



DRUPAL DEVELOPER DAYS VIENNA • 2023



UI Suite from the trenches

Joys and struggles implementing a design system inside Drupal



Florent Torregrosa ([Grimreaper](#))

Since Drupal 6 (2011)

Tech expert at [Smile](#)

Contributions:

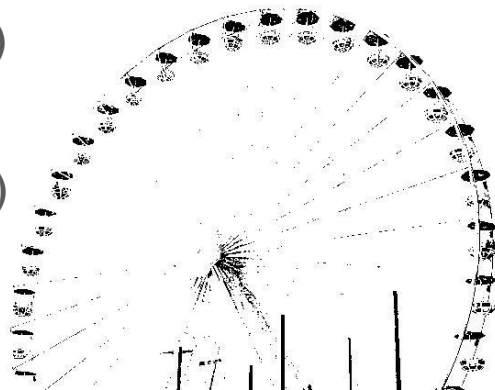
- 40+ projects maintainer
- contrib and core patches
- French translation moderator
- event co-organizer
- former Drupal France board member

Since 2021: UI Suite (projects & contrib)



Summary

1. UI Suite
2. Layout & grids
3. Style utilities & helpers
4. Component & variants
 - Declaration
 - Identification (& Bootstrap 5 off-canvas example)
 - Templating
 - Site building (with Bootstrap 5 carousel example)




Moving target

Those best practices may evolve because of:

- new features from design systems
- new way of implementing features due to evolutions in underlying modules
- new Twig functions and filters
- cases not yet encountered in design systems

Not only Bootstrap

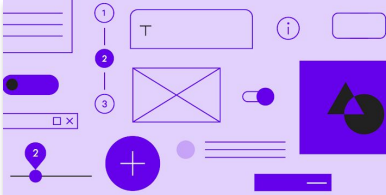


Bootstrap 5

Bootstrap 5
ui suite bootstrap

30 components
32 styles

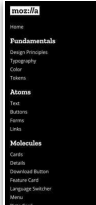
Progress: **95%**



Material Design 2
ui suite material

22 components
4 styles

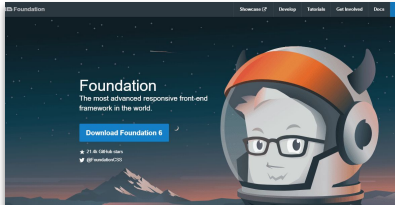
Progress: **70%**



Mozilla Protocol
ui suite protocol

22 components
2 styles

Progress: **90%**



Zurb Foundation
ui suite zurb foundation

23 components
10 styles

Progress: **15%**



DSFR
ui suite dsfr

41 components
17 styles

Progress: **70%**

And many private ones...

UI Suite

A design system in a Drupal theme

UI Suite is:

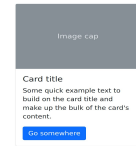
- A methodology to implement reusable Design Systems in a single and shareable place (the Drupal theme) and in a front-dev friendly way (no PHP)
- An ecosystem of modules to make those implementations available as Drupal API for site building and back development

See: [/project/ui_suite](#)

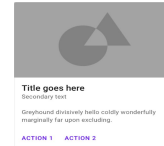
Parts of a design system

Components (patterns & variants)

Bootstrap



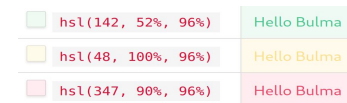
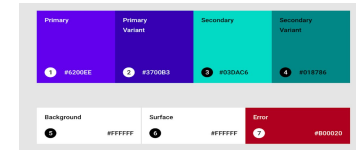
Material



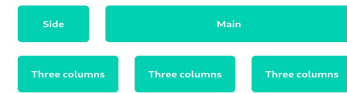
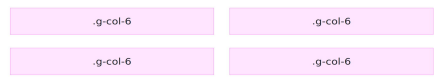
Bulma



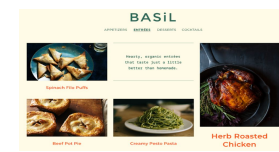
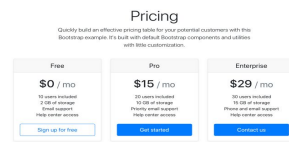
Styles (utilities or helpers)



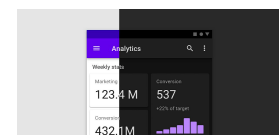
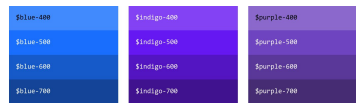
Layouts (often grid based)



