

Level Up Your Layout!

Component Based SiteBuilding
with Layout Builder
and View Modes





Aimee Hannaford

PMP, SCPM, CSM, CSPO, CPWA

CEO / Principal Architect

 aimeerae  aimeeraed

Ryan Bateman

Architect

 porkloin  _porkloin_

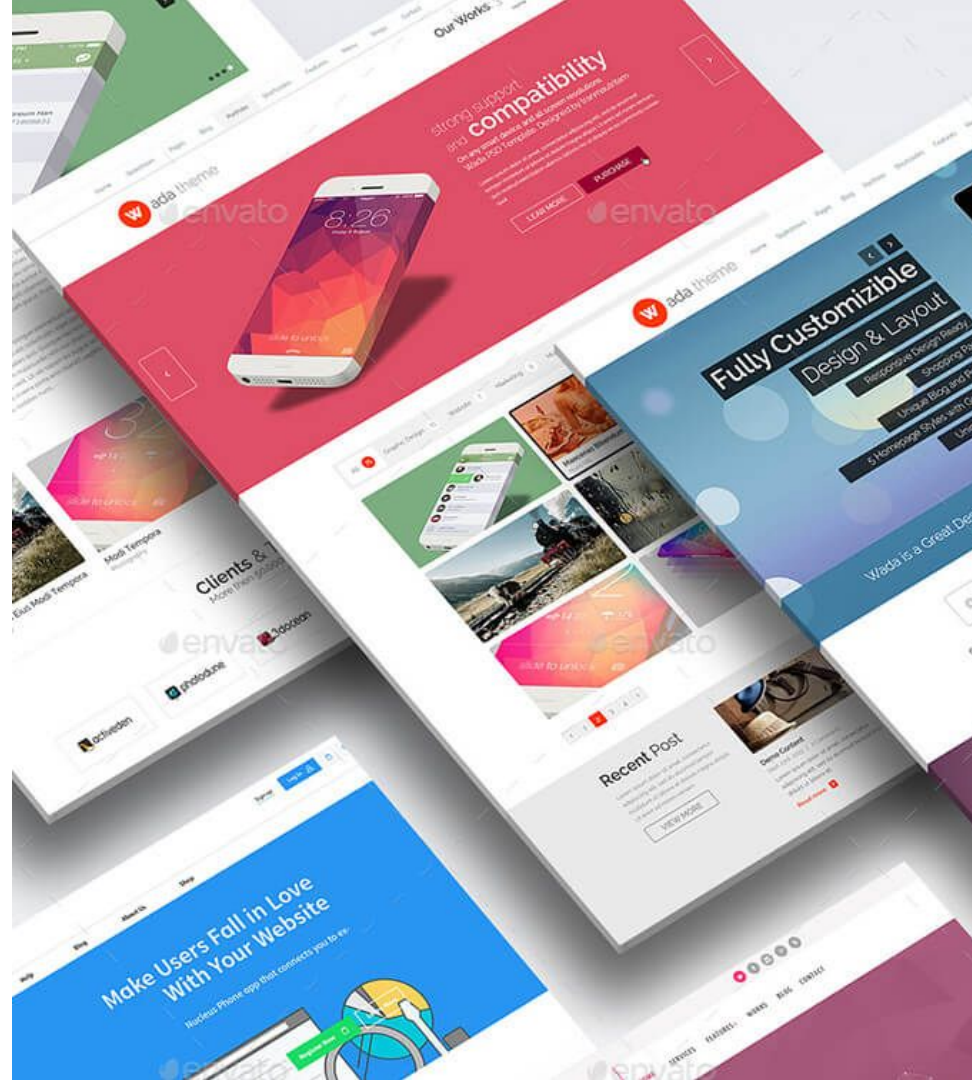


Topics Covered

- Design to Development
- View Modes
- Layout Builder
- View Modes + Layout Builder Basics
- Demo!
- Leveled Up Layouts
- Considerations

Design to Development

Get the Design



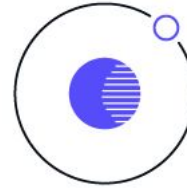
Review the Site's Design

Identify Similarities

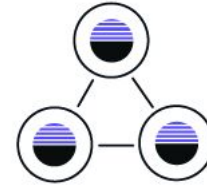
- Banners
- Lists
- Grids
- Full content layouts
- Embellishments / callouts

Component based design / Atomic Design

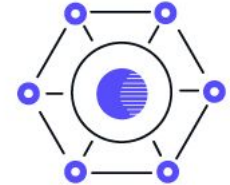
<http://atomicdesign.bradfrost.com/>



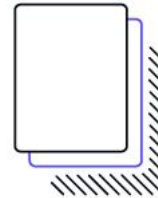
ATOMS



MOLECULES



ORGANISMS

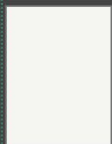


TEMPLATES



PAGES

Source: <https://dribbble.com/shots/6324875-Atomic-Design-Icons>



One column



Two column



Three column



Four column



Define Layouts

Define Components

Dairy-free and delicious
milk chocolate



[VIEW ARTICLE >](#)

Deep mediterranean quiche

[VIEW RECIPE >](#)



Super easy vegetarian
pasta bake

Difficulty: Easy



[VIEW RECIPE >](#)



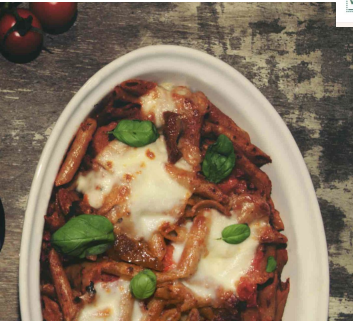
Fiery chili sauce

[VIEW RECIPE >](#)

Super easy vegetarian pasta
bake

A wholesome pasta bake is the ultimate comfort food. This delicious bake is super quick to prepare and an ideal midweek meal for all the family.

