheet" type="text/
css">
<link href="http://fonts.googleapis.com/
css?family=Droid+Serif:700" rel="stylesheet" type="text/css">
<link href="http://fonts.googleapis.com/
```

```
css?family=Droid+Serif:700italic" rel="stylesheet" type="text/
css">
<![endif]-->
```

Notice how all browsers except IE are instructed to use the usual method for accessing Google Web Fonts. This keeps bold and italic fonts loading correctly (and more quickly) in Firefox, Opera, Chrome and Safari. Meanwhile, IE is instructed to access each weight and style separately. This fixes the faux bold and italic problem in IE 7 and 8, and it doesn't create any new problems in more recent versions of the browser.

Same *font* set large for a **heading**

Throughout history, we have used the written word to record and preserve who we are and what we care about: *possessions, laws, commitments, ideas, and*

Same *font* set large for a **heading**

Throughout history, we have used the written word to record and preserve who we are and what we care about: *possessions, laws, commitments, ideas, and memories.*

Same *font* set large for a **heading**

Throughout history, we have used the written word to record and preserve who we are and what we care about: *possessions, laws, commitments, ideas, and memories.*

Same *font* set large for a **heading**

Throughout history, we have used the written word to record and preserve who we are and what we care about: *possessions, laws, commitments, ideas, and memories.*

Same *font* set large for a **heading**

Throughout history, we have used the written word to record and preserve who we are and what we care about: *possessions, laws, commitments, ideas, and memories.*

Same *font* set large for a **heading**

Throughout history, we have used the written word to record and preserve who we are and what we care about: *possessions, laws, commitments, ideas, and memories.*

Using a conditional comment, we get true bold and italic to load correctly across browsers. Top to bottom: IE 8, IE 9, Firefox 11, Google Chrome 18, Safari 5, Opera 11.62.

Help Visitors Enjoy Their Reading Experience

If people are on your website, they're probably either skimming quickly looking for something or they've found what they're looking for and want to read it as easily as possible. Either way, keeping the text readable will help them achieve their goal.

BOLD AND ITALIC HELP ORGANIZE CONTENT

From a macro perspective, bold and italic forms of a font are important for people skimming your website. Bold and italic forms add emphasis — both **strong** and *subtle* — to text. They can help visitors understand the organization of content before they even start to read it.

Scanning with purpose is scanning down or across a text, jumping from section to section, *looking for a specific piece of information*. The reader may glance at the first letter or word of each section, dismissing incorrect matches and moving on.

Casual reading is skimming over a text, reading sentences here and there (the first sentence of each paragraph, the caption, the pull quote) *to get a general idea and flavor of the text.*

Sustained reading is engaged reading. It includes *pleasure reading* (pursued for its own sake) and *reading for understanding.* Readers slow down, read the entire text, and may go into a trancelike state.

Scanning with purpose is scanning down or across a text, jumping from section to section, looking for a specific piece of information. The reader may glance at the first letter or word of each section, dismissing incorrect matches and moving on.

Casual reading is skimming over a text, reading sentences here and there (the first sentence of each paragraph, the caption, the pull quote) to get a general idea and flavor of the text.

Sustained reading is engaged reading. It includes pleasure reading (pursued for its own sake) and reading for understanding. Readers slow down, read the entire text, and may go into a trancelike state.

Bold and italic create levels of emphasis, which helps to visually organize text (left). The same text without bold or italic (right) would feel more like a narrative.

TRUE BOLD AND ITALIC ARE EASIER TO READ

From a micro perspective, true bold and italic forms are important for people engaged in more sustained reading on your website. A proper balance between strokes and spaces improves legibility and makes text easier to read, thus minimizing brain fatigue and giving visitors a more pleasurable experience of the website. Type designers use their time and talent to create Web font families that are both legible and beautiful; for us, it's just a matter of getting the true weights and styles to load properly.

> **As typographers, our first responsibility is to our readers.** Our most important job is to help our readers *find, understand,* and *connect* with the words, ideas, and information they seek. Our second responsibility then, is to honor the content. We must help to clarify and share the meaning of the texts people read.

True bold and italic text is not just more visually pleasing, but also easier to read.

Waiting until Google fixes this problem might be tempting, but it's been on Google's radar since June 2010. Making sure that the bold and italic fonts served up by Google Web Fonts work across browsers is up to us. And it takes only a minute. Don't let 1 in 5 visitors to your website down. ❧

FURTHER RESOURCES

- "Say No to Faux Bold[152]" Alan Stearns

- "It's About Legibility[153]" Allan Haley

- "Is the Font Easy to Read? Anatomy and Legibility[154]" (PDF), Laura Franz
  This excerpt from the book *Typographic Web Design* is in PDF format. View it in Adobe Acrobat to compare fonts via rollovers.

---

152. http://www.alistapart.com/articles/say-no-to-faux-bold/
153. http://new.fonts.com/content/learning/fontology/level-4/fine-typography/legibility
154. http://goodwebfonts.com/legibility_twd.pdf

Setting Weights And Styles With The @font-face Declaration

BY LAURA FRANZ ❧

If people are on your website, they're probably either skimming quickly, looking for something, or they've found what they're looking for and want to read it as easily as possible. Either way, keeping text readable will help them achieve their goal.

BOLD AND ITALIC HELP TO ORGANIZE CONTENT

The previous chapter ended by showing that weights and styles are an important UX element when setting text. Bold and italic forms of a font help people to skim your website. They add emphasis — both **strong** and *subtle* — that can help visitors understand the organization of content before even starting to read it.

Scanning with purpose is scanning down or across a text, jumping from section to section, *looking for a specific piece of information*. The reader may glance at the first letter or word of each section, dismissing incorrect matches and moving on.

Casual reading is skimming over a text, reading sentences here and there (the first sentence of each paragraph, the caption, the pull quote) *to get a general idea and flavor of the text*.

Sustained reading is engaged reading. It includes *pleasure reading* (pursued for its own sake) and *reading for understanding*. Readers slow down, read the entire text, and may go into a trancelike state.

Scanning with purpose is scanning down or across a text, jumping from section to section, looking for a specific piece of information. The reader may glance at the first letter or word of each section, dismissing incorrect matches and moving on.

Casual reading is skimming over a text, reading sentences here and there (the first sentence of each paragraph, the caption, the pull quote) to get a general idea and flavor of the text.

Sustained reading is engaged reading. It includes pleasure reading (pursued for its own sake) and reading for understanding. Readers slow down, read the entire text, and may go into a trancelike state.

Weights and styles are an important UX element. Bold and italic help readers to see structure and to skim the text more efficiently (left). The same text without bold or italic (right) feels more like a narrative.

In this chapter, we'll start where we left off. Because weights and styles help our visitors to read our content, we should make sure they work! And getting weights and styles to work correctly using the @font-face

declaration can be a bit crazy-making. Let's look at two popular approaches to setting weights and styles with the `@font-face` declaration. I'll show you why they are not the best solutions, and show you a third more effective approach to follow.

Unique Font-Family Names, Normal Weights And Styles

If you've used one of FontSquirrel[155]'s amazing `@font-face` kits, then you're familiar with this approach to setting weights and styles. The CSS provided in every kit uses a *unique* font-family name for each weight and style, and sets the weight and style in the `@font-face` declaration to `normal`.

For example, the syntax for Ubuntu Italic and Ubuntu Bold looks like this:

