EMULSIFY 2.0

Component Driven Development

CHRIS MARTIN (CCJJMARTIN)
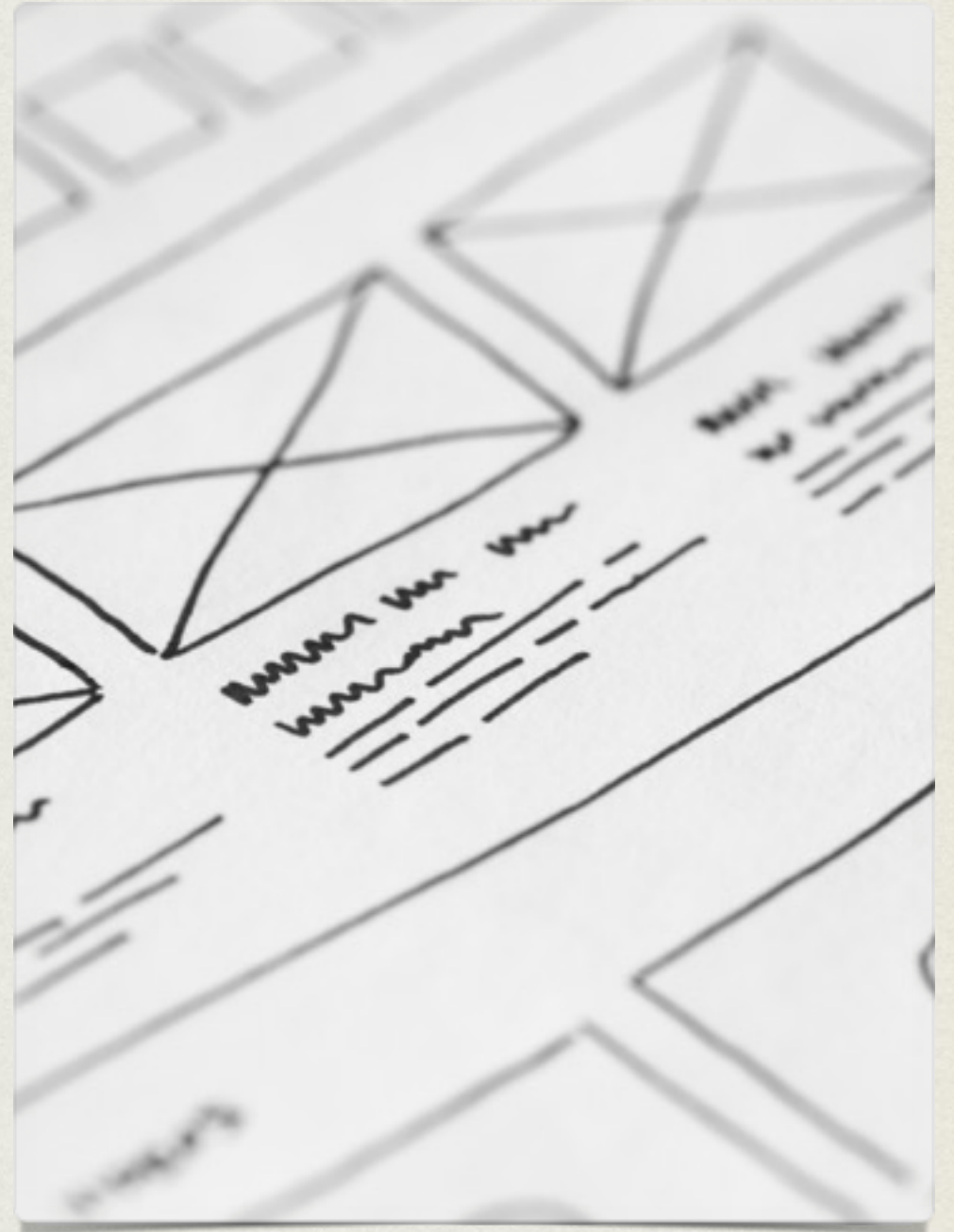
Support Engineer at Four Kitchens

AGENDA

- Component Driven Design/Development (CDD)

- Emulsify's implementation of CDD

- Getting Started with Emulsify

- Connecting Emulsify to Drupal

- New Features in Emulsify 2.0

- Emulsify 3.0 Roadmap

- Building your first Emulsify component

- Tips, Additional Resources, and Credits

PREREQUISITES

- NVM (Node Version Manager)

- Basic BEM (Block Element Modifier CSS Methodology)

- Basic Pattern Lab (Atomic Design System)

- Basic Drupal 8 Theming

- Basic Command Line Knowledge

COMPONENT DRIVEN DEVELOPMENT

Advantages

Disadvantages

BENEFITS OF CDD

- Builds your brand by standardizing your site appearance

- Easy to maintain (change once apply everywhere)

- Easily portable (you are building a style guide first and integrating it with a CMS second)

- Easy to remove one piece later (code is contained)

HIDDEN BENEFITS OF CDD

- Developers job is made easier due to less back and forth on ideas since they are now visible

- Most clients have problems deciding if something is built right until they see it, CDD allows clients to see the end result early in the build process.

