

# A NEW THEME LAYER FOR DRUPAL 8

Jen Lampton ~ @jenlampton



**DRUPAL + DESIGN + LOS ANGELES**

I AM @jenlampton

working with Drupal since D4.7 ~2006  
contrib module maintainer (youtube, total\_control)  
training instructor / curriculum author  
accidental “core developer”

I care a lot about Drupal’s “learnability”

WHO ARE YOU?



# YOU ARE:

Theme developers?  
(D7? D6? D5?)

# YOU ARE:

Theme developers?  
(D7? D6? D5?)

New to Drupal?  
(just want to know what the heck is going on in D8)

# YOU ARE:

Theme developers?  
(D7? D6? D5?)

New to Drupal?  
(just want to know what the heck is going on in D8)

Drupal Experts?  
(also want to know what the heck is going on in D8)



# DRUPAL 8 : A NEW THEME LAYER

DRUPAL 8 :  
A NEW THEME LAYER  
(probably)



WHY?

# DRUPAL 7: POSITIVE



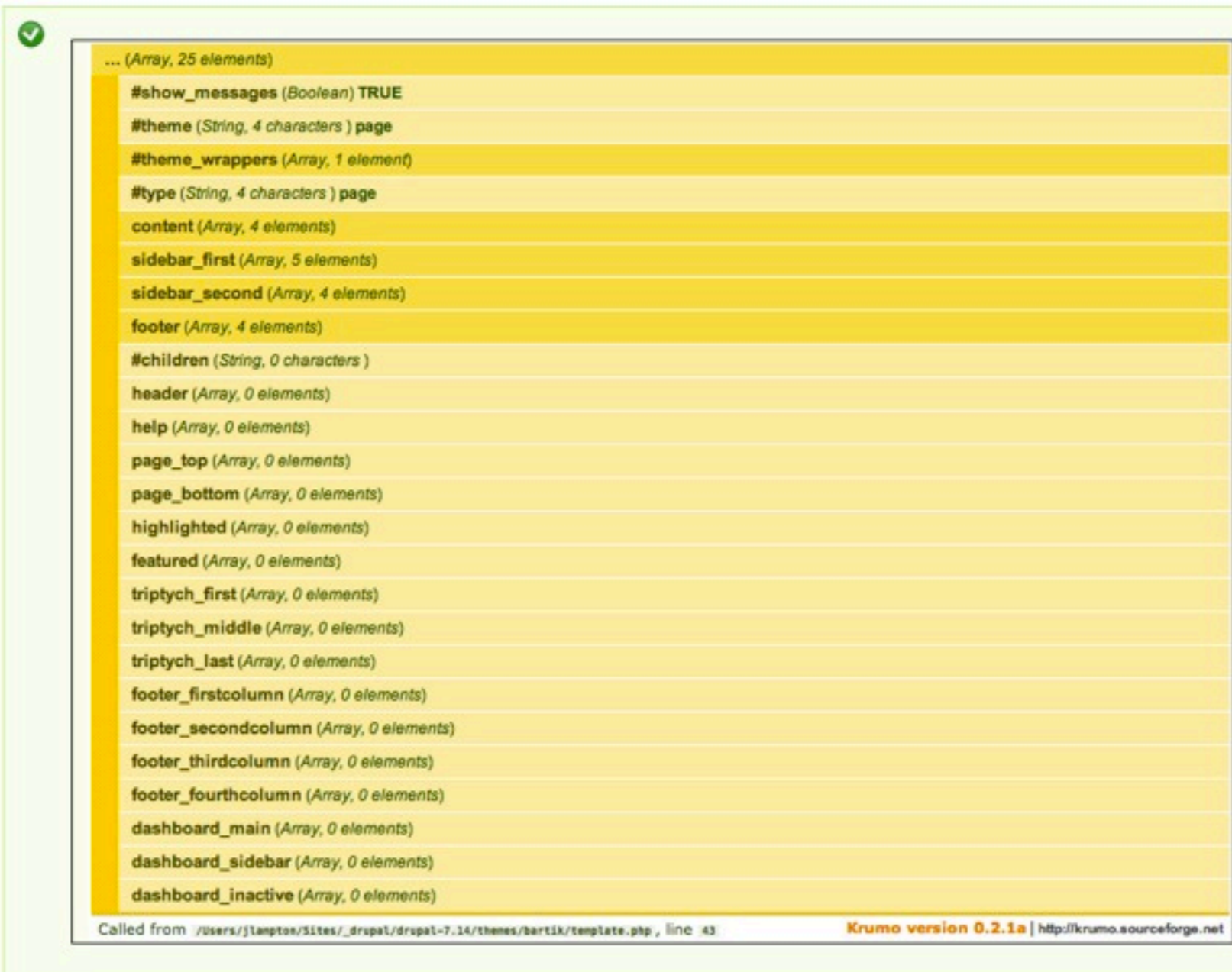
# DRUPAL 7: POSITIVE



flexible



# DRUPAL 7: POSITIVE



The screenshot shows a Krumo debug output for a Drupal 7 page. It displays a large array with 25 elements. The array contains various configuration and content-related data, including a boolean for showing messages, a string for the theme name, and several arrays representing content blocks and page structure. The Krumo version is 0.2.1a, and the output is called from a template file in Drupal 7.14.

```
... (Array, 25 elements)
#show_messages (Boolean) TRUE
#theme (String, 4 characters) page
#theme_wrappers (Array, 1 element)
#type (String, 4 characters) page
content (Array, 4 elements)
sidebar_first (Array, 5 elements)
sidebar_second (Array, 4 elements)
footer (Array, 4 elements)
#children (String, 0 characters)
header (Array, 0 elements)
help (Array, 0 elements)
page_top (Array, 0 elements)
page_bottom (Array, 0 elements)
highlighted (Array, 0 elements)
featured (Array, 0 elements)
triptych_first (Array, 0 elements)
triptych_middle (Array, 0 elements)
triptych_last (Array, 0 elements)
footer_firstcolumn (Array, 0 elements)
footer_secondcolumn (Array, 0 elements)
footer_thirdcolumn (Array, 0 elements)
footer_fourthcolumn (Array, 0 elements)
dashboard_main (Array, 0 elements)
dashboard_sidebar (Array, 0 elements)
dashboard_inactive (Array, 0 elements)

Called from /Users/jlampton/Sites/drupal/drupal-7.14/themes/bartik/template.php, line 43  Krumo version 0.2.1a | http://krumo.sourceforge.net
```

unified~ish data structures everywhere



# DRUPAL 7: POSITIVE



anything is possible



# DRUPAL 7: NEGATIVE



# DRUPAL 7: NEGATIVE

```
?>
<div id="node-<?php print $node->nid; ?>" class="<?php print $classes; ?> clearfix"<?php print $attributes; ?>>
  <?php print render($title_prefix); ?>
  <?php if (!$page): ?>
    <h2<?php print $title_attributes; ?>>
      <a href="<?php print $node_url; ?>"><?php print $title; ?></a>
    </h2>
  <?php endif; ?>
  <?php print render($title_suffix); ?>

  <?php if ($display_submitted): ?>
    <div class="meta submitted">
      <?php print $user_picture; ?>
      <?php print $submitted; ?>
    </div>
  <?php endif; ?>

  <?php if ($page['featured']): ?>
    <div id="featured"><div class="section clearfix">
      <?php print render($page['featured']); ?>
    </div></div> <!-- /.section, /#featured -->
  <?php endif; ?>
```

Drupal only (weird) syntax

# DRUPAL 7: NEGATIVE

```
?>
<div id="node-<?php print $node->nid; ?>" class="<?php print $classes; ?> clearfix"<?php print $attributes; ?>>
  <?php print render($title_prefix); ?>
  <?php if (!$page): ?>
    <h2<?php print $title_attributes; ?>>
      <a href="<?php print $node_url; ?>"><?php print $title; ?></a>
    </h2>
  <?php endif; ?>
  <?php print render($title_suffix); ?>

  <?php if ($display_submitted): ?>
    <div class="meta submitted">
      <?php print $user_picture; ?>
      <?php print $submitted; ?>
    </div>
  <?php endif; ?>

  <?php if ($page['featured']): ?>
    <div class="section clearfix">
      <?php print render($page['featured']); ?>
    </div></div> <!-- /.section, /#featured -->
  <?php endif; ?>
```

Object or Array?

