

Twig Recipes

Making Drupal 8 Render the Markup You Want



<http://bit.ly/twig-recipes>

Mauricio Dinarte

@dinarcon

mauricio@agaric.com

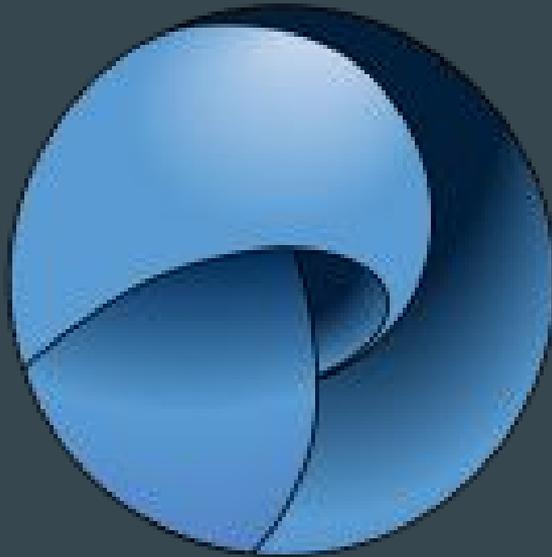
<https://www.drupal.org/u/dinarcon>



Nicaragua



Agaric

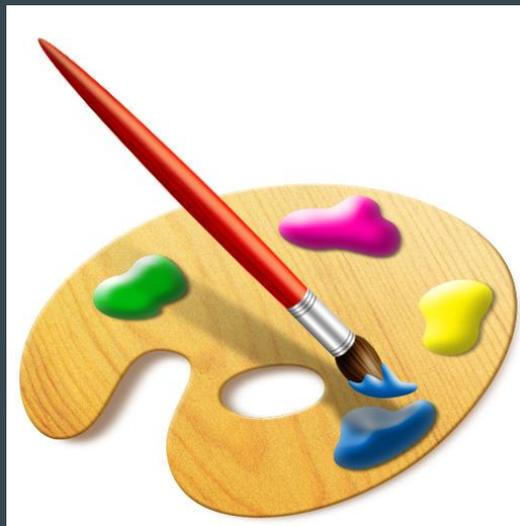


USA | Nicaragua | México | Germany | England

Themes

...

Control appearance



A theme...

- Defines the layout of the website
- Sets the color scheme
- Specifies the fonts to be used
- Might provide interactive user experiences

<https://www.drupal.org/theme-guide/8>

<http://sqndr.github.io/d8-theming-guide>

Templates

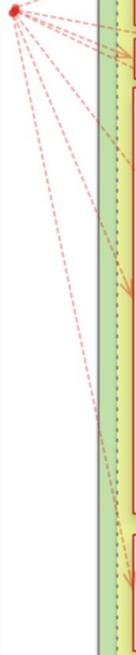
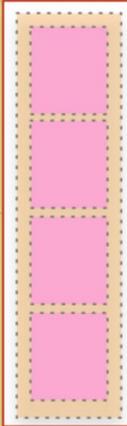
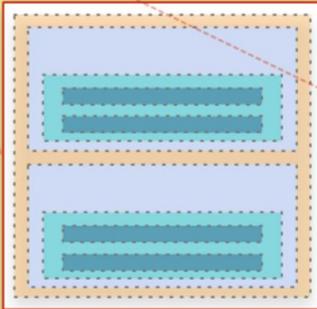
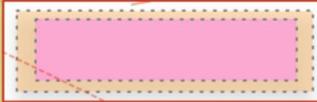
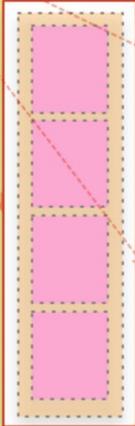
They separate the presentation logic from the business logic.

It consists of HTML markup with placeholders that would be replaced by dynamic content.

