

# Process Work

## 1.0 Letter Construction

### 1.1 Practicing with Procreate

This is a record of my practice using Procreate (Image 1). The Apple Pencil allows me to write without worrying about ink smudging on the paper.

However, since the brush in Procreate simulates calligraphy strokes, and the Apple Pencil does not provide physical feedback on actual pen pressure, I spent a significant amount of time practicing to gain better control. I focused on writing neat and elegant English letters as accurately as possible.

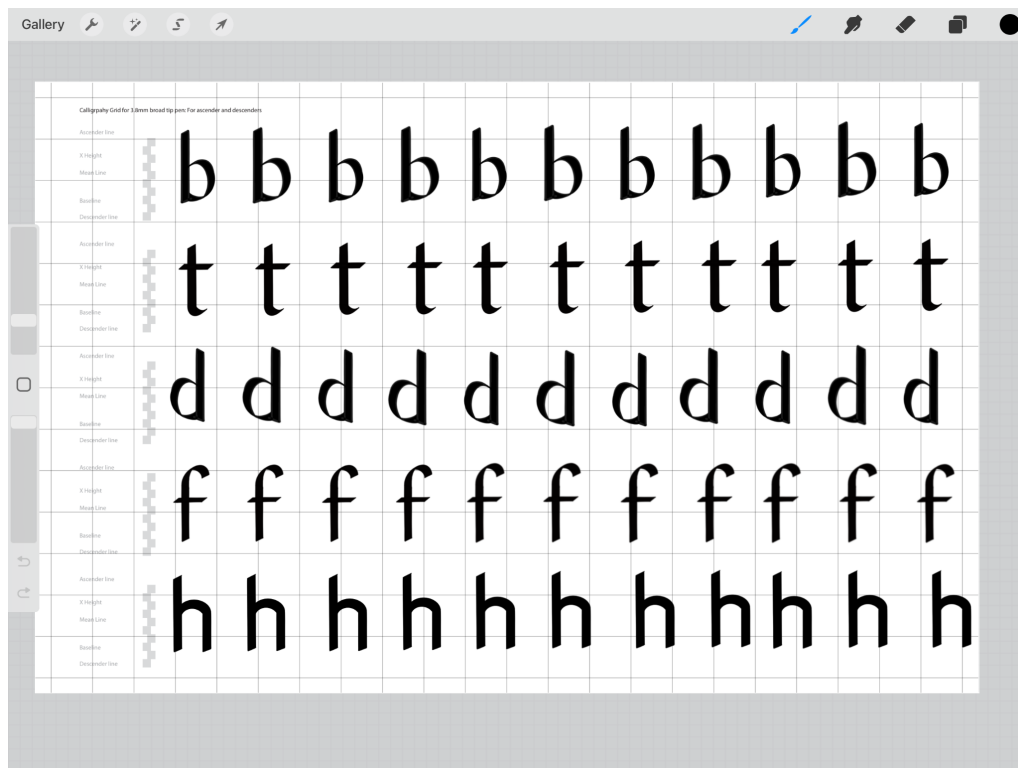


Image 1. Ascender Calligraphy Practice

## 1.2 Writing "Hamburgetfontstiv"

After gaining more confidence in my practice, I began writing “**Hamburgetfontstiv**” (Image 2).

I used the **grid system in Procreate** to help control spacing. Since I find it difficult to write precisely when following the grid lines too strictly, I adjusted my approach by setting my baseline slightly above the bottom line—this is a personal habit that makes writing more comfortable for me.



Image 2. “Hamburgetfontstiv” Practice

### 1.3 Labeling the Anatomy Structure

Once I finished writing, I proceeded to label the **Anatomy structure** (Image 3).

In fact, during my practice, I had already memorized all the structural elements. However, for accuracy, I referred to the **in-class quiz** to verify my annotations and ensure everything was labeled correctly.

