

STYLE UTILITIES

THE NEXT DRUPAL THEMING REVOLUTION



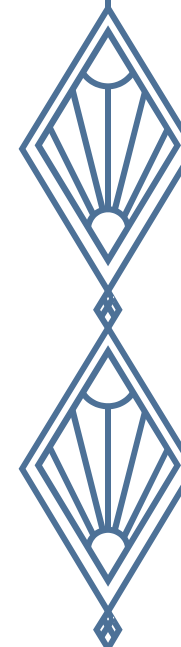
GRIMREAPER - FLORENT TORREGROSA

Tech expert

Drupaler since 2011

UI Suite founder

Core maintainer of Asset Library



EVERYBODY LOVE SDC

“THE BEST THEMING API OF MODERN DRUPAL”

Easy to author

Component driven
development

Pure UI logic & model

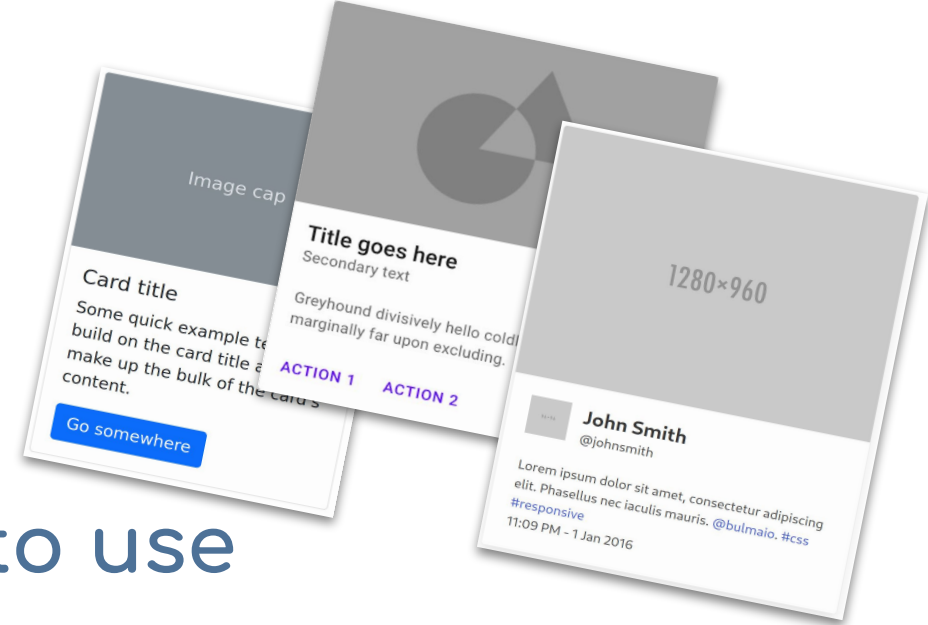
No drupalism. Stateless &
business agnostic

