Advanced Drupal Front-End Development

FLDC

FLORIDA

DRUPAL CAMP

About Me

- Andy Blum, Front End Dev @ Lullabot
- Ohio, USA
- Science teacher 2013-2016
- Front End/Drupal developer 2016-present
    - Drupal Dev
        - Site building
        - Custom theming
        - Basic module development
        - Core contribution (olivero)
    - Front-end Dev
        - HTML/CSS/JS
        - SSG (Jekyll, Gatsby)
        - React, Next.js
        - Node.js

About Me

@andy-blum (1 hyphen)

- Github
- Drupal.org
- Drupal Slack
- Drupaltwig Slack
- LinkedIn

@andy__blum (2 underscores)

- Twitter

Topics To Be Covered

- Advanced theme setup & configuration like the Libraries API and Theme Settings
- Utilizing PHP and Drupal’s core & contrib functionality to enhance your theme
- JavaScript and Drupal Behaviors
- Advanced twig concepts
- Debugging and troubleshooting methods

First: What’s Your Experience?

- Skill Levels
  - Novice → Intermediate → Advanced
  - Low → High
- Skill Domains
  - PHP
  - JavaScript
  - Drupal
  - Twig

Skill Level: PHP

NOVICE:

I’ve never touched PHP in my life

↓

I can write variables & navigate arrays/objects

INTERMEDIATE:

I write functions that manipulate data in some basic ways

↓

I’ve done a little bit of object-oriented code

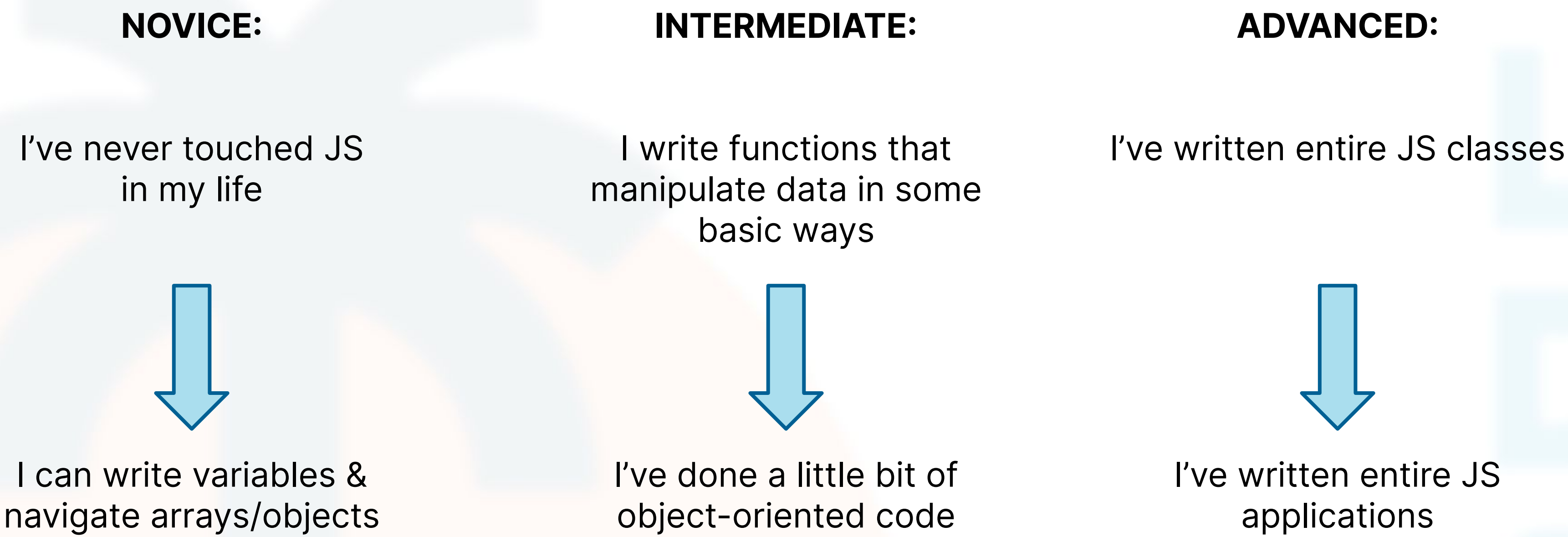
ADVANCED:

I’ve written entire PHP classes

↓

I’ve written entire PHP applications

Skill Level: JavaScript (Vanilla)



NOVICE:

I’ve never touched JS in my life

I can write variables & navigate arrays/objects

INTERMEDIATE:

I write functions that manipulate data in some basic ways

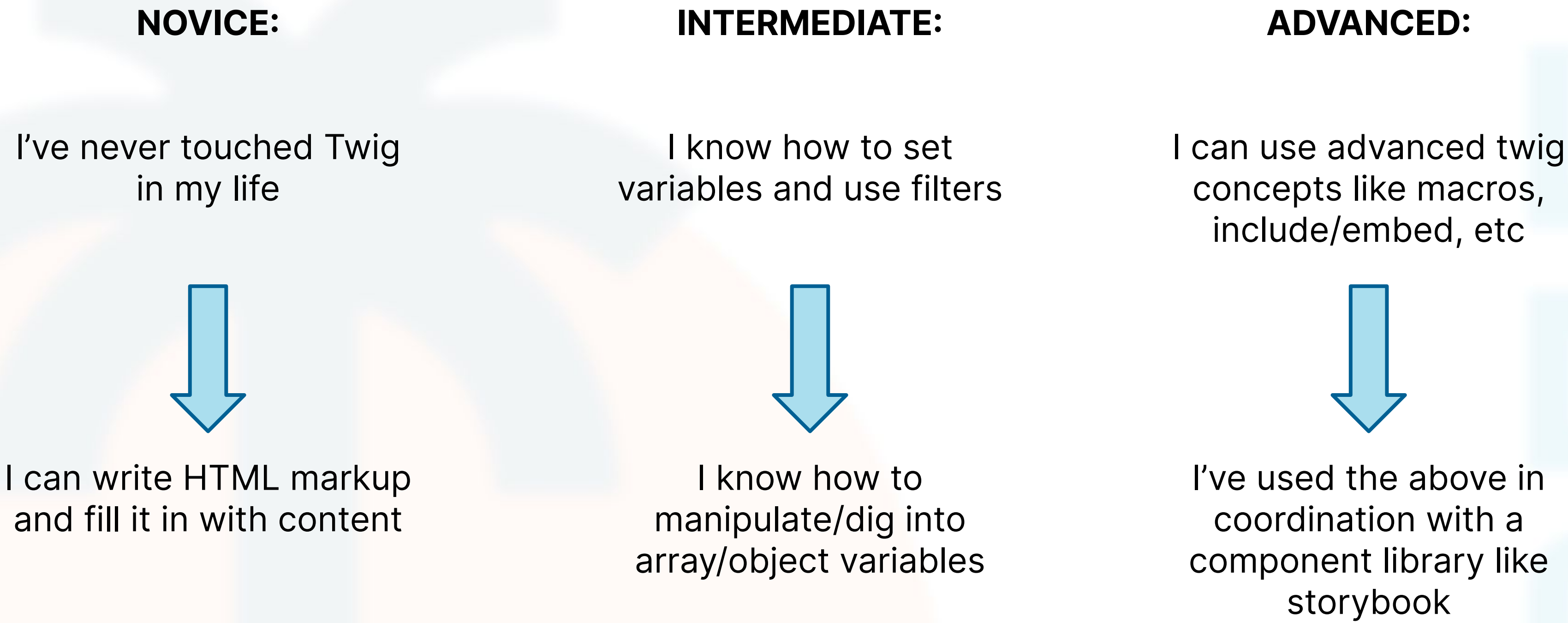
I’ve done a little bit of object-oriented code

