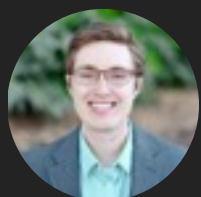


FIGHT
THE **ZOMBIE**



PATTERN LIBRARY

Responsive Web Design Summit • 03.29.16



Marcelo Somers

Solution Architect, User Experience
[@marcelosomers](#)

“

Front-end style guides are a modular collection of all the elements in your product's user interface, together with code snippets for developers to copy and paste as needed to implement those elements. They include common UI components like buttons, form-input elements, navigation menus, modal overlays, and icons.

– Lean UX



ZOMBIE

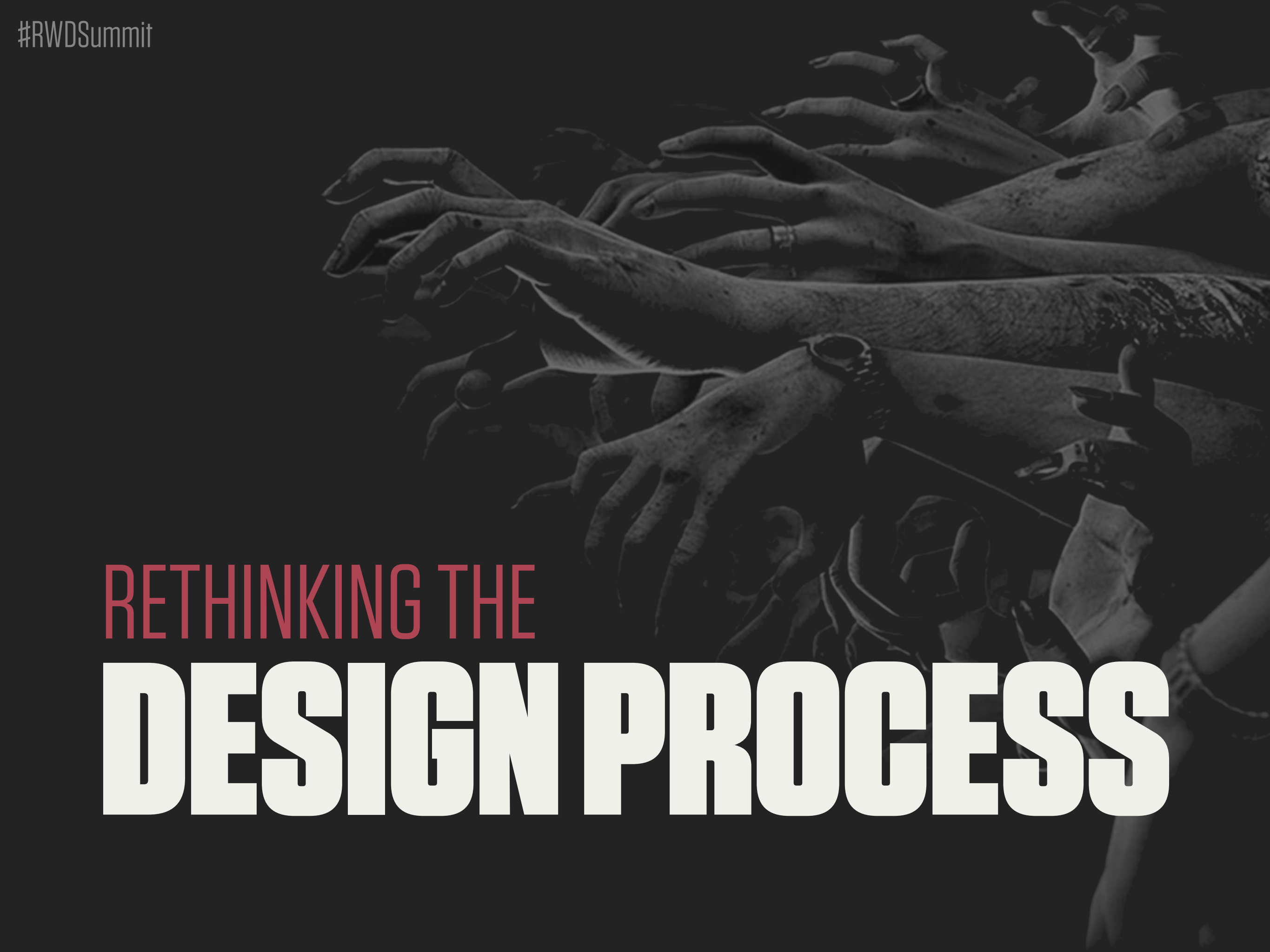
PATTERN LIBRARIES



building interfaces

“How do you keep ~~living~~,
**knowing that's what
the world is like?**”

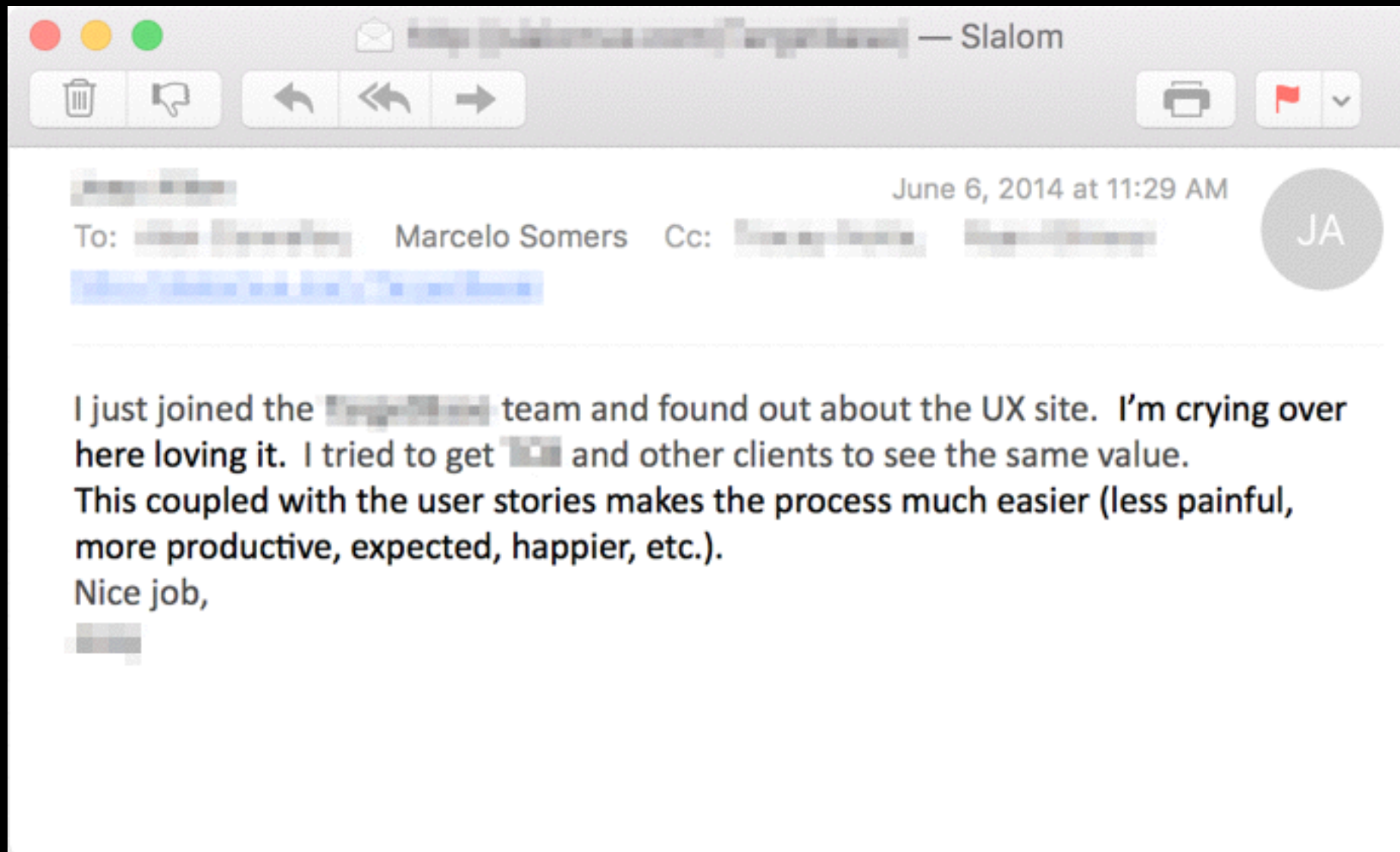
#RWDSummit



RETHINKING THE DESIGN PROCESS

BUILD
SYSTEMS
NOT
PAGES

-Andy Clarke



BUILD A **TINY** **BOOTSTRAP**

– Mark Otto & Dave Rupert

DESIGNING WITH PATTERNS

DESIGN

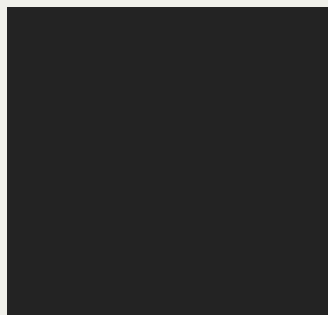
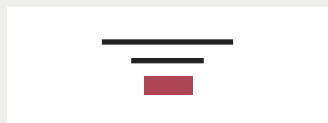