Easy to use

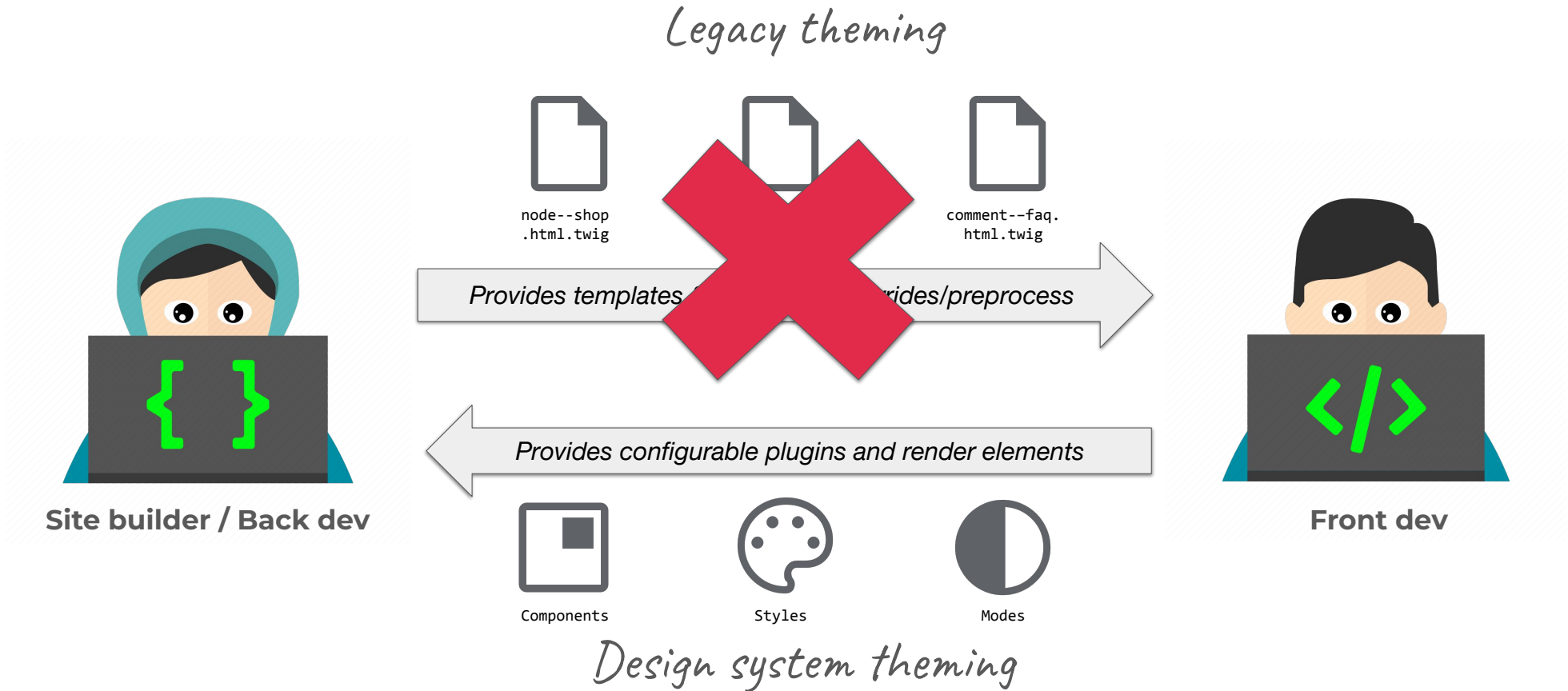
Clear contract between
front & back

Available as plugin &
render elements

Sharable between
projects



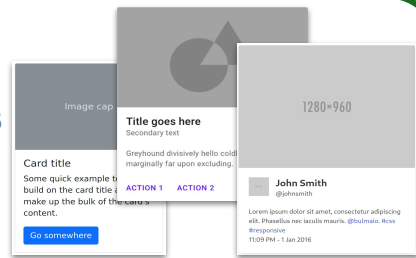
INVERSION OF WORKFLOW



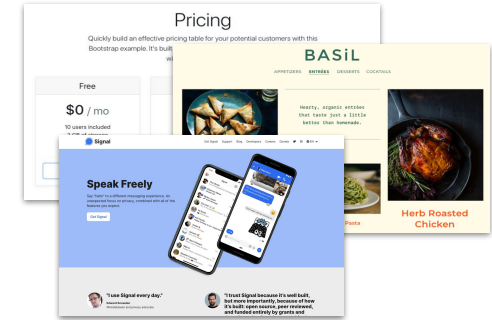
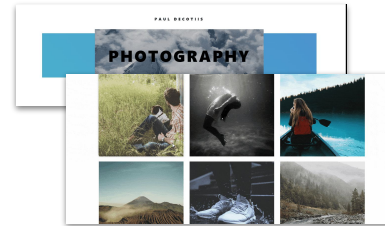
PARTS OF A WEB DESIGN SYSTEM

ALREADY IN CORE

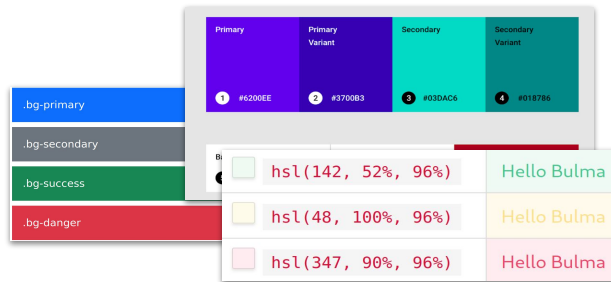
Components & variants



Animations & transitions



Examples

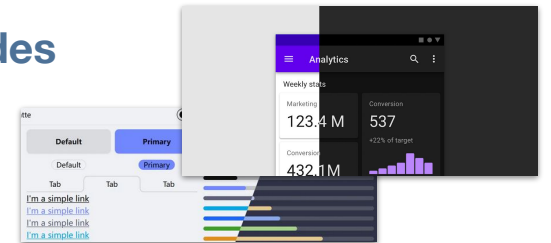


Styles utilities & helpers

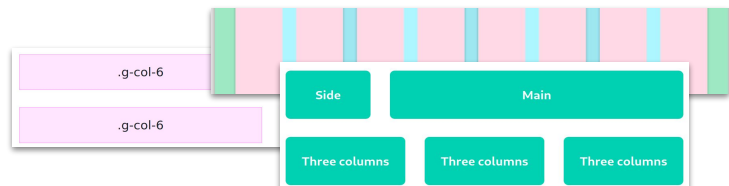


Icon packs

Themes/modes switching



Grid & layouts systems



Variables (runtime tokens)



ARE YOU SATISFIED?



NO



WHAT DO YOU WANT?



MORE DESIGN SYSTEMS API

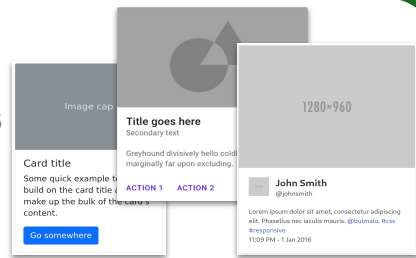


imgflip.com

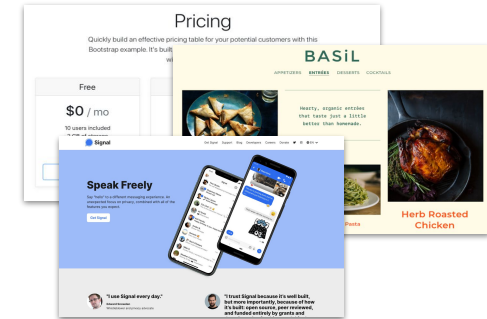
PARTS OF A WEB DESIGN SYSTEM

TODAY - STYLES UTILITIES

Components & variants

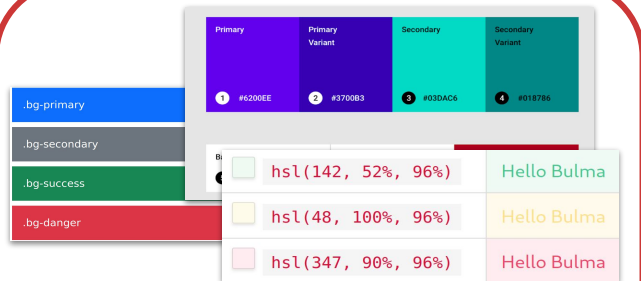


Animations & transitions



Examples

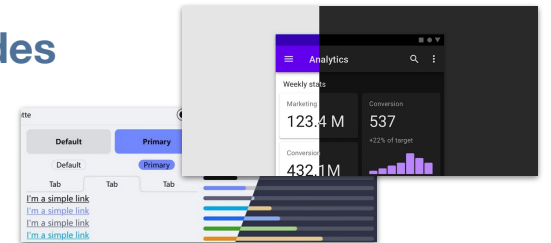
Styles utilities & helpers



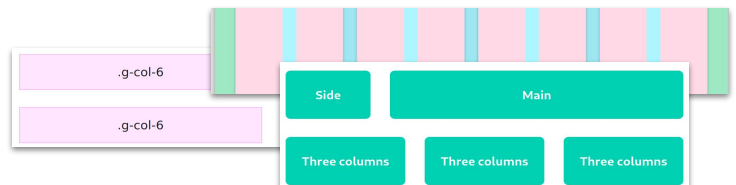
Icon packs



Themes/modes switching



Grid & layouts systems



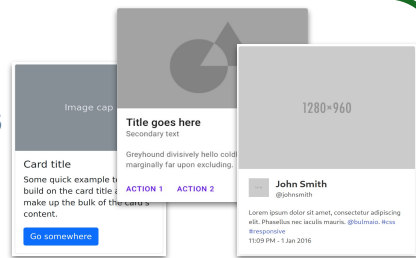
Variables (runtime tokens)