```
@font-face {
    font-family: 'UbuntuItalic';
      src: url('Ubuntu-RI-webfont.eot');
      src: url('Ubuntu-RI-webfont.eot?#iefix')
format('embedded-opentype'),
           url('Ubuntu-RI-webfont.woff') format('woff'),
           url('Ubuntu-RI-webfont.ttf') format('truetype'),
           url('Ubuntu-RI-webfont.svg#UbuntuItalic')
format('svg');
    font-weight: normal;
    font-style: normal;
}

@font-face {
    font-family: 'UbuntuBold';
      src: url('Ubuntu-B-webfont.eot');
      src: url('Ubuntu-B-webfont.eot?#iefix')
format('embedded-opentype'),
           url('Ubuntu-B-webfont.woff') format('woff'),
           url('Ubuntu-B-webfont.ttf') format('truetype'),
           url('Ubuntu-B-webfont.svg#UbuntuBold') format('svg');
    font-weight: normal;
    font-style: normal;
}
```

---

155. http://www.fontsquirrel.com

Notice that the font-family names are unique, with each font-family name accessing the appropriate Web font files. For example, `Ubuntu-Italic` accesses `Ubuntu-RI-webfont.woff`, while `UbuntuBold` accesses `Ubuntu-B-webfont.woff`.

Also, notice that the `font-weight` and `font-style` for both `@font-face` declarations are set to `normal`.

STYLING TEXT USING THIS METHOD

To style text using this method, use the appropriate font-family name, and keep all weights and styles set to `normal`. For example, the Regular, Regular Italic, Bold and Bold Italic headings below are set with classes. The classes are styled like so:

```css
.u400 {
    font-family: 'UbuntuRegular', arial, sans-serif;
    font-weight: normal;
    font-style: normal;
}

.u400i {
    font-family: 'UbuntuRegularItalic', arial, sans-serif;
    font-weight: normal;
    font-style: normal;
}

.u700 {
    font-family: 'UbuntuBold', arial, sans-serif;
    font-weight: normal;
    font-style: normal;
}