ADVANCED:

I’ve written entire JS classes

I’ve written entire JS applications

Skill Level: Twig



NOVICE:

I've never touched Twig in my life

I can write HTML markup and fill it in with content

INTERMEDIATE:

I know how to set variables and use filters

I know how to manipulate/dig into array/object variables

ADVANCED:

I can use advanced twig concepts like macros, include/embed, etc

I've used the above in coordination with a component library like storybook

Skill Level: Drupal (Site Building & Admin)

NOVICE:

I’ve never touched Drupal in my life

↓

I’ve logged in and poked around, updated content, etc

INTERMEDIATE:

I’ve done some config like creating content types, views, etc

↓

I’ve made a basic site like a blog with a couple fancy features

ADVANCED:

I’ve built several sites on my own

↓

I’m Dries

But First…

- Source Code
  - github.com/andy-blum/fl-demo
- Tooling
  - VS Code on macOS (with a workspace already prepped!)
  - DDEV (ddev.readthedocs.io)
- Prefer your own setup?
  - That’s fine, but I might not be able to help you do some specific things
- Need Help?
  - Try getting it set up, but there are several of us here if you need help

training-start

But First...

- This repository is setup with git tags as checkpoints.
  - Cloning the repository puts you at the *end of this presentation*
  - `git checkout -b my-changes`
  - `git checkout training-start`

- If at any point something goes awry, just create a new feature branch from the tagged checkpoint

- **Please, Demo Gods, let everything go smoothly today**

What I’m Giving You To Start

- Composer-managed Drupal (v9.3.5)
- Drush (v11.0.4)
- DDEV (1.18 or 1.19) 👉 *need v19 if you’re not on docker desktop*
    - PHP 8.0

- Start it up!

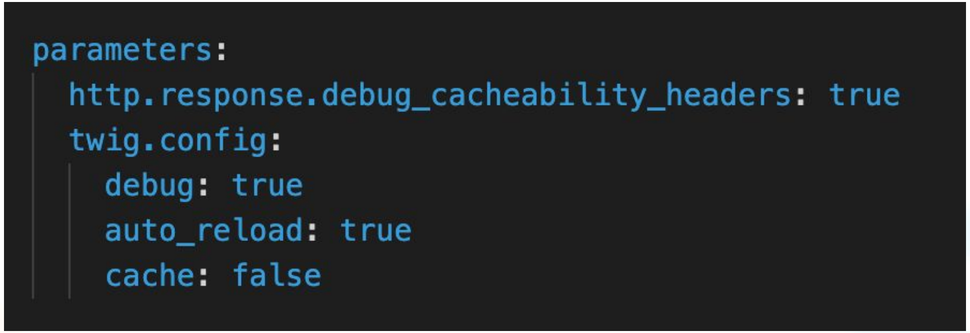
    ```
    git checkout training-start
    ddev start
    ddev composer install
    ddev drush si demo_umami
    ```

Get into developer mode

debugging-01

Configure development services

- Enabling twig debug messages
- Force twig files to re-compile every time the source code changes



```yaml
parameters:
  http.response.debug_cacheability_headers: true
  twig.config:
    debug: true
    auto_reload: true
    cache: false
```

`/web/sites/development.services.yml`

- You can see the documentation for all services in [default.services.yml]()

