The Art and Tradition of Typography

FontBlog 25 Jun 2010 6:31 PM | 5

The Art and Tradition of Typography

For over 25 years Microsoft has been very focused on the development of type and type technologies. In order to fully understand the technical foundations of typography in Windows, a brief overview of some of the highlights of "typographic engineering" from the past 500 years can add some useful insight. Now, by referring to 500 years of type, there is a clear reference to Johannes Gutenberg and his involvement in the development of moveable metal type in Europe. Although much of this discussion is centered on the development of type and typography for Latin based scripts, there is an equivalent rich history of other type scripts throughout the world, and I'll attempt to make reference to these throughout this article.

The craftsmen who created the type for printing presses were part engineer and part artisan, and had many technical challenges to solve. The first step in creating a piece of lead type involves the process of punchcutting. This process involves carving a three dimensional image (in reverse) of each character of the font into the end of a steel punch—a different image and a separate punch for each font's size. The characters at different sizes were not typically scaled clones of other sizes; instead each size had its own attributes for the font, based on the size at which the reader would view the text. In optical scaling, as it is called, smaller text sizes typically have larger x-heights, wider stem widths, and less typographic stem contrast and larger display sizes have smaller x-heights with more variation in stem widths.

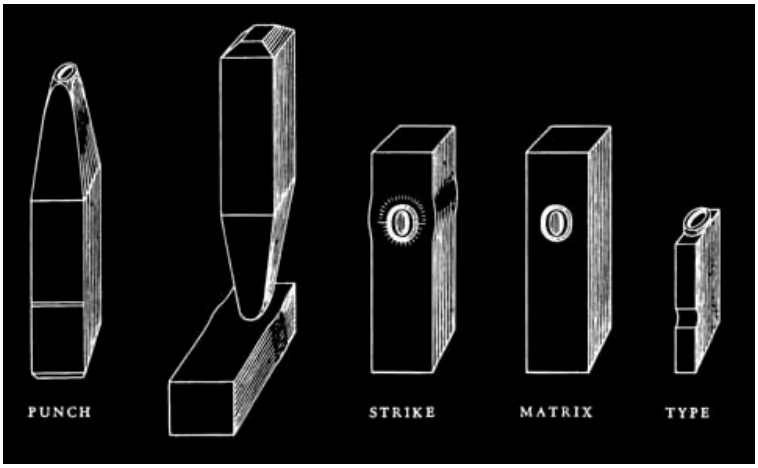

Figure 1: *Example of a Punch*

The measurement of font sizes we use in the computer world is generally inherited from the days of punch cutting and the printing press. The punches were used to create a mold, and the mold was used to create a reversed image on the end of a metal rod, called a sort. The size of the typeface is measured by the size of the end of the sort (vertically in relation to the character), which is called an "em" in typography. Different typefaces may take up a different proportion of that em space; this is why some typefaces of the same "size" look as though they're different sizes. In the computer domain, the unit of measurement for font sizes is traditionally points. A point is equal to 1/72 of an inch or around 0.353 mm. It is generally much better to use true sizes such as points or millimeters or inches when referencing text, instead of pixels, in order for techniques like optical scaling to work well for the reader of the text. Human eyes, after all, are not scalable.

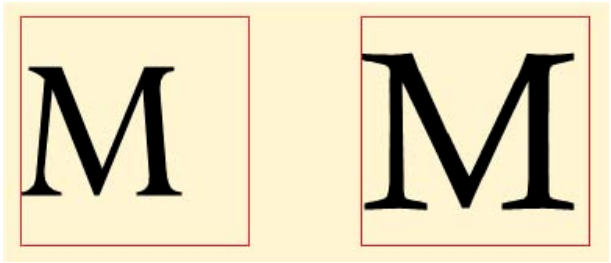

Figure 2: *The letter "M", on the left in Perpetua and on the right in Calisto, inside squares of one em on each side.*

The punchcutters also had to be aware of the technical capabilities of the printing press and the papers and inks used on the press. Different inks and papers have different ink-spread attributes and the typeface had to be designed to account for this. Punchcutters would make tiny adjustments to how the strokes fitted together, as well as to the weights and typographic contrast of the stems.

