7 to 8: Transitioning From theme() and Theme Functions to Render Arrays and Templates


Gus Childs
@guschilds
CHROMATIC

CHROMATIC

chromaticsites.com

CHROMATIC

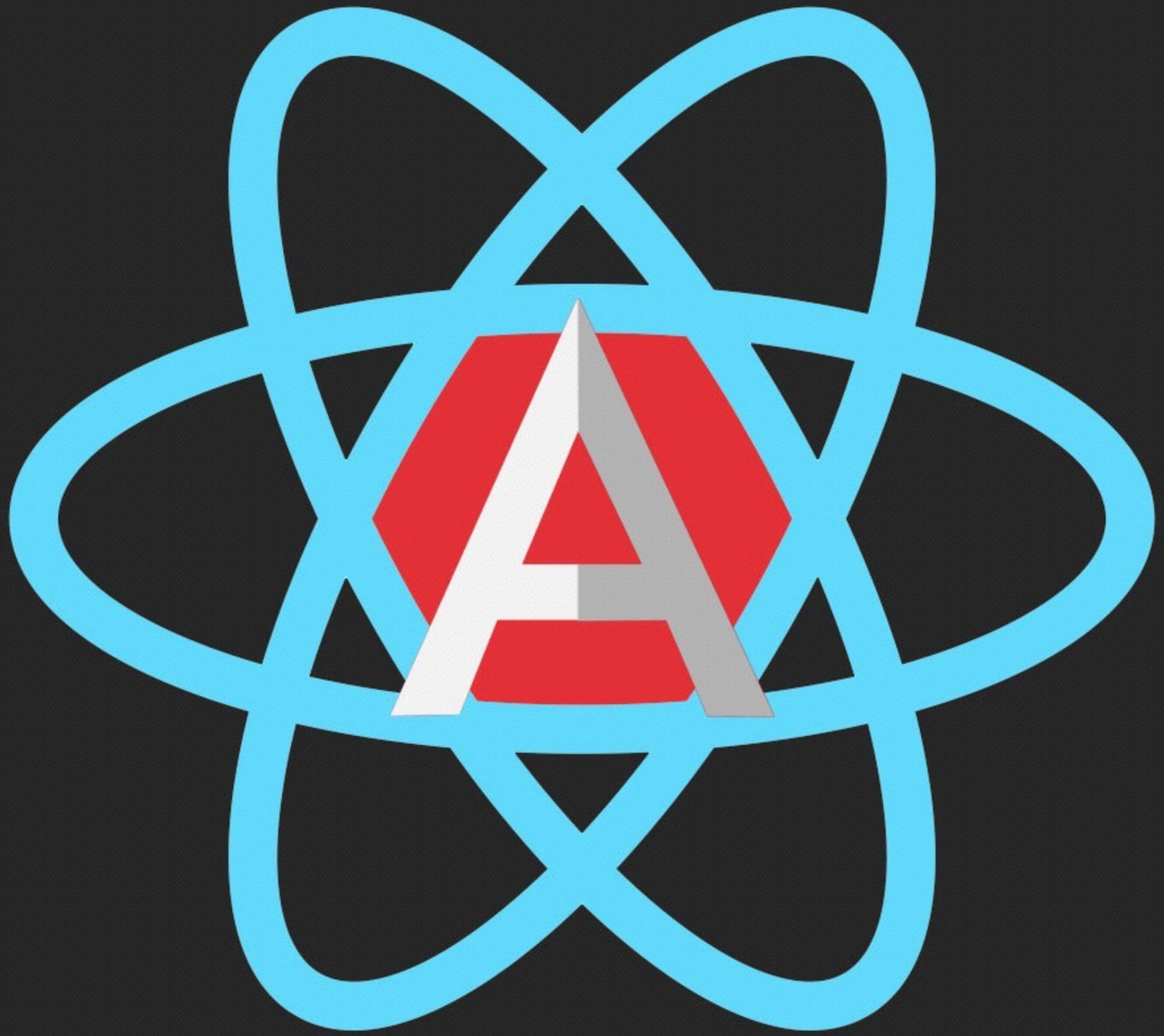
chromaticsites.com

front-end
back-end

front-end