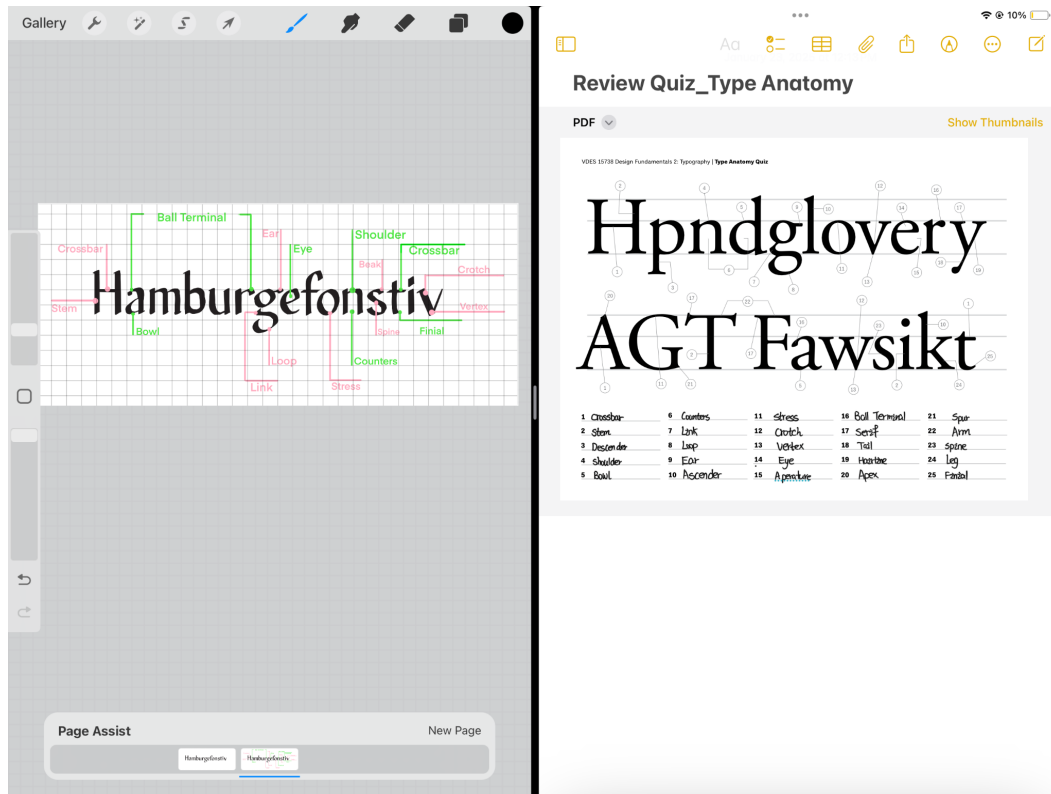


Image 3. label the Anatomy structure

## 2.0 Paragraph Setting

### 2.1 Typeface Selection & Initial Considerations

The professor provided the following typefaces:

- 'Abril Fatface' – A highly decorative serif typeface, typically used for **headings**, not body text.
- 'Georgia' – A **serif typeface designed for screen reading**, making it suitable for body text.
- 'Roboto' & 'Ubuntu' – **Sans-serif fonts** that may work well depending on the design.
- 'Bitter' – Another **serif typeface suitable for body text**.
- 'Kelly Slab' – A decorative typeface that is **less readable for long texts**.

Initially, the default paragraph style in the HTML used 'Abril Fatface', which is not suitable for body text due to its **strong decorative nature**, making it **harder to read in long-form content**. Thus, I needed to replace it with a more **readable and elegant** typeface, narrowing my choices to **Georgia, Bitter, and Roboto**.

However, I eventually eliminated **Bitter** because:

1. **It has a more pronounced personality**, which could be distracting for extended reading.
2. **Georgia is explicitly designed for digital readability**, making it the best serif choice.
3. **Roboto is a strong sans-serif option**, but I wanted to test serif vs. sans-serif readability before making a final decision.

After finalizing the typeface choices, I also considered other parameters:

- **Column width** (optimal range: **600-800px**)
- **Font size** (generally **16-18px** for readability)
- **Line height** (usually **1.5 or more** to avoid text crowding)
- **Letter spacing** (adjustable based on the typeface's natural spacing)

## 2.2 First Version: Initial Adjustments

### CSS Code:

```
/* First Attempt at Paragraph Formatting */
```

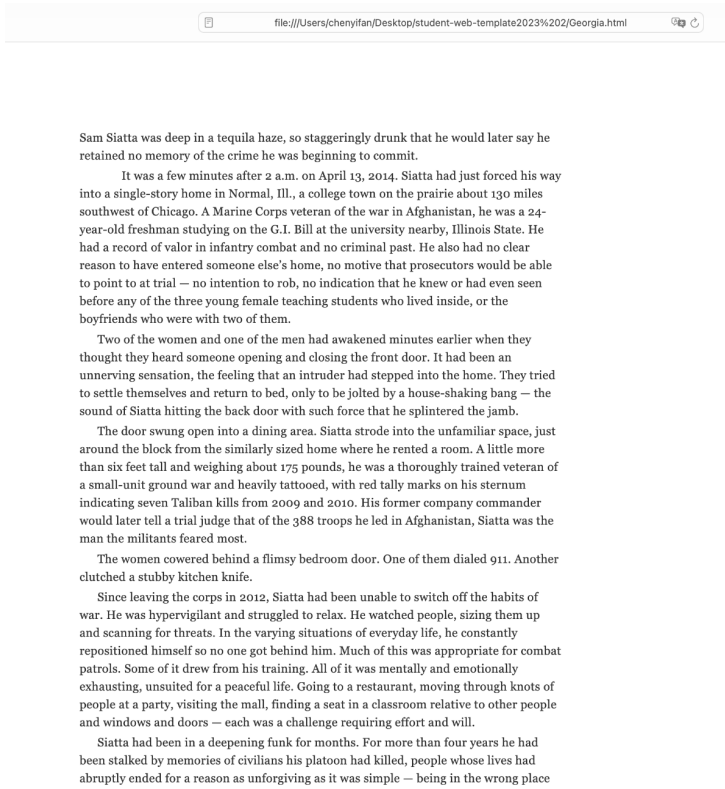
```
#theparagraph {  
    width: 650px; /* Setting an optimal column width for readability */  
}
```

```
#theparagraph p {  
    font-family: 'Georgia', serif;  
    font-size: 16px; /* Standard size, but may feel too small */  
    line-height: 1.5; /* Balanced line spacing */  
    letter-spacing: 0.3px; /* Slight letter spacing for improved readability */  
    font-weight: 400;  
    margin-bottom: 1.2em; /* Ensuring space between paragraphs */  
}
```

```
/* Set paragraph indentation */
```

```
#theparagraph p + p {  
    text-indent: 1.5em;  
    margin-top: -1em;  
}
```