Many characters themselves presented challenges for lead type. Some characters, when placed adjacent, will interfere with each other. In Latin text, the 'f' and 'i' combination is an example of such an awkward combination. To deal with this issue, ligatures were created that combined the two character forms into one glyph. (Historically ligatures, in the form of abbreviations, were also used for text spacing by providing alternate glyphs — such as the ampersand, originally a combination of **e** + **t** — that could be used to justify the line of text. This technique was not original to metal typesetting, as early medieval scribes often would

make combinations of letters in order to make a line of text fit on a page.)

Figure 3: *Ligatures. Top: Times New Roman Italic fi and fl combination without ligatures. Bottom: Times New Roman Italic, fi and fl combination with ligatures.*

Type designers and typographers throughout the era of metal type strove to advance the beauty and utility of text on paper. The use of small caps, designed to fit with the weight of the lower case glyphs, the use of old style numerals, and the careful and deliberate use of hyphenation and justification all helped to create a page of text with what typographers call "even color" — meaning that there is a smooth, even tone to the text, with no blotches of dark or rivers of light breaking up the page. The beauty of a well-designed book or magazine page makes the reading experience easier.

Figure 4: *Top, Constantia, Cap M, Cap M electronically scaled to Small Cap height, lowercase m. Bottom: Constantia, Cap M, True Small Cap M, lowercase m. (Note that the overall weight of the true small cap is designed to be in harmony with the lowercase.*

Ultimately it is the role of great typography to convey the content in the most pleasing and effective way possible. Appropriate use of type and typography enhances our ability to comprehend the content. On the web, we've had many limitations to deal with—from a wide variety of screen sizes and limited display resolutions to a serious lack of high quality screen typefaces to help convey the messages of web authors. We have the ability, though, to bring these all together on the web, and providing rich typographic experiences to all readers.

Often the concern is raised when providing tools for rich typography is that they will be abused. And no doubt this will happen. Good typography has an invisibility that does not get in the way of reading. It will be important to help educate web designers on the importance of high quality typography and create tools to make this task easier. This reminds me of the quote from Beatrice Warde in her essay *The Crystal Goblet.*

> *Printing* [or web page design] *demands a humility of mind, for the lack of which many of the fine arts are even now floundering in self-conscious and maudlin experiments. There is nothing simple or dull in achieving the transparent page. Vulgar ostentation is twice as easy as discipline. When you realise that ugly typography never effaces itself, you will be*

able to capture beauty as the wise men capture happiness by aiming at something else.

Reading and the Brain

Even with a beautifully designed page it is useless if the reader of the page is unable to understand the contents of the material. Reading itself is not a simple process of staring at a page and ingesting the contents—instead it involves a complex dance between the eye and the brain, with the eye leading off with moves called saccades, fixations, and regressive saccades.

The human eye is attuned to reading text about 4 mm high at a distance of 50 cm. In order to perceive the spatial frequencies of text, we need to rely on the central portion of our vision called the fovea centralis.

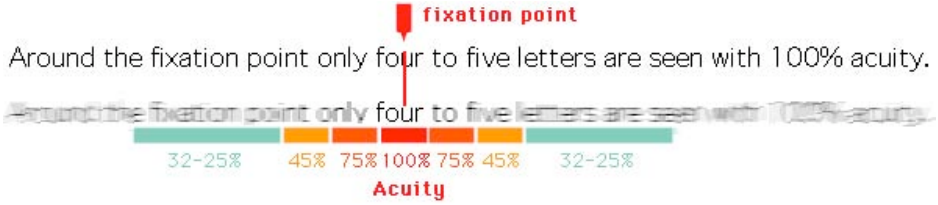

Figure 5: *This picture shows the acuity of foveal vision in reading (during one eye stop). The lower line of text simulates the acuity of vision with the relative acuity percentages. Image credit: Hans-Werner34 at en.wikipedia.*