DESIGNING WITH PATTERNS

DESIGN



SLICE

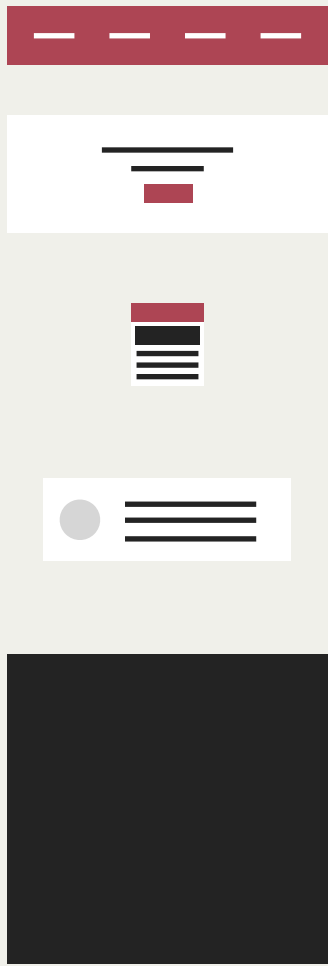


DESIGNING WITH PATTERNS

DESIGN



SLICE



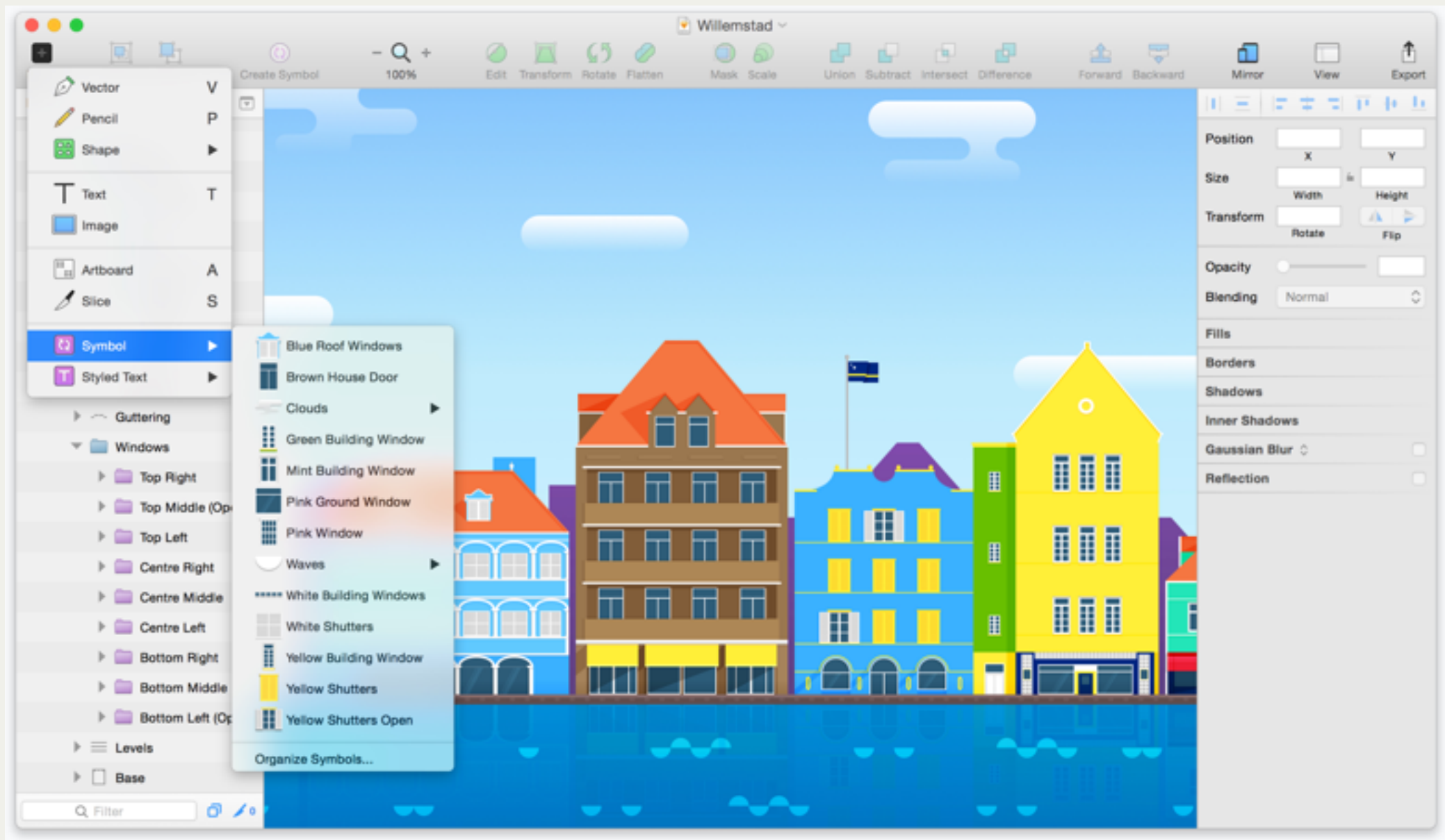
ASSEMBLE



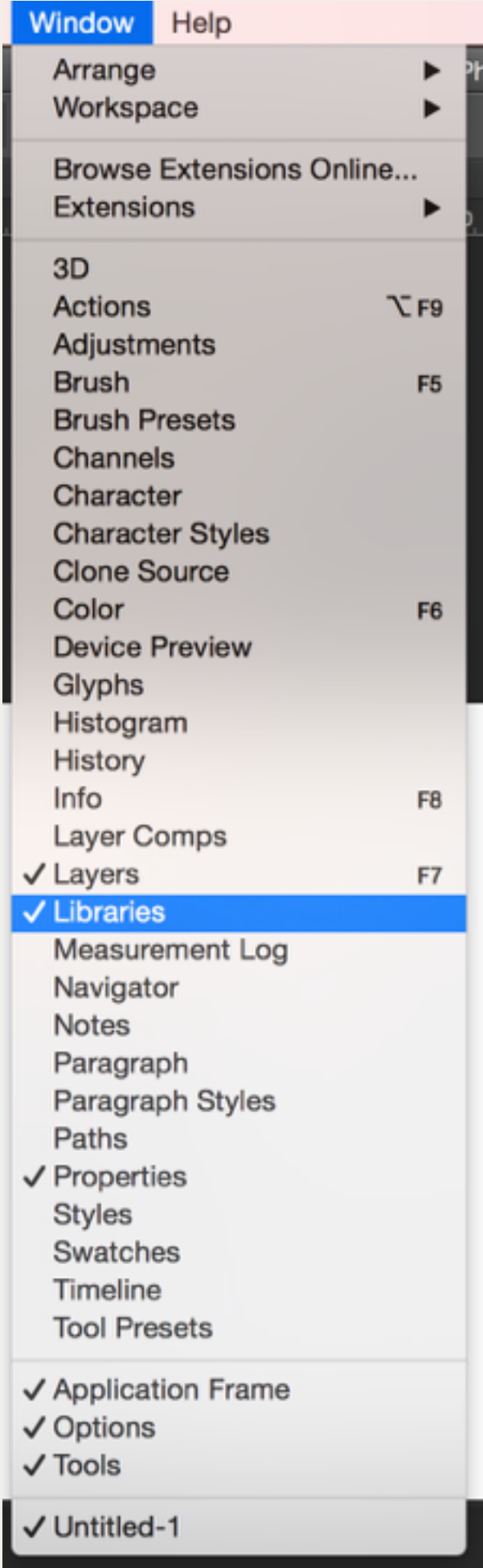
DESIGN TOOLS



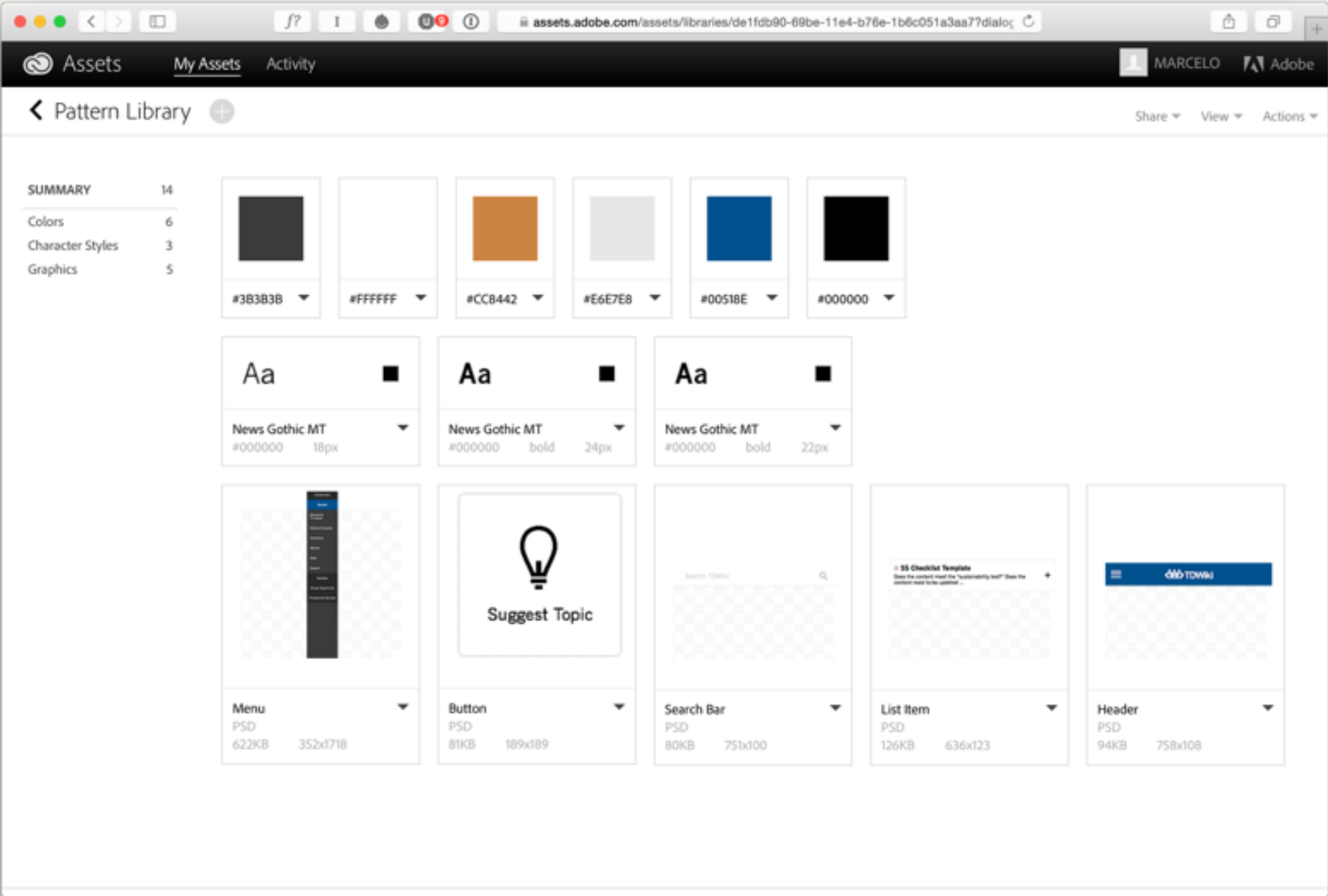






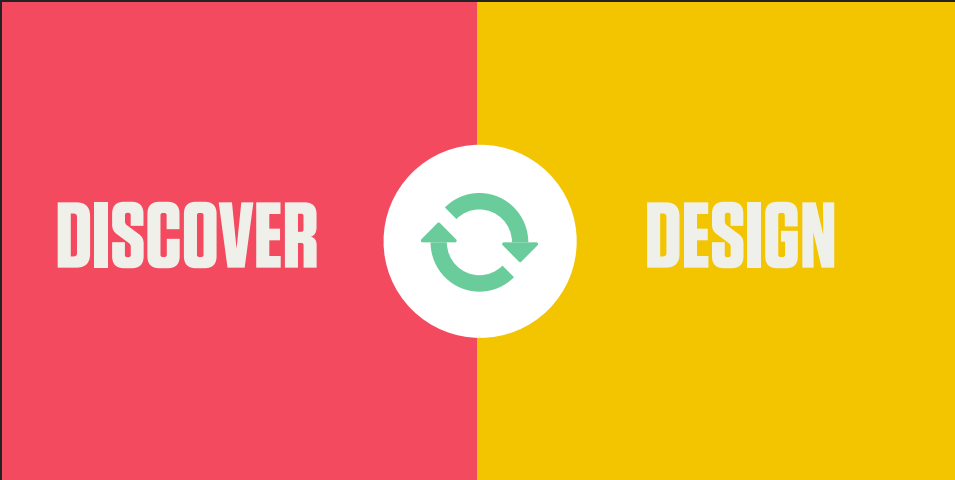




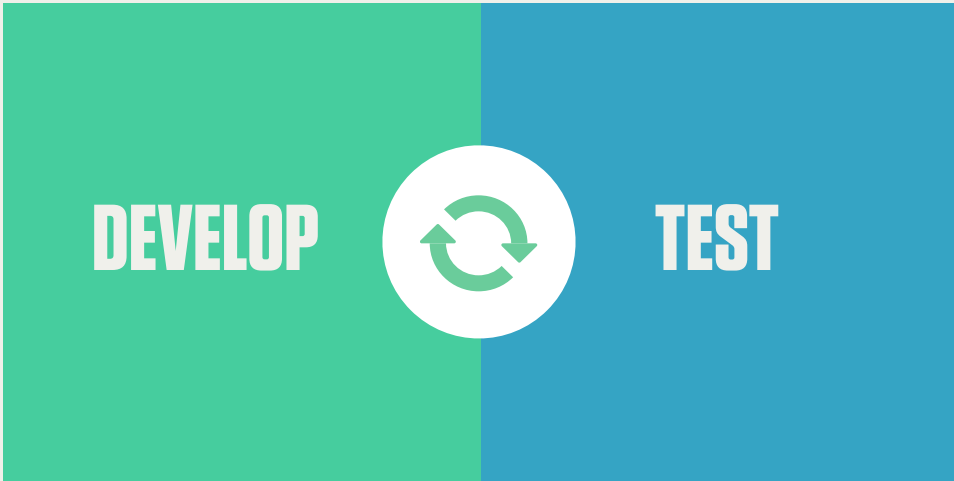


WHERE THE DESIGN PROCESS
BREAKS DOWN

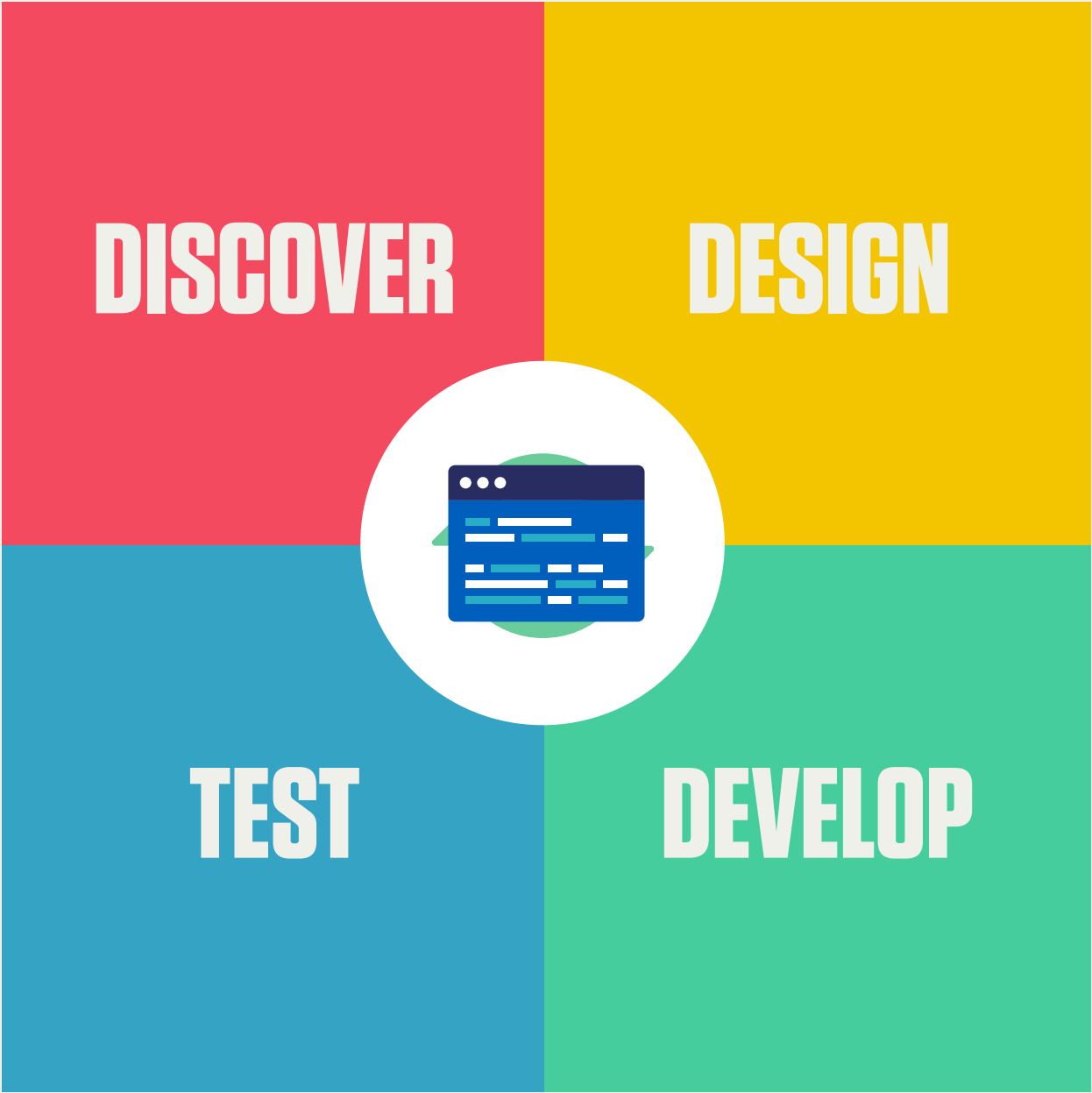
DESIGNERS



DEVELOPERS



TRADITIONAL HANDOFF IS
BROKEN

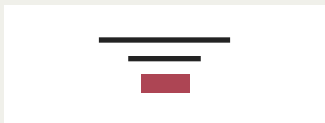


DESIGNING WITH PATTERNS

DESIGN



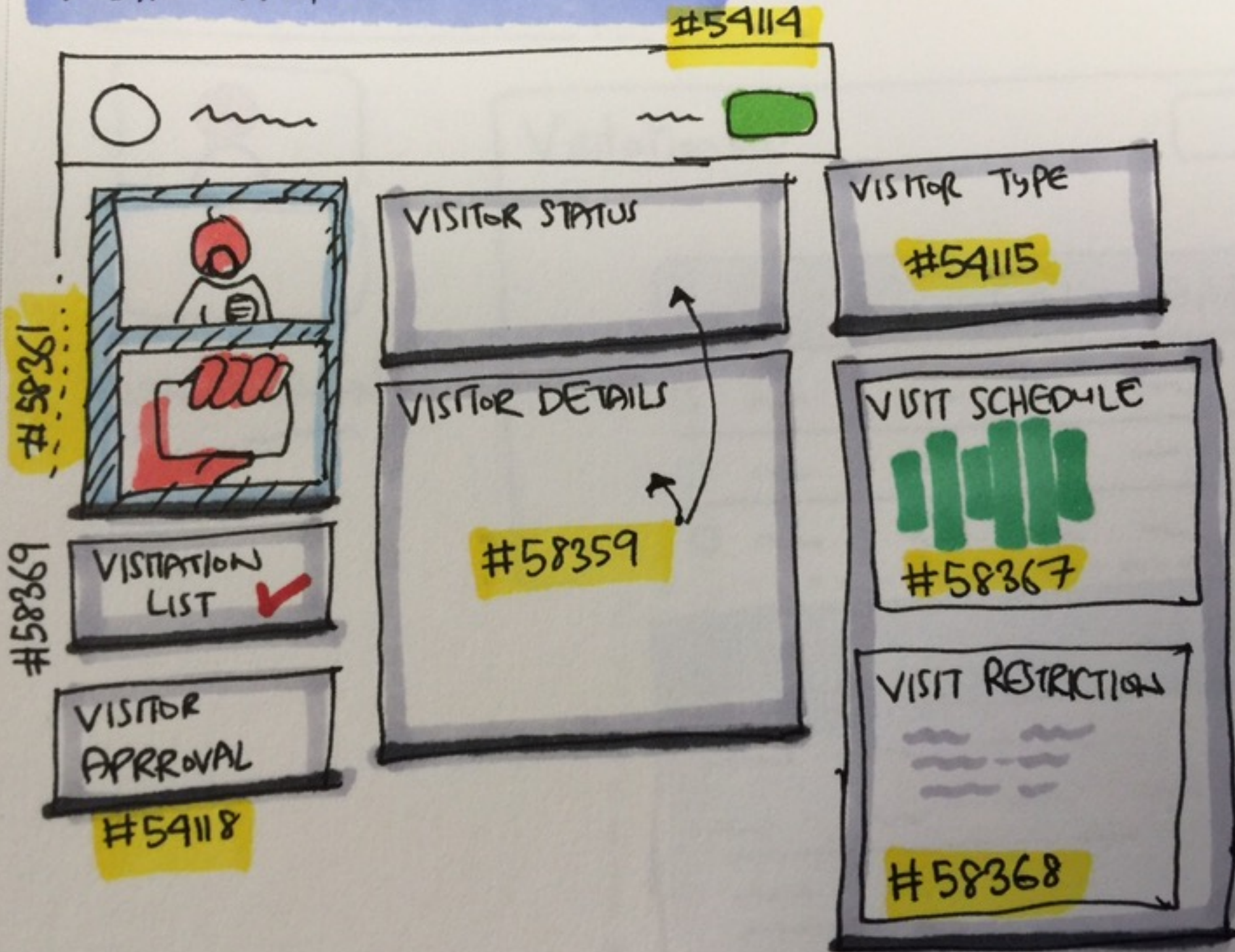
PATTERN LIBRARY



ASSEMBLE

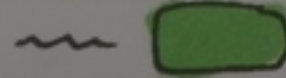
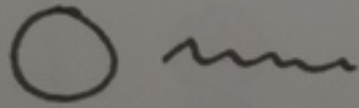


Visitor Profile



Visitor Profile

#54114



#58359



#58359

VISITATION
LIST ✓

VISITOR
APPROVAL

#54118

VISITOR STATUS

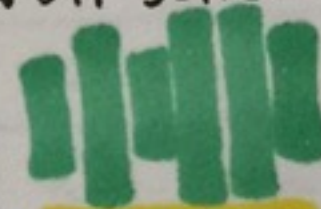
VISITOR DETAILS

#58359

VISITOR TYPE

#54115

VISIT SCHEDULE



#58367

VISIT RESTRICTIONS

#58368

Details Panel

slalomux.com/patternpack/projects/prison-telecommunications/blocks/details-panel.html

☆

I

ⓘ

🔒

🔒

f?

🔄

🔄

🔄

🔄

Style Guide ▾

Blocks ▾

Templates ▾

DETAILS PANEL

Example of the tab navigation detail panel.

Example

PIN

Call Settings

PIN Settings

PIN Number Required

Ex: 328491239

Min. Length 4; Max. 20

PIN Generation

Generate

PIN append length

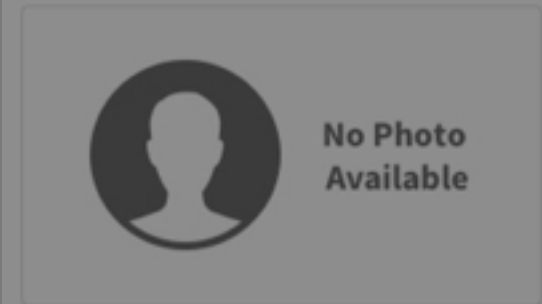
Append

Code

```

<ul class="nav nav-tabs call-details-tabs" style="width:300px">
  <li class="active"><a href="#pin-settings" data-toggle="tab">PIN</a></li>
  <li class="notes-tab"><a href="#call-settings" data-toggle="tab">Call Settings</a></li>
</ul>
<div class="tab-content" style="width:400px">
  <div role="panel" class="tab-pane active" id="pin-settings">
    <div class="bff-include">
      <div class="well securus-panel-style securus-panel-padding">
        <h4 class="settings-headline">
          <a data-toggle="collapse" href="#inmate-pin-details">
            ADD TITLE
          <div class="pull-right">
            <span class="toggle-hide"></span>
            <span class="toggle-show"></span>
          </div>
        </h4>
      </div>
    </div>
  </div>

```



Visitation Blocked Allowed List: **6 active entries**
last updated 10-27-2014

Visitor Approval

✖ Reject

✔ Approve

Visitor Status ⓘ

✔ Active

Visitor Details

First name:	Walter
Last name:	White
Username (Email):	wwhitejr@email.com
Phone:	555-555-5555
Address:	1500 Main Street, #600
City:	Dallas
State:	TX
Zip:	75040
Country:	USA

Visitor Type Schedule Restrictions

Visitor Type

Attorney

Edit Visitor Type

Attorney Information

License Number	Expiration Date
TX-220933847	12/05/2015
Name of Firm	
Keller & Keller LLC	

Cancel

Save

ELIMINATE

WASTE

THE EVOLVING DESIGNER

“I can shift the bulk of my time and energy to looking at the bigger picture. I can think more widely about our users' journey through our product and give attention to more than baseline functionality.

– Katey Basye (Salesforce UX)

WHERE DOES THE PATTERN LIBRARY

LIVE

IN YOUR PROCESS?

STYLE GUIDE MATURITY MODEL



Governed

The pattern library process is built in to your organization



Automatic

The pattern library is a part of your app build process

This is the “chasm” that is hard for organizations to cross



Manual

The pattern library has code, but must be manually updated



Static

A one-time PDF of your brand guidelines



Inconsistent

The absence of a design system

Speaker Deck

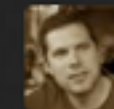
Published on Sep 22, 2015

UNLESS IT'S PART OF YOUR BUILD,
YOUR STYLEGUIDE IS JUST MORE
DOCUMENTATION TO MAINTAIN

@philhawsworth



share



Phil Hawksw...