this session

back-end



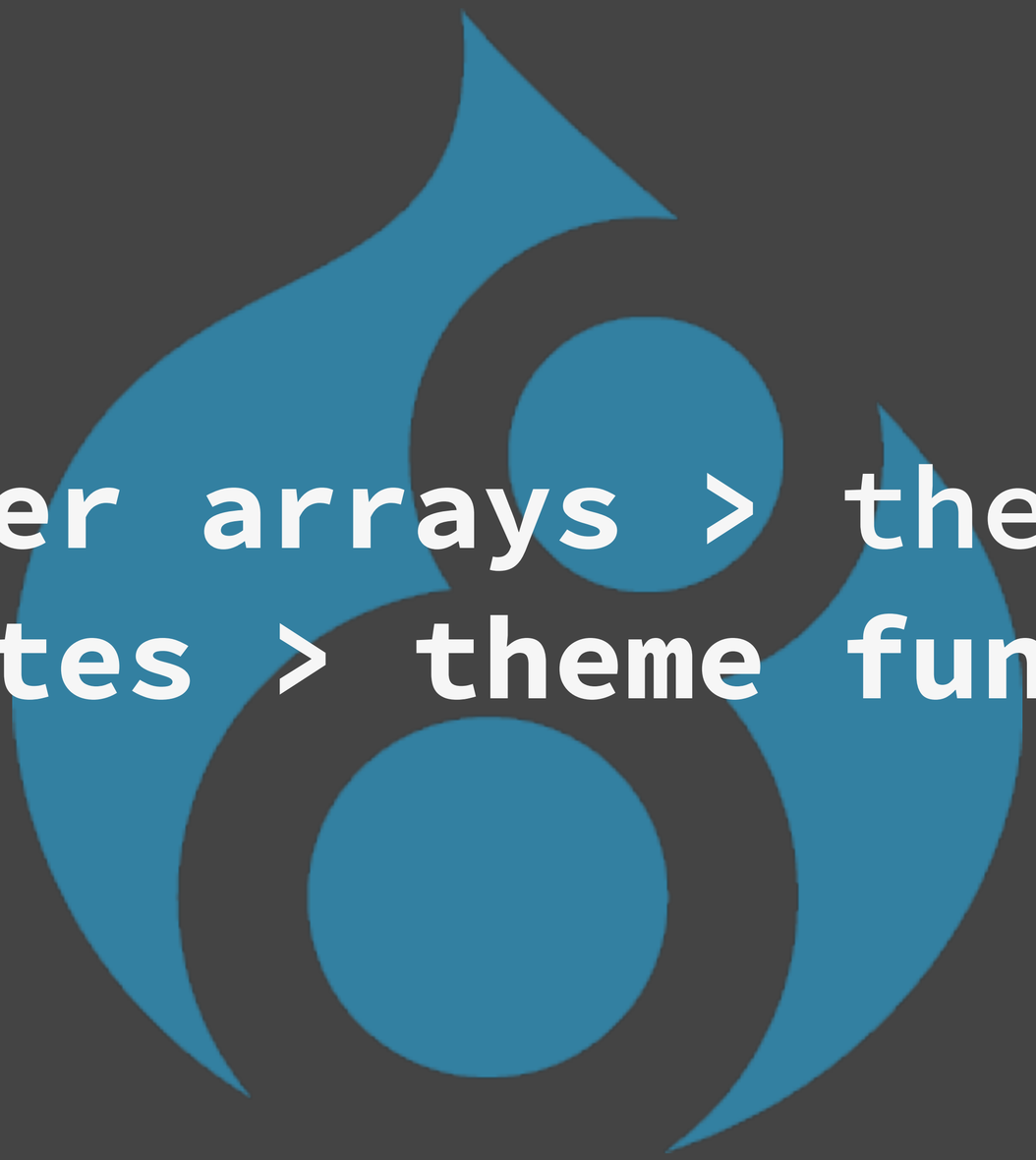
CHROMATIC

<h1>  </h1>

CHROMATIC

render arrays > theme()

templates > theme functions



render arrays > theme()
templates > theme functions

CHROMATIC

render arrays > theme()

