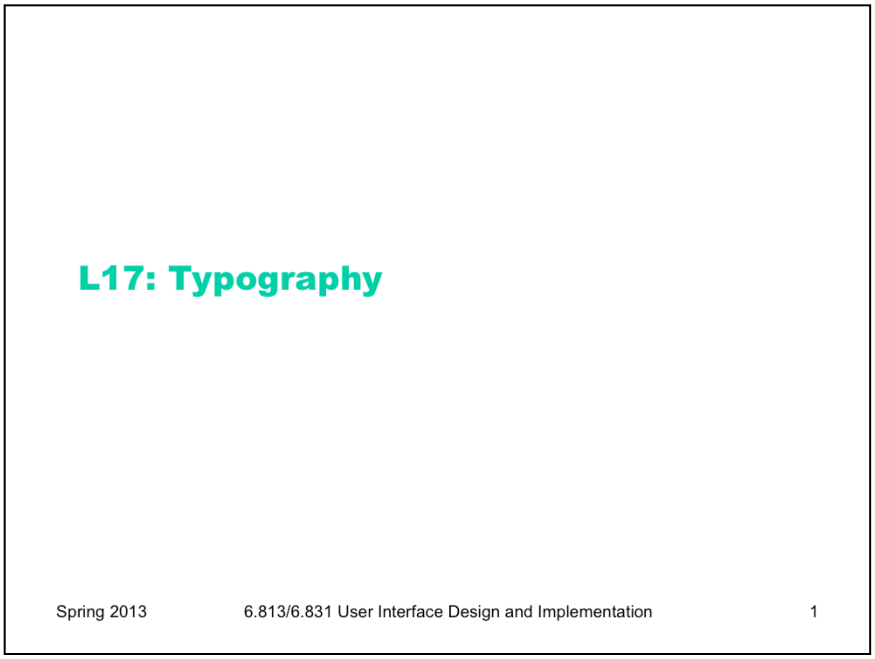
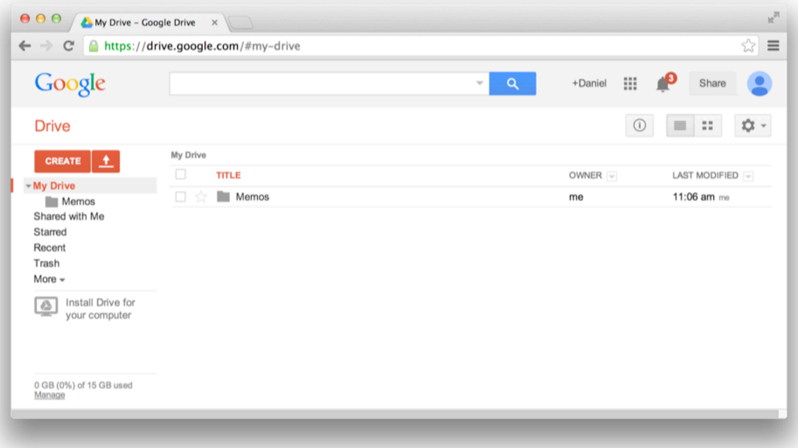


L17: Typography

UI Hall of Fame or Shame?

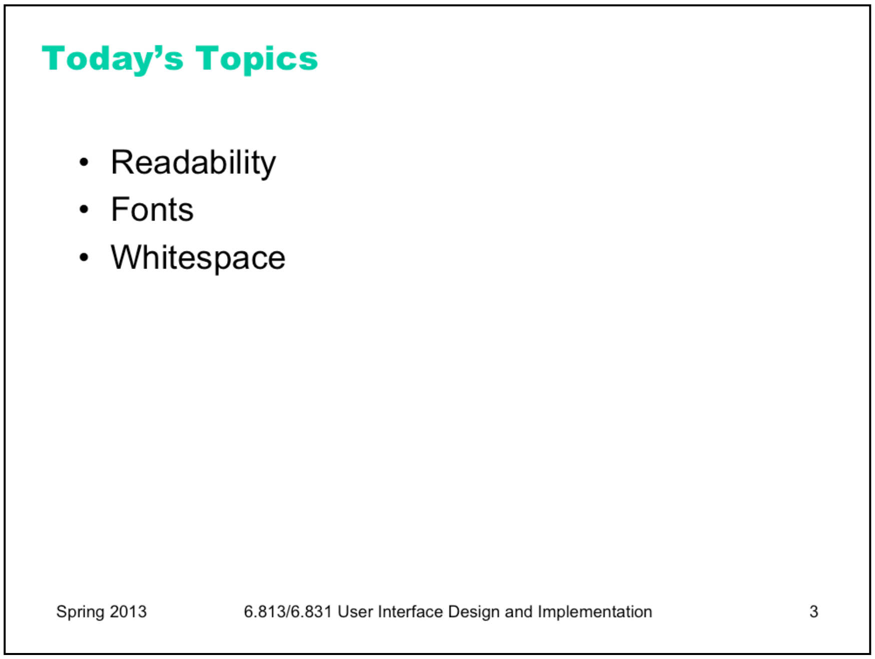


My Drive

TITLE	OWNER	LAST MODIFIED
Memos	me	11:06 am me

Consider the web interface for Google Drive. Analyze its usability with respect to:

- Graphical layout: the squint test, visual variables, spacing, etc.
- Adherence to user interface conventions
- Simplicity (what functionality is offered, how simply is it expressed?)
- Learnability and efficiency; information scent

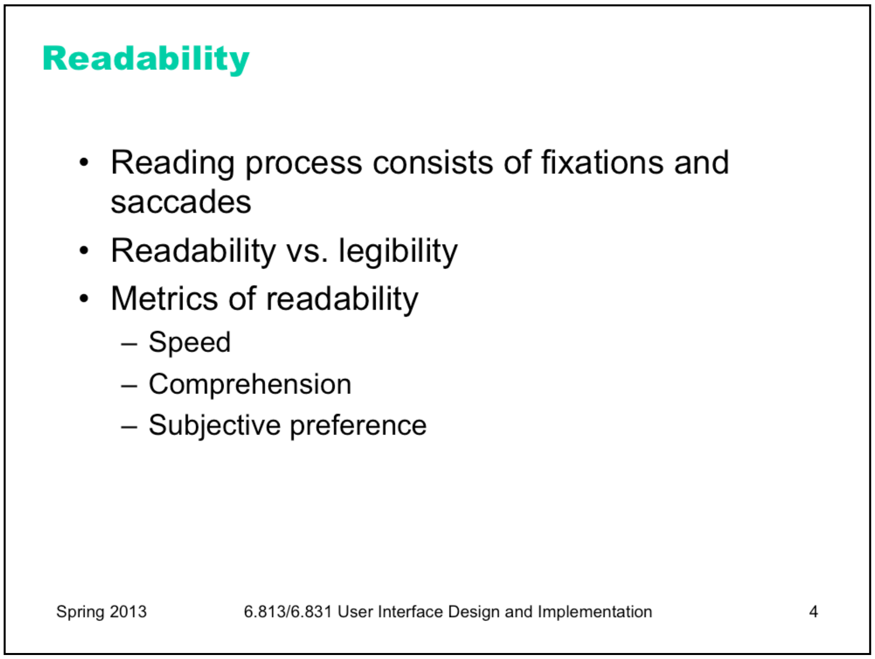


Today's Topics

- Readability
- Fonts
- Whitespace

Typography is the art and science of displaying text (or "setting type" as print designers call it). The key decisions of typography concern font (the shapes of letters and other characters) and spacing (the white space around letters, words, lines, and paragraphs). Both are important to successful text display; without adequate spacing, the shape of the text is much harder for the eye to discriminate.