- Designers don't need to have a Drupal site running to build your design

DISADVANTAGES OF CDD

- CDD is likely to increase your time spent in the Design phase of a project

  - I list this as a con because at first glance you will think that your project is moving slower but this is actually a great thing because CDD allows prototyping to happen faster and logic flaws discovered at this phase of development are much cheaper than if they are discovered later.

EMULSIFY'S COMPONENT DRIVEN DEVELOPMENT

Pattern Lab

Living Style Guide

BEM

PATTERN LAB

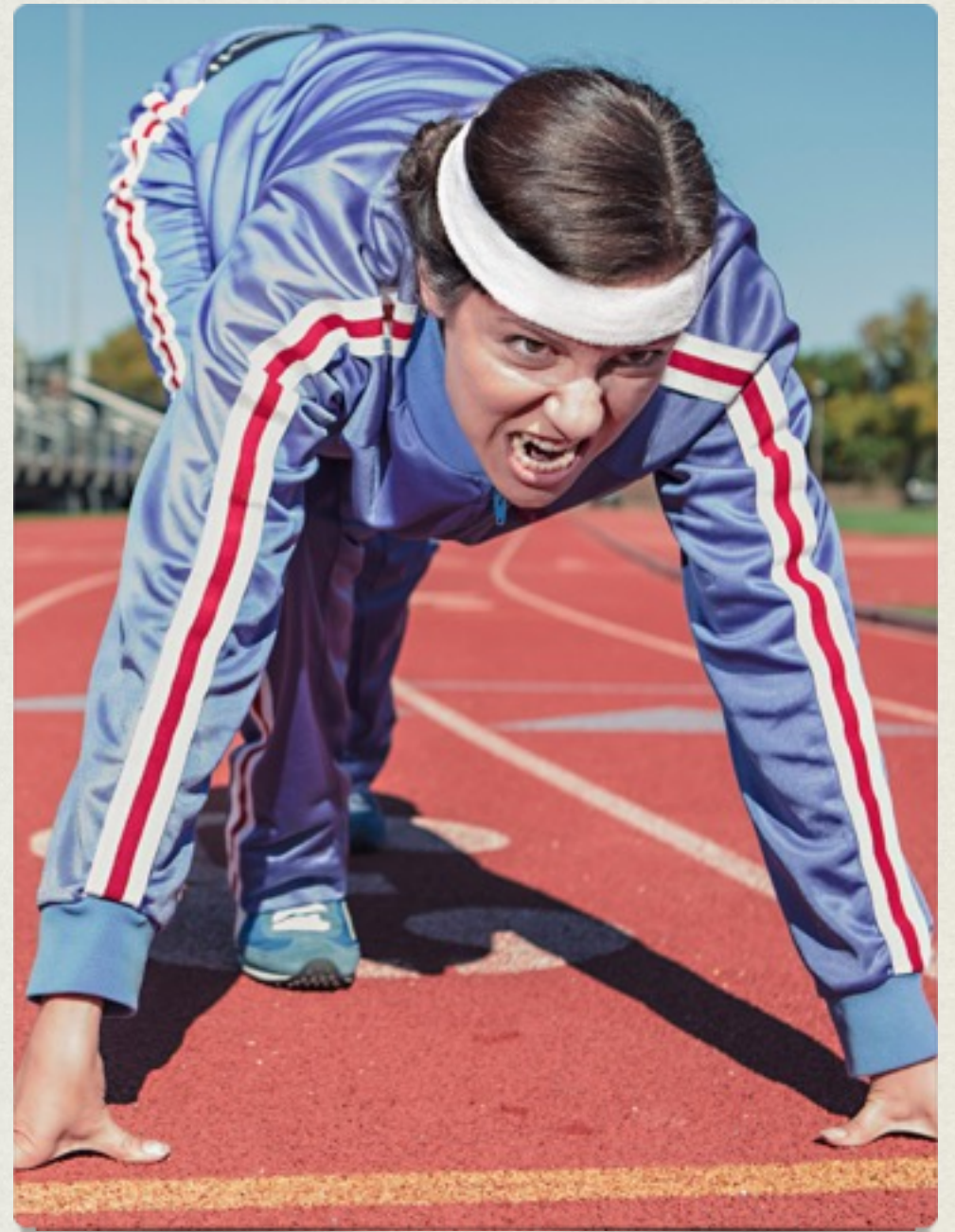- Demo: https://fourkitchens.github.io/emulsify/pattern-lab/public/

BEM

- Possible CDD replacements: OOCSS, SMACS, etc.