.u700i {
    font-family: 'UbuntuBoldItalic', arial, sans-serif;
    font-weight: normal;
    font-style: normal;
}
```

Ubuntu Regular (400)

Ubuntu Regular Italic

Ubuntu Bold (700)

Ubuntu Bold Italic

Ubuntu Regular (400)

Ubuntu Regular Italic

Ubuntu Bold (700)

Ubuntu Bold Italic

Ubuntu Regular, Italic, Bold and Bold Italic on Windows 7 in Internet Explorer 8 (top) and on Mac OS X in Chrome 23 (bottom). Unique font-family names with normal weights and styles (á la FontSquirrel) work fine.

MAKE SURE THE WEIGHTS AND STYLES MATCH!

Because the weights and styles are set to `normal` in the `@font-face` declarations, keeping the weights and styles set to `normal` when styling the text is important. Otherwise, the bolds may double-bold (some browsers will add a bold weight to the already bold Web font), and the italics may double-italic (some browsers will add an italic style to the already italic Web font).

For example, let's say we *incorrectly* set the `font-style` to `italic` and the `font-weight` to `700` (bold) in our corresponding classes:

```
.u400 {
  font-family: 'UbuntuRegular', arial, sans-serif;
  font-weight: normal;
  font-style: normal;
```

```css
}

.u400i {
    font-family: 'UbuntuRegularItalic', arial, sans-serif;
    font-weight: normal;
    font-style: italic;
}

.u700 {
    font-family: 'UbuntuBold', arial, sans-serif;
    font-weight: 700;
    font-style: normal;
}

.u700i {
    font-family: 'UbuntuBoldItalic', arial, sans-serif;
    font-weight: 700;
    font-style: italic;
}
```

The fonts will render incorrectly in Mac OS X browsers and on iPad Safari browsers.

When using normal weights and styles in the `@font-face` *declaration, give text elements normal weights and styles, too. Otherwise, your text may end up with a double-bold and double-italic. If you set text elements as bold or italic, then Ubuntu Italic, Bold and Bold Italic won't double-bold or double-italic on Windows 7 in Internet Explorer 8 (top). But look at what happens on Mac OS X in Firefox 17 and on iPad 3 with iOS 5.1 in Safari (bottom) — yikes!*

USING <EM> AND <STRONG> ELEMENTS

While `<em>` and `<strong>` can be styled to communicate emphasis and importance in a variety of ways, they are often used in their default forms: with `<em>` set to italic text and `<strong>` set to bold text.

For example, the paragraph below is styled like so:

```css
p {
    font-family: 'UbuntuRegular', arial, sans-serif;
    font-weight: normal;
    font-style: normal;
}
```

And the `<em>` and `<strong>` elements are left in their default states:

```css
em {
    font-style: italic;
}


strong {
    font-weight: bold;
}
```

A paragraph of text set using unique font-family names. *Using the <em> element to set italics. Will it work?* **Using the <strong> element to set bold. Will it work?**

Applying `<em>` and `<strong>` in their default states results in faux italic and faux bold.

The result is a faux italic and a faux bold. Why? Because the paragraph is set to Ubuntu Regular, with the weight and style set to `normal`. When the `<em>` element is applied, the text remains Ubuntu Regular but is slanted to look like it's italic. Notice the angular "e" and the double-story "a." When the `<strong>` element is applied, the text remains Ubuntu Regular but is stretched to look like it's bold. Notice the letter "e" is no longer monoline — the sides of the letter look thicker than the top and bottom of the letter.

We can fix this problem by making sure the `<em>` and `<strong>` elements use the correct font-family name. For example, the paragraph below continues to be styled like so:

```css
p {
    font-family: 'UbuntuRegular', arial, sans-serif;
    font-weight: normal;
    font-style: normal;
}
```

And the `<em>` and `<strong>` elements are styled to use the correct corresponding font-family names:

```
em {
    font-family: 'UbuntuRegularItalic', arial, sans-serif;
    font-weight: normal;
    font-style: normal;
}