\$blue-400	\$indigo-400	\$purple-400
\$blue-500	\$indigo-500	\$purple-500
\$blue-600	\$indigo-600	\$purple-600
\$blue-700	\$weight-light	font-weight 300
	\$weight-normal	font-weight 400
	\$weight-medium	font-weight 500
	\$weight-sembold	font-weight 600
	\$weight-bold	font-weight 700

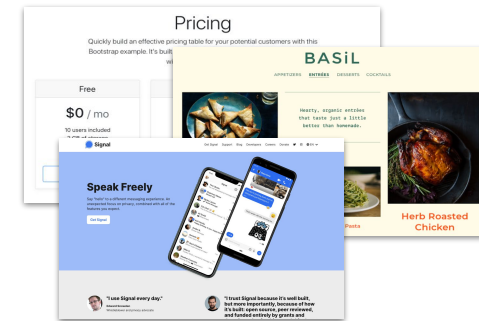
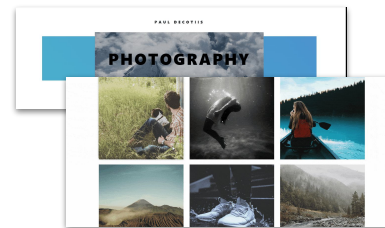
PARTS OF A WEB DESIGN SYSTEM

TOMORROW - DESIGN TOKENS

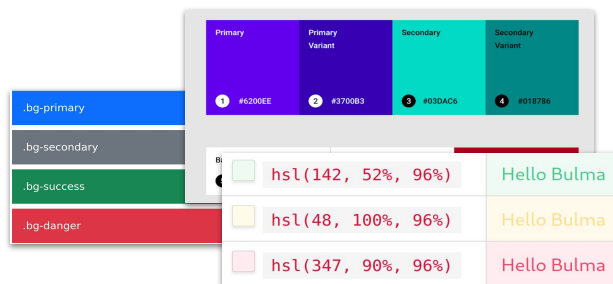
Components & variants



Animations & transitions



Examples

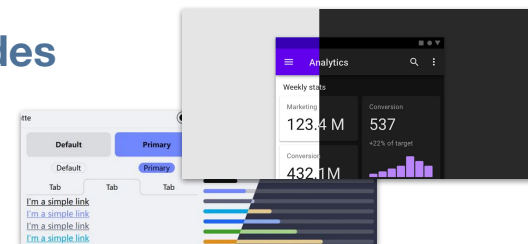


Styles utilities & helpers

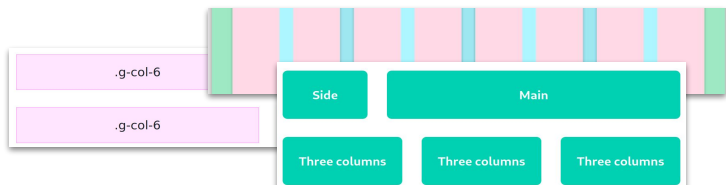


Icon packs

Themes/modes switching



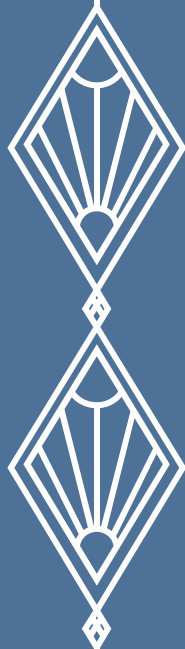
Grid & layouts systems



Variables (runtime tokens)

\$blue-400	\$indigo-400	\$purple-400
\$blue-500	\$indigo-500	\$purple-500
\$blue-600	\$indigo-600	\$purple-600
\$blue-700	\$indigo-700	\$purple-700
\$weight-light	font-weight	300
\$weight-normal	font-weight	400
\$weight-medium	font-weight	500
\$weight-sembold	font-weight	600
\$weight-bold	font-weight	700

01

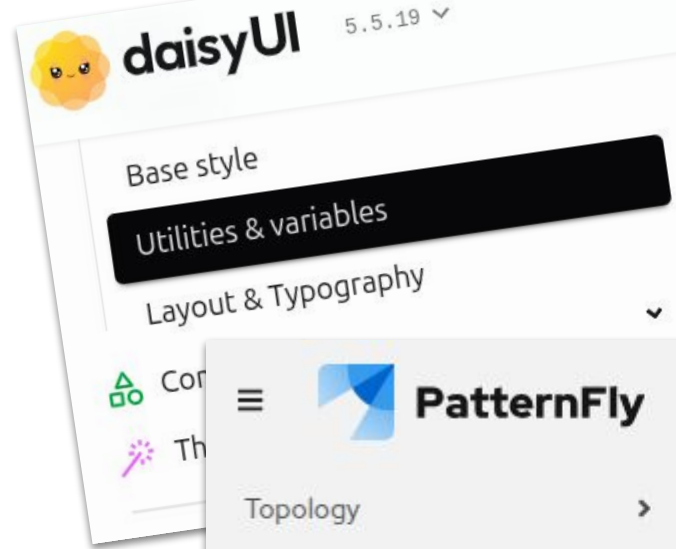
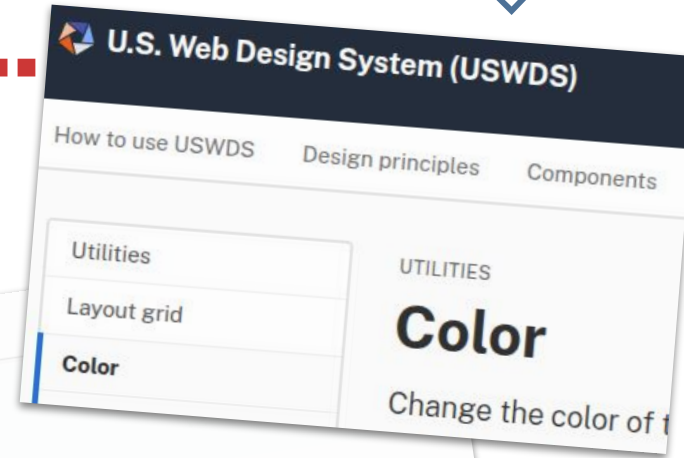


WHAT IS A STYLE?



