

PLAY



The Future of Drupal Theming

AI, Experience Builder, and Beyond

MIKE HERCHEL // FOUNDER @ DRIPYARD



The Future of Drupal Theming

AI, Experience Builder, and Beyond

MIKE HERCHEL // FOUNDER @ DRIPYARD

Mike Herchel

- 18+ years in Drupal development
- Primary dev / maintainer of Drupal's default front-end theme, Olivero
- CSS Subsystem maintainer for Drupal core
- Helped get various modern frontend features into Drupal core
- Dad to a kickass teenager, and two bad dogs
- Fan of the open web, fast and accessible websites and long walks on the beach
- Founder of DRIPYARD



An aerial, high-angle view of a city at night. The buildings are illuminated with various colors of light, including blue, purple, and yellow. The lights create a vibrant, futuristic atmosphere. The perspective is from a high vantage point, looking down on the city's grid.

Dripyard®

Agenda

- **Past:** Templates everywhere
 - Problems, challenges, and solutions
- **Present:** Components everywhere
 - Tips, tricks, and best practices
- **Future:** XB, AI, and the web platform
 - Future proofing your code and gotchas



The dark past of Drupal's frontend

Theming Drupal 5-7

- PHP Template
- Monolith CSS files generated from build process
 - Most styles added through *.info file
 - Can be more granular with preprocess
- jQuery
- Robust templating system