Foveal vision takes up about 2° of arc, and at the reading distance of around 50 cm with characters about 4 mm high (about 12 point), it ends up that we can decode about 7–8 characters at a time. The fixation point is the center of our foveal vision and is usually to the left of center of English words and never on a word boundary. We fixate on this set of characters for around 200–250 ms and start the process of decoding the individual characters, building up the possible words made up of these characters. While we are fixating we also target the location of the next 7–8 characters, sometimes skipping over short words or words that are predictable given the context. Getting to this next location is called the saccade, a ballistic motion of the eye during which vision is suppressed. Although most saccades are forward through the text, skilled readers will still move their eyes backwards about 10–15% of the time, usually related to comprehension problems or blinking—these are called regressive saccades.

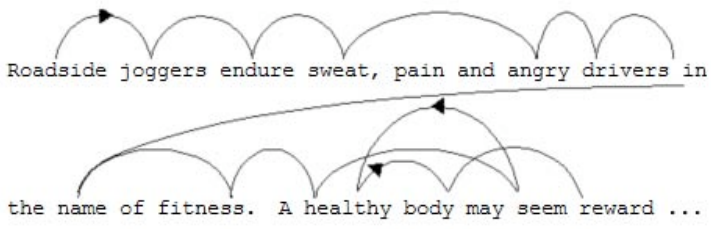

Figure 6: *Simulation of saccades through text.*

Several factors not related to the content of the text can impact the performance of reading, among the most common being excessive fixation time and the number of letters you can recognize during a fixation, but other factors, including reduced saccade distance and excessive regressive saccades, can lead to a diminished reading experience. One of the difficulties of reading on computer displays is that at the typical text sizes needed for reading at 50 cm, the size of the features in the glyph forms are less than or equal to the size of a whole pixel. Improvements in text rendering techniques or display pixel density can improve all of the eye-movement aspects of reading.

The human visual system plays a key role in the reading experience, and when reading longer pieces of text, there is strain put on our visual system. In general, the muscles of the eye are *very* robust. The saccadic eye movementss we just discussed don't end when we stop reading, they continue on in much of what we perceive. Even "reading" someone's face involves fixations and saccades. Certain ways of displaying text, though, can have a negative impact on the muscles of our visual system and lead to eye fatigue. Studies have shown that text that has too much gray on a white background—or blurry text—leads to a form of eye fatigue in the *orbicularis oculi* muscle, which controls our blinking and squinting. Likewise, text that is too small to read comfortably has the same impact on the *orbicularis oculi.*

There are other aspects of reading performance that are more brain centered than eye centered. One area of research that is quite relevant to web page design is tying the content of the text to the intended usage of the typeface. Most text, either consciously or subconsciously, is conveying one or more tones or emotions such as; serious or light-hearted, humorous or sad, informal or formal. This is not surprising, and authors, through their choice of words, use this technique to give deeper meaning to the text. In a similar way, although maybe less intuitive to people not familiar with typography, typefaces convey similar tones and emotions. When studies have been conducted asking users to categorize a typeface with a set of emotions, users are very consistent in the emotions they associate with a given typeface. Additional studies have shown that people read faster when the personality of the typeface matches the content of the text, and slower when the typeface's personality doesn't match the content of the text. The takeaway here is that a good selection of typefaces is important to properly present the content of the web page; acquiring appropriate fonts and using font embedding capabilities has a significant impact on how we read and comprehend text.