- Block Element Modifier:

  - Block: Card (.card) - Stand alone entity

  - Element: Image (.card__image) - Tied to a block

  - Modifier: Large (.card__image--large) - Changes appearance or behavior

LEGACY PSEUDO BEM

- We are moving to full BEM but in the mean time you may find some pseudo BEM that looks like:

  - Block: Contact Form (.contact-form)

  - Element: Submit Button (.submit .contact-form__submit)

  - Modifier: Disabled (.submit .submit--disabled .contact-form__submit)

GETTING STARTED WITH EMULSIFY

BASIC COMPONENT

http://patternlab.io/docs/pattern-organization.html

patterns/[Type]/[Subtype]/[Name].[Extension]

```
/patterns
 /organisms
  /pattern-name
   pattern-name.twig
   pattern-name.scss
   pattern-name.js
   pattern-name.md
   pattern-name.yml
   pattern-name~variation.yml
```

YML FILES

- https://github.com/fourkitchens/emulsify/blob/develop/components/_patterns/01-atoms/01-links/link/link.twig#L3

- Pass data into your twig files in pattern lab only, your templates directory will be expected to connect the same variables during your integration with Drupal phase.

HOW TO USE THE COMPONENT

- Include - pull in a file and replace the variables

- Extends - pull in a file and replace blocks

- Embed - pull in a file and replace variables and/or blocks

ACTUAL USEAGE INCLUDE

```twig
{% if card_link_url %}
  {% include "@atoms/01-links/link/link.twig" with {
    "link_content": card_link_text,
    "link_url": card_link_url,
    "link_attributes": card_link_attributes,
    "link_blockname": card_link_blockname|
default(card_base_class),
    "link_base_class": card_link_base_class|
default('link'),
  } %}
{% endif %}
```
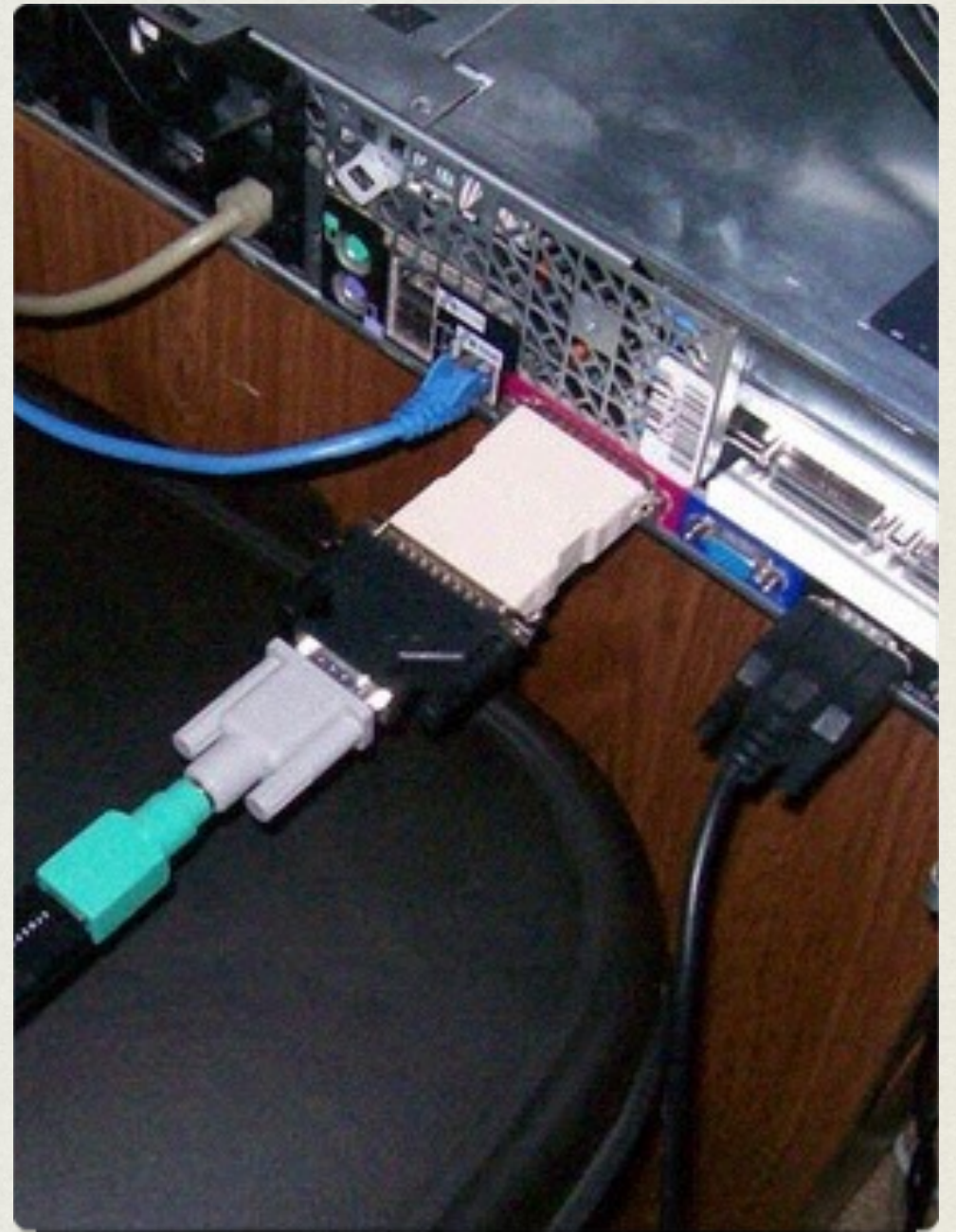
ACTUAL USEAGE DOES NOT USE EXTENDS