Here is the screenshot of the first version of the paragraph:



Pros & Cons of First Version

| Pros                            | Cons                                                               |
|---------------------------------|--------------------------------------------------------------------|
| Elegant and readable serif font | Font size felt slightly too small, making reading more difficult   |
| Comfortable line height (1.5)   | Column width felt too wide, affecting readability                  |
|                                 | Line spacing could be slightly increased for better visual comfort |

## 2.3 Second Version: Switching to a Sans-Serif Approach

### CSS Code:

```
/* Optimized paragraph layout: Width, font, spacing */

#theparagraph {

    max-width: 65ch; /* Restrict column width based on character count */

    margin: 0 auto; /* Center align the paragraph */

    padding: 1rem; /* Add padding for better spacing */

}


#theparagraph p {

    font-family: 'Roboto', sans-serif; /* Testing a sans-serif option */

    font-weight: 300; /* Lighter weight for better readability */

    font-size: 1rem; /* Equivalent to 16px */

    line-height: 1.6; /* Comfortable line height */

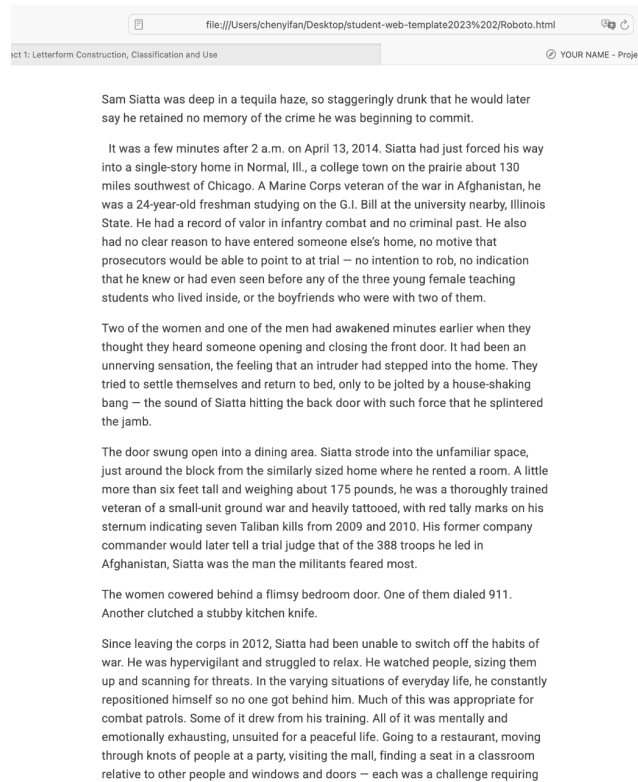
    letter-spacing: 0.01em; /* Minor letter spacing adjustment */

    color: #444;

    margin-bottom: 1rem; /* Maintain paragraph spacing */

}
```

## Here is the screenshot of the Second version of the paragraph:



## Pros & Cons of Second Version

| Pros                                                       | Cons                                                                             |
|------------------------------------------------------------|----------------------------------------------------------------------------------|
| Good paragraph spacing, making it more visually structured | Last line of some paragraphs had only a few words, making it visually unbalanced |
| Column width was more appropriate for comfortable reading  | Roboto felt somewhat "stiff" and lacked the elegance of Georgia                  |