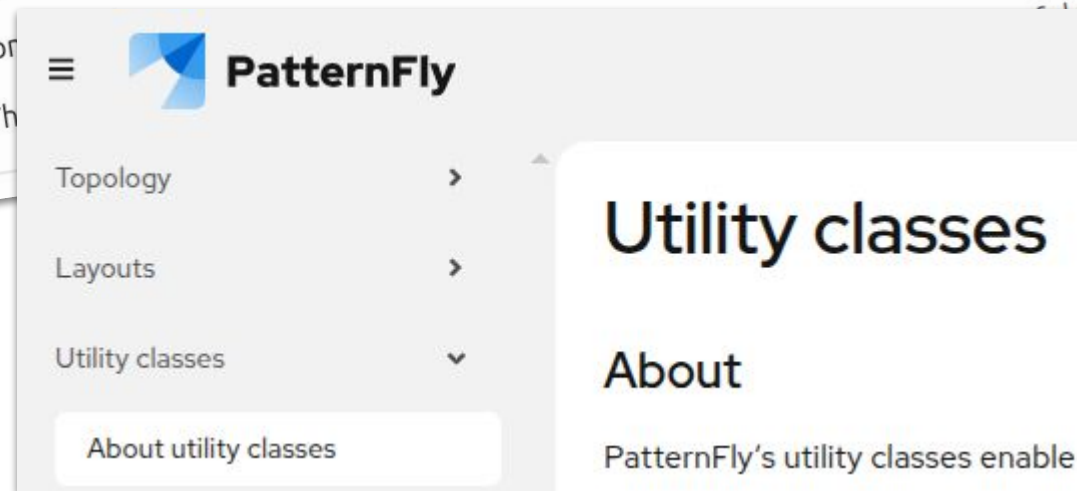
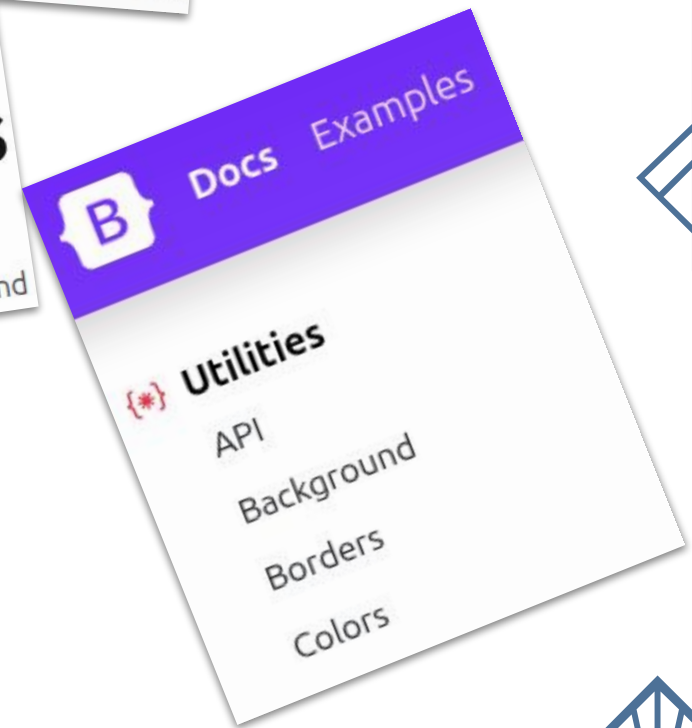
SOMETIMES CALLED...

“UTILITIES”