22 Presentations



Star this Talk

11 Stars



Published in

Programming



Stats

888 Views

Share



Twitter, Facebook



Embed



Direct Link



Download PDF

Static sites go all Hollywood by Phil Hawksworth

Published September 22, 2015 in Programming

The popularity of building web sites with static site generators is on the rise. Their reduced complexity, easier compliance, cheaper hosting, and other benefits are getting people's attention, but they do have limits.

This talk will explore how we can break through some of those limits with the use of a new breed of

f? I A Medium Corporation medium.com/@marcelosomers/chasing-the-holy-grail-bbc0b7cce365

Edit Search Medium 13

Marcelo Somers

Dec 3, 2015 · 8 min read

Recommended by Jason Cypret, Sherif Tariq, and 10 others

Chasing the Holy Grail

Strategies For Distributing Your Pattern Library and Keeping It in Sync



In our work with Pattern Libraries, we strive to achieve a level of maturity where the Pattern Library documentation is automatically in sync with the app(s) that it powers. Generally, this means sharing the CSS code so developers implementing patterns can just copy and paste snippets of HTML so it “just works.”

<https://medium.com/@marcelosomers/chasing-the-holy-grail-bbc0b7cce365#wvedzdvsx>

#RWDSummit



GETTING STARTED

(TODAY)



TAKE AN INVENTORY

13

14

15

16

17

18

19

20

21

22

23

24

Messaging

Send On
+ Tap Here Search for filter

Send On
+ Tap Here Search for filter

Mobile List

Send On
+ Tap Here Search for filter

Send On
+ Tap Here Search for filter

Modal

Send On
+ Tap Here Search for filter

Option List Group

Send On
+ Tap Here Search for filter

Send On
+ Tap Here Search for filter

Progress Bar

Send On
+ Tap Here Search for filter

Send On
+ Tap Here Search for filter

Secondary Menu

Send On
+ Tap Here Search for filter

Send On
+ Tap Here Search for filter

Status Bar

Send On
+ Tap Here Search for filter

Send On
+ Tap Here Search for filter

Tiles

Send On
+ Tap Here Search for filter

Send On
+ Tap Here Search for filter

Transaction List

Send On
+ Tap Here Search for filter

Send On
+ Tap Here Search for filter

Transaction Totals

Send On
+ Tap Here Search for filter

Send On
+ Tap Here Search for filter

Table List

Send On
+ Tap Here Search for filter

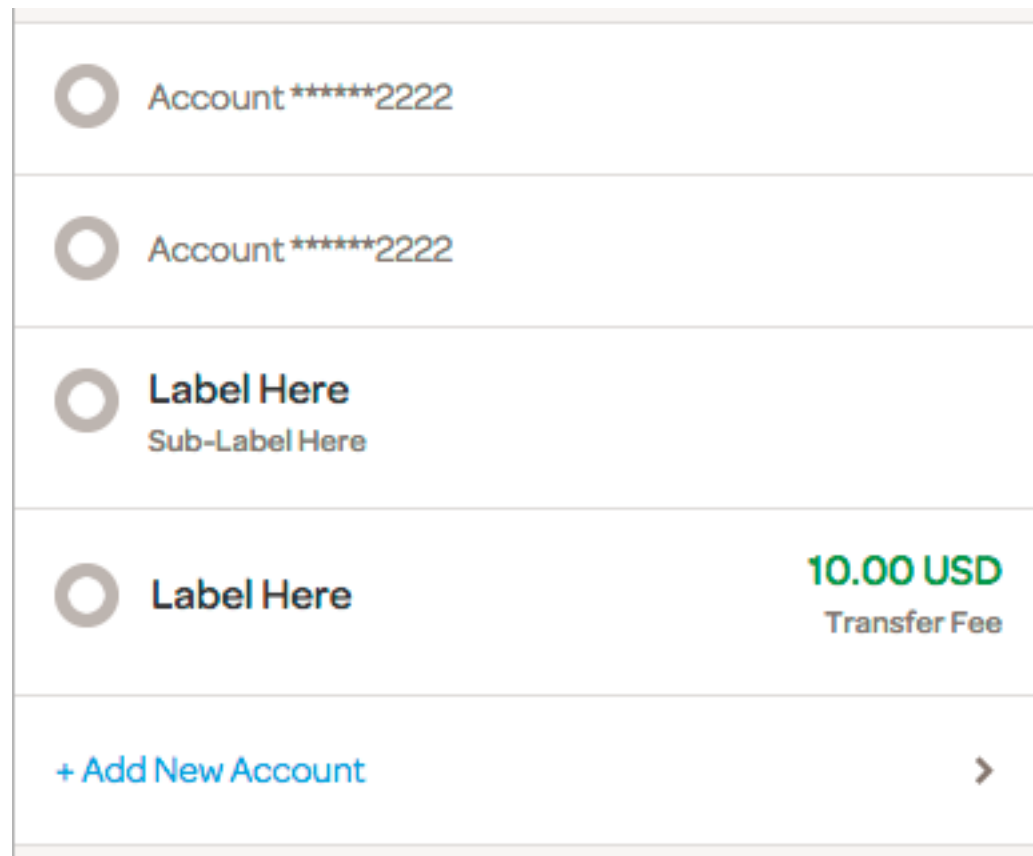
Send On
+ Tap Here Search for filter

Molecules

Desktop



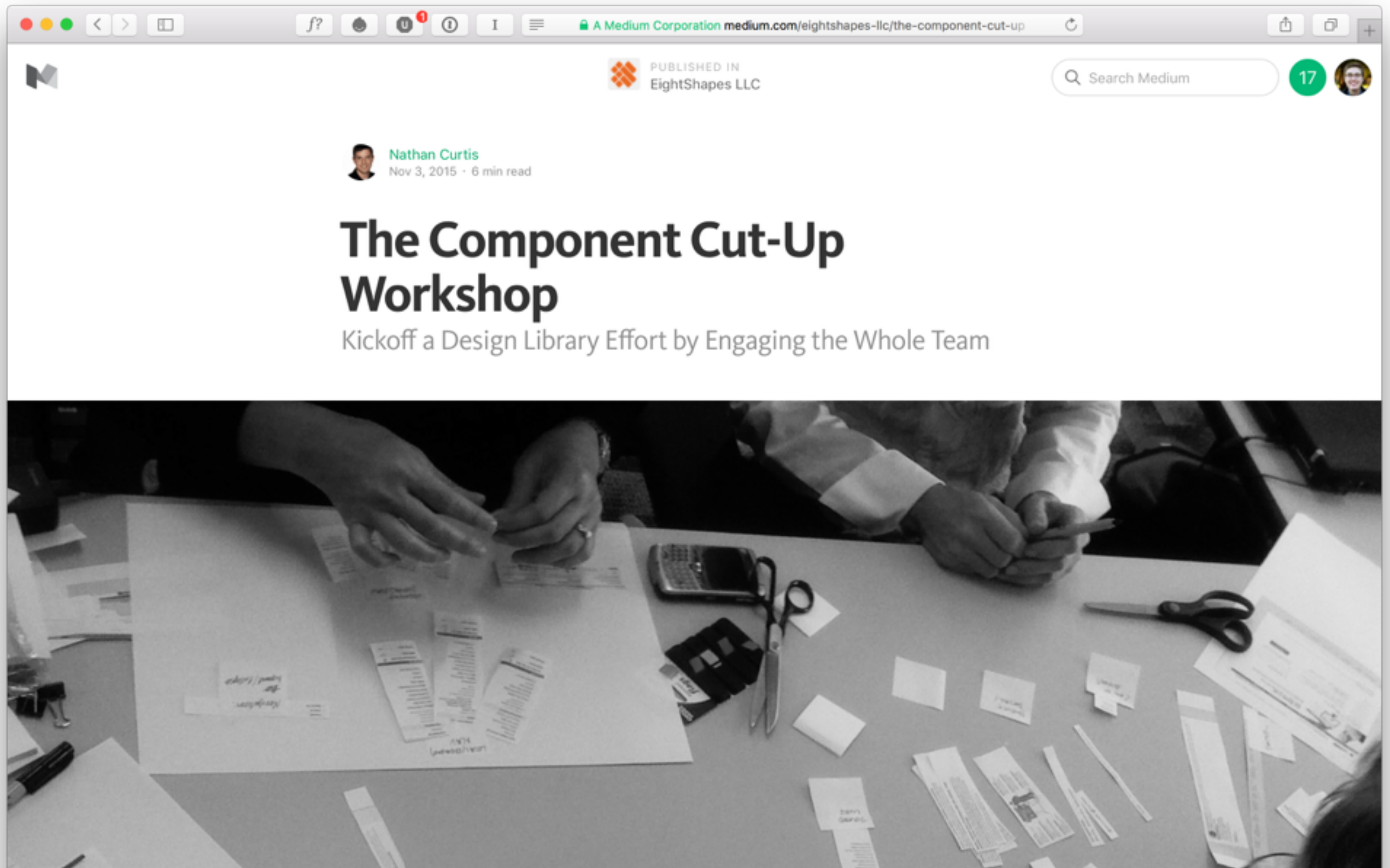
Mobile



Option List Group

Seen On:

- My Account » Account List
- Send Money » Select Payment Option



<https://medium.com/eightshapes-llc/the-component-cut-up-workshop-1378ae110517#.c887jlrum>

WHAT TO DOCUMENT

1

BASE STYLES

2

COMPONENTS

3

PAGE TEMPLATES

The screenshot shows a web browser window with the URL www.nngroup.com/articles/front-end-style-guides/. The page header features the Nielsen Norman Group logo and tagline, a search bar, and a navigation menu. The left sidebar contains sections for Topics, Author, and Recent Articles. The main content area displays the article title, author, date, topics, a summary, a definition, and an introduction. A component preview for 'Button Groups' is shown at the bottom.

NN/g Nielsen Norman Group
Evidence-Based User Experience Research, Training, and Consulting

HOME TRAINING CONSULTING REPORTS **ARTICLES** ABOUT NN/G

Topics
[E-commerce](#)
[Intranets](#)
[Mobile & Tablet](#)
[User Testing](#)
[Web Usability](#)
[See all topics...](#)

Author
[Jakob Nielsen](#)
[Don Norman](#)
[Bruce "Tog" Tognazzini](#)
[See all authors...](#)

Recent Articles
[Front-End Style-Guides: Definition, Requirements, Component Checklist](#)
[3 Ways to Inspire Intranet Content Authors](#)
[Games User Research: What's Different?](#)
[How Chunking Helps Content Processing](#)
[Microsoft Desirability Toolkit](#)
[Product Reaction Words](#)
[See all articles...](#)

Front-End Style-Guides: Definition, Requirements, Component Checklist

by [PAGE LAUBHEIMER](#) on March 27, 2016
Topics: [Agile](#) [Visual Design](#)

Summary: Front-end style-guides help efficient design and testing, and enforce UI consistency. We present 8 style-guide requirements and 25 common components.

What Is a Front-End Style Guide?

Front-end style guides have become an increasingly commonplace deliverable in UX practice. As described by Jeff Gothelf and Josh Seiden in their book *Lean UX*, they originated in [Agile or Lean environments](#).

Definition: Front-end style guides are a modular collection of all the elements in your product's user interface, together with code snippets for developers to copy and paste as needed to implement those elements. They include common UI components like buttons, form-input elements, navigation menus, modal overlays, and icons.

A front-end style guide is both a **deliverable** created by the UX team (in concert with the engineering team, typically) and a **tool** used by the entire team for maintaining consistent, nimble product design in a modular format.

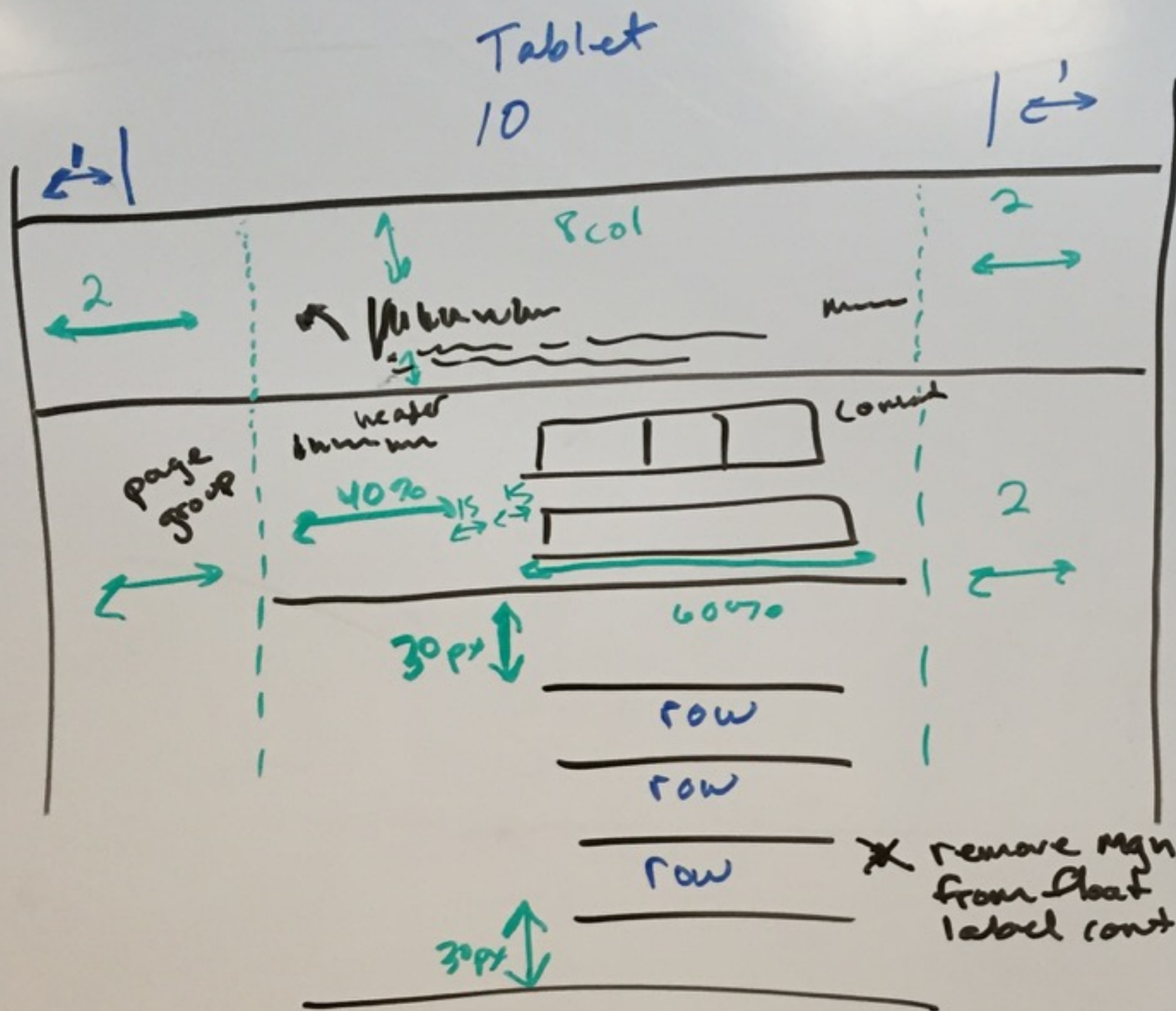
Front-end style guides are distinct from design pattern libraries, which are a long-standing tool used by UX practitioners to define broad design ideas, rather than specific implementation details. Unfortunately, [some proponents](#) of front-end style guides refer to them as *pattern libraries*, which has caused some confusion in the UX community.