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

```php
$variables['image'] = theme('image', array(
  'path' => 'logo.png',
  'alt' => t('My logo!'),
));
```

```
// Drupal 7.
<?php print $image; ?>
```

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

```
$variables['image'] = array(
   '#theme' => 'image',
   '#path' => 'logo.png',
   '#alt' => t('My logo!'),
);
```

```
// Drupal 7.
<?php print $image; ?>
```

CHROMATIC

```
// Drupal 7.
<?php print render($image); ?>
```

```
$variables['image'] = array(
   '#theme' => 'image',
   '#path' => 'logo.png',
   '#alt' => t('My logo!'),
);
```

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

```
// Drupal 8.
{{ image }}
```

```
$variables['header'] = array(
  'title' => array(...),
  'slogan' => array(...),
  'logo' => array(
    '#theme' => 'image',
    '#path' => 'logo.png',
    '#alt' => t('My logo!'),
  ),
);
```

```
$variables['header'] = array(
  'title' => array(...),
  'slogan' => array(...),
  'logo' => array(
    '#theme' => 'image',
    '#path' => 'logo.png',
    '#alt' => t('My logo!'),
  ),
);
```

```php
// From page.tpl.php:
<?php print render($page['sidebar']); ?>

// From node.tpl.php:
<?php print render($content); ?>
```

render arrays > theme()

templates > theme functions

```
// Theme functions?
// @see drupal.org/node/2575081
theme_image()
theme_item_list()
theme_link()
```

David Hwang
@eatings

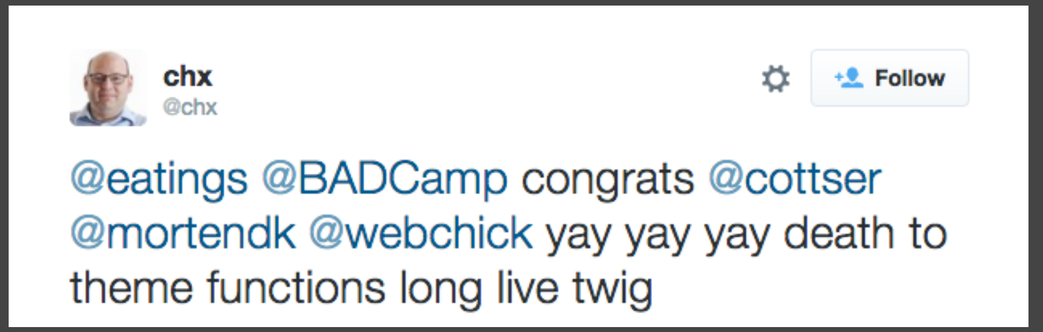
Follow

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

CHROMATIC


mortendk
@mortendk
Follow
die theme function die .... #drupaltwig @badcamp
$ git commit -m 'Issue #1938900 by
ntel, neochief, tstoeckler, andypost,
a template (DIE THEME FUNCTIONS DIE)


John Albin Wilkins
@JohnAlbin
Following
Good riddance!
David Hwang @eatings
As of @BADCamp 2015, #Drupal core no longer has any theme functions.


chx
@chx
Follow
@eatings @BADCamp congrats @cottser @mortendk @webchick yay yay yay death to theme functions long live twig

CHROMATIC


OILERS
9
SportsCha
CHROMATIC

```php
// Uses a render array.
$variables['image'] = array(
  '#theme' => 'image',
  '#path' => 'logo.png',
  '#alt' => t('My logo!'),
);

// Uses the theme() function.
$variables['image'] = theme('image', array(
  'path' => 'logo.png',
  'alt' => t('My logo!'),
));
```

```php
// Uses a render array.
$variables['image'] = array(
  '#theme' => 'image',
  '#path' => 'logo.png',
  '#alt' => t('My logo!'),
);

// Uses the theme() function.
$variables['image'] = theme('image', array(
  'path' => 'logo.png',
  'alt' => t('My logo!'),
));
```

```
// Theme function.
theme_image()

// Template.
image.tpl.php
```

```
// Theme function.
theme_image()

// Template.
// image.tpl.php
```