For typography, an outstanding book is Robert Bringhurst, *The Elements of Typographic Style*, Hartley & Marks, 2002. Also useful is "Principles of Typography for User Interface Design" by Paul Kahn and Krzysztof Lenk, *interactions*, which you can find online.



Readability

- Reading process consists of fixations and saccades
- Readability vs. legibility
- Metrics of readability
  - Speed
  - Comprehension
  - Subjective preference

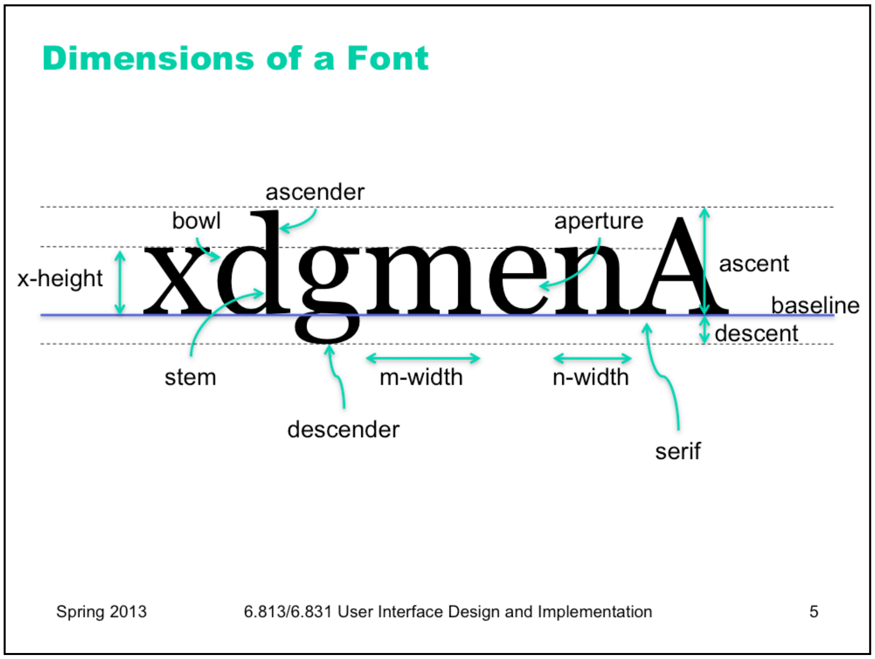
Spring 2013 6.813/6.831 User Interface Design and Implementation 4

In keeping with our brief tours of cognitive science for each design topic, let’s say a bit about what’s known about reading. Note that these high-level comments are applicable to most written languages, not just English. Most of the rest of the lecture will be specific to languages that use the Latin alphabet and its corresponding fonts, however.

First, reading is not a smoothly linear process, even though it may feel that way to introspection. The eye does not move steadily along a line of text; it proceeds in fits and starts, called **fixations** (stopping and focusing on one place to recognize a word or several words at a time) and **saccades** (an abrupt jump to the next fixation point). At the end of a line, the eye must saccade back to the beginning of the next line.

Researchers studying reading and typography often make a distinction between *legibility*, which is low-level and concerns how easy it is to recognize and distinguish individual letter shapes, and *readability*, which concerns the effectiveness of the whole reading process. A single fixation can consume whole words or multiple words, so fluent readers recognize the shape of an entire word, not necessarily its individual letters. Readability can be measured by several metrics, including speed, comprehension, error rate, and subjective preference. Readability is essentially the usability of a display of text.

A good discussion of the reading process can be found at http://www.microsoft.com/typography/ctfonts/wordrecognition.aspx .



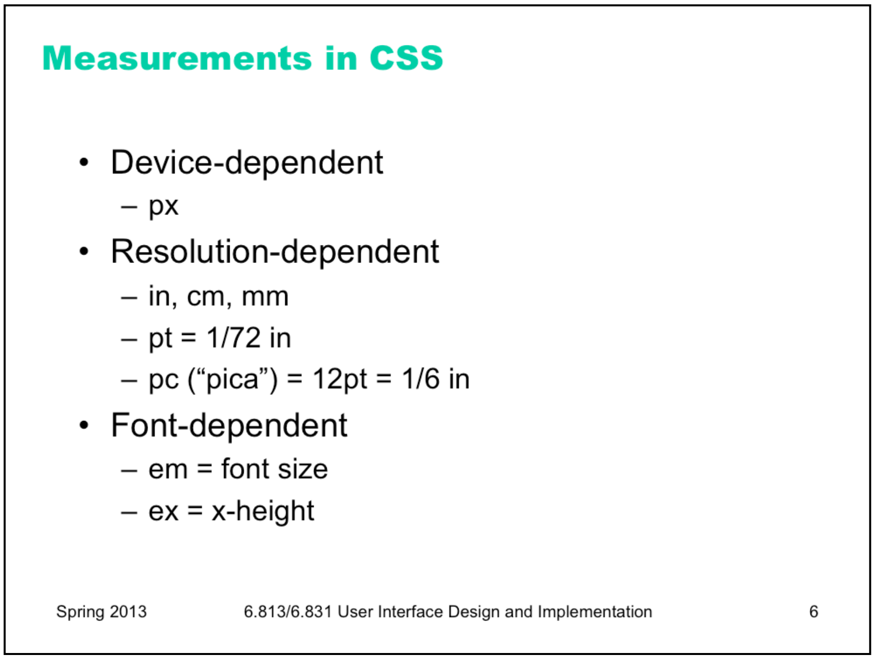
Dimensions of a Font

Now a few definitions. Characters in the Latin alphabet all sit on a common baseline. Some characters have descenders dipping below the baseline; others have ascenders rising above the typical height of a lowercase character (the **x-height**). Capital letters also ascend above the x-height. The typical height of ascenders above the baseline is called the **ascent**, and the typical height of descenders below it is called the **descent** of the font.

The **font size** is typically ascent + descent (but not always, alas, so two fonts with the same numerical size but using different typefaces may not line up in height!). Font size is denoted in points; a point is 1/72 inch, so a 12-point font occupies 1/6 of an inch vertically.

