Aha! Understanding and Using Render Arrays in Drupal 8

CHROMATIC

Gus Childs

@guschilds

CHROMATIC

chromatichq.com
@chromatichq

```
// various regions in this

@include jacket(critical) {
  .layout-header__inner,
  .layout-main,
  .layout-footer__inner {
    max-width: $page-max-width;
    padding: 0 $gutter-half;
    margin: 0 auto;

    @include breakpoint($breakpoint-medium) {
      padding: 0 $gutter-full;
    }
  }

  .layout-footer__inner {
    padding: $gutter-full;

    @include breakpoint($breakpoint-large) {
      padding: 2 * $gutter-full;
    }
  }

  .page {
    position: relative;
  }

  .page--sidebar {
    @include breakpoint($breakpoint-medium) {
      .layout-content {
        margin-right: -$sidebar-width;
```

```
      'pattern' => '/v([0-9\.]+)/',
      'lines' => 1,
    ),
    'files' => array(
      'js' => array('lazysizes.min.js'),
    ),
  );
  return $libraries;
}

/**
 * Overrides image theming to add lazyloading to all images.
 *
 * @see theme_image()
 */
function xyz_lazyload_image($variables) {
  if (path_is_admin(current_path()) || !$variables['lazyload']) {
    return theme_image($variables);
  }
  $attributes = $variables['attributes'];
  // Lazyloading is added with the data-src attribute and a lazyload class. If
  // responsive images are involved, the srcset attribute is moved to
  // data-srcset in the same fashion. The lazysizes JS will do the rest.
  $attributes['class'][] = XYZ_LAZYLOAD_CLASS;
  $attributes['data-src'] = file_create_url($variables['path']);
  if (isset($attributes['srcset'])) {
    $attributes['data-srcset'] = $attributes['srcset'];
    unset($attributes['srcset']);
  }
  (array('width', 'height', 'alt', 'title') as $key) {
```

let’s talk about render arrays

CHROMATIC

but why render arrays?

problem:

build feature X

like the last site

but

slightly different

solution:
copy/paste?

CHROMATIC

Don’t Repeat Yourself!

CHROMATIC

solution:

abstraction

render arrays are abstract

(picturesbyann, Flickr)

CHROMATIC

(davidchief, Flickr)

“arrays in Drupal 8?!” –you

CHROMATIC

“yeah, sorry.” -maintainers

CHROMATIC

Aha! Understanding and Using Render Arrays in Drupal 8

CHROMATIC

Render Arrays:

what they look like

how they work

why they're useful

the Render API

their future

CHROMATIC

what do render arrays look like?

(Pulp Fiction)

theme()
render arrays

```
// Using theme() in D7.
$logo = theme('image', array(
  'path' => 'logo.png',
  'alt' => t('My logo!'),
));
```

```
// Using theme() in D7.
$logo = theme('image', array(
  'path' => 'logo.png',
  'alt' => t('My logo!'),
));
```

```
// Using theme() in D7.
$logo = theme('image', array(
  'path' => 'logo.png',
  'alt' => t('My logo!'),
));
```

```
// The equivalent render array in D7.
$logo = array(
  '#theme' => 'image',
  '#path' => 'logo.png',
  '#alt' => t('My logo!'),
);
```

```
// The equivalent render array in D7.
$logo = array(
  '#theme' => 'image',
  '#path' => 'logo.png',
  '#alt' => t('My logo!'),
);
```

```
// The equivalent render array in D7.
$logo = array(
  '#theme' => 'image',
  '#path' => 'logo.png',
  '#alt' => t('My logo!'),
);
```

```
// The equivalent render array in D7.
$logo = array(
  '#theme' => 'image',
  '#path' => 'logo.png',
  '#alt' => t('My logo!'),
);
```

```
// Using theme() in D7.
$logo = theme('image', array(
  'path' => 'logo.png',
  'alt' => t('My logo!'),
));
```

```php
// Printing a string variable in D7.
<?php print $logo; ?>
```

```
// The resulting markup.
<img src="logo.png" alt="My logo!" />
```

```
// The equivalent render array in D7.
$logo = array(
  '#theme' => 'image',
  '#path' => 'logo.png',
  '#alt' => t('My logo!'),
);
```

```php
// How to NOT print a render array in D7.
<?php print $logo; ?>
```

```
// The result. Huh?
Array
```

```php
// Rendering/printing a render array in D7.
<?php print render($logo); ?>
```

```
// The resulting markup. That’s better.
<img src="logo.png" alt="My logo!" />
```

```
// Printing/rendering anything in D8.
{{ logo }}
```

```
// Printing/rendering a render array in D7.
<?php print render($logo); ?>

// Printing/rendering a render array in D8.
{{ logo }}
```

CHROMATIC

```
// A simplified page array.
array(
  'header' => array(...),
  'content' => array(
    'node' => array(
      '#theme' => 'node',
      '#node' => Object,
      '#view_mode' => 'full',
    ),
  ),
  'sidebar_first' => array(...),
  'footer' => array(...),
);
```

drupal7/modules/system/page.tpl.php

```
// Rendering the content region in D7.
<?php print render($page['content']); ?>
```

drupal8/core/themes/stable/templates/layout/page.html.twig

```
// Rendering the content region in D8.
{{ page.content }}
```

```
// A simplified page array.
array(
  'header' => array(...),
  'content' => array(
    'node' => array(
      '#theme' => 'node',
      '#node' => Object,
      '#view_mode' => 'full',
    ),
  ),
  'sidebar_first' => array(...),
  'footer' => array(...),
);
```

```
// A simplified page array.
array(
  'header' => array(...),
  'content' => array(
    'node' => array(
      '#theme' => 'node',
      '#node' => Object,
      '#view_mode' => 'full',
    ),
  ),
  'sidebar_first' => array(...),
  'footer' => array(...),
);
```

how do render arrays work?