# DRUPAL 7: NEGATIVE

```
?>
<div id="node-<?php print $node->nid; ?>" class="<?php print $classes; ?> clearfix"<?php print $attributes; ?>>
  <?php print render($title_prefix); ?>
  <?php if (!$page): ?>
    <h2<?php print $title_attributes; ?>>
      <a href="<?php print $node_url; ?>"><?php print $title; ?></a>
    </h2>
  <?php endif; ?>
  <?php print render($title_suffix); ?>
  <?php if ($display_submitted): ?>
    <div class="meta submitted">
      <?php print $user_picture; ?>
      <?php print $submitted; ?>
    </div>
  <?php endif; ?>
  <?php if ($page['featured']): ?>
    <div id="featured"><div class="section clearfix">
      <?php print render($page['featured']); ?>
    </div></div> <!-- /.section, /#featured -->
  <?php endif; ?>
```

print or print render() ?



# DRUPAL 7: NEGATIVE

```
<?php db_query("DROP TABLE {node}"); ?>
```

PHP is insecure

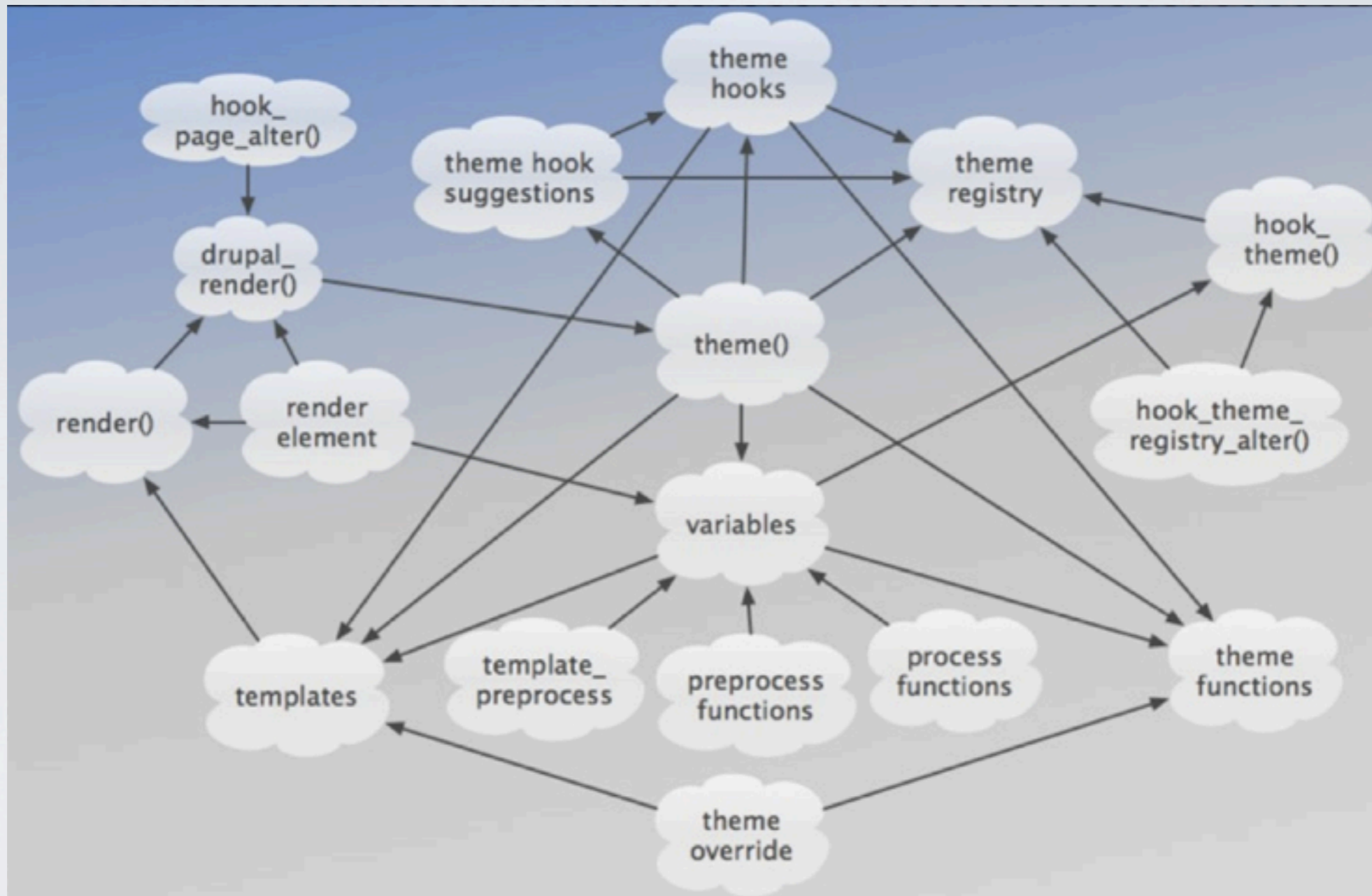
# DRUPAL 7: NEGATIVE

```
> find . -name "*.tpl.php"
./aggregator/aggregator-feed-source.tpl.php
./aggregator/aggregator-item.tpl.php
./aggregator/aggregator-summary-item.tpl.php
./aggregator/aggregator-summary-items.tpl.php
./aggregator/aggregator-wrapper.tpl.php
./block/block-admin-display-form.tpl.php
./block/block.tpl.php
./block/tests/themes/block_test_theme/page.tpl.php
./book/book-all-books-block.tpl.php
./book/book-export-html.tpl.php
./book/book-navigation.tpl.php
./book/book-node-export-html.tpl.php
./comment/comment-wrapper.tpl.php
./comment/comment.tpl.php
./field/theme/field.tpl.php
./forum/forum-icon.tpl.php
./forum/forum-list.tpl.php
./forum/forum-submitted.tpl.php
./forum/forum-topic-list.tpl.php
./forum/forums.tpl.php
./node/node.tpl.php
./overlay/overlay.tpl.php
./poll/poll-bar--block.tpl.php
./poll/poll-bar.tpl.php
./poll/poll-results--block.tpl.php
./poll/poll-results.tpl.php
./poll/poll-vote.tpl.php
./profile/profile-block.tpl.php
./profile/profile-listing.tpl.php
./profile/profile-wrapper.tpl.php
./search/search-block-form.tpl.php
./search/search-result.tpl.php
./search/search-results.tpl.php
./simpletest/tests/theme_test.template_test.tpl.php
./system/html.tpl.php
./system/maintenance-page.tpl.php
./system/page.tpl.php
./system/region.tpl.php
./taxonomy/taxonomy-term.tpl.php
./toolbar/toolbar.tpl.php
./user/user-picture.tpl.php
./user/user-profile-category.tpl.php
./user/user-profile-item.tpl.php
./user/user-profile.tpl.php
```

Too many template files



# DRUPAL 7: NEGATIVE



complex mix of subsystems

# DRUPAL 7: NEGATIVE



Drupal 7 is too hard to learn!