[View recipe](#)



Home

Explore recipes across every type of occasion, ingredient, and skill level.

Name Layouts & Components

1. Non-technical people consume the names.
2. KISS (Keep it Stupid Simple*)
3. Keep it obvious
4. Keep it descriptive
5. Plan for reuse
6. Proper level of specificity



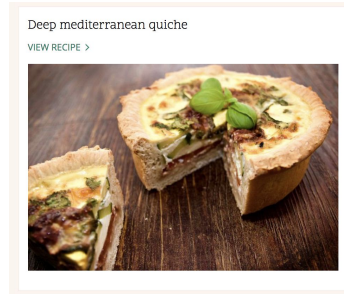
2 Col 75 - 25

Map Components to Content

A single item of content (recipe) can be displayed in many ways.

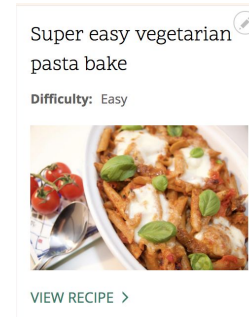
Identify and document each time the content item can be displayed.

- Full Content
- Card
- Teaser



Fiery chili sauce

VIEW RECIPE >



VIEW RECIPE >

Deep mediterranean quiche

An Italian inspired quiche with sun dried tomatoes and courgette. A perfect light meal for a summer's day.

Recipe category: [Main courses](#)

Tags: [Pastry](#) [Egg](#)



Preparation time: 40 minutes

Cooking time: 30 minutes

Number of servings: 8

Difficulty: Medium

Ingredients

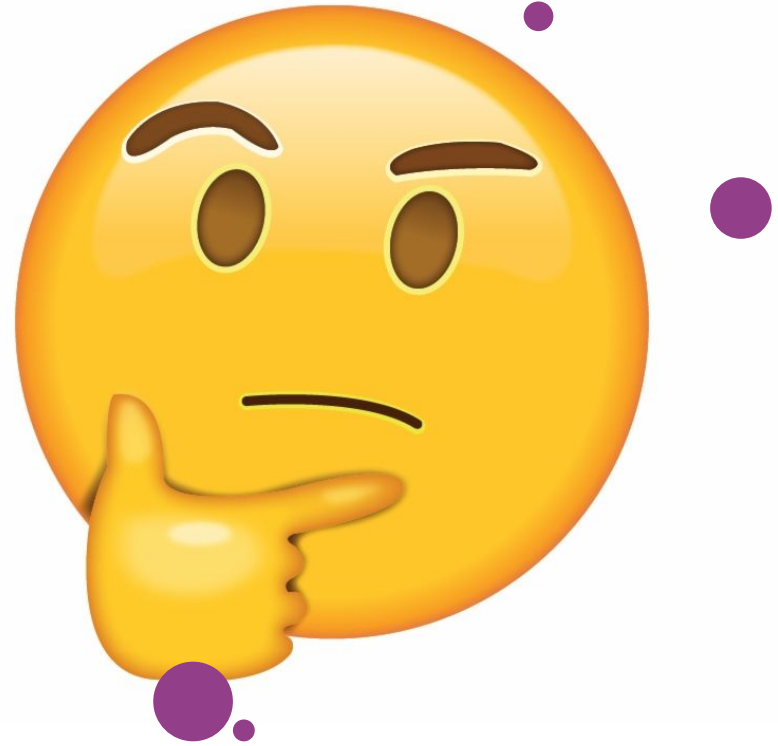
For the pastry:	Half a courgette
280g plain flour	450ml soya milk
140g butter	500g grated parmesan
Cold water	2 eggs
For the filling:	200g sun dried tomatoes
1 onion	100g feta
2 garlic cloves	

Recipe instruction

- 1 Preheat the oven to 400°F/200°C. Starting with the pastry; rub the flour and butter together in a bowl until crumbling like breadcrumbs. Add water, a little at a time, until it forms a dough.

Considerations

- Avoid too much specificity
- Leverage existing best practices
- Focus on reusability & patterns



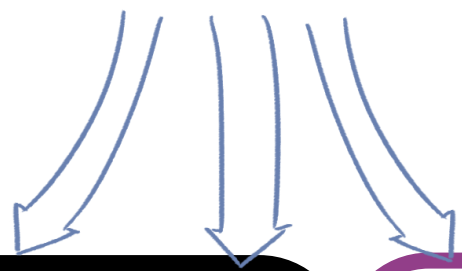
View Modes

“Definition”

“It is all the stuff on the
Manage Display tab.”

What is a View Mode?

A **View Display** is the **rendered output** of an entity following the **configuration rules** defined in a **View Mode**.