(Pete Birkinshaw, Flickr)

```
// Printing/rendering a render array in D7.
<?php print render($logo); ?>

// Printing/rendering a render array in D8.
{{ logo }}
```

```
// Our simple render array in D7.
$logo = array(
  '#theme' => 'image',
  '#path' => 'logo.png',
  '#alt' => t('My logo!'),
);
```

```
// The resulting markup.
<img src="logo.png" alt="My logo!" />
```

Theme System

(Erin Khoo, Flickr)

drupal7/modules/system/system.module
drupal7/includes/theme.inc
(simplified)

```php
/**
 * Implements hook_theme().
 */
function system_theme() {
  return array(
    'html' => array(
      'render element' => 'html',
      'template' => 'html',
    ),
    'page' => array(
      'render element' => 'page',
      'template' => 'page',
    ),
    // (Other theme hook definitions).
  );
}
```

drupal8/core/modules/system/system.module
drupal8/core/includes/theme.inc
(simplified)

```
/**
 * Implements hook_theme().
 */
function system_theme() {
  return array(
    'html' => array(
      'render element' => 'html',
    ),
    'page' => array(
      'render element' => 'page',
    ),
    // (Other theme hook definitions).
  );
}
```

drupal7/modules/system/system.module
drupal7/includes/theme.inc
(simplified)

```php
/**
 * Implements hook_theme().
 */
function system_theme() {
  return array(
    'image' => array(
      'variables' => array(
        'path' => NULL,
        'width' => NULL,
        'height' => NULL,
        'alt' => '',
        'title' => NULL,
        'attributes' => array(),
      ),
    ),
  );
}
```

CHROMATIC

drupal7/modules/system/system.module
drupal7/includes/theme.inc
(simplified)

```
/**
 * Implements hook_theme().
 */
function system_theme() {
  return array(
    'image' => array(
      'variables' => array(
        'path' => NULL,
        'width' => NULL,
        'height' => NULL,
        'alt' => '',
        'title' => NULL,
        'attributes' => array(),
      ),
    ),
  );
}
```

drupal7/modules/system/system.module
drupal7/includes/theme.inc
(simplified)

```
/**
 * Implements hook_theme().
 */
function system_theme() {
  return array(
    'image' => array(
      'variables' => array(
        'path' => NULL,
        'width' => NULL,
        'height' => NULL,
        'alt' => '',
        'title' => NULL,
        'attributes' => array(),
      ),
    ),
  );
}
```

```
// Our simple render array in D7.
$logo = array(
  '#theme' => 'image',
  '#path' => 'logo.png',
  '#alt' => t('My logo!'),
);
```

```
// Our simple render array in D7.
$logo = array(
  '#theme' => 'image',
  '#path' => 'logo.png',
  '#alt' => t('My logo!'),
);
```

```
// Our simple render array in D7.
$logo = array(
  '#theme' => 'image',
  '#path' => 'logo.png',
  '#alt' => t('My logo!'),
);
```

drupal7/modules/system/system.module
drupal7/includes/theme.inc
(simplified)

```php
/**
 * Implements hook_theme().
 */
function system_theme() {
  return array(
    'image' => array(
      'variables' => array(
        'path' => NULL,
        'width' => NULL,
        'height' => NULL,
        'alt' => '',
        'title' => NULL,
        'attributes' => array(),
      ),
    ),
  );
}
```

drupal7/modules/system/system.module
drupal7/includes/theme.inc
(simplified)

```
/**
 * Implements hook_theme().
 */
function system_theme() {
  return array(
    'image' => array(
      'variables' => array(
        'path' => NULL,
        'width' => NULL,
        'height' => NULL,
        'alt' => '',
        'title' => NULL,
        'attributes' => array(),
      ),
    ),
  );
}
```

CHROMATIC

drupal8/core/modules/system/system.module
drupal8/core/includes/theme.inc
(simplified)

```
/**
 * Implements hook_theme().
 */
function system_theme() {
  return array(
    'image' => array(
      'variables' => array(
        'uri' => NULL,
        'width' => NULL,
        'height' => NULL,
        'alt' => '',
        'title' => NULL,
        'attributes' => array(),
        'sizes' => NULL,
        'srcset' => array(),
        'style_name' => NULL,
      ),
    ),
  );
}
```

CHROMATIC

drupal8/core/modules/system/system.module
drupal8/core/includes/theme.inc
(simplified)

```
/**
 * Implements hook_theme().
 */
function system_theme() {
  return array(
    'image' => array(
      'variables' => array(
        'uri' => NULL,
        'width' => NULL,
        'height' => NULL,
        'alt' => '',
        'title' => NULL,
        'attributes' => array(),
        'sizes' => NULL,
        'srcset' => array(),
        'style_name' => NULL,
      ),
    ),
  );
}
```

drupal8/core/modules/system/system.module
drupal8/core/includes/theme.inc
(simplified)

```
/**
 * Implements hook_theme().
 */
function system_theme() {
  return array(
    'image' => array(
      'variables' => array(
        'uri' => NULL,
        'width' => NULL,
        'height' => NULL,
        'alt' => '',
        'title' => NULL,
        'attributes' => array(),
        'sizes' => NULL,
        'srcset' => array(),
        'style_name' => NULL,
      ),
    ),
  );
}
```

CHROMATIC

```
// Our simple render array in D7.
$logo = array(
  '#theme' => 'image',
  '#path' => 'logo.png',
  '#alt' => t('My logo!'),
);
```

```
// Our simple render array in D8.
$logo = array(
  '#theme' => 'image',
  '#uri' => 'public://logo.png',
  '#alt' => t('My logo!'),
);
```

```php
// A responsive image render array in D8.
$logo_uri = 'public://logo.png';
$logo = array(
  '#theme' => 'image',
  '#uri' => $logo_uri,
  '#alt' => t('My logo!'),
  '#sizes' => '100vw',
  '#srcset' => array(
    array(
      'uri' => ImageStyle::load('thumbnail')->buildUri($logo_uri),
      'width' => '250w',
    ),
    array(
      'uri' => ImageStyle::load('large')->buildUri($logo_uri),
      'width' => '750w',
    ),
  ),
);
```

```php
// A responsive image render array in D8.
$logo_uri = 'public://logo.png';
$logo = array(
  '#theme' => 'image',
  '#uri' => $logo_uri,
  '#alt' => t('My logo!'),
  '#sizes' => '100vw',
  '#srcset' => array(
    array(
      'uri' => ImageStyle::load('thumbnail')->buildUri($logo_uri),
      'width' => '250w',
    ),
    array(
      'uri' => ImageStyle::load('large')->buildUri($logo_uri),
      'width' => '750w',
    ),
  ),
);
```

```
// A responsive image render array in D8.
$logo_uri = 'public://logo.png';
$logo = array(
  '#theme' => 'image',
  '#uri' => $logo_uri,
  '#alt' => t('My logo!'),
  '#sizes' => '100vw',
  '#srcset' => array(
    array(
      'uri' => ImageStyle::load('thumbnail')->buildUri($logo_uri),
      'width' => '250w',
    ),
    array(
      'uri' => ImageStyle::load('large')->buildUri($logo_uri),
      'width' => '750w',
    ),
  ),
);
```

CHROMATIC

```php
// A responsive image render array in D8.
$logo_uri = 'public://logo.png';
$logo = array(
  '#theme' => 'image',
  '#uri' => $logo_uri,
  '#alt' => t('My logo!'),
  '#sizes' => '100vw',
  '#srcset' => array(
    array(
      'uri' => ImageStyle::load('thumbnail')->buildUri($logo_uri),
      'width' => '250w',
    ),
    array(
      'uri' => ImageStyle::load('large')->buildUri($logo_uri),
      'width' => '750w',
    ),
  ),
);
```

