



Drupal 8 Theming Deep Dive



#drupaldevdays

wunderkraut 

Smile
OPEN SOURCE SOLUTIONS

adyax

Kaliop
INTERACTIVE MEDIA

mSYSTEMS

actency
Strategies & Solutions Web

TRAINED
PEOPLE

Amazee
labs

Romain JARRAUD

Trainer/consultant

Trained People

@romainjarraud (mainly in french)

What is theming?

Theming?

Fonctionnal

Drupal and modules

Display

Theme

Theming?

- **THEMING = HTML output**
- Other outputs could be RSS feed, JSON...
- HEADfull Drupal8 here!

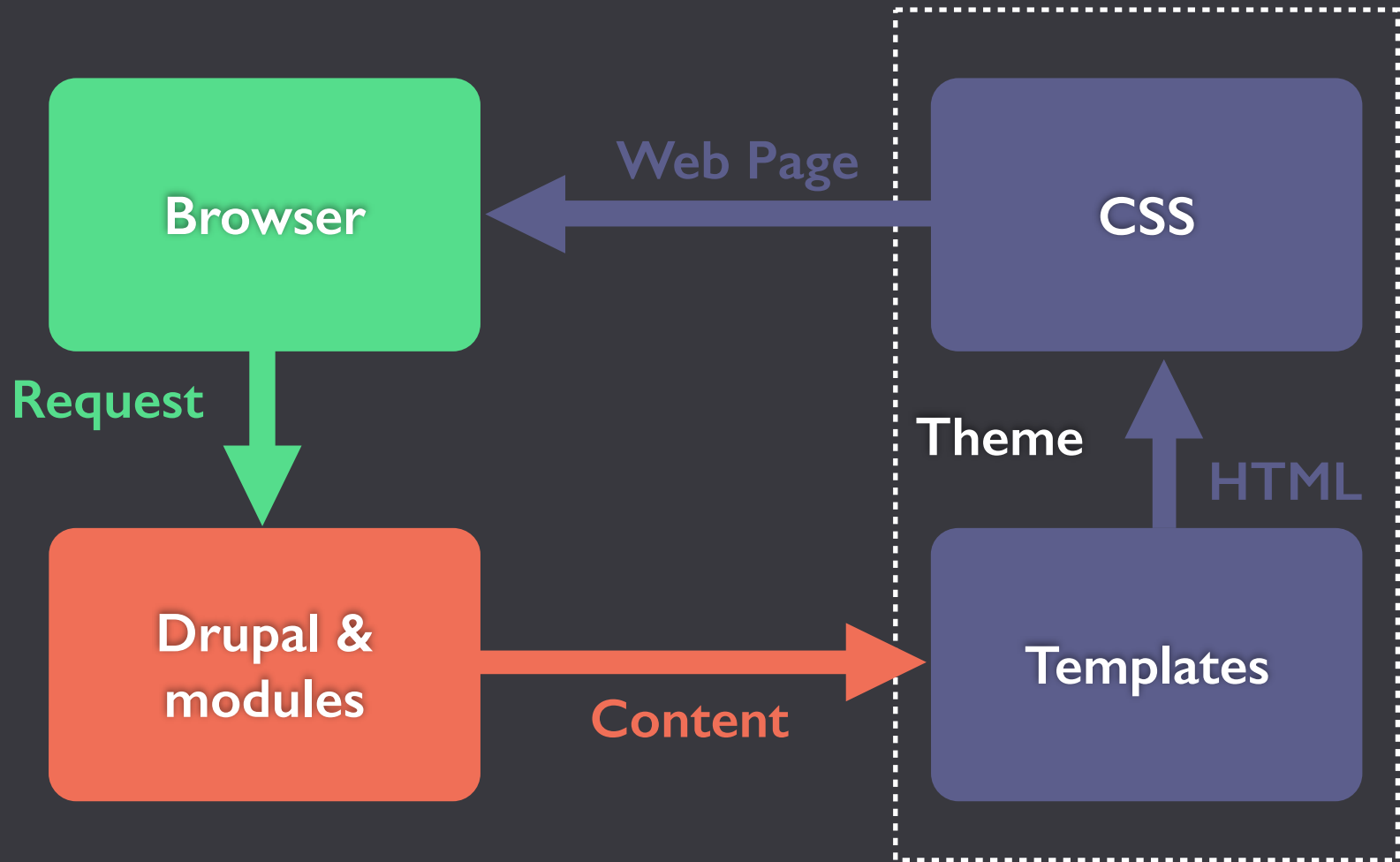
Theming?