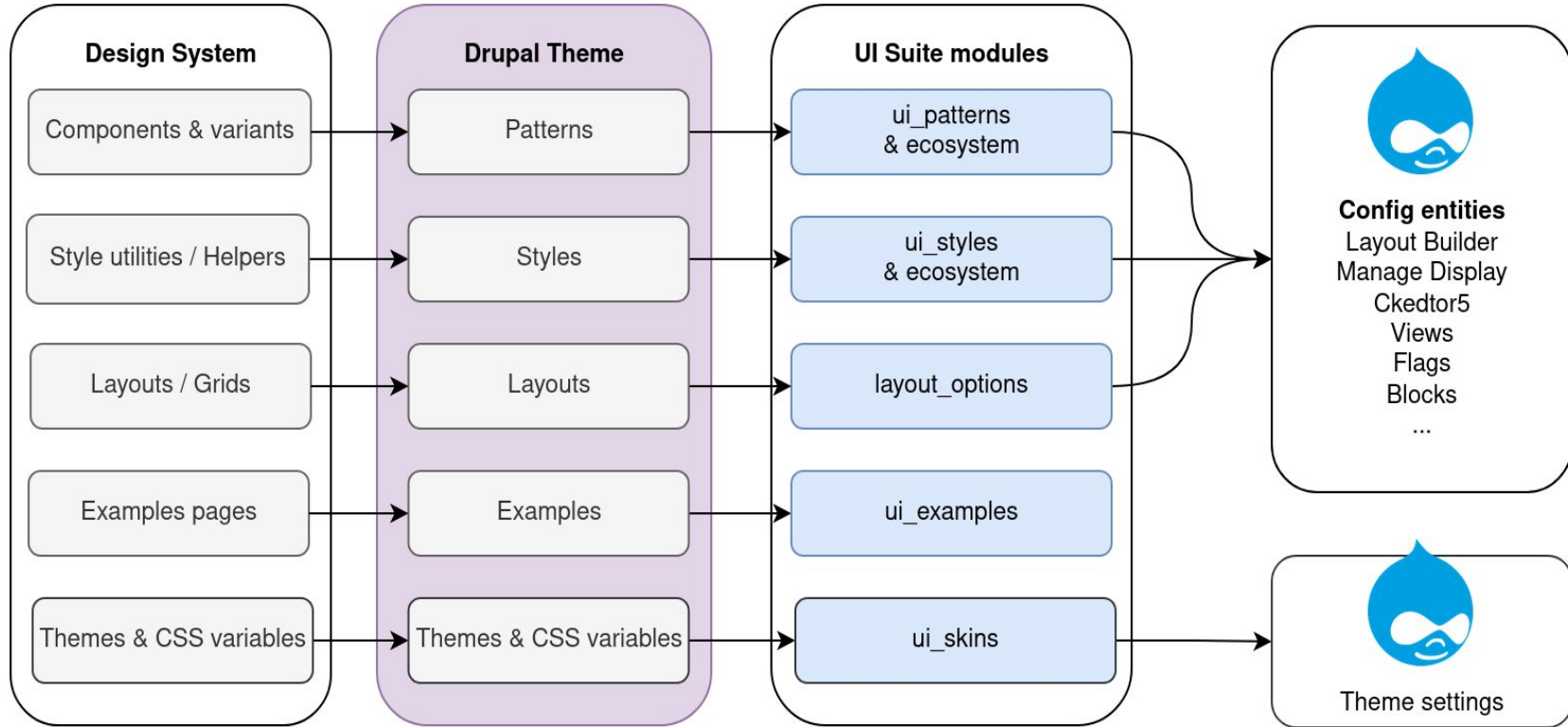
Examples pages



Themes & CSS variables



Ecosystem



Workflow inversion

“Classic” theming

Site building & back
development

Provides markup, templates
naming and CSS selectors

Theming, with preprocesses
and template suggestions

UI Suite theming

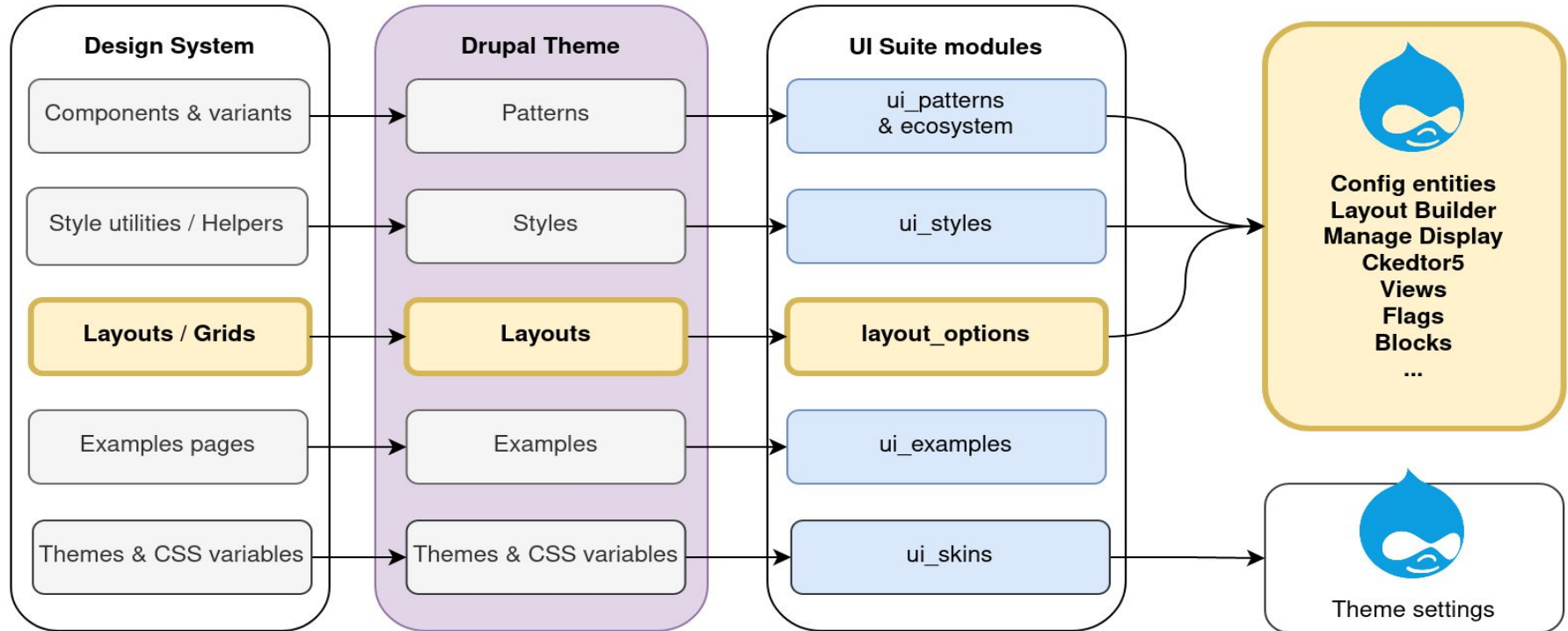
Theming implementation of
a design system

Provides configurable
plugins and render
elements

Site building & back
development

The diagram illustrates a grid layout with 4 columns. The first two columns are labeled `.g-col-6`. The last two columns are labeled `Side` and `Main`. Below the grid, there are three buttons labeled `Three columns`, `Three columns`, and `Three columns`.

Layouts



Class annotation

```
/**
 * @Layout(
 *   id = "my_layout",
 *   label = @Translation("My Layout"),
 *   template = "templates/my-layout",
 * )
 */
class MyLayout extends LayoutBase {
  public function defaultConfiguration() {}
  public function buildConfigurationForm($form,
    $form_state) {}
  public function
  submitConfigurationForm(&$form, $form_state) {}
}
```

YML files

layout_discovery.layouts.yml

```
layout_onecol:
  label: 'One column'
  path: layouts/onecol
  template: layout--onecol
  library: layout_discovery/onecol
  category: 'Columns: 1'
  default_region: content
  icon_map:
    - [content]
  regions:
    content:
      label: Content
```

Layout Options

Not all plugin discoveries are equals.

Only PHP classes annotation allows layout configuration.

For optimum frontend developer experience, let's avoid PHP.

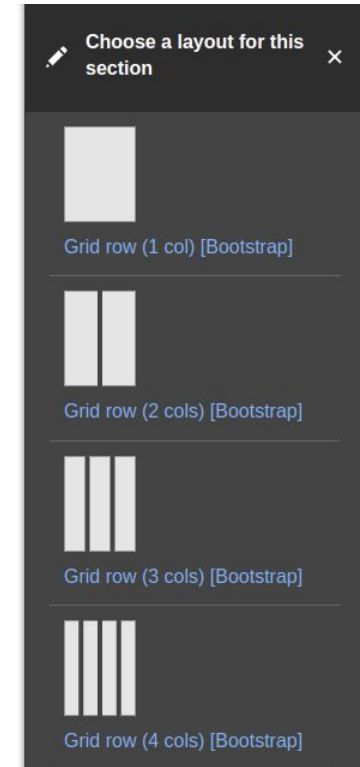
[Layout Options](#) to the rescue!

2 distinct YAML files:

```
YML ui_suite_bootstrap.layout_options.yml  
YML ui_suite_bootstrap.layouts.yml
```

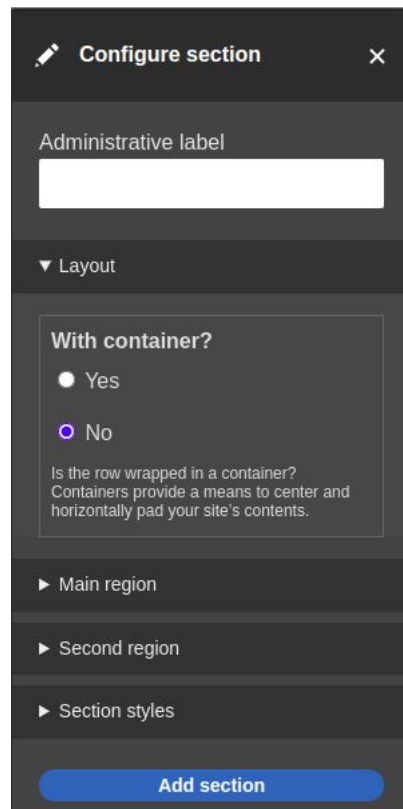
*.layouts.yml

```
bootstrap_grid_row_3:
  label: "Grid row (3 cols) [Bootstrap]"
  path: templates
  template: layout--grid
  class: '\Drupal\layout_options\Plugin\Layout\LayoutOptions'
  category: "Columns: 3"
  default_region: main
  icon_map:
    - [main, second, third]
  regions:
    main:
      label: Main
    second:
      label: Second
    third:
      label: Third
```



*.layout_options.yml

```
layout_option_definitions:  
  with_container:  
    title: "With container?"  
    description: "Is the row wrapped in a container?"  
    default: ""  
    plugin: "layout_options_class_radios"  
    multi: false  
    layout: true  
    regions: false  
    inline: true  
    options:  
      "with-container": "Yes"  
      "": "No"
```



Configure section ×

Administrative label

▼ Layout

With container?