CHROMATIC

```
// The resulting responsive image markup.
<img src="logo.png" alt="My logo!" sizes="100vw"
  srcset="styles/thumbnail/public/logo.png 250w,
    styles/large/public/logo.png 750w">
```

```php
// A responsive image render array in D8.
$logo_uri = 'public://logo.png';
$logo = array(
  '#theme' => 'image',
  '#uri' => $logo_uri,
  '#alt' => t('My logo!'),
  '#sizes' => '100vw',
  '#srcset' => array(
    array(
      'uri' => ImageStyle::load('thumbnail')->buildUri($logo_uri),
      'width' => '250w',
    ),
    array(
      'uri' => ImageStyle::load('large')->buildUri($logo_uri),
      'width' => '750w',
    ),
  ),
);
```

```
// The resulting responsive image markup.
<img src="logo.png" alt="My logo!" sizes="100vw"
  srcset="styles/thumbnail/public/logo.png 250w,
    styles/large/public/logo.png 750w">
```

drupal7/modules/system/system.module
drupal7/includes/theme.inc
(simplified)

```
/**
 * Implements hook_theme().
 */
function system_theme() {
  return array(
    'html' => array(
      'render element' => 'html',
      'template' => 'html',
    ),
    'page' => array(
      'render element' => 'page',
      'template' => 'page',
    ),
    'image' => array(
      'variables' => array(
        // Bunches 'o variables.
      ),
    ),
  );
}
```

CHROMATIC

drupal7/modules/system/system.module
drupal7/includes/theme.inc
(simplified)

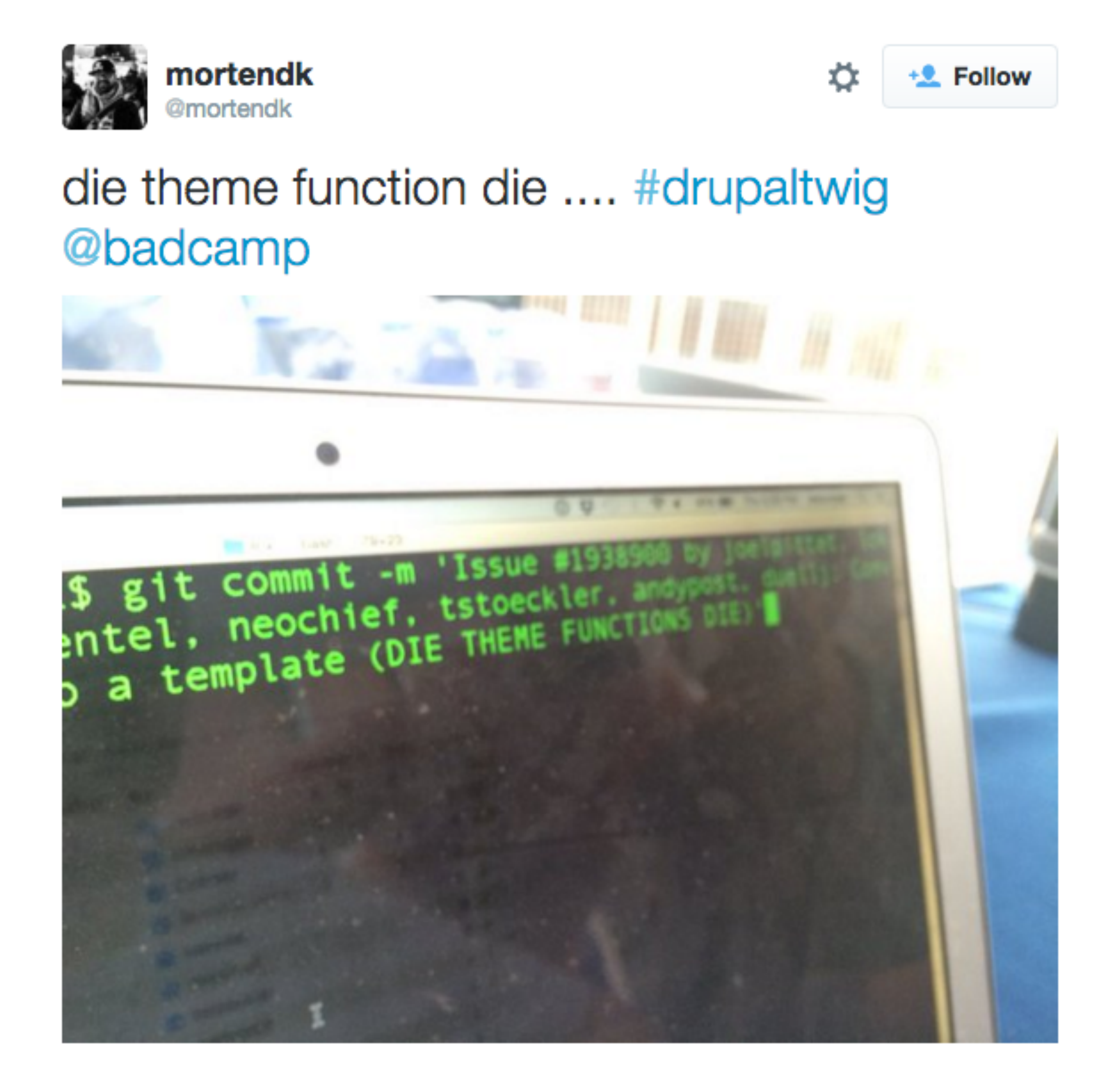
```
/**
 * Implements hook_theme().
 */
function system_theme() {
  return array(
    'html' => array(
      'render element' => 'html',
      'template' => 'html',
    ),
    'page' => array(
      'render element' => 'page',
      'template' => 'page',
    ),
    'image' => array(
      'variables' => array(
        // Bunches 'o variables.
      ),
    ),
  );
}
```

drupal7/modules/system/system.module
drupal7/includes/theme.inc
(simplified)

```php
/**
 * Implements hook_theme().
 */
function system_theme() {
  return array(
    'html' => array(
      'render element' => 'html',
      'template' => 'html',
    ),
    'page' => array(
      'render element' => 'page',
      'template' => 'page',
    ),
    'image' => array(
      'variables' => array(
        // Bunches 'o variables.
      ),
    ),
  );
}
```