- **Modules** produce **datas**.
- Modules have default **renderings**.
- For example, **Block** module define **block.twig.html** template which render each block.
- The theme **OVERRIDES** the default renderings.

Theming?

Theme overrides defaults:

- **html code** for every element on the page.
- **styles**: font size, colors, images...
- **effects** using javascript.



Create a theme

Create a theme

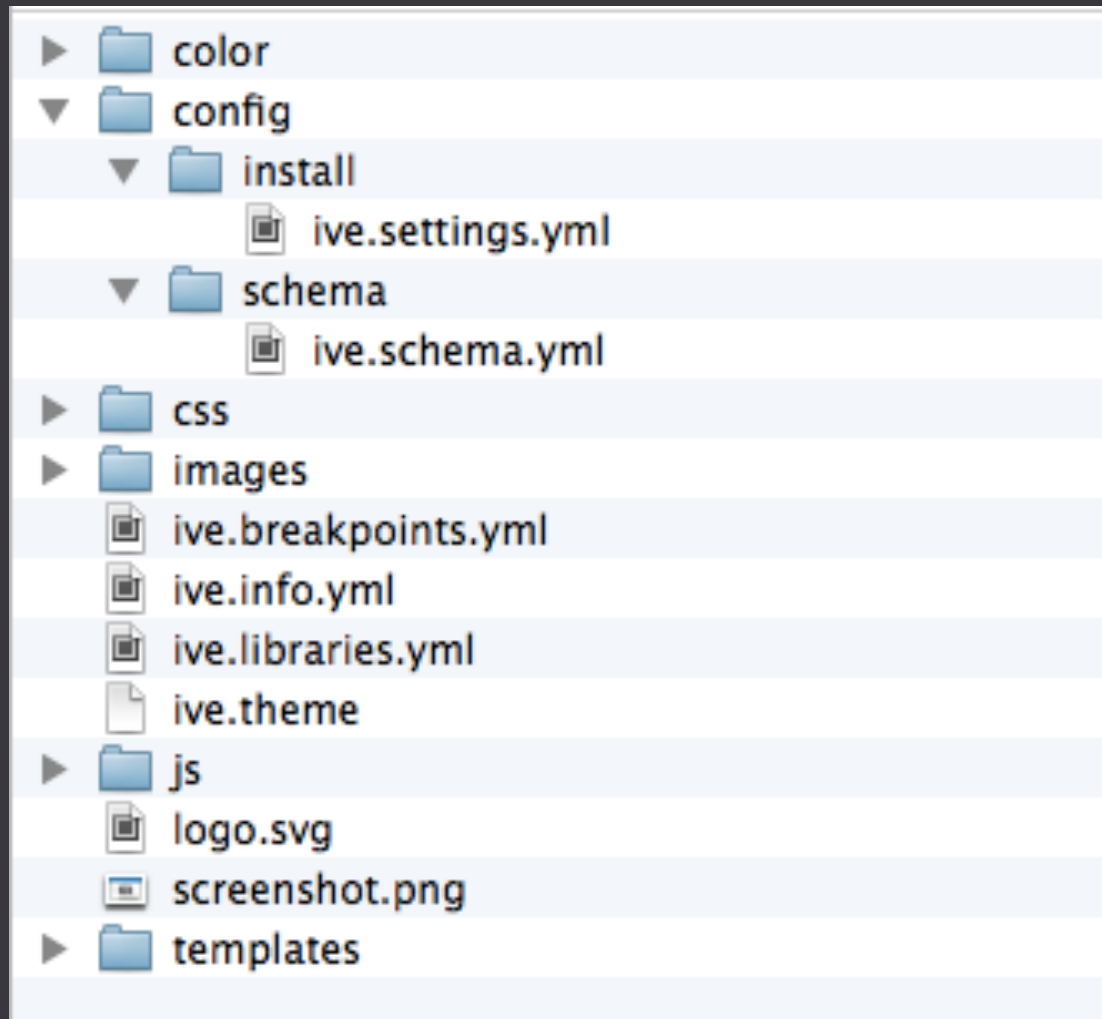
- `/themes/ive/ive.info.yml`

```
name: Ive
type: theme
description: 'Example theme.'
version: VERSION
core: '8.x'
regions:
  header: Header
  primary_menu: 'Primary menu'
  secondary_menu: 'Secondary menu'
  help: Help
  page_top: 'Page top'
  page_bottom: 'Page bottom'
  featured_top: 'Featured top'
  breadcrumb: Breadcrumb
  content: Content
  sidebar_first: 'Sidebar first'
  sidebar_second: 'Sidebar second'
  featured_bottom_first: 'Featured bottom first'
  featured_bottom_second: 'Featured bottom second'
  featured_bottom_third: 'Featured bottom third'
  footer_first: 'Footer first'
  footer_second: 'Footer second'
  footer_third: 'Footer third'
  footer_fourth: 'Footer fourth'
  footer_fifth: 'Footer fifth'
```