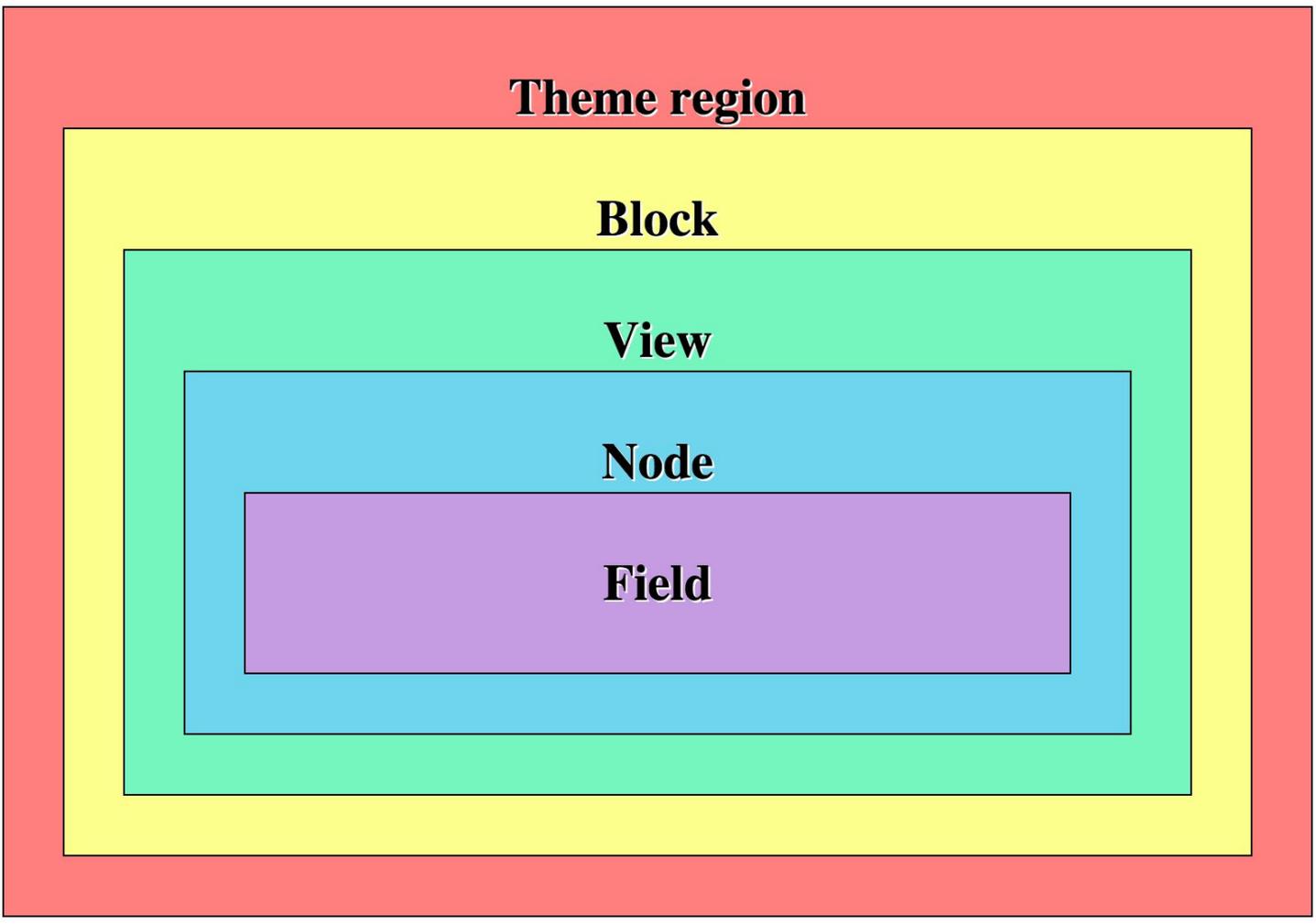
CSS, JavaScript, and other assets are added as needed.