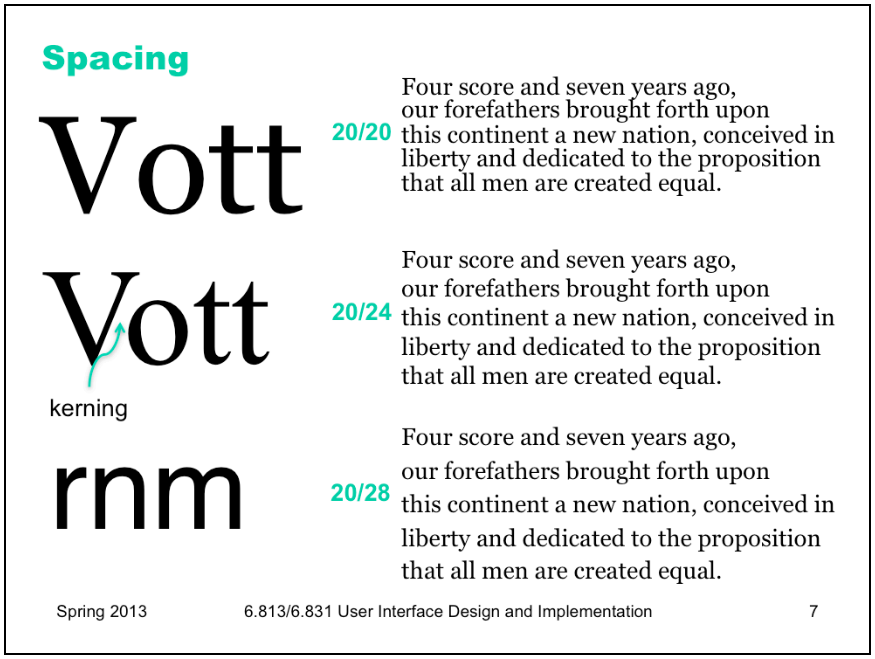
X-height, m-width, and n-width are useful font-dependent length metrics. You can find them used in CSS, for example. They allow specifying lengths in a way that will automatically adapt when the font is changed.

More font terminology can be found at http://www.davidairey.com/images/design/letterform.gif.



Measurements in CSS

- Device-dependent
  - px
- Resolution-dependent
  - in, cm, mm
  - pt = 1/72 in
  - pc (“pica”) = 12pt = 1/6 in
- Font-dependent
  - em = font size
  - ex = x-height



Spacing

Vott

20/20

Four score and seven years ago,
our forefathers brought forth upon
this continent a new nation, conceived in
liberty and dedicated to the proposition
that all men are created equal.

Vott

kerning

20/24

Four score and seven years ago,
our forefathers brought forth upon
this continent a new nation, conceived in
liberty and dedicated to the proposition
that all men are created equal.

rnm

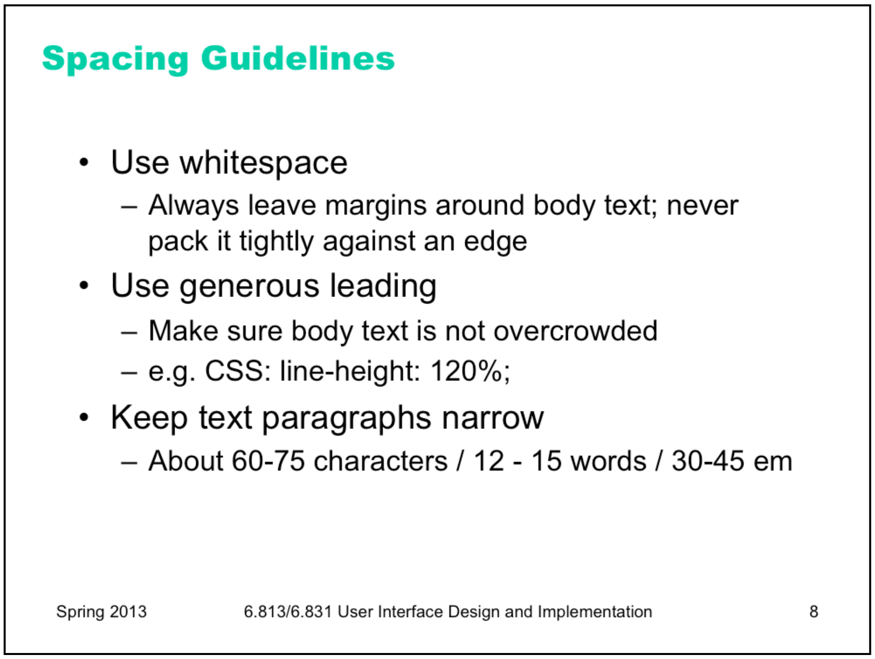
20/28

Four score and seven years ago,
our forefathers brought forth upon
this continent a new nation, conceived in
liberty and dedicated to the proposition
that all men are created equal.

Several kinds of spacing matter in typography. At the lowest level is character spacing. The gaps between characters must be managed to prevent uneven gaps or characters appearing to run into each other. **Kerning** is the process of adjusting character spacing for particular pairs of characters; sometimes it needs to be narrowed (as V and o shown here), and sometimes widened (e.g. to keep “rn” from looking too much like “m”). A good font has kerning rules built into it, and a good GUI toolkit uses them automatically, so this is rarely something GUI programmers need to worry about, except when choosing a font. Note that the top Vott is displayed in Georgia, which at least on my system appears *not* to have any kerning for V and o. The second Vott is displayed in Times New Roman.

Spacing between words and lines matters too. Words must have adequate space around them to allow the word shape to be easily recognized, but too much space interferes with the regular rhythm of reading. Similarly, adequate line spacing is necessary to make word shapes recognizable in a vertical dimension, but too much line spacing makes it harder for the eye to track back to the start of the next line. Line spacing is also called **leading**; technically speaking, the leading is the distance between baselines of adjacent lines. Both font size and leading are important. Print designers say, for example, “12 point type on 14 points of leading” (or “12/14”) to indicate that the font size is 12 points (typically ascent + descent) with 2 points of space between the descent of one line and the ascent of the next. Using the same line height as font size (like 20/20) is almost always a mistake; characters from adjacent lines touch each other, and the paragraph is much too crowded. Note that leading also strongly affects the overall **value** of the body text (which type designers somewhat confusingly call the “color” of the text; historically print is mainly black-and-white, of course, but it’s confusing when talking about modern printing and modern computer displays). Tight spacing looks much darker than loose spacing.

In most toolkits, choosing a font size implicitly chooses a leading, but the default leading may not always be the best choice. Look at it and adjust if necessary. CSS makes this possible using the line-height property.