# DRUPAL 8

what do we want?

# DRUPAL 8

what do we want?

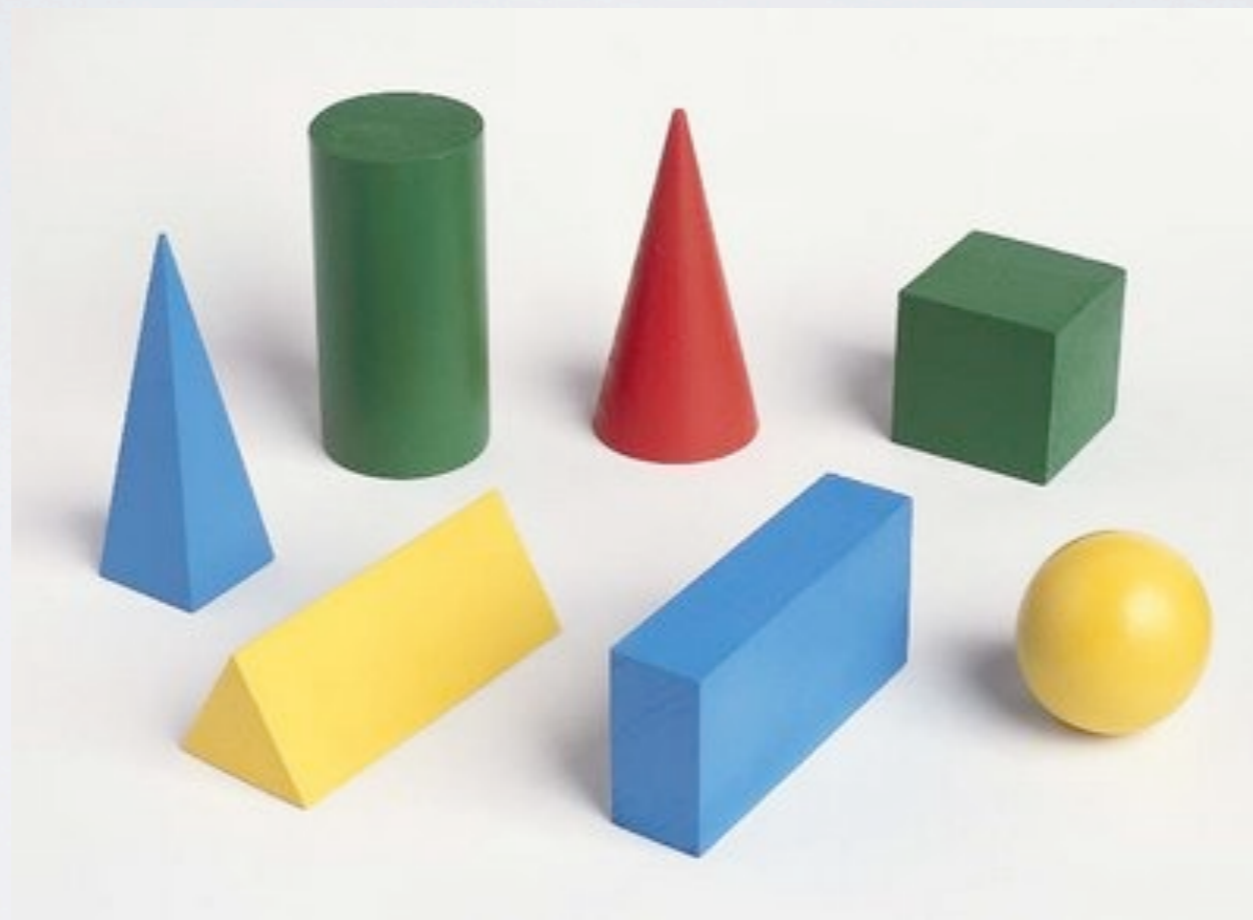


Something proudly found elsewhere



# DRUPAL 8

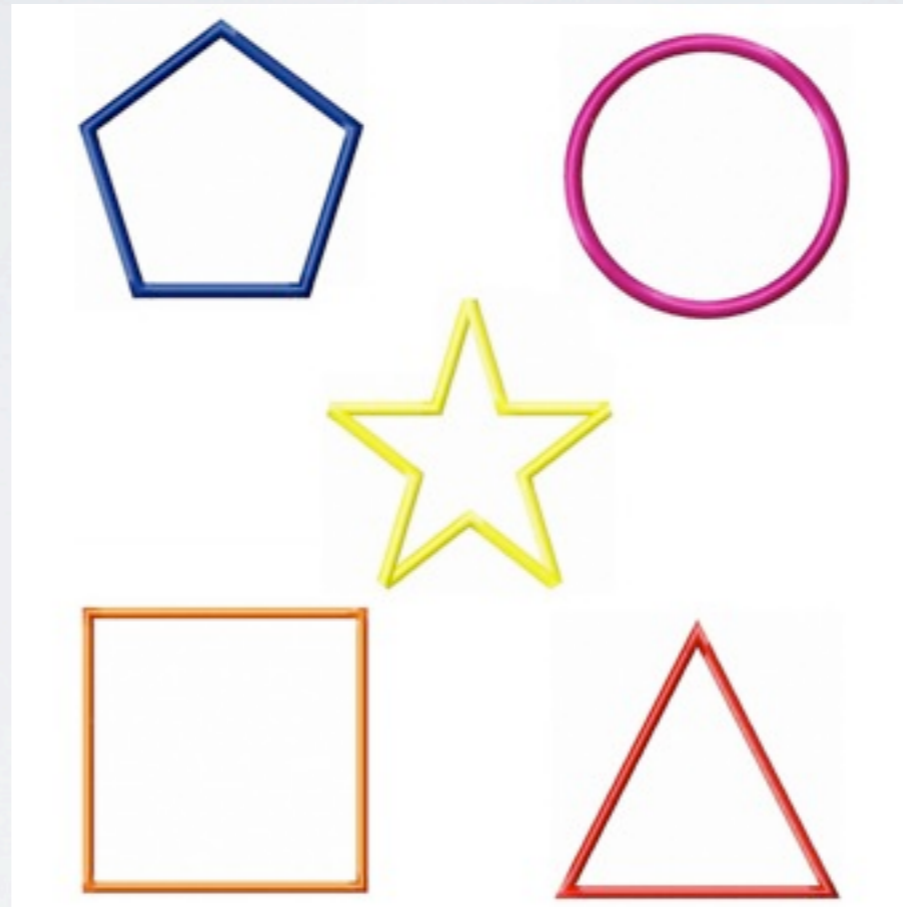
what do we want?



Objects

# DRUPAL 8

what do we want?



Objects  
(but not in the templates)



# DRUPAL 8

what do we want?



Secure

# DRUPAL 8

what do we want?

```
user.tpl.php  
node.tpl.php  
comment.tpl.php  
wrapper.tpl.php
```

The right templates



# DRUPAL 8

what do we want?