Components
Button Groups
[Component Overview](#)

<https://www.nngroup.com/articles/front-end-style-guides/>



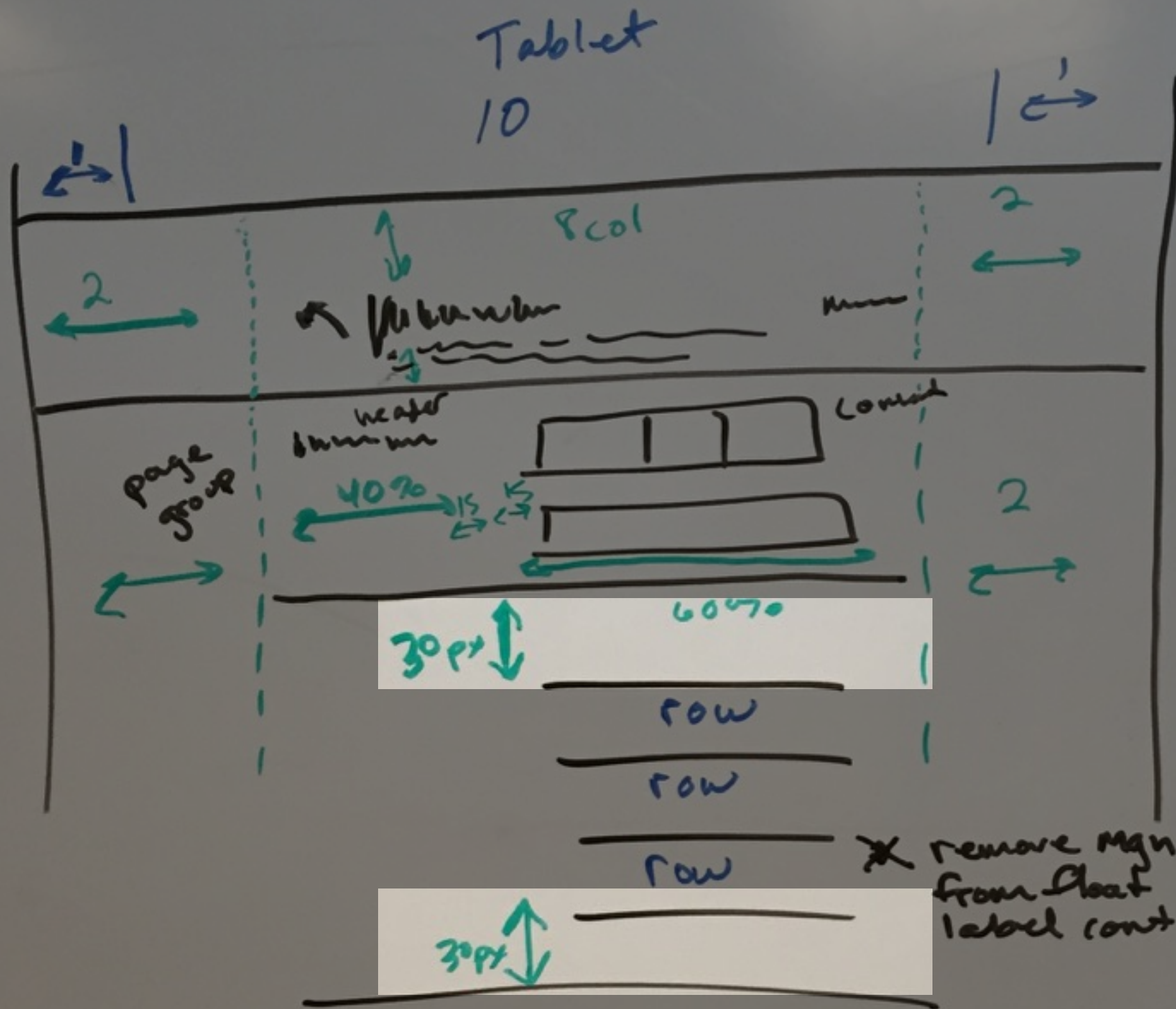
STANDARDIZE



Core-
page-
header

Core-
page-
content

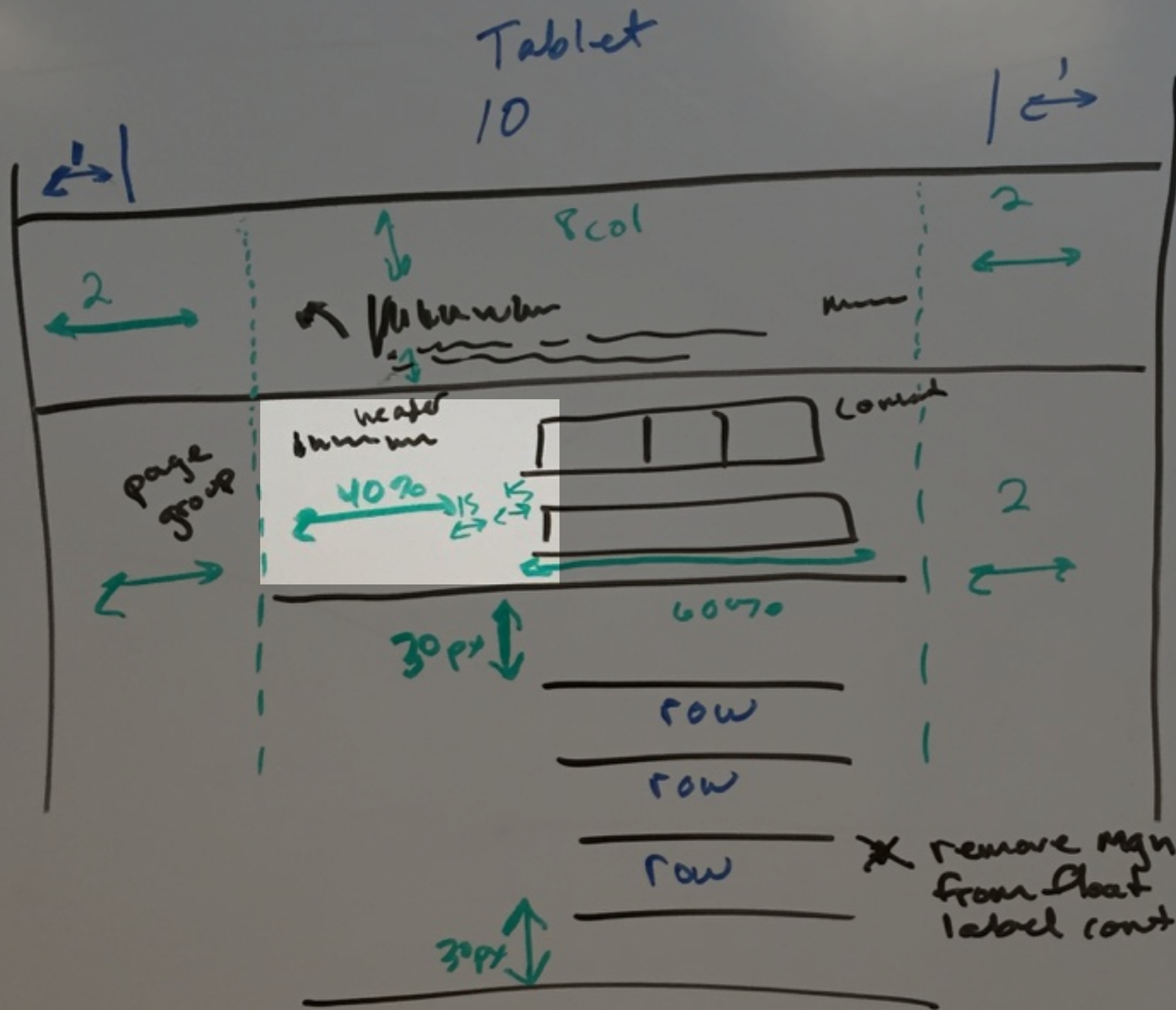
Core-
page-
actions



Core-
page-
header

Core-
page-
content

Core-
page-
actions



Core-
page-
header

Core-
page-
content

Core-
page-
actions

Tablet
10

10

8 col

Header

Header

page group

Header

Content

40%

30px

60%

row

row

row

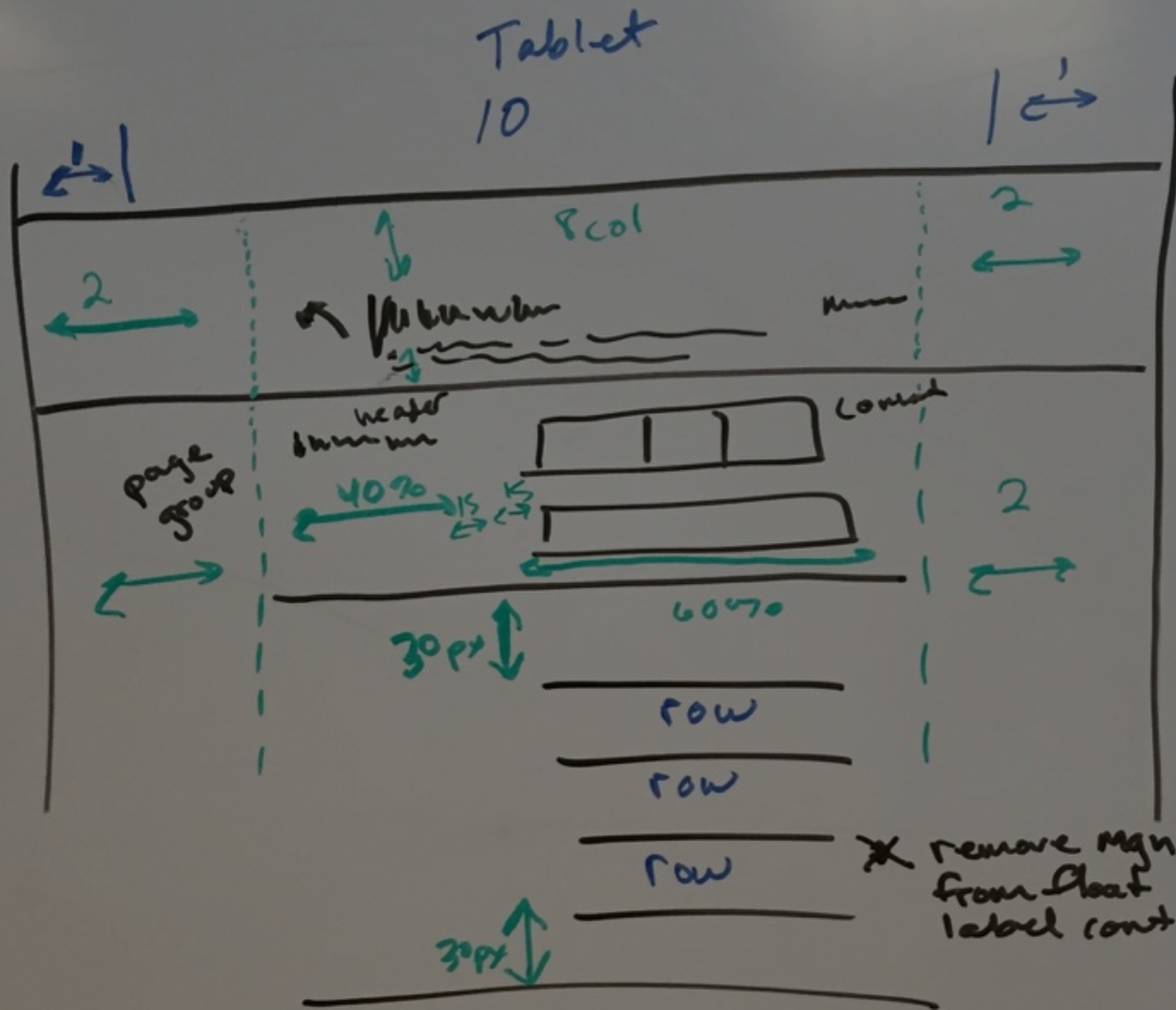
30px

remove mgn from float label cont

Core-
Page-
header

Core-
Page-
Content

Core-
Page-
actions



Core-
page-
header

Core-
page-
content

Core-
page-
actions

3 DOCUMENT

salesforce

design system

Getting Started

Design

Components

- Activity Timeline
- Badges
- Breadcrumbs
- Buttons
- Button Groups
- Cards
- Data Tables
- Datepickers
- Dropdowns
- Forms
- Grid System
- Icons
- Images
- Lookups
- Media Objects
- Modals
- Notifications
- Page Headers
- Picklists
- Pills

Components

Buttons

Create a button with a **button** or **a** element to retain the native click function. Use a disabled attribute when a button can't be clicked.

ACCESSIBILITY

If an icon button doesn't include a label, use a span with **.slds-assistive-text** to describe the icon for screen readers.

Base

DEV READY

The base **.slds-button** looks like a plain text link. It removes all the styling of the native button. It's typically used to trigger a modal or display a "like" link. All button variations are built by adding another class to **.slds-button**.

LARGE

Button

HTML

SCSS

```
<button class="slds-button">Button</button>
```

VARIANTS

- Base
- Neutral
- Neutral Icon
- Brand
- Inverse
- Stateful
- Stateful Inverse
- Icon
- Icon Sizing
- Icon More
- Icon Stateful
- Icon Inverse
- Hint

Component Overview

salesforce design system

Getting Started

Design

Components

- Activity Timeline
- Badges
- Breadcrumbs
- Buttons
- Button Groups
- Cards
- Data Tables
- Datepickers
- Dropdowns
- Forms
- Grid System
- Icons
- Images
- Lookups
- Media Objects
- Modals
- Notifications
- Page Headers
- Picklists
- Pills

Buttons

Create a button with a `button` or `a` element to retain the native click function. Use a disabled attribute when a button can't be clicked.

ACCESSIBILITY

If an icon button doesn't include a label, use a span with `.slds-assistive-text` to describe the icon for screen readers.

Base DEV READY

The base `.slds-button` looks like a plain text link. It removes all the styling of the native button. It's typically used to trigger a modal or display a "like" link. All button variations are built by adding another class to `.slds-button`.

🖥️ LARGE

Button

HTML

SCSS

```
<button class="slds-button">Button</button>
```

VARIANTS

Base

Neutral

Neutral Icon

Brand

Inverse

Stateful

Stateful Inverse

Icon

Icon Sizing

Icon More

Icon Stateful

Icon Inverse

Hint

Component Overview

salesforce

design system

Getting Started

Design

Components

Buttons

Create a button with a **button** or **a** element to retain the native click function. Use a disabled attribute when a button can't be clicked.

Buttons

Button Groups

Cards

Data Tables

Datepickers

Dropdowns

Forms

Grid System

Icons

Images

Lookups

Media Objects

Modals

Notifications

Page Headers

Picklists

Pills

ACCESSIBILITY

If an icon button doesn't include a label, use a span with `.slds-assistive-text` to describe the icon for screen readers.

Base

DEV READY

The base `.slds-button` looks like a plain text link. It removes all the styling of the native button. It's typically used to trigger a modal or display a "like" link. All button variations are built by adding another class to `.slds-button`.

LARGE

Button

HTML

SCSS

`<button class="slds-button">Button</button>`

Neutral Icon

Brand

Inverse

Stateful

Stateful Inverse

Icon

Icon Sizing

Icon More

Icon Stateful

Icon Inverse

Hint

Component Overview

salesforce design system

Getting Started

Design

Components

Activity Timeline

Badges

Breadcrumbs

Buttons

Components

Buttons

Create a button with a **button** or **a** element to retain the native click function. Use a disabled attribute when a button can't be clicked.

ACCESSIBILITY

VARIANTS

Base

Neutral

Neutral Icon

Neutral

DEV READY

Add the `.slds-button--neutral` class to create a neutral button, which has a white background and gray border.

🖥️ LARGE

Button neutral

Anchor Button Neutral

Disabled

Button Small

```
<button class="slds-button">Button</button>
```

salesforce design system

Getting Started

Design

Components

- Activity Timeline
- Badges
- Breadcrumbs
- Buttons
- Button Groups
- Cards
- Data Tables
- Datepickers
- Dropdowns
- Forms
- Grid System
- Icons
- Images

Components

Buttons

Create a button with a **button** or **a** element to retain the native click function. Use a disabled attribute when a button can't be clicked.

ACCESSIBILITY

If an icon button doesn't include a label, use a span with **.slds-assistive-text** to describe the icon for screen readers.

Base DEV READY

The base **.slds-button** looks like a plain text link. It removes all the styling of the native button. It's typically used to trigger a modal or display a "like" link. All button variations are built by adding another class to **.slds-button**.

LARGE

Button

VARIANTS

Base

Neutral

Neutral Icon

Brand

Inverse

Stateful

Stateful Inverse

Icon

Icon Sizing

Icon More

Icon Stateful

Icon Inverse

HTML

SCSS

```
<button class="slds-button">Button</button>
```

salesforce design system

Getting Started

Design

Components

Activity Timeline

Badges

Datepickers

Dropdowns

Forms

Grid System

Icons

Images

Lookups

Media Objects

Modals

Notifications

Page Headers

Picklists

Pills

Components

Buttons

Create a button with a **button** or **a** element to retain the native click function. Use a disabled attribute when a button can't be clicked.

VARIANTS

Base

Base

DEV READY

The base **.slds-button** looks like a plain text link. It removes all the styling of the native button. It's typically used to trigger a modal or display a "like" link. All button variations are built by adding another class to **.slds-button**.

LARGE

Button

HTML

SCSS

```
<button class="slds-button">Button</button>
```

Stateful Inverse

Icon

Icon Sizing

Icon More

Icon Stateful

Icon Inverse

Hint

Component Overview

ACCESSIBILITY

If an icon button doesn't include a label, use a span with **.slds-assistive-text** to describe the icon for screen readers.

salesforce design system

Getting Started

Design

Components

- Activity Timeline
- Badges
- Breadcrumbs
- Buttons
- Button Groups
- Cards
- Data Tables
- Datepickers
- Dropdowns
- Forms
- Grid System
- Icons
- Images
- Lookups
- Media Objects
- Modals
- Notifications
- Page Headers
- Picklists
- Pills

Components

Buttons

Create a button with a **button** or **a** element to retain the native click function. Use a **disabled** attribute when a button can't be clicked.

ACCESSIBILITY

If an icon button doesn't include a label, use a span with **.slds-assistive-text** to describe the icon for screen readers.

Base DEV READY

The base **.slds-button** looks like a plain text link. It removes all the styling of the native button. It's typically used to trigger a modal or display a "like" link. All button variations are built by adding another class to **.slds-button**.