Drupal



Theme region



```
graph TD; Theme[Theme region] --- Block[Block]; Block --- View[View]; View --- Node[Node]; Node --- Field[Field];
```

Block

View

Node

Field

```
<!DOCTYPE html>
<html{{ html_attributes }}>
  <head>
    <head-placeholder token="{{ placeholder_token|raw }}">
    <title>{{ head_title|safe_join(' | ') }}</title>
    <css-placeholder token="{{ placeholder_token|raw }}">
    <js-placeholder token="{{ placeholder_token|raw }}">
  </head>
  <body{{ attributes }}>
    {#
      Keyboard navigation/accessibility link to main content section in
      page.html.twig.
    #}
    <a href="#main-content" class="visually-hidden focusable">
      {{ 'Skip to main content'|t }}
    </a>
    {{ page_top }}
    {{ page }}
    {{ page_bottom }}
    <js-bottom-placeholder token="{{ placeholder_token|raw }}">
  </body>
</html>
```

html.html.twig

```
<div class="layout-container">
  <header role="banner">
    {{ page.header }}
  </header>
  {{ page.primary_menu }}
  {{ page.secondary_menu }}

  {{ page.breadcrumb }}

  {{ page.highlighted }}

  {{ page.help }}

  <main role="main">
    <a id="main-content" tabindex="-1"></a>{# link is in html.html.twig #}

    <div class="layout-content">
      {{ page.content }}
    </div>{# /.layout-content #}

    {% if page.sidebar_first %}
      <aside class="layout-sidebar-first" role="complementary">
        {{ page.sidebar_first }}
      </aside>
    {% endif %}

    {% if page.sidebar_second %}
      <aside class="layout-sidebar-second" role="complementary">
        {{ page.sidebar_second }}
      </aside>
    {% endif %}
  </main>

  {% if page.footer %}
    <footer role="contentinfo">
      {{ page.footer }}
    </footer>
  {% endif %}
</div>{# /.layout-container #}
```

page.html.twig

```
{% if content %}  
  <div{{ attributes }}>  
    {{ content }}  
  </div>  
{% endif %}
```

region.html.twig

```
<div{{ attributes }}>
  {{ title_prefix }}
  {% if label %}
    <h2{{ title_attributes }}>{{ label }}</h2>
  {% endif %}
  {{ title_suffix }}
  {% block content %}
    {{ content }}
  {% endblock %}
</div>
```

block.html.twig

```
<article{{ attributes }}>

  {{ title_prefix }}
  {% if not page %}
    <h2{{ title_attributes }}>
      <a href="{{ url }}" rel="bookmark">{{ label }}</a>
    </h2>
  {% endif %}
  {{ title_suffix }}

  {% if display_submitted %}
    <footer>
      {{ author_picture }}
      <div{{ author_attributes }}>
        {% trans %}Submitted by {{ author_name }} on {{ date }}{% endtrans %}
        {{ metadata }}
      </div>
    </footer>
  {% endif %}

  <div{{ content_attributes }}>
    {{ content }}
  </div>

</article>
```

node.html.twig

Twig

...

Twig

Twig is a template engine for PHP and it is part of the Symfony framework. In Drupal 8 Twig replaces PHPTemplate as the default templating engine.

<https://www.drupal.org/theme-guide/8/twig>

Twig Template

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     {% block head %}
5       <link rel="stylesheet" href="style.css" />
6       <title>{% block title %}{% endblock %} - Twig Recipes</title>
7     {% endblock %}
8   </head>
9   <body>
10    <div id="content">{% block content %}{% endblock %}</div>
11    <div id="footer">
12      {% block footer %}
13        This work is licensed under a
14        <a rel="license" href="http://creativecommons.org/licenses/by-nc-sa/4.0/">
15          Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License
16        </a>.
17      {% endblock %}
18    </div>
19  </body>
20 </html>
```

<http://twig.sensiolabs.org/doc/templates.html>

Variables

```
1  {{ node.bundle }}
2  {{ node['bundle'] }}
3  {{ attribute(node, 'bundle') }}
4
5  {# equivalent to the non-working form.operator.#type #}
6  {{ form.operator['#type'] }}
7
8  {# equivalent to the non-working foo.data-foo #}
9  {{ attribute(foo, 'data-foo') }}
```