debugging-02

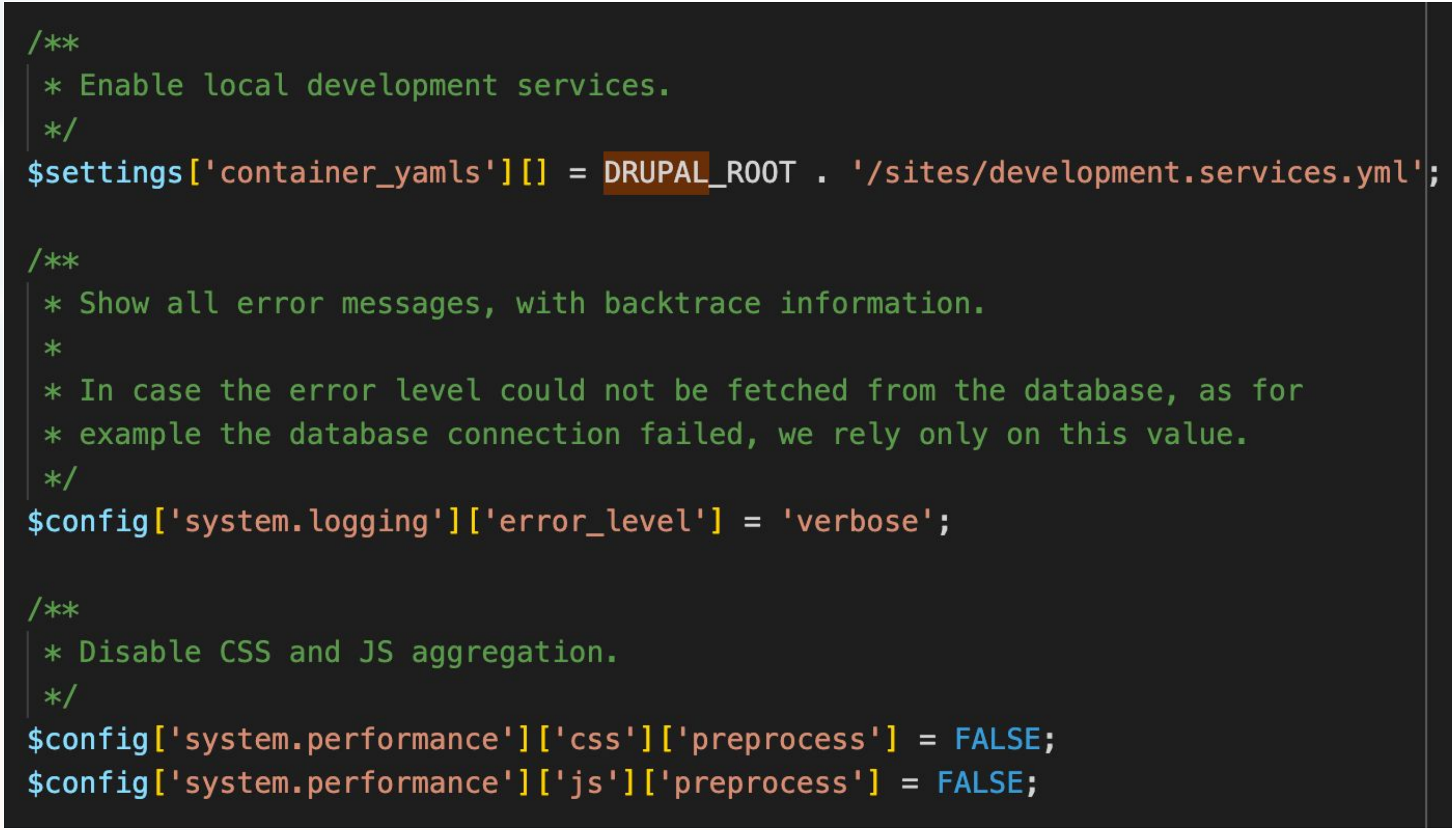
Configure local settings

- /web/sites/default/settings.local.php
- Duplicate from example.settings.local.php

- Enable local development services
- Show all error messages
- Disable css/js aggregation
- Disable render cache
- Disable internal page cache
- Disable dynamic page cache

debugging-02

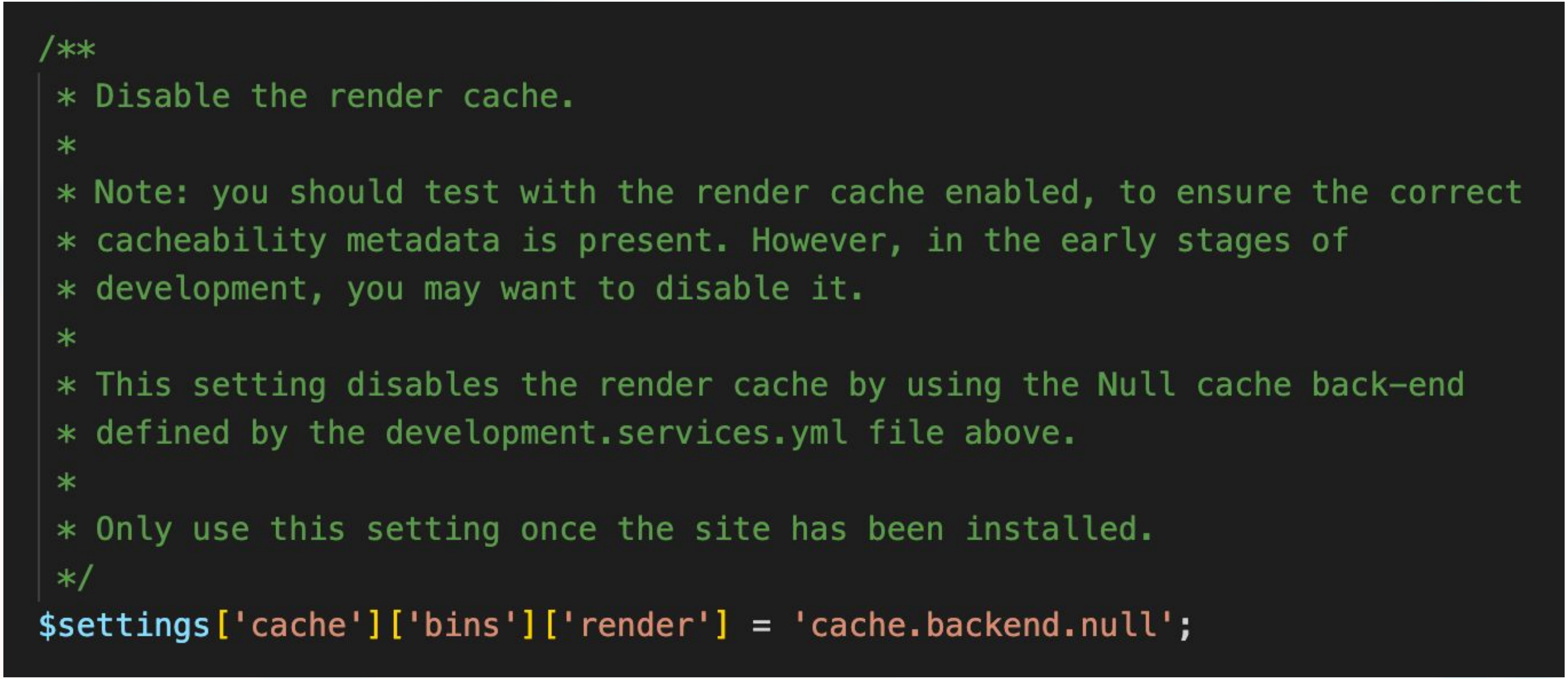
Configure local settings



```
/**
 * Enable local development services.
 */
$settings['container_yamls'][] = DRUPAL_ROOT . '/sites/development.services.yml';

/**
 * Show all error messages, with backtrace information.
 *
 * In case the error level could not be fetched from the database, as for
 * example the database connection failed, we rely only on this value.
 */
$config['system.logging']['error_level'] = 'verbose';

/**
 * Disable CSS and JS aggregation.
 */
$config['system.performance']['css']['preprocess'] = FALSE;
$config['system.performance']['js']['preprocess'] = FALSE;
```