- Not used in practice because we always have variables that need to be passed or blocks that need to be renamed

ACTUAL USEAGE EMBED

```
{% embed "@atoms/01-links/link/link.twig" with {
  "link_url": img_url,
  "link_attributes": image_link_attributes,
  "link_base_class": image_link_base_class,
  "link_modifiers": image_link_modifiers,
  "link_blockname": image_link_blockname,
} %}
  {% block link_content %}
   <img
    {{ bem(image_base_class, (image_modifiers), image_blockname|
default(image_link_base_class)) }}
     src="{{ img_src }}"
     {% if img_alt %}
      alt="{{ img_alt }}"
     {% endif %}
    />
  {% endblock %}
{% endembed %}
```

CONNECTING EMULSIFY TO DRUPAL

It's all about the templates directory
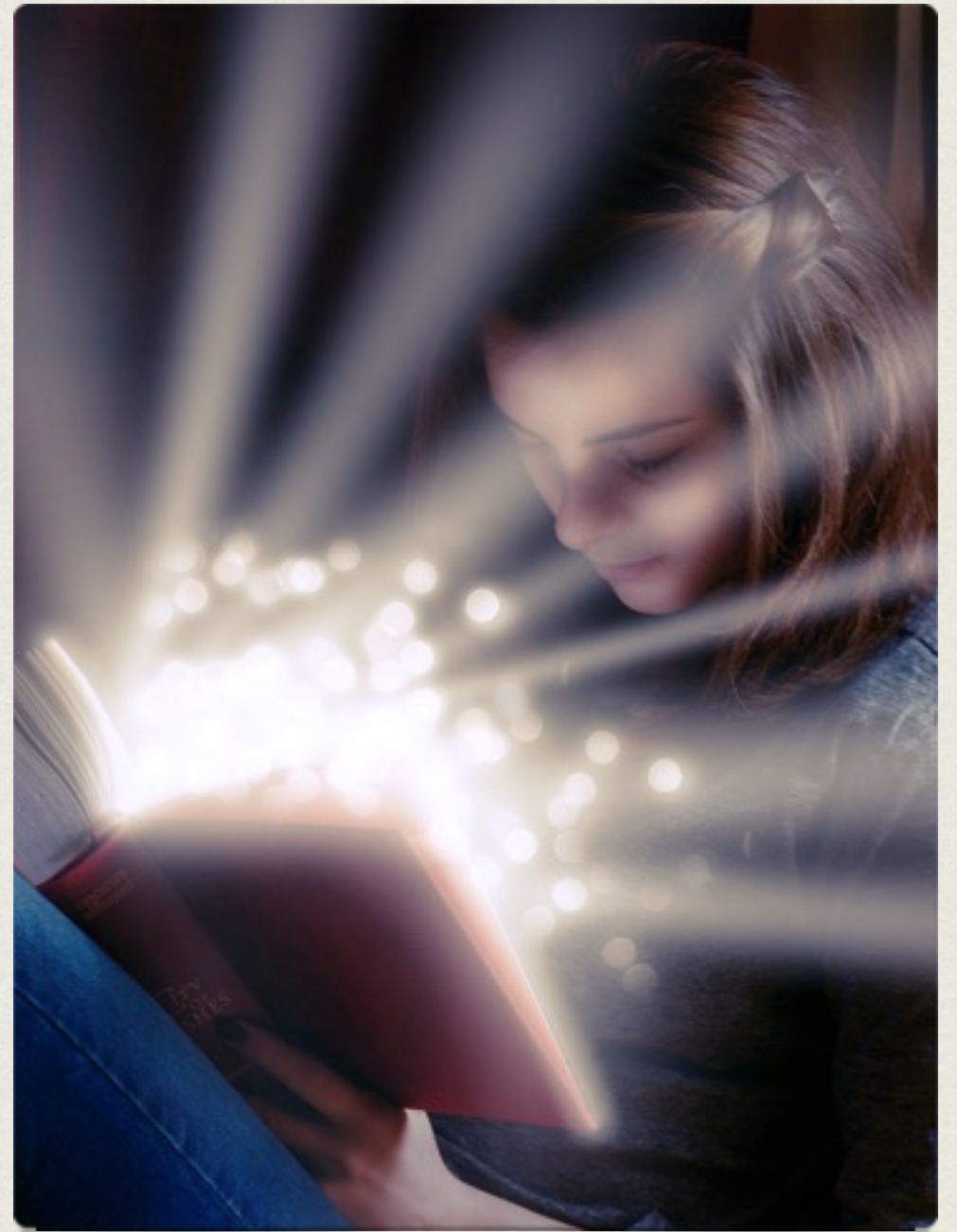
DRUPAL INTEGRATION

```
{% include "@atoms/02-text/00-headings/heading-1/
heading-1.twig"
  with {
    "heading_1_blockname": "page-title",
    "heading_1": title,
  }
%}
```