strong {
    font-family: 'UbuntuBold', arial, sans-serif;
    font-weight: normal;
    font-style: normal;
}
```

Notice that the `font-weight` and `font-style` for both `<em>` and `<strong>` are set to `normal`. This is counterintuitive, but necessary so that the text doesn't end up with a double-italic or a double-bold.

A paragraph of text set using unique font-family names. *Using the <em> element to set italics. Will it work?* **Using the <strong> element to set bold. Will it work?**

Using the correct font-family names — and setting weights and styles to `normal` *— results in true italic and true bold.*

The result is a true italic and true bold. Why? Because while the paragraph is set to Ubuntu Regular, the `<em>` element is set to Ubuntu Italic and the `<strong>` element is set to Ubuntu Bold — and all weights and styles are set to `normal`.

PROBLEM: IF THE FALLBACK FONT LOADS, WEIGHTS AND STYLES WILL BE LOST

While purposely stripping out weights and styles is counterintuitive, using a unique font-family name and normal weights and styles does work — as long as the Web font loads.

If the fallback font loads, then all bolds and italics will be lost — because we had set all weights and styles to `normal` — thus making it harder for readers to see the structure of your website's content and making it harder for them to skim the text.

If the Web font doesn't load, then the fallback font (here, Times New Roman) will load with a normal weight and style — undermining hierarchy and readability. Remember, we had set all weights and styles to `normal` *when we styled the elements!*

THE SHORT VERSION

Using unique font-family names combined with setting `font-weight` and `font-style` to `normal` is unforgiving. Mismatching weights and styles could easily result in either faux italic and faux bold or double-italic and double-bold. If the Web font doesn't load, then the result will be *no* italic or bold! So, this approach to setting weights and styles using the `@font-face` declaration isn't the best solution.

Style Linking

Another way to set weights and styles is to use the same font-family name multiple times, setting the weights and styles in each `@font-face` declaration to match the weight and style of the Web font file being accessed. This approach is called style linking.

For example, using style linking, the syntax for Ubuntu Italic and Ubuntu Bold would look like this:

```
@font-face {
  font-family: 'Ubuntu';
    src: url('Ubuntu-RI-webfont.eot');
```

```
    src: url('Ubuntu-RI-webfont.eot?#iefix')
format('embedded-opentype'),
        url('Ubuntu-RI-webfont.woff') format('woff'),
        url('Ubuntu-RI-webfont.ttf') format('truetype'),
        url('Ubuntu-RI-webfont.svg#UbuntuItalic')
format('svg');
    font-weight: 400;
    font-style: italic;
}


@font-face {
    font-family: 'Ubuntu';
    src: url('Ubuntu-B-webfont.eot');
    src: url('Ubuntu-B-webfont.eot?#iefix')
format('embedded-opentype'),
        url('Ubuntu-B-webfont.woff') format('woff'),
        url('Ubuntu-B-webfont.ttf') format('truetype'),
        url('Ubuntu-B-webfont.svg#UbuntuBold') format('svg');
    font-weight: 700;
    font-style: normal;
}
```

Notice that the font-family names are the same, regardless of what Web font file is being accessed. For example, `Ubuntu` accesses `Ubuntu-RI-webfont.woff`, and `Ubuntu` also accesses `Ubuntu-B-webfont.woff`. How does that work?

Notice that the `font-weight` and `font-style` for each `@font-face` declaration is set to match the weight and style of the Web font file being accessed. The `Ubuntu` that accesses the *italic* Web font file has `font-style: italic`. The `Ubuntu` that accesses the *bold* Web font file has `font-weight: 700`.

In this method, the weights and styles in the `@font-face` declarations act as "markers." When a browser encounters those weights and styles elsewhere in the CSS, it knows which @font-family declaration to use and which Web font files to access.

STYLING TEXT USING STYLE LINKING

To style text with this method, use the same font family for all versions of the font. Set weights and styles to trigger the correct Web font files that the browser should access. If you want the italic version of the font, make sure to set `font-style: italic`. For example, the Regular, Regular Italic, Bold and Bold Italic headings below are set with classes. The classes are styled like so:

```css
.u400 {
   font-family: 'Ubuntu', arial, sans-serif;
   font-weight: 400;
   font-style: normal;
}

.u400i {
   font-family: 'Ubuntu', arial, sans-serif;
   font-weight: 400;
   font-style: italic;
}

.u700 {
   font-family: 'Ubuntu', arial, sans-serif;
   font-weight: 700;
   font-style: normal;
}