Additional aspects of typography, beyond typeface selection and text rendering, play a significant role in understanding textual content. The layout of the text, with good line lengths, good hyphenation, appropriate use of leading, and proper use of OpenType Layout features such as ligatures, kerning, *true* small caps, and old-style numerals are all critical characteristics for

good reading and an aesthetically beautiful experience. Surprisingly, research studies have had a difficult time attributing reading performance benefits to these characteristics. We have, though, conducted research that shows there is a critical benefit for well-designed and aesthetically pleasing text; the ability to perform better on cognitive tasks. This clearly impacts higher-order brain functions and plays a critical role in integrating the contents of what we read into our world view.

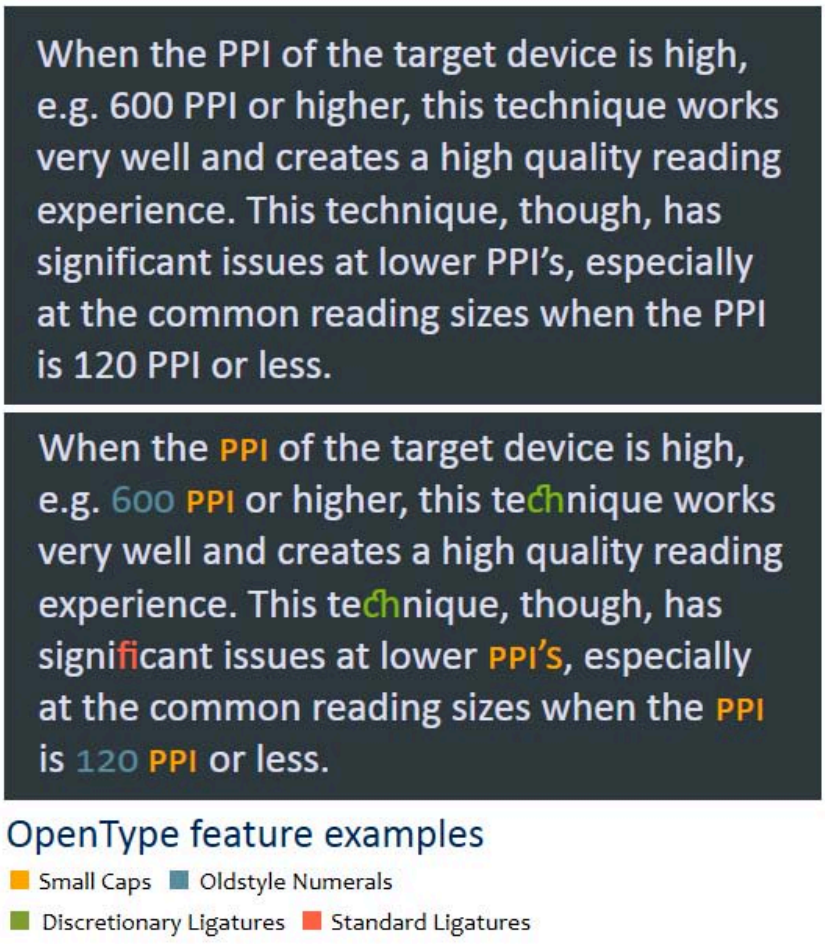

Figure 7: *OpenType Features.*

The Screen and the Technology of Type

The computer screen has been a challenging technology for reading text, with low pixel densities being one of the biggest obstacles for great reading. Other factors such as decreased edge contrast on characters and the ergonomic discomfort of reading on large fixed displays enhance the difficulty. Twenty-five years ago we all expected that hardware technology would solve this problem in much the same way that CPUs, memory, and computer hard drives have made exponential improvements over that same time period—but progress has been very slow in making the improvements we've hoped for in text on computer screens.

