Typography and Fonts

Elements of Graphics
CS324e
Spring 2019
What is Typography?

- Visual composition of words
- Reinforcement of word meaning via appearance
- Design and aesthetics of written word
- More than just fonts
Glyphs

- Characters available in a given typeface
  - Note that a font is a specific form of a typeface family (e.g. Helvetica Regular and Helvetica Italic)
- Glyphs change based on the language the font depicts
- Also includes numbers and special characters
Serif vs Sans-Serif

- Serif
  - Short finishing lines at the end of characters
  - More readable in print
  - Causes less eye strain for large bodies of text
- Sans-serif
  - Lack finishing lines
  - More readable on computer screens
  - Adds emphasis
  - More recognizable

(Kensington Design)
Kerning and Tracking

- Kerning is the adjustment of space between characters
- Kerning pairs allow for better spacing between specific characters
- Tracking is the adjustment of space between groups of characters

Three kerning distances (Wikipedia)
Alignment

- Align left used for left-to-right languages
- Align right used for right-to-left languages
- Justified aligns the ends of the text
  - Avoids paragraph rag
  - Can create spacing issues
- Centered aligns the text to the middle of the column
  - Useful for emphasizing titles
  - Hard to read, so otherwise avoid
Vector Fonts

- Vector fonts are fonts made using vectors rather than bitmaps
- Vectors mathematically defined allowing for scaling
- TrueType (.ttf) is older font standard
- OpenType (.otf) is newer font standard with more glyphs and features
- Both are usable in Processing
Vector Font Example

(https://blog.typekit.com/2005/12/13/quality_in_type/)
Bitmap Font Example

(http://www.sibcode.com/font-editor/)
Question

Which of these is vector and which is bitmap?

(a) Text

(b) Text
Initializing Fonts

- Display all available fonts on a machine:
  ```java
  String[] fontlist = PFont.list();
  printArray(fontlist);
  ```
- `createFont()` converts PFont to desired font
- `textFont()` sets the PFont as the text font type
PFont courier;

void setup() {
    size(100, 100);
    courier = createFont("Courier", 32);
    textFont(courier);
}

void draw() {
    text("Hello", 0, 32);
}
Activity

- Run this code (or modify to include a font that you have)
- Make sure the text is actually displaying in the correct font (Processing is not the best tool for handling fonts!)
Text Attributes

- `textLeading()` sets spacing between lines of text
- `textAlign()` sets alignment based on parameters `LEFT`, `CENTER` and `RIGHT`
- `textSize()` sets text size
- `textWidth()` returns width of character or String
Hands-on: Displaying Text

❖ Today’s activities:

1. Write some text to the sketch’s screen then experiment with font-type, text size, spacing, etc

2. Once you are happy with the text’s appearance, create a “typewriter” that will display characters to the screen as the user types

3. Add additional features (such as delete, carriage return, etc) if time is available