☐ Yes

☒ No

Is the row wrapped in a container?
Containers provide a means to center and horizontally pad your site's contents.

► Main region

► Second region

► Section styles

Add section

With fixed number of regions

Define your **layouts** in my_theme.layouts.yml

Usually 4 layouts (1, 2, 3, 4 columns),

Define your **layout options** in my_theme.layout_options.yml

- classes for each regions/columns VS the ones for the layout itself
- for potential wrapper/container div, parse the layout classes

```
{% set with_container = attributes.hasClass('with-container') %}  
{% set attributes = attributes.removeClass('with-container') %}
```

```
{% if with_container %}  
  <div class="container">  
{% endif %}
```

Write your **layout template**

In templates/layouts/layout-grid.html.twig

With dynamic number of regions

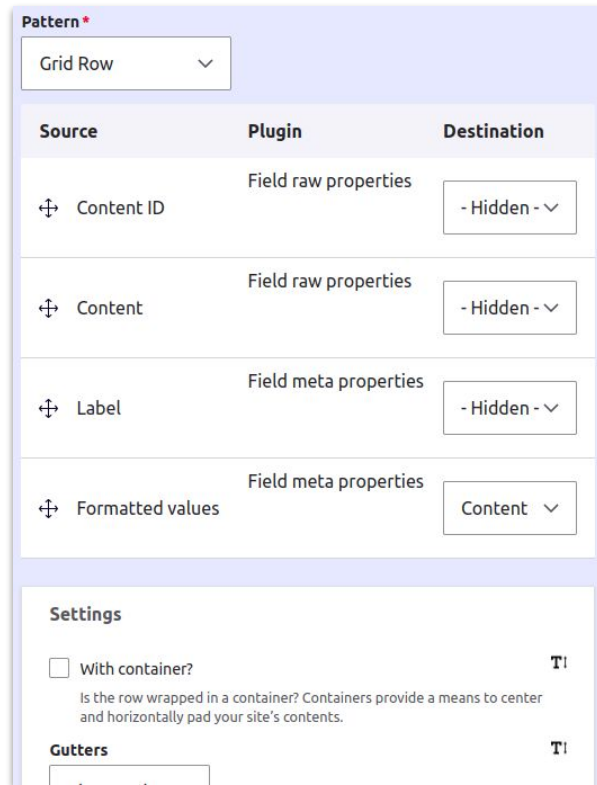
Views styles plugins

Entity reference field formatters
plugins.

The number of regions depends on the
number of queried items.





Define a “row” component with:

- layout options as component props (settings)
- layout regions as component slots (field)



Pattern *

Grid Row

Source	Plugin	Destination
 Content ID	Field raw properties	- Hidden -
 Content	Field raw properties	- Hidden -
 Label	Field meta properties	- Hidden -
 Formatted values	Field meta properties	Content

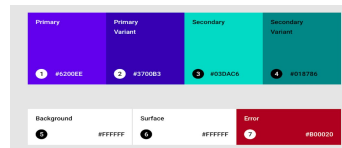
Settings

☐ With container? T!

Is the row wrapped in a container? Containers provide a means to center and horizontally pad your site's contents.

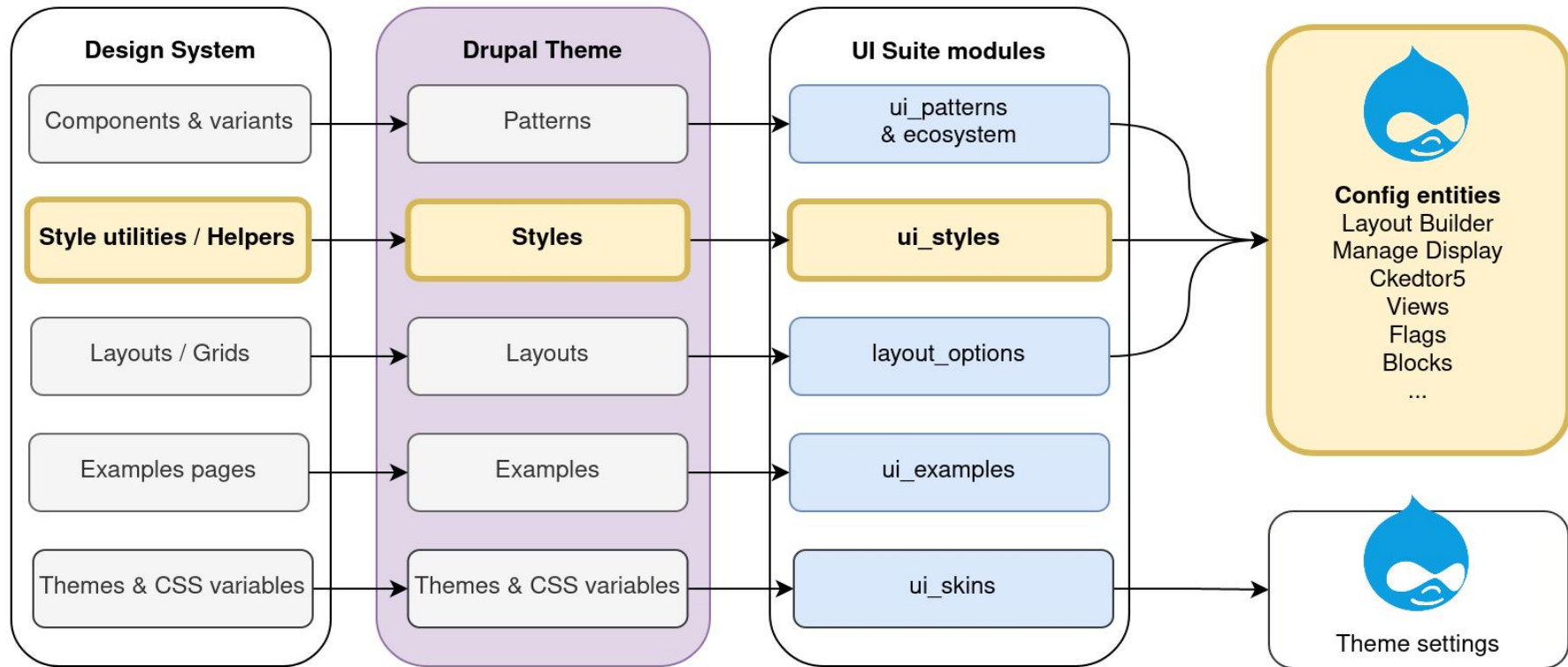
Gutters T!