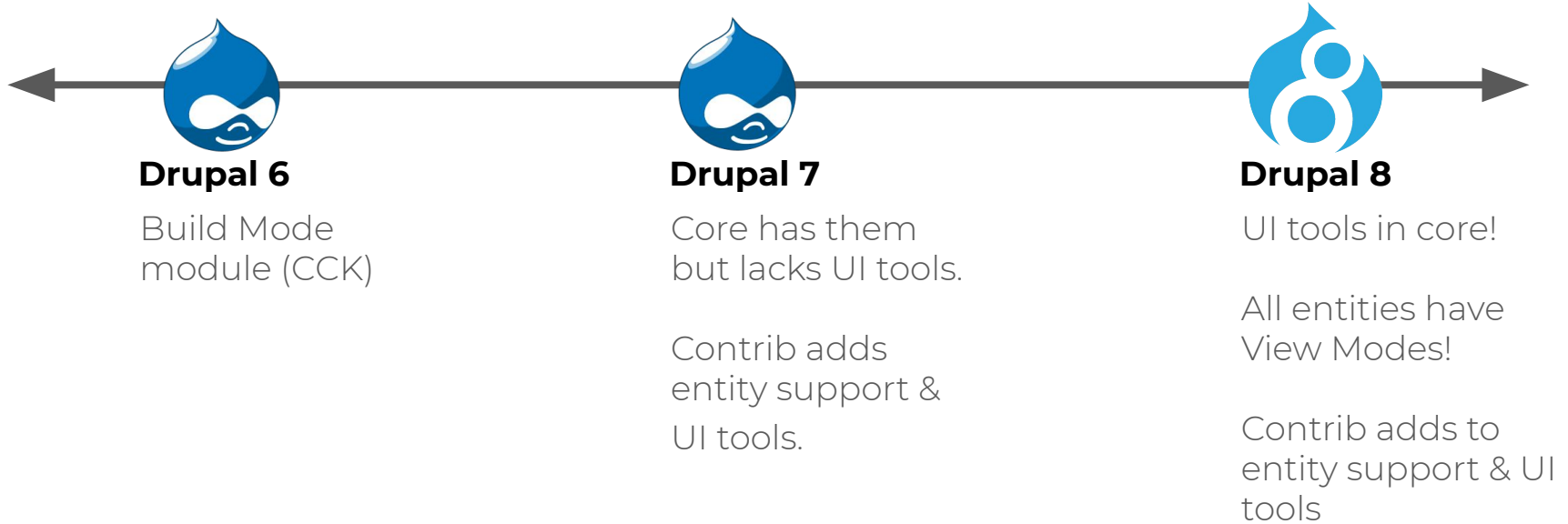


View Display

View Mode



Brief History of View Modes



View Modes are useful!

- Configure layouts with no code!
- Provide multiple layouts for the same content!
- Add a layer of tools for content display:
 - QuickEdit
 - Layout configuration
 - Minimal styles
 - Field formatter



Where do View Modes Live?

Structure > Display modes >
View Modes

`/admin/structure/display-modes/view`

Prerequisites

- Name
- Bundle Type

View modes ☆

Home > Administration > Structure > Display modes

+ Add view mode

Content

NAME	OPERATIONS
Full content	Edit ▾
RSS	Edit ▾
Search index	Edit ▾
Search result highlighting input	Edit ▾
Teaser	Edit ▾

[Add new Content view mode](#)

Custom block

NAME	OPERATIONS
------	------------

Why don't they show up?

Enable the View Modes Display!

1. Structure > Content types > [type] > Manage display
2. Open CUSTOM DISPLAY SETTINGS fieldset.
3. Check the View Modes you would like to configure for this content type.

Content items can be displayed using different view modes: Teaser, Full content, Print, RSS, etc. Teaser is a short format that is typically used in lists of multiple content items. Full content is typically used when the content is displayed on its own page.

Here, you can define which fields are shown and hidden when Product content is displayed in each view mode, and define how the fields are displayed in each view mode.

FIELD	LABEL	FORMAT	
+ Subtitle	- Hidden -	Plain text	⚙
+ Body	- Hidden -	Default	
+ Hero Image	- Hidden -	Image	Image style: Large (480x480) Linked to file ⚙
+ Currency	Above	Default	
+ Interests	Inline	Label	Link to the referenced entity ⚙
+ Price	Above	Default	1234 Display with prefix and suffix. ⚙
+ Product Id	Inline	Plain text	⚙
+ Links			
Disabled			
+ Language	Above	Language	

CUSTOM DISPLAY SETTINGS

Use custom display settings for the following view modes

- Teaser Short
- Full content
- RSS

Checked here.

Applied View Modes

Site: Unami Drupal 8 Install

Content Type: Article

View: Article aside / Card

Block: not in Unami

Custom Entity: not in Unami

Manage display ☆

Edit Manage fields Manage form display Manage display Translate content type

Default Card Card comment Card comment with image Full content RSS Teaser

View Mode

[Home](#) » [Administration](#) » [Structure](#) » [Content types](#) » [Article](#)

Content items can be displayed using different view modes: Teaser, Full content, Print, RSS, etc. *Teaser* is a short format that is typically used in lists of multiple content items. *Full content* is typically used when the content is displayed on its own page.

Here, you can define which fields are shown and hidden when *Article* content is displayed in each view mode, and define how the fields are displayed in each view mode.

[Show row weights](#)

FIELD	LABEL	FORMAT	
⊕ Tags	Above	Label	Link to the referenced entity
⊕ Image	- Hidden -	Responsive image	Responsive image style: 3:2 Image
⊕ Body	- Hidden -	Default	
⊕ Links			
Disabled			
⊕ Language	Above	Language	
⊕ Moderation control			

▶ CUSTOM DISPLAY SETTINGS

▶ LAYOUT OPTIONS

Save

Articles aside (Content) ☆

Edit Translate view

Home » Administration » Structure » Views

Displays

Block + Add

Display name: Block

TITLE
Title: More featured articles

FORMAT
Format: Unformatted list | Settings
Show: Content | Card

FIELDS
The selected style or row format does not use fields.