## 2.4 Third Version: Adjustments to Improve Readability

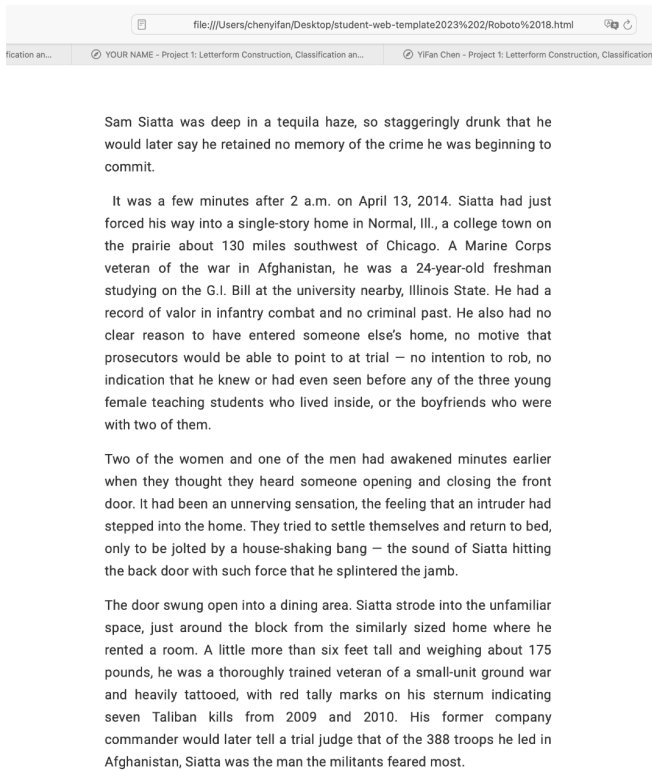
### CSS Code:

/\* Optimized paragraph layout: Width, font, spacing \*/

```
#theparagraph {  
    max-width: 65ch; /* Restrict column width */  
    margin: 0 auto; /* Center alignment */  
    padding: 1rem; /* Adding padding for whitespace */  
}
```

```
#theparagraph p {  
    font-family: 'Roboto', sans-serif; /* Keeping sans-serif but refining parameters */  
    font-size: 18px; /* Increasing font size for readability */  
    line-height: 1.7; /* Slightly more spacing between lines */  
    letter-spacing: 0.3px; /* Adjusting letter spacing */  
    max-width: 700px; /* Setting a defined max-width */  
    text-align: justify; /* Justifying text for a more structured look */  
    margin-bottom: 1rem; /* Maintaining paragraph spacing */  
}
```

Here is the screenshot of the Third version of the paragraph:



Pros & Cons of Third Version

| Pros                                           | Cons                                                                        |
|------------------------------------------------|-----------------------------------------------------------------------------|
| Increased font size improved readability       | Still encountered occasional cases where the last line had only a few words |
| Balanced column width made reading comfortable | Roboto felt less elegant compared to Georgia                                |

## 2.5 Fourth & Final Version: The Best Balance

### CSS Code:

```
/* *****Final Version of Paragraph Formatting***** */

#theparagraph {
    width: 650px; /* Optimal column width for readability */
}

#theparagraph p {
    font-family: 'Georgia', serif; /* Choosing a refined serif font */
    font-size: 18px; /* Ideal size for comfortable reading */
    line-height: 1.75; /* Well-balanced spacing */
    letter-spacing: 0.4px; /* Slight increase in letter spacing */
    max-width: 700px; /* Controlling the column width */
    text-align: left; /* Keeping left alignment for better flow */
    margin-bottom: 1.5rem; /* Increasing space between paragraphs */

    hyphens: auto; /* Automatic hyphenation to improve text wrapping */
    word-break: break-word; /* Allowing words to break naturally */

    widows: 2; /* Ensuring the last line doesn't contain a single word */
    orphans: 2; /* Avoiding single-line paragraphs at the top of a new section */
}
```

```
/* Indenting subsequent paragraphs */
```

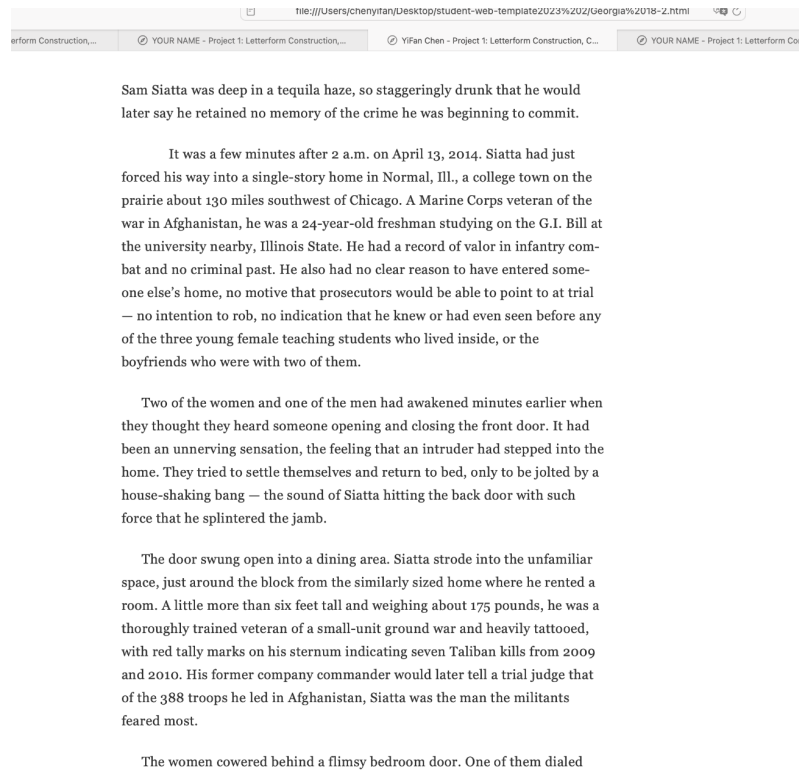
```
#theparagraph p + p {
```

```
    text-indent: 1.5em;
```

```
    margin-top: 0.5rem; /* Ensuring paragraph separation */
```

```
}
```