```
1 // For {{ node.bundle }} Twig will perform the following
2 // checks and return the first that is valid.
3
4 // Check if node is an array and bundle a valid element.
5 return node['bundle'];
6 // If not, and if node is an object, check that bundle is a valid property.
7 return node->bundle;
8 // If not, and if node is an object, check that bundle is a valid method.
9 return node->bundle();
10 // If not, and if node is an object, check that getBundle is a valid method.
11 return node->getBundle();
12 // If not, and if node is an object, check that isBundle is a valid method.
13 return node->isBundle();
14 // If not, and the variable passed is a render array, call the render method on it.
15 return render(node->bundle); // This is Drupal specific.
16 // If not, return a NULL value.
17 return NULL;
```

Setting Variables

```
1  {%  
2    set classes = [  
3      'node',  
4      'node--type-' ~ node.bundle|clean_class,  
5      node.isPromoted() ? 'node--promoted',  
6      node.isSticky() ? 'node--sticky',  
7      not node.isPublished() ? 'node--unpublished',  
8      view_mode ? 'node--view-mode-' ~ view_mode|clean_class,  
9    ]  
10  %}  
11  
12  <article{{ attributes.addClass(classes) }}>
```

Filters

Drupal specific filters <http://bit.ly/2kcWV0K>

```
1  {{ sometext|upper }}
2  {{ "This text becomes uppercase"|upper }}
3
4  {# Filters can be applied using the filter tag #}
5  {% filter upper %}
6     This text becomes uppercase
7  {% endfilter %}
8
9  {# Filters can be chained #}
10 {{ name|striptags|upper }}
11
12 {# Filters can receive parameters #}
13 {{ modules|safe_join(', ') }}
```

Functions

Drupal specific filters <http://bit.ly/2jLkizT>

```
1  {# Outputs a number between 0 and 7 inclusive #}
2  {{ random(7) }}
3
4  {# Outputs 0, 1, 2, 3, #}
5  {% for i in range(0, 7) %}
6      {{ i }},
7  {% endfor %}
8
9  {# Attaches a Drupal library #}
10 {{ attach_library('classy/node') }}
```

Tests

```
1  {% if q is not empty %}
2
3  {% if label_display in ['before', 'invisible'] %}
4
5  {% if description_display == 'before' and description.content %}
6
7  {% if (summary is not empty) or (description is not empty) %}
8
9  {% if some_number is odd %}
```

Filter or Function or Test?

	Filter	Function	Test
Requires an expression preceding it?	Yes	No	Yes
Can receive arguments?	Yes	Yes	Yes
Return value	Scalar	Mixed	Boolean

Control Structures

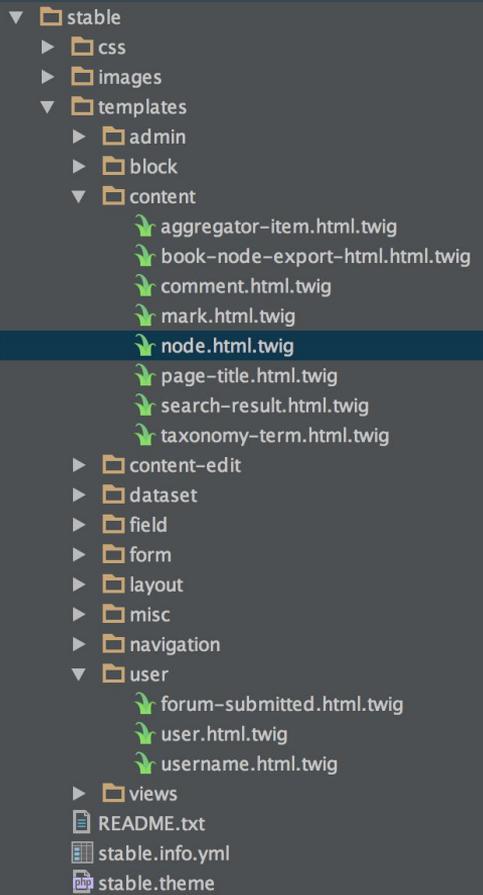
```
1  {% for item in breadcrumb %}
2    <li>
3      {% if item.url %}
4        <a href="{{ item.url }}">{{ item.text }}</a>
5      {% else %}
6        {{ item.text }}
7      {% endif %}
8    </li>
9  {% endfor %}
```

Control Structures