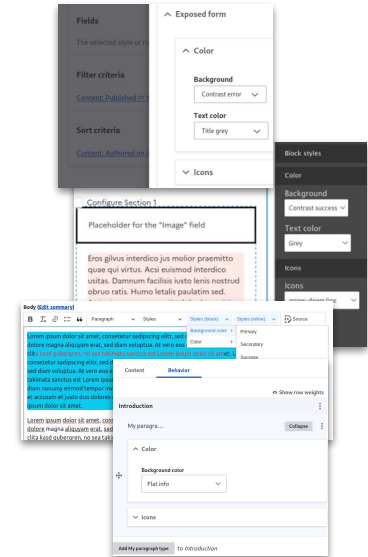
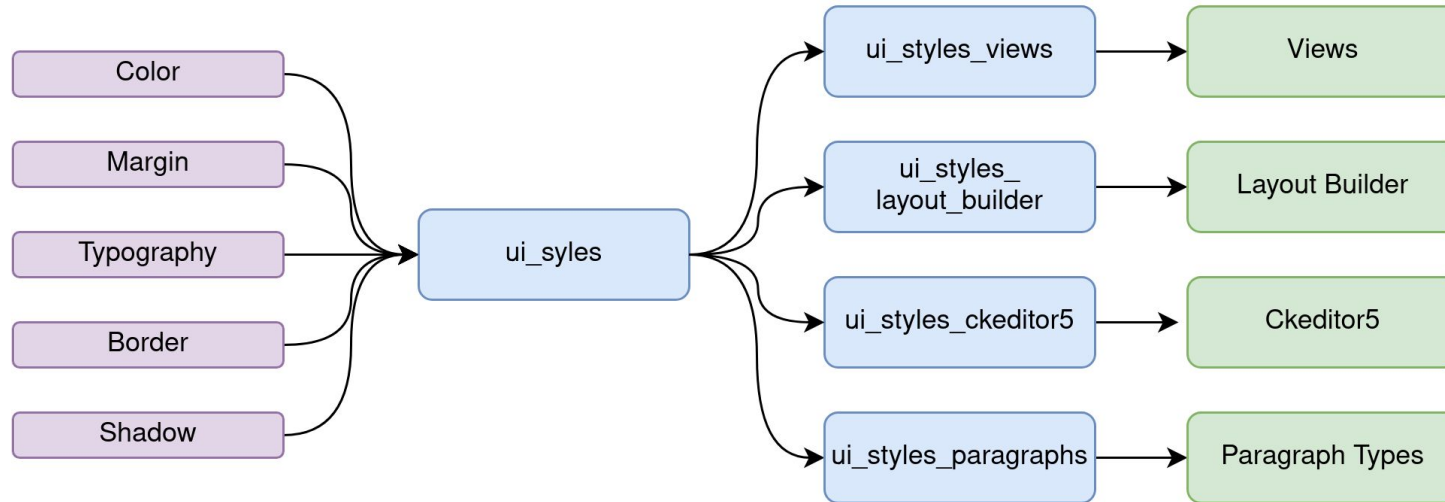
Styles utilities & helpers



<div></div> hsl(142, 52%, 96%)	Hello Bulma
<div></div> hsl(48, 100%, 96%)	Hello Bulma
<div></div> hsl(347, 90%, 96%)	Hello Bulma

Styles utilities & helpers





*.ui_styles.yml

```
colors_color:
  category: "Text"
  label: "Color"
  options:
    text-primary: "Primary"
    text-primary-emphasis: "Primary
emphasis"
    text-secondary: "Secondary"
    text-secondary-emphasis: "Secondary
emphasis"
    text-success: "Success"
    text-success-emphasis: "Success
emphasis"
    text-danger: "Danger"
    text-danger-emphasis: "Danger emphasis"
    text-warning: "Warning"
    text-warning-emphasis: "Warning
emphasis"
    text-info: "Info"
    text-info-emphasis: "Info emphasis"
    text-light: "Light"
    text-light-emphasis: "Light emphasis"
    ...
```

Color

Convey meaning through color with a handful of color utility classes.

Primary (text-primary)

Primary emphasis (text-primary-emphasis)

Secondary (text-secondary)

Secondary emphasis (text-secondary-emphasis)

Success (text-success)

Success emphasis (text-success-emphasis)

Danger (text-danger)

Danger emphasis (text-danger-emphasis)

Warning (text-warning)

Warning emphasis (text-warning-emphasis)

Info (text-info)

Info emphasis (text-info-emphasis)

Light (text-light)

Light emphasis (text-light-emphasis)

*.ui_styles.yml

```
text_font_monospace :  
  category: "Text"  
  label: "Monospace"  
  options:  
    font-monospace: "Monospace"
```

```
colors_opacity:  
  category: "Text"  
  label: "Opacity"  
  options:  
    text-opacity-100: "100%"  
    text-opacity-75: "75%"  
    text-opacity-50: "50%"  
    text-opacity-25: "25%"  
  previewed_with:  
    - text-primary
```

Monospace

Change a selection to the monospace font stack.

Monospace (font-monospace)

Opacity

100% (text-opacity-100)

75% (text-opacity-75)

50% (text-opacity-50)

25% (text-opacity-25)

Styles are sets of universal HTML classes

Each style option is a self-descriptive, single-purpose CSS class.

To apply on any element to handle generic CSS effects like background color, text effects, spacing, etc.

If specific context or markup expected, it may not be a style but:

1. A layout option
2. Or related to a component

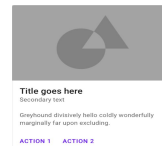
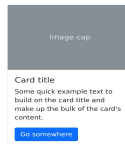
Bootstrap 5 example:

```
<button class="btn btn-outline-secondary  
btn-lg px-4">Default</button>  
  
<ol class="breadcrumb p-3 bg-body-tertiary  
rounded-3">  
  <li class="breadcrumb-item">  
    <a class="link-body-emphasis  
fw-semibold text-decoration-none"  
href="#">Library</a>  
  </li>  
  <li class="breadcrumb-item active"  
aria-current="page">  
    Data  
  </li>  
</ol>
```