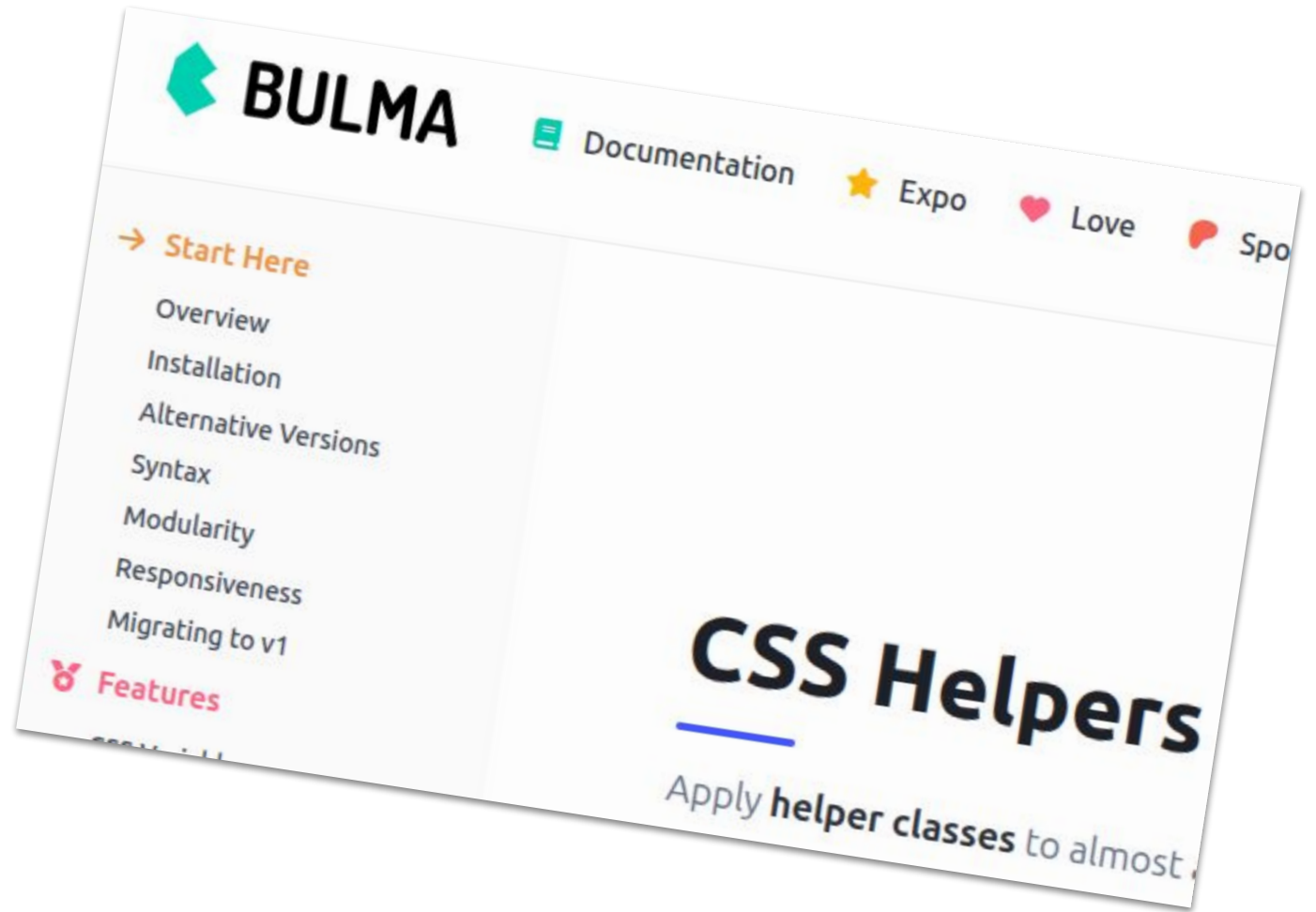
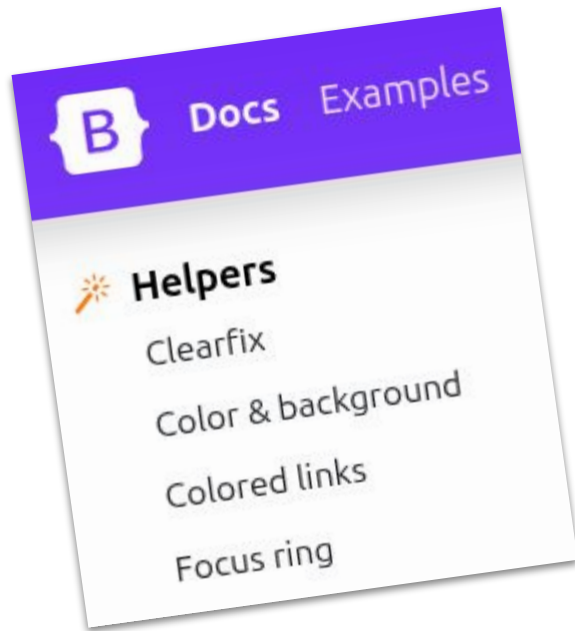
Search...

Utility classes



SOMETIMES CALLED...

“HELPERS”



SOMETIMES CALLED...

“FUNDAMENTALS”



IN A FEW WORDS...

- 👍 Single UI purpose
- 👍 Universal
- 👍 Set of mutually exclusive HTML attributes



EXAMPLE (1)

99% OF THE TIME CSS CLASSES

```
class="bg-primary"  
class="bg-primary-subtle"  
class="bg-secondary"  
class="bg-secondary-subtle"  
...
```

.bg-primary

.bg-primary-subtle

.bg-secondary

.bg-secondary-subtle

.bg-success

.bg-success-subtle

.bg-danger

EXAMPLE (2)

AT LEAST ONE EXCEPTION: CLARITY

```
# Custom attribute
```

```
cds-text="display"  
cds-text="headline"  
cds-text="message"
```

```
# Inline CSS variables
```

```
style="--size: var(--cds-global-space-7);"  
style="--size: var(--cds-global-space-14);"
```

Style Name (in Figma)	Attribute
Display	[cds-text="display"]
Headline	[cds-text="headline"]
Title	[cds-text="title"]

AND ALSO... THEMES!

SUSPICIOUSLY SIMILAR

```
# Bootstrap  
data-bs-theme="dark"
```

Scopable

```
# Daisy UI  
data-theme="retro"
```

```
# DSFR  
<html data-fr-mourning data-fr-scheme="light">
```

Not scopable

```
# Clarity  
<body cds-theme="dark">
```

EXCLUSION PRINCIPLE

Multiple styles on
one element



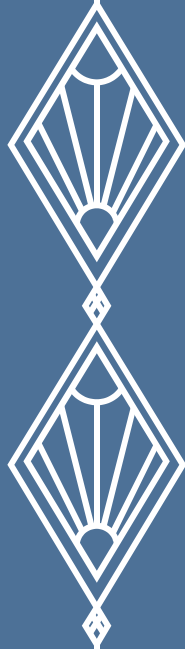
```
class="bg-primary border-2"  
class="bg-secondary border-1"
```

Multiple options
of the same style



```
class="border-primary border-secondary"
```

02



**WE ALREADY HAVE
SOMETHING**



IN CONTRIB



CONTRIB (1): STYLE OPTIONS

2 600 KNOWN INSTALLS

Linked to PHP plugins

Linked to Drupal usage

Multiple options of the same style possible

Not design system driven and front dev friendly

```
# *.style_options.yml

options:
  example_class:
    plugin: css_class
    label: 'My custom class'
    multiple: true
    options:
      - label: None
      - label: Style 1
        class: style_1
contexts:
  layout: { ... }
  paragraphs: { ... }
```