.u700i {
   font-family: 'Ubuntu', arial, sans-serif;
   font-weight: 700;
   font-style: italic;
}
```

Ubuntu Regular, Italic, Bold and Bold Italic on Windows 7 in Internet Explorer 8 (top) and on Mac OS X in Chrome 23 (bottom). Using style linking to set weight and style appears to work fine.

AGAIN, MAKE SURE THE WEIGHTS AND STYLES MATCH!

Because weight and style are used to "trigger" the correct `@font-face` declaration, setting weights and styles to match those used in the `@font-face` declarations is important. As a bonus, when default rules for weights and styles are applied by browsers — like italic for `<em>` and bold for `<strong>` — then the correct fonts will automatically load (as long as your font has a bold and italic version), because the default rules will also trigger the correct `@font-face` declaration.

A paragraph of text set using style linking. *Using the <em> element to set italics. Will it work?* **Using the <strong> element to set bold. Will it work?**

When default rules for weights and styles are applied by browsers — like italic for `<em>` *and bold for* `<strong>` *— then the correct fonts will automatically load (as long as your font has a bold and italic version). Again, using style linking to set weight and style appears to work fine.*

BONUS: IF THE FALLBACK FONT LOADS, WEIGHTS AND STYLES WILL BE RETAINED

Unlike the first approach, setting weights and styles with style linking means that weights and styles will be retained even when the Web font fails to load. Why? Because all weights and styles were set correctly (for example, not `normal`) when styling the text.

Note: If the Web font doesn't load, then the fallback font (here, Times New Roman) will still provide hierarchy — bolds and italics will remain intact. Remember, we set all weights and styles to match the Web fonts when we styled the elements!

But style linking has its own problems…

PROBLEM: INTERNET EXPLORER 7 AND 8 CAN ONLY APPLY UP TO FOUR WEIGHTS AND STYLES TO A SINGLE FONT-FAMILY NAME

Style linking works — as long as you're not using more than four weights and styles (and as long as the person reading your Web page isn't the one of the one in eight people who still use Internet Explorer (IE) 7 or 8). If you use more than four weights and styles on the website, then IE 7 and 8 will convert all `light` and `medium` weights to normal, and all `black` and `heavy` weights to bold — so, you'll lose some of your carefully set text.

Ubuntu Light (300)
Ubuntu Light Italic
Ubuntu Regular (400)
Ubuntu Regular Italic
Ubuntu Medium (500)
Ubuntu Medium Italic
Ubuntu Bold (700)
Ubuntu Bold Italic

IE 7 and 8 can't apply more than four weights and styles to a single font-family name. Ubuntu has eight weights and styles. If you used all eight with the font-family name `Ubuntu`, *then the Light, Light Italic, Medium and Medium Italic would convert to Regular and Regular Italic.*

PROBLEM: CRASHES BROWSERS ON BLACKBERRY AND IPAD 1

No matter how well your fonts load, if the page crashes the browser, people won't be able to see your content! And as of this writing, Web pages that use style linking crash BlackBerry 9900 browsers on a regular basis. They also crash iPad 1 browsers 100% of the time.

THE SHORT VERSION

Using style linking appears to be more forgiving. It greatly reduces the potential for setting text that is faux italic and faux bold, or double-italic and double-bold. In addition, if the Web font doesn't load, the text will retain both style and weight. But if you're using more than four weights and styles, then text won't render properly in IE 7 and 8. In addition, style linking crashes the browsers on BlackBerry and iPad 1 de-

vices. So, this approach to setting weights and styles using the `@font-face` declaration isn't the best solution at this time (although the future — when IE 7 and 8, BlackBerry 9900 and iPad 1 are all defunct — looks bright!).

Unique Font-Family Names, Matching Weights And Styles

A third — and currently the most effective — way to set weights and styles is to merge the first two methods. If you've used FontDeck[156] for Web fonts, then you'll be familiar with this approach.

Use the *unique* font-family names — which allows IE to show all of the weights and styles you need. Also, use matching weights and styles (for example, do *not* set weights and styles to `normal`), which will keep the weights and styles intact should the Web font fail.

For example, the syntax for Ubuntu Italic and Ubuntu Bold would like this:

```
@font-face {
    font-family: 'UbuntuItalic';
        src: url('Ubuntu-RI-webfont.eot');
        src: url('Ubuntu-RI-webfont.eot?#iefix')
format('embedded-opentype'),
            url('Ubuntu-RI-webfont.woff') format('woff'),
            url('Ubuntu-RI-webfont.ttf') format('truetype'),
            url('Ubuntu-RI-webfont.svg#UbuntuItalic')
format('svg');
    font-weight: 400;
    font-style: italic;
}

@font-face {
    font-family: 'UbuntuBold';
        src: url('Ubuntu-B-webfont.eot');
        src: url('Ubuntu-B-webfont.eot?#iefix')
format('embedded-opentype'),
            url('Ubuntu-B-webfont.woff') format('woff'),
            url('Ubuntu-B-webfont.ttf') format('truetype'),
            url('Ubuntu-B-webfont.svg#UbuntuBold') format('svg');
    font-weight: 700;
```

---

156. http://fontdeck.com

```
  font-style: normal;
}
```

Notice that the font-family names are unique, with each font-family name accessing the appropriate Web font files. For example, `Ubuntu-Italic` accesses `Ubuntu-RI-webfont.woff`, while `UbuntuBold` accesses `Ubuntu-B-webfont.woff`.

Also, notice the weight and style: The `font-weight` and `font-style` for each `@font-face` declaration is set to match the weight and style of the Web font file being accessed. The `UbuntuItalic` that accesses the italic Web font file has `font-style: italic`. The `UbuntuBold` that accesses the bold Web font file has `font-weight: 700`.

STYLING TEXT USING THE COMBINED METHOD

To style text with this method, use the unique font-family names, and set weights and styles to match those used in the `@font-face` declarations. For example, the Light, Light Italic, Regular, Regular Italic, Medium, Medium Italic, Bold and Bold Italic headings below are set with classes. The classes are styled like so:

```
.u300 {
  font-family: 'UbuntuLight', arial, sans-serif;
  font-weight: 300;
  font-style: normal;
}

.u300i {
  font-family: 'UbuntuLightItalic', arial, sans-serif;
  font-weight: 300;
  font-style: italic;
}

.u400 {
  font-family: 'UbuntuRegular', arial, sans-serif;
  font-weight: 400;
  font-style: normal;
}

.u400i {
  font-family: 'UbuntuRegularItalic', arial, sans-serif;
  font-weight: 400;
  font-style: italic;
}
```

```css
.u700 {
    font-family: 'UbuntuMedium', arial, sans-serif;
    font-weight: 500;
    font-style: normal;
}

.u700i {
    font-family: 'UbuntuMediumItalic', arial, sans-serif;
    font-weight: 500;
    font-style: italic;
}

.u900 {
    font-family: 'UbuntuBold', arial, sans-serif;
    font-weight: 700;
    font-style: normal;
}

.u900i {
    font-family: 'UbuntuBoldItalic', arial, sans-serif;
    font-weight: 700;
    font-style: italic;
}
```

Ubuntu Light (300)

Ubuntu Light Italic

Ubuntu Regular (400)

Ubuntu Regular Italic

Ubuntu Medium (500)

Ubuntu Medium Italic

Ubuntu Bold (700)

Ubuntu Bold Italic

By combining unique font-family names with matching weights and styles, all eight weights and styles of Ubuntu will work — even on Windows 7 in IE 8. And BlackBerry and iPad 1 browsers won't crash.

AGAIN, MAKE SURE THE WEIGHTS AND STYLES MATCH!

Although using unique font-family names means we no longer need the weights and styles to "trigger" the correct `@font-face` declaration, setting all weights and styles to match is important. Applying weights and styles to text elements will keep them intact in case the Web font fails. Also, because the weights and styles match, the text won't end up with either a double-bold or double-italic, or a faux bold or faux italic.

PROBLEM: USING `<EM>` AND `<STRONG>` ELEMENTS

Using unique font-family names brings back an earlier problem: getting `<em>` and `<strong>` elements to work properly. Default styling on these elements will result in faux italic and faux bold text.

For example, the three paragraphs below are styled like so:

```css
p.light {
   font-family: 'UbuntuLight', arial, sans-serif;
   font-weight: 300;
   font-style: normal;
}


p {
   font-family: 'UbuntuRegular', arial, sans-serif;
   font-weight: normal;
   font-style: normal;
}


p.medium {
   font-family: 'UbuntuMedium', arial, sans-serif;
   font-weight: 500;
   font-style: normal;
}
```

And the `<em>` and `<strong>` elements are left in their default states:

```css
em {
   font-style: italic;
}


strong {
   font-weight: bold;
}
```

A paragraph of light (300) text set using unique font-names and style linking. *Using the <em> element to set italics. Will it work?* Using the <strong> element to set bold. Will it work?

A paragraph of text set using unique font-names and style linking. *Using the <em> element to set italics. Will it work?* **Using the <strong> element to set bold. Will it work?**

A paragraph of medium (500) text set using unique font-names and style linking. *Using the <em> element to set italics. Will it work?* **Using the <strong> element to set bold. Will it work?**

Applying <em> and <strong> in their default states will result in faux italic and (usually) faux bold.

The result is a faux italic and a faux bold (when it's applied). Why? Because we're using unique font-family names. Let's look at the middle paragraph. It's set to Ubuntu Regular, with weight and style set to match (`weight: normal` and `style: normal`). When the <em> element is applied, the text remains Ubuntu Regular but is slanted to look like it's italic. When the <strong> element is applied, the text remains Ubuntu Regular but is stretched to look like it's bold.

We can fix the bold text by making sure the <strong> element uses the correct font-family name, weight and style:

```
strong {
    font-family: 'UbuntuBold', arial, sans-serif;
    font-weight: 700;
    font-style: normal;
}
```

The <em> is a bit more difficult to fix. We can set it to Ubuntu Regular Italic and use the matching weight and style:

```
em {
    font-family: 'UbuntuRegularItalic', arial, sans-serif;
    font-weight: 400;
    font-style: italic;
}
```


Setting <em> to UbuntuRegularItalic and <strong> to UbuntuBold will result in an italic that's too heavy for light text and too light for medium text.

The result is a bold and italic that are consistent throughout, regardless of the weight of the paragraph text. This may be fine for the <strong> element, but the <em> element looks a bit out of place on both the light and medium text.

The only way to fix this is to create a variety of classes for the <em> element, like so:

```
em {
    font-family: 'UbuntuRegularItalic', arial, sans-serif;
    font-weight: 400;
    font-style: italic;
}
```