Spacing Guidelines

- Use whitespace
  - Always leave margins around body text; never pack it tightly against an edge
- Use generous leading
  - Make sure body text is not overcrowded
  - e.g. CSS: line-height: 120%;
- Keep text paragraphs narrow
  - About 60-75 characters / 12 - 15 words / 30-45 em

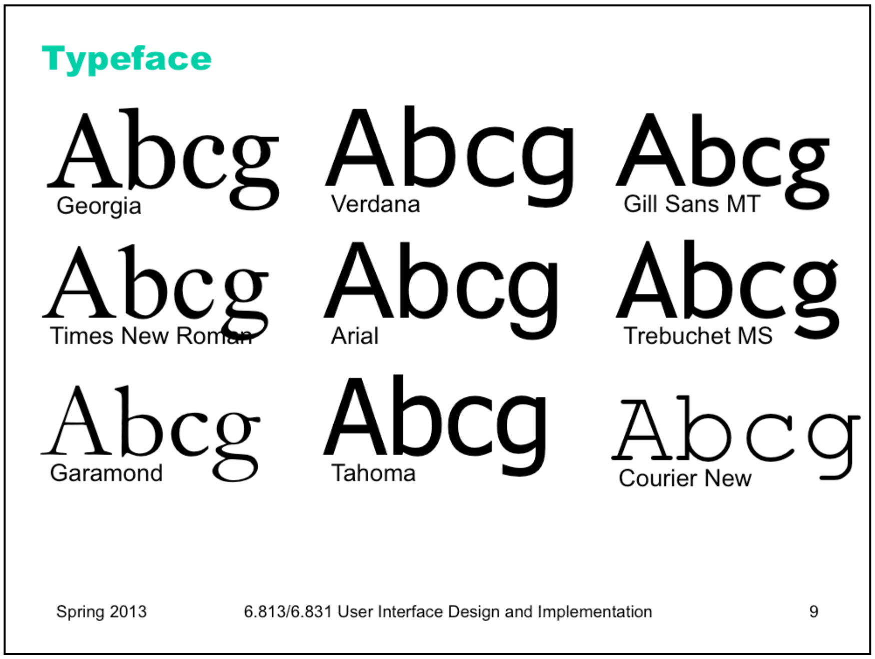
Spring 2013 6.813/6.831 User Interface Design and Implementation 8

Here is some advice for choosing spacing for body text (text set in paragraphs). Always leave margins around body text; never pack it tightly against a rule, an edge, or a window boundary. The margin helps frame the text and also helps the reader find the ends of the lines, which is essential for the saccade back to the beginning of the next line.

Use generous leading, but not too generous. 120% of the font size is a good rule of thumb; this would correspond to the 20/24 leading shown on the previous slide.

Line length (or equivalently, paragraph width) is another important consideration. Hundreds of years of experience from print typography suggest that fairly short lines (3-4 inches) are both faster to read and preferred by users. Unfortunately the studies of onscreen reading yield mixed answers; apparently, on screen, longer lines (about twice the ideal length for print) help users read faster, but users still prefer the short lines (perhaps because their consistent feel with print). These same studies show that onscreen reading is slower than print reading, however, and recent studies have shown less and less effect of line length on speed, possibly because display and font technology is improving rapidly. So it’s possible that making the lines longer merely offsets the poorer resolution and legibility of computer displays relative to print, and as the displays approach print in quality, this distinction will go away. (Bailey, “Optimal Line Length: Research Supporting How Line Length Affects Usability”, December 2002, http://webusability.com/article_line_length_12_2002.htm)

Translating the 3-4 inch rule into characters or m-widths for typical 10-point to 12-point type gives 60-75 characters or 30-45 em widths.



Typeface

Abcg
Georgia

Abcg
Verdana

Abcg
Gill Sans MT

Abcg
Times New Roman

Abcg
Arial

Abcg
Trebuchet MS

Abcg
Garamond

Abcg
Tahoma

Abcg
Courier New

Spring 2013 6.813/6.831 User Interface Design and Implementation 9

After spacing, a key decision is what **typeface** to use. Typeface refers to a family of fonts sharing the same name, like "Arial" or "Georgia." A **font** is a choice of typeface *and* size *and* style, like roman, italic, oblique, boldface, etc.

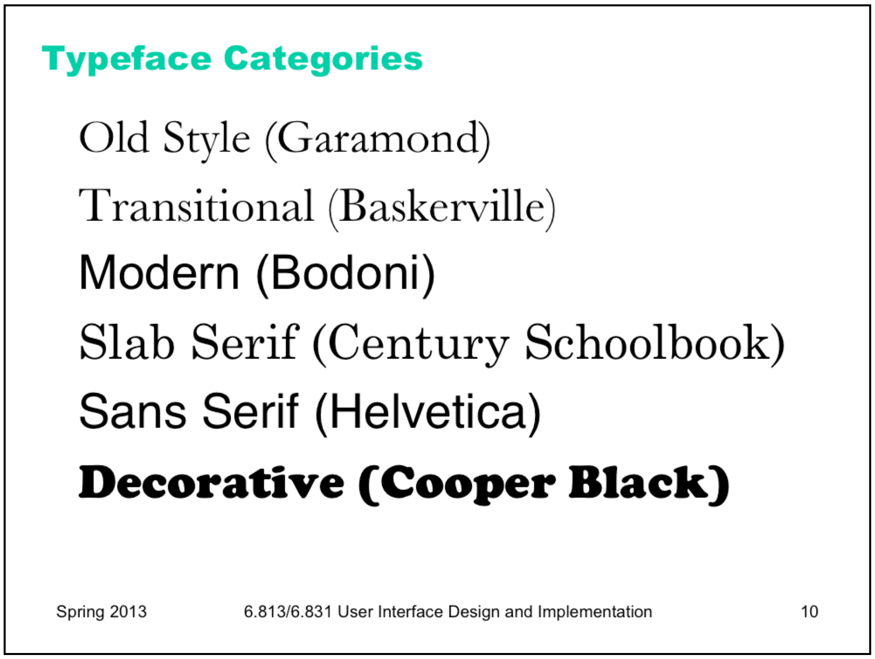
Typefaces can be classified in many ways, and can convey strong associations that influence how the user perceives the text. One important classification you should know is between **serif** fonts, like Georgia and Times, and **sans serif** fonts, like Verdana and Arial. Historically, in print typography, serif fonts have been used for **body text** (text set in paragraphs), because they offer stronger cues to word shape that allow measurably faster reading. Sans serif fonts were generally used for **display text** (text that stands alone, like headings and labels), for which reading speed is less important and contrast from body text is useful.

In the early days of computer typography, sans serif fonts were often preferred for all uses, because their simpler letter shapes were far more legible on low-resolution displays. As displays become higher resolution, however, serif text may once again assert itself; even now, there is evidence that serif fonts are faster to read on screen (Bernard et al, "A Comparison of Popular Online Fonts: Which is Best and When?", Usability News, 2001, http://www.surl.org/usabilitynews/32/font.asp).

Another key distinction is between proportional fonts (in which each character has a different width) and monospace fonts (in which all characters have the same width, like Courier New shown here). Monospace fonts waste screen space and generally look worse than well-designed proportional fonts, so avoid them unless you have a good reason.