CONTRIB (2): BLOCK STYLE PLUGINS

600 KNOWN INSTALLS

```
# *.blockstyle.yml

checkbox_with_exclude:
  label: 'Checkbox with Exclude'
  class: '\Drupal\block_style_plugins_test\Plugin\BlockStyle\CheckboxWithExclude'
  exclude:
    - 'system_powered_by_block'
    - 'basic'
form_fields_created_with_yaml:
  label: 'Styles Created by Yaml'
  include:
    - 'system_powered_by_block'
  form:
    'test_field':
      '#type': 'textfield'
      '#title': 'Title Created by Yaml'
      '#default_value': 'text goes here'
```

Not design system driven
and front dev friendly

CONTRIB (3): LAYOUT BUILDER STYLES

20 000 KNOWN INSTALLS

Styles declared with config entities:

- LB Style
- LB Style group

Usage limited to Layout Builder.

Not front dev friendly

Label *

Add border around image(s)

CSS classes *
Enter one per line.

my-image-border-class

Type *

Block

Section

Determines if this style can be applied to sections or blocks.

Group *

Block Margins and Padding

Borders

Limit Items Per Row

CONTRIB (4): UI STYLES

1 300 KNOWN INSTALLS

Main module provided the API only.

Sub-modules for Drupal usage:

- Blocks
- Layout Builder
- Views
- ...

Front dev friendly!

```
# *.ui_styles.yml

background_color:
  label: Background color
  options:
    bg-primary: Primary
    bg-secondary: Secondary
    bg-success: Success
    bg-danger: Danger
    bg-warning: Warning
    bg-info: Info
```

CONTRIB SPACE: NOT GOOD ENOUGH

Not a proper ecosystem

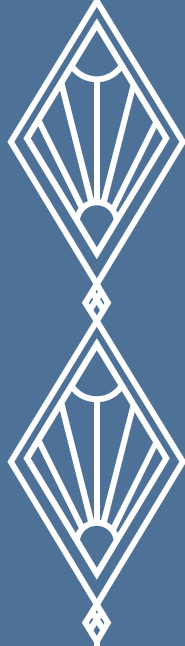
Not sharing the same philosophy

Too tied to visual builders

Not covering all visual builders

We need Core to fix this!

02



CORE API



LET CORE KNOWS ABOUT THE
STYLES



CORE ISSUE: #3517033

Add a style utility API

[View](#) [Edit](#) [View history](#)

Problem/Motivation

After SDC in Drupal 10.1 (July 2023), and the Icon API in Drupal 11.1 (Dec 2025), let's continue to implement design systems API in Core, in order to be able to build business agnostic, shareable, Drupal themes, providing design implementations which can be leveraged by display building tools.

Styles utilities are a common artefact of design systems. Examples:

- Bootstrap's utilities: <https://getbootstrap.com/docs/5.3/utilities/api/>
- Tailwind in DaisyUI: <https://daisyui.com/docs/utilities/>
- Material's styles: <https://m3.material.io/styles>
- Bulma's helpers: <https://bulma.io/documentation/helpers>
- PatternFly's utility classes : <https://www.patternfly.org/utility-classes/about-utility-classes>
- USWDS's utilities: <https://designsystem.digital.gov/utilities/>
- ...

Each style is a **set of mutually exclusive, self-descriptive, single-purpose, universal, CSS classes**. Examples: Typography, orders, Colors, Spacing, Elevation....

Analysis of the current solutions in the contrib space

Needs review

Project:	Drupal core
Version:	main
Component:	theme system
Priority:	Normal
Category:	Feature request
Assigned:	Unassigned
Issue tags:	TX (Themer Experience), 11.4.0 release priority
Reporter:	pdureau
Created:	3 Apr 2025 at 02:56 CEST
Updated:	23 Feb 2026 at 11:18 CET

★ Following 41 followers

 48 files **+5912** **-22**

66% LoC are tests (2026/03)

PLUGIN DEFINITION

FOR FRONT DEVS

Metadata

Styles

Application

Asset

```
my_style:  
  category: ...  
  label: ...  
  description: ...  
  links: { ... }  
  options:  
    option_1:  
      label: ...  
      description: ...  
      value: ...  
      target: ...  
      attribute: ...  
      status: ...  
  weight: ...  
  type: ...  
  target: ...  
  attribute: ...  
  status: ...  
  library: ...
```

Override per option

FEATURE: TARGET

Possible targets:

- any (scopable)
- html
- body
- meta

```
my_style:  
  label: ...  
  options:  
    option_1:  
      label: ...  
      target: ...  
      value: ...  
  target: ...
```

FEATURE: LIFECYCLE

Status on the style

Status per option

- experimental
- stable
- deprecated (logged)
- obsolete (logged)

```
my_style:  
  label: ...  
  options:  
    option_1:  
      label: ...  
      value: ...  
      status: ...  
  status: ...
```

FEATURE: LIBRARY MANAGEMENT

Automatic attachment of a library when using a style.

```
core/lib/Drupal/Core/Render/Element/Textarea.php
@@ -44,21 +44,12 @@ public function getInfo() {
44 44 ],
45 45 '#pre_render' => [
46 46 [static::class, 'preRenderGroup'],
47 - [static::class, 'preRenderAttachments'],
48 47 ],
49 48 '#theme' => 'textarea',
50 49 '#theme_wrappers' => ['form_element'],
51 50 ];
52 51 }
53 52
54 - /**
55 - * Adds the textarea resize library.
56 - */
57 public static function preRenderAttachments($element): array {
58 - $element['#attached']['library'][] = 'core/drupal.textarea-resize';
59 - return $element;
60 }
```

```
my_style:
  label: ...
  options:
    option_1:
      label: ...
      value: ...
  library: ...
```

FEATURE: JSON SCHEMA

Like with SDC and icon API, we introduce a JSON schema to validate automatically the styles declaration.

```
type: "object"
required:
  - "label"
  - "options"
properties:
  enabled:
    type: "boolean"
  label: { ... }
  description: { ... }
...
```

FEATURE: FORM ELEMENT

```
$form['styles'] = [  
  '#type' => 'styles',  
  '#title' => $this->t('Styles'),  
  '#default_value' => [  
    'style_id' => 'option_id'  
  ],  
  '#wrapper_type' => 'details',  
  '#open' => FALSE,  
];
```

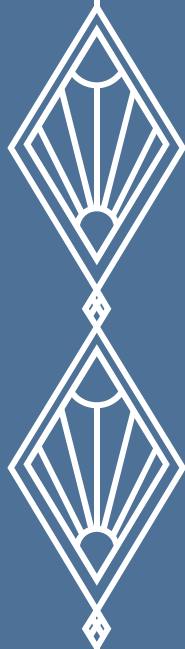
Simple form element as a first step.