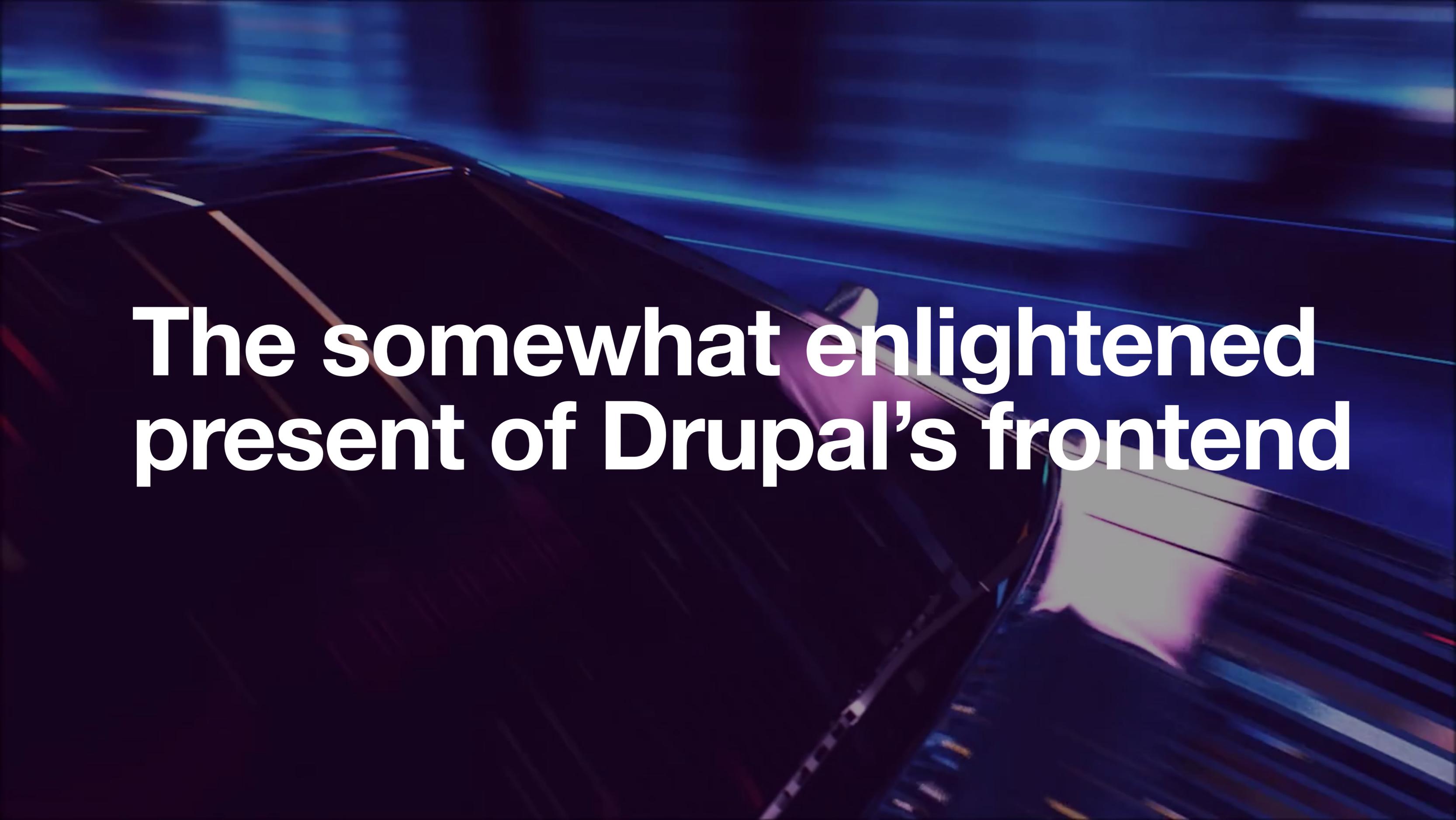
Theming Drupal 8

- Twig
- Introduction of Libraries API
- Monolith CSS files generated from build process
- Beginning of phasing out of jQuery

Theming Drupal 9



- Olivero and Claro themes
- Easier template debugging with `dump()` (but not until v9.5)

The background is an abstract digital composition. It features a dark blue and purple color palette. There are several bright, glowing lines and streaks of light, primarily in shades of cyan and magenta, that create a sense of motion and depth. A faint, grid-like pattern is visible, suggesting a digital or architectural space. The overall effect is futuristic and high-tech.

**The somewhat enlightened
present of Drupal's frontend**

Theming Drupal 10-11

- Easy peasy template discovery and debugging with new UI settings
- New Icon API in D11.1
- New Twig filters (`add_suggestion`, `add_class`, `set_attribute`)
- Single Directory Components!

Development settings

These settings should only be enabled on development environments and never on production.

Do not cache markup

Disables render cache, dynamic page cache, and page cache.

Twig development mode

Exposes Twig development settings.

Twig development mode

Twig debug mode

Provides Twig's `dump()` function for debugging, outputs template suggestions to HTML comments, and automatically recompile Twig templates after changes.

Disable Twig cache

Twig templates are not cached and are always compiled when rendered.

Save settings

Icon API

```
$build['icon'] = [  
  '#type' => 'icon',  
  '#pack_id' => 'material_symbols',  
  '#icon_id' => 'home',  
  '#settings' => [  
    'size' => 64,  
  ],  
];
```

```
{{ icon('material_symbols', 'home', { size: 64 }) }}
```

EXPLORER: XB-DE...

- web
 - themes
 - contrib
 - theme-uno
 - components
 - config
 - css
 - icons
 - images
 - js
 - src
 - templates
 - .gitignore
 - composer.json
 - README.md
 - ! uno.icons.yml**
 - ! uno.info.yml
 - ! uno.libraries.yml
 - uno.theme
 - README.txt
 - .csslintrc
 - .eslintignore
 - .eslintrc.json
 - .ht.router.php
 - .htaccess
 - autoload.php
 - example.gitignore
 - index.php
 - INSTALL.txt
 - README.md
 - robots.txt
 - update.php
 - .editorconfig
 - .gitattributes
 - composer.json

```

! uno.icons.yml
web > themes > theme-uno > ! uno.icons.yml > {} uno_font_awesome > {} license
You, last month | 1 author (You)
1 uno_font_awesome:
2   enabled: true
3   label: "Font Awesome SVG"
4   links:
5     - https://fontawesome.com/icons
6     - https://docs.fontawesome.com/web/add-icons/svg-bare
7   version: 6.7.2
8   license: You, last month • Icons.
9     name: Font Awesome Free License
10    url: https://github.com/FortAwesome/Font-Awesome/blob/6.x/LICENSE.txt
11    gpl-compatible: true
12  extractor: svg
13  config:
14    sources:
15      - icons/fontawesome/svg/{group}/{icon_id}.svg
16  settings:
17    size:
18      title: "Size"
19      type: "integer"
20      default: 32
21    color:
22      title: "Color"
23      type: "string"
24      format: "color"
25    alt:
26      title: "Alt text"
27      type: "string"
28      description: "Accessibility alternative text, leave empty for decorative icon."
29  template: >-
30    <svg
31      {{ attributes
32        .addClass('icon')
33        .setAttribute('viewBox', attributes.viewBox|default('0 0 50 50'))
34        .setAttribute('height', size|default('32'))
35        .setAttribute('fill', color|default('currentColor'))
36        .setAttribute('aria-hidden', 'true')
37      }}
38    >
39    {{ content }}
40  </svg>