LARGE

Button

HTML SCSS

```
<button class="slds-button">Button</button>
```

VARIANTS

Base

Neutral

Neutral Icon

Brand

Inverse

Stateful

Stateful Inverse

Icon

Icon Sizing

Icon More

Icon Stateful

Icon Inverse

Hint

Component Overview

**BASIC
DOCUMENTATION**



NAME



DESCRIPTION



EXAMPLE



CODE SNIPPET



CENTRALIZE YOUR CSS

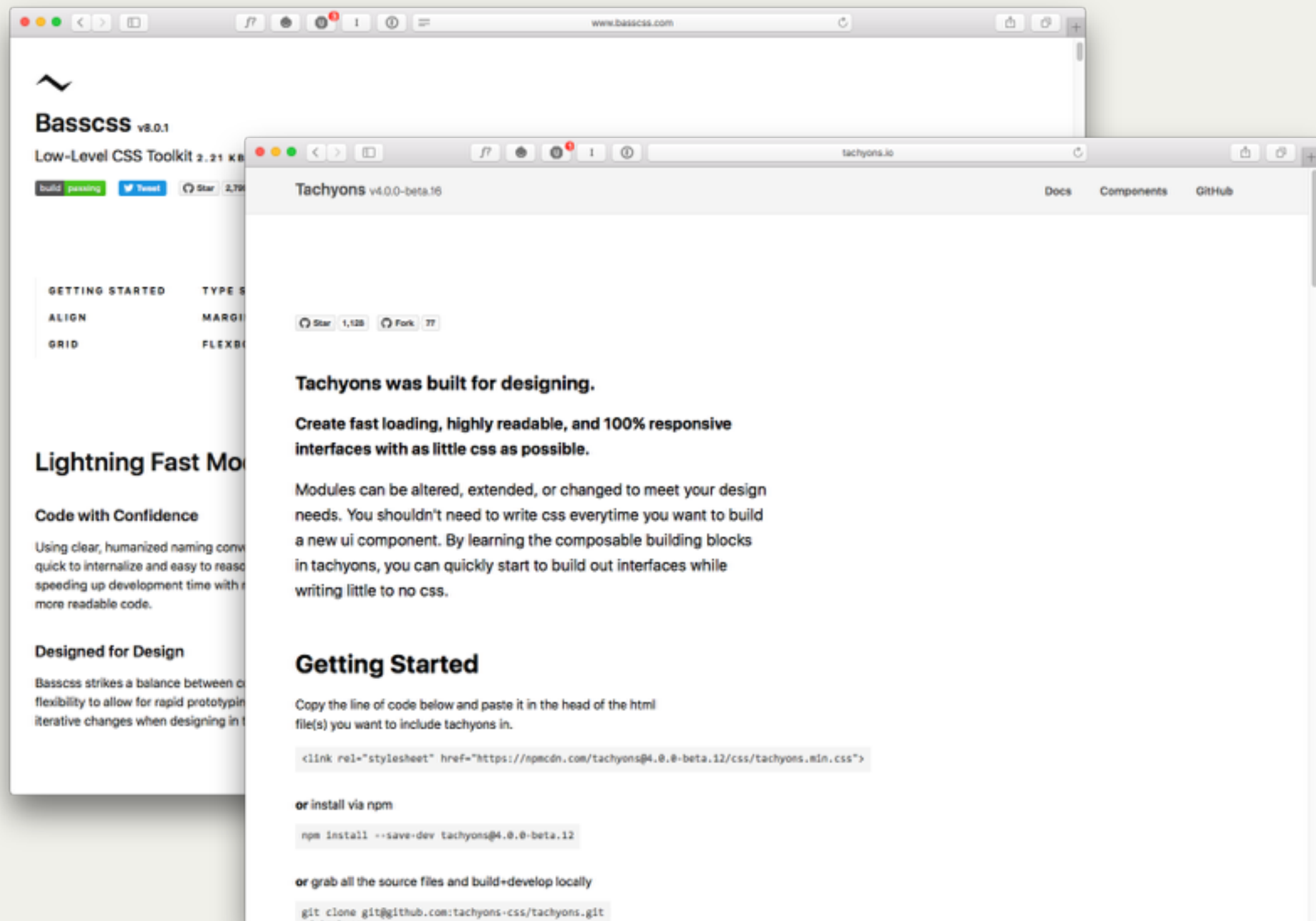
4

DEFINE CSS STANDARDS

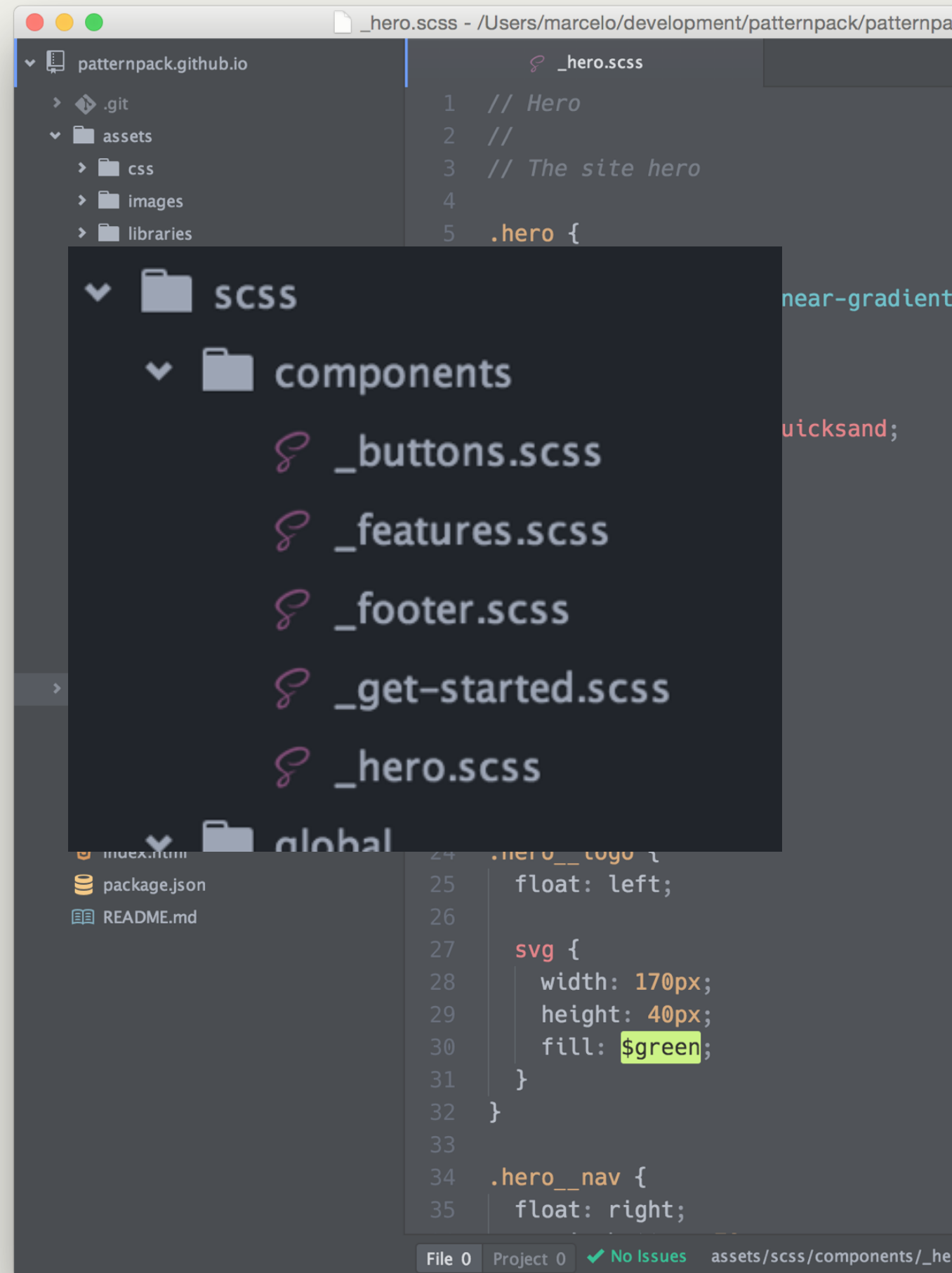
5

REFACTOR TO PERFECTION

START FUNCTIONAL



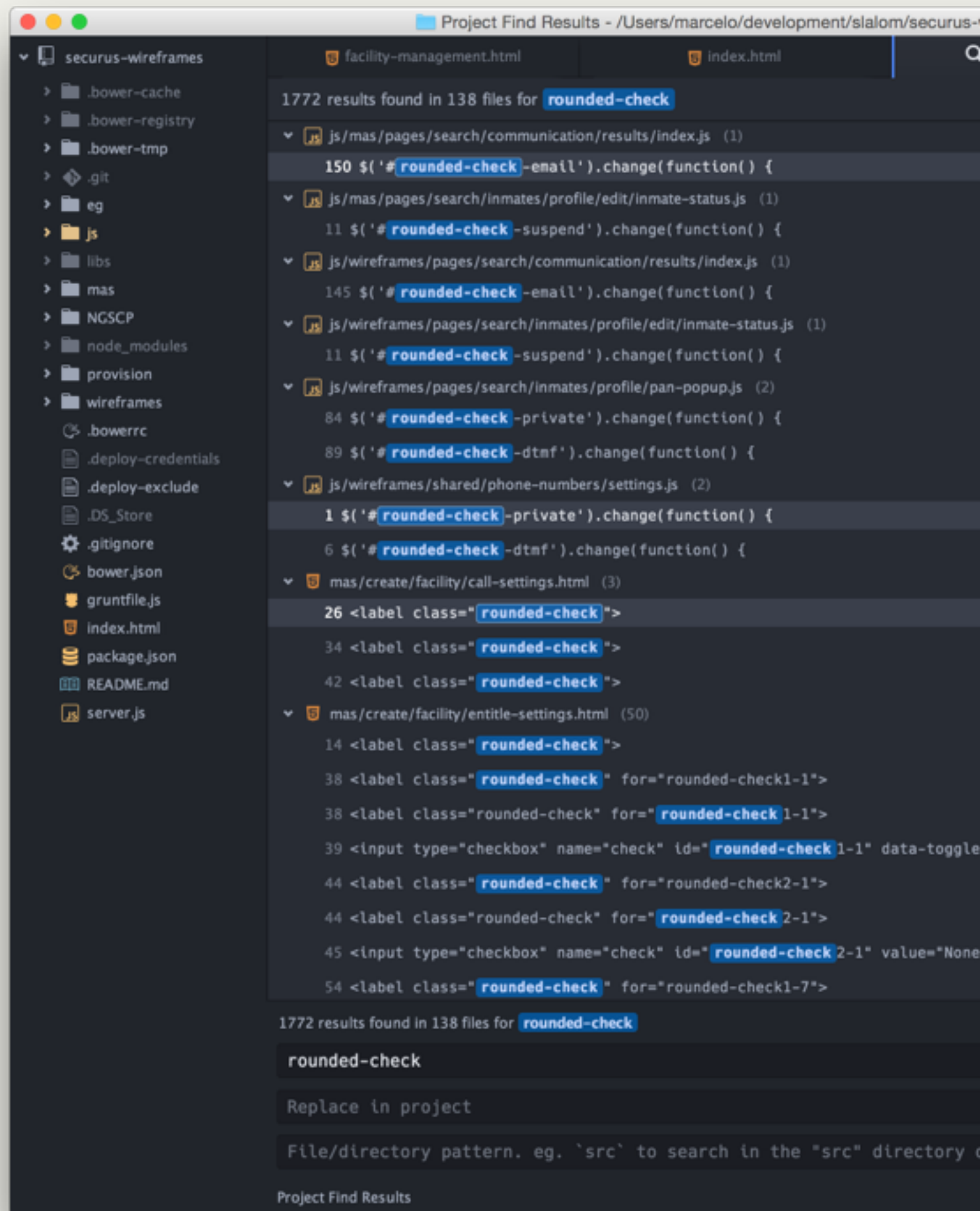
BREAK UP YOUR COMPONENTS



NAMESPACE THE CSS



FIND IN PROJECT IS YOUR FRIEND



REGEX SEARCH FOR CLASSES

```
class\s*?=\s*?["'"].*?table
```

will find all instances where class="" contains table

**DON'T FORGET
JAVASCRIPT!**



GOVERN YOUR LIBRARY

The screenshot shows a web browser window with the address bar displaying `css-tricks.com/what-a-css-code-review-might-look-like/`. The page features the CSS-Tricks logo at the top, followed by navigation links: Blog, Videos, Almanac, Snippets, and Forums. A search bar is located to the right of these links. The main heading of the article is "What a CSS Code Review Might Look Like", written by Geoff Graham on July 6, 2015. The article text discusses the importance of code reviews for CSS and provides a starting point for reviewing CSS code. A sidebar on the right contains a section titled "Front End Design & Development Jobs" with links to job listings from InteSolv and Chewy, and buttons for "See more jobs" and "Post a job".

CSS-TRICKS

Blog Videos Almanac Snippets Forums

What a CSS Code Review Might Look Like

 **GEOFF GRAHAM** // JULY 6, 2015

Many programming languages go through a code review before deployment. Whether it's a quick once-over, in-depth peer review, or complete unit testing, code reviews help us release code into the wild with confidence.

I started to imagine what a CSS code review might look like. CSS can be written in a number of ways, and the best way is often subjective to the project. I'm definitely not trying to get dogmatic with a post like this, but instead lay the foundation for what could be a starting point for getting the most out of CSS before it is released.

Why should CSS be reviewed at all?

It's fair to wonder why we would want to review CSS in the first place. A review is yet

Front End Design & Development Jobs

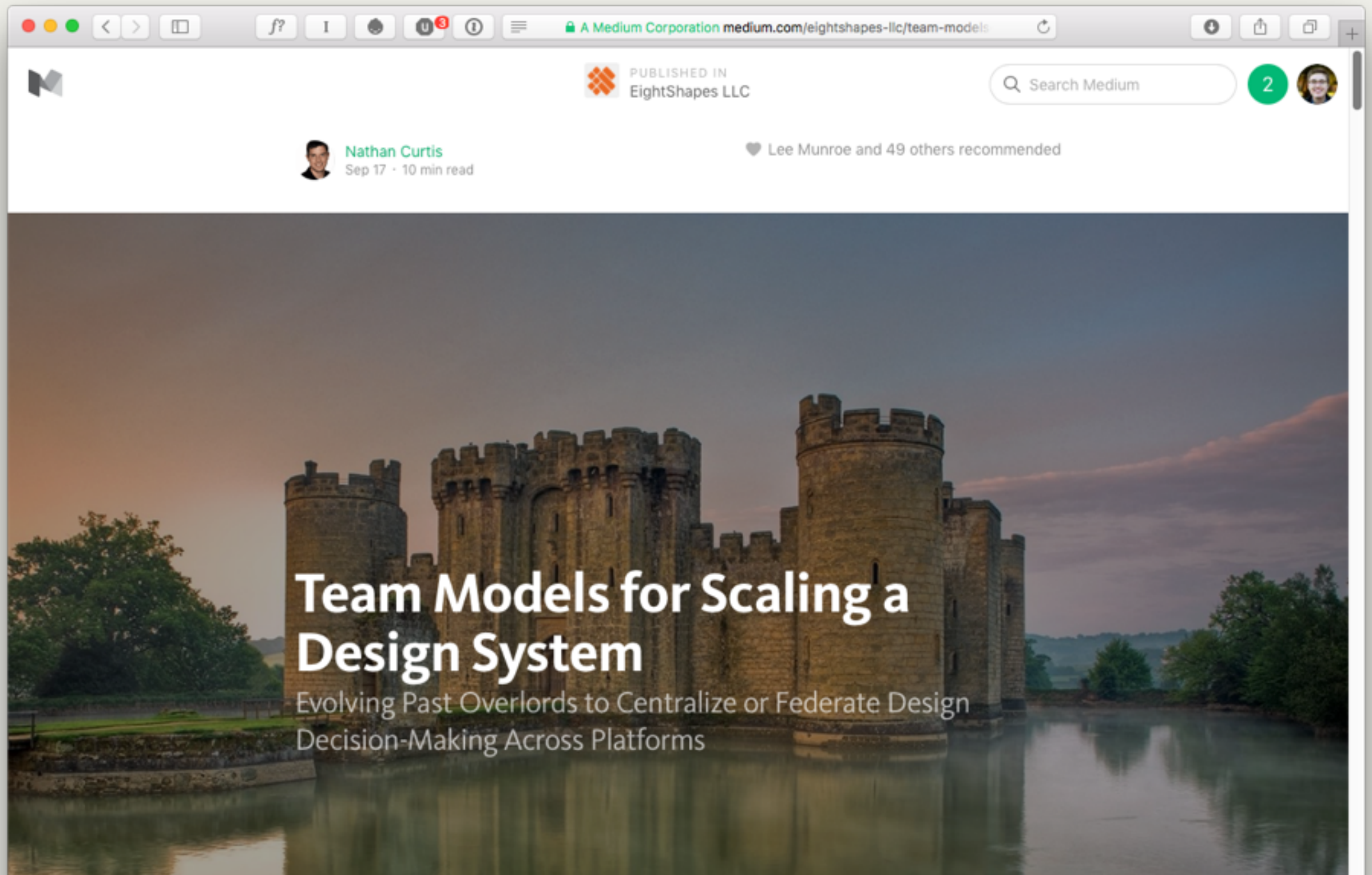
InteSolv is hiring a Full Stack .Net Developer with Strong Front End Skills

Chewy is hiring a Senior Front End Developer

Chewy is hiring a Senior Interaction Designer

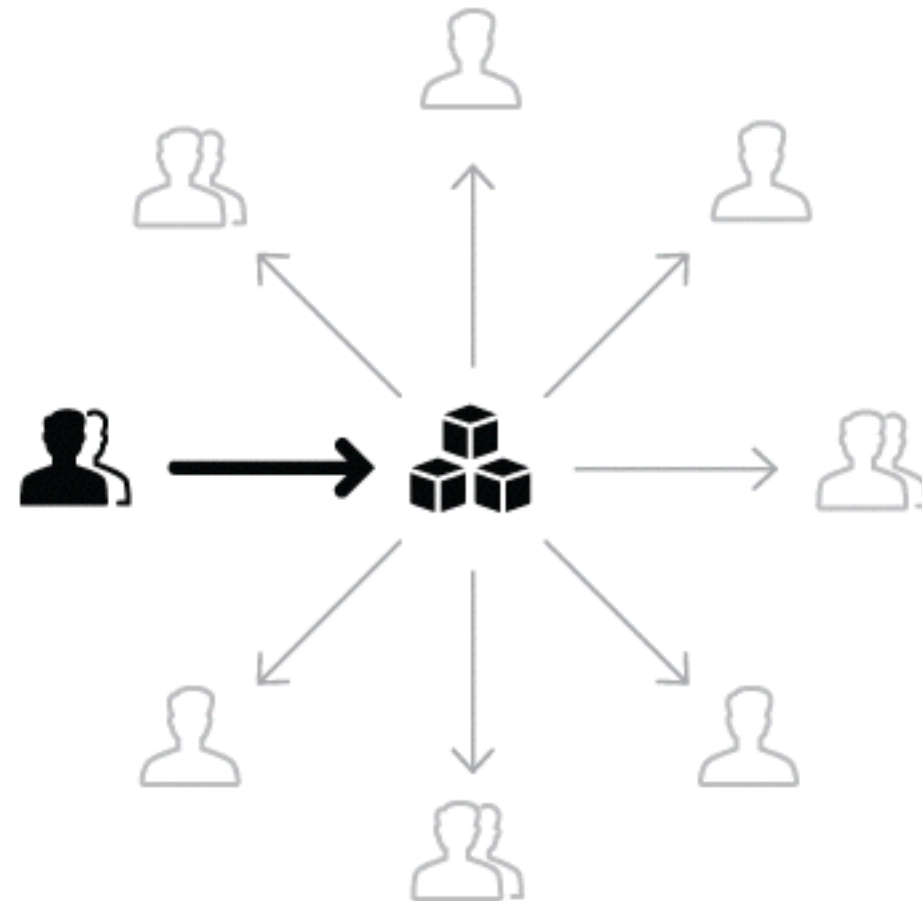
[See more jobs](#) [Post a job](#)