debugging-02

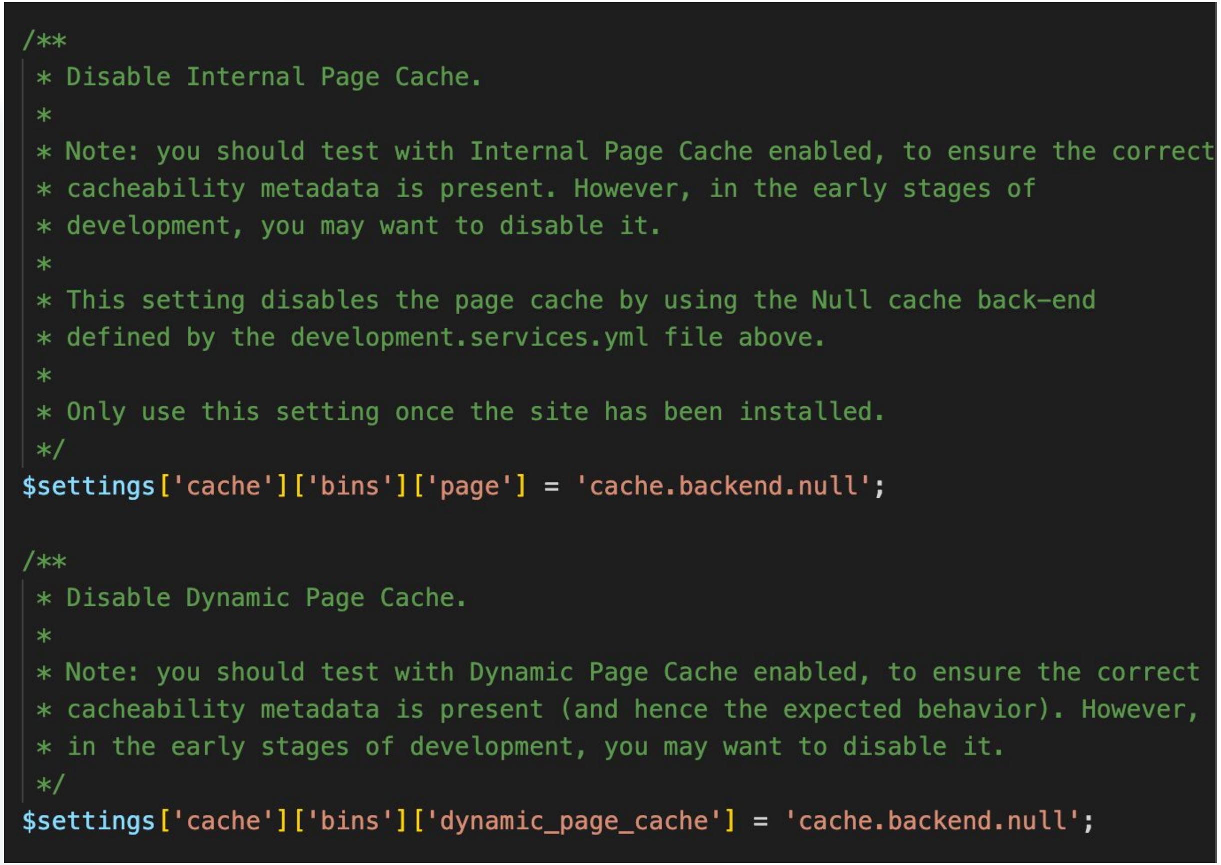
Configure local settings



```
/**
 * Disable the render cache.
 *
 * Note: you should test with the render cache enabled, to ensure the correct
 * cacheability metadata is present. However, in the early stages of
 * development, you may want to disable it.
 *
 * This setting disables the render cache by using the Null cache back-end
 * defined by the development.services.yml file above.
 *
 * Only use this setting once the site has been installed.
 */
$settings['cache']['bins']['render'] = 'cache.backend.null';
```

debugging-02

Configure local settings



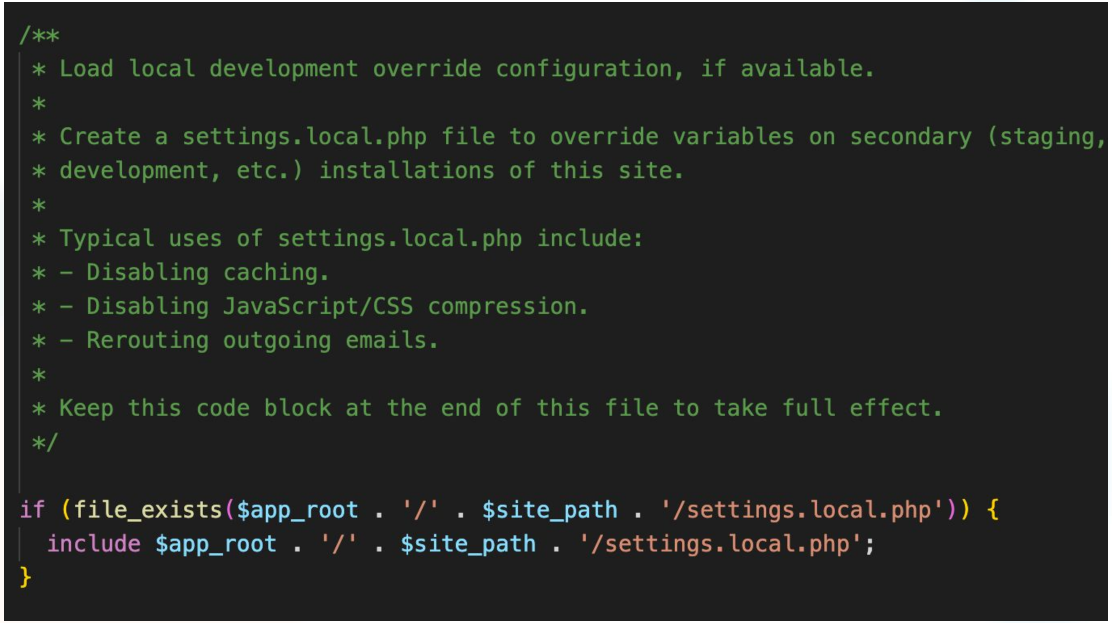
```
/**
 * Disable Internal Page Cache.
 *
 * Note: you should test with Internal Page Cache enabled, to ensure the correct
 * cacheability metadata is present. However, in the early stages of
 * development, you may want to disable it.
 *
 * This setting disables the page cache by using the Null cache back-end
 * defined by the development.services.yml file above.
 *
 * Only use this setting once the site has been installed.
 */
$settings['cache']['bins']['page'] = 'cache.backend.null';

/**
 * Disable Dynamic Page Cache.
 *
 * Note: you should test with Dynamic Page Cache enabled, to ensure the correct
 * cacheability metadata is present (and hence the expected behavior). However,
 * in the early stages of development, you may want to disable it.
 */
$settings['cache']['bins']['dynamic_page_cache'] = 'cache.backend.null';
```