Your choice of font family conveys a tone. Some designers think Times and Arial look cheap because they're so widely used; using Georgia or Garamond will give your UI a more "designed" look (i.e., you actually made an informed choice, rather than choosing a default). Another consideration is whether the font was designed for screen use. On this slide, Verdana, Georgia, Tahoma, and Trebuchet were commissioned by Microsoft primarily for onscreen use. Most of the other fonts shown here are digital updates of old fonts originally designed for print. Note some distinct features of Georgia/Verdana/etc. relative to the others – larger x-height (as a fraction of total ascent) and generous bowls and apertures, intended to make the fonts more legible at small sizes on lower-resolution displays.

All the fonts shown here are appropriate and useful for body text (though they aren't the only possibilities, of course). Fonts for body text are designed to have evenly-distributed "color" (the value of the text in a squint test) so that they look good in bulk. You will find many other fonts installed on your computer, some of them very wacky. Some of these may be useful for occasional display text, but certainly not body text. Unfortunately today's word processors give users many fonts but very little help selecting the right one for the right use. Instead we get a long undifferentiated list of installed fonts that hides gems like Garamond and Georgia in a sea of Comic Sans, Goudy Stout, and Old English Text MT -- some of which probably should not be used in any imaginable circumstance.



Typeface Categories

Old Style (Garamond)

Transitional (Baskerville)

Modern (Bodoni)

Slab Serif (Century Schoolbook)

Sans Serif (Helvetica)

Decorative (Cooper Black)

Spring 2013 6.813/6.831 User Interface Design and Implementation 10

To help you sort through the profusion of typefaces out there, here is a commonly-used classification, based on visual characteristics of the faces. (See http://graphicdesign.spokanefalls.edu/tutorials/process/type_basics/type_families.htm for more discussion of this classification.)

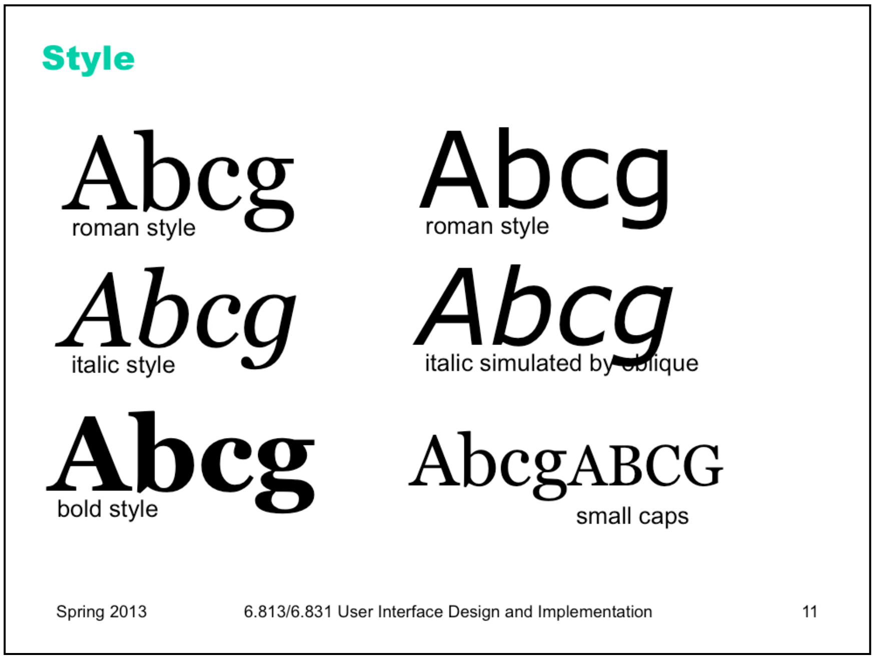
Old Style fonts mimic calligraphy – drawing letters with a pen that has a slanted tip -- very closely. Calligraphy was the writing technology that preceded the printing press, so Old Style fonts were maintaining consistency with it. If you look carefully at the a’s in Garamond, you’ll see that the diagonal strokes in one direction are very thin, and diagonal strokes in the other direction are very fat. This is characteristic of a pen held at an angle by a calligrapher’s hand. Other characteristic of Old Style fonts are low contrast (in thickness) between the thick and thin strokes, and small x-heights (relative to the full ascent).

Transitional fonts moved toward greater contrast between thick and thin strokes. The Modern class of fonts is the full evolution of this move – not only are the thick parts of the letters much thicker than the thin parts, but the serifs are virtually hairlines, with sharp corners between the serif and the rest of the letter rather than a graceful curve. Sharp contrasts are typical of Modern fonts.

The Slab Serif family have fat, rather than hairline, serifs, and x-heights tend to be large.

The Sans Serif family is the most recent and modern-looking. Not only have Sans Serif fonts done away with serifs, but they also do away with the variation within a stroke – no more calligraphic contrasts of thick and thin. Instead the letters are geometric, smooth, shaped.

The first five categories are suitable body fonts, readable in long texts. The last category, Decorative, should only be used for headings, titles, or logos, where it’s important to be eye-catching but ease of reading is not so important.



Style

Abcg
roman style

Abcg
roman style

Abcg
italic style

Abcg
italic simulated by oblique

Abcg
bold style

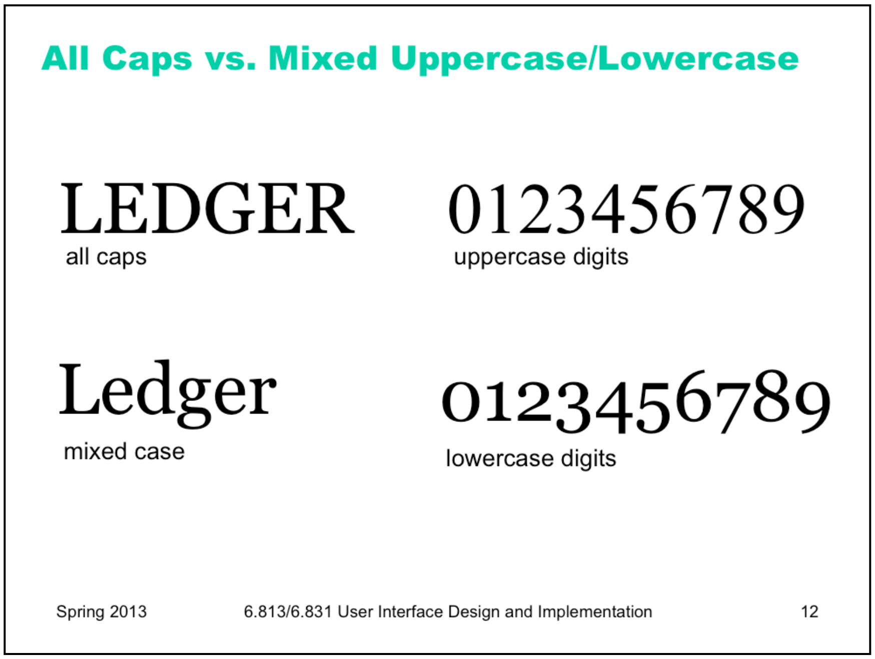
AbcgABCG
small caps

Spring 2013 6.813/6.831 User Interface Design and Implementation 11

We said that a font consists of typeface, size, and **style**. Here are a few common styles you can use to establish contrast.