End of the story!

Maybe not...

Theme files



Templates

TWIG

- Drupal 8 uses **TWIG**.
- TWIG created by **Fabien Potencier**.
- Templating system to generate HTML.

TWIG

- Template name *.html.twig*: example of *page.html.twig* for the page template.
- They hold the HTML tags along with **TWIG code**.

TWIG (quickly)

- Display the content of var: `{{ var }}`.
- Display any kind of variable (string, array, object):
`{{ node.title }}`.
- Function: `{% if var %} {% endif %}`.
- Comments: `{# comment #}`
- Translation:
 - `{% trans %} translatable {{ string }} {% endtrans %}`.
 - Ready to be translated through Drupal UI!

TWIG (quickly)

- Filters:
 - `{{ date|format_date('medium') }}`
 - string:
 - Escaping: `{{ text }}` (default).
 - No escaping: `{{ text|passthrough }}` (be careful!).
 - Placeholder : `{{ text|placeholder }}`.
 - `{{ content|without('links') }}`.
 - `{{ string|lower }}` (upper as well).
 - `{{ class_name|clean_class }}`.
 - `{{ id_name|clean_id }}`.

TWIG

/themes/ive/templates/block.html.twig

```
{%
  set classes = [
    'block',
    'block-' ~ configuration.provider|clean_class,
  ]
%}
<div{{ attributes.addClass(classes) }}>
  {{ title_prefix }}
  {% if label %}
    <h2{{ title_attributes }}>{{ label }}</h2>
  {% endif %}
  {{ title_suffix }}
  {% block content %}
    <div{{ content_attributes.addClass('content') }}>
      {{ content }}
    </div>
  {% endblock %}
</div>
```

TWIG Debug

Activate Twig Debug in local environment by editing the file `/sites/default/services.yml`.

```
#  
# Not recommended in production environments  
# @default false  
debug: true  
# Twig auto-reload:
```

TWIG Debug

- Not enough because Drupal caches entity rendering.
- Uncomment the loading of a `local.settings.php` file in `/sites/default/settings.php`.

```
*/  
if (file_exists(__DIR__ . '/settings.local.php')) {  
    include __DIR__ . '/settings.local.php';  
}  
$databases['default']['default'] = array (
```

That file adds a backend cache render service which avoid Drupal to cache.

Template suggestions

- Depending on context you can ask Drupal to load a specific template.
- For example to display a node Drupal uses *node.html.twig*. But it can use *node--article.html.twig* (if it exists!) for any node of type article.

Template suggestions

- A module defines suggestions with `hook_theme_suggestions_HOOK()`.
- Other modules or themes can add suggestions with `hook_theme_suggestions_alter()` or `hook_theme_suggestions_HOOK_alter()`.

Template suggestions

```
▼ <div class="content">
  <!-- THEME DEBUG -->
  <!-- THEME HOOK: 'node' -->
  <!-- FILE NAME SUGGESTIONS:
    * node--1--full.html.twig
    * node--1.html.twig
    * node--article--full.html.twig
    * node--article.html.twig
    * node--full.html.twig
    x node.html.twig
  -->
  <!-- BEGIN OUTPUT from 'themes/ive/templates/node.html.twig' -->
  ▶ <article data-history-node-id="1" data-quickedit-entity-id="node/1" role="art
```