```

Icon API

Drupal 10.3-11.0

https://www.drupal.org/project/ui_icons 1.0.x branch
has a backport submodule

Drupal 11.1+

<https://www.drupal.org/docs/develop/drupal-apis/icon-api>

Twig Filters

|add_suggestion()

```
{# node--article.html.twig #}  
{{ content.field_items|add_suggestion('unordered-list') }}
```

```
{# field--unordered-list.html.twig #}  
<ul{{ attributes }}  
  {% for item in items %}  
    <li{{ item.attributes }}>{{ item.content }}</li>  
  {% endfor %}  
</ul>
```

Twig Filters

|add_class() and |set_attribute()

```
{# node--article.html.twig #}  
{{ content.field_tags|add_class('my-class') }}
```

```
{# image-formatter--fetchpriorityhigh.html.twig #}  
{{ image|set_attribute('fetchpriority', 'high') }}
```

Twig Filters

Chain for extra fun!

```
{# node--article.html.twig #}  
{{ content.field_items|add_suggestion('unordered-list')|add_class('my-items__item') }}
```

TEMPLATES

ALL THE WAY DOWN



Is SDC our Savior?



EXPLORER: DRUPA...

- web
 - profiles
 - sites
- themes
 - contrib
 - theme-uno
 - components
 - accordion
 - breadcrumb
 - button
 - card
 - images
 - ! card.component.yml** M
 - # card.css
 - ✓ card.example.twig
 - JS card.js
 - ✓ card.twig
 - ctas
 - details
 - fieldset
 - footer
 - header
 - heading
 - hero
 - hero-2
 - icon

! card.component.yml M X

web > themes > theme-uno > components > card > ! card.component.yml > \$schema

```
1 $schema: https://git.drupalcode.org/project/drupal/-/raw/HEAD/core/assets/schemas/v1/metadata.schema.json
2 name: Card
3 status: stable
4 group: Components
5 props:
6   type: object
7   required:
8     - title
9   properties:
10    attributes:
11      type: Drupal\Core\Template\Attribute
12    image:
13      $ref: json-schema-definitions://experience_builder.module/image
14      title: 'Image'
15      type: object
16    examples:
17      - src: images/palm3.jpg
18        alt: 'Boring placeholder'
19        width: 2670
20        height: 1780
21    title:
22      title: Title
23      type: string
24      examples:
25        - Card title
26    body:
27      $ref: json-schema-definitions://experience_builder.module/textarea
28      title: Body text
29      type: string
```

EXPLORER: DRUPA...

- web
 - profiles
 - sites
- themes
 - contrib
 - theme-uno
 - components
 - accordion
 - breadcrumb
 - button
 - card
 - images
 - ! card.component.yml M
 - # card.css
 - ✓ card.example.twig
 - JS card.js
 - ✓ card.twig M
 - ctas
 - details
 - fieldset
 - footer
 - header
 - heading
 - hero
 - hero-2
 - icon

! card.component.yml M ✓ card.twig M X

web > themes > theme-uno > components > card > ✓ card.twig

```

1  <article{{ attributes.addClass('card', 'theme--light') }}>
2  <div class="card__top">
3  |   {{ include('uno:image-or-media', { image }, with_context = false) }}
4  </div>
5  <div class="card__bottom">
6  |   {% if link_href %}
7  |     <a href="{{ link_href }}" {{ open_new_window ? 'target="_blank"' }} class="card__link">
8  |     |   {% endif %}
9  |     <h3 class="card__title">{{ title }}</h3>
10 |     </a>{% endif %}
11 |
12 |   {% if body %}
13 |     <div class="card__body">
14 |     |   {{ body }}
15 |     </div>
16 |   {% endif %}
17 |
18 |   {% if cta_text %}
19 |     <div class="card__cta-text">
20 |     |   {{ cta_text }}
21 |     </div>
22 |   {% endif %}
23 |   {% if link_href %}
24 |     </a>
25 |   {% endif %}
26 </div>
27 </article>
28

```

EXPLORER: DRUPA...

- web
 - profiles
 - sites
- themes
 - contrib
 - theme-uno
 - components
 - accordion
 - breadcrumb
 - button
 - card
 - images
 - ! card.component.yml M
 - # card.css
 - card.example.twig
 - JS card.js
 - card.twig
 - ctas
 - details
 - fieldset
 - footer
 - header
 - heading
 - hero
 - hero-2
 - icon

! card.component.yml M # card.css

web > themes > theme-uno > components > card > # card.css > .card

```
1 .card {  
2   --card-padding-block: 16px;  
3   --card-padding-inline: 24px;  
4   --card-border-radius: 10px;  
5   --card-box-shadow: 0 24px 36px -18px #00000066;  
6   --card-translate-distance: 36px;  
7   --card-background: var(--theme-surface);  
8   --card-text-color: var(--theme-text-color-medium);  
9  
10  position: relative; /* Anchor hyperlink pseudo-element. */  
11  display: flex;  
12  flex-direction: column;  
13  width: 100%;  
14  border: 1px solid transparent;  
15  border-radius: var(--card-border-radius);  
16  box-shadow: var(--card-box-shadow);  
17  overflow: clip;  
18  color: var(--card-text-color);  
19 }  
20  
21 .card--has-link {  
22   cursor: pointer;  
23 }  
24  
25 .card__top {  
26   position: relative;  
27   aspect-ratio: 16 / 9;  
28   /* translate: 0 var(--card-translate-distance); */  
29   transition: translate 0.2s;
```

EXPLORER: DRUPA...