```
1  {% if messages|length > 1 %}  
2      <ul>  
3          {% for message in messages %}  
4              <li>{{ message }}</li>  
5          {% endfor %}  
6      </ul>  
7      {% else %}  
8          {{ messages|first }}  
9      {% endif %}  
10 {% endfor %}
```

Overriding Drupal Markup

...



148 templates in the **stable** Core theme

Theme Debug

```
twig.config:  
  debug: true  
  auto_reload: true  
  cache: true
```

<https://www.drupal.org/node/1906392>

<https://www.drupal.org/node/2358785>

Template Suggestions

```
▼<div id="block-bartik-content" class="block block-system block-system-main-block">
  ▼<div class="content">
    <!-- THEME DEBUG -->
    <!-- THEME HOOK: 'node' -->
    <!-- FILE NAME SUGGESTIONS:
      * node--1--full.html.twig
      * node--1.html.twig
      * node--article--full.html.twig
      * node--article.html.twig
      * node--full.html.twig
      x node.html.twig
    -->
    <!-- BEGIN OUTPUT from 'core/themes/bartik/templates/node.html.twig' -->
    ▼<article data-history-node-id="1" data-quickedit-entity-id="node/1" role="article" class="contextual-
region node node--type-article node--promoted node--view-mode-full clearfix" about="/node/1" typeof=
"schema:Article" data-quickedit-entity-instance-id="0">
```

<https://www.drupal.org/node/2354645>

<http://bit.ly/2aimNTu> (* views templates suggestions)



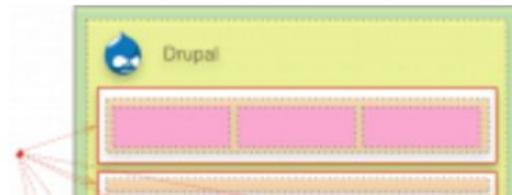
Twig Recipes

Making Drupal 8 render the markup that you want

Home

Templates

Submitted by [admin](#) on Thu, 07/21/2016 - 21:33



Used to render HTML markup.

```
▼<div id="block-bartik-content" class="block block-system block-system-main-block">
```

```
▼<div class="content">
```

```
  <!-- THEME DEBUG -->
```

```
  <!-- THEME HOOK: 'node' -->
```

```
  <!-- FILE NAME SUGGESTIONS:
```

```
    * node--1--full.html.twig
```

```
    * node--1.html.twig
```

```
    * node--article--full.html.twig
```

```
    * node--article.html.twig
```

```
    * node--full.html.twig
```

```
    x node.html.twig
```

```
  -->
```

```
  <!-- BEGIN OUTPUT from 'core/themes/bartik/templates/node.html.twig' -->
```

```
▼<article data-history-node-id="1" data-quickedit-entity-id="node/1" role="article" class="contextual-  
region node node--type-article node--promoted node--view-mode-full clearfix" about="/node/1" typeof=  
"schema:Article" data-quickedit-entity-instance-id="0">
```

```
90     {% if display_submitted %}
91     <div class="node__meta">
92         {{ author_picture }}
93         <span{{ author_attributes }}>
94             {% trans %}Submitted by {{ author_name }} on {{ date }}{% endtrans %}
95         </span>
96         {{ metadata }}
97     </div>
98     {% endif %}
```

`./core/themes/bartik/templates/node.html.twig`

```
6 * Available variables:
7 * - node: The node entity with limited access to object properties and methods.
8 * Only method names starting with "get", "has", or "is" and a few common
9 * methods such as "id", "label", and "bundle" are available. For example:
10 * - node.getCreatedTime() will return the node creation timestamp.
11 * - node.hasField('field_example') returns TRUE if the node bundle includes
12 * field_example. (This does not indicate the presence of a value in this
13 * field.)
14 * - node.isPublished() will return whether the node is published or not.
```

`./core/themes/bartik/templates/node.html.twig`

