

New Note Format!

You might notice CS 61B notes look a little different from my old CS 61A notes, and there's a good reason for that! I've switched from writing notes in Boostnote to Typora, and here are some of the big reasons why:

Typora code is color-coded

Boostnote code is just normal text written using a fixed-width font, which isn't representative of how modern text editors work.

```
/** Traditional first program.
 * @author P. N. Hilfinger */

public class Hello {
    /** Print greeting.
     * ARGV is ignored. */

    public static void main(String[] args){
        System.out.println("Hello, world!");
    }
}
```

Typora lets me select a language for writing code, then color-codes it so comments, classes etc. are more clearly delineated from one another:

```
/** Traditional first program.
 * @author P. N. Hilfinger */

public class Hello {
    /** Print greeting.
     * ARGV is ignored. */

    public static void main(String[] args){
        System.out.println("Hello, world!");
    }
}
```

java

Typora has better image management

It's not easy to integrate Markdown with image support, but Typora does provide stronger controls and from a first impression seems more robust: a lot of images in Boostnote are corrupted/lost to the void when files are saved, which makes it harder for me to include images in notes.

Typora is just nicer to work in

Boostnote uses a strange, dual-column view that feels cluttered at times, while Typora neatly integrates preview and typing in one window. It makes it easier for me to write nicer notes with real-time previews!

Typora has better LaTeX support

For some reason, Boostnote won't export LaTeX code sometimes, so I generally avoided using them. Using Typora will make it easier for me to include LaTeX code.

$$x = 2!$$

Typora has spell-check

Yeah, minor, but some of my old notes have typos from writing too fast. Typora actually has spell-check support, so expect more polished notes!