Italic and boldface create contrast in orientation and value, respectively, without substantially changing the shape of the typeface. Some typefaces lack a true italic, and instead substitute an *oblique* font which is just a slanted version of the normal roman style (sometimes even automatically-transformed from the roman font, not hand-designed). Georgia, shown on the left, has a true italic – notice that the b loses its lower serif, and the g actually changes shape. Sans serif fonts have an oblique rather than italic; look at Arial for an example.

Small caps is another useful style. Small caps are uppercase letters that are as tall as the x-height, rather than the full ascent of the font. Like italic, small caps are sometimes a hand-designed font included with the typeface family (often slightly wider and lighter than capital letters), and sometimes simply automatically generated by shrinking the font.



All Caps vs. Mixed Uppercase/Lowercase

LEDGER
all caps

0123456789
uppercase digits

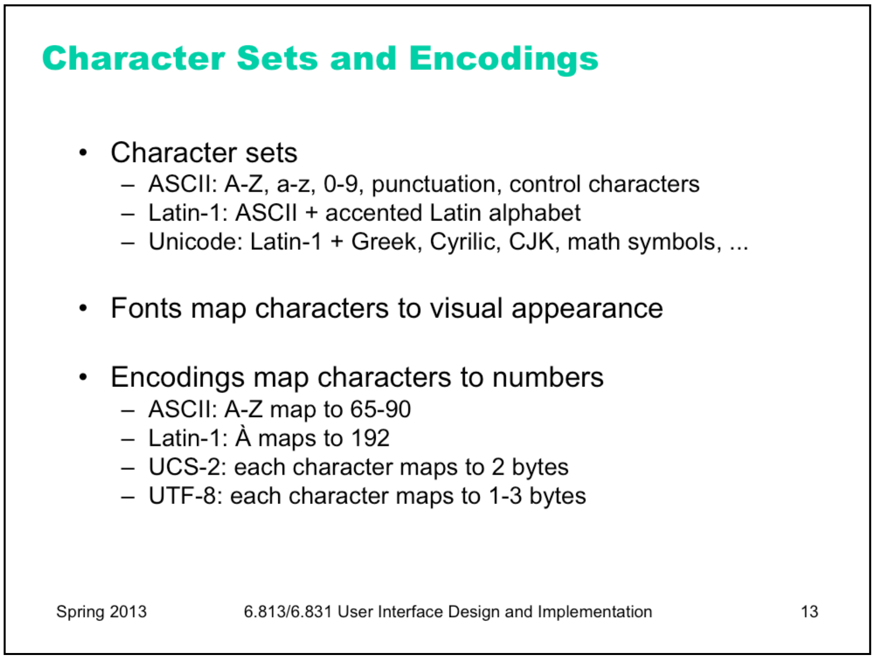
Ledger
mixed case

0123456789
lowercase digits

Spring 2013 6.813/6.831 User Interface Design and Implementation 12

While we're talking about capital letters, it's worth discussing when it's appropriate to set text in all capitals. All-caps has very little variation in word shape, because all the letters have the same top (the full ascent of the font) and the same bottom (the baseline, with almost no descenders). For this reason, it's both slow and unsatisfying to read body text set in all-caps. All-caps should be reserved only for display text (headings, labels, etc), and even then used very sparingly.

Older print typography actually had lowercase *digits*, not just letters. Notice that the lowercase digits predominantly follow the x-height, with ascenders and descenders for certain digits, just like lowercase letters. You may have seen typesetting like this in older books, published in the first half of the 20th century or earlier. Lowercase digits fell out of fashion in print in favor of more uniform uppercase digits, which may be monospaced horizontally as well, so that columns of digits line up easily; all the digits in Times New Roman have equal width, for example, even though the rest of the typeface is proportional. But lowercase digits are worth some consideration. They are more readable in body text than uppercase digits, for the same reasons as lowercase letters, and they convey a feel that is simultaneously retro and "designed." Unfortunately the character sets we use (ASCII and Unicode) make no distinction between lowercase 5 and uppercase 5 (unlike a and A), so when you choose a typeface, you either get *only* lowercase digits (like Georgia on the bottom) or *only* uppercase digits (like Times on top).



Character Sets and Encodings

- Character sets
  - ASCII: A-Z, a-z, 0-9, punctuation, control characters
  - Latin-1: ASCII + accented Latin alphabet
  - Unicode: Latin-1 + Greek, Cyrilic, CJK, math symbols, ...
- Fonts map characters to visual appearance
- Encodings map characters to numbers
  - ASCII: A-Z map to 65-90
  - Latin-1: À maps to 192
  - UCS-2: each character maps to 2 bytes
  - UTF-8: each character maps to 1-3 bytes

Spring 2013 6.813/6.831 User Interface Design and Implementation 13

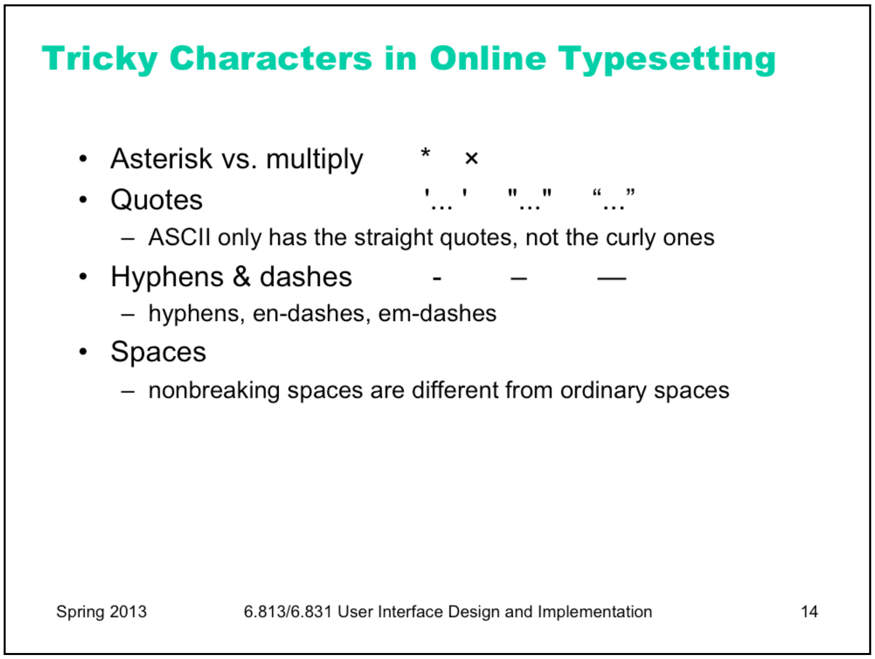
Note the difference between **character sets** and **fonts**. The Unicode character 'A' doesn't actually say how to *draw* A on the screen; a font does that. So even though you can represent many different alphabets in a single Unicode string, the *font* you're drawing the string with doesn't necessarily know how to draw all those characters. The appearance of a particular character in a font is called a glyph. Many fonts only have glyphs for a small subset of Unicode. For characters that aren't supported by the font, you'll see an error glyph, which might look like a little empty square or a question mark.