- web
 - profiles
 - sites
- themes
 - contrib
 - theme-uno
 - components
 - accordion
 - breadcrumb
 - button
 - card
 - images
 - ! card.component.yml M
 - # card.css
 - ✓ card.example.twig
 - JS card.js
 - ✓ card.twig
 - ctas
 - details
 - fieldset
 - footer
 - header
 - heading
 - hero
 - hero-2
 - icon

! card.component.yml M JS card.js

```
web > themes > theme-uno > components > card > JS card.js > ...  
You, 3 months ago | 1 author (You)  
1 /** You, 3 months ago • WIP on cards and layout 3 col.  
2 * @file  
3 * Enables navigation when user clicks the entire card.  
4 */  
5  
6 ((Drupal, once) => {  
7   /**  
8    * Initialize the cards.  
9    *  
10   * @param {HTMLElement} card  
11   *   The card DOM element.  
12   */  
13   function init(card) {  
14     const link = card.querySelector('[href]');  
15  
16     if (link) {  
17       card.classList.add('card--has-link');  
18       card.onclick = () => link.click();  
19     }  
20   }  
21  
22  
23   /**  
24   * Initialize the cards.  
25   *  
26   * @type {Drupal~behavior}  
27   *  
28   * @prop {Drupal~behaviorAttach} attach  
29   */
```

EXPLORER: DRUPA...

- web
 - profiles
 - sites
- themes
 - contrib
 - theme-uno
 - components
 - accordion
 - breadcrumb
 - button
 - card
 - images
 - ! card.component.yml M
 - # card.css
 - card.example.twig
 - JS card.js
 - card.twig M
 - ctas
 - details
 - fieldset
 - footer
 - header
 - heading
 - hero
 - hero-2
 - icon

! card.component.yml M | card.twig M | card.example.twig X

```
web > themes > theme-uno > components > card > card.example.twig  
You, 2 weeks ago | 1 author (You)  
1 {{ include('uno:card', { You, last month • More cleanup.  
2 title: 'Accessible by default',  
3 image: {  
4   src: '/' ~ directory ~ '/components/card/images/palm2.jpg',  
5   alt: 'Palm tree at sunset',  
6   width: 2670,  
7   height: 1780,  
8 },  
9 body: 'DripYard themes are built with accessibility in mind, meeting a minimum conformance of WCAG 2.2 AA.  
10 link_href: 'https://www.dripyard.com',  
11 open_new_window: false,  
12 cta_text: 'Learn more',  
13 }, with_context = false) }}  
14
```

Is this theming
nirvana?





The future

Finding the nirvana of Drupal
theming



**Writing future-proof
code
(As much as possible)**



Disclaimer:

**Best practices are still
evolving**

Challenges

- What should be a component?
- How can we use the same component with and without XB?
- How do I tell XB to use a specific widget (e.g. textarea)?
- How will images be handled (image styles etc)?
- How will layout be handled?



**Not everything should
be a component!**

But most things should be

Examples of what are NOT:

- Base global styles
 - Buttons
 - Text inputs and form elements
- Variables



**Making your
SDCs XBB ready**

Making your SDCs XB ready

Required props

```
props:  
  type: object  
  required:  
    - title  
  properties:  
    title:  
      title: Title  
      type: string  
      examples: [ This is the title. ]
```

Components status

[Enabled components](#)[Disabled components](#)