debugging-03

Include our local settings

- /web/sites/default/settings.php
- Include local settings

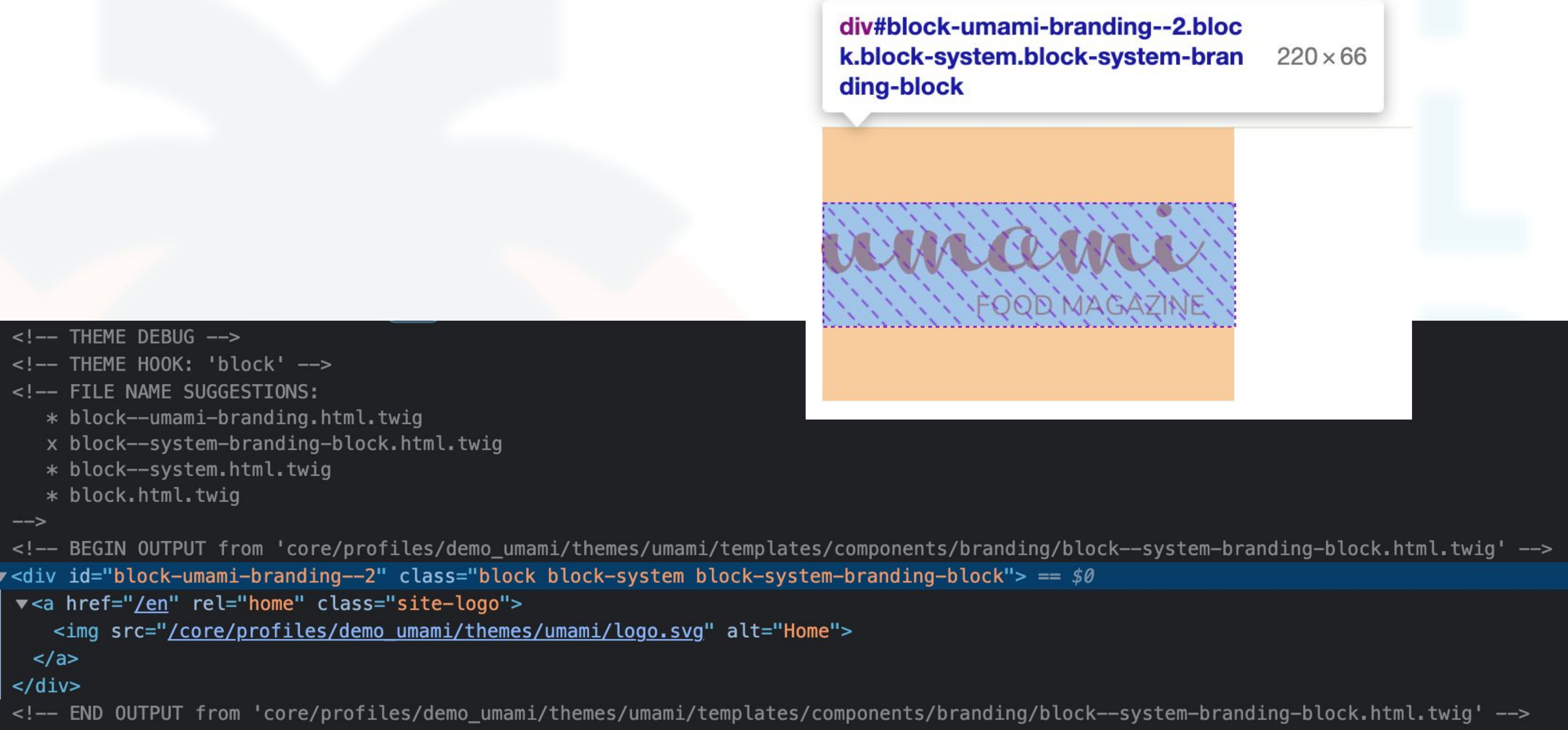
- Clear cache for all these changes to take effect



```php
/**
 * Load local development override configuration, if available.
 *
 * Create a settings.local.php file to override variables on secondary (staging,
 * development, etc.) installations of this site.
 *
 * Typical uses of settings.local.php include:
 * - Disabling caching.
 * - Disabling JavaScript/CSS compression.
 * - Rerouting outgoing emails.
 *
 * Keep this code block at the end of this file to take full effect.
 */

if (file_exists($app_root . '/' . $site_path . '/settings.local.php')) {
  include $app_root . '/' . $site_path . '/settings.local.php';
}
```

debugging-03

Hints & Help in our rendered markup!