ATOMIC DESIGN IN DRUPAL (NOT SET IN STONE)

- Work in progress looking for community feedback:

  - Atom = Drupal Fields and Basic Markup

  - Molecules = Drupal Nodes

  - Organisms = Drupal Views / Drupal Blocks

  - Templates = Not connected to Drupal

  - Pages = Not connected to Drupal

NEW FEATURES IN EMULSIFY 2.0

Drush

Composer

BEM

Attributes

DRUSH

- New Drush command to assist you with scaffolding Emulsify

- Useage `drush9 @devdesktop.loc.emulsify-example help emulsify`

COMPOSER

- You can now install Emulsify via composer

- Stand alone install: `composer create-project fourkitchens/emulsify --stability dev --no-interaction emulsify && cd emulsify && yarn install`

- Into your existing project: `composer require fourkitchens/emulsify`

BEM

- New BEM function to assist you with class management

- https://github.com/drupal-pattern-lab/bem-twig-extension/blob/master/bem.function.php

- https://fourkitchens.github.io/emulsify/pattern-lab/public/

- https://github.com/fourkitchens/emulsify/blob/develop/components/_patterns/01-atoms/01-links/link/link.twig#L19

- Importance of name spacing your BEM parameters

ATTRIBUTES

- Attributes are an essential part of html so it only makes sense to support customizing them.

- https://github.com/drupal-pattern-lab/add-attributes-twig-extension/blob/master/add_attributes.function.php

EMULSIFY 3.0

Roadmap

ROADMAP

- Complete overhaul of components:

  - https://github.com/fourkitchens/emulsify/pull/150

INTERMISSION

Let's take a quick break

BUILDING COMPONENTS WITH EMULSIFY

Time to dig in

TIPS, ADDITIONAL RESOURCES, AND CREDITS

Twig Dump
Additional Resources

TWIG DUMP FUNCTION

- https://twig.symfony.com/doc/2.x/functions/dump.html

- https://github.com/pattern-lab/patternengine-php-twig#enable-dump

- This can be enabled in Pattern Lab by going to:

  - pattern-lab/config/config.yml

- Change `twigDebug: false` to `twigDebug: true`

- You can now use: `{{ dump(variable_name) }}`

ADDITIONAL RESOURCES

- Pattern Lab: http://patternlab.io/

- Style Guides: http://styleguides.io/

- Specificity: https://css-tricks.com/specifics-on-css-specificity/

CREDITS

- Emulsify 1.0: https://www.youtube.com/watch?v=m0kJpc7J1rM

- Design Systems Building The Future: https://css-tricks.com/design-systems-building-future/

- Component Driven Design: https://events.drupal.org/barcelona2015/sessions/component-driven-design-and-development

- Images provided by: https://www.pexels.com/