**Here is the screenshot of the Final version of the paragraph:**



### Pros & Cons of Final Version

| Pros                                                   | Cons                                                                                                                                     |
|--------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| Maintains paragraph spacing while ensuring readability | Still occasionally has paragraphs where the last line contains only two words, but this is reasonable as it happens in only one instance |
| Uses an elegant serif typeface, 'Georgia'              |                                                                                                                                          |
| Balanced line spacing (1.75) makes text easy to read   |                                                                                                                                          |
| Widow and orphan rules reduce awkward line breaks      |                                                                                                                                          |

## 2.6 Expanding Beyond the Given Parameters: A Thought Process

At the beginning of this project, I focused on the five required parameters: **typeface, font size, leading, tracking, and column width**. These are fundamental elements in typography that directly impact readability and aesthetic appeal. However, as I experimented with different settings, I realized that **purely adjusting these five aspects was not enough to create an optimal reading experience**.

The first issue I encountered was **uneven line breaks** at the end of paragraphs. Even after adjusting the font size and column width, I noticed that some paragraphs still ended with **only one or two words on the last line**, which disrupted the overall balance of the text block. This led me to explore solutions beyond the given parameters.

(Here are the links that helped me:

<https://stackoverflow.com/questions/4212909/can-css-force-a-line-break-after-each-word-in-an-element>

<https://developer.mozilla.org/en-US/docs/Web/CSS/line-break>)

I started by implementing **hyphenation**, which allowed words to break naturally at appropriate syllables, preventing large gaps or awkward spacing. This adjustment immediately improved paragraph consistency, especially for longer words that would otherwise force irregular line breaks.

(Here are the links that helped me:

<https://developer.mozilla.org/en-US/docs/Web/CSS/hyphens>

<https://medium.com/clear-left-thinking/all-you-need-to-know-about-hyphenation-in-css-2baee2d89179>)

Another problem I noticed was **text flow**. With a fixed column width, there were instances where words stretched unnaturally across the line, creating an uncomfortable reading rhythm. To address this, I added **word-breaking rules**, ensuring that text would break naturally if it exceeded the line width. This helped maintain even spacing while preventing large empty spaces at the end of lines.

(Here are the links that helped me:

<https://developer.mozilla.org/en-US/docs/Web/CSS/text-overflow>

<https://developer.mozilla.org/en-US/docs/Web/CSS/word-break>

[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_text/Wrapping\\_breaking\\_text](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_text/Wrapping_breaking_text)

<https://stackoverflow.com/questions/3775810/stop-word-wrap-dividing-words>)

Additionally, I discovered that **widows and orphans were affecting paragraph balance**. Some paragraphs had **a single word** left alone on the last line, while others started a new section with only one line of text before moving onto the next column. This made the layout look disconnected and inconsistent. By adding **widow and orphan controls**, I was able to enforce a more structured appearance, ensuring that single words or isolated lines would not disrupt the overall text flow.

(Here are the links that helped me:

<https://stackoverflow.com/questions/70338798/how-to-avoid-orphan-word-on-a-line-with-css>

<https://developer.mozilla.org/en-US/docs/Web/CSS/orphans>

<https://medium.com/design-for-experience/word-flow-on-the-web-7adae064367c>

<https://stackoverflow.com/questions/2914988/css-how-to-align-one-word-left-and-another-word-right-within-the-same-div>

<https://developer.mozilla.org/en-US/docs/Web/CSS/text-align>

[https://css-tricks.com/almanac/properties/o/orphans/#:~:text=In%20short%2C%20in%20CSS%20specifically,split%20on%20the%20old%20page.\)](https://css-tricks.com/almanac/properties/o/orphans/#:~:text=In%20short%2C%20in%20CSS%20specifically,split%20on%20the%20old%20page.)

While making these adjustments, I also considered **margin-bottom and paragraph indentation** to further enhance readability. Increasing the **space between paragraphs** gave the text a more natural rhythm, making it easier to scan. I also introduced a **small paragraph indent** to reinforce structural clarity, improving the flow between ideas.