Component	Status	Reason
block.dashboard_text_block	Incompatible	<ul style="list-style-type: none">Block plugin settings must be fully validatable
sdccivictHEME.button	Incompatible	<ul style="list-style-type: none">Component has "obsolete" status
sdccivictHEME.experience_builder.shoe_button	Incompatible	<ul style="list-style-type: none">Experience Builder does not know of a field type/widget to allow populating the icon prop, with the shape {"type":"object","\$ref":"json-schema-definitions://experience-builder/icon"}
sdccivictHEME.experience_builder.shoe_details	Incompatible	<ul style="list-style-type: none">Experience Builder does not know of a field type/widget to allow populating the expand_icon prop, with the shape {"type":"object","\$ref":"json-schema-definitions://experience-builder/icon"}Experience Builder does not know of a field type/widget to allow populating the collapse_icon prop, with the shape {"type":"object","\$ref":"json-schema-definitions://experience-builder/icon"}
sdccivictHEME.experience_builder.video	Incompatible	<ul style="list-style-type: none">Experience Builder does not know of a field type/widget to allow populating the src prop, with the shape {"type":"string","format":"uri","pattern":"\\. (mp4 webm) (\\. /)+\$"}.
sdccivictHEME.navigation.badge	Incompatible	<ul style="list-style-type: none">Prop "status" must have title
sdccivictHEME.navigation.title	Incompatible	<ul style="list-style-type: none">Experience Builder does not know of a field type/widget to allow populating the modifiers prop, with the shape {"type":"array","items":{"type":"string","enum":["ellipsis","dots"]}}Experience Builder does not know of a field type/widget to allow populating the extra_classes prop, with the shape {"type":"array","items":{"type":"string"}}.
sdccivictHEME.navigation.toolbar-button	Incompatible	<ul style="list-style-type: none">Experience Builder does not know of a field type/widget to allow populating the modifiers prop, with the shape {"type":"array","items":{"type":"string","enum":["collapse","expand"]}}Experience Builder does not know of a field type/widget to allow populating the extra_classes prop, with the shape {"type":"array","items":{"type":"string"}}.
sdccivictHEME.starshot_demo.button	Incompatible	<ul style="list-style-type: none">Component has "obsolete" status
sdccivictHEME.starshot_demo.old-starshot-button	Incompatible	<ul style="list-style-type: none">Component has "obsolete" status

Making your SDCs XB ready

Selecting widgets

```
image:  
  $ref: json-schema-definitions://experience_builder.module/image  
  title: 'Image'  
  type: object
```

```
content:  
  $ref: json-schema-definitions://experience_builder.module/textarea  
  title: Content  
  type: string
```

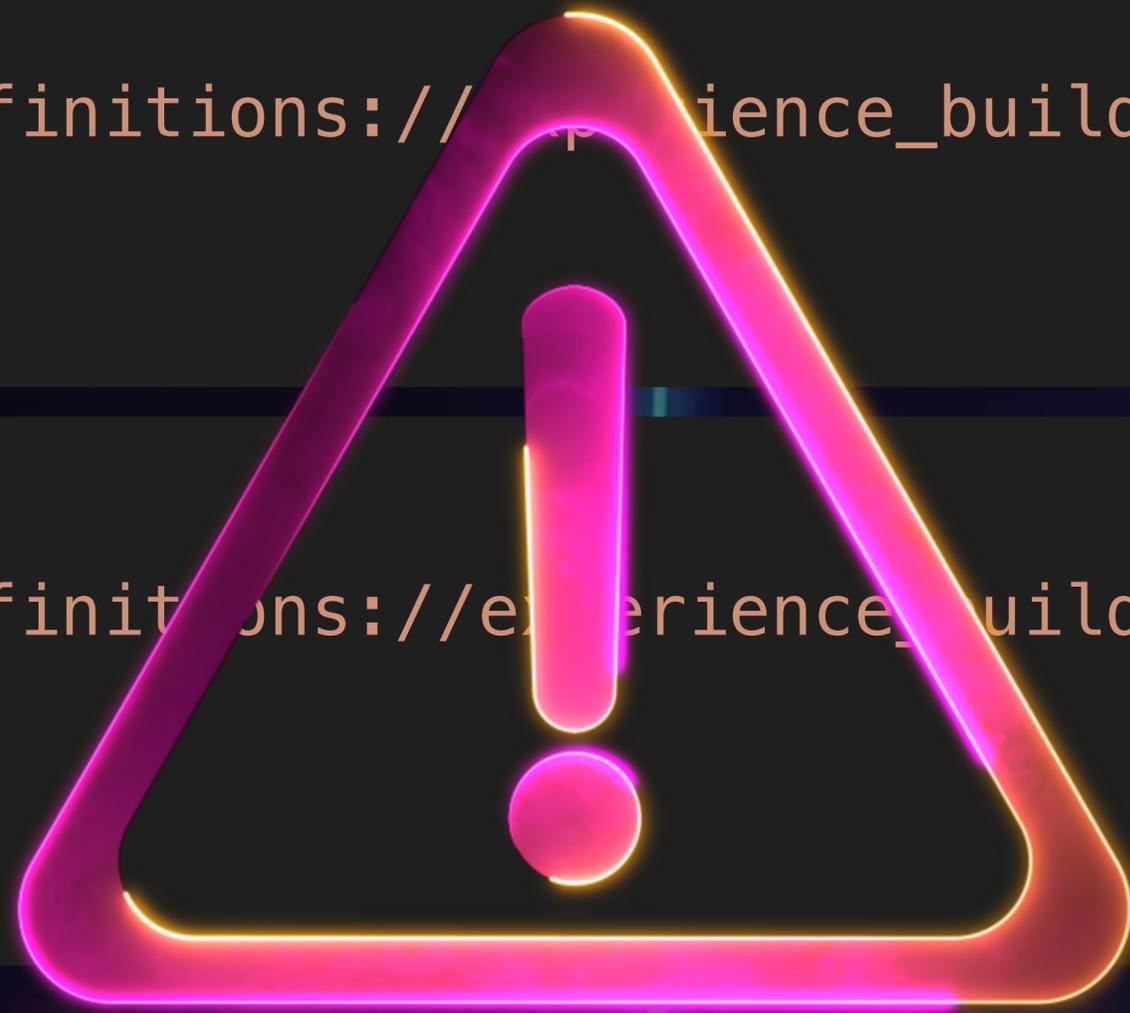
https://git.drupalcode.org/project/experience_builder/-/blob/0.x/schema.json