drupal7/includes/theme.inc

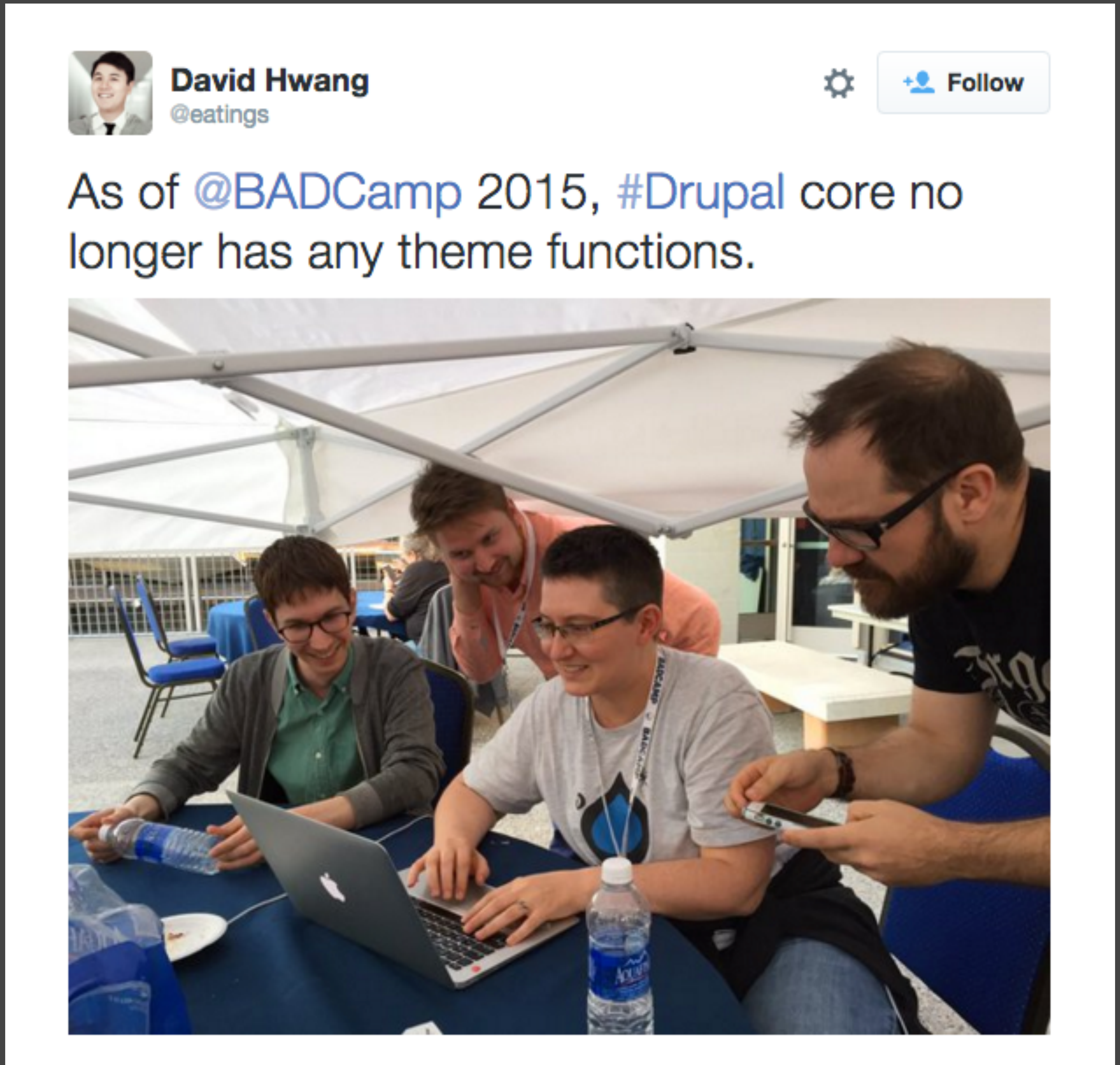
```
/**
 * Returns HTML for an image.
 */
function theme_image($variables) {
  $attributes = $variables['attributes'];
  $attributes['src'] = file_create_url($variables['path']);

  foreach (array('width', 'height', 'alt', 'title') as $key) {

    if (isset($variables[$key])) {
      $attributes[$key] = $variables[$key];
    }
  }

  return '<img' . drupal_attributes($attributes) . ' />';
}
```



mortendk
@mortendk

die theme function die .... #drupaltwig @badcamp

```
$ git commit -m 'Issue #1938900 by joelpittet, [illegible]
entel, neochief, tstoeckler, andypost, duellj: Con
o a template (DIE THEME FUNCTIONS DIE)'
```



David Hwang
@eatings

As of @BADCamp 2015, #Drupal core no longer has any theme functions.

CHROMATIC

drupal7/includes/theme.inc

```php
function theme_item_list($variables) {
  $items = $variables['items'];
  $title = $variables['title'];
  $type = $variables['type'];
  $attributes = $variables['attributes'];

  // 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.
  $output = '<div class="item-list">';
  if (isset($title) && $title !== '') {
    $output .= '<h3>' . $title . '</h3>';
  }

  if (!empty($items)) {
    $output .= "<$type" . drupal_attributes($attributes) . '>';
    $num_items = count($items);
    $i = 0;
    foreach ($items as $item) {
      $attributes = array();
      $children = array();
      $data = '';
      $i++;
      if (is_array($item)) {
        foreach ($item as $key => $value) {
          if ($key == 'data') {
            $data = $value;
          }
          elseif ($key == 'children') {
            $children = $value;
          }
          else {
            $attributes[$key] = $value;
          }
        }
      }
      else {
        $data = $item;
      }
      if (count($children) > 0) {
        // Render nested list.
        $data .= theme_item_list(array('items' => $children, 'title' => NULL, 'type' => $type, 'attributes' => $attributes));
      }
      if ($i == 1) {
        $attributes['class'][] = 'first';
      }
      if ($i == $num_items) {
        $attributes['class'][] = 'last';
      }
      $output .= '<li' . drupal_attributes($attributes) . '>' . $data . "</li>\n";
    }
    $output .= "</$type>";
  }
  $output .= '</div>';
  return $output;
}
```

Don’t Repeat Yourself!