Note also the difference between character sets and **encodings**. A character set is an abstract set of possible characters. ASCII had 128 characters; Latin-1 had 256 characters, and Unicode has thousands of characters. An encoding maps each character in a character set to a number (or a small sequence of numbers). Internally, Java uses a 16-bit encoding for Unicode characters, representing each character by two bytes in memory. But the most common encoding for Unicode text in files and web pages is UTF-8, which does *not* use two bytes per character. Instead, UTF-8 uses 1, 2, or 3 bytes to represent each character. Single bytes are used to represent all the 7-bit ASCII characters, so UTF-8 can be thought of as an extension to ASCII.

There are other encodings as well. ASCII maps its characters to the numbers 0-127, which are stored in bytes. Latin-1 (also called ISO 8859-1 after its ISO standard) maps its characters to 0-255 (compatibly).

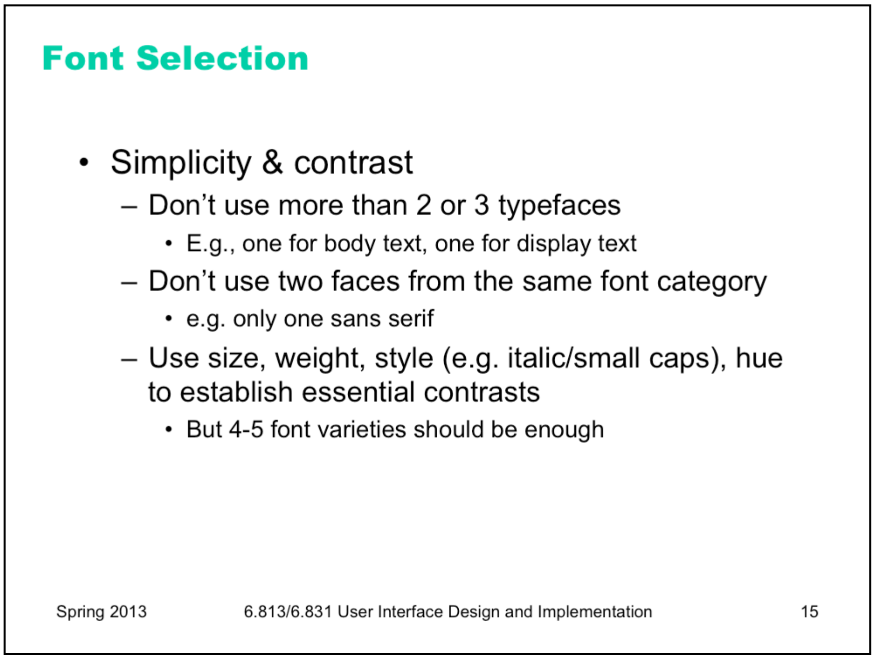
In general, **you cannot correctly interpret a text file or web page without knowing its encoding**. If your code ignores encodings and assumes everything is ASCII, you will find that it mostly works as long as you only use English, because encodings generally strive for backwards compatibility with ASCII. In other words, an English text would probably look identical in ASCII, UTF-8, and Latin-1. But it may break horribly on text in other languages. Even English text has problems when the author uses punctuation that isn't available in the basic ASCII character set. For example, ASCII only had one kind of double-quote mark (a vertical one), but many word processors now use left and right double quotes that are available in Unicode and other character sets, which often turn into garbage characters when you load the text into encoding-ignorant programs.

For more about encodings, Joel Spolsky has a good article (http://www.joelonsoftware.com/articles/Unicode.html).



Tricky Characters in Online Typesetting

- Asterisk vs. multiply * ×
- Quotes '... ' "..." “...”
  - ASCII only has the straight quotes, not the curly ones
- Hyphens & dashes - – —
  - hyphens, en-dashes, em-dashes
- Spaces
  - nonbreaking spaces are different from ordinary spaces



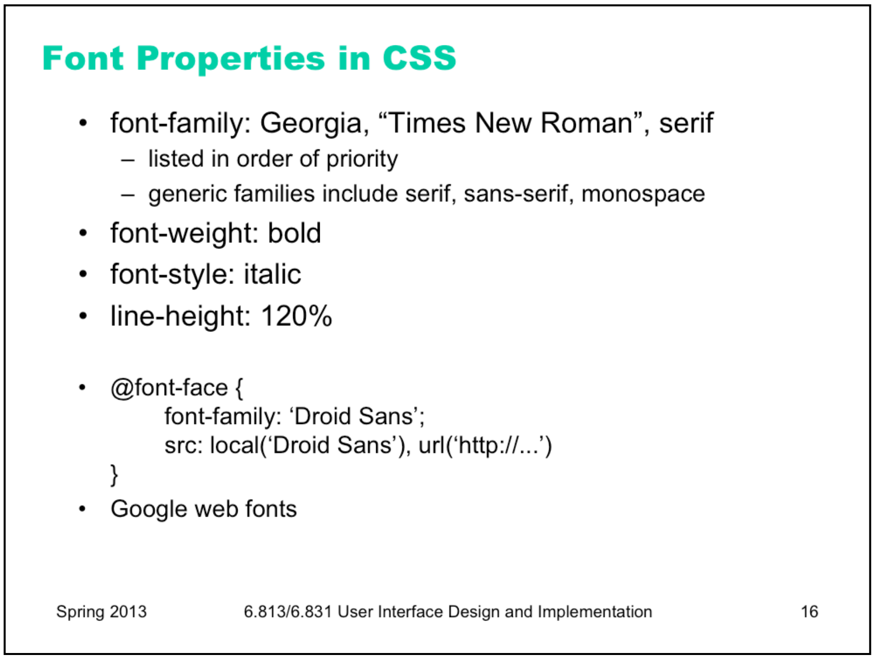
Font Selection

- Simplicity & contrast
  - Don’t use more than 2 or 3 typefaces
    - E.g., one for body text, one for display text
  - Don’t use two faces from the same font category
    - e.g. only one sans serif
  - Use size, weight, style (e.g. italic/small caps), hue to establish essential contrasts
    - But 4-5 font varieties should be enough

Spring 2013 6.813/6.831 User Interface Design and Implementation 15

In general, decisions about typography are like other decisions in graphic design: use font selection to make important contrasts, and otherwise keep your font choices simple. Don’t use more than 2 or 3 typefaces (if that many). You might use a serif face for body text, and a sans serif face for display text. Many interfaces have no real need for body text at all, in which case you can easily get away with a single typeface.

Within the typefaces you chose, use variation of size and style (and color) to establish the necessary contrasts. Size, in particular, makes it easy to establish a hierarchy, such as headings and subheadings. Even so, 4-5 fonts in all should be all you need.



