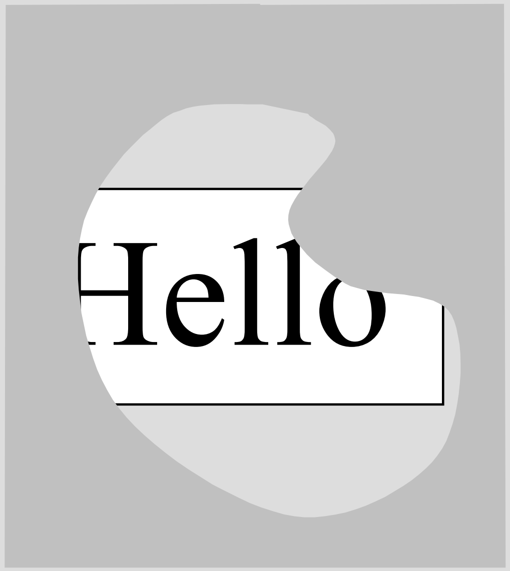
Clipping, Text, Fonts

Clipping Path

The rendered image may be clipped by a clipping path



Clipping Path

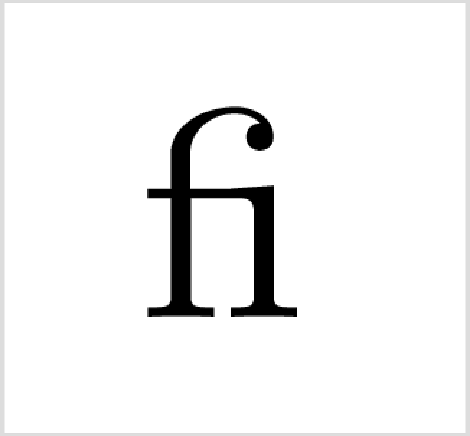
- Graphics2D methods:
  - void setClip(Shape s)
    - Replaces current clipping region with the Shape
  - void clip(Shape s)
    - Clips the current clipping region...
    - i.e., Clipping region becomes intersection of current clipping region and this shape
- The current Clip affects subsequent drawing only

Fonts

- Glyph:
  - Geometry describing shape of a character
- Font:
  - A collection of glyphs for an entire alphabet
- Relationship between glyphs and characters is not necessarily 1-to-1
  - e.g., Some glyphs correspond to a sequence of characters
  - e.g., a Ligature is when 2 characters are rendered in combination

Ligature

- A glyph may contain multiple letters
- A common ligature:



Font

- Logical fonts in Java
  - Serif
  - SansSerif
  - Monospaced
  - Dialog
  - DialogInput
- Logical Fonts improve portability
  - Map to physical fonts on system
  - e.g., SansSerif on Windows maps to Arial
- Font styles
  - PLAIN
  - ITALIC
  - BOLD
- Derived font
- Font metrics

Fonts

- Font(String name, int style, int size)
- Style:
  - Combine with bitwise OR, |
  - Font.BOLD, Font.ITALIC, Font.PLAIN
- Size:
  - In points
- In Graphics2D
  - void setFont(Font f)
- Drawing Text:
  - void drawString(String s, int x, int y)

Derived Fonts

- Can derive new fonts from existing ones
  - Font deriveFont(int style)
  - Font deriveFont(float size)
  - Font deriveFont(int style, float size)
  - Font deriveFont(AffineTransform tx)
  - Font deriveFont(int style, AffineTransform tx)

Glyphs as Shape Objects

- Java2D allows us to create a shape object from glyphs of characters from a font
- Here's how:
  - Begin with a Font
  - Extract the FontRenderContext from the Graphics2D object
  - Call the method of the Font class
    - GlyphVector createGlyphVector(FontRenderContext frc, String str)
  - This gives you a GlyphVector for str
  - Call either of these on the GlyphVector:
    - Shape getOutline()
    - Shape getOutline(float x, float y)

Glyph

- Geometry describing shape of character
- Obtain a glyph

```
Font font = new Font("Serif", Font.BOLD, 144);
FontRenderContext frc = g2.getFontRenderContext();
GlyphVector gv = font.createGlyphVector(frc, "Java");
Shape glyph = gv.getOutline(100,200);
```