FILTER CRITERIA Add ▾
Content: Published (= Yes)
Content: Content type (= Article)
Content: Translation language (= Content language selected for page)

SORT CRITERIA Add ▾
Content: Authored on (desc)
Content: ID (asc)

BLOCK SETTINGS
Block name: Articles aside
Block category: Lists (Views)
Allow settings: Items per page
Access: Permission | View published content

HEADER Add

FOOTER Add

NO RESULTS BEHAVIOR Add

PAGER
Use pager: Display a specified number of items | 3 items
More link: No
Link display: None

LANGUAGE
Rendering Language: Content language of view row

▶ ADVANCED

Save Cancel

Block: Row style options

View mode
Card ▾

Apply Cancel

Block: Row style options




- Default
- ✓ Card
- Card common
- Card common alt
- Full content
- RSS
- Search index
- Search result highlighting input
- Teaser

Manage display ☆

[Edit](#)[Manage fields](#)[Manage form display](#)[Manage display](#)[Translate custom block type](#)

[Home](#) » [Administration](#) » [Structure](#) » [Block layout](#) » [Custom block library](#) » [Edit Card](#)

[Show row weights](#)

FIELD	LABEL	FORMAT
 Referenced Content	<input type="text" value="- Hidden -"/>	<input type="text" value="Rendered entity"/> Rendered as Card 
Disabled		
 Language	<input type="text" value="Above"/>	<input type="text" value="Language"/>

▶ CUSTOM DISPLAY SETTINGS

▼ LAYOUT OPTIONS

Use Layout Builder

View Mode

[Save](#)

Name Your View Modes

1. Naming is important and is hard. Use industry standards.
2. “Teaser 1” doesn’t mean anything.
3. “Article Teaser”? Why?! One view mode per content type may not be necessary. You can reuse view modes!
4. Name them in a Broad to Narrow fashion.
5. Reduce unnecessary View Modes to reduce clutter.



Hello
my name is

2 Col 75 - 25

Words.

- **Full content** displays the most content of a single entity. Commonly **“the” URL for the content.** (Default)
- **Teasers** display content in a **listing** format.
- **Cards** display content in a **grid** format.
- **Banners** display content in **banded layouts** for call to actions. Used in cross-linking and lead generation.
- **Admin** variants display content used by internal teams to manage content. Inventory, tagging, tech specs.

Naming Convention



[**Content** Specificity]

+ [**Entity** Specificity]

+ [View Mode **Family**]

+ [View Mode **Variant**]

+ [**Display Context**]



Domain	Display Modes	
Kingdom	View Modes Form Modes	
Phylum	Content Type Block Term Comment	Per content entity
Class	Teaser Full Content Banner Card Admin	Per display similarity
Order	Short Teaser Long Teaser	Variant per display class
Family	<p>Product Short Teaser</p> <p><i>Commonly becomes the conversational name/ticket name of a view mode applied to content.</i></p> <p><i>Represents specific content entity + specific display configuration</i></p>	<p>Content entity + Display</p> <p>Differentiator: Variant per display “class” and per entity bundle</p> <p>Example: A view mode applied to content.</p>

[Content Specificity] + [Entity Specificity] + [View Mode **Family**] + [View Mode **Variant**] + [Display Context]



Domain	Display Modes	
Kingdom	View Modes Form Modes	
Phylum	Content Type Block Term Comment	Per content entity
Class	Teaser Full Content Banner Card Admin	Per display similarity
Order	Short Teaser	Variant per display class
Family	Product Short Teaser	Content entity + Display
Genus	Video Product Short Teaser Clothing Product Short Teaser	Content Specificity + Content Entity + Display
Species	<p>Video Product Short Teaser Short - ES</p> <p><i>Conversational name for project management / tickets.</i></p> <p>Implementation would be: <i>Video Short Teaser View Mode - ES applied to Product Content Type</i></p>	<p>Differentiator: Content Specificity + Content Entity + Display + Display Context</p> <p>Example: Specific type of content has a very specific display variant that is different per language. (German)</p>

Considerations

- Avoid too much specificity
- Leverage existing best practices
- Focus on reusability & patterns
- Focus on user success