Font Properties in CSS

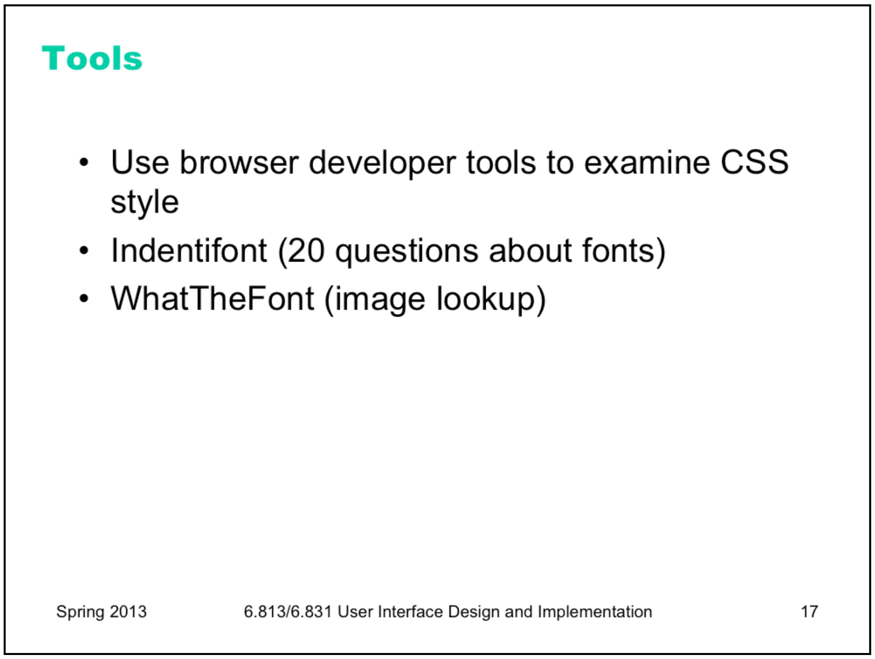
- font-family: Georgia, “Times New Roman”, serif
  - listed in order of priority
  - generic families include serif, sans-serif, monospace
- font-weight: bold
- font-style: italic
- line-height: 120%

- @font-face {

```
@font-face {
        font-family: ‘Droid Sans’;
        src: local(‘Droid Sans’), url(‘http://...’)
}
```

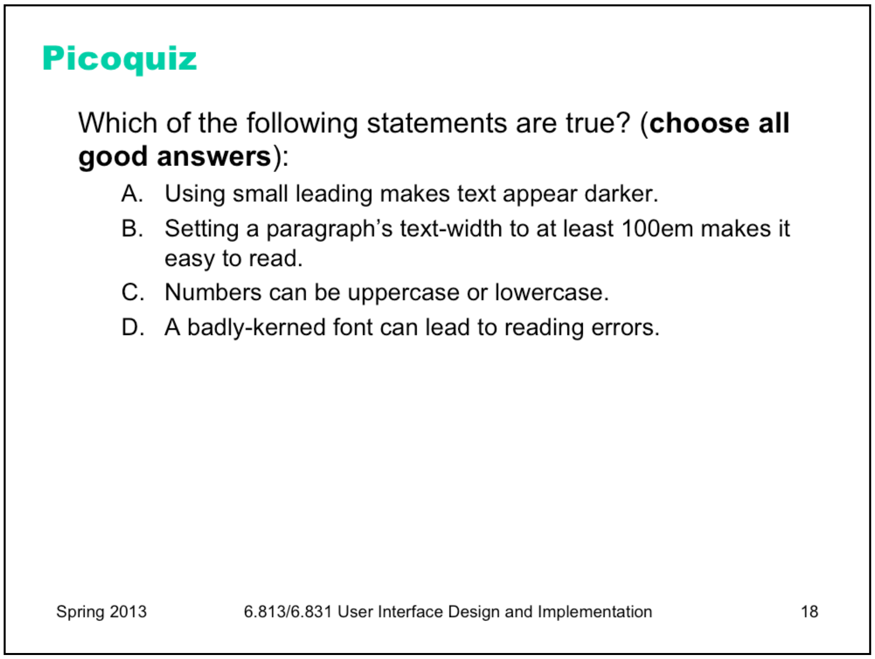
- Google web fonts

Spring 2013 6.813/6.831 User Interface Design and Implementation 16



Tools

- Use browser developer tools to examine CSS style
- Indentifont (20 questions about fonts)
- WhatTheFont (image lookup)



Picoquiz

Which of the following statements are true? (**choose all good answers**):

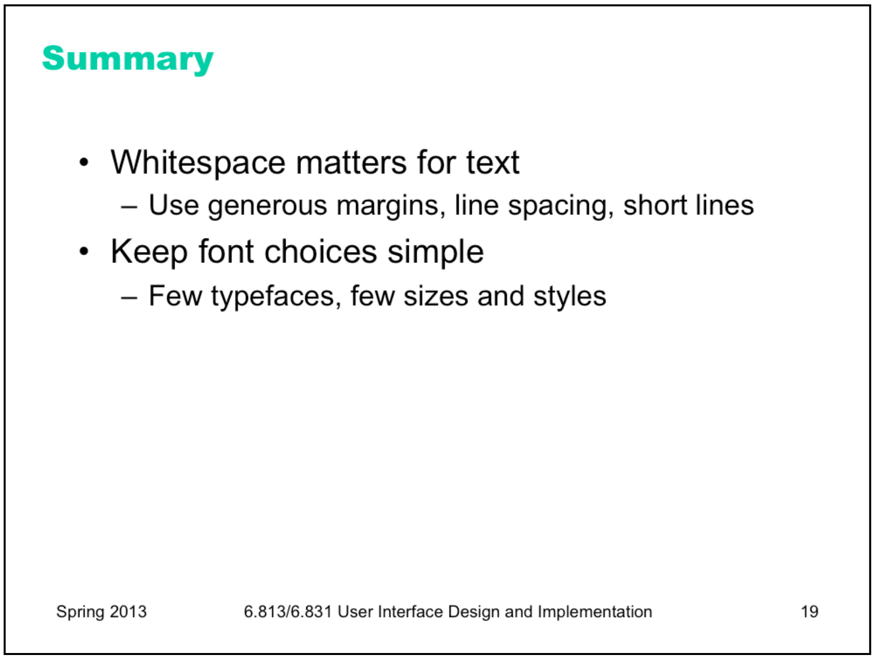
A. Using small leading makes text appear darker.

B. Setting a paragraph's text-width to at least 100em makes it easy to read.

C. Numbers can be uppercase or lowercase.

D. A badly-kerned font can lead to reading errors.

To answer the picoquiz questions in this lecture, go to:
http://courses.csail.mit.edu/6.831/2013/picoquiz?**lectureId=17**



Summary

- Whitespace matters for text
  - Use generous margins, line spacing, short lines
- Keep font choices simple
  - Few typefaces, few sizes and styles