Utilities

Modes



04



HOW TO USE IT



FRONT & BACK UNITE 🦊



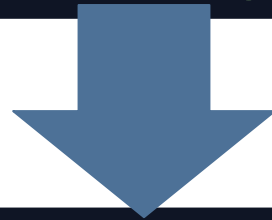
FOR FRONTEND DEVELOPERS

IN TWIG

New method `addStyle` on the Attribute object.

Attribute object will handles attachments!

```
{# before #}  
attributes.addClass('my-font-heavy-class')  
attach_library('provider/my_library')
```



```
{# after #}  
attributes.addStyle('font_weight', 'heavy')
```

FOR BACKEND DEVELOPERS

RENDER ARRAYS

```
# Before
$build = [
  '#type' => 'container',
  '#attributes' => [
    'class' => [
      'class-1',
      'class-2',
    ],
  ],
  '#attached' => [
    'library' => [
      'provider/my_library',
    ],
  ],
];
```



```
# After
$build = [
  '#type' => 'container',
  '#styles' => [
    'style' => 'option_1',
    'style_2' => 'option_2',
  ],
];
```

FOR BACKEND DEVELOPERS

BUBBLING

```
$build = [  
  '#type' => 'container',  
  '#attached' => [  
    'styles' => [  
      'style' => 'option_1',  
      'style_2' => 'option_2',  
    ],  
  ],  
];
```

To put styles in head or body and for non scopable mode.

MANY BENEFITS

👍 Check that style exists and is not deprecated.

👍 Check that option exists and is not deprecated.

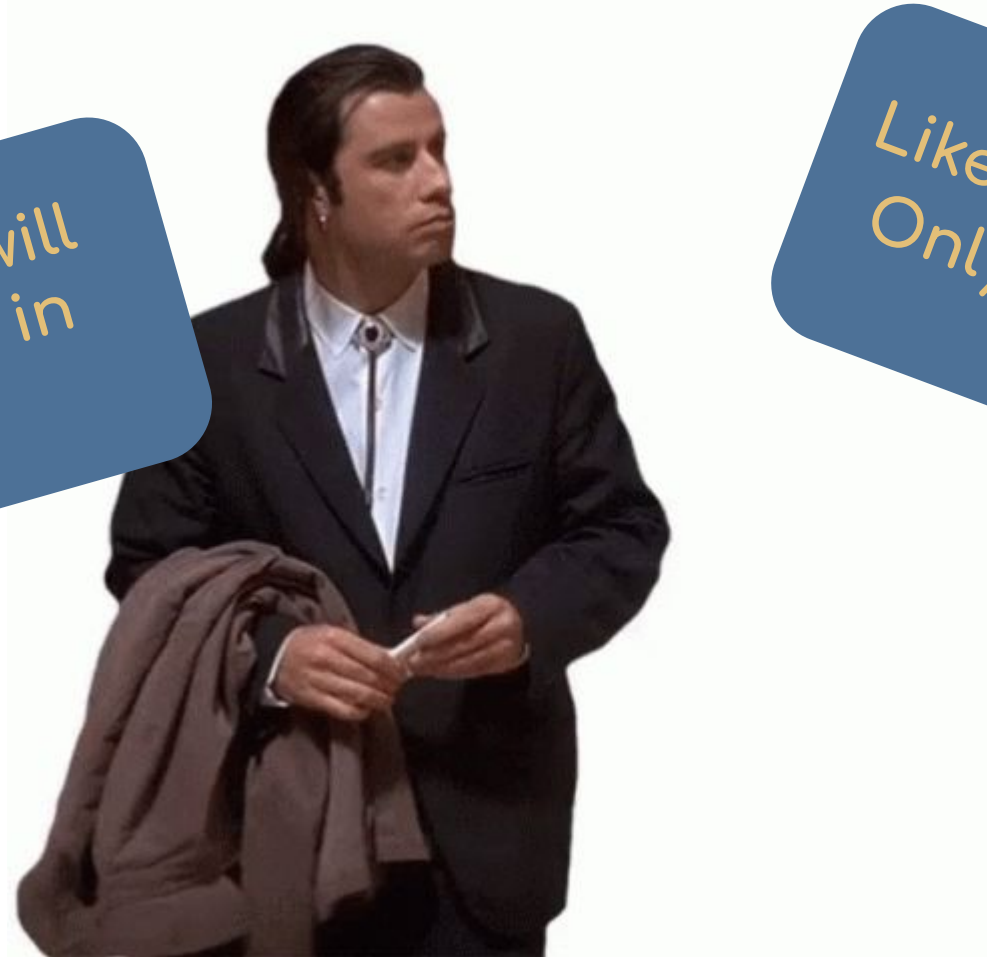
👍 Avoid incompatible options applied on the same element.

👍 No more need to know how the style is applied. If it is a class or something else.

👍 Asset library automatically handled.

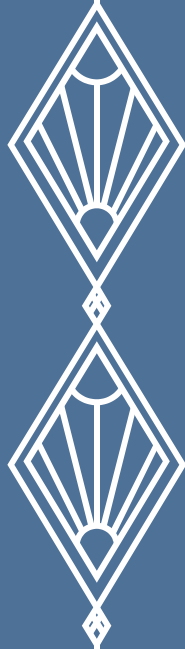
WHAT ABOUT SITE BUILDERS?