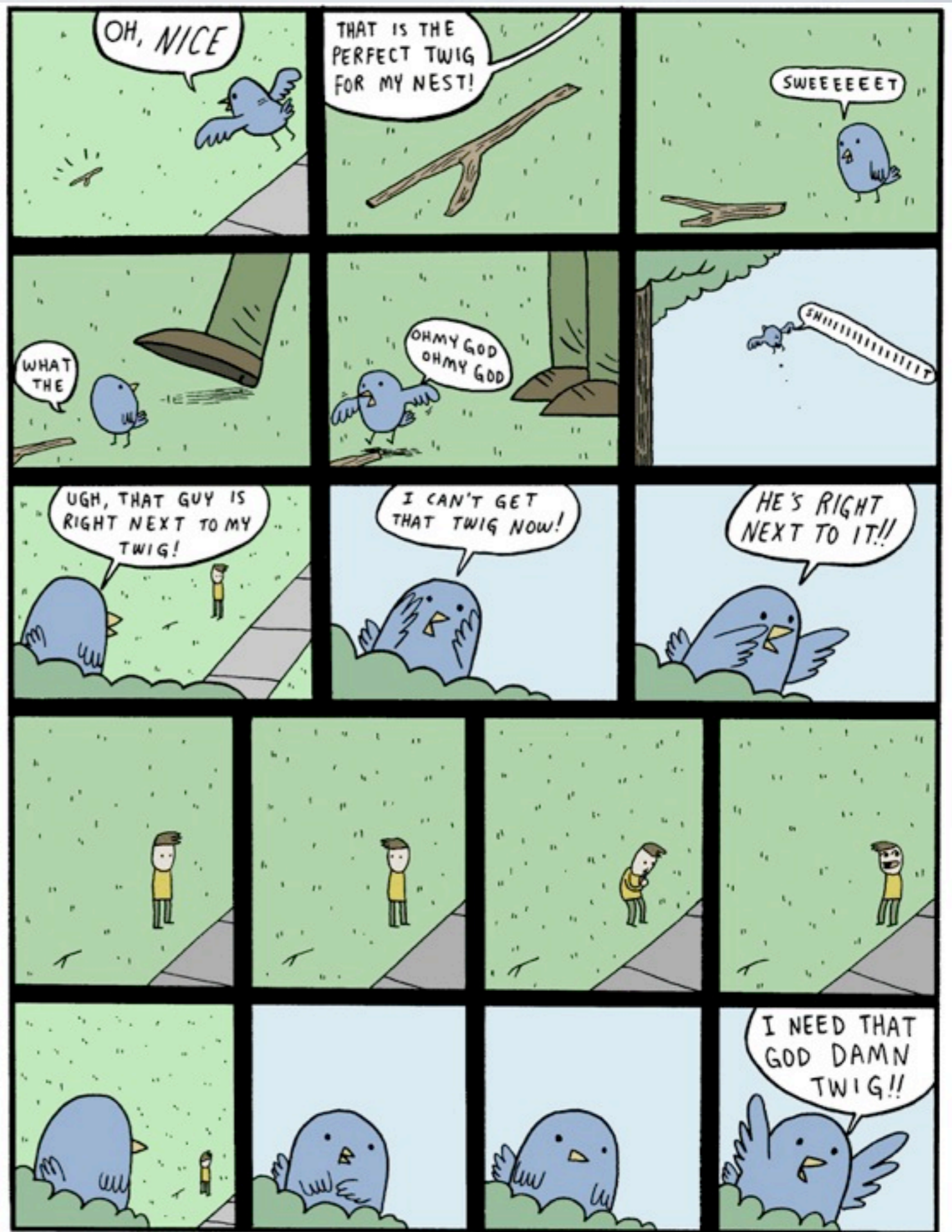
Consistency



# TWIG

A modern  
template engine  
for PHP.

<http://twig.sensiolabs.org>





# TWIG

[ABOUT](#) [DOCUMENTATION](#) [BLOG](#) [DEVELOPMENT](#) [CONTRIBUTORS](#)



## Twig Documentation

Read the online documentation to learn more about Twig.

- [Introduction / Installation](#)
- [Twig for Template Designers](#)
- [Twig for Developers](#)
- [Extending Twig](#)
- [Twig Internals](#)
- [Twig Recipes](#)
- [Coding Standards](#)
- [API](#)
- [Twig contributed extensions documentation](#)
- [License](#)

## Twig Reference

Browse the online reference to learn more about built-in features.

You can download the documentation for offline reading:



well documented

# TWIG

## Extending Twig ¶

Twig can be extended in many ways; you can add extra tags, filters, tests, operators, global variables, and functions. You can even extend the parser itself with node visitors.



The first section of this chapter describes how to extend Twig easily. If you want to reuse your changes in different projects or if you want to share them with others, you should then create an extension as described in the following section.

extensible



# TWIG

- **Secure:** When it comes to security, Twig has some unique features:

- *Automatic output escaping:* To be on the safe side, you can enable automatic output escaping globally or for a block of code:

```
{% autoescape true %}
  {% var %}
  {% var|raw %}      {# var won't be escaped #}
  {% var|escape %}  {# var won't be doubled-escaped #}
{% endautoescape %}
```

- *Sandboxing:* Twig can evaluate any template in a sandbox environment where the user has access to a limited set of tags, filters, and object methods defined by the developer. Sandboxing can be enabled globally or locally for just some templates:

```
{% include "user.html" sandboxed %}
```

secure

# TWIG

- **Unit tested:** Twig is fully unit-tested. The library is stable and ready to be used in large projects.

well-tested



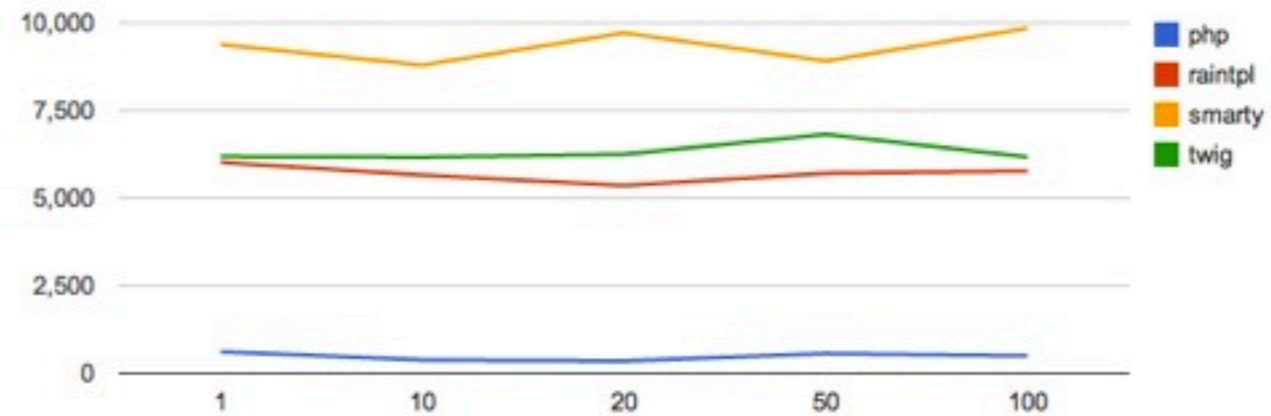
# TWIG

## Summary (assign)

Test	tot. time	tot. memory	package size
<a href="#">php</a> 5.3.3-7+squeeze9	483 $\mu$ s	11.97 KB	4 KB
<a href="#">raintpl</a> 2.7.0	5707 $\mu$ s	282.77 KB	37 KB
<a href="#">twig</a> 1.5.1	6323 $\mu$ s	715.93 KB	647 KB
<a href="#">smarty</a> 3.1.7	9336 $\mu$ s	1.28 MB	971 KB

## Execution Time (assign)

Execution Time ( $\mu$ s)



fast

# TWIG



IDE integration



# TWIG



python



**smarty**  
TEMPLATE ENGINE



**django**

recognizable syntax

# TWIG



by Symfony's author, Fabien Potencier



# TWIG

what does it look like?