<https://css-tricks.com/what-a-css-code-review-might-look-like/>

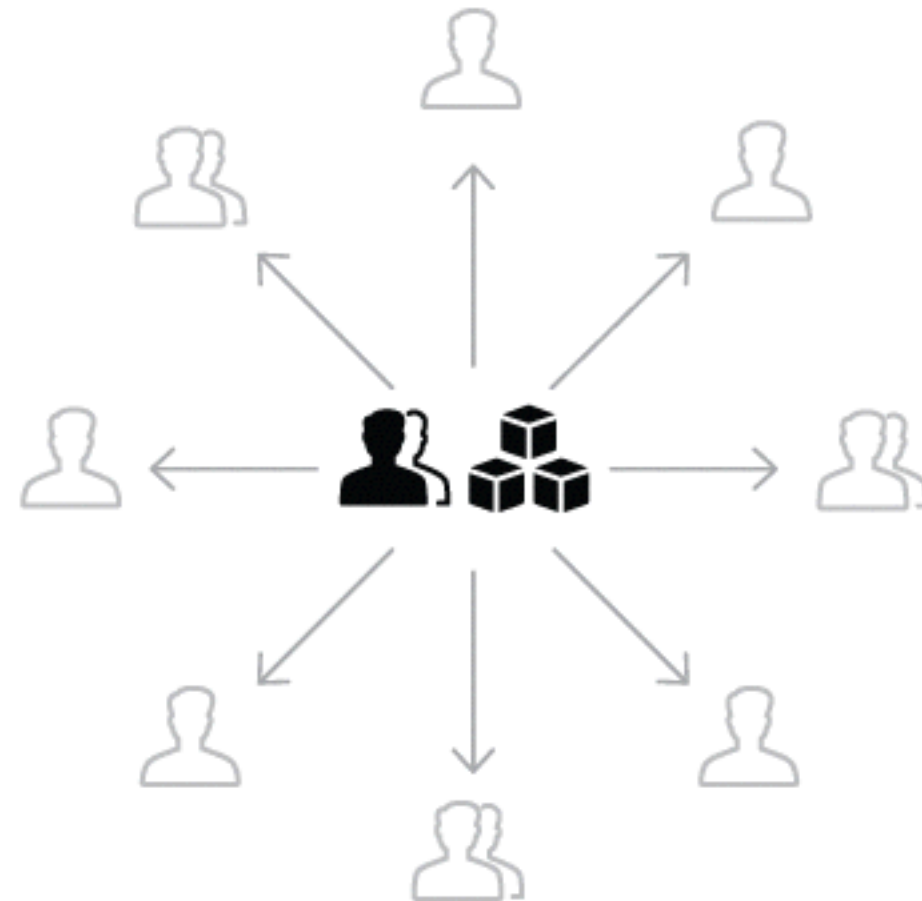


<https://medium.com/eightshapes-llc/team-models-for-scaling-a-design-system>

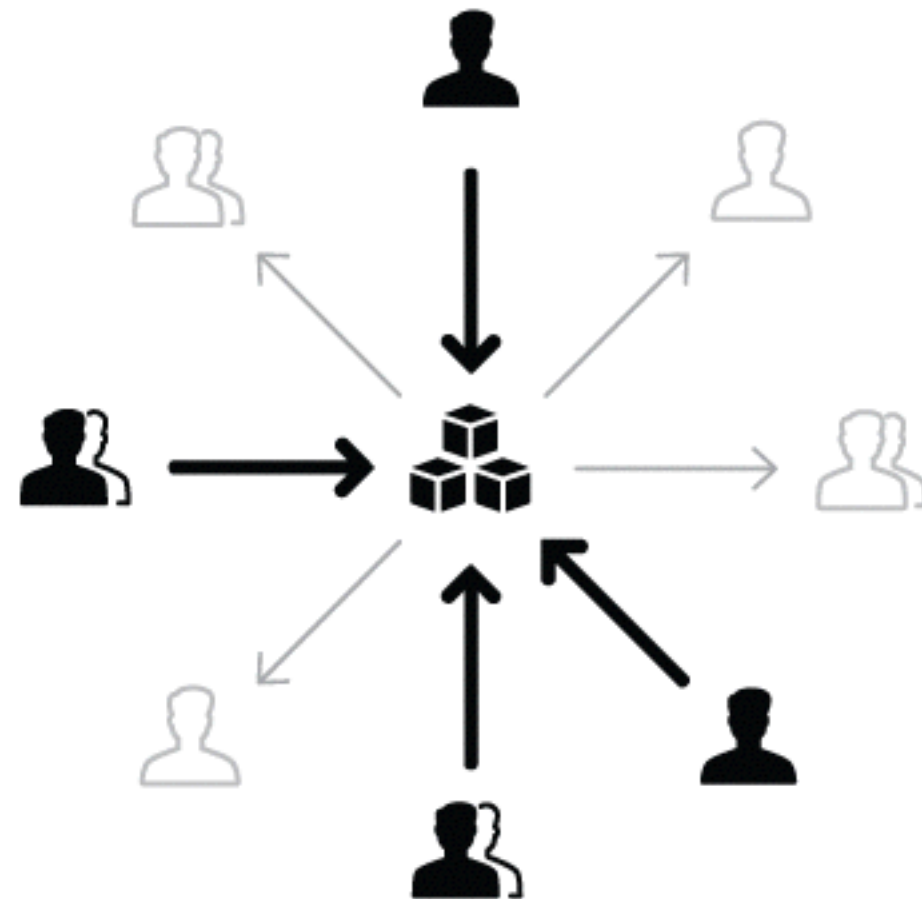
SOLITARY



CENTRALIZED



FEDERATED



“

Effective organizational design is more than drawing boxes and arrows, it flows from a thorough understanding of strategic objectives, takes into consideration both the formal and informal elements of the organizational units involved in the redesign and weighs the potential impact on the rest of the organization and its ability to fulfill strategic goals.

– Competing by Design, Nadler and Tushman

“

Effective organizational design is more than drawing boxes and arrows, it **flows from a thorough understanding of strategic objectives**, takes into consideration both the formal and informal elements of the organizational units involved in the redesign and weighs the potential impact on the rest of the organization and its ability to fulfill strategic goals.

– Competing by Design, Nadler and Tushman

“

Effective organizational design is more than drawing boxes and arrows, it flows from a thorough understanding of strategic objectives, **takes into consideration both the formal and informal elements of the organizational units involved in the redesign and weighs the potential impact on the rest of the organization and its ability to fulfill strategic goals.**

– Competing by Design, Nadler and Tushman

“

Effective organizational design is more than drawing boxes and arrows, it flows from a thorough understanding of strategic objectives, takes into consideration both the formal and informal elements of the organizational units involved in the redesign and **weighs the potential impact on the rest of the organization and its ability to fulfill strategic goals.**

– Competing by Design, Nadler and Tushman

1

WHY ARE WE DOING THIS?

2

HOW WILL IT FIT IN OUR ORGANIZATION?

3

WHAT INVESTMENT ARE WE WILLING TO MAKE?

4

WHO'S DOING THE WORK?

OPEN SOURCE

CULTURE

GREAT

DOCUMENTATION

MAKE IT

EASY TO CONTRIBUTE

FIND EASY
MISTAKES

EMBRACE

NON-DEVELOPERS

MAKE IT EASY TO
COMMUNICATE

OPEN & TRANSPARENT **ROADMAP**



PATTERN LIBRARY TOOLS



CSS DOCUMENTATION

KSS · Knyle Style Sheets

warpspire.com/kss/

John

KSS

Introduction Documentation Syntax Generate Styleguides

View the code on GitHub

KSS Knyle Style Sheets

Documentation for any flavor of CSS that you'll love to write. Human readable, machine parsable, and easy to remember.

Works great with CSS, SCSS, LESS, and much more.

```
// A button suitable for giving a star to someone.
//
// :hover           - Subtle hover highlight.
// .star-given      - A highlight indicating you've already given a star.
// .star-given:hover - Subtle hover highlight on top of star-given styling.
// .disabled        - Dims the button to indicate it cannot be used.
//
// Styleguide 2.1.3.
a.button.star{
  ...
  &.star-given{
    ...
  }
  &.disabled{
    ...
  }
}
```

KSS

Documentation for humans

Documentation is all about communication. Between people, not computers. So why should your documentation format cater to computers?

Works with SCSS, LESS, and more

KSS is designed to work with every flavor of CSS out there — preprocessor or not. Choose whatever works for you.

Automatically generate styleguides

Create example HTML for your CSS and automatically generate variations of each element.



STATIC SITE GENERATION

Pattern Lab | Build Atomic D x

John

← → ↺ 🏠

patternlab.io

☆ » ☰

 pattern lab

documentation

about

resources

demo

on github

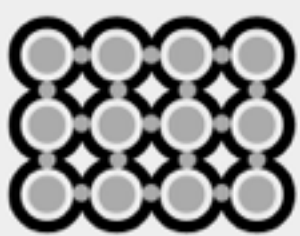
 **pattern lab**
create atomic design systems


atoms

▶


molecules

▶


organisms

▶


templates

▶


pages

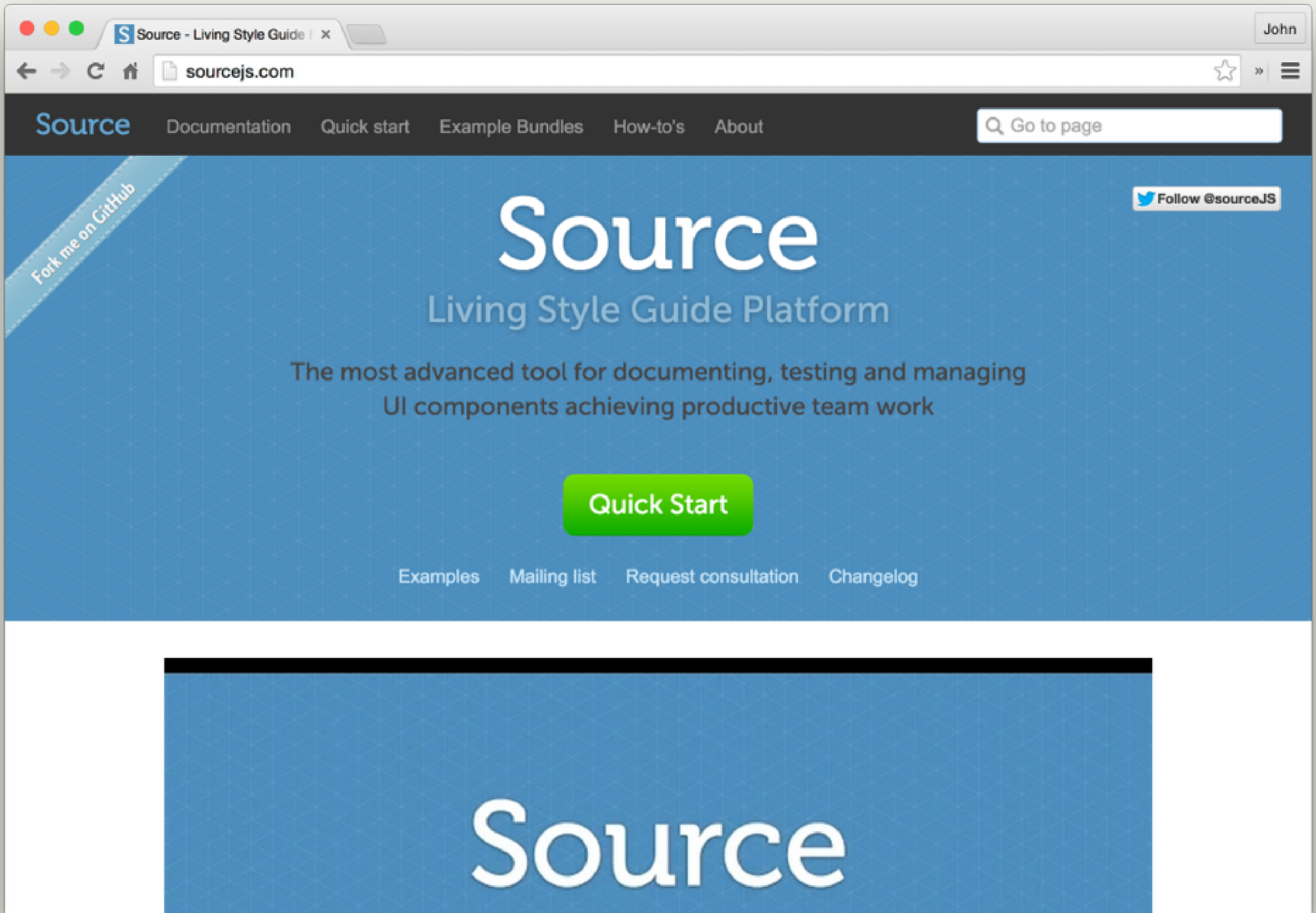
 download

 view on github

 view demo



INTEGRATED





THERE HAS TO BE A
BETTER WAY

Build and Document Your Interface. Then Share the Code.

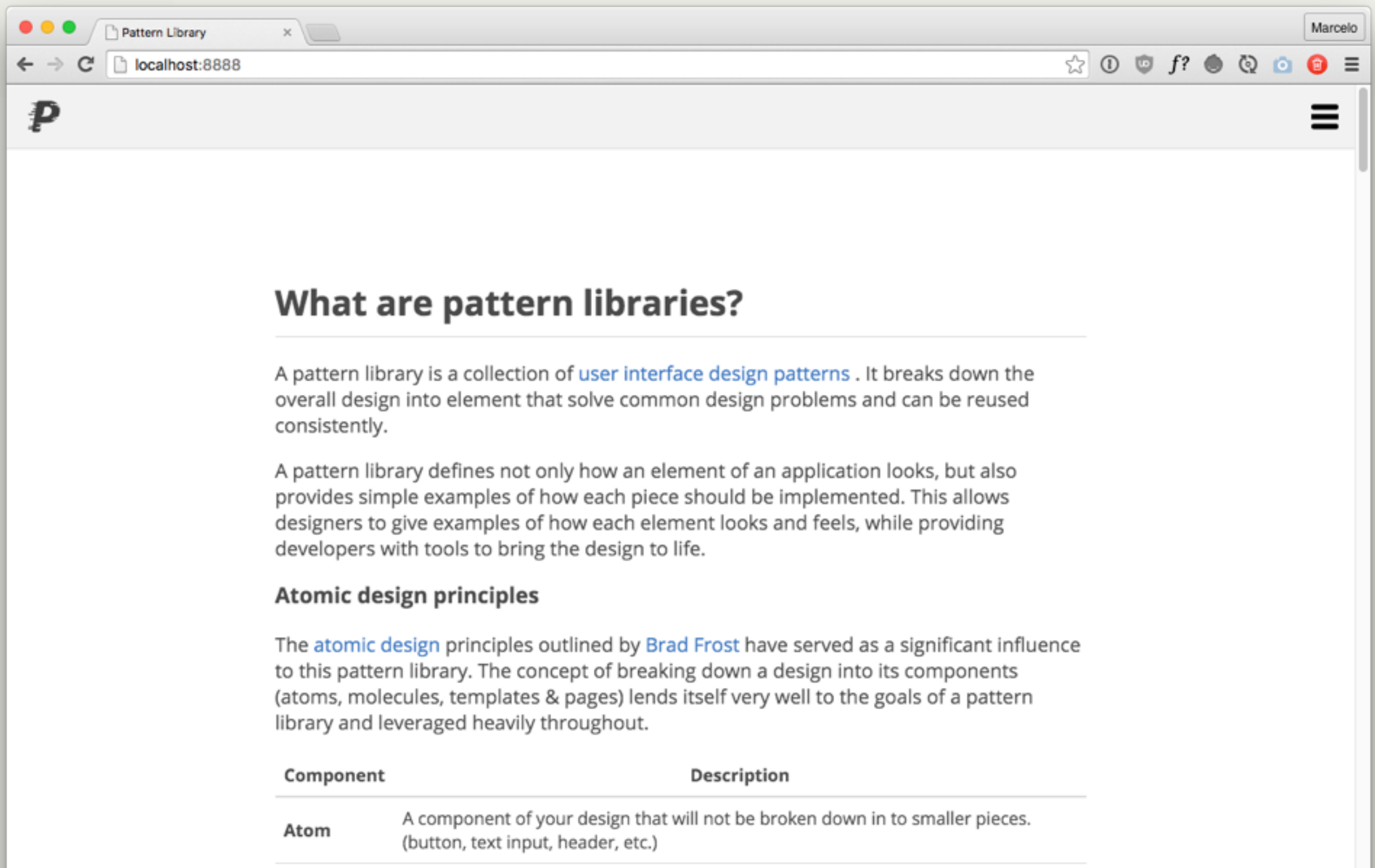
Get Started



<http://patternpack.org/>



GET STARTED WITH
**PATTERN
PACK**



Pattern Library

localhost:8888

Marcelo

P

What are pattern libraries?