(Here are the links that helped me:

<https://developer.mozilla.org/en-US/docs/Web/CSS/margin-bottom>

<https://stackoverflow.com/questions/41253908/changing-spacing-between-paragraphs-and-inside-of-paragraphs>

<https://blog.hubspot.com/website/how-to-indent-css#:~:text=How%20to%20Indent%20Paragraphs%20in,to%20the%20value%20you%20want.>)

Through this process, I learned that typography is not just about adjusting a **few predefined settings**—it requires a **holistic approach**, considering how different elements interact to create a **cohesive and visually appealing text layout**. The small refinements I added were not just technical optimizations, but **deliberate design choices aimed at crafting a comfortable reading experience**.

(Here are the links that helped me:

[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_cascading\\_variables/Using\\_CSS\\_custom\\_properties](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_cascading_variables/Using_CSS_custom_properties)

<https://aguayo.co/en/blog-aguayo-user-experience/systemic-thinking-holistic-approach-to-improve-user-experience/#:~:text=Consideration%20of%20All%20Aspects%3A%20Holistic,that%20influence%20the%20user%20experience.>)

Ultimately, this experiment reinforced the idea that **good typography goes beyond basic parameters**—it is about **understanding the reader's experience and fine-tuning every detail to achieve harmony**.

## **3.0 Typeface Classification**

### **3.1 Letter: "g"**

#### **Initial Observations & Hypothesis**

The lowercase "g" in the provided image stands out due to its distinctive two-story structure, high contrast in stroke weight, and sharp serifs. These characteristics suggest that it belongs to the serif family. Additionally, the vertical stress and noticeable contrast between thick and thin strokes indicate that it might be part of the Transitional Serif category.

#### **Key Characteristics Supporting the Conclusion**

##### **Increased Stroke Contrast**

- Transitional serif typefaces, which emerged in the 18th century, exhibit higher contrast between thick and thin strokes compared to Old Style serifs.
- The "g" in this image shows a clear distinction between its thick and thin strokes, a defining feature of Transitional typefaces.

##### **Vertical Stress**

- Unlike Old Style fonts, which have a diagonal stress, Transitional typefaces emphasize verticality.
- The bowl and loop of this "g" are symmetrically aligned, confirming a strong vertical axis.

##### **Distinctive Double-Story "g"**

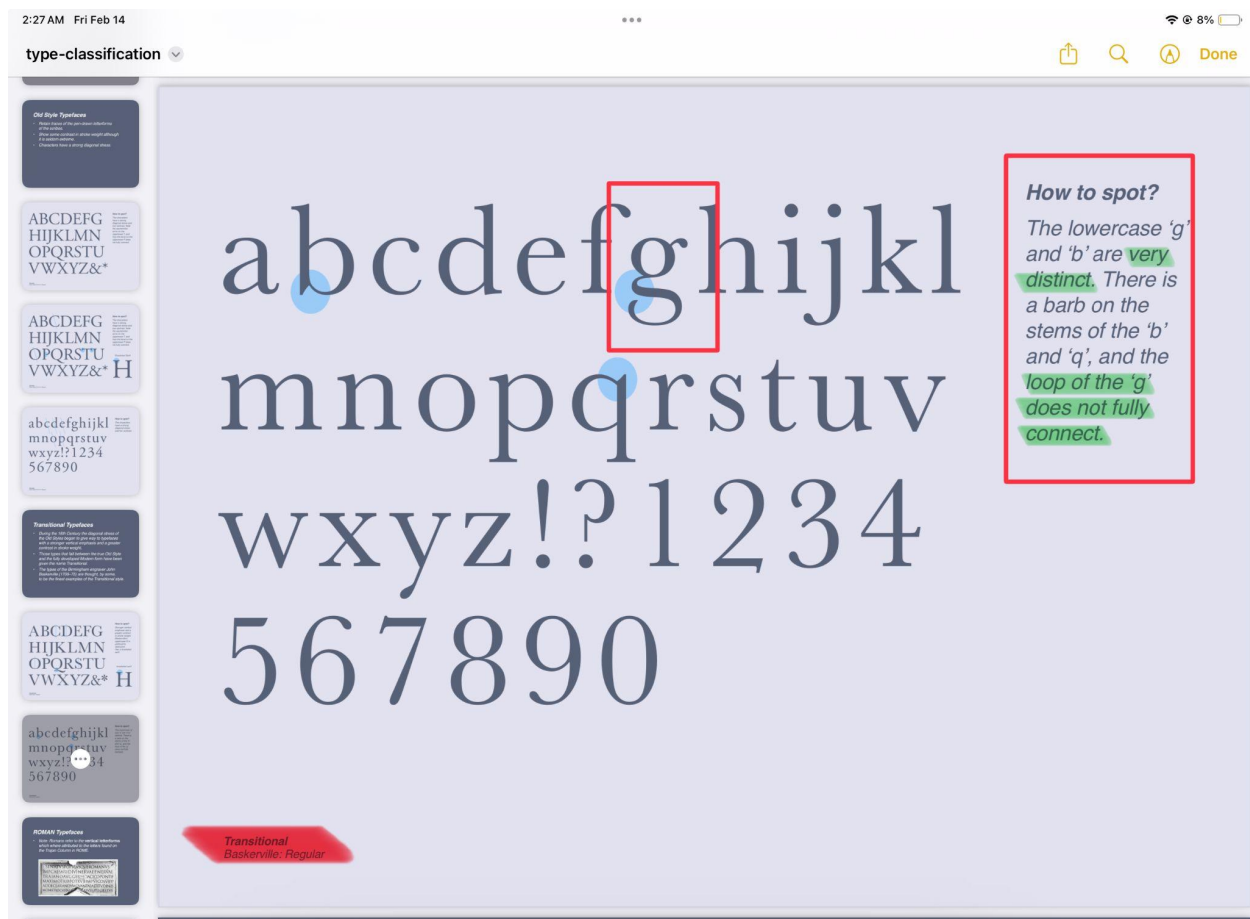
- The lowercase "g" has a well-defined two-story form, with a large, open loop.
- In Baskerville, the loop does not fully connect at the bottom, leaving a small gap, which is visible in this image.