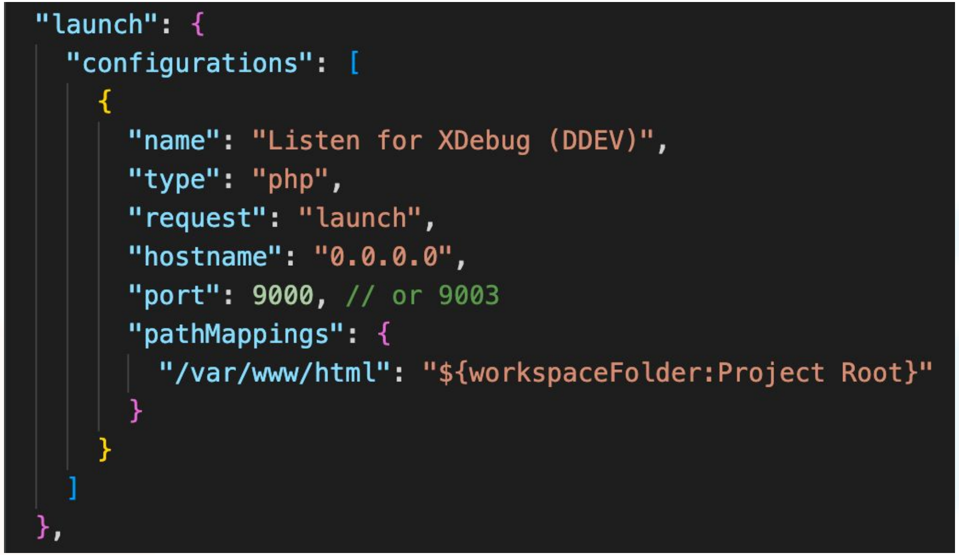
div#block-umami-branding--2.block.block-system.block-system-branding-block 220 × 66

```
<!-- THEME DEBUG -->
<!-- THEME HOOK: 'block' -->
<!-- FILE NAME SUGGESTIONS:
   * block--umami-branding.html.twig
   x block--system-branding-block.html.twig
   * block--system.html.twig
   * block.html.twig
-->
<!-- BEGIN OUTPUT from 'core/profiles/demo_umami/themes/umami/templates/components/branding/block--system-branding-block.html.twig' -->
<div id="block-umami-branding--2" class="block block-system block-system-branding-block"> == $0
 <a href="/en" rel="home" class="site-logo">
    <img src="/core/profiles/demo_umami/themes/umami/logo.svg" alt="Home">
  </a>
</div>
<!-- END OUTPUT from 'core/profiles/demo_umami/themes/umami/templates/components/branding/block--system-branding-block.html.twig' -->
```

debugging-04

Add PHP Step Debugging

- Add xDebug to DDEV
  - If you’re not using DDEV or VS code you’re on your own here
- Allows step debugging PHP
- /fl-demo.code-workspace
- Install PHP Debug extension
- xDebug config changed between ddev 1.18 and 1.19
- **We’ll skip using this for now**



```
"launch": {
  "configurations": [
    {
      "name": "Listen for XDebug (DDEV)",
      "type": "php",
      "request": "launch",
      "hostname": "0.0.0.0",
      "port": 9000, // or 9003
      "pathMappings": {
        "/var/www/html": "${workspaceFolder:Project Root}"
      }
    }
  ]
},
```

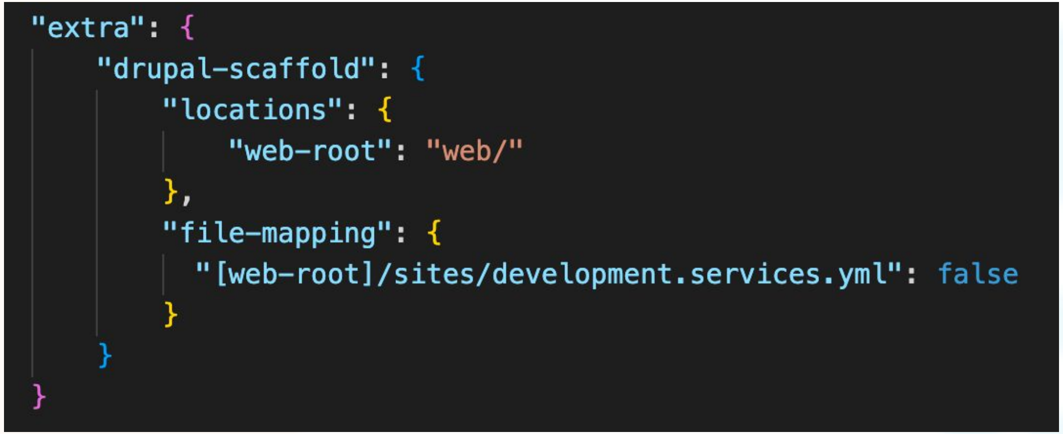
debugging-05

Add helpful debugging print outs

- Add the devel module
  - `ddev composer require drupal/devel`
  - `ddev drush en devel devel_generate`

- Now look at development.services.yml
  - “Composer Scaffold”
  - [Drupal issue #3094699]()
  - [Scaffolding docs]()

- Prevent scaffolding from Overwriting this file



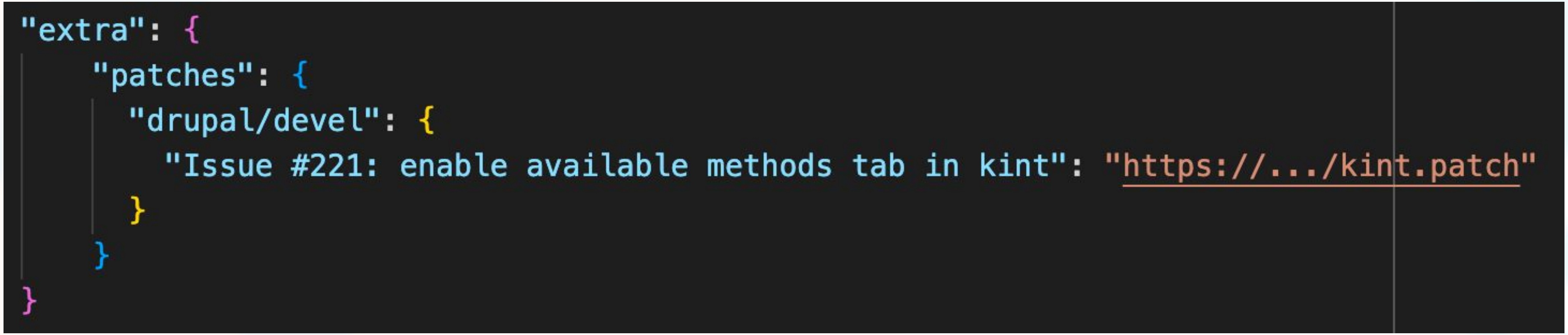
```
"extra": {
    "drupal-scaffold": {
        "locations": {
            "web-root": "web/"
        },
        "file-mapping": {
          "[web-root]/sites/development.services.yml": false
        }
    }
}
```

debugging-05

My Favorite: Kint

- OPTIONAL
  - Add kint as a vardumper
  - With a patch, displays methods & object iterators