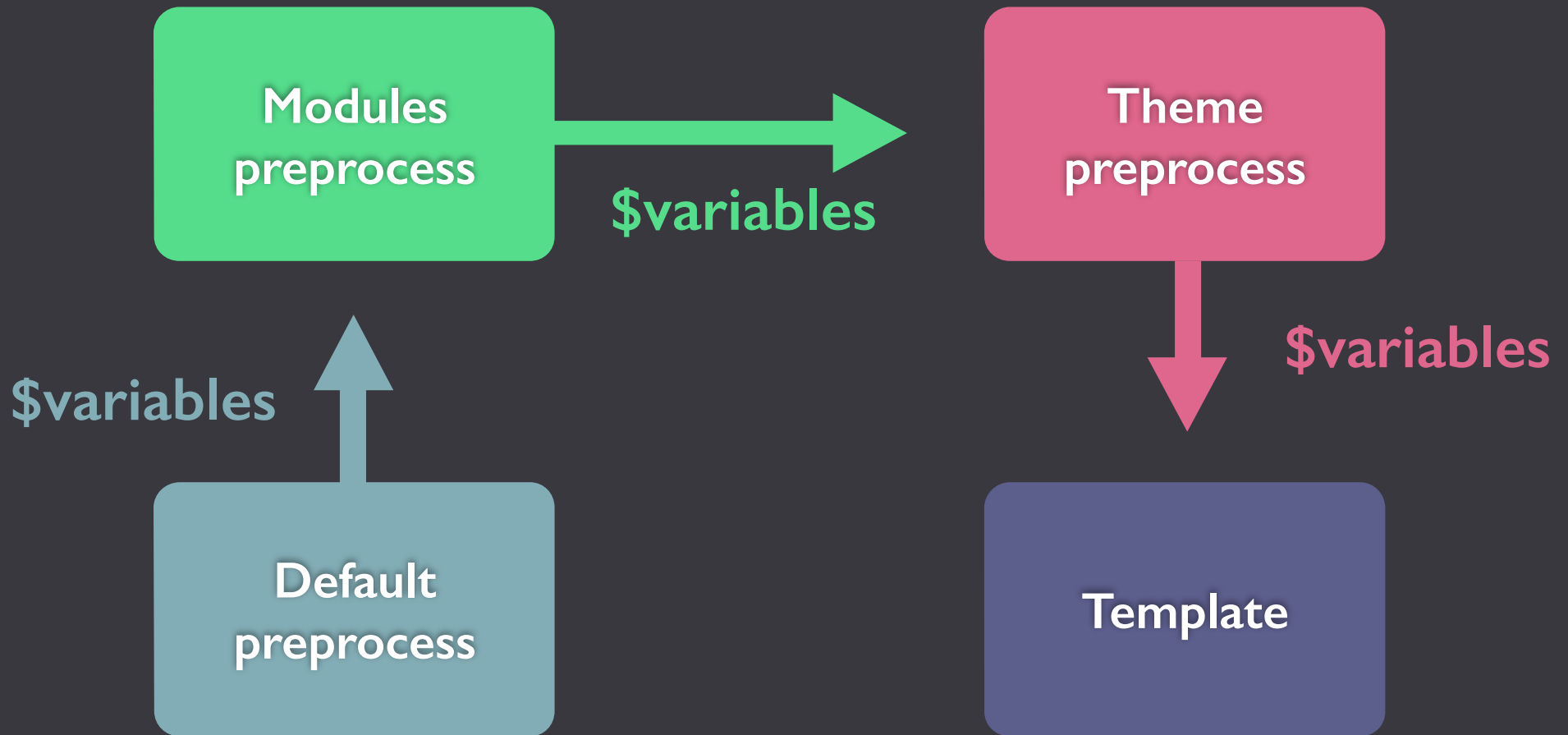
(Those comments are shown
thanks to TWIG Debug mode)

Override

- **Copy original file** into theme template folder.
- Rename it if necessary.
- **Empty theme registry.**
- Do what you want!

Preprocess functions

Preprocess functions



Called functions order

Theme suggestions

- **MODULE**_theme_suggestions_**HOOK**(array \$variables)
- **OTHERMODULE**_theme_suggestions_alter(array &\$suggestions, array \$variables, \$hook)
- **OTHERMODULE**_theme_suggestions_**HOOK**_alter(array &\$suggestions, array \$variables)
- **THEME**_theme_suggestions_alter(array &\$suggestions, array \$variables, \$hook)
- **THEME**_theme_suggestions_**HOOK**_alter(array &\$suggestions, array \$variables)

Preprocess

- **template**_preprocess_**HOOK**(array &\$variables)
- **OTHERMODULE**_preprocess(array &\$variables, \$hook)
- **OTHERMODULE**_preprocess_**HOOK**(array &\$variables)
- **THEME**_preprocess(array &\$variables, \$hook)
- **THEME**_preprocess_**HOOK**(array &\$variables)

Librairies

Libraries

- **No more manually asset loading.**
- Bye bye *drupal_add_css()*, *drupal_add_js()* and *drupal_add_library()*!
- Must add any **asset** through a **library**.
- Drupal takes care of libraries loading and dependencies.