```
90  {% if display_submitted %}
91    <div class="node__meta">
92      {{ author_picture }}
93      <span{{ author_attributes }}>
94        {% set custom_date = node.createdTime|date("F d, Y") %}
95        {% trans %}Submitted by {{ author_name }} on {{ custom_date }}{% endtrans %}
96      </span>
97      {{ metadata }}
98    </div>
99  {% endif %}
```

./themes/YOUR_THEME/templates/node.html.twig

+ Rebuild caches



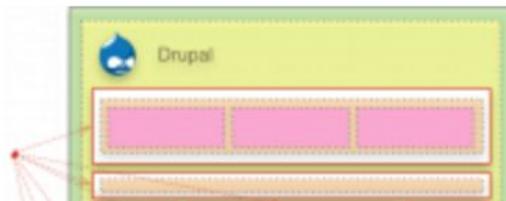
Twig Recipes

Making Drupal 8 render the markup that you want

Home

Templates

Submitted by [admin](#) on July 21, 2016



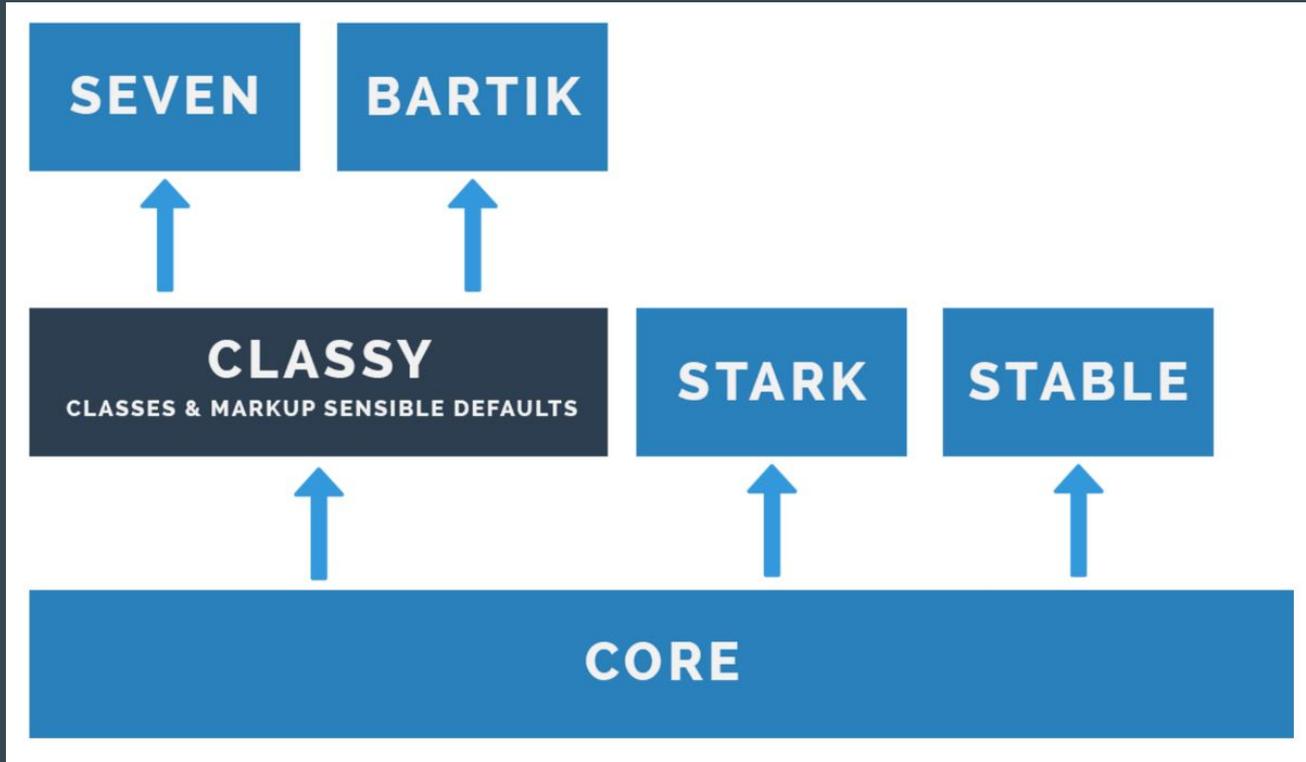
Used to render HTML markup.

All of these are equivalent