# TWIG

what does it look like?

```
{% if items %}  
  <{{ type }} class="{{ items.attributes.class }}" {{ items.attributes }}>  
  {% for item in items %}  
    <li class="{{ item.attributes.class }}" {{ item.attributes }}>  
      {{- item -}}  
    </li>  
  {% endfor %}  
</{{ type }}>  
{% endif %}
```

print with {{ }}



# TWIG

what does it look like?

```
{% if items %}
<{{ type }} class="{{ items.attributes.class }}" {{ items.attributes }}>
{% for item in items %}
  <li class="{{ item.attributes.class }}" {{ item.attributes }}>
    {{- item -}}
  </li>
{% endfor %}
</{{ type }}>
{% endif %}
```

commands with {% %}

# TWIG

what does it look like?

```
{% if items %}
<{{ type }} class="{{ items.attributes.class }}" {{ items.attributes }}>
{% for item in items %}
  <li class="{{ item.attributes.class }}" {{ item.attributes }}>
    {{- item -}}
  </li>
  {% endfor %} {# end for loop #}
</{{ type }}>
{% endif %}
```

comments with {# #}



# TWIG

what does it look like?

```
{% if items %}
  <{{ type }} class="{{ items.attributes.class }}" {{ items.attributes }}>
  {% for item in items %}
    <li class="{{ item.attributes.class }}" {{ item.attributes }}>
      {{- item -}}
    </li>
  {% endfor %}
</{{ type }}>
{% endif %}
```

simple and intuitive

# INTERLUDE:

attributes

```
{% if items %}
<{{ type }} class="{{ items.attributes.class }}" {{ items.attributes }}>
{% for item in items %}
  <li class="{{ item.attributes.class }}" {{ item.attributes }}>
    {{- item -}}
  </li>
{% endfor %}
</{{ type }}>
{% endif %}
```

attributes are messy



# INTERLUDE:

attributes

HTML5 attribute types:

String

```
<div class="item-list"></div>
```

Boolean

```
<input type="checkbox" checked />
```

Enumerated

```
<div id="box" contenteditable="true" ></div>
```

attributes are messy

# INTERLUDE:

attributes

We need attributes to be “drillable”

attributes

or

attributes.class

or

attributes.id



# INTERLUDE:

attributes

PHP5 to the rescue!

ArrayAccess

allows objects to be treated as arrays

\_\_toString

allows “flattening” of arrays or objects into HTML strings  
(almost)

# INTERLUDE:

attributes

but, this has nothing to do with TWIG.

so

let's do it now:

<http://drupal.org/node/1290694>



# TWIG

how would it work?

all theme functions become template files.  
a single way to override markup!

# TWIG

D7

```
function theme_username($variables) {  
  if (isset($variables['link_path'])) {  
    $output = l($variables['name'] . $variables['extra'], $variables['link_path'], $variables['link_options']);  
  }  
  else {  
    $output = '<span' . drupal_attributes($variables['attributes_array']) . '>' . $variables['name'] . $variables['extra'] . '</span>';  
  }  
  return $output;  
}
```

D8

```
{% if link %}  
  <a href="{{ link.path }}" {{ link.attributes }}>{{ name }} {{- extra }}</a>  
{% else %}  
  <span {{ attributes }}>{{ name }} {{- extra }}</span>  
{% endif %}
```

theme\_username becomes username.twig



# TWIG

D7

```
function theme_image($variables) {  
  $attributes = $variables['attributes'];  
  $attributes['src'] = file_create_url($variables['uri']);  
  
  foreach (array('width', 'height', 'alt', 'title') as $key) {  
    if (isset($variables[$key])) {  
      $attributes[$key] = $variables[$key];  
    }  
  }  
  
  return '<img' . drupal_attributes($attributes) . ' />';  
}
```

D8

```

```

theme\_image becomes image.twig

# TWIG

D7

```
function theme_link($variables) {  
  return '<a href="' . check_plain(url($variables['path'], $variables['options'])) . "'  
  . drupal_attributes($variables['options']['attributes']) . '>  
  . ($variables['options']['html'] ? $variables['text'] : check_plain($variables['text']))  
  . '</a>';  
}
```

D8

```
<a href="{{ path }}" {{ attributes }}>{{ text }}</a>
```

theme\_link becomes link.twig



# TWIG

D7

```
function theme_item_list($variables) {  
  $items = $variables['items'];  
  $title = (string) $variables['title'];  
  $type = $variables['type'];  
  $list_attributes = $variables['attributes'];  
  
  $output = '';  
  if ($items) {  
    $output .= '<' . $type . drupal_attributes($list_attributes) . '>';  
  
    $num_items = count($items);  
    $i = 0;  
    foreach ($items as $key => $item) {  
      $i++;  
      $attributes = array();  
  
      if (is_array($item)) {  
        $value = '';  
        if (isset($item['data'])) {  
          $value .= $item['data'];  
        }  
        $attributes = array_diff_key($item, array('data' => 0, 'children' => 0));  
  
        // Append nested child list, if any.  
        if (isset($item['children'])) {  
          // HTML attributes for the outer list are defined in the 'attributes'  
          // theme variable, but not inherited by children. For nested lists,  
          // all non-numeric keys in 'children' are used as list attributes.  
          $child_list_attributes = array();  
          foreach ($item['children'] as $child_key => $child_item) {  
            if (is_string($child_key)) {  
              $child_list_attributes[$child_key] = $child_item;  
              unset($item['children'][$child_key]);  
            }  
          }  
          $value .= theme('item_list', array(  
            'items' => $item['children'],  
            'type' => $type,  
            'attributes' => $child_list_attributes,  
          ));  
        }  
        else {  
          $value = $item;  
        }  
  
        $attributes['class'][] = ($i % 2 ? 'odd' : 'even');  
        if ($i == 1) {  
          $attributes['class'][] = 'first';  
        }  
        if ($i == $num_items) {  
          $attributes['class'][] = 'last';  
        }  
  
        $output .= '<li' . drupal_attributes($attributes) . '>' . $value . '</li>';  
      }  
      $output .= "</$type>";  
    }  
  
    // Only output the list container and title, if there are any list items.  
    // Check to see whether the block title exists before adding a header.  
    // Empty headers are not semantic and present accessibility challenges.  
    if ($output !== '') {  
      if ($title !== '') {  
        $title = '<h3>' . $title . '</h3>';  
      }  
      $output = '<div class="item-list">' . $title . $output . '</div>';  
    }  
  }  
  return $output;  
}
```

D8

```
{% if items %}  
<{{ type }} class="{{ attributes.class }}" {{ attributes }}>  
  {% for item in items %}  
    <li class="{{ item.attributes.class }}" {{ item.attributes }}>  
      {{- item -}}  
    </li>  
  {% endfor %}  
</{{ type }}>  
{% endif %}
```

theme\_item\_list becomes item\_list.twig

# TWIG

how would it work?