Libraries

Declaration

Loading

Libraries declaration

`/themes/ive/ive.libraries.yml`

```
ive:  
  version: VERSION  
  css:  
    base:  
      css/base.css: {}  
    layout:  
      css/layout.css: {}  
    component:  
      css/component.css: {}  
    state:  
      css/state.css: {}  
    theme:  
      css/theme.css: {}  
  js:  
    js/ive.js: {}
```

Libraries loading

Loading from `/themes/ive/ive.info.yml`

```
name: Ive
type: theme
description: 'Example theme.'
version: VERSION
core: '8.x'
libraries:
  - ive/global-styling
  - ive/ive
regions:
  header: Header
  primary_menu: 'Primary menu'
  secondary_menu: 'Secondary menu'
  help: Help
  page_top: 'Page top'
  page_bottom: 'Page bottom'
  featured_top: 'Featured top'
  breadcrumb: Breadcrumb
  content: Content
  sidebar_first: 'Sidebar first'
  sidebar_second: 'Sidebar second'
  featured_bottom_first: 'Featured bottom first'
  featured_bottom_second: 'Featured bottom second'
  featured_bottom_third: 'Featured bottom third'
  footer_first: 'Footer first'
  footer_second: 'Footer second'
  footer_third: 'Footer third'
  footer_fourth: 'Footer fourth'
  footer_fifth: 'Footer fifth'
```


Libraries loading

Loading from a template file

/themes/ive/templates/block.html.twig

```
{{ attach_library('ive/ive') }}
```

Libraries loading

Loading from `/themes/ive/ive.theme` using
`THEME_page_attachments_alter()`

```
/**  
 * Implements hook_page_attachments_alter().  
 *  
 * @param array &$attachments  
 */  
function ive_page_attachments_alter(array &$attachments) {  
    $attachments ['#attached']['library'][] = 'ive/ive';  
}
```

Libraries loading

Loading from **/themes/ive/ive.theme** using a
preprocess fonction

```
/**  
 * Implements hook_preprocess_HOOK() for page templates.  
 */  
function ive_preprocess_page(&$variables) {  
    $variables['#attached']['library'][] = 'ive/ive';  
}
```

Library dependency

- No javascript loaded for anonymous users.
- Dependencies should be explicit.

/themes/ive/ive.libraries.yml

```
ive:
  version: VERSION
  css:
    base:
      css/base/base.css: {}
    layout:
      css/layout/layout.css: {}
    component:
      css/components/component.css: {}
    state:
      css/state/state.css: {}
    theme:
      css/theme/theme.css: {}
  js:
    js/ive.js: {}
  dependencies:
    - core/jquery
```

From PHP to JS

/themes/ive/ive.theme

```
/**
 * Implements hook_page_attachments_alter().
 *
 * @param array &$attachments
 */
function ive_page_attachments_alter(array &$attachments) {
    $attachments['#attached']['drupalSettings']['ive']['foo'] = 'bar';
}
```

/themes/ive/js/ive.js

```
(function ($, Drupal, drupalSettings) {

    $foo = drupalSettings.ive.foo;
    alert($foo);

})(jQuery, Drupal, drupalSettings);
```

Breakpoints

Breakpoints

Breakpoints are exposed on the **server side**.

For example, images can be loaded depending on those breakpoints (using *Responsive Image* module).

Breakpoints

[/themes/ive/ive.breakpoints.yml](#)

```
ive.mobile:  
  label: mobile  
  mediaQuery: ''  
  weight: 2  
  multipliers:  
    - 1x  
ive.narrow:  
  label: narrow  
  mediaQuery: 'all and (min-width: 560px) and (max-width: 850px)'  
  weight: 1  
  multipliers:  
    - 1x  
ive.wide:  
  label: wide  
  mediaQuery: 'all and (min-width: 851px)'  
  weight: 0  
  multipliers:  
    - 1x
```


Breakpoints

Label *
 Machine name: article
Example: 'Hero image' or 'Author image'.