Drupal 7 » theme.inc

function theme_image

```
 7 theme.inc theme_image($variables)
4.6 theme.inc theme_image($path, $alt = '', $title = '', $attr = '', $getsize = true)
4.7 theme.inc theme_image($path, $alt = '', $title = '', $attributes = NULL, $getsize = TRUE)
  5 theme.inc theme_image($path, $alt = '', $title = '', $attributes = NULL, $getsize = TRUE)
  6 theme.inc theme_image($path, $alt = '', $title = '', $attributes = NULL, $getsize = TRUE)
```

Returns HTML for an image.

Parameters

$variables: An associative array containing:

- **path**: Either the path of the image file (relative to base_path()) or a full URL.
- **width**: The width of the image (if known).
- **height**: The height of the image (if known).
- **alt**: The alternative text for text-based browsers. HTML 4 and XHTML 1.0 always require an alt attribute. The HTML 5 draft allows the alt attribute to be omitted in some cases. Therefore, this variable defaults to an empty string, but can be set to NULL for the attribute to be omitted. Usually, neither omission nor an empty string satisfies accessibility requirements, so it is strongly encouraged for code calling theme('image') to pass a meaningful value for this variable.
  - http://www.w3.org/TR/REC-html40/struct/objects.html#h-13.8
  - http://www.w3.org/TR/xhtml1/dtds.html
  - http://dev.w3.org/html5/spec/Overview.html#alt
- **title**: The title text is displayed when the image is hovered in some popular browsers.
- **attributes**: Associative array of attributes to be placed in the img tag.

core's theme.inc

```
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
// Never do this!
$image = theme_image(array(
   'path' => 'logo.png',
   'alt' => t('My logo!'),
));
```

my_theme's template.php

```php
function my_theme_image($variables) {
  $attributes = $variables['attributes'];
  $attributes['data-src'] = file_create_url($variables['path']);
  $attributes['class'][] = 'lazyload';

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

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

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

```php
// Better, but still :(.
$image = theme('image', array(
  'path' => 'logo.png',
  'alt' => t('My logo!'),
));
```

core's theme.inc

```
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) . ' />';
}
```

core's 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;
}
```

CHROMATIC

```
$output = '<div class="item-list">';
```

core's 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;
}
```

CHROMATIC

```
// Theme function.
theme_image()

// Template.
image.tpl.php
```

```
// Theme function.
// theme_image()

// Template.
image.tpl.php
```

```
// Disclaimer: this is just an example.
// Core already does this, but in a different
// way, and you won't need to in order to
// render an image.
```

my_module.module

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

my_module.module

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

Drupal 7 » theme.inc

function theme_image

```
 7 theme.inc theme_image($variables)
4.6 theme.inc theme_image($path, $alt = '', $title = '', $attr = '', $getsi
4.7 theme.inc theme_image($path, $alt = '', $title = '', $attributes = NULL
TRUE)
 5 theme.inc theme_image($path, $alt = '', $title = '', $attributes = NULL
TRUE)
 6 theme.inc theme_image($path, $alt = '', $title = '', $attributes = NULL
TRUE)
```

Returns HTML for an image.

Parameters

$variables: An associative array containing:

- **path**: Either the path of the image file (relative to base_path()) or a full URL.
- **width**: The width of the image (if known).
- **height**: The height of the image (if known).
- **alt**: The alternative text for text-based browsers. HTML 4 and XHTML 1.0 always require an alt
HTML 5 draft allows the alt attribute to be omitted in some cases. Therefore, this variable defa
string, but can be set to NULL for the attribute to be omitted. Usually, neither omission nor an
satisfies accessibility requirements, so it is strongly encouraged for code calling theme('image'
meaningful value for this variable.
  - http://www.w3.org/TR/REC-html40/struct/objects.html#h-13.8
  - http://www.w3.org/TR/xhtml1/dtds.html
  - http://dev.w3.org/html5/spec/Overview.html#alt
- **title**: The title text is displayed when the image is hovered in some popular browsers.
- **attributes**: Associative array of attributes to be placed in the img tag.

core's theme.inc

