This repo contains the code challenge for the second day of preprocessing | Edit

Add topics

⊙ **16** commits	⑂ **2** branches	⬡ **0** releases	👥 **2** contributors

---

Branch: master ⏷ | New pull request | Create new file | Upload files | Find file | Clone or download ⏷

This branch is even with LambdaSchool:master.

Clone with HTTPS ⓘ	Use SSH

Use Git or checkout with SVN using the web URL.

`https://github.com/ErikKimsey/Preproc` 📋

| Open in Desktop | Download ZIP |

🖼 **mixelpixel** Merge pull request #229 from LambdaSchool/mixelpixel-patch-1 ...

📁 css	Updates for Project 2	
📁 img	Updated preprocessor image	
📁 less	Update to contact padding	3 months ago
📄 .DS_Store	Updates for Project 2	4 months ago
📄 README.md	Stretch Tasks header	2 months ago
📄 index.html	Added the viewport meta tag	3 months ago

📖 **README.md**

Preprocessing II - Advanced LESS

At this point, you have been introduced to LESS variables, nesting, mixins, escaping, importing and more. This project incorporates a small amount of everything. You will need to be able to navigate the file structure and make the following updates:

Project 2 - Organize imports, use mixins, functions, escaping, namespacing by building and updating a single web page.

This project will be broken into two main tasks:

1. Use several imports to build a single index.less file.
2. Demonstrate that you can use mixins, escaping, functions and stretching to see if you can use namespaces.

Start by forking and cloning this repo then get started on the tasks. Good luck!

Task 1 - Importing

Notice that the LESS folder in this project has several .less files now. Task 1 is to correctly import the .less files into index.less in the following order:

1. reset.less
2. variables.less
3. mixins.less
4. general-styles.less
5. header.less
6. cta.less
7. main-content.less
8. contact.less

9. footer.less

Once you have imported your less files, be sure to run less-watch-compiler

Task 2 - Demonstrate your skills using Mixins, escaping and functions by doing the following:

1. Use all 5 mixins from the mixins.less file in anyway you want throughout the site (Hint, google the CSS properties if you don't know what they do!).

2. Use escaping to create 2 unique media query variables named "tablet" at a max-width of 768px and "phone" at 400px max-width. Now use your variables in combination with @media to introduce responsive styling to your page. (You can use code from your last project for the responsivness)

3. Create your own custom button mixin that passes values for width, height, and background color. Once you have created this mixin, introduce a button to the bottom of each `.text-content` div. Create unique text for each button to match the header. Example: The heading "Features" could have a button that reads "See Features." Be creative!

4. Use a function to darken the background by 20% on hover for all those new buttons you just made.

Stretch Tasks

Create your own namespace for the following elements:

- Headings
- Buttons
- p tags

Once you have created your own unique namespace styles, apply them throughout the site using the > accessor to overrite the current styles in place as you see fit. This is open ended so try and find practical ways this could help you or just go wild and change everything ;)