The common computer screens of twenty-five years ago in the PC space were primarily CRTs and had pixel densities of around 70 Pixels per Inch (PPI). For many of these displays those pixel densities were only horizontal, the vertical arrangement of pixels might be half of that density—giving us rectangular pixels. Ten years later, with LCD screens starting to take off, we were up to around 85 PPI for most laptop displays, an increase of 15 PPI. Fifteen years after that, most desktop displays are under 110 PPI, and increase of only 40 PPI over 25 years, just over a 60% improvement—where we would have expected around 8000 times improvement in computer processing capability over that same time period. Granted, smaller displays for mobile phones and even laptops have made more significant improvements, but if we assume 300 PPI for the best screens today that is only around a 325% improvement.

Without the strong support of hardware improvements, we've had to rely on several technological solutions that can improve the appearance of text on lower-resolution computer displays. The first step to improving text quality comes from a capability in the TrueType technology developed by Apple, which Microsoft first shipped with Windows 3.1 in 1992.

The TrueType technology works by defining a high resolution grid based on the em of the typeface. This grid is called the "Units per Em" and in a TrueType font it is typically a power of two like 2048. Other values are possible, each providing a tradeoff between the space taken to represent the data and the level of detail that can be shown. Each character is digitized by tracing the outer edges of the characters in the typeface and connecting these using graphics primitives of points, lines and curves. This creates a series of outlines for each character that describes the area encompassed by the "ink" used to represent the font. Sometimes the metal sort is digitized, sometimes the optical master from an earlier technology called phototypesetting is used, other times the ink on the page is digitized, and finally some fonts are designed directly in font creation tools where the designer makes the necessary lines and curves to describe the typeface.

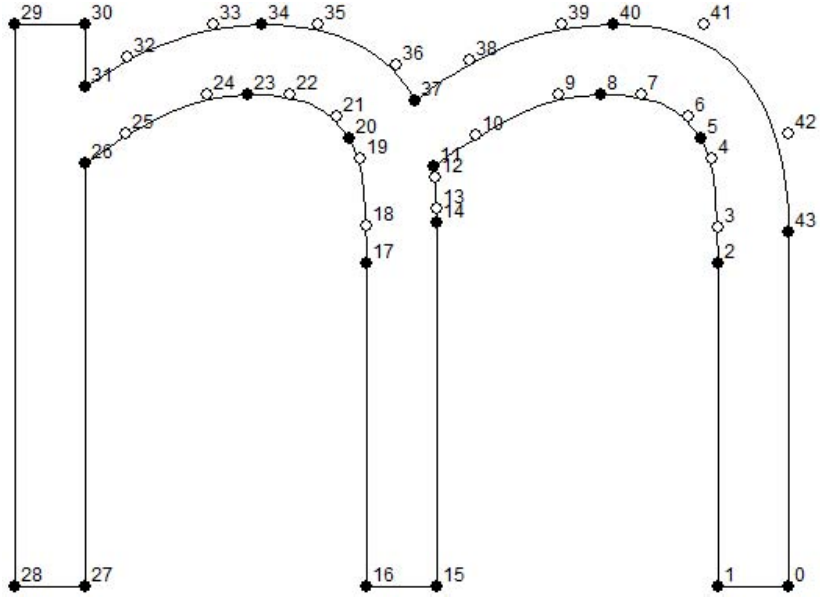

Figure 8: *Outline of lowercase 'm' with points, lines and curves.*

In order to display the "outline" of the character on the screen or printer, the points in the outline have to be scaled to the appropriate size. This requires knowledge of the Units per Em that the font was designed at, the point size to display, and the PPI of the target device. From these values a scale factor can be determined and applied to every point. Once the outline is sized for the device, the TrueType technology uses a fill operation to create the font—if the center of a pixel is inside the character outline, the pixel is set to the text color, otherwise it is left as the background color.

When the PPI of the target device is high, e.g. 600 PPI or higher, this technique works very well and creates a high quality reading experience. This technique, though, has significant issues at lower PPIs, especially at the common reading sizes when the PPI is 120 PPI or less. Specifically, when the feature size that we are trying to accurately represent is less than or equal to the pixel size, rounding problems have a severe detrimental impact on the text quality.