```
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) . ' />';
}
```

core's theme.inc

```
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) . ' />';
}
```

```
// Drupal 7's theoretical image.tpl.php.
<img<?php print $attributes; ?> />

// Drupal 8's theoretical image.html.twig.
<img{{ attributes }} />
```

core's theme.inc

```
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) . ' />';
}
```

my_module.module

```php
/**
 * Implements hook_preprocess_HOOK() for image.
 */
function my_module_preprocess_image(&$variables) {
  $variables['attributes_array']['src'] = file_create_url($variables['path']);

  foreach (array('width', 'height', 'alt', 'title') as $key) {
    if (isset($variables[$key])) {
      $variables['attributes_array'][$key] = $variables[$key];
    }
  }
}
```

my_module.module

```php
/**
 * Implements hook_process_HOOK() for image.
 */
function my_module_process_image(&$variables) {
  $variables['attributes'] = drupal_attributes($variables['attributes_array']);
}
```

my_module.module

```php
/**
 * Implements hook_preprocess_HOOK() for image.
 */
function my_module_preprocess_image(&$variables) {
  $variables['attributes_array']['src'] = file_create_url($variables['path']);

  foreach (array('width', 'height', 'alt', 'title') as $key) {
    if (isset($variables[$key])) {
      $variables['attributes_array'][$key] = $variables[$key];
    }
  }
}

/**
 * Implements hook_process_HOOK() for image.
 */
function my_module_process_image(&$variables) {
  $variables['attributes'] = drupal_attributes($variables['attributes_array']);
}
```

my_theme's template.php

```
function my_theme_image($variables) {
  $attributes = $variables['attributes'];
  $attributes['data-src'] = file_create_url($variables['path']);
  $attributes['class'][] = 'lazyload';

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

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

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

my_theme's template.php

```php
/**
 * Implements hook_preprocess_HOOK() for image.
 */
function my_theme_preprocess_image(&$variables) {
  $variables['attributes_array']['class'][] = 'lazyload';
  $variables['attributes_array']['data-src'] = $variables['attributes_array']['src'];
  unset($variables['attributes_array']['src']);
}
```

```php
/**
 * Implements hook_preprocess_HOOK() for image.
 */
function my_module_preprocess_image(&$variables) {
  $variables['attributes_array']['src'] = file_create_url($variables['path']);

  foreach (array('width', 'height', 'alt', 'title') as $key) {
    if (isset($variables[$key])) {
      $variables['attributes_array'][$key] = $variables[$key];
    }
  }
}

/**
 * Implements hook_preprocess_HOOK() for image.
 */
function my_theme_preprocess_image(&$variables) {
  $variables['attributes_array']['class'][] = 'lazyload';
  $variables['attributes_array']['data-src'] = $variables['attributes_array']['src'];
  unset($variables['attributes_array']['src']);
}

/**
 * Implements hook_process_HOOK() for image.
 */
function my_module_process_image(&$variables) {
  $variables['attributes'] = drupal_attributes($variables['attributes_array']);
}
```

```php
/**
 * Implements hook_preprocess_HOOK() for image.
 */
function my_module_preprocess_image(&$variables) {
  $variables['attributes_array']['src'] = file_create_url($variables['path']);

  foreach (array('width', 'height', 'alt', 'title') as $key) {
    if (isset($variables[$key])) {
      $variables['attributes_array'][$key] = $variables[$key];
    }
  }
}

/**
 * Implements hook_preprocess_HOOK() for image.
 */
function my_theme_preprocess_image(&$variables) {
  $variables['attributes_array']['class'][] = 'lazyload';
  $variables['attributes_array']['data-src'] = $variables['attributes_array']['src'];
  unset($variables['attributes_array']['src']);
}

/**
 * Implements hook_process_HOOK() for image.
 */
function my_module_process_image(&$variables) {
  $variables['attributes'] = drupal_attributes($variables['attributes_array']);
}
```

```
$variables['image'] = array(
   '#theme' => 'image',
   '#path' => 'logo.png',
   '#alt' => t('My logo!'),
);
```

```
// Drupal 7.
<?php print render($image); ?>

// Drupal 8.
{{ image }}
```

```
// Without the lazyload preprocessing.
<img src="logo.png" alt="My logo!" />

// With the lazyload preprocessing.
<img data-src="logo.png" class="lazyload" alt="My logo!" />
```

```
// Drupal 7's theoretical image.tpl.php.
<img<?php print $attributes; ?> />

// Drupal 8's theoretical image.html.twig.
<img{{ attributes }} />
```