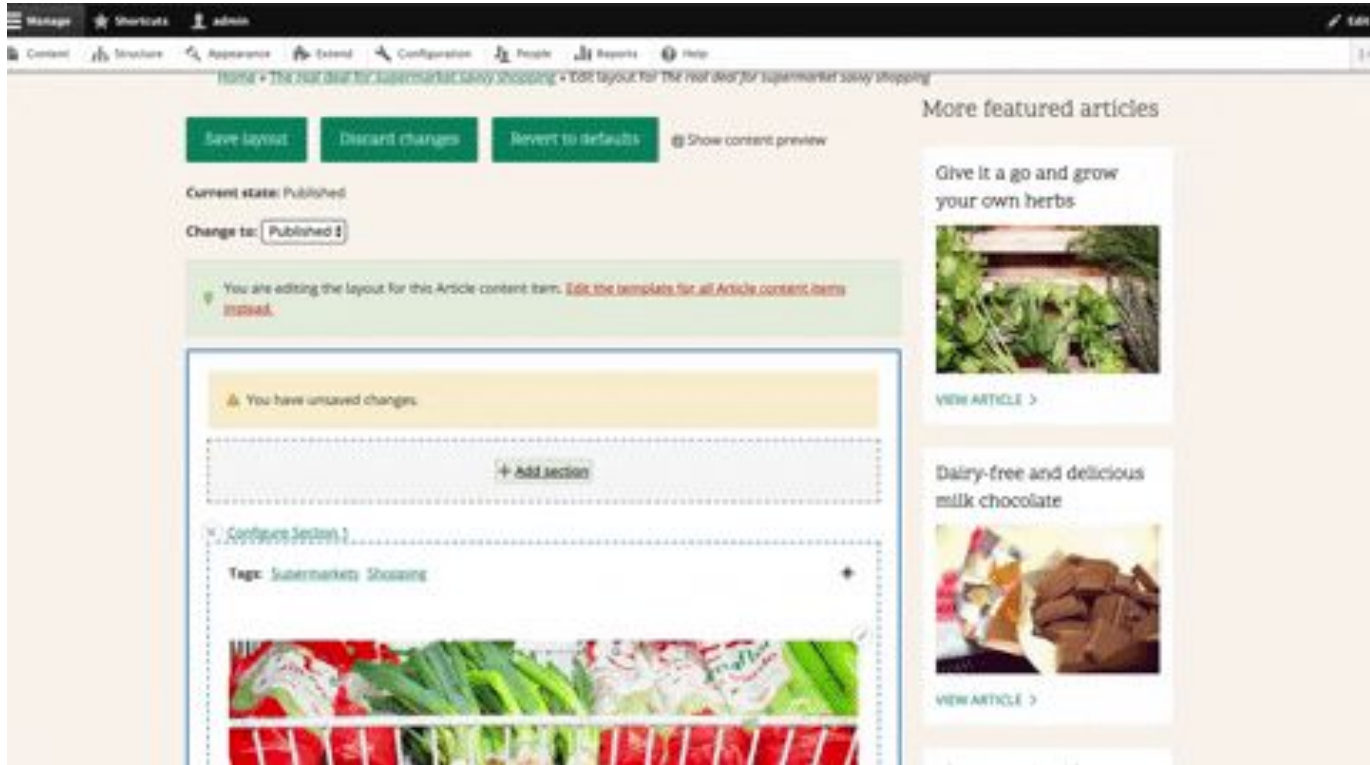


Layout Builder

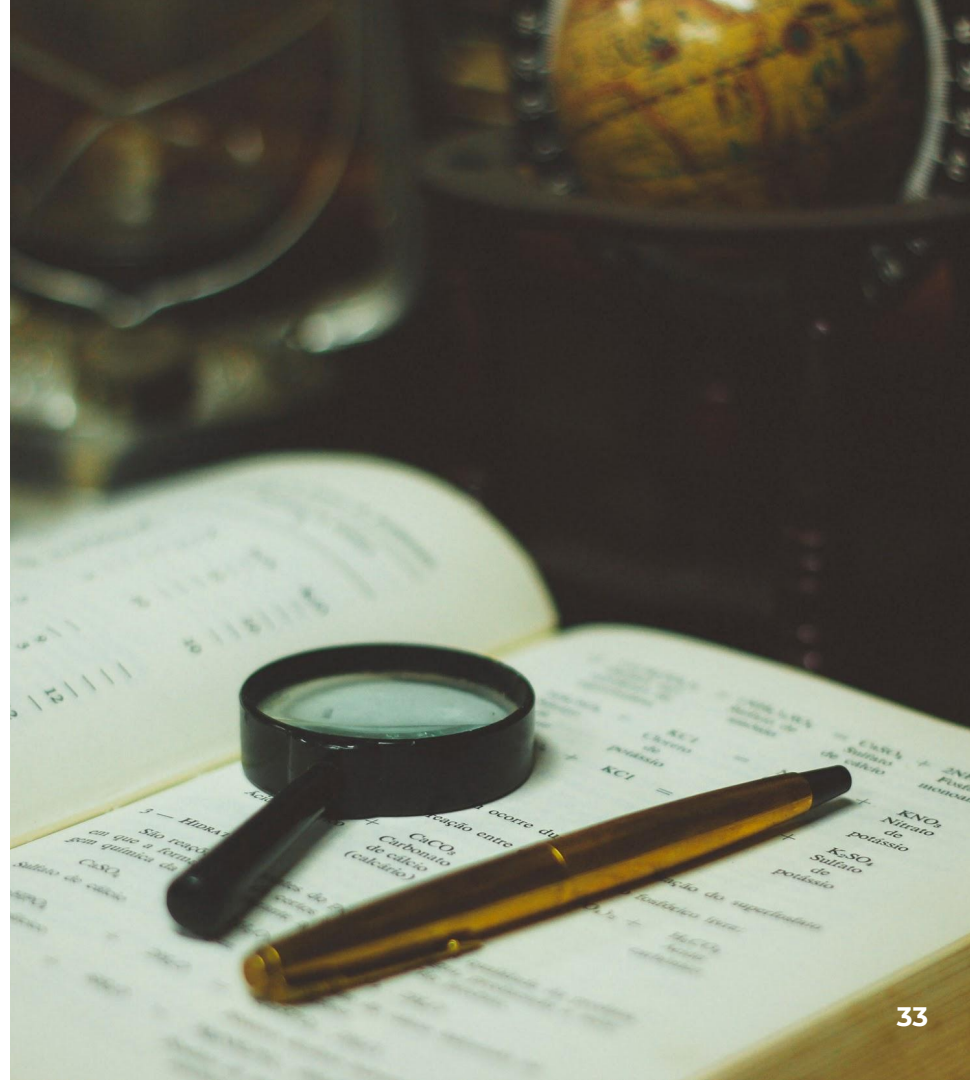
What is Layout Builder?

Layout Builder is new **display management interface** for Drupal 8, allowing **users** to drag-and-drop fields, blocks, and other content to **build page layouts** in the browser.