##### **Serifs & Barbs**

- The serifs are sharp and slightly bracketed, unlike the rounded Old Style serifs.
- Baskerville's b, q, and g often feature small barbs on the stems, which is evident in this "g".

#### **Final Classification: Transitional Serif (Baskerville Regular)**

Cross-check answers with the PowerPoint from class:



## **3.2 Letter: "a"**

### **Initial Observations & Hypothesis**

The lowercase "a" in the provided image features thick, blocky serifs that are unbracketed, giving it a chunky and geometric appearance. This suggests that it differs from Old Style or Transitional Serif fonts, which typically have more delicate serifs. Instead, this feature aligns with Slab Serif typefaces, which are known for their strong, rectangular serifs and low contrast between strokes.

### **Key Characteristics Supporting the Conclusion**

#### **Thick, Unbracketed Serifs**

- Unlike Old Style or Transitional typefaces, which have curved (bracketed) serifs, Slab Serif fonts have flat, unbracketed serifs.
- The “a” in this image exhibits these thick, unbracketed serifs, making it a strong candidate for Slab Serif classification.

#### **Even Stroke Weight**

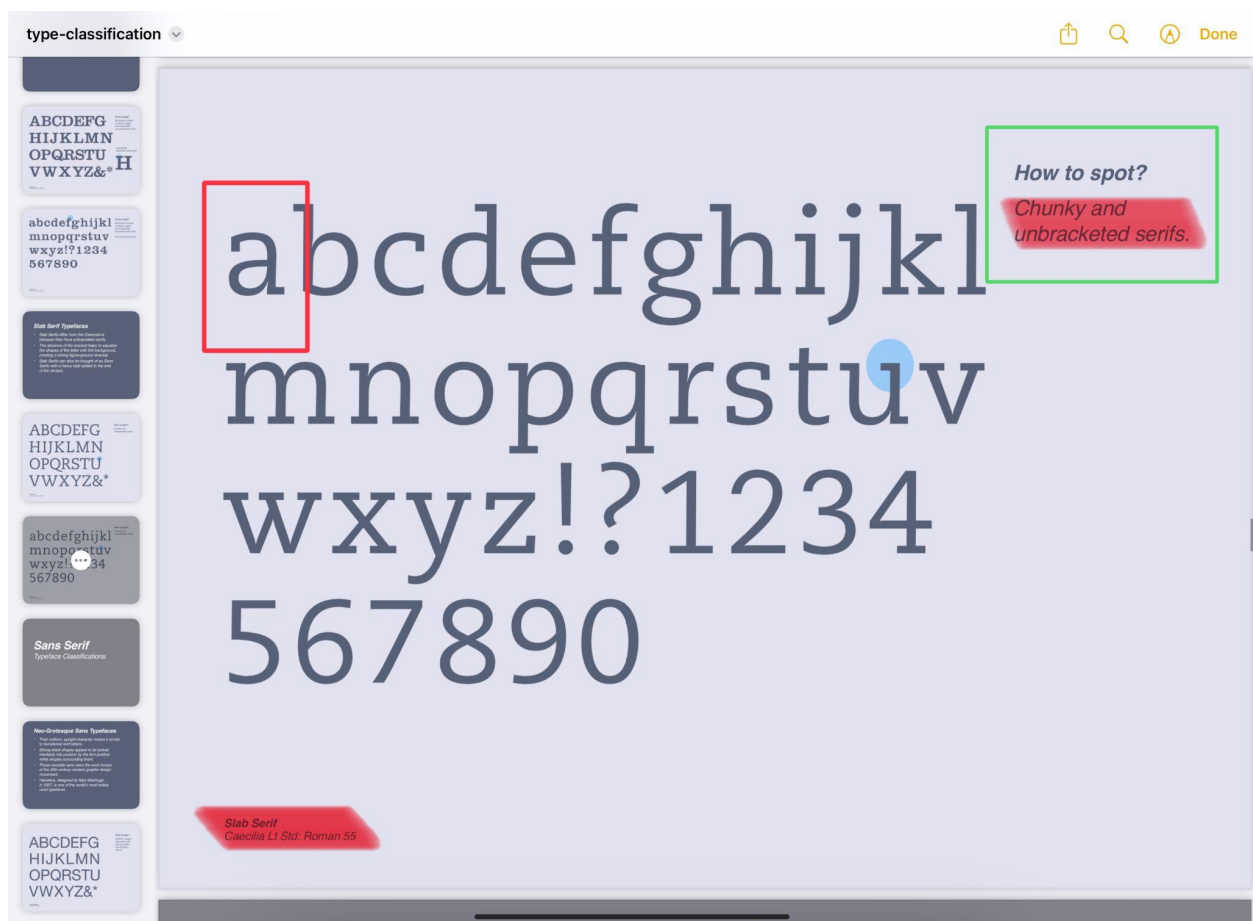
- The letter has consistent stroke weight throughout, lacking the dramatic contrast of Transitional or Modern serifs.
- This uniformity in stroke width is a defining trait of Slab Serif fonts.