```css
em.light {
    font-family: 'UbuntuLightItalic', arial, sans-serif;
    font-weight: 300;
    font-style: italic;
}

em.medium {
    font-family: 'UbuntuMediumItalic', arial, sans-serif;
    font-weight: 500;
    font-style: italic;
}
```

A paragraph of light (300) text set using unique font-names and style linking. *Using the <em> element to set italics. Will it work?* **Using the <strong> element to set bold. Will it work?**

A paragraph of text set using unique font-names and style linking. *Using the <em> element to set italics. Will it work?* **Using the <strong> element to set bold. Will it work?**

A paragraph of medium (500) text set using unique font-names and style linking. *Using the <em> element to set italics. Will it work?* **Using the <strong> element to set bold. Will it work?**

Creating three versions of the `<em>` *element (* `em`*,* `em.light`*,* `em.medium`*) will result in italics that "match" the weight of the text.*

This will result in italics that match the weight of the text in each paragraph. This creates much better hierarchy, but it undermines the beauty of cascading style sheets and the simplicity of using a single `<em>` element.

BONUS: IF THE FALLBACK FONT LOADS, WEIGHTS AND STYLES WILL BE RETAINED

Setting all weights and styles correctly (for example, not `normal`) when styling the text means that basic weights and styles will be retained even if the Web font doesn't load.

Ubuntu Light (300)

Ubuntu Light Italic

Ubuntu Regular (400)

Ubuntu Regular Italic

Ubuntu Medium (500)

Ubuntu Medium Italic

Ubuntu Bold (700)

Ubuntu Bold Italic

A paragraph of light (300) text set using unique font-names and style linking. *Using the <em> element to set italics. Will it work?* **Using the <strong> element to set bold. Will it work?**

A paragraph of text set using unique font-names and style linking. *Using the <em> element to set italics. Will it work?* **Using the <strong> element to set bold. Will it work?**

A paragraph of medium (500) text set using unique font-names and style linking. *Using the <em> element to set italics. Will it work?* **Using the <strong> element to set bold. Will it work?**

If the Web font doesn't load, then the fallback font (here, Times New Roman) will still provide hierarchy — bolds and italics will remain intact. But because fallback fonts don't usually have "extra" weights (such as Light, Medium and Black), some weights will revert to Normal and Bold.

Using unique font-family names *and* setting weights and styles to match (for example, not setting them to `normal`) is the most effective method of getting weights and styles to work with the `@font-face` declaration. An extra step is required (i.e. setting and using unique font-family names — even with elements such as `<em>` and `<strong>`) to avoid faux italic and faux bold, but the solution does reduce the potential for double-italic and double-bold text. In addition, if the Web font doesn't load, then the text will retain both style and weight. Finally, text will render properly in IE 7 and 8 (even if you're using more than four weights and styles), and your pages won't crash the browsers on BlackBerry 9900 and iPad 1 devices.

Make Sure Your Weights And Styles Work

Weights and styles help visitors read the content on your website. So, make sure your weights and styles load correctly — whether for IE 8, Chrome, Mac OS X Firefox or an iPad and whether or not the Web font loads correctly (or visitors see a fallback font).

When choosing which method to use for your website, ask yourself two questions:

1. Will I be using more than four weights and styles on the website?

2. Are BlackBerry and iPad 1 users part of my target audience?

If the answer to both questions is "no," then go ahead and use style linking. It's elegant and forgiving.

If the answer to either question is "yes," then use the combined method. Currently, the only way to make sure your weights and styles work cross-browser is to use unique font-family names in conjunction with matching weights and styles (and, of course, to make sure your font has a bold and italic version). This might sound like a chore, requiring extra care and typing (`UbuntuRegularItalic` is longer to type than `Ubuntu`, after all), but as with most typographic details, the best results take time and effort.

Even Typekit[157], which uses JavaScript to handle the intricacies of Web fonts, requires variation-specific font-family names to support older versions of IE.

Once IE 7 and 8, BlackBerry 9900 browsers and the iPad 1 are defunct, using style linking alone *should* be enough to get the job done.

---

157. https://typekit.com

But as of 28 January 2013, StatCounter[158] reports that IE 7 and 8 alone were used for 12.3% of the 45 billion page views collected from November 2012 to January 2013. Information on mobile browsers is not clear, but a minimum of 1 in 8 visitors still need us to go the extra mile. So, for now, we've got some extra typing to do. ❧

FURTHER RESOURCES

- "Avoiding Faux Weights and Styles with Google Web Fonts[159]," Laura Franz, Smashing Magazine

- "Say No to Faux Bold[160]," Alan Stearns, A List Apart

- "New From Typekit: Variation-Specific Font-Family Names in IE 6–8[161]," Typekit blog