Making your SDCs XB ready

Selecting widgets

```
image:  
  $ref: json-schema-definitions://experience_builder.module/image  
  title: 'Image'  
  type: object
```

```
content:  
  $ref: json-schema-definitions://experience_builder.module/textarea  
  title: Content  
  type: string
```



Making your SDCs XB ready

Working with images

```
image: {  
  width: 700,  
  height: 700,  
  alt: 'boat',  
  src: '/files/boat.jpg'  
}
```

```
<div class="field field--name-field-media-image field--type-entity-reference field--label-above">  
  <div class="field__label">media image</div>  
  <div class="field__item">  
    <div class="contextual-region media media--type-image media--view-mode-default">  
      <div data-contextual-id="media:media=2:changed=1741531435&langcode=en" data-contextual-token="YCZIS6Gk8avfGV3QYf-ku1_15_BHwCzQP994tMNpWc0" data-drupal-ajax-container="" data-once="contextual-render" class="contextual"><button class="trigger focusable visually-hidden" type="button" aria-pressed="false">Open configuration options</button>  
      <ul class="contextual-links" hidden="">  
        <li><a href="/media/2/edit?destination=/node/8">Edit</a></li>  
        <li><a href="/media/2/delete?destination=/node/8">Delete</a></li>  
      </ul>  
    </div>  
    <div class="field field--name-field-media-image field--type-image field--label-visually_hidden">  
      <div class="field__label visually-hidden">Image</div>  
      <div class="field__item">  
          
      </div>  
    </div>  
  </div>  
</div>  
</div>
```

Making your SDCs XB ready

Working with images

```
{# components/image-or-media/image-or-media.twig #}
{% if image.src %}
  {% set image_attributes = create_attribute() %}
  <img
    {% for attribute, value in image %}
      {{ image_attributes
        .setAttribute(attribute, value)
        .addClass(classes ? classes)
        .setAttribute('loading', loading ?: 'lazy' )
      }}
    {% endfor %}
  />
{% else %}
  {{ image|add_suggestion('bare') }}
{% endif %}
```

Making your SDCs XB ready

Working with images

```
{# field--bare.html.twig #}
{% for item in items %}
  {% if item.content['#theme'] %}
    {{ item.content|add_suggestion('bare') }}
  {% else %}
    {{ item.content }}
  {% endif %}
{% endfor %}
```

```
{# media--bare.html.twig #}
{% if content %}
  {% if content['#theme'] %}
    {{ content|add_suggestion('bare') }}
  {% else %}
    {{ content }}
  {% endif %}
{% endif %}
```

Making your SDCs XB ready

Working with images

```
{# *Hopefully* coming soon to XB (but with different syntax) #}  

```

Tip: containers should dictate component spacing



```
.my-component {  
  margin-block: var(--spacing-xl);  
}
```

Tip: containers should dictate component spacing



```
.component-container {  
  > * {  
    margin-block: var(--spacing-xl);  
  }  
}
```