A pattern library is a collection of [user interface design patterns](#) . It breaks down the overall design into element that solve common design problems and can be reused consistently.

A pattern library defines not only how an element of an application looks, but also provides simple examples of how each piece should be implemented. This allows designers to give examples of how each element looks and feels, while providing developers with tools to bring the design to life.

Atomic design principles

The [atomic design](#) principles outlined by [Brad Frost](#) have served as a significant influence to this pattern library. The concept of breaking down a design into its components (atoms, molecules, templates & pages) lends itself very well to the goals of a pattern library and leveraged heavily throughout.

Component	Description
Atom	A component of your design that will not be broken down in to smaller pieces. (button, text input, header, etc.)

<https://github.com/patternpack/patternpack-example-library>

```
$ npm init  
$ git init
```

```
$ npm install grunt patternpack --save-dev
```

gruntfile.js

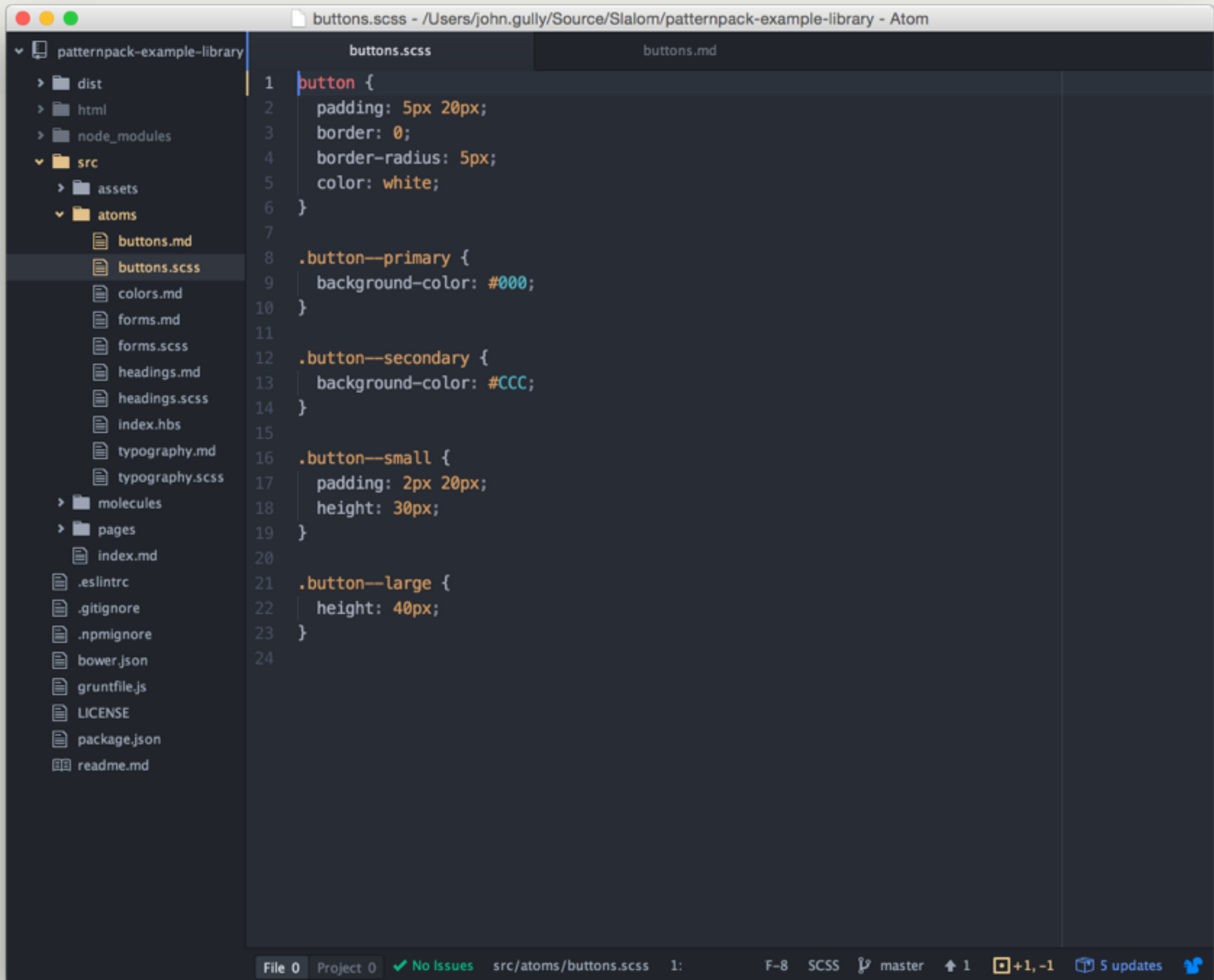
```
module.exports = function(grunt) {  
  grunt.initConfig({  
    patternpack: {  
      run: {},  
      build: {},  
      release: {}  
    }  
  });  
  
  grunt.loadNpmTasks('patternpack');  
  
  grunt.registerTask('default', ['patternpack:run']);  
}
```

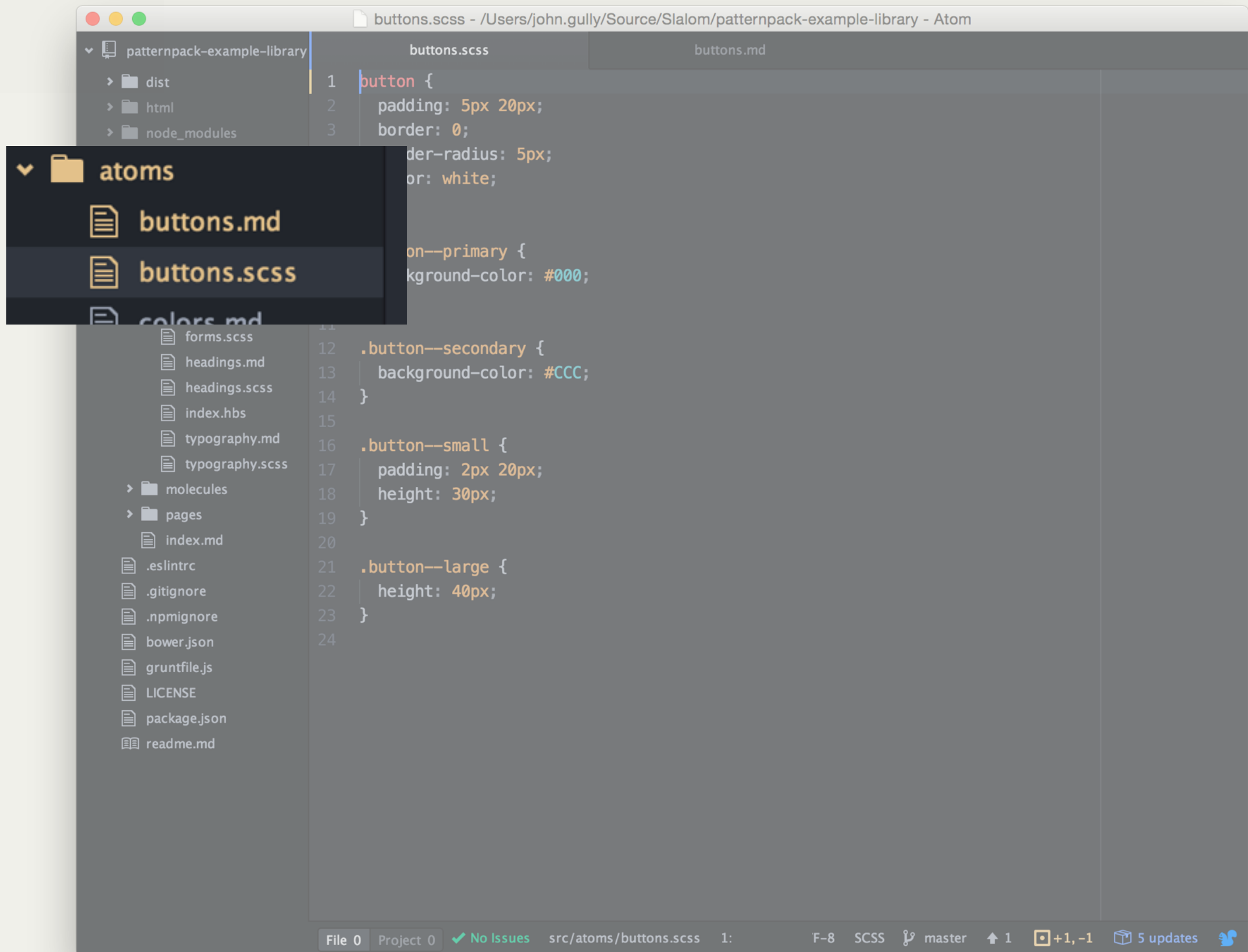
```
$ grunt patternpack:run
```

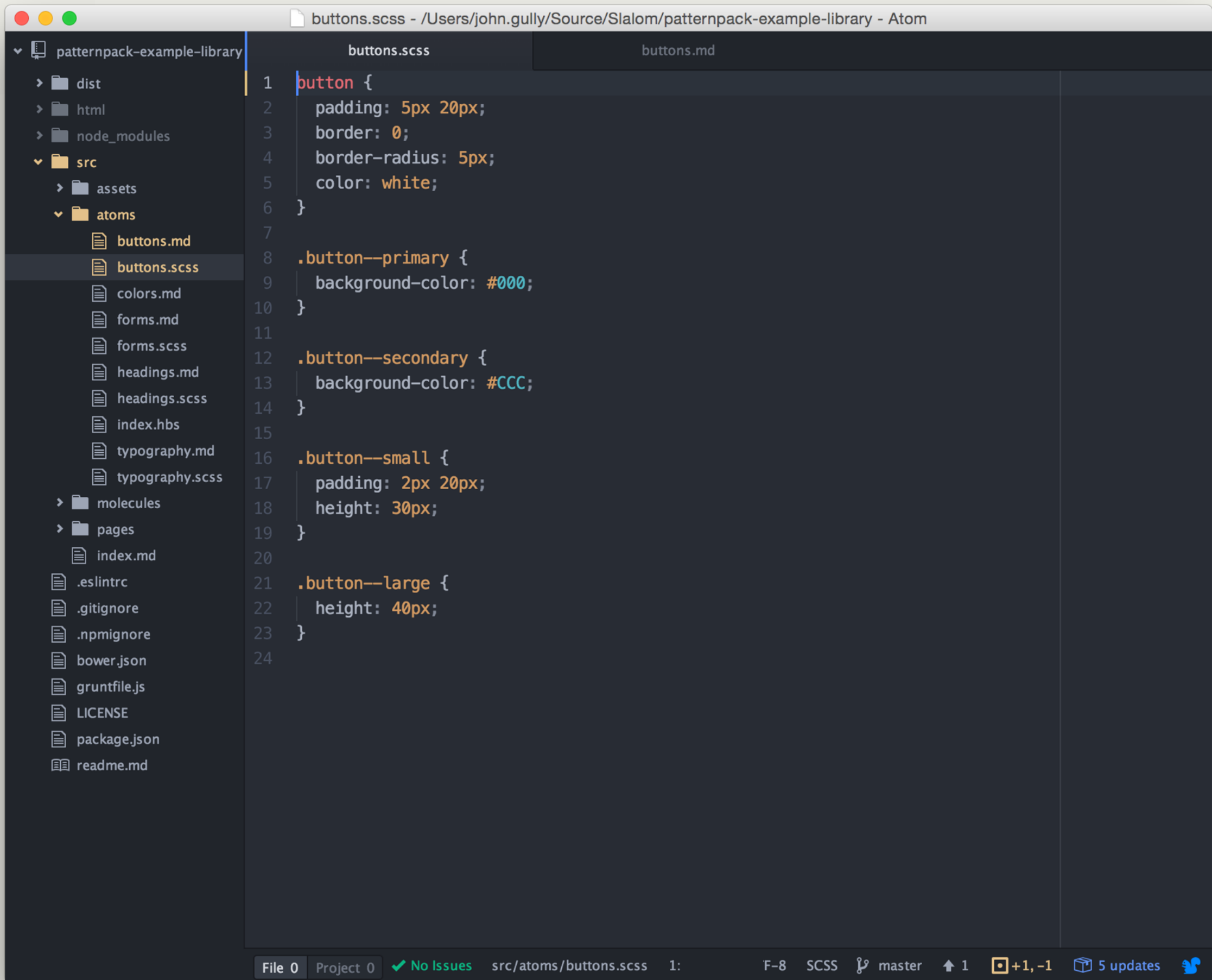
Welcome to PatternPack



CREATE YOUR FIRST **PATTERN**







patternpack-example-library

dist

html

node_modules

src

assets

atoms

buttons.md

buttons.scss

colors.md

forms.md

forms.scss

headings.md

headings.scss

index.hbs

typography.md

typography.scss

molecules

pages

index.md

.eslintrc

.gitignore

.npmignore

bower.json

gruntfile.js

LICENSE

package.json

readme.md

buttons.md - /Users/john.gully/Source/Slalom/patternpack-example-library - Atom

buttons.scss

buttons.md

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

title: Buttons

Buttons

TODO: Describe what buttons are and how they are used in the application.

Examples

<div class="library__example">

<button type="button" class="button--primary">Primary</button>

<button type="button" class="button--secondary">Secondary</button>

<button type="button" class="button--primary button--large">Primary Large</button>

<button type="button" class="button--secondary button--large">Secondary Large</button>

<button type="button" class="button--primary button--small">Primary Small</button>

<button type="button" class="button--secondary button--small">Secondary Small</button>

</div>

Code

```html

<button type="button" class="button--primary">Primary</button>

<button type="button" class="button--secondary">Secondary</button>

<button type="button" class="button--primary button--large">Primary Large</button>

<button type="button" class="button--secondary button--large">Secondary Large</button>

<button type="button" class="button--primary button--small">Primary Small</button>

<button type="button" class="button--secondary button--small">Secondary Small</button>

```

File 0

Project 0

✓ No Issues

src/atoms/butto

LF

UTF-8

GitHub Markdown

master

↑ 1

+1, -1

5 updates

```
1 ---
2 title: Buttons
3 ---
4 ## Buttons
5 TODO: Describe what buttons are and how they are used in the application.
6
```

- forms.scss
- headings.md
- headings.scss
- index.hbs
- typography.md
- typography.scss
- ▼ molecules
 - index.hbs
 - nav.md
- ▶ pages
 - index.md
- .eslintrc
- .gitignore
- .npmignore
- bower.json
- gruntfile.js
- LICENSE
- package.json
- readme.md

```
11 <br />
12 <button type="button" class="button--primary button--large">Primary Large</button>
13 <button type="button" class="button--secondary button--large">Secondary Large</button><br />
14 <br />
15 <button type="button" class="button--primary button--small">Primary Small</button>
16 <button type="button" class="button--secondary button--small">Secondary Small</button>
17 </div>
18
19 ### Code
20 ```html
21 <button type="button" class="button--primary">Primary</button>
22 <button type="button" class="button--secondary">Secondary</button>
23
24 <button type="button" class="button--primary button--large">Primary Large</button>
25 <button type="button" class="button--secondary button--large">Secondary Large</button>
26
27 <button type="button" class="button--primary button--small">Primary Small</button>
28 <button type="button" class="button--secondary button--small">Secondary Small</button>
29 ```
30
```

Examples

```
<div class="library__example">
```

```
  <button type="button" class="button--primary">Primary</button>
```

```
  <button type="button" class="button--secondary">Secondary</button><br />
```

```
<br />
```

```
  <button type="button" class="button--primary button--large">Primary Large</button>
```

```
  <button type="button" class="button--secondary button--large">Secondary Large</button><br />
```

```
<br />
```

```
  <button type="button" class="button--primary button--small">Primary Small</button>
```

```
  <button type="button" class="button--secondary button--small">Secondary Small</button>
```

```
</div>
```

nav.md

pages

index.md

.eslintrc

.gitignore

.npmignore

bower.json

gruntfile.js

LICENSE

package.json

readme.md

```
20 HTML
21 <button type="button" class="button--primary">Primary</button>
22 <button type="button" class="button--secondary">Secondary</button>
23
24 <button type="button" class="button--primary button--large">Primary Large</button>
25 <button type="button" class="button--secondary button--large">Secondary Large</button>
26
27 <button type="button" class="button--primary button--small">Primary Small</button>
28 <button type="button" class="button--secondary button--small">Secondary Small</button>
29 ``
30
```

patternpack-example-library

disthtmlnode_modules

src

assets

atoms

buttons.md

buttons.scss

colors.md

forms.md

forms.scss

headings.md

headings.scss

index.html

buttons.md

1

2

3

4

5

6

7

8

9

10

11

12

13

14

title: Buttons

Buttons

TODO: Describe what buttons are and how they are used in the application.