drupal8/core/modules/system/system.module
drupal8/core/includes/theme.inc
(simplified)

```
/**
 * Implements hook_theme().
 */
function system_theme() {
  return array(
    'html' => array(
      'render element' => 'html',
    ),
    'page' => array(
      'render element' => 'page',
    ),
  );
}
```

drupal7/includes/theme.inc

```php
/**
 * Returns HTML for an image.
 */
function theme_image($variables) {
  $attributes = $variables['attributes'];
  $attributes['src'] = file_create_url($variables['path']);

  foreach (array('width', 'height', 'alt', 'title') as $key) {

    if (isset($variables[$key])) {
      $attributes[$key] = $variables[$key];
    }
  }

  return '<img' . drupal_attributes($attributes) . ' />';
}
```

drupal8/core/themes/stable/templates/field/image.html.twig

```
<img{{ attributes }} />
```

drupal7/includes/theme.inc

```
/**
 * Returns HTML for an image.
 */
function theme_image($variables) {
  $attributes = $variables['attributes'];
  $attributes['src'] = file_create_url($variables['path']);

  foreach (array('width', 'height', 'alt', 'title') as $key) {

    if (isset($variables[$key])) {
      $attributes[$key] = $variables[$key];
    }
  }

  return '<img' . drupal_attributes($attributes) . ' />';
}
```

```php
// A responsive image render array in D8.
$logo_uri = 'public://logo.png';
$logo = array(
  '#theme' => 'image',
  '#uri' => $logo_uri,
  '#alt' => t('My logo!'),
  '#sizes' => '100vw',
  '#srcset' => array(
    array(
      'uri' => ImageStyle::load('thumbnail')->buildUri($logo_uri),
      'width' => '250w',
    ),
    array(
      'uri' => ImageStyle::load('large')->buildUri($logo_uri),
      'width' => '750w',
    ),
  ),
);
```

drupal8/core/includes/theme.inc

```php
/**
 * Prepares variables for image templates.
 */
function template_preprocess_image(&$variables) {
  if (!empty($variables['uri'])) {
    $variables['attributes']['src'] = file_url_transform_relative(file_create_url($variables['uri']));
  }

  // There is normally a lot of code here that takes the responsive variables
  // and turns them into properly formatted attribute values.

  foreach (array('width', 'height', 'alt', 'title', 'sizes') as $key) {
    if (isset($variables[$key])) {
      // If the property has already been defined in the attributes,
      // do not override, including NULL.
      if (array_key_exists($key, $variables['attributes'])) {
        continue;
      }
      $variables['attributes'][$key] = $variables[$key];
    }
  }
}
```

CHROMATIC

drupal8/core/includes/theme.inc

```php
/**
 * Prepares variables for image templates.
 */
function template_preprocess_image(&$variables) {
  if (!empty($variables['uri'])) {
    $variables['attributes']['src'] = file_url_transform_relative(file_create_url($variables['uri']));
  }

  // There is normally a lot of code here that takes the responsive variables
  // and turns them into properly formatted attribute values.

  foreach (array('width', 'height', 'alt', 'title', 'sizes') as $key) {
    if (isset($variables[$key])) {
      // If the property has already been defined in the attributes,
      // do not override, including NULL.
      if (array_key_exists($key, $variables['attributes'])) {
        continue;
      }
      $variables['attributes'][$key] = $variables[$key];
    }
  }
}
```

drupal8/core/includes/theme.inc

```php
/**
 * Prepares variables for image templates.
 */
function template_preprocess_image(&$variables) {
  if (!empty($variables['uri'])) {
    $variables['attributes']['src'] = file_url_transform_relative(file_create_url($variables['uri']));
  }

  // There is normally a lot of code here that takes the responsive variables
  // and turns them into properly formatted attribute values.

  foreach (array('width', 'height', 'alt', 'title', 'sizes') as $key) {
    if (isset($variables[$key])) {
      // If the property has already been defined in the attributes,
      // do not override, including NULL.
      if (array_key_exists($key, $variables['attributes'])) {
        continue;
      }
      $variables['attributes'][$key] = $variables[$key];
    }
  }
}
```

drupal8/core/includes/theme.inc

```php
/**
 * Prepares variables for image templates.
 */
function template_preprocess_image(&$variables) {
  if (!empty($variables['uri'])) {
    $variables['attributes']['src'] = file_url_transform_relative(file_create_url($variables['uri']));
  }

  // There is normally a lot of code here that takes the responsive variables
  // and turns them into properly formatted attribute values.

  foreach (array('width', 'height', 'alt', 'title', 'sizes') as $key) {
    if (isset($variables[$key])) {
      // If the property has already been defined in the attributes,
      // do not override, including NULL.
      if (array_key_exists($key, $variables['attributes'])) {
        continue;
      }
      $variables['attributes'][$key] = $variables[$key];
    }
  }
}
```

drupal8/core/themes/stable/templates/field/image.html.twig

```
<img{{ attributes }} />
```

drupal8/core/themes/classy/templates/content/node.html.twig

```
{%
  set classes = [
    'node',
    'node--type-' ~ node.bundle|clean_class,
    node.isPromoted() ? 'node--promoted',
    node.isSticky() ? 'node--sticky',
    not node.isPublished() ? 'node--unpublished',
    view_mode ? 'node--view-mode-' ~ view_mode|clean_class,
  ]
%}
{{ attach_library('classy/node') }}
<article{{ attributes.addClass(classes) }}>
  // Node markup.
</article>
```

Theme System

(Erin Khoo, Flickr)

drupal8/core/modules/system/system.module
drupal8/core/includes/theme.inc
(simplified)

```php
/**
 * Implements hook_theme().
 */
function system_theme() {
  return array(
    'image' => array(
      'variables' => array(
        'uri' => NULL,
        'width' => NULL,
        'height' => NULL,
        'alt' => '',
        'title' => NULL,
        'attributes' => array(),
        'sizes' => NULL,
        'srcset' => array(),
        'style_name' => NULL,
      ),
    ),
  );
}
```

CHROMATIC

```php
// A responsive image render array in D8.
$logo_uri = 'public://logo.png';
$logo = array(
  '#theme' => 'image',
  '#uri' => $logo_uri,
  '#alt' => t('My logo!'),
  '#sizes' => '100vw',
  '#srcset' => array(
    array(
      'uri' => ImageStyle::load('thumbnail')->buildUri($logo_uri),
      'width' => '250w',
    ),
    array(
      'uri' => ImageStyle::load('large')->buildUri($logo_uri),
      'width' => '750w',
    ),
  ),
);
```

```
// Printing/rendering our logo in D8.
{{ logo }}
```

drupal8/core/includes/theme.inc

```php
/**
 * Prepares variables for image templates.
 */
function template_preprocess_image(&$variables) {
  if (!empty($variables['uri'])) {
    $variables['attributes']['src'] = file_url_transform_relative(file_create_url($variables['uri']));
  }

  // There is normally a lot of code here that takes the responsive variables
  // and turns them into properly formatted attribute values.

  foreach (array('width', 'height', 'alt', 'title', 'sizes') as $key) {
    if (isset($variables[$key])) {
      // If the property has already been defined in the attributes,
      // do not override, including NULL.
      if (array_key_exists($key, $variables['attributes'])) {
        continue;
      }
      $variables['attributes'][$key] = $variables[$key];
    }
  }
}
```

drupal8/core/themes/stable/templates/field/image.html.twig

```
<img{{ attributes }} />
```

CHROMATIC

```
// The resulting responsive image markup.
<img src="logo.png" alt="My logo!" sizes="100vw"
  srcset="styles/thumbnail/public/logo.png 250w,
    styles/large/public/logo.png 750w">
```

why are render arrays useful?