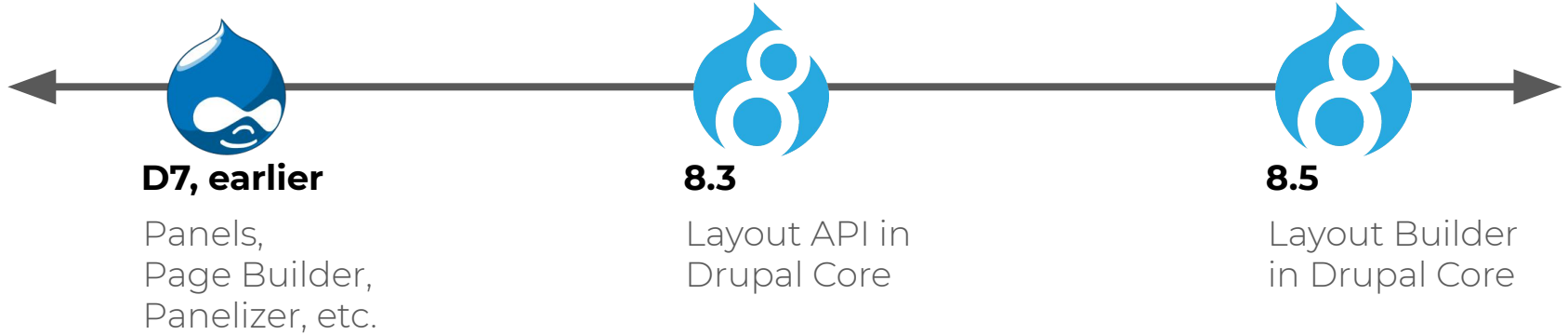
It looks like this!



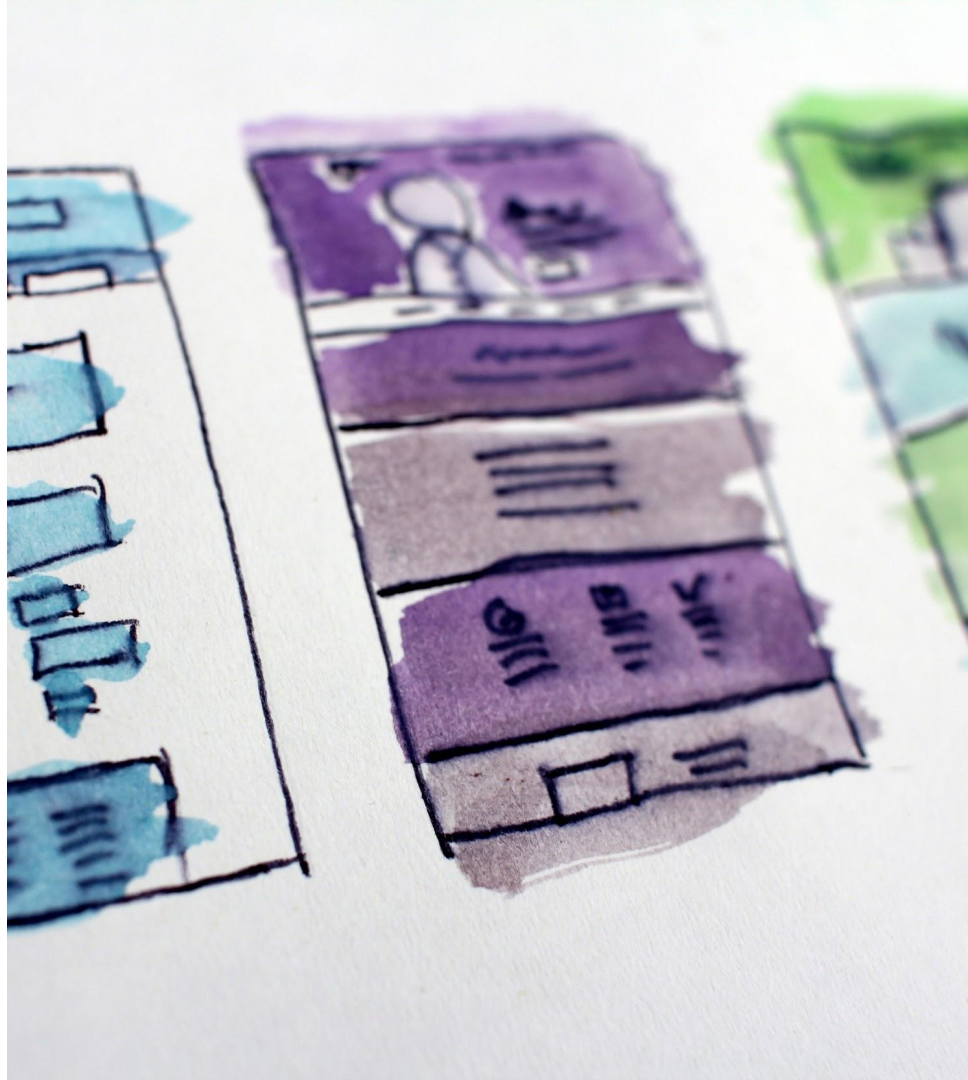
A Brief History of Layout Builder



Brief History of Layout Builder



What's in a Layout?



What's in a Layout?

- Twig (template)
- YAML (definition)
- CSS/JS (styling, etc.)

```
{% if content %}
<div {{ attributes.addClass('two-column') }}>
{% if content.main %}
  <div {{ region_attributes.main.addClass('main-region') }}>
    {{ content.main }}
  </div>
{% endif %}
{% if content.sidebar %}
  <div {{ region_attributes.sidebar.addClass('sidebar-region') }}>
    {{ content.sidebar }}
  </div>
{% endif %}
</div>
{% endif %}
```

```
advanced_layout_2:
  label: Advanced Layout 2
  category: My Layouts
  template: templates/advanced-layout-2
  # Replace "MY_MODULE_OR_THEME" with the machine name of your module
  # or theme.
  library: MY_MODULE_OR_THEME/advanced-layout-library
  regions:
    main:
      label: Main content
```

