Introduction

Before jumping in, let's talk about what we're actually learning in this course.

WE'LL COVER THE FOLLOWING

Preface

CSS isn't what it used to be. Today, you'll be hard-pressed to find a strong, straightforward, and expressive UI language that can beat modern CSS.

As the tech world moves more and more towards web-based frameworks for UI, seen in the embrace of electronics for modern sleek looking desktop apps, we start to see a bigger emphasis on highly customized looks for apps. Everyone's racing to create that *dark theme*, right?

The goal of this course is two-fold:

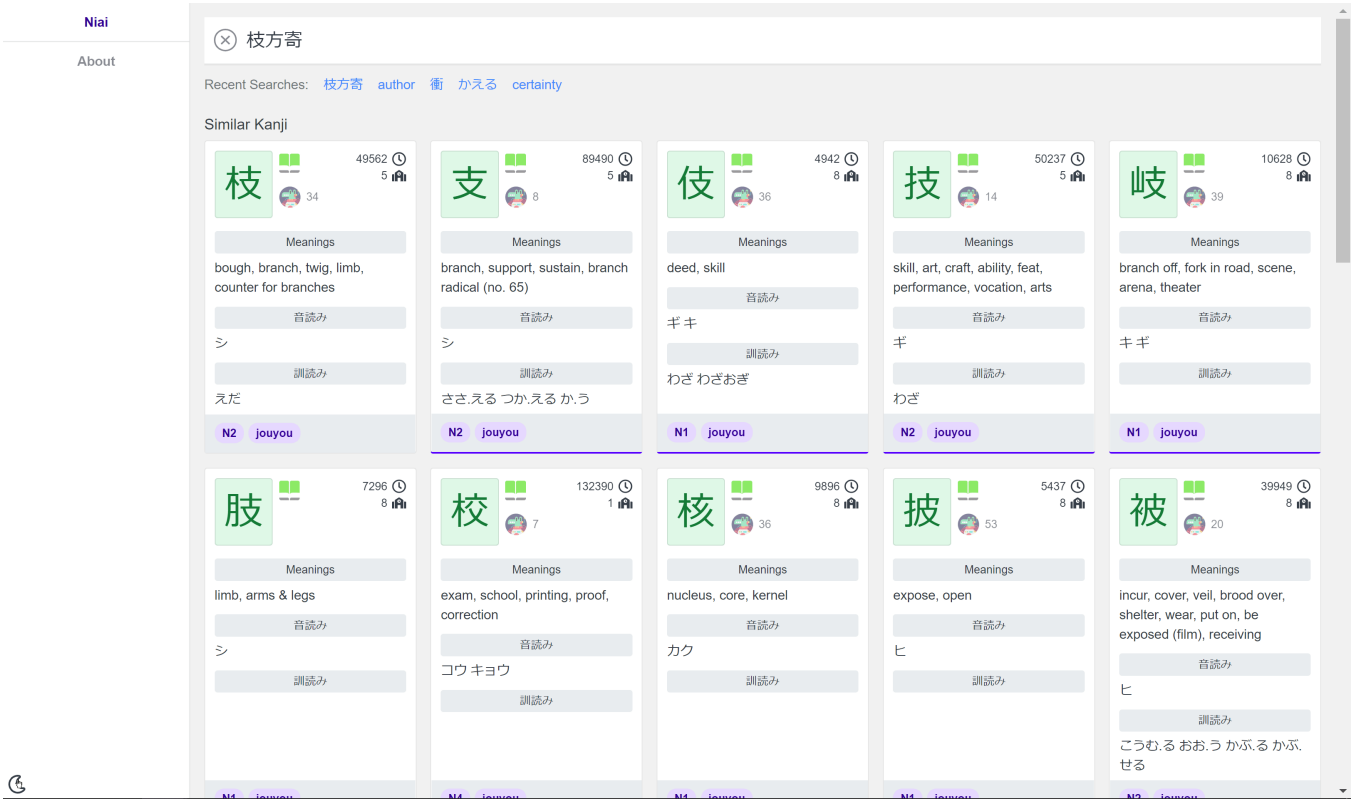
- Devise practical methods and patterns around theming and outline some common problems and their solutions.
- Learn how to use `css-theming`. We wouldn't want to repeat the theming plumbing each time we build a new app, so `css-theming` is a publicly distributed package that implements all patterns discussed.

While discussing problems and solutions in the first part, we'll be building our *theoretical* system, the system that is implemented in `css-theming`.

As such, this course is not a documentation of CSS features! It's rather the practical results of my experience over years developing highly customized complex UIs using CSS. I will express some opinionated views, but you can use

the knowledge obtained to develop any variation if you so desire.

Some of our discussions will be around colors, as this is the most important part of a theme. Something I want to emphasize here is that developing a consistent system that can also understand different **brightness** levels is hard and requires intricate planning. For example, Slack has had support for themes since the start but only introduced a true *dark* theme not long ago. There's an inherit complexity in designing a well thought out *dark* theme as we'll see.



from niai.mrahhal.net

But a *dark* theme is all the rage today and it's being embraced more and more as time goes on, so our system will definitely have to support this. In addition, the system we create must make it easy to reason about what color/shade/swatch to use when and where.

Finally, we're going to use SCSS on top of CSS to make our life a bit easier. SCSS is a superset of CSS that gives us more control than CSS. It won't be an essential piece here, and you're welcome to use LESS or any other framework, but `css-theming` is implemented using SCSS.

GitHub

This course comes with the css-theming github repo. You'll use the `css-theming` package to access the SCSS and JS/TS tools in your project.

`theming` package to access the SCSS and JS/TS tools in your project.

How we'll do this

I like to hammer out the theory before implementation. In the first part, we'll learn multiple different patterns and solve problems around theming. The second part is Using css-theming; this will act as a full guide to using and customizing `css-theming` to suit every app's needs.

Now that you're familiar with what we'll be learning throughout this course, let's dig deep into the main concepts for mastering CSS theming.