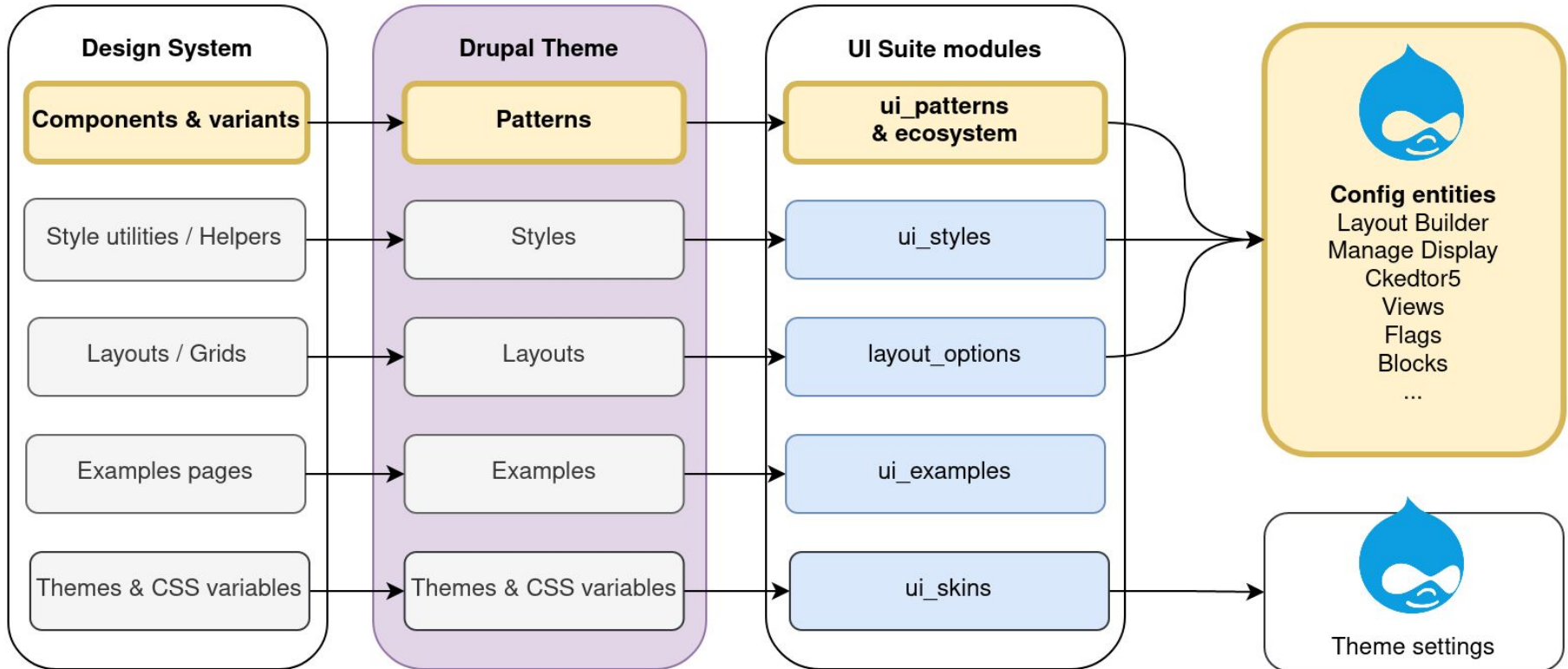
Options are mutually exclusive

Do	Don't
<pre>borders_border : category: "Borders" label: "Border" options: border: "All" border-top: "Top" border-end: "End" border-bottom: "Bottom" border-start: "Start" borders_border_subtractive : category: "Borders" label: "Border subtractive" options: border-0: "All" border-top-0: "Top" border-end-0: "End" border-bottom-0: "Bottom" border-start-0: "Start"</pre>	<pre>borders_border : category: "Borders" label: "Border" options: border: "Additive All" border-top: "Additive Top" border-end: "Additive End" border-bottom: "Additive Bottom" border-start: "Additive Start" border-0: "Subtractive All" border-top-0: "Subtractive Top" border-end-0: "Subtractive End" border-bottom-0: "Subtractive Bottom" border-start-0: "Subtractive Start"</pre>

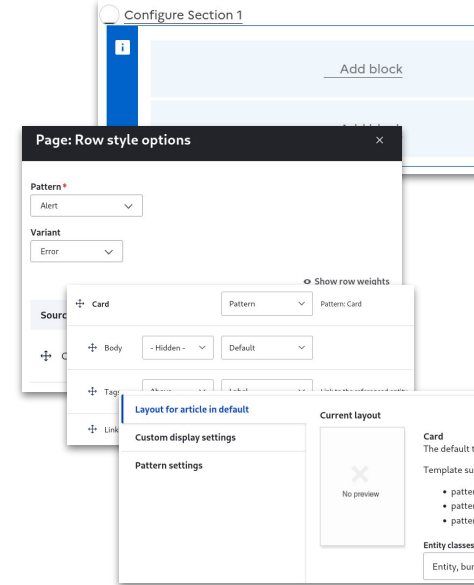
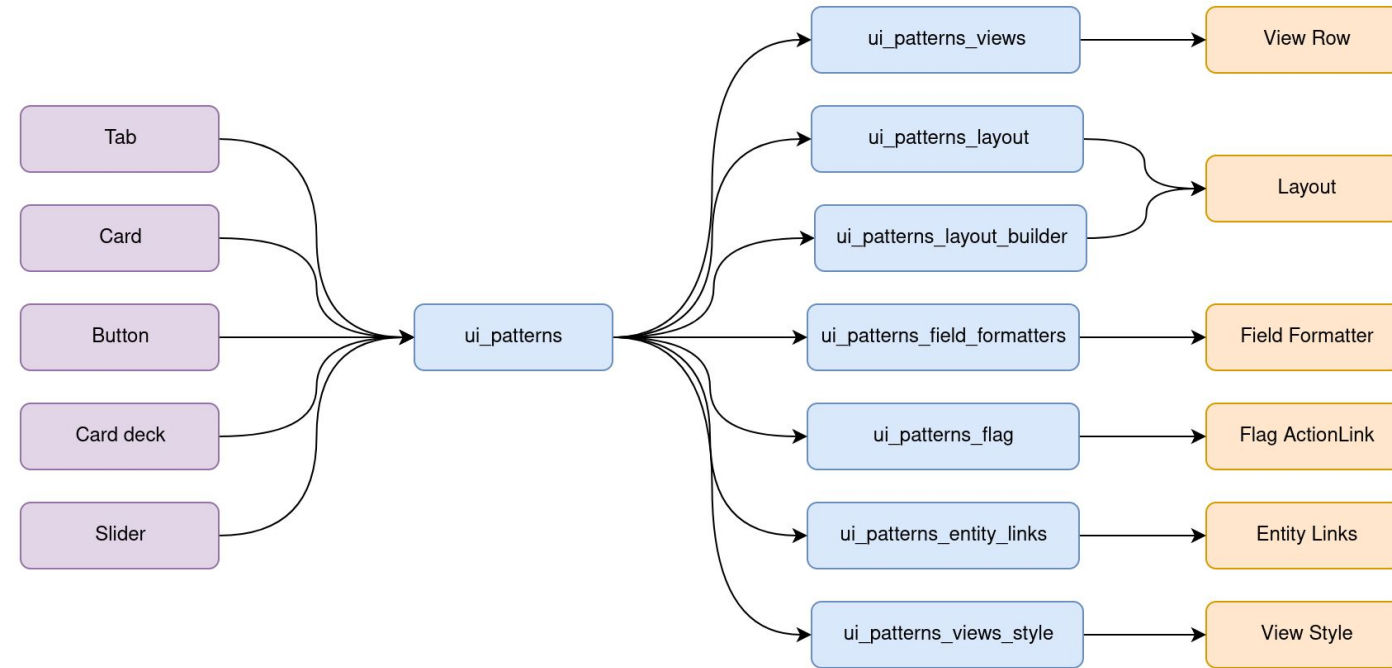
Components & variants



Components & variants



Ecosystem



Terminology

UI Patterns 1.x	Drupal SDC	WebComponents	ReactJS	Vuejs
pattern	component	component	component	component
fields	slots	slots	children	slots
settings	props	attributes	props	props

New terminology for **UI Patterns 2.x**

Variants, slots & props

Variants

A different visual version of the component.

Examples: horizontal card, wide banner...

Slots (fields)

Free, optional, renderable values.

Examples: card body, button label...

Props (settings)

Strict & controlled data structures (boolean, enums, attributes, menu...)

Example: image position, carousel indicator...

offcanvas.ui_patterns.yml

```
offcanvas:  
  label: "Offcanvas"  
  description: "Build hidden  
sidebars into your project for  
navigation...."  
  category: "Dialog"  
  variants:  
    ...  
  settings:  
    ...  
  fields:  
    ...  
  libraries:  
    ...
```

Offcanvas

Build hidden sidebars into your project for navigation, shopping carts, and more with a few classes and our JavaScript plugin.
<https://getbootstrap.com/docs/5.3/components/offcanvas/>

Type	Name	Label	Type	Description / Options
Field	<code>title</code>	Title	<code>string</code>	Offcanvas title.
Field	<code>body</code>	Body	<code>render</code>	The content of the offcanvas.
Setting	<code>responsive</code>	Responsive	<code>select</code>	Hide content in offcanvas below the selected breakpoint. Above that breakpoint, the contents within will behave as usual. <ul style="list-style-type: none"><code>offcanvas-sm</code>: Hide below small<code>offcanvas-md</code>: Hide below medium<code>offcanvas-lg</code>: Hide below large<code>offcanvas-xl</code>: Hide below extra large<code>offcanvas-xxl</code>: Hide below extra extra large
Setting	<code>backdrop</code>	Backdrop	<code>select</code>	When backdrop is set to static, the offcanvas will not close when clicking outside of it. <ul style="list-style-type: none"><code>false</code>: No backdrop<code>static</code>: Static
Setting	<code>scroll</code>	Body scrolling	<code>boolean</code>	By default, body scrolling is disabled.
Setting	<code>heading_level</code>	Heading level	<code>select</code>	Heading level of the offcanvas. <ul style="list-style-type: none">1: h12: h23: h34: h45: h5 (Default)6: h6
Setting	<code>offcanvas_id</code>	ID	<code>textfield</code>	ID used by external buttons to toggle the visibility.