(ndrwfgg, Flickr)

theme functions such as theme_image()

hook_theme() defines theme hooks

the theme() function

theme functions such as theme_image()

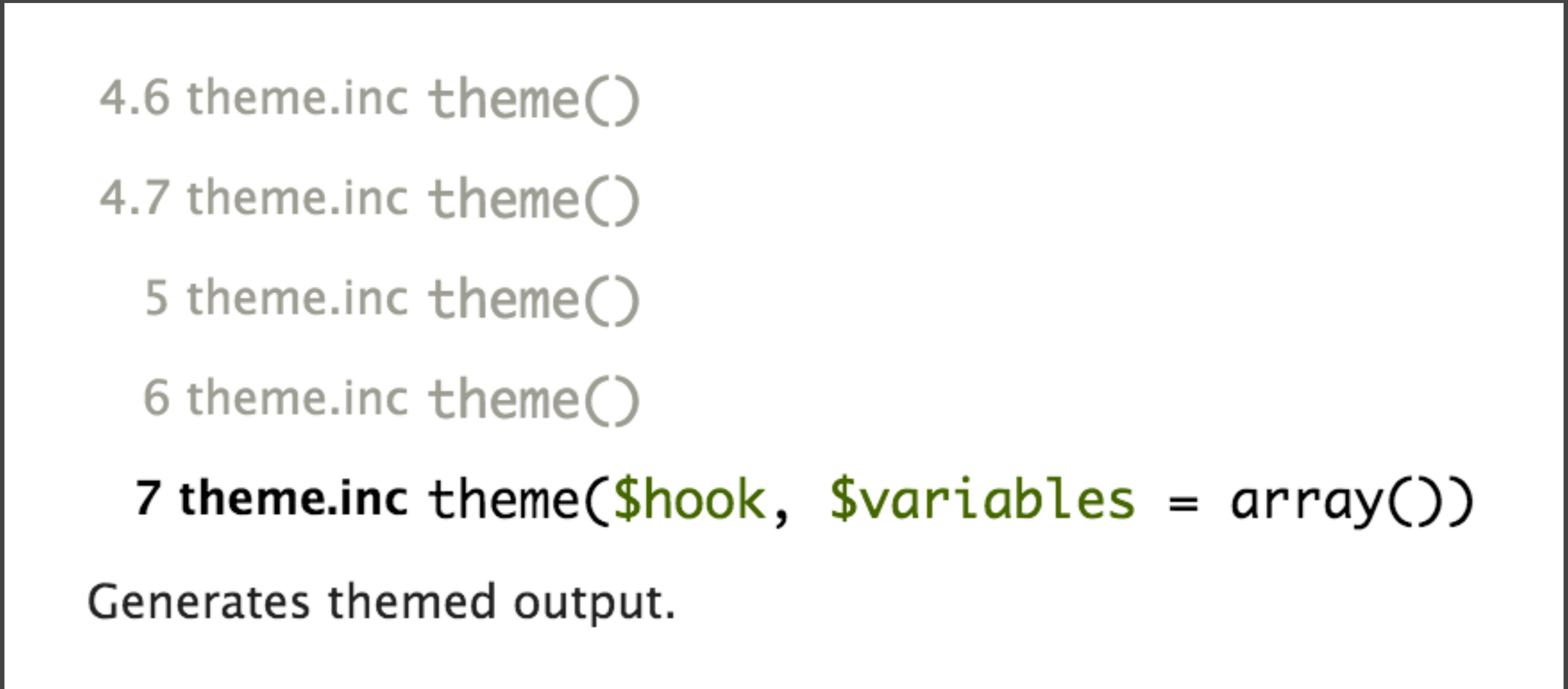
hook_theme() defines theme hooks

the theme() function

CHROMATIC

```
// Using theme() in Drupal 7.
$logo = theme('image', array(
  'path' => 'logo.png',
  'alt' => t('My logo!'),
));
```

https://api.drupal.org/api/drupal/includes!theme.inc/function/theme/7.x



```
4.6 theme.inc theme()
4.7 theme.inc theme()
  5 theme.inc theme()
  6 theme.inc theme()
  7 theme.inc theme($hook, $variables = array())
```

Generates themed output.

https://api.drupal.org/api/drupal/includes!theme.inc/function/theme/7.x

“Avoid calling this function directly. It is preferable to replace direct calls to the theme() function with calls to drupal_render() by passing a render array with a #theme key to drupal_render(), which in turn calls theme().”

drupal7/modules/system/page.tpl.php

```
<?php if ($logo): ?>
  <a href="<?php print $front_page; ?>" title="<?php print t('Home'); ?>"
    rel="home" id="logo">
    <img src="<?php print $logo; ?>" alt="<?php print t('Home'); ?>" />
  </a>
<?php endif; ?>
```

CHROMATIC

drupal7/modules/system/page.tpl.php

```
<?php if ($logo): ?>
  <a href="<?php print $front_page; ?>" title="<?php print t('Home'); ?>"
    rel="home" id="logo">
    <img src="<?php print $logo; ?>" alt="<?php print t('Home'); ?>" />
  </a>
<?php endif; ?>
```

drupal7/includes/theme.inc

```
/**
 * Preprocess variables for page.tpl.php
 */
function template_preprocess_page(&$variables) {
  $variables['logo'] = theme_get_setting('logo');
}
```

```php
/**
 * Implements hook_preprocess_page().
 */
function my_theme_preprocess_page(&$variables) {
  if (drupal_is_front_page()) {
    $variables['logo'] = 'logo-front.jpg';
  }
}
```

CHROMATIC

drupal7/modules/system/page.tpl.php

```
<?php if ($logo): ?>
  <a href="<?php print $front_page; ?>" title="<?php print t('Home'); ?>"
    rel="home" id="logo">
    <img src="<?php print $logo; ?>" alt="<?php print t('Home'); ?>" />
  </a>
<?php endif; ?>
```

theoretical drupal7/modules/system/page.tpl.php

```php
<?php if ($logo): ?>
  <a href="<?php print $front_page; ?>" title="<?php print t('Home'); ?>"
    rel="home" id="logo">
    <?php print render($logo); ?>
  </a>
<?php endif; ?>
```

theoretical drupal7/includes/theme.inc

```php
/**
 * Preprocess variables for page.tpl.php
 */
function template_preprocess_page(&$variables) {
  $variables['logo'] = theme('image', array(
    'path' => theme_get_setting('logo'),
    'alt' => t('Home'),
  ));
}
```

```
/**
 * Implements hook_preprocess_page().
 */
function my_theme_preprocess_page(&$variables) {
  dpm($variables['logo']);
}
```