Examples

<div class="library__example">

<button type="button" class="button--primary">Primary</button>

<button type="button" class="button--secondary">Secondary</button>

<button type="button" class="button--primary button--large">Primary Large</button>

<button type="button" class="button--secondary button--large">Secondary Large</button>


```
19 ### Code
20 ```html
21 <button type="button" class="button--primary">Primary</button>
22 <button type="button" class="button--secondary">Secondary</button>
23
24 <button type="button" class="button--primary button--large">Primary Large</button>
25 <button type="button" class="button--secondary button--large">Secondary Large</button>
26
27 <button type="button" class="button--primary button--small">Primary Small</button>
28 <button type="button" class="button--secondary button--small">Secondary Small</button>
29 ```
```

patternpack-example-library

> dist

> html

> node_modules

> src

> assets

> atoms

buttons.md

buttons.scss

colors.mdforms.mdforms.scssheadings.mdheadings.scssindex.hbstypography.mdtypography.scss

> molecules

index.hbs

nav.md

> pages

index.md

.eslintrc

.gitignore

.npmignore

bower.json

gruntfile.js

LICENSE

package.json

readme.md

buttons.md

1 ---

2 title: Buttons

3 ---

4 ## Buttons

5 TODO: Describe what buttons are and how they are used in the application.

6

7 ### Examples

8 <div class="library__example">

9 <button type="button" class="button--primary">Primary</button>

10 <button type="button" class="button--secondary">Secondary</button>

11

12 <button type="button" class="button--primary button--large">Primary Large</button>

13 <button type="button" class="button--secondary button--large">Secondary Large</button>

14

15 <button type="button" class="button--primary button--small">Primary Small</button>

16 <button type="button" class="button--secondary button--small">Secondary Small</button>

17 </div>

18

19 ### Code

20 ```html

21 <button type="button" class="button--primary">Primary</button>

22 <button type="button" class="button--secondary">Secondary</button>

23

24 <button type="button" class="button--primary button--large">Primary Large</button>

25 <button type="button" class="button--secondary button--large">Secondary Large</button>

26

27 <button type="button" class="button--primary button--small">Primary Small</button>

28 <button type="button" class="button--secondary button--small">Secondary Small</button>

29 ```

30

File 0

Project 0

✓ No Issues

src/atoms/buttons.i

LF

UTF-8

GitHub Markdown

master

↑ 1

+16, -13

5 updates

used in the application.

```
Primary</button>
Secondary</button><br />
```

```
button--large">Primary Large</button>
button--large">Secondary Large</button><br />
```

```
button--small">Primary Small</button>
button--small">Secondary Small</button>
```

```
Primary</button>
Secondary</button>
```

```
button--large">Primary Large</button>
button--large">Secondary Large</button>
```

```
button--small">Primary Small</button>
button--small">Secondary Small</button>
```



Buttons

TODO: Describe what buttons are and how they are used in the application.

Examples

Primary

Secondary

Primary Large

Secondary Large

Primary Small

Secondary Small

Code

```
<button type="button" class="button--primary">Primary</button>
<button type="button" class="button--secondary">Secondary</button>
<button type="button" class="button--primary button--large">Primary Large</button>
<button type="button" class="button--secondary button--large">Secondary Large</button>
<button type="button" class="button--primary button--small">Primary Small</button>
<button type="button" class="button--secondary button--small">Secondary Small</button>
```


UNLEASH YOUR
CREATION

USE **SEMANTIC VERSIONING** FOR YOUR DESIGN SYSTEM

1.0.5

major
release

minor
release

patch
release

105

Customize to your needs

major
release

minor
release

patch
release

```
$ grunt patternpack:release  
$ git push --follow-tags
```

1

BUILD YOUR STATIC SITE

2

INCREMENT YOUR VERSION

3

CREATE A NEW COMMIT

4

TAG THE COMMIT

INTEGRATE INTO YOUR
APPLICATION

f? I A Medium Corporation medium.com/@marcelosomers/chasing-the-holy-grail-bbc0b7cce365

Edit

Search Medium

13

Marcelo Somers

Dec 3, 2015 · 8 min read

Recommended by Jason Cypret, Sherif Tariq, and 10 others

Chasing the Holy Grail

Strategies For Distributing Your Pattern Library and Keeping It in Sync



In our work with Pattern Libraries, we strive to achieve a level of maturity where the Pattern Library documentation is automatically in sync with the app(s) that it powers. Generally, this means sharing the CSS code so developers implementing patterns can just copy and paste snippets of HTML so it “just works.”

<https://medium.com/@marcelosomers/chasing-the-holy-grail-bbc0b7cce365#wvedzdvsx>

SUPPORT FOR NPM & BOWER

```
$ npm install my-awesome-pattern-library
```

```
$ npm install https://user:pass@github.com/user/my-  
awesome-library.git
```

ANATOMY OF A PACKAGE.JSON DEPENDENCY

```
"devDependencies": {  
  "my-awesome-pattern-library": "https://user:pass@bitbucket.org/jondoe/  
my-awesome-pattern-library.git#1.0.0"  
}
```

ANATOMY OF A PACKAGE.JSON DEPENDENCY

```
"devDependencies": {  
  "my-awesome-pattern-library": "https://user:pass@bitbucket.org/jondoe/  
my-awesome-pattern-library.git#1.0.0"  
}
```



**USERNAME &
PASSWORD**

ANATOMY OF A PACKAGE.JSON DEPENDENCY

```
"devDependencies": {  
  "my-awesome-pattern-library": "https://user:pass@bitbucket.org/jondoe/  
  my-awesome-pattern-library.git#1.0.0"  
}
```

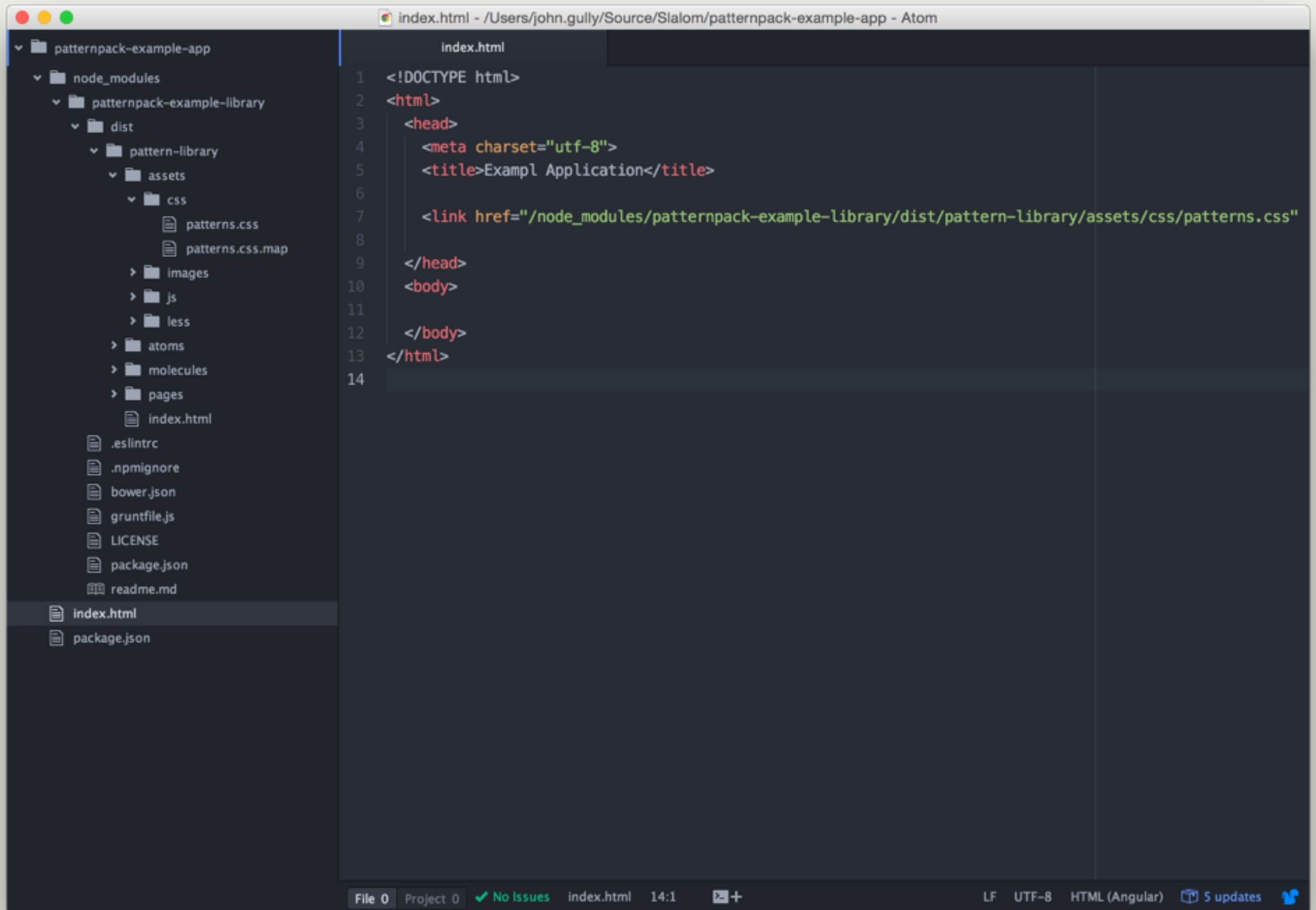
**PATH TO GIT
REPOSITORY**

ANATOMY OF A PACKAGE.JSON DEPENDENCY

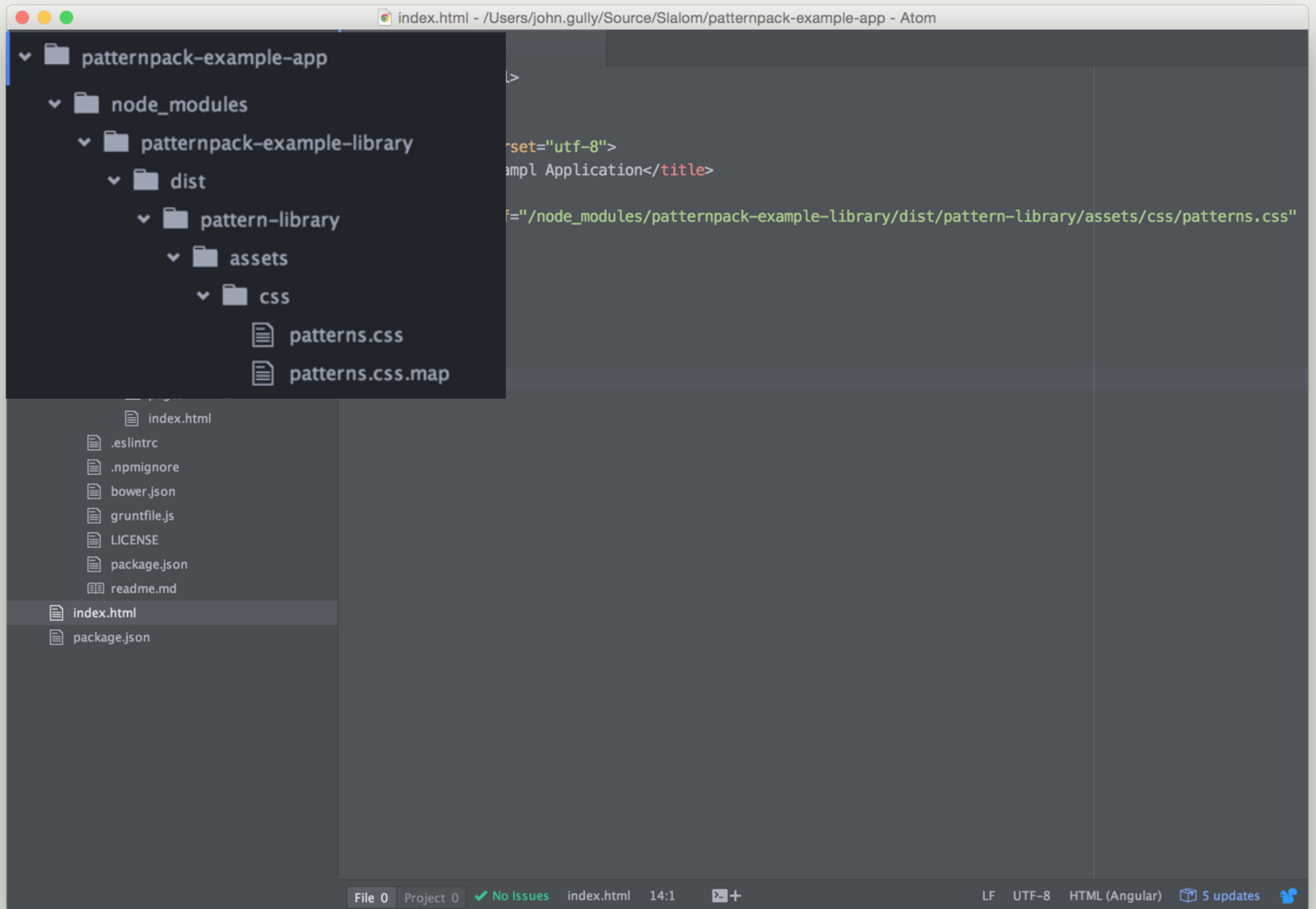
```
"devDependencies": {  
  "my-awesome-pattern-library": "https://user:pass@bitbucket.org/jondoe/  
my-awesome-pattern-library.git#1.0.0"  
}
```

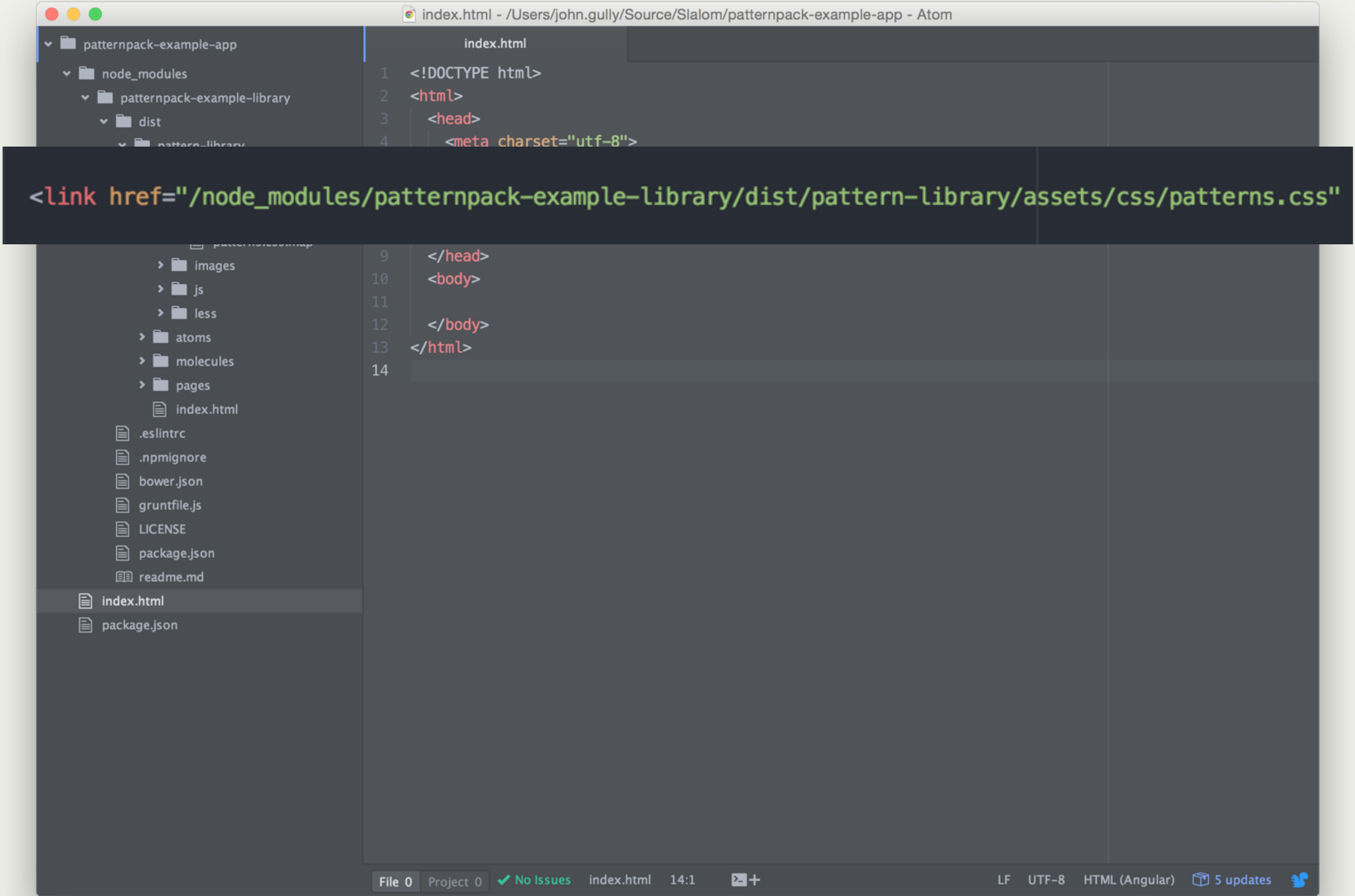


**VERSION
COMMITISH**



#ZombieLibrary





Build and Document Your Interface. Then Share the Code.

Get Started



<http://patternpack.org/>

THIS IS MORE THAN JUST A
TECH PROBLEM



FIGHT ZOMBIE

INTERFACES



MODULARIZE

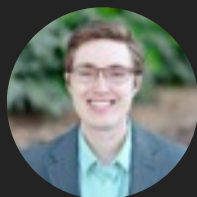
A group of survivors from the TV show The Walking Dead are walking in a line on a paved road. From left to right, there is a man with a backpack, a man in a tan shirt, a man in a brown shirt holding a machete, a woman in a black tank top, and a man in a light shirt holding a knife. In the background, another man is visible. The word "DOCUMENT" is overlaid in large white letters across the center of the image.

DOCUMENT



SHARE

THANK YOU!



Marcelo Somers
@marcelosomers

Slides

<http://j.mp/zombie-library-rwdsummit>

PatternPack

<http://patternpack.org/>