Breakpoint group *

Select a breakpoint group from the installed themes. Warning: if you change the breakpoint group you lose all your selected image style mappings.

1x wide [all and (min-width: 851px)]

Select an image style for this breakpoint.

1x narrow [all and (min-width: 560px) and (max-width: 850px)]


Select an image style for this breakpoint.

1x mobile []

Select an image style for this breakpoint.

[Save](#) [Delete](#)

[Show row weights](#)

FIELD	LABEL	FORMAT
 Image	<input type="text" value="- Hidden -"/>	Format settings: Responsive image Responsive image style * <input type="text" value="Article"/> Fallback image style <input type="text" value="Automatic"/> Link image to <input type="text" value="Nothing"/> Update Cancel

Configuration

Configuration

- **How to add settings** in the backoffice.
- Each theme gets its own default form.
- Alter the form with `THEME_form_system_theme_settings_alter()` (using the **Form API**).

Configuration

`/themes/ive/config/install/ive.settings.yml`

```
ive: one
```

Defines:

- `ive.settings` configuration.
- « `ive` » with its default value « `one` ».

This configuration is loaded during installation.

Configuration

/themes/ive/ive.theme

```
function ive_form_system_theme_settings_alter(array &$form, FormStateInterface $form_state) {
    $config = \Drupal::config('ive.settings');

    $form['ive'] = array(
        '#type' => 'select',
        '#title' => t('Add a class to node template'),
        '#options' => array(
            '' => t('No class'),
            'one' => t('One'),
            'two' => t('Two'),
            'three' => t('Three'),
        ),
        '#default_value' => $config->get('ive'),
    );
}

/**
 * Implements hook_preprocess_HOOK() for node.html.twig.
 */
function ive_preprocess_node(&$variables) {
    // Add a class from the Ive Theme settings form.
    $config = \Drupal::config('ive.settings');
    $variables['attributes']['class'][] = $config->get('ive');

    // Remove the "Add new comment" link on teasers or when the comment form is
    // displayed on the page.
    if ($variables['teaser'] || !empty($variables['content']['comments']['comment_form'])) {
        unset($variables['content']['links']['comment']['#links']['comment-add']);
    }
}
```

Configuration

▼ LOGO IMAGE SETTINGS

Use the default logo supplied by the theme

▼ SHORTCUT ICON SETTINGS

Your shortcut icon, or 'favicon', is displayed in the address bar and bookmarks of most browsers.

Use the default shortcut icon supplied by the theme

Add a class to node template

One ▼

Save configuration

```
<!-- BEGIN OUTPUT from 'themes/ive/templates/node.html.twig' -->
▶<article data-history-node-id="1" data-quickedit-entity-id="node/1" role="article" class="contextual-region one node node--
type-article node--promoted node--view-mode-full clearfix" about="/node/1" typeof="schema:Article" data-quickedit-entity-
instance-id="0">...</article>
<!-- END OUTPUT from 'themes/ive/templates/node.html.twig' -->
</div>
```

Theme hook declaration

Theme hook declaration

- Each theme hook is associated with a **template** file.
- Each theme hook got its own preprocess function (*template_preprocess_HOOK()*) and theme suggestions function (*MODULE_theme_suggestions_HOOK()*).

Theme hook declaration

/modules/simple/simple.module

```
/**
 * Implements hook_theme().
 */
function simple_theme($existing, $type, $theme, $path) {
  return array(
    'simple' => array(
      'variables' => array('data' => NULL),
      'template' => 'simple',
    ),
  );
}
```

Theme hook declaration

[/modules/simple/src/Controller/simple.php](#)

```
<?php

namespace Drupal\simple\Controller;

use Drupal\Core\Controller\ControllerBase;

class simple extends ControllerBase {
    public function content() {
        return array(
            '#theme' => 'simple',
            '#data' => 'simple',
        );
    }
}
```

A few more things...

- **Everything is a bloc** (nearly): breadcrumb, messages...
- **Logo** format is **SVG**.
- Base theme **Classy**.
- No more theme functions but templates instead.
- No more *theme()* fonction but **Render Arrays**.
- No more process functions.
- PLEASE, no more database queries in templates!!!

Conclusion

Not so many things

to learn compared to Drupal 7.

Easier with Drupal 8!

Thanks!

Q&A