CHROMATIC

```
// The current state of the logo variable.
<img src="logo.png" alt="Home" />
```

```php
/**
 * Implements hook_preprocess_page().
 */
function my_theme_preprocess_page(&$variables) {
  $new_logo = str_replace(' />', ' class="logo" />', $variables['logo']);
  $variables['logo'] = $new_logo;
}
```

theoretical drupal7/includes/theme.inc

```php
/**
 * Preprocess variables for page.tpl.php
 */
function template_preprocess_page(&$variables) {
  $variables['logo'] = array(
    '#theme' => 'image',
    '#path' => theme_get_setting('logo'),
    '#alt' => t('Home'),
  );
}
```

```
/**
 * Implements hook_preprocess_page().
 */
function my_theme_preprocess_page(&$variables) {
  dpm($variables['logo']);
}
```

```php
// The current state of the logo variable.
array(
  '#theme' => 'image',
  '#path' => 'logo.jpg',
  '#alt' => 'Home',
);
```

```
/**
 * Implements hook_preprocess_page().
 */
function my_theme_preprocess_page(&$variables) {
  $variables['logo']['#attributes']['class'][] = 'logo';
}
```

```
// The current state of the logo variable.
array(
  '#theme' => 'image',
  '#path' => 'logo.jpg',
  '#alt' => 'Home',
  '#attributes' => array(
    'class' => array('logo'),
  ),
);
```

theoretical drupal7/modules/system/page.tpl.php

```
<?php if ($logo): ?>
  <a href="<?php print $front_page; ?>" title="<?php print t('Home'); ?>"
    rel="home" id="logo">
    <?php print render($logo); ?>
  </a>
<?php endif; ?>
```

```
// The resulting markup with our class.
<img src="logo.png" alt="Home" class="logo" />
```

problem:

build feature X
like the last site
but
slightly different

```
// The current state of the logo variable.
array(
  '#theme' => 'image',
  '#path' => 'logo.jpg',
  '#alt' => 'Home',
);
```

```php
/**
 * Implements hook_preprocess_page().
 */
function my_theme_preprocess_page(&$variables) {
  $variables['logo']['#attributes']['class'][] = 'logo';
}
```

drupal8/core/themes/classy/templates/content/node.html.twig

```
{%
  set classes = [
    'node',
    'node--type-' ~ node.bundle|clean_class,
    node.isPromoted() ? 'node--promoted',
    node.isSticky() ? 'node--sticky',
    not node.isPublished() ? 'node--unpublished',
    view_mode ? 'node--view-mode-' ~ view_mode|clean_class,
  ]
%}
{{ attach_library('classy/node') }}
<article{{ attributes.addClass(classes) }}>
  // Node markup.
</article>
```

solution:

abstraction

the Render API

(john mcsporran, Flickr)

Theme System?

Render API?

CHROMATIC

Theme System:

hook_theme()

hook_preprocess_HOOK()

hook_theme_suggestions_HOOK()

Twig templating engine

Render API:

Render arrays

Render elements

Caching

Attachments

Placeholders

#theme
#type
#[variable]
#cache
#attached
#weight
#markup
#prefix
#suffix

#theme
#type
#[variable]
#cache
#attached
#weight
#markup
#prefix
#suffix

drupal8/core/lib/Drupal/Core/Render/theme.api.php

> “The core structure of the Render API
> is the render array, which is a
> hierarchical associative array
> containing data to be rendered
> and properties describing
> how the data should be rendered.”

Render API:

Render arrays

Render elements

Caching

Attachments

Placeholders

CHROMATIC

“Modules have the capability of defining elements, which are essentially prepackaged default render arrays.”

drupal8/core/lib/Drupal/Core/Render/Element/Checkbox.php

```
/**
 * @FormElement("checkbox")
 */
class Checkbox extends FormElement {

  public function getInfo() {
    $class = get_class($this);
    return array(
      '#input' => TRUE,
      '#return_value' => 1,
      '#process' => array(
        array($class, 'processCheckbox'),
        array($class, 'processAjaxForm'),
        array($class, 'processGroup'),
      ),
      '#pre_render' => array(
        array($class, 'preRenderCheckbox'),
        array($class, 'preRenderGroup'),
      ),
      '#theme' => 'input__checkbox',
      '#theme_wrappers' => array('form_element'),
      '#title_display' => 'after',
    );
  }
```

drupal8/core/lib/Drupal/Core/Render/Element/Checkbox.php

```php
/**
 * @FormElement("checkbox")
 */
class Checkbox extends FormElement {

  public function getInfo() {
    $class = get_class($this);
    return array(
      '#input' => TRUE,
      '#return_value' => 1,
      '#process' => array(
        array($class, 'processCheckbox'),
        array($class, 'processAjaxForm'),
        array($class, 'processGroup'),
      ),
      '#pre_render' => array(
        array($class, 'preRenderCheckbox'),
        array($class, 'preRenderGroup'),
      ),
      '#theme' => 'input__checkbox',
      '#theme_wrappers' => array('form_element'),
      '#title_display' => 'after',
    );
  }
```

CHROMATIC

```
// Our simple render element in D8.
$form['setting'] = array(
  '#type' => 'checkbox',
  '#title' => t('Are we having fun yet?'),
);
```

```
// Our simple render element in D8.
$form['setting'] = array(
  '#type' => 'checkbox',
  '#title' => t('Are we having fun yet?'),
);
```

```
// Our simple render element in D8.
$form['setting'] = array(
  '#type' => 'checkbox',
  '#title' => t('Are we having fun yet?'),
);
```

drupal8/core/lib/Drupal/Core/Render/Element/Checkbox.php

```
/**
 * @FormElement("checkbox")
 */
class Checkbox extends FormElement {

  public function getInfo() {
    $class = get_class($this);
    return array(
      '#input' => TRUE,
      '#return_value' => 1,
      '#process' => array(
        array($class, 'processCheckbox'),
        array($class, 'processAjaxForm'),
        array($class, 'processGroup'),
      ),
      '#pre_render' => array(
        array($class, 'preRenderCheckbox'),
        array($class, 'preRenderGroup'),
      ),
      '#theme' => 'input__checkbox',
      '#theme_wrappers' => array('form_element'),
      '#title_display' => 'after',
    );
  }
```

drupal8/core/lib/Drupal/Core/Render/Element/Checkbox.php

```
/**
 * @FormElement("checkbox")
 */
class Checkbox extends FormElement {

  public function getInfo() {
    $class = get_class($this);
    return array(
      '#input' => TRUE,
      '#return_value' => 1,
      '#process' => array(
        array($class, 'processCheckbox'),
        array($class, 'processAjaxForm'),
        array($class, 'processGroup'),
      ),
      '#pre_render' => array(
        array($class, 'preRenderCheckbox'),
        array($class, 'preRenderGroup'),
      ),
      '#theme' => 'input__checkbox',
      '#theme_wrappers' => array('form_element'),
      '#title_display' => 'after',
    );
  }
```