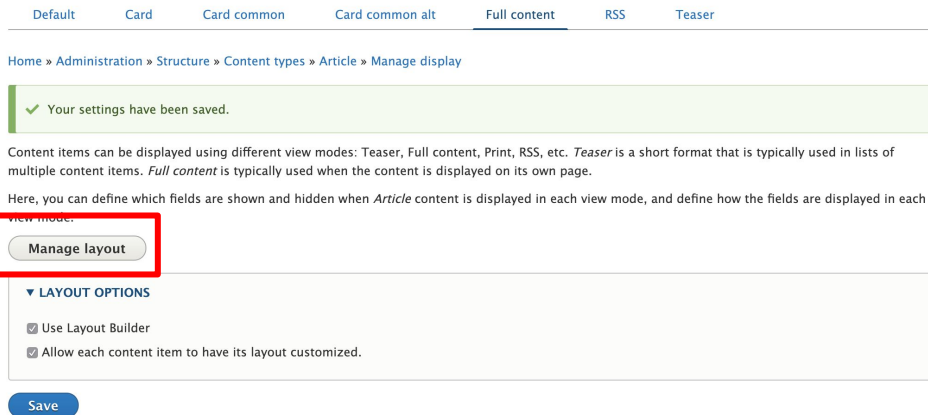
Where does Layout Builder Live?



Two Places to Use Layout Builder: Pt 1

Layout Defaults:

- Drop-in replacement for “Manage Display”
- Drag-and-drop placement of content fields (from edit form)
- Optional placement of additional blocks and content.
- Applied as the “default” layout



Default Card Card common Card common alt Full content RSS Teaser

Home » Administration » Structure » Content types » Article » Manage display

✓ Your settings have been saved.

Content items can be displayed using different view modes: Teaser, Full content, Print, RSS, etc. *Teaser* is a short format that is typically used in lists of multiple content items. *Full content* is typically used when the content is displayed on its own page.

Here, you can define which fields are shown and hidden when *Article* content is displayed in each view mode, and define how the fields are displayed in each view mode.

Manage layout

▼ LAYOUT OPTIONS

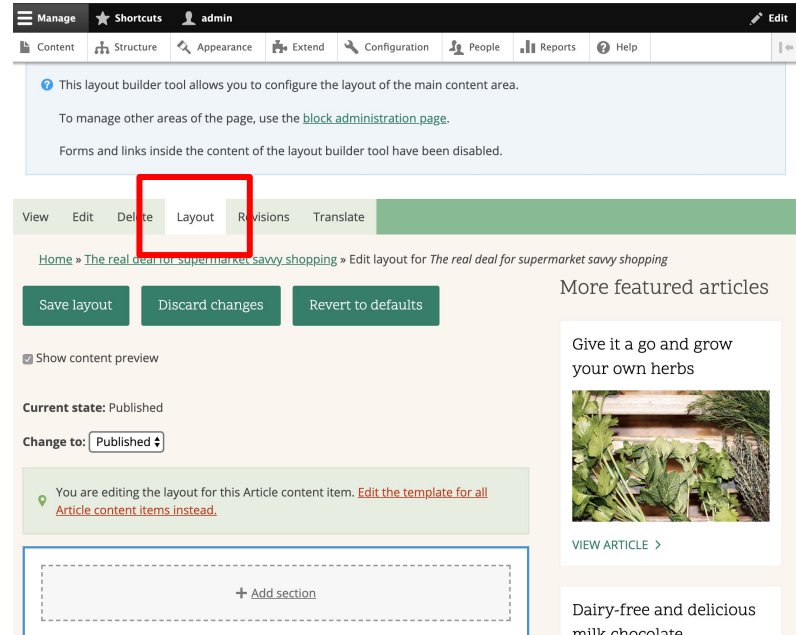
- Use Layout Builder
- Allow each content item to have its layout customized.

Save

Two Places to Use Layout Builder: Pt 1

Layout Overrides:

- Custom, one-off layout for one specific piece of content.
- Can remove, add, or extend content from default layout.



The screenshot shows the Drupal administration interface for editing a content item. At the top, there is a navigation bar with tabs for 'Content', 'Structure', 'Appearance', 'Extend', 'Configuration', 'People', 'Reports', and 'Help'. Below this, a blue informational box explains the layout builder tool. A red box highlights the 'Layout' tab in the top navigation bar. Below the tabs, there are buttons for 'Save layout', 'Discard changes', and 'Revert to defaults'. A 'Show content preview' checkbox is checked. The 'Current state' is 'Published', and the 'Change to' dropdown is set to 'Published'. A green warning box states: 'You are editing the layout for this Article content item. [Edit the template for all Article content items instead.](#)' At the bottom, there is a dashed box with a '+ Add section' button. On the right side, there is a 'More featured articles' section with a featured article titled 'Give it a go and grow your own herbs' and a 'VIEW ARTICLE >' link. Below that, there is another article titled 'Dairy-free and delicious milk chocolate'.