core's 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;
}
```

CHROMATIC

templates > theme functions

render arrays > theme()

```
// Example: You're building a module for a
// multi-site installation that renders the
// logo in a particular way. This logo will
// be printed in each site's page template
// and may be altered by each theme.
```

page.tpl.php

```
<header>
  <h1><?php print $title; ?></h1>
  <?php
    print theme('image', array(
      'path' => 'logo.png',
      'alt' => t('My logo!'),
    ));
  ?>
</header>
```

CHROMATIC

cool_logo.module

```php
/**
 * Implements hook_preprocess_HOOK() for page.
 */
function cool_logo_preprocess_page(&$variables) {
  $variables['logo'] = theme('image', array(
    'path' => 'logo.png',
    'alt' => t('My logo!'),
  ));
}
```

page.tpl.php

```
<header>
  <h1><?php print $title; ?></h1>
  <?php print $logo; ?>
</header>
```

cool_logo.module

```php
/**
 * Implements hook_preprocess_HOOK() for page.
 */
function cool_logo_preprocess_page(&$variables) {
  $variables['logo'] = theme('image', array(
    'path' => 'logo.png',
    'alt' => t('My logo!'),
  ));
}
```

my_theme's template.php

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

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

my_theme's template.php

```
/**
 * Implements hook_preprocess_HOOK() for page.
 */
function my_theme_preprocess_page(&$variables) {
  $find = 'My logo!';
  $replace = t('Woah! Logo!');
  $logo = str_replace($find, $replace, $variables['logo']);
  $variables['logo'] = $logo;
}
```

cool_logo.module

```php
/**
 * Implements hook_preprocess_HOOK() for page.
 */
function cool_logo_preprocess_page(&$variables) {
  $variables['logo'] = theme('image', array(
    'path' => 'logo.png',
    'alt' => t('My logo!'),
  ));
}
```

cool_logo.module

```php
/**
 * Implements hook_preprocess_HOOK() for page.
 */
function cool_logo_preprocess_page(&$variables) {
  $variables['logo'] = array(
    '#theme' => 'image',
    '#path' => 'logo.png',
    '#alt' => t('My logo!'),
  );
}
```

my_theme's template.php

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

```
array(
    '#theme' => 'image',
    '#path' => 'logo.png',
    '#alt' => t('My logo!'),
);
```

my_theme's template.php

```php
/**
 * Implements hook_preprocess_HOOK() for page.
 */
function my_theme_preprocess_page(&$variables) {
  // Change the alt text of the logo.
  $variables['logo']['#alt'] = t('Woah! Logo!');
}
```

page.tpl.php

```
<header>
  <h1><?php print $title; ?></h1>
  <?php print render($logo); ?>
</header>
```

overridden page.tpl.php

```
<header>
  <h1><?php print $title; ?></h1>
  <?php print render($logo); ?>
</header>
```

page.tpl.php

```
<header>
  <h1><?php print $title; ?></h1>
  <?php print render($logo); ?>
</header>
```

page.html.twig

```
<header>
  <h1>{{ title }}</h1>
  {{ logo }}
</header>
```

woah.

huh?