D7

```
'comment' => array(  
  'template' => 'comment',  
  'render element' => 'elements',  
)
```

D8

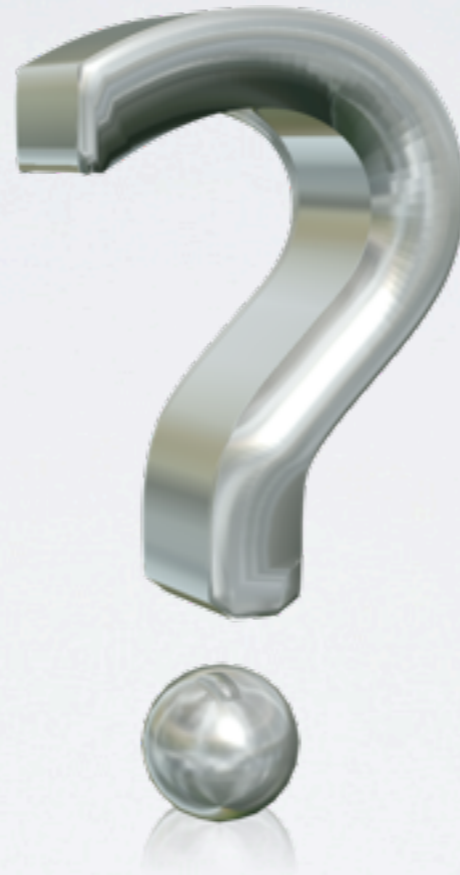
```
'comment' => array(  
  'template' => 'comment',  
  'variables' => array(  
    'title' => NULL,  
    'attributes' => array(),  
    'content' => NULL,  
  ),  
)
```

more like Drupal 6!



# TWIG

how would it work?



there's a lot we still don't know.

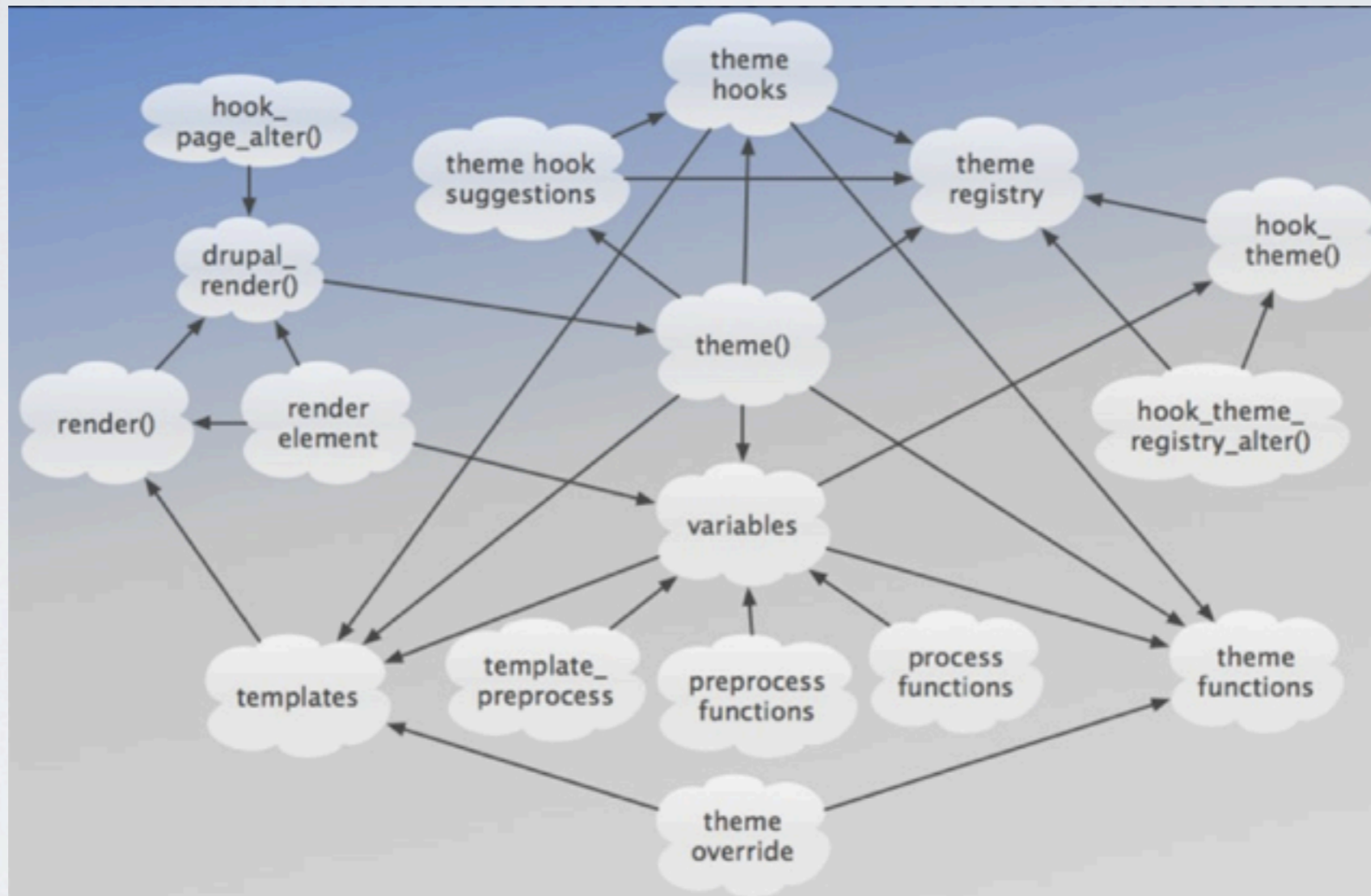
# TWIG

why is this good?