Gotcha: SDC CSS Source Order

```
<link rel="stylesheet" media="all" href="/core/modules/system/css/components/system-status-counter.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/system/css/components/system-status-report-counters.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/system/css/components/system-status-report-general-info.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/system/css/components/abledrag.module.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/system/css/components/tablesrt.module.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/system/css/components/tree-child.module.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/components/ajax-progress.module.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/components/autocomplete-loading.module.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/contextual/css/contextual.module.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/views/css/views.module.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/sdc/../../../../core/themes/olivero/components/card/card.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/claro/css/components/toolbar.module.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/contextual/css/contextual.toolbar.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/claro/css/state/toolbar.menu.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/contextual/css/contextual.theme.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/contextual/css/contextual.icons.theme.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/claro/css/theme/toolbar.theme.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/claro/css/theme/toolbar.icons.theme.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/user/css/user.icons.admin.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/shortcut/css/shortcut.theme.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/shortcut/css/shortcut.icons.theme.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/base/fonts.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/base/variables.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/base/base.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/layout/layout.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/layout/grid.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/layout/layout-content-narrow.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/layout/layout-content-medium.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/layout/layout-footer.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/layout/region.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/layout/region-content.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/layout/region-hero.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/layout/region-secondary-menu.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/layout/social-bar.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/layout/views.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/components/progress.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/components/block.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/components/breadcrumb.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/components/embedded-media.css?s36t5d" />
```



Gotcha: SDC CSS Source Order

- <https://www.drupal.org/project/drupal/issues/3374901>
- https://www.drupal.org/project/sdc_css_relocator

```
<link rel="stylesheet" media="all" href="/core/modules/system/css/components/system-status-counter.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/system/css/components/system-status-report-counters.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/system/css/components/system-status-report-general-info.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/system/css/components/abledrag.module.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/system/css/components/tablesrt.module.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/system/css/components/tree-child.module.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/components/ajax-progress.module.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/components/autocomplete-loading.module.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/contextual/css/contextual.module.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/views/css/views.module.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/sdc/../../../../core/themes/olivero/components/card/card.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/claro/css/components/toolbar.module.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/contextual/css/contextual.toolbar.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/claro/css/state/toolbar.menu.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/contextual/css/contextual.theme.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/contextual/css/contextual.icons.theme.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/claro/css/theme/toolbar.theme.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/claro/css/theme/toolbar.icons.theme.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/user/css/user.icons.admin.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/shortcut/css/shortcut.theme.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/modules/shortcut/css/shortcut.icons.theme.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/base/fonts.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/base/variables.css?s36t5d" />
<link rel="stylesheet" media="all" href="/core/themes/olivero/css/base/base.css?s36t5d" />
```



Gotcha: SDC CSS Source Order

```
.hero__button[class] {  
  /* Custom button styles */  
}
```

- <https://www.drupal.org/project/drupal/issues/3374901>
- https://www.drupal.org/project/sdc_css_relocator

Gotcha: SDC CSS Source Order

```
.hero__button {  
  /* Custom but  
}
```



- <https://www.drupal.org/project/drupal/issues/3374901>



JavaScript in Experience Builder?



Layers Library

+ Add new

Sections >

Components ▾

- ❖ Accordion group
- ❖ Accordion item
- ❖ Button
- ❖ Footer
- ❖ Heading
- ❖ Layout 3 Column
- ❖ Tab
- ❖ Tab group
- ❖ Text

Code ▾

</> test

Overrides >

Editor

JavaScript CSS Global CSS

```
1 export default function MyComponent() {
2   return (
3     <div class="p-4">
4       <h1>{ this.props.testTitle }</h1>
5       <p>This is a paragraph, coming from Experience Builder, imagine all the possibilities this will bring for content editors!</p>
6     </div>
7   );
8 }
```

Add to components

Preview

This is the title

This is a paragraph, coming from Experience Builder, imagine all the possibilities this will bring for content editors!

Component data

Props Slots

Prop name	Type	Required
testTitle	Text	<input type="checkbox"/>
Example value		
<input type="text" value="This is the title"/>		

+ Add

Artificial Intelligence: Where does it fit in?



New! Introducing Figma to Bolt

What do you want to build?

Prompt, run, edit, and deploy full-stack **web** and **mobile** apps.

```
Create an accordion group component using only vanilla JS, HTML, and css. It should have an 80s miami vice look and feel. →
```

Use **Shift** + **Return** for a new line

- Import from Figma
- Build a mobile app with Expo
- Start a blog with Astro
- Create a docs site with Vitepress
- Scaffold UI with shadcn
- Draft a presentation with Slidev

or start a blank app with your favorite stack



component. I'll update the necessary files with vibrant colors, neon effects, and a synthwave aesthetic.