```
{# - node.getCreatedTime() will return the node creation timestamp. #}  
...  
{% set custom_date = node.createdTime|date("F d, Y") %}  
{% set custom_date = node.getCreatedTime|date("F d, Y") %}  
{% set custom_date = node.getCreatedTime()|date("F d, Y") %}
```

Warning!

For demonstration purpose only. DO NOT subtheme Bartik.



Twig Recipes

...

1. How can I pass information from Drupal to Twig?

- Modify the *.theme file.
- Implement `hook_preprocess_HOOK(&$variables)`
- Modify the `$variables` array. This is a key-value container of information. Keys will become variables in the template.
- Print the variable in the template.

1. How can I pass information from Drupal to Twig?

```
1 /**
2  * Implements template_preprocess_page().
3  */
4 function nicaragua_preprocess_page(&$variables) {
5     if (isset($variables['node']) && $variables['node']->bundle() == 'page') {
6         $variables['cover_image'] = $variables['node']->get('field_image')->view('full');
7         $variables['cover_title'] = $variables['node']->get('field_teaser')->view('full');
8     }
9 }
```

1. How can I pass information from Drupal to Twig?

```
1  {% if cover_image|render %}
2  <div class="wrapper cover" style="background-image: url('...');">
3    {% if cover_title|render %}
4      <div class="layout">
5        {{ cover_title }}
6      </div>
7    {% endif %}
8  </div>
9  {% endif %}
```

2. How can I create and assign classes directly in the Twig template?

- In the template itself, use **set** to create a variable that contains the class name.
- Print the variable where needed.
- Usually this happen in `{{ attributes.addClass(classes) }}`

2. How can I create and assign classes directly in the Twig template?

```
1  {%  
2      set classes = [  
3          'node',  
4          'node--type-' ~ node.bundle|clean_class,  
5          node.isPromoted() ? 'node--promoted',  
6          node.isSticky() ? 'node--sticky',  
7          not node.isPublished() ? 'node--unpublished',  
8          view_mode ? 'node--view-mode-' ~ view_mode|clean_class,  
9      ]  
10  %}  
11  
12  <article{{ attributes.addClass(classes) }}>
```

3. How can I render image fields as background images?

From Drupal 8.3 onwards

- Configure the image field to link to use the “URL to Image” format.
- Render the variable in the template.

<https://www.drupal.org/node/2824621>

3. How can I render image fields as background images?

FIELD	LABEL	FORMAT	
 Horizontal Image	- Hidden -	URL to image	Image style: Large (480x480) 

3. How can I render image fields as background images?

```
1 | <section class="feature_image"  
2 |   style="background-image: url('{% spaceless %}{{ content.field_horizontal_image }}{% endspaceless %}')";>  
3 |   ...  
4 | </section>
```

```
2 | url('{% spaceless %}{{ content.field_horizontal_image }}{% endspaceless %}')');
```

3. How can I render image fields as background images?

Before Drupal 8.3

- Install the Image URL Formatter module.
- Configure the image field to link to use the “Image URL” format and do not link to anything.
- Render the variable in the template.

3. How can I render image fields as background images?

A screenshot of a configuration dialog for a 'Horizontal Image' field. The dialog is light blue and contains several settings:

- Field Name:** Horizontal Image
- Visibility:** - Hidden -
- Format settings:** Image URL
- URL type:** Full URL
- Image style:** Hero Rectangle_996x498
- Link image to:** Nothing
- Buttons:** Update (blue), Cancel (light blue)

[Configure Image Styles](#)

3. How can I render image fields as background images?

```
1 | <section class="feature_image"  
2 |   style="background-image: url('{% spaceless %}{{ content.field_horizontal_image }}{% endspaceless %}')";>  
3 |   ...  
4 | </section>
```

```
2 | url('{% spaceless %}{{ content.field_horizontal_image }}{% endspaceless %}')');
```

4. How can I render node content as HTML attributes?

File field (SVG)

 Icon

```
1 
```

4. How can I render node content as HTML attributes?

Entity reference field as link



- Hidden -



Format settings: **Label**

Featured
content

Link label to the referenced entity

field--node--field-featured-content--feature.html.twig

```
1 | {% for item in items %}
2 |   {{ item.content['#url'] }}
3 | {% endfor %}
```

4. How can I render node content as HTML attributes?

```
1 <a title="Features content"  
2 href="{% spaceless %}{{ content.field_featured_content }}{% endspaceless %}"  
3 class="{{ lightbox_class }}">  
4     {{ content.field_teaser }}  
5 </a>
```

5. How can I conditionally render fields based on the content of other fields?

Boolean field



Lightbox

Format settings: **Boolean**

Output format

5. How can I conditionally render fields based on the content of other fields?

```
1 | {% set lightbox = content.field_lightbox|render|trim %}  
2 | {% if lightbox %}  
3 |     {{ content.field_video }}  
4 | {% endif %}
```

6. How can I use Twig filters to format numbers?

```
1  {{ 200.35|number_format }}
2  {{ 9800.333|number_format(2, '.', ',') }}
3
4  {#
5  Defaults:
6      0 decimal places.
7      . as the decimal point.
8      , as the thousands separator.
9  #}
```

7. How do I create an absolute link to a node?

```
1 | <a href="{ url('entity.node.canonical', {'node': node.id}) }" >{ 'Link text' | t } </a>
```

Drupal Console is your friend to find routes in the system.

```
$ drupal router:debug | grep '/node'
```

```
$ drupal router:debug entity.node.canonical
```

Want to learn more?

Mastering Twig

by Javier Eguiluz @ DrupalCon Barcelona 2015

<https://www.youtube.com/watch?v=4KfkV1T3bhE>

Want to learn more?

Drupalnator Themestation:

Understanding Drupal 8's New Theming Layers

by Darryl Norris @ Twin Cities DrupalCamp 2016

<https://www.youtube.com/watch?v=Ot--bH9379U>

Need support?

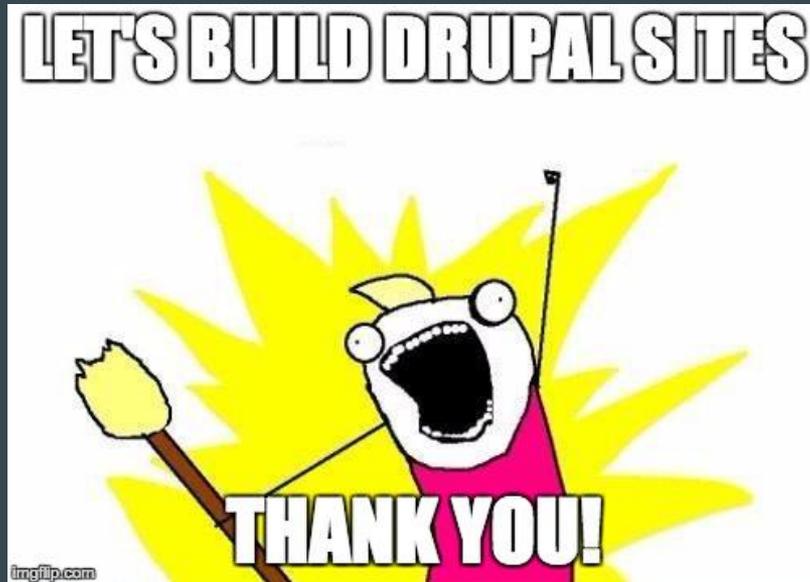
#drupal-twig @ freenode.net

<https://drupaltwig-slack.herokuapp.com>

Thank You!

Let's continue the conversation on twitter

#DrupalTwigRecipes



Please Provide Feedback

@dinarcon | #DrupalTwigRecipes | mauricio@agaric.com



<http://bit.ly/twig-recipes>