Site builders will
found usage in
contrib.



Like for SDC and Icon API.
Only API in Core for now.

05



A NEW CONTRIB SPACE



CONTRIB: UI STYLES 2

Add styles on html or body tags.

Modes available everywhere where UI Styles could already add styles!

HTML styles

Body styles

Styles to apply on the body tag.

Modes

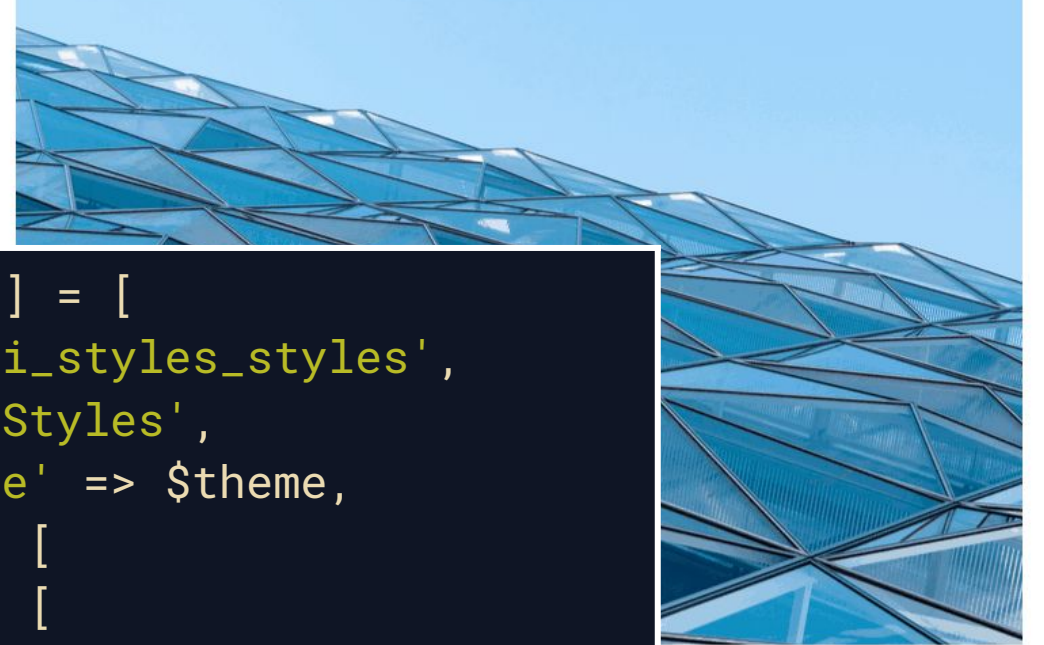
Color mode

- None -

Aa Light

Aa Dark

CONTRIB: UI STYLES 2



Improved form
element

Filtering

Widgets

```
$form['setting'] = [  
  '#type' => 'ui_styles_styles',  
  '#title' => 'Styles',  
  '#drupal_theme' => $theme,  
  '#filters' => [  
    'target' => [  
      Target::Any,  
      Target::Body,  
    ],  
    'type' => [Type::Mode],  
    'attribute' => [  
      'my_attribute',  
    ],  
    'status' => [  
      Status::Obsolete,  
    ],  
  ],  
];
```

CONTRIB: DISPLAY BUILDER

The screenshot displays the Display Builder interface. At the top, there are tabs for 'Libraries' and 'Logs', and a 'Builder' / 'Preview' toggle. Below this, there are several panels: 'Library panels' (with sub-tabs 'Components' and 'Blocks'), 'View panels', 'Token' (with options 'Base / Token / 1', 'Remove Token', 'Copy Token'), 'Toolbar buttons' (with 'Config' and 'Styles' buttons), and 'Contextual panels'. A green box highlights the 'Styles' button in the 'Toolbar buttons' panel. Below the main interface, there is a configuration table for 'Contextual panels'.

Enabled	Island	Configuration
<input checked="" type="checkbox"/>	Contextual form Configure the active component or block.	
<input checked="" type="checkbox"/>	Style utilities Apply style utilities to the active component or block.	

Let's make the Style utilities island configurable!

WHAT ELSE?

LET THE CONTRIB SPACE BLOOM



About the Core's Style API proposal

Active	
Project:	Style Options
Version:	2.0.x-dev
Component:	Miscellaneous
Priority:	Normal
Category:	Task

About the Core's Style API proposal

Active	
Project:	Block Style Plugin
Version:	8.x-2.x-dev
Component:	Miscellaneous
Priority:	Normal
Category:	Task

Try Core's Style API proposal

Needs review	
Project:	Layout Builder Styles
Version:	2.1.0
Component:	Code
Priority:	Normal
Category:	Task

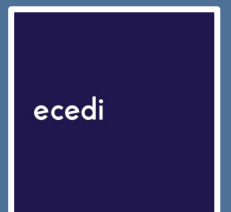
THANKS!



PLEASE EVALUATE
THE SESSION!



THANKS TO ALL
OUR SPONSORS



WANT MORE UI SUITE?



Display Builder, the
HTMX-powered visual builder by
Pierre Dureau

Tomorrow! 4pm!

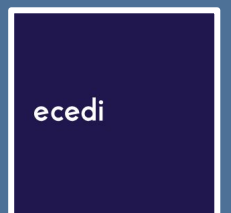
Amphi "Bleu de Sassenage"



PLEASE EVALUATE
THE SESSION!



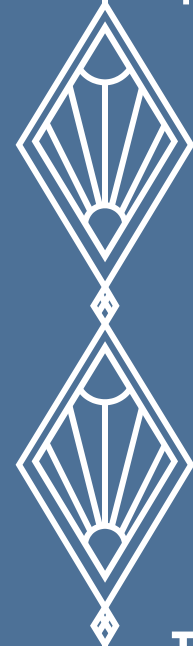
THANKS TO ALL
OUR SPONSORS



ANY QUESTIONS?



PLEASE EVALUATE
THE SESSION!



THANKS TO ALL
OUR SPONSORS