Create 80s Miami Vice Accordion Component

- ✓ Create initial files
- ✓ Install dependencies

```
npm install
```

- ✓ Update `style.css`
- ✓ Update `main.js`

> Start application

```
npm run dev
```

I've created a Miami Vice-themed accordion component with:

1. Retro 80s styling featuring:

- Neon pink and cyan colors
- Gradient backgrounds
- Glowing effects
- Bold typography
- Smooth animations

2. Functionality:

- Click to expand/collapse sections
- Smooth animations for opening/closing
- Only one section can be open at a time
- Plus/minus icons that animate on click

3. Responsive design that works on all screen sizes

The development server is now running, and you should see the Miami Vice-styled accordion component with three example sections. Click on any header to expand its content!

Report Issue

Code Preview

5173 /

Miami Vice Accordion

CRUISING SOUTH BEACH

Feel the ocean breeze as you cruise down Ocean Drive in your white Ferrari Testarossa. The neon lights reflect off the chrome, and the palm trees sway to the rhythm of the night.

SUNSET VIBES

NIGHT MOVES

How can Bolt help you today?



component. I'll update the necessary files with vibrant colors, neon effects, and a synthwave aesthetic.

Create 80s Miami Vice Accordion Component

- ✓ Create initial files
- ✓ Install dependencies

```
npm install
```

- ✓ Update `style.css`
- ✓ Update `main.js`
- > Start application

```
npm run dev
```

I've created a Miami Vice-themed accordion component with:

1. Retro 80s styling featuring:

- Neon pink and cyan colors
- Gradient backgrounds
- Glowing effects
- Bold typography
- Smooth animations

2. Functionality:

- Click to expand/collapse sections
- Smooth animations for opening/closing
- Only one section can be open at a time
- Plus/minus icons that animate on click

3. Responsive design that works on all screen sizes

The development server is now running, and you should see the Miami Vice-styled accordion component with three example sections. Click on any header to expand its content!

[Report Issue](#)

How can Bolt help you today?



Code Preview

Files

public

.gitignore

counter.js

index.html

javascript.svg

main.js

package-lock.json

package.json

style.css

main.js

```
1 import './style.css'
2
3 document.querySelector('#app').innerHTML = `
4   <div>
5     <h1>Miami Vice Accordion</h1>
6     <div class="accordion">
7       <div class="accordion-item">
8         <div class="accordion-header">
9           Cruising South Beach
10        <div class="accordion-icon"></div>
11      </div>
12      <div class="accordion-content">
13        <p>Feel the ocean breeze as you cruise down Ocean Drive in your white Ferrari Testarossa.
14        The neon lights reflect off the chrome, and the palm trees sway to the rhythm of the night.</p>
15      </div>
16    </div>
17
18    <div class="accordion-item">
19      <div class="accordion-header">
20        Sunset Vibes
21      <div class="accordion-icon"></div>
22    </div>
23    <div class="accordion-content">
24      <p>Watch the sun dip below the horizon, painting the sky in brilliant shades of pink and purple.
25      The perfect backdrop for Miami's vibrant nightlife scene.</p>
26    </div>
27  </div>
28
29  <div class="accordion-item">
30    <div class="accordion-header">
31      Night Moves
32    <div class="accordion-icon"></div>
33  </div>
34  <div class="accordion-content">
35    <p>When the stars come out, the city comes alive. Synthesizer music fills the air,
36    and the streets pulse with the energy of a thousand neon dreams.</p>
37  </div>
38 </div>
39 </div>
40 </div>
41 `
42
43 // Accordion functionality
44 const accordionHeaders = document.querySelectorAll('.accordion-header');
45
46 accordionHeaders.forEach(header => {
```

Bolt

Terminal



VITE v5.4.14 ready in 776 ms

```
+ Local: http://localhost:5173/
+ Network: use --host to expose
+ press h + enter to show help
```



Layers Library

+ Add new

Sections >

Components ▾

- ❖ Accordion group
- ❖ Accordion item
- ❖ Button
- ❖ Footer
- ❖ Heading
- ❖ Layout 3 Column
- ❖ Tab
- ❖ Tab group
- ❖ Text

Code ▾

</> test

Overrides >

Editor

JavaScript CSS Global CSS

```
1 export default function MyComponent() {
2   return (
3     <div class="p-4">
4       <h1>{ this.props.testTitle }</h1>
5       <p>This is a paragraph, coming from Experience Builder, imagine all the possibilities this will bring for content editors!</p>
6     </div>
7   );
8 }
```

Add to components

Preview

This is the title

This is a paragraph, coming from Experience Builder, imagine all the possibilities this will bring for content editors!

Component data

Props Slots

Prop name	Type	Required
testTitle	Text	<input type="checkbox"/>
Example value		
<input type="text" value="This is the title"/>		

+ Add

Thank you! Questions?



MIKE HERCHEL // FOUNDER @ DRIPYARD