```php
// Uses a render array.
$variables['image'] = array(
  '#theme' => 'image',
  '#path' => 'logo.png',
  '#alt' => t('My logo!'),
);

// Uses the theme() function.
$variables['image'] = theme('image', array(
  'path' => 'logo.png',
  'alt' => t('My logo!'),
));
```

```
// Theme function.
theme_image()

// Template.
image.tpl.php
```

my_module.module

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

core's theme.inc

```
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) . ' />';
}
```

core's 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;
}
```

CHROMATIC

```
// Drupal 7's theoretical image.tpl.php.
<img<?php print $attributes; ?> />

// Drupal 8's theoretical image.html.twig.
<img{{ attributes }} />
```

my_module.module

```php
/**
 * Implements hook_preprocess_HOOK() for image.
 */
function my_module_preprocess_image(&$variables) {
  $variables['attributes_array']['src'] = file_create_url($variables['path']);

  foreach (array('width', 'height', 'alt', 'title') as $key) {
    if (isset($variables[$key])) {
      $variables['attributes_array'][$key] = $variables[$key];
    }
  }
}

/**
 * Implements hook_process_HOOK() for image.
 */
function my_module_process_image(&$variables) {
  $variables['attributes'] = drupal_attributes($variables['attributes_array']);
}
```

my_theme's template.php

```
function my_theme_image($variables) {
  $attributes = $variables['attributes'];
  $attributes['data-src'] = file_create_url($variables['path']);
  $attributes['class'][] = 'lazyload';

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

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

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

my_theme's template.php

```php
/**
 * Implements hook_preprocess_HOOK() for image.
 */
function my_theme_preprocess_image(&$variables) {
  $variables['attributes_array']['class'][] = 'lazyload';
  $variables['attributes_array']['data-src'] = $variables['attributes_array']['src'];
  unset($variables['attributes_array']['src']);
}
```

```php
/**
 * Implements hook_preprocess_HOOK() for image.
 */
function my_module_preprocess_image(&$variables) {
  $variables['attributes_array']['src'] = file_create_url($variables['path']);

  foreach (array('width', 'height', 'alt', 'title') as $key) {
    if (isset($variables[$key])) {
      $variables['attributes_array'][$key] = $variables[$key];
    }
  }
}

/**
 * Implements hook_preprocess_HOOK() for image.
 */
function my_theme_preprocess_image(&$variables) {
  $variables['attributes_array']['class'][] = 'lazyload';
  $variables['attributes_array']['data-src'] = $variables['attributes_array']['src'];
  unset($variables['attributes_array']['src']);
}

/**
 * Implements hook_process_HOOK() for image.
 */
function my_module_process_image(&$variables) {
  $variables['attributes'] = drupal_attributes($variables['attributes_array']);
}
```

page.tpl.php

```
<header>
  <h1><?php print $title; ?></h1>
  <?php
    print theme('image', array(
      'path' => 'logo.png',
      'alt' => t('My logo!'),
    ));
  ?>
</header>
```

CHROMATIC

cool_logo.module

```php
/**
 * Implements hook_preprocess_HOOK() for page.
 */
function cool_logo_preprocess_page(&$variables) {
  $variables['logo'] = theme('image', array(
    'path' => 'logo.png',
    'alt' => t('My logo!'),
  ));
}
```

my_theme's template.php

```
/**
 * Implements hook_preprocess_HOOK() for page.
 */
function my_theme_preprocess_page(&$variables) {
  $find = 'My logo!';
  $replace = t('Woah! Logo!');
  $logo = str_replace($find, $replace, $variables['logo']);
  $variables['logo'] = $logo;
}
```

cool_logo.module

```php
/**
 * Implements hook_preprocess_HOOK() for page.
 */
function cool_logo_preprocess_page(&$variables) {
  $variables['logo'] = array(
    '#theme' => 'image',
    '#path' => 'logo.png',
    '#alt' => t('My logo!'),
  );
}
```

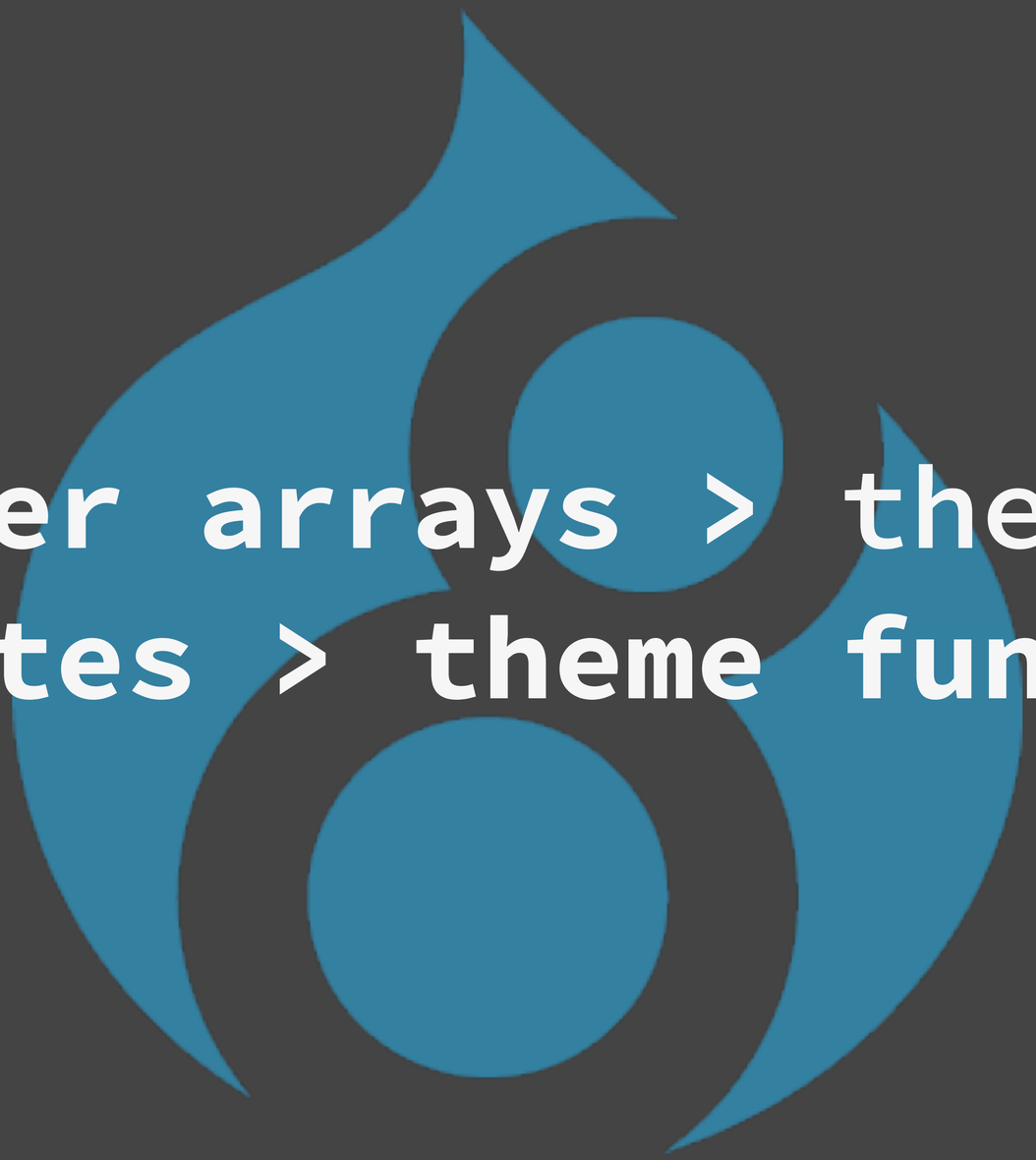
my_theme's template.php

```php
/**
 * Implements hook_preprocess_HOOK() for page.
 */
function my_theme_preprocess_page(&$variables) {
  // Change the alt text of the logo.
  $variables['logo']['#alt'] = t('Woah! Logo!');
}
```

page.tpl.php

```
<header>
  <h1><?php print $title; ?></h1>
  <?php print render($logo); ?>
</header>
```

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



render arrays > theme()
templates > theme functions

CHROMATIC

further reading

Render Arrays in Drupal 7.
drupal.org/node/930760
Twig best practices – preprocess functions, templates, and render arrays.
drupal.org/node/1920746

CHROMATIC

thanks!


questions?
CHROMATIC