- **We’ll skip using this for now**



```
"extra": {
    "patches": {
      "drupal/devel": {
        "Issue #221: enable available methods tab in kint": "https://.../kint.patch"
      }
    }
}
```

Custom Theme Settings

add-theme

Duplicate core’s umami theme

- We don’t want to modify the core theme - anytime we update/install with composer our changes will be overwritten

- /web/core/profiles/demo_umami/themes/umami
- /web/themes/custom/foomami

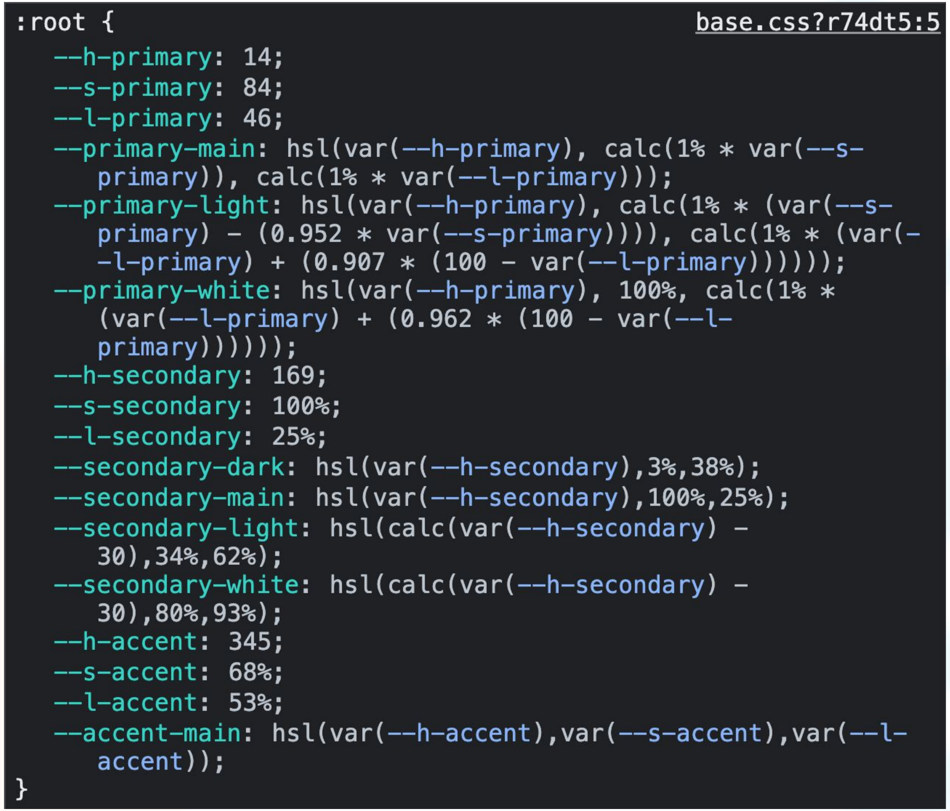
- I’m making some CSS changes as well that we will briefly cover but don’t need to do together.

- Just checkout the tagged commit
  - `git checkout add-theme`

Enable our theme!

- Enable the `foomami` theme & set to default
- Enable `claro` theme & make it the admin theme *(optional)*
- Disable `umami` *(optional)*

- Only changes I made to the theme besides renaming
  - Add `core_version_requirement` to info.yml
  - Consolidate & convert colors to HSL



```
:root {                                                  base.css?r74dt5:5
   --h-primary: 14;
   --s-primary: 84;
   --l-primary: 46;
   --primary-main: hsl(var(--h-primary), calc(1% * var(--s-
      primary)), calc(1% * var(--l-primary)));
   --primary-light: hsl(var(--h-primary), calc(1% * (var(--s-
      primary) - (0.952 * var(--s-primary)))), calc(1% * (var(-
      -l-primary) + (0.907 * (100 - var(--l-primary))))));
   --primary-white: hsl(var(--h-primary), 100%, calc(1% *
      (var(--l-primary) + (0.962 * (100 - var(--l-
      primary))))));
   --h-secondary: 169;
   --s-secondary: 100%;
   --l-secondary: 25%;
   --secondary-dark: hsl(var(--h-secondary),3%,38%);
   --secondary-main: hsl(var(--h-secondary),100%,25%);
   --secondary-light: hsl(calc(var(--h-secondary) -
      30),34%,62%);
   --secondary-white: hsl(calc(var(--h-secondary) -
      30),80%,93%);
   --h-accent: 345;
   --s-accent: 68%;
   --l-accent: 53%;
   --accent-main: hsl(var(--h-accent),var(--s-accent),var(--l-
      accent));
}
```

Add custom settings to our theme

- http://.../admin/appearance/settings/foomami
  - Starts with just the global settings
  - We can add custom controls here!

- Add theme-settings.php to our theme root
- Add hook_form_FORM_ID_alter function
  - Form_id: system_theme_settings
  - Use my VS Code Extension
  - WTF is a hook?