#### **Strong Figure-Ground Reversal**

- The absence of brackets in Slab Serifs helps to equalize the shapes of the letter with the background, creating a bold and structured appearance.
- This makes Slab Serif fonts highly legible and impactful, which is why they are often used for headlines and branding.

**Final Classification: Slab Serif (Caecilia Lt Std: Roman 55)**

Cross-check answers with the PowerPoint from class:



### **3.3 Letter: "o"**

#### **Initial Observations & Hypothesis**

The lowercase "o" in the provided image appears to be perfectly circular, with even stroke weight throughout, and no serifs. These features suggest that it likely belongs to the Sans-Serif family. More specifically, its highly geometric design indicates that it fits within the Geometric Sans Serif category.

#### **Key Characteristics Supporting the Conclusion**

##### **Perfect Circular Form**

- Unlike Humanist or Transitional Sans Serifs, which often have slight variations in stroke weight or curvature, Geometric Sans Serifs strive for mathematical perfection.
- The "o" in this image appears almost like a perfect circle, which is a classic feature of Geometric Sans typefaces like Futura.

##### **Even Stroke Weight**

- The lack of contrast between thick and thin strokes is another key identifier.
- Geometric Sans Serifs maintain uniform stroke width, unlike Transitional or Modern serifs, which have high contrast.

##### **Minimalistic, Modern Look**

- Geometric Sans Serif fonts emerged in the 20th century, heavily influenced by Bauhaus design principles.
- The simple, clean, and almost futuristic appearance of this "o" suggests that it belongs to this category.

#### **Final Classification: Geometric Sans Serif (Futura Medium)**

Cross-check answers with the PowerPoint from class:

type-classification 28 of 35

## Geometric Sans Typefaces

- Types of an apparently even stroke weight with the appearance of **perfect circle O's**, Isosceles A's, etc
- This form is found in the earliest Greek inscriptions and has been periodically revived, notably in Imperial Rome, during the English Classical Revival, and at the Bauhaus.
- Futura is considered to be the archetypal geometric sans.

ABCDEFG  
HIJKLMN  
OPQRSTU  
VWXYZ&\*

abcdefghijklmnopqrstuvwxyz  
1234567890

Classify the following letters

O

O

type-classification

How to spot?  
The Os are perfect circles, and the apex of the A and M are sharp triangles.

ABCDEFG  
HIJKLMN  
OPQRSTU  
VWXYZ&\*

Geometric  
Futura Medium

### **3.4 Letter: "e"**

#### **Initial Observations & Hypothesis**

The lowercase "e" in the provided image has a noticeable diagonal stress and low contrast in stroke weight, with subtle curves that resemble handwritten forms. These characteristics suggest that it likely belongs to the Old Style Serif category, which retains calligraphic influences from the Renaissance era.

#### **Key Characteristics Supporting the Conclusion**

##### **Diagonal Stress**

- Unlike Transitional and Modern Serifs, which have a strong vertical stress, Old Style typefaces retain a diagonal stress, resembling the natural angle of handwritten text.
- The “e” in this image clearly demonstrates this diagonal stress, making it a key identifier for Old Style classification.

##### **Low Stroke Contrast**

- While Transitional and Modern typefaces have high contrast between thick and thin strokes, Old Style fonts tend to have gentler contrast.
- The low contrast in this “e” confirms that it does not belong to the later serif styles.

##### **Handwritten Influence**

- Old Style typefaces originated from scribes and hand-drawn letterforms, which is why their strokes appear more organic and less rigid.
- The curved terminals and soft serifs in this "e" reflect these origins, making it a strong candidate for Adobe Garamond Pro.

**Final Classification: Old Style Serif (Adobe Garamond Pro: Regular)**

Cross-check answers with the PowerPoint from class:

