Part 3 - Designing Text

Most interfaces use way too many **font sizes**. Unless a team has a rigid design system in place, it's not uncommon to find that every pixel value from 10px to 24px has been used in the UI somewhere. Choosing font sizes without a system is a bad idea for two reasons:

1. It leads to annoying inconsistencies in your designs.
2. It slows down your workflow.

Here's an example of a scale that works well for most projects and aligns nicely with the spacing and sizing scale recommended in "Establishing a spacing and sizing system": It's constrained just enough to speed up your decision making, but isn't so limited as to make you feel like you're missing a useful size.

```
12px
14px
16px
18px
20px
24px
30px
36px
48px
60px
72px
```

It's constrained just enough to speed up your decision making, but isn't so limited as to make you feel like you're missing a useful size.

When you're building a type scale, don't use **em** units to define your sizes. Because em units are relative to the current font size, the computed font size of nested elements is often not actually a value in your scale.

Stick to px or rem units — it's the only way to guarantee you're actually sticking to the system.

Use good fonts

Play it safe For UI design, your safest bet is a fairly **neutral sans-serif** — think something like Helvetica. If you really don't trust your own taste, one great option is to rely on the system font stack:

```
-apple-system, Segoe UI, Roboto, Noto Sans, Ubuntu, Cantarell, Helvetica Neue;
```

- Ignore typefaces with less than five weights
- A great way to limit the number of options you have to choose from is to crank that up to 10+ (to account for italics)
- On **Google Fonts** specifically, that cuts out 85% of the available options, leaving you with less than 50 sans-serifs to choose from.

- Trust the wisdom of the crowd: If a font is popular, it's probably a good font. Most font directories will let you sort by popularity,

- Steal from people who care: Inspect some of your favorite sites and see what typefaces they are using (look for **"font-family"** attribute)

- Keep your line length in check: When styling paragraphs, it's easy to make the mistake of fitting the text to your layout instead of trying to create the best reading experience. For the best reading experience, **make your paragraphs wide enough to fit between 45 and 75 characters per line** (around 12-13 words per line). The easiest way to do this on the web is using em units, which are relative to the current font size. A width of 20-35em will get you in the right ballpark

- Baseline, not center: A better approach is to align mixed font sizes by their baseline, which is the imaginary line that letters rest on. --> use **align-items: baseline** instead of *align-items: center*. The result is a simpler, cleaner look than what you get when you center two pieces of text and offset their baselines.

- Accounting for line length: your line-height and paragraph width should be proportional — narrow content can use a shorter line-height like 1.5, but wide content might need a line-height as tall as 2. Line-height is proportional 106

- Accounting for font size: When text is small, extra line spacing is important because it makes it a lot easier for your eyes to find the next line when the text wraps. But as text gets larger, your eyes don't need as much help. This means that for large headline text you might not need any extra line spacing, and a line-height of 1 is perfectly fine.

Line-height and font size are *inversely proportional* — use a taller line-height for small text and a shorter line-height for large text

- Don't center long form text: Center-alignment can look great for headlines or short, independent blocks of text. But if something is longer than two or three lines, it will almost always look better **left-aligned**.

- Right-align numbers: If you're designing a table that includes numbers, right-align them.