To address this problem, the TrueType technology includes an integrated programming language specifically designed to assist skilled designers or automated tools in improving the quality of text at smaller sizes and/or smaller PPIs. The process of improving font scaling is usually called "hinting" and it is accomplished by "instructing" the font glyph shapes using the built in interpretative programming language. This language is implemented as a stack-based byte-code interpreter that includes primitives and data structures for applying rules and constraints to the features of the glyphs in order to achieve high quality, readable text. Broad classes of instructions for TrueType include stack management, subroutines & conditional branching, basic math & Booleans, control of state variables, outline manipulation, and data structure management. The outline manipulation and state variables are the components specifically optimized for working with TrueType outlines.

Because the language is very rich, it does not force the "hinter" into only one way to solve a particular problem. There are several techniques in hinting a TrueType font for use on screens that are quite common. One technique is called grid-fitting, which aligns the outline edges of glyphs to pixel or subpixel boundary. By carefully grid-fitting the outline and making sure the outline snaps to multiples of the grid, rounding problems are minimized. Another technique that works with grid-fitting is using the data structures provided with TrueType to maintain grid level consistency, both within the glyph and across the glyphs in the font. For example, if the vertical stem of a lowercase 'm' is two pixels wide, the rest of the stems should (usually) be two pixels wide. Other similar glyphs in the font should have stems that are two pixels wide. Although this seems obvious, it does not naturally happen when scaling fonts and it needs to be carefully designed in the hinting. Also related to this is stem regularization. The outline may have subtle or even significant differences between features that may look like they are the same. Regularization is a technique using data structures that force the values to be the same for lower sizes, and then as the size increases (meaning there are more pixels to describe the glyph) differentiation between stems is allowed. Finally, an important aspect of hinting, especially at small sizes with lower resolution, is "delta hinting." Delta hinting is making small adjustments to the outline that fine-tunes the shape of the outline at a particular size or range of sizes. Sometimes this is used to fix an errant pixel turned on by rounding errors, other times it is used to adjust global features in the typeface like x-height or stem width. This list of hinting techniques is not meant to be fully inclusive; there are many more concepts than we have the time for in this simple overview.

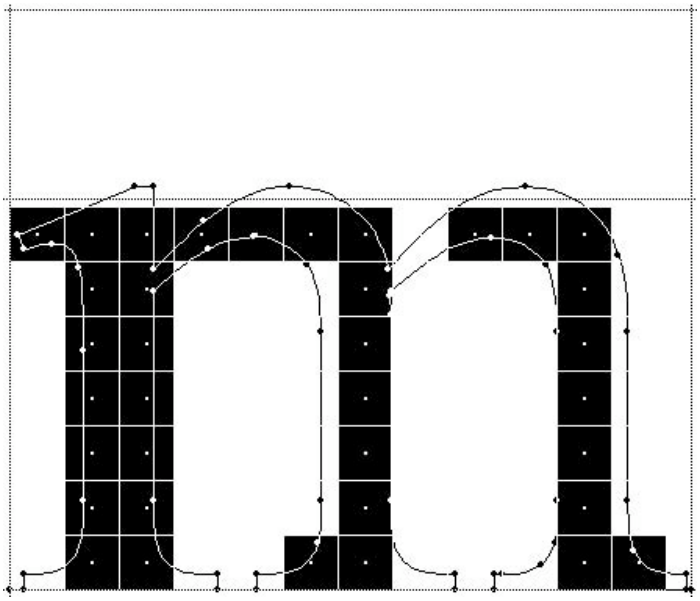

Figure 9: *Unhinted lowercase 'm' with bi-level rendering.*

When Microsoft integrates "hints" into the fonts we ship, skilled typographers are responsible for writing the code for the hints. Like the punchcutter of old, meticulously carving the letterform into steel, making adjustments and tradeoffs for the technology of the era and the necessities of human reading, the font "hinter" has similar concerns. Basic tradeoffs like determining if symmetry is more important to maintain than glyph proportion, deciding if overshoots and undershoots are appropriate for a given size, fine tuning pixel shapes, and making appropriate adjustments to the letterform for the size at which the glyph will be displayed—similar to optical scaling—are all the bailiwick of the font "hinter."