158. http://gs.statcounter.com
159. http://www.smashingmagazine.com/2012/07/11/avoiding-faux-weights-styles-google-web-fonts/
160. http://www.alistapart.com/articles/say-no-to-faux-bold/
161. http://blog.typekit.com/2011/06/27/new-from-typekit-variation-specific-font-family-names-in-ie-6-8/

About The Authors

Carolyn Knight & Jessica Glaser

Carolyn Knight and Jessica Glaser are academics from the University of Wolverhampton in the UK, graphic designers, and prolific design writers. Their numerous books focus on topics including the use of space in graphic design, mnemonics and memory devices and the understanding and creation of visual hierarchy. Their latest book is the best selling 'Graphic Design Exercise Book', published in English by RotoVision and in Spanish by Editorial Gustavo Gili. Their company Bright Pink Communication Design, works in such areas as healthcare, construction, education, financial services and the public sector.

Heydon Pickering

Heydon is a Web designer from Norwich in the UK. He blogs about beautiful code and ugly music at heydonworks.com[162]. His hermit crabs are called Ian and Phillip and he's made a children's book[163] all about different crab species. Twitter: @heydonworks[164]

David Kadavy

David Kadavy is the author of Design for Hackers[165], which debuted at #18 on all of Amazon. Sign up for email updates[166] and he'll send you a pretty little PDF cheat sheet highlighting these typography tips. You can visit his website[167] and follow him on Twitter[168].

Laura Franz

Laura Franz is a Professor at UMass Dartmouth, where she teaches a wide range of type classes — including a Web Typography class in The Graduate Certificate In Web and Interaction Design. Inspired by the intersection of tradition and technology, Laura shares her Web font rec-

162. http://www.heydonworks.com/
163. http://www.crabsareodd.com/
164. http://www.twitter.com/heydonworks
165. http://www.amazon.com/Design-Hackers-Reverse-Engineering-Beauty/dp/1119998956
166. http://digioh.com/v/k/10033
167. http://www.kadavy.net
168. http://www.twitter.com/kadavy

ommendations on goodwebfonts.com[169] and her typography knowledge via "Typographic Web Design: How to Think Like a Typographer in HTML and CSS" (Wiley), "Typography for Web Designers"[170], and "Choosing and Using Web Fonts"[171].

Dan Reynolds

Dan Reynolds is a Berlin-based type designer. After studies in Germany, the UK and the US, he now works as a doctoral candidate and research assistant at the Braunschweig University of Art. You can visit his blog[172] and follow him on Twitter[173].

Nathan Ford

Nathan Ford is a Texas-born designer/developer happily working at Mark Boulton Design[174] in Penarth, Wales. His free time is consumed creating things – some you may have used before, including tools like Fount[175], MIN[176], and Unify[177]. Read more of his work at Art=Work[178] or follow the author on Twitter[179]

169. http://www.goodwebfonts.com/
170. http://www.lynda.com/Web-Fonts-tutorials/Typography-for-Web-Designers/79411-2.html
171. http://www.lynda.com/course-tutorials/Choosing-Using-Web-Fonts/97715-2.html
172. http://www.typeoff.de/
173. http://www.twitter.com/typeoff
174. http://markboultondesign.com/
175. http://fount.artequalswork.com/
176. http://min.artequalswork.com/
177. http://unify.unitinteractive.com/
178. http://artequalswork.com/
179. http://www.twitter.com/nathan_ford

About Smashing Magazine

Smashing Magazine[180] is an online magazine dedicated to Web designers and developers worldwide. Its rigorous quality control and thorough editorial work has gathered a devoted community exceeding half a million subscribers, followers and fans. Each and every published article is carefully prepared, edited, reviewed and curated according to the high quality standards set in Smashing Magazine's own publishing policy[181].

Smashing Magazine publishes articles on a daily basis with topics ranging from business, visual design, typography, front-end as well as back-end development, all the way to usability and user experience design. The magazine is — and always has been — a professional and independent online publication neither controlled nor influenced by any third parties, delivering content in the best interest of its readers. These guidelines are continually revised and updated to assure that the quality of the published content is never compromised.

About Smashing Media GmbH

Smashing Media GmbH[182] is one of the world's leading online publishing companies in the field of Web design. Founded in 2009 by Sven Lennartz and Vitaly Friedman, the company's headquarters is situated in southern Germany, in the sunny city of Freiburg im Breisgau. Smashing Media's lead publication, Smashing Magazine, has gained worldwide attention since its emergence back in 2006, and is supported by the vast, global Smashing community and readership. Smashing Magazine had proven to be a trustworthy online source containing high quality articles on progressive design and coding techniques as well as recent developments in the Web design industry.

180. http://www.smashingmagazine.com
181. http://www.smashingmagazine.com/publishing-policy/
182. http://www.smashing-media.com