# TWIG

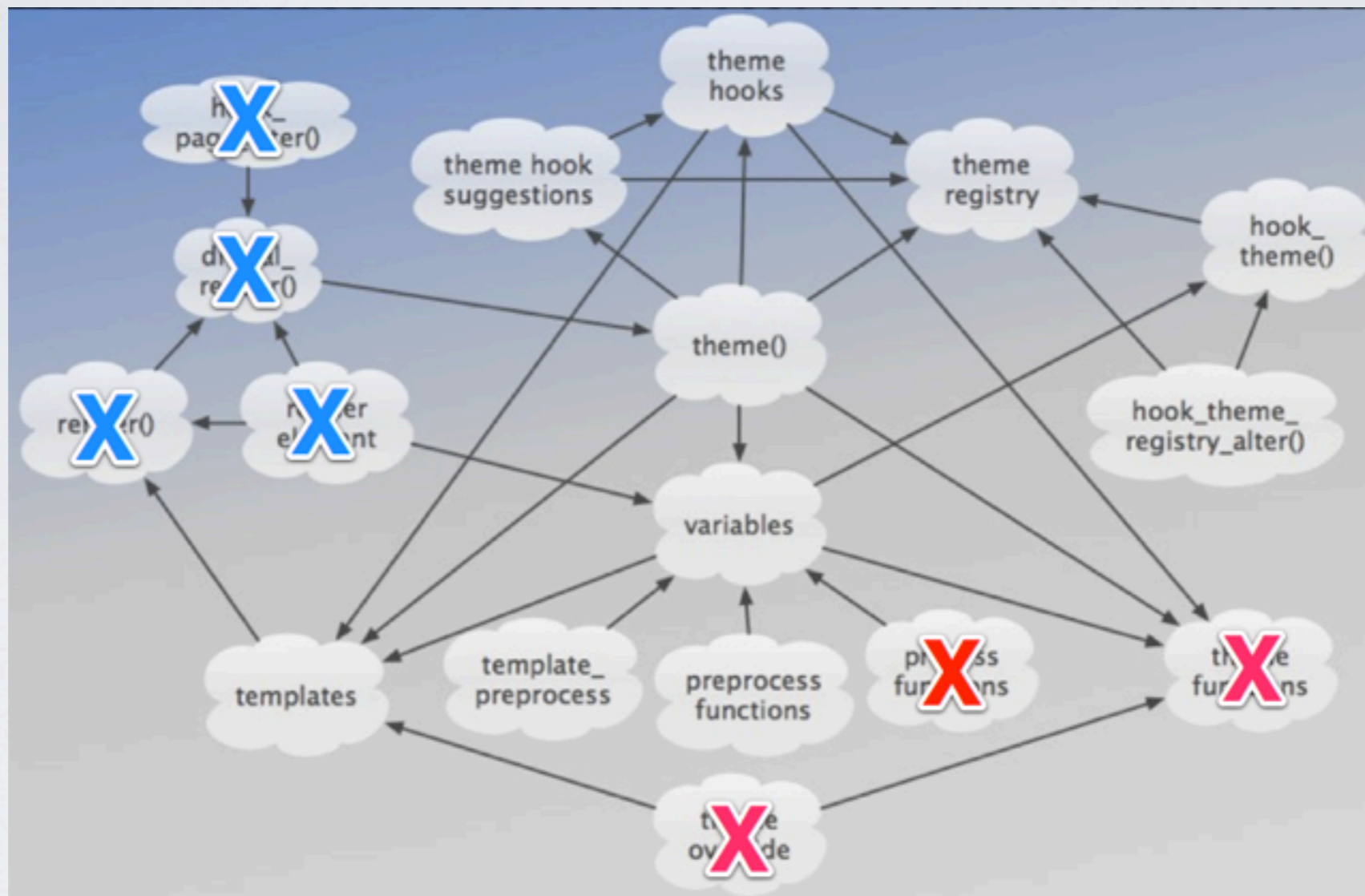
why is this good?



remember the complexity of Drupal 7?

# TWIG

why is this good?



look what would happen in Drupal 8.



# TWIG

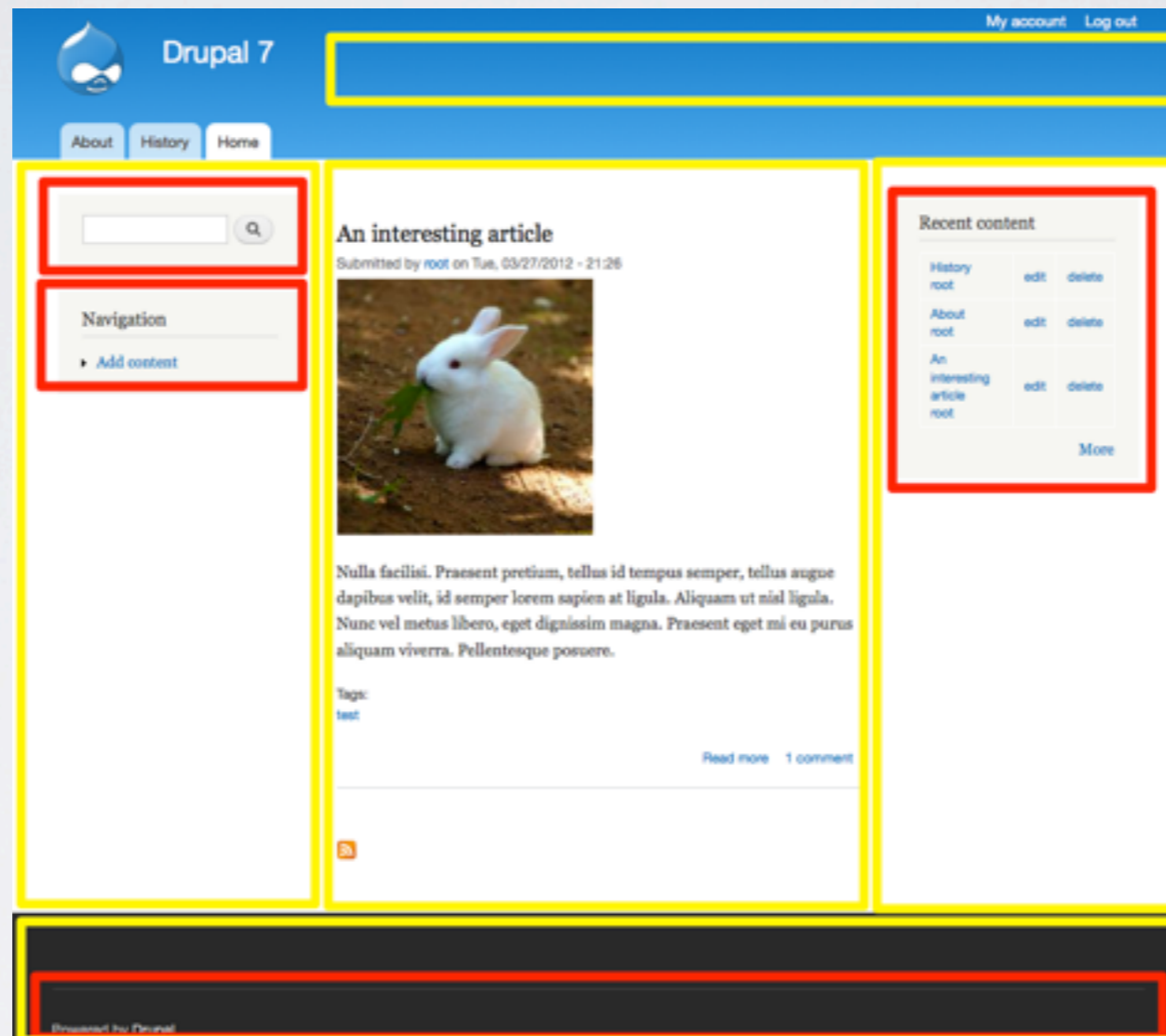
where does this fit in with SCOTCH?



The “Blocks & Layouts” initiative complicates things

# TWIG

where does this fit in with SCOTCH?