Figure 10: *Japanese Font Meiryo, the left column is unhinted and the right is hinted.*

Some people have argued against font hinting because they claim it adds a non-linear component to the "true design" of the typeface. Historically there has never been a true master design, instead each font is tuned for the hardware and optics of viewing by the designer, and there have always been nonlinearities in the designs to account for nonlinearities in our visual system or the technologies we develop. The optical scaling concept is inherently non-linear and designed for the physiology of humans. Other "features" of typeface design like "ink traps" are nonlinearities designed to handle problems with how ink and metal interact with paper. When carefully crafted by a skilled typographer, these hints and nonlinearities add to the beauty and functionality of a typeface.

OpenType Layout Features are another important aspect of beauty and functionality. The OpenType Layout capability was designed by Microsoft in 1994 as a way of supporting advanced typographic layout while remaining true to the principles of Unicode. The goal of Unicode was not to provide presentation forms (ligatures, small caps, old-style numerals, &c.) for every glyph, instead there was to be another way of encoding presentation forms—for Microsoft that was the development of OpenType Layout, originally called TrueType Open. In 1996, Adobe joined Microsoft with the joint announcement of OpenType, which included the TrueType Open technology. In conjunction with Unicode, OpenType Layout allows the selection of both rich international and Latin based typography. Some script systems like Arabic and Indic require such a system in order to display the minimal necessities of the language system, and when taken to the full extent, it can provide beautifully displayed type. For Latin based languages, many of the rich typographic traditions that come from hot metal typography require a system like OpenType Layout.

Figure 11: *Font: Arabic Typesetting, example shows alternate shapes based on context: The initial letter Jeem takes different shape depending on what letter follows it.*

OpenType Layout works by listing features in the font that are supported by different script and language systems. A feature can be on by default and apply to the whole font or a feature can be applied to a range of Unicode characters to possibly change the default glyph for that character or to change the position of that character. Changing glyphs can happen on a one-to-one, a one to many, a many to one basis, or based on the context of a glyph and its neighbors. Features like kerning are implemented by changing the position of multiple glyphs with respect to each other.

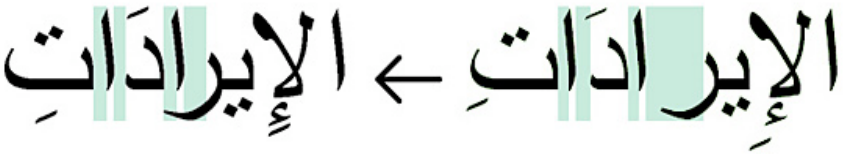

Figure 12: *Font: Simplified Arabic, OpenType kerning.*

Once we have great type and layout, we need to render it properly on the screen to get the best possible reading experience. Other posts have discussed ClearType technology in Windows 7 and some of the new ClearType features available in DirectWrite. There are clearly both personal preferences to the type of rendering used and rendering that works well on some display types rather than others, in the same way that some people prefer to read hardback books and others paperbacks. In the big picture, though, when close attention is paid to the quality of every detail, to every pixel of text, the results pay off with the best possible reading experience.

All of these typographic features that have been discussed here must work in harmony to make a great reading experience. Reading is often something we take for granted—most of us learn how to read as a child, and soon forget the difficulties of that task. This is good, as we don't want to focus on the mechanics of reading when we should be focusing on the content of the text. If we as font developers, browser developers and web page designers do our job properly, the content can be king, and we can focus our energies on comprehension rather than decoding.

Greg Hitchcock & al.

Edit: Fixed a few typographic errors in the layout