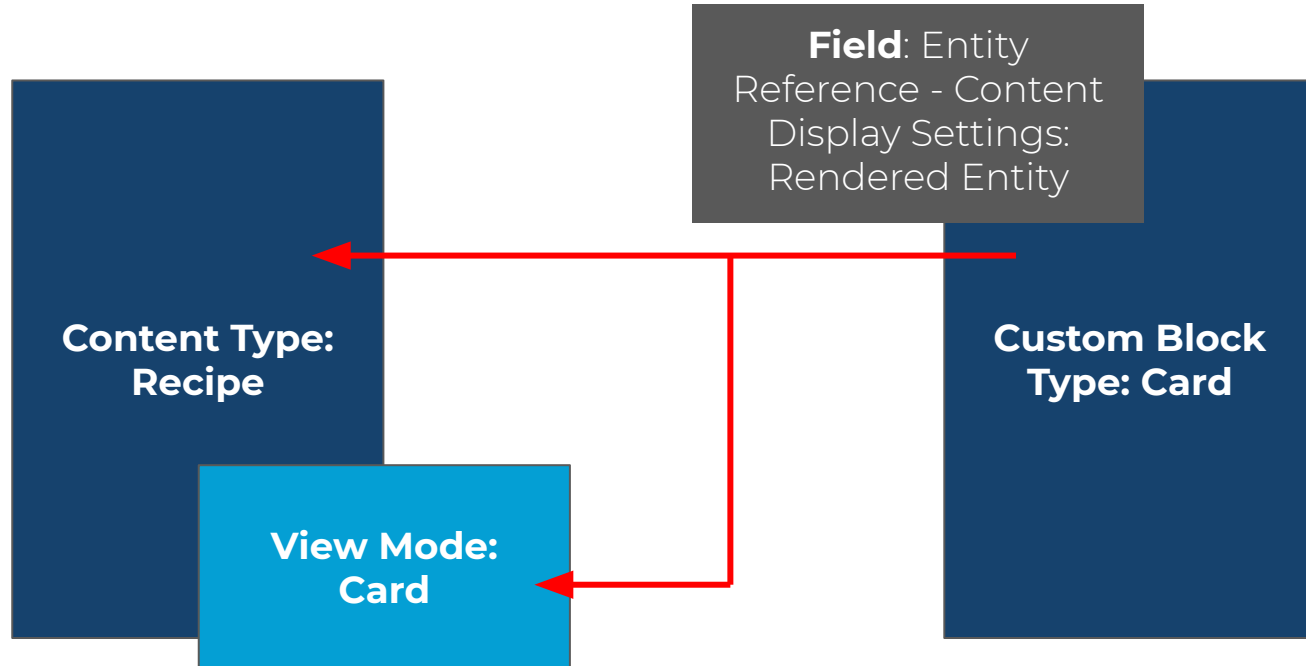
View Modes + Layout Builder Basics

How can we leverage our existing content (nodes) to generate Layout Builder Content?



Map **view modes** to **custom block types**, using an **entity reference** to render the entity with a given view mode.

View Mode/Custom Block Type Mapping





HOOK42

Demo time!

Considerations

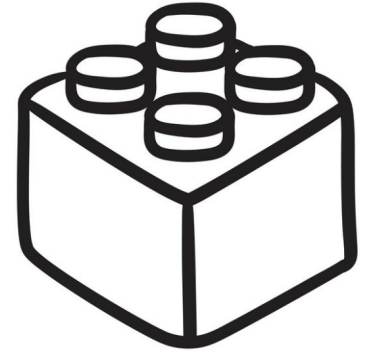
- Understand your applied scope
- Consistent location of customizations (config vs. css)
- Permissions
- Visibility to editor



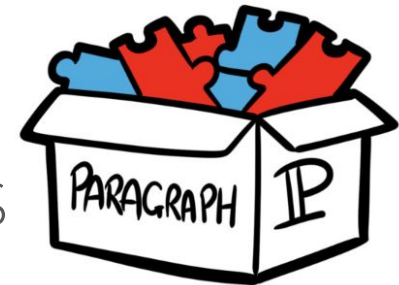
Leveled Up Layouts

Layout Builder vs. Paragraphs

- You can use them together (contrib)
- Custom Blocks for “chunks” of content
- Don't have to install more modules!
- Less performance impact with native blocks
- In-context UI vs. form-field only interface
- Still can leverage Twig templates and CSS

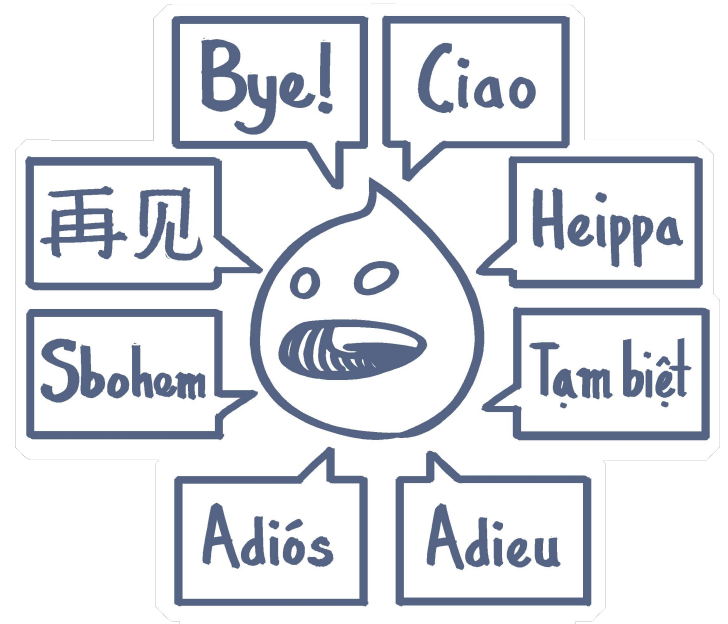


VS.



Layout Builder & Multilingual

- Core Multilingual
- Contrib Extensions
- Translation (same layout, translated words)
- Localization (different layout per language)
- Mindful multilingual configuration



Considerations

- Beware of block-ception
- Defaults vs. overrides in the wild
- Complexity & confusion
- Document your implementation
- Content Editors are using it



Takeaways

Something Witty.

- Layout Builder is constantly improving.
- Using it will help improve it.
- Create your strategy BEFORE site building.
- Read the CURRENT docs and watch the CURRENT videos.
- Contribute to the issue queues.



Questions?



HOOK**42**

Fin.