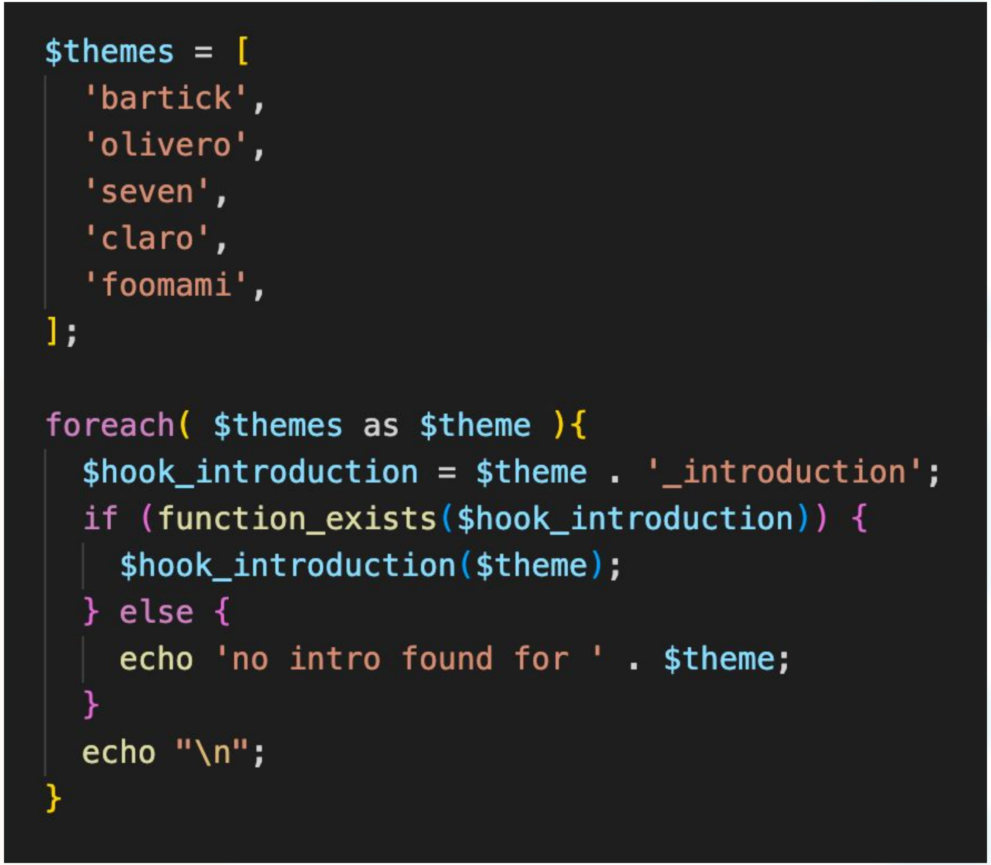
Hooks: Explained

- Hooks are specific functions that let us interject functionality into a program
- Three main types for themes
  - Form Alters
  - Template Preprocesses
  - Theme Suggestions
- We'll use the first two in this stage

Hooks: Explained

- Hooks are specific functions that let us interject functionality into a program

- Three main types for themes
  - Template Preprocesses
  - Form Alters
  - Theme Suggestions

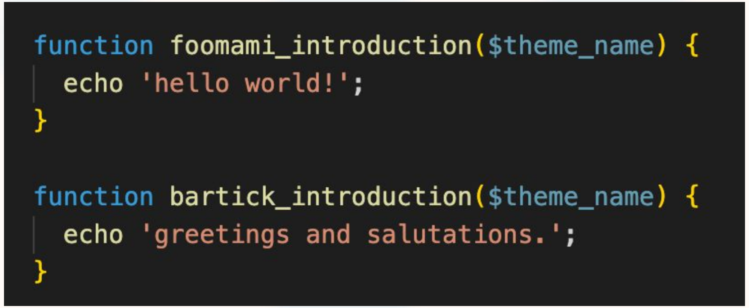


```php
$themes = [
  'bartick',
  'olivero',
  'seven',
  'claro',
  'foomami',
];

foreach( $themes as $theme ){
  $hook_introduction = $theme . '_introduction';
  if (function_exists($hook_introduction)) {
    $hook_introduction($theme);
  } else {
    echo 'no intro found for ' . $theme;
  }
  echo "\n";
}
```

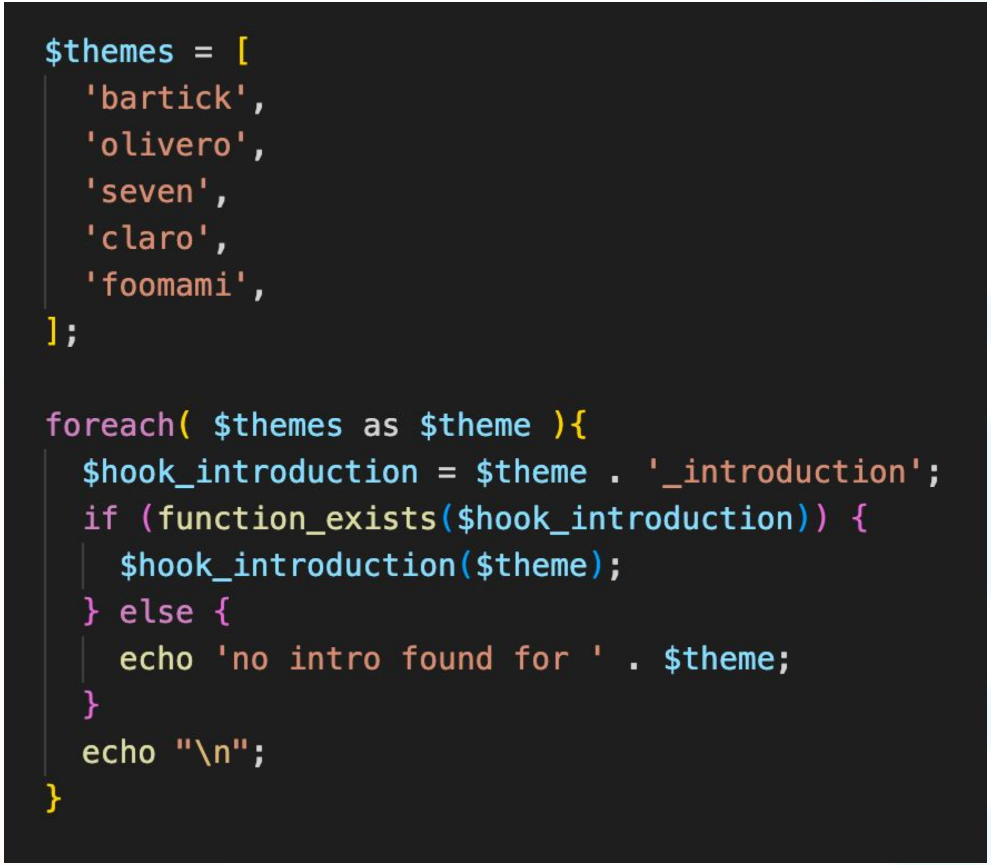
Hooks: Explained

- Hooks are specific functions that let us interject functionality into a program

- Three main types for themes
    - Template Preprocesses
    - Form Alters
    - Theme Suggestions