CHROMATIC

```
// Our simple render element in D8.
$form['setting'] = array(
  '#type' => 'checkbox',
  '#title' => t('Are we having fun yet?'),
);
```

drupal8/core/lib/Drupal/Core/Render/Element/Checkbox.php

```php
/**
 * @FormElement("checkbox")
 */
class Checkbox extends FormElement {

  public function getInfo() {
    $class = get_class($this);
    return array(
      '#input' => TRUE,
      '#return_value' => 1,
      '#process' => array(
        array($class, 'processCheckbox'),
        array($class, 'processAjaxForm'),
        array($class, 'processGroup'),
      ),
      '#pre_render' => array(
        array($class, 'preRenderCheckbox'),
        array($class, 'preRenderGroup'),
      ),
      '#theme' => 'input__checkbox',
      '#theme_wrappers' => array('form_element'),
      '#title_display' => 'after',
    );
  }
```

The Form API is a subset of the Render API.

```
// Our simple render element in D8.
$form['setting'] = array(
  '#type' => 'checkbox',
  '#title' => t('Are we having fun yet?'),
);
```

Render API:

Render arrays

Render elements

Caching

Attachments

Placeholders

CHROMATIC

```
// Caching a render array.
'#cache' => array(
  'keys' => array('entity_view', 'node', $node->id()),
  'contexts' => array('languages'),
  'tags' => array('node:' . $node->id()),
  'max-age' => Cache::PERMANENT,
),
```

Cacheability of render arrays

drupal.org/developing/api/8/render/arrays/cacheability

```php
// Attaching libraries and JS settings to render arrays.
$build['#attached']['library'][] = 'core/jquery';
$build['#attached']['drupalSettings']['foo'] = 'bar';
```

```php
// Using placeholders in render array markup.
$build['my_element'] = array(
  '#attached' => array('placeholders' => array('@foo' => 'replacement')),
  '#markup' => array('Something about @foo'),
);
```

```
// Using placeholders in render array markup.
$build['my_element'] = array(
  '#attached' => array('placeholders' => array('@foo' => 'replacement')),
  '#markup' => array('Something about @foo'),
);
```

#theme

#type

#[variable]

#cache

#attached

#weight

#markup

#prefix

#suffix

CHROMATIC

Render API:

Render arrays

Render elements

Caching

Attachments

Placeholders

the future

(Davis Doherty, Flickr)

CHROMATIC

[meta] Refactor Render API to be OO

drupal.org/node/1843798

Unify & simplify render & theme system: component-based rendering (enables pattern library, style guides, interface previews, client-side re-rendering)

drupal.org/node/2702061

Unify & simplify render & theme system: component-based rendering (enables pattern library, style guides, interface previews, client-side re-rendering)

drupal.org/node/2702061

CHROMATIC

Render arrays,
the current theme system,
and the current Render API
may be removed from Drupal 9.

I didn’t just waste
an hour of your time.

CHROMATIC

Render arrays are still important

…for now. :)

CHROMATIC

Render Arrays:

what they look like

how they work

why they’re handy

the Render API

their future

CHROMATIC

```
// Our simple render array in D8.
$logo = array(
  '#theme' => 'image',
  '#uri' => 'public://logo.png',
  '#alt' => t('My logo!'),
);
```

drupal8/core/modules/system/system.module
drupal8/core/includes/theme.inc
(simplified)

```
/**
 * Implements hook_theme().
 */
function system_theme() {
  return array(
    'image' => array(
      'variables' => array(
        'uri' => NULL,
        'width' => NULL,
        'height' => NULL,
        'alt' => '',
        'title' => NULL,
        'attributes' => array(),
        'sizes' => NULL,
        'srcset' => array(),
        'style_name' => NULL,
      ),
    ),
  );
}
```

CHROMATIC

```
// Printing/rendering anything in D8.
{{ logo }}
```

drupal8/core/includes/theme.inc

```php
/**
 * Prepares variables for image templates.
 */
function template_preprocess_image(&$variables) {
  if (!empty($variables['uri'])) {
    $variables['attributes']['src'] = file_url_transform_relative(file_create_url($variables['uri']));
  }

  // There is normally a lot of code here that takes the responsive variables
  // and turns them into properly formatted attribute values.

  foreach (array('width', 'height', 'alt', 'title', 'sizes') as $key) {
    if (isset($variables[$key])) {
      // If the property has already been defined in the attributes,
      // do not override, including NULL.
      if (array_key_exists($key, $variables['attributes'])) {
        continue;
      }
      $variables['attributes'][$key] = $variables[$key];
    }
  }
}
```

CHROMATIC

drupal8/core/themes/stable/templates/field/image.html.twig

```
<img{{ attributes }} />
```

```
// The resulting markup.
<img src="logo.png" alt="My logo!" />
```

my_theme/template.php

```php
/**
 * Implements hook_preprocess_page().
 */
function my_theme_preprocess_page(&$variables) {
  $variables['logo']['#attributes']['class'][] = 'logo';
}
```

CHROMATIC

```
// Our simple render/form element in D8.
$form['setting'] = array(
  '#type' => 'checkbox',
  '#title' => t('Are we having fun yet?'),
);
```

#theme
#type
#[variable]
#cache
#attached
#weight
#markup
#prefix
#suffix

```php
// Caching a render array.
'#cache' => array(
  'keys' => array('entity_view', 'node', $node->id()),
  'contexts' => array('languages'),
  'tags' => array('node:' . $node->id()),
  'max-age' => Cache::PERMANENT,
),

// Attaching libraries and JS settings to render arrays.
$build['#attached']['library'][] = 'core/jquery';
$build['#attached']['drupalSettings']['foo'] = ‘bar';

// Using placeholders in render array markup.
$build['my_element'] = array(
  '#attached' => array('placeholders' => array('@foo' => 'replacement')),
  '#markup' => array('Something about @foo'),
);
```

Aha! Understanding and Using Render Arrays in Drupal 8

CHROMATIC

further reading:

Render API Overview (8.x)

api.drupal.org/api/drupal/core!lib!Drupal!Core!Render!theme.api.php/group/theme_render/8.2.x

Theme system overview (8.x)

api.drupal.org/api/drupal/core!lib!Drupal!Core!Render!theme.api.php/group/themeable/8.2.x

Render Arrays in Drupal 7

drupal.org/node/930760

CHROMATIC

further reading:

Render API Overview (8.x)

api.drupal.org/api/drupal/core!lib!Drupal!Core!Render!theme.api.php/group/theme_render/8.2.x

Theme system overview (8.x)

api.drupal.org/api/drupal/core!lib!Drupal!Core!Render!theme.api.php/group/themeable/8.2.x

Render Arrays in Drupal 7

drupal.org/node/930760

CHROMATIC