Start (start)

Launch offcanvas

End (end)

Launch offcanvas

Variants declaration

```
offcanvas:  
  label: "Offcanvas"  
  variants:  
    start:  
      label: "Start"  
    end:  
      label: "End"  
    top:  
      label: "Top"  
    bottom:  
      label: "Bottom"
```

Offcanvas

Build hidden sidebars into your project for navigation, shopping carts, and more with a few classes and our JavaScript plugin.
<https://getbootstrap.com/docs/5.3/components/offcanvas/>

Type	Name	Label	Type	Description / Options
Field	<code>title</code>	Title	<code>string</code>	Offcanvas title.
Field	<code>body</code>	Body	<code>render</code>	The content of the offcanvas.

Start (`start`)

Launch offcanvas

End (`end`)

Launch offcanvas

Slots (fields) declaration

```
offcanvas:  
  label: "Offcanvas"  
  fields:  
    title:  
      type: "string"  
      label: "Title"  
      description: "Offcanvas  
title."  
      preview: "Offcanvas"  
    body:  
      type: "render"  
      label: "Body"  
      description: "The content of  
the offcanvas."  
      preview: "Content for the  
offcanvas goes here..."
```

Offcanvas

Build hidden sidebars into your project for navigation, shopping carts, and more with a few classes and our JavaScript plugin.
<https://getbootstrap.com/docs/5.3/components/offcanvas/>

Type	Name	Label	Type	Description / Options
Field	title	Title	string	Offcanvas title.
Field	body	Body	render	The content of the offcanvas.
Setting	responsive	Responsive	select	Hide content in offcanvas below the selected breakpoint. Above that breakpoint, the contents within will usual. <ul style="list-style-type: none">offcanvas-sm: Hide below smalloffcanvas-md: Hide below mediumoffcanvas-lg: Hide below largeoffcanvas-xl: Hide below extra largeoffcanvas-xxl: Hide below extra extra large
Setting	backdrop	Backdrop	select	When backdrop is set to static, the offcanvas will not close when clicking outside of it. <ul style="list-style-type: none">false: No backdropstatic: Static
Setting	scroll	Body scrolling	boolean	By default, body scrolling is disabled.
Setting	heading_level	Heading level	select	Heading level of the offcanvas. <ul style="list-style-type: none">1: h12: h23: h34: h45: h5 (Default)6: h6
Setting	offcanvas_id	ID	textfield	ID used by external buttons to toggle the visibility.

Start (start)

Launch offcanvas

End (end)

Launch offcanvas

Props (settings) declaration

```
offcanvas :
  label: "Offcanvas"
  settings:
    responsive :
      type: "select"
      label: "Responsive"
      description: "..."
      options:
        offcanvas-sm: "Hide below small"
        offcanvas-md: "Hide below
medium"
        offcanvas-lg: "Hide below large"
        offcanvas-xl: "Hide below extra
large"
        offcanvas-xxl: "Hide below extra
extra large"
    allow_expose: true
    allow_token: true
```

...

Offcanvas

Build hidden sidebars into your project for navigation, shopping carts, and more with a few classes and our JavaScript plugin.
<https://getbootstrap.com/docs/5.3/components/offcanvas/>

Type	Name	Label	Type	Description / Options
Field	title	Title	string	Offcanvas title.
Field	body	Body	render	The content of the offcanvas.
Setting	responsive	Responsive	select	Hide content in offcanvas below the selected breakpoint. Above that breakpoint, the contents within will behave as usual. <ul style="list-style-type: none"> offcanvas-sm: Hide below small offcanvas-md: Hide below medium offcanvas-lg: Hide below large offcanvas-xl: Hide below extra large offcanvas-xxl: Hide below extra extra large
Setting	backdrop	Backdrop	select	When backdrop is set to static, the offcanvas will not close when clicking outside of it. <ul style="list-style-type: none"> false: No backdrop static: Static
Setting	scroll	Body scrolling	boolean	By default, body scrolling is disabled.
Setting	heading_level	Heading level	select	Heading level of the offcanvas. <ul style="list-style-type: none"> 1: h1 2: h2 3: h3 4: h4 5: h5 (Default) 6: h6
Setting	offcanvas_id	ID	textfield	ID used by external buttons to toggle the visibility.

Start (start)

Launch offcanvas

End (end)

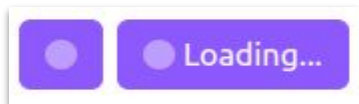
Launch offcanvas

Components & variants Identification

Components identification

Upstream documentation may show nested component markup.

Usually, look for the HTML tag with the component name as a class.



```
<button class="btn btn-primary" type="button" disabled>
  <span class="spinner-grow spinner-grow-sm" role="status" aria-hidden="true"></span>
  <span class="visually-hidden">Loading...</span>
</button>
<button class="btn btn-primary" type="button" disabled>
  <span class="spinner-grow spinner-grow-sm" role="status" aria-hidden="true"></span>
  Loading...
</button>
```

Components identification

Upstream documentation may contain several components: [Navbar & navbar nav](#)

```
<nav class="navbar navbar-expand-lg bg-body-tertiary">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" data-bs-whatever="#"></button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a href="#">Home</a>
        </li>
        <li class="nav-item">
          <a href="#">About</a>
        </li>
        <li class="nav-item dropdown">
          <a href="#">Services</a>
          <ul>
            <li>Service 1</li>
            <li>Service 2</li>
            <li>Service 3</li>
          </ul>
        </li>
        <li class="nav-item">
          <a href="#">Contact</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

Sub components

Extract sub components for site building logic. Example: [Accordion](#)

```
<div class="accordion accordion-flush" id="accordionFlushExample">  
  <div class="accordion-item">  
    <h2 class="accordion-header">  
      <button class="accordion-button collapsed" type="button" data-  
        Accordion Item #1  
      </button>  
    </h2>  
    <div id="flush-collapseOne" class="accordion-collapse collapse">  
      <div class="accordion-body">Placeholder content for this acc  
    </div>  
  </div>  
  <div class="accordion-item"...>  
  <div class="accordion-item"...>  
</div>
```

Trust the code, not the doc

Upstream documentation may be confusing because their main target is the design system user, not the design system implementer.

```
<table class="table table-success table-striped-columns">  
  ...  
</table>
```



Content

Reboot

Typography

Images

Tables

Figures



Components

Accordion

Alerts

Badge

Breadcrumb

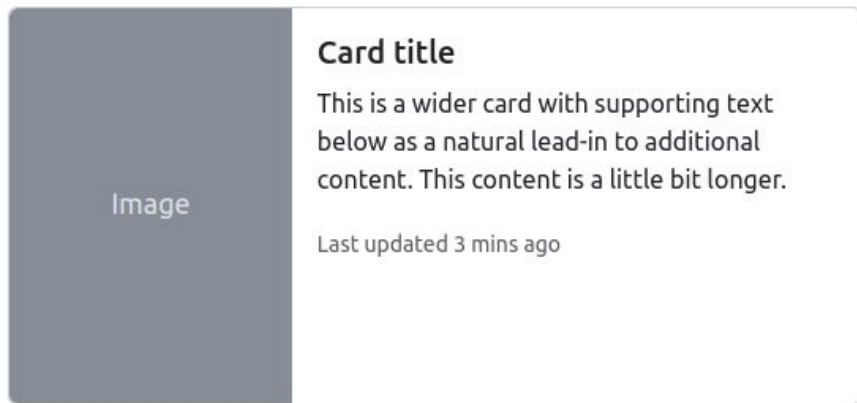
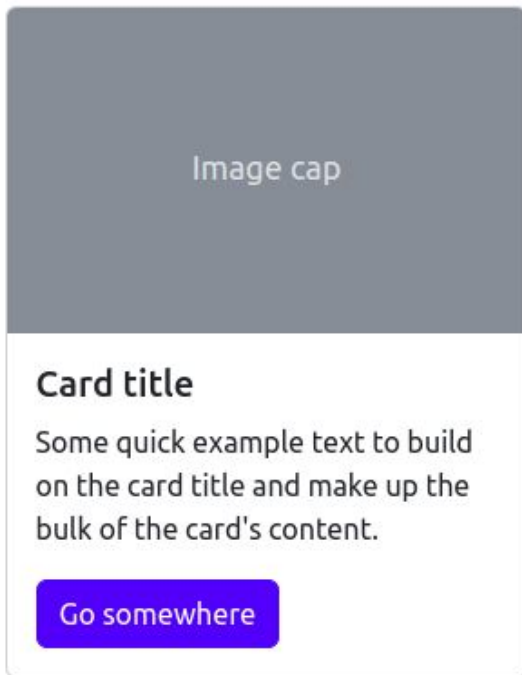
Buttons

Button group

Card

Variants identification

The component presents different visual variations. Example: [Horizontal card](#)



Variants identification

Variants classes derived from the component base class.

```
<button type="button" class="btn btn-primary btn-lg">Large button</button>  
<button type="button" class="btn btn-secondary btn-lg">Large button</button>
```

```
<button type="button" class="btn btn-primary btn-sm">Small button</button>  
<button type="button" class="btn btn-secondary btn-sm">Small button</button>
```

```
light:  
  label: "Light"  
dark:  
  label: "Dark"  
link:  
  label: "Link"  
primary__lg:  
  label: "Primary large"  
secondary__lg:  
  label: "Secondary large"  
success__lg:  
  label: "Success large"
```

Slots (fields) identification

Look for HTML tags with classes similar to the component class.

Can I replace this markup from documentation by any markup?

Well done!

Aww yeah, you successfully read this important alert message. This example text is going to run a bit longer so that you can see how spacing within an alert works with this kind of content.

Whenever you need to, be sure to use margin utilities to keep things nice and tidy.

```
<div class="alert alert-success" role="alert">
  <h4 class="alert-heading">Well done!</h4>
  <p>Aww yeah, you successfully read this impor
<hr>
  <p class="mb-0">Whenever you need to, be sure
</div>
```

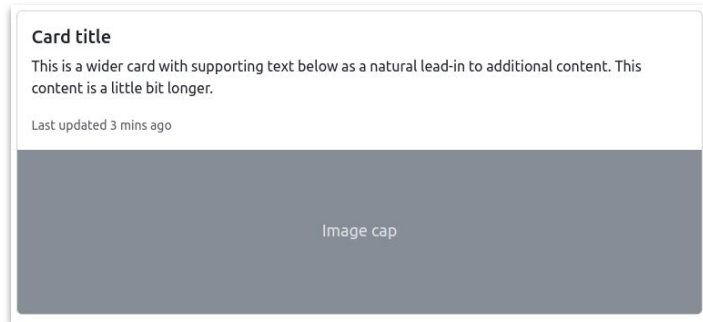
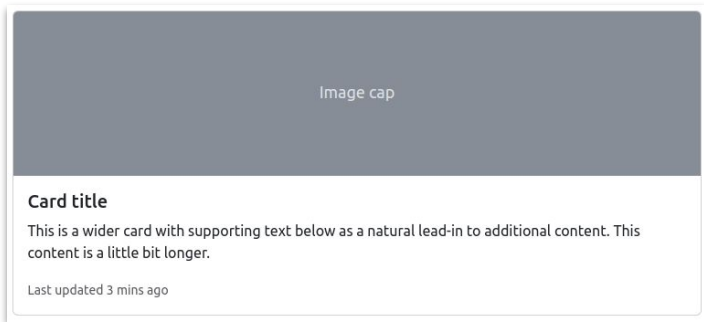
```
<div>{{ attributes.addClass('alert').setAttribute('role', alert) }}>
  {% if heading %}
  <h4 class="alert-heading">{{ heading }}</h4>
  {% endif %}
  {{ message }}
</div>
```

Props (settings) identification

Dynamic data which is not a variant nor a slot.

Examples:

- heading level
- position of the image
- attributes on a specific tag (which is not the component base tag)
- ...



Components & variants: Identification

Example: offcanvas from Bootstrap 5

Component from upstream

First example from [Offcanvas](#) documentation:

```
<div class="offcanvas offcanvas-start show"
tabindex="-1" id="offcanvas"
aria-labelledby="offcanvasLabel">
  <div class="offcanvas-header">
    <h5 class="offcanvas-title"
id="offcanvasLabel">
      Offcanvas
    </h5>
    <button type="button" class="btn-close"
data-bs-dismiss="offcanvas"
aria-label="Close"></button>
  </div>
  <div class="offcanvas-body">
    Content for the offcanvas goes here.
  </div>
</div>
```

Offcanvas



Content for the offcanvas goes here. You can place just about any Bootstrap component or custom elements here.

Component from upstream

Offcanvas

Second example from documentation:

```
<a class="btn btn-primary" data-bs-toggle="offcanvas" href="#offcanvasExample" role="button"
aria-controls="offcanvasExample" >
```

Link with href

```
</a>
```

```
<button class="btn btn-primary" type="button" data-bs-toggle="offcanvas"
data-bs-target="#offcanvasExample" aria-controls="offcanvasExample" >
```

Button with data-bs-target

```
</button>
```

```
<div class="offcanvas offcanvas-start" tabindex="-1" id="offcanvasExample"
aria-labelledby="offcanvasExampleLabel" >
```

```
<div class="offcanvas-header" >...</div>
```

```
<div class="offcanvas-body" >...</div>
```

```
</div>
```

Identify the component

```
<div class="offcanvas offcanvas-start show" tabindex="-1" id="offcanvas"
aria-labelledby="offcanvasLabel">
  <div class="offcanvas-header">
    <h5 class="offcanvas-title" id="offcanvasLabel">Offcanvas</h5>
    <button type="button" class="btn-close" data-bs-dismiss="offcanvas"
aria-label="Close"></button>
  </div>
  <div class="offcanvas-body">
    Content for the offcanvas goes here.</div>
</div>
```


Identify the variants

Responsive section:

offcanvas

← Component base class

.offcanvas-sm
.offcanvas-md
.offcanvas-lg
.offcanvas-xl
.offcanvas-xxl

Placement section:

.offcanvas-start
.offcanvas-end
.offcanvas-top
.offcanvas-bottom

```
<div class="offcanvas-lg offcanvas-end"  
...>  
...  
</div>
```

So it is additive, and for responsive, the component base class is removed.

2 possibilities:

- “inner join” to generate variants:
 - offcanvas_sm__offcanvas_start
 - etc.
- **Use placement as variants as it is mandatory and Responsive as prop**

Identify the slots (fields)

```
<div class="offcanvas offcanvas-start show" >  
  <div class="offcanvas-header" >  
    <h5 class="offcanvas-title" >Offcanvas</h5>  
    <button type="button" class="btn-close" data-bs-dismiss="offcanvas"  
aria-label="Close"></button>  
  </div>  
  <div class="offcanvas-body" >  
    Content for the offcanvas goes here. You can place just about any Bootstrap component or  
    custom elements here.  
  </div>  
</div>
```



Identify the props (settings)

Body Scrolling

```
<div class="offcanvas ..." data-bs-scroll="true" data-bs-backdrop="false" ...>
```

Body Scrolling and backdrop

```
<div class="offcanvas ..." data-bs-scroll="true" ...>
```

Static backdrop

```
<div class="offcanvas ..." data-bs-backdrop="static" ...>
```

Avoid props paradoxes by leveraging orthogonality.

Even if not shown explicitly by the documentation, the 2 data attributes are independent.

Identify the props (settings)

```
settings:
  responsive:
    type: "select"
    label: "Responsive"
    options:
      offcanvas-sm: "Hide below small"
      offcanvas-md: "Hide below medium"
      offcanvas-lg: "Hide below large"
      offcanvas-xl: "Hide below extra large"
      offcanvas-xxl: "Hide below extra large"
  backdrop:
    type: "select"
    label: "Backdrop"
    options:
      "false": "No backdrop"
      static: "Static"
  scroll:
    type: "boolean"
    label: "Body scrolling"
    preview: false
  heading_level:
    type: "select"
    label: "Heading level"
    options:
      1: "h1"
      2: "h2"
      3: "h3"
      4: "h4"
      5: "h5 (Default)"
      6: "h6"
    preview: 5
  offcanvas_id:
    type: "textfield"
    label: "ID"
```

From variants identification step

2 orthogonal data attributes => 2 props

Possibility to control heading level, arbitrary choice of implementation

A random ID is generated if not provided

```
<div class="offcanvas offcanvas-start" data-bs-scroll="true" data-bs-backdrop="false"
id="offcanvasScrolling">
  <div class="offcanvas-header">
    <h5 class="offcanvas-title">Offcanvas with body scrolling</h5>
    <button type="button" class="btn-close" data-bs-dismiss="offcanvas"
aria-label="Close"></button>
  </div>
  <div class="offcanvas-body">
    <p>Try scrolling the rest of the page to see this option in action.</p>
  </div>
</div>
```

Slots

Props

Variant

Pattern

Components & variants

Twig templating

Variants

Always use the attributes object:

```
<div>{{attributes}}>
```

A snippet for adding variants as HTML classes:

```
{% set attributes = attributes.addClass('btn').addClass('btn-' ~  
variant|lower|replace({'_': '-'})) %}
```

The same, hardcore version, for specific use cases:

```
{% set variants = variant|split('__')|map(v =>  
  v|lower|replace((v): 'btn-' ~ v)|replace({'_': '-'})) %}  
{% set attributes = attributes.addClass('btn').addClass(variants) %}
```

Don't loop when no markup wrapper

```
{% for slide in slides %}  
{{ slide }}  
{% endfor %}
```

Do this instead:

```
{{ slides }}
```

Careful with looping

```
{% if loop.first and slide is  
iterable %}
```

Don't put slots into HTML attributes

```
<span title="<h1>Lorem</h1>">  
Foo</span>
```

Don't rely on slots typing, it is for
documentation only

Use default values when props are “mandatory”

```
{% set heading_level =  
heading_level|default( 5) %}
```

Replace shorthand ternary operators by the default filter:

```
{% set heading_level =  
heading_level ?: 5 %}
```

No useless tests: `!= ''`, is empty, is defined...

```
{% if with_controls %}
```

No default values for boolean

- FALSE and NULL are evaluated the same for Twig.
- Boolean props are FALSE by default.

Keep components stateless

Data is pushed to component, never pulled from components

- No Drupal route
- No call to external services

No complex objects or Drupal API in templates.

No dependencies between components

Don't: HTML markup

```
<article{{attributes.addClass('card')}}>
  {{ body }}
  <a href="{{ link_url }}" class="btn
btn-primary">{{ link_text }}</a>
</article>
```

Don't: Twig include

```
<article{{attribute.addClass('card')}}>
  {{ body }}
  {% include '../button/button.html.twig'
with {
  'url': 'link_url', 'text': 'link_text',
} only %}
</article>
```

Do: Inject nested components into slots.

```
<article{{attribute.addClass('card')}}>
  {{ body }}
</article>
```

For unavoidable hardcoded dependencies, use `pattern()` function.

```
{{ pattern('close_button', {
  attributes: create_attribute({
    'data-bs-dismiss': 'offcanvas',
    'data-bs-target': '#' ~ offcanvas_id
  })
}) }}
```

No hardcoded ID values

Declare them as prop and inject values from outside.

```
settings:
  modal_id:
    type: "textfield"
    label: "ID"
    description: "ID used by external buttons to toggle the visibility."
```

Use Twig random() function.

```
{% set modal_id = modal_id|default("modal-" ~ random()) %}
```

Components & variants: Site building

Example: Bootstrap 5 Carousel

Each component in a view mode display

Production
team

field_members

Team
member

Formatter *

Pattern (one for all) ▾

Formatter

Field formatter from view display ▾

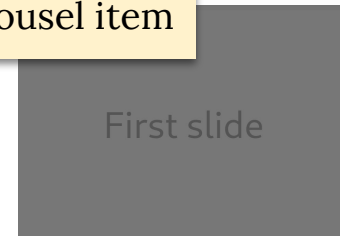
Configure Section 1

"Image" field	"Body" field
"Tags" field	"Authored by" field
"Link" field	"Authored by" field
"Introduction" field	Add block
Add block	

Carousel



Carousel item



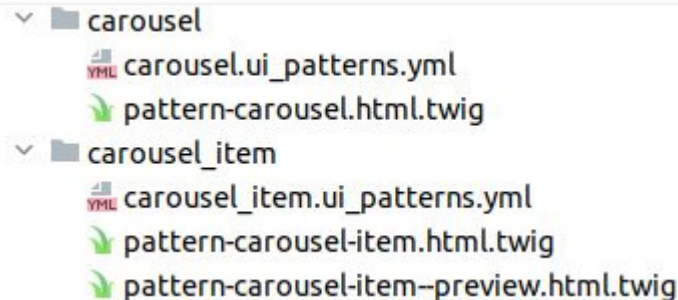
Keep components flat

Carousel

```
<div class="carousel slide">
  <div class="carousel-inner">
    <div class="carousel-item active">
      
      <div class="carousel-caption">
        ...
      </div>
    ...
  </div>
  <div class="carousel-item">
    
  </div>
  ...
</div>
...
</div>
```

No nested slots

Extract carousel_item sub-component



Site building issues

Carousel

```
<div class="carousel slide">
  <div class="carousel-inner">
    <div class="carousel-item active" >
      
      <div class="carousel-caption">
        ...
      </div>
    ...
  </div>
  <div class="carousel-item">
    
  </div>
  ...
</div>
...
</div>
```

“active” class should be put by the carousel component.

Bootstrap carousel expects “carousel-item” class to be a direct child of “.carousel-inner”.

So not possible to delegate this div to the sub-component.

Inject attributes from parent

Results: pattern-carousel.html.twig

...

```
<div class="carousel-inner">
  {% for slide in slides %}
    {% if loop.first %}
      {{ slide|add_class('carousel-item', 'active') }}
    {% else %}
      {{ slide|add_class('carousel-item') }}
    {% endif %}
  {% endfor %}
</div>
```

...

Result on the sub-component

Results: pattern-carousel-item.html.twig

It does not have the “carousel-item” class as we may expect!

```
<div{{ attributes|attributes }}>
  {{ image }}
  {% if caption %}
    <div class="carousel-caption">
      {{ caption }}
    </div>
  {% endif %}
</div>
```

Problems could hide anywhere!

Not as easy as it may look!

Currently most advanced public implementation: [UI Suite Bootstrap 5](#)

Can be used to see the latest state of best practices.

Those tips will be published as drupal.org documentation soon.

Any questions?

HOW I IMAGINED
IMPLEMENTING
A DESIGN
SYSTEM WOULD BE



HOW IT
COULD BE



imgflip.com

Thank you for joining us!

Stay connected with Drupal Austria for more exciting events, news, and updates!

Also visit our website: <https://ddd23.drupalcamp.at/>



And don't forget to visit our sponsors' stands.

Their support helps make events like this possible!



amazee.io
part of Mirantis



Droptica



agiledrop

Trusted Drupal teammates



Kraut.Hosting
No Clouds Just Sunshine



Druid

drop
solid