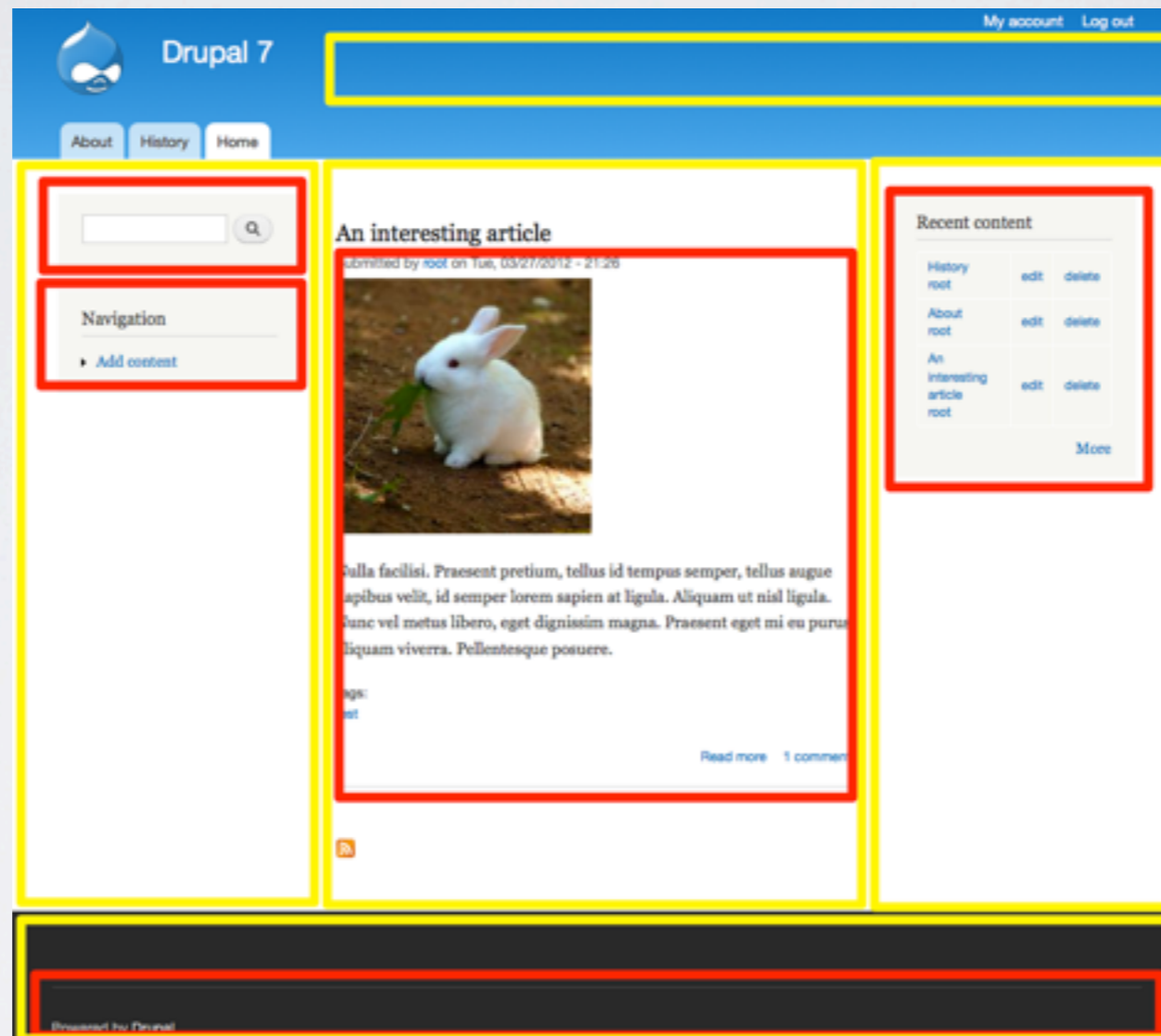


Blocks in D6 (in red)



# TWIG

where does this fit in with SCOTCH?



Blocks in D7 (in red)

# TWIG

where does this fit in with SCOTCH?



Blocks in D8 (in red)



# TWIG

where does this fit in with SCOTCH?

## **Two kinds of templates:**

- Page or region “templates” as plugins
  - selectable via the user interface
- Block or element “templates”
  - provided by or overridden in your theme

# TWIG

while we're at it...

PHPTemplate	Twig template	Status
page.tpl.php	-	layouts provided by Sctotch
region.tpl.php	-	eliminated in favor of wrapper.twig
block.tpl.php	block.twig	same as wrapper.twig, but uses 'aside' HTML5 tag
node.tpl.php	node.twig	same as wrapper.twig, but uses 'article' HTML5 tag
comment.tpl.php	comment.twig	same as node.twig, here for ease of learning?
comment- wrapper.tpl.php	-	eliminated in favor of wrapper.twig
field.tpl.php	-	eliminated in favor of item_list.twig
forums.tpl.php	-	eliminated in favor of wrapper.twig
search-results.tpl.php	-	eliminated in favor of wrapper.twig, and item_list.twig
search-result.tpl.php	-	eliminated in favor of wrapper.twig

...let's clean up the template files!



# TWIG

improve markup

- Use HTML5!
- Create markup patterns
- Consistent use of tags & classes

# TWIG

reduce & reuse templates

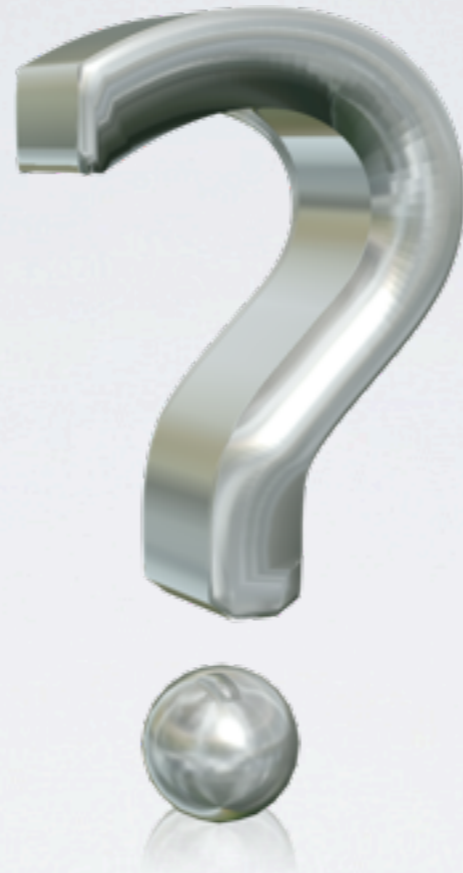
- Consolidate similar templates
- Fewer templates & wrappers
- More theme\_hook\_suggestions



# TWIG

Focus on “Learnability”

- Easier to learn than PHP
- Consistent
- Secure



QUESTIONS?



# A NEW THEME LAYER FOR DRUPAL 8

@jenlampton | <http://www.jenlampton.com>

# photo credits:

lolcat-flexible

<http://cheezburger.com/2679924736>

anything is possible pebbles

<http://www.invergordontours.com/aip.html>

lolcat questionmark

<http://icanhascheezburger.com/2007/10/31/11197/>

wheel-reinvented

<http://www.brainwads.net/drewhawkins/2012/01/dont-re-invent-the-wheel-make-something-better/>

objects

[http://2teachers4classroom.blogspot.com/2009\\_02\\_01\\_archive.html](http://2teachers4classroom.blogspot.com/2009_02_01_archive.html)

shapes

[http://englishclass.jp/reading/topic/For\\_Screening\\_Purposes\\_Only](http://englishclass.jp/reading/topic/For_Screening_Purposes_Only)

secure

<http://blog.stratopedia.org/2010/06/03/what-is-a-secure-site/>

consistency

<http://icsigns.org/press/2010/03/23/consistency-staying-on-the-mark/>

twig bird comic

<http://s302.photobucket.com/albums/nn105/walkseva/?action=view&current=thebirdneedsthattwig.gif&currenttag=bird%20park%20twig%20comic%20need%20it>

twig docs screenshots

<http://twig.sensiolabs.org/documentation>

twig speed graphs

<http://phpcomparison.net/>

python icon

<http://python-hosting.org/>

ruby icon

<http://itmediaconnect.ro/en/web>

django logo

<http://py-arahat.blogspot.com/2010/08/django-vs-pylons.html>

symfony logo

<http://symfony.com/logo>

scotch glass

<http://www.thespir.it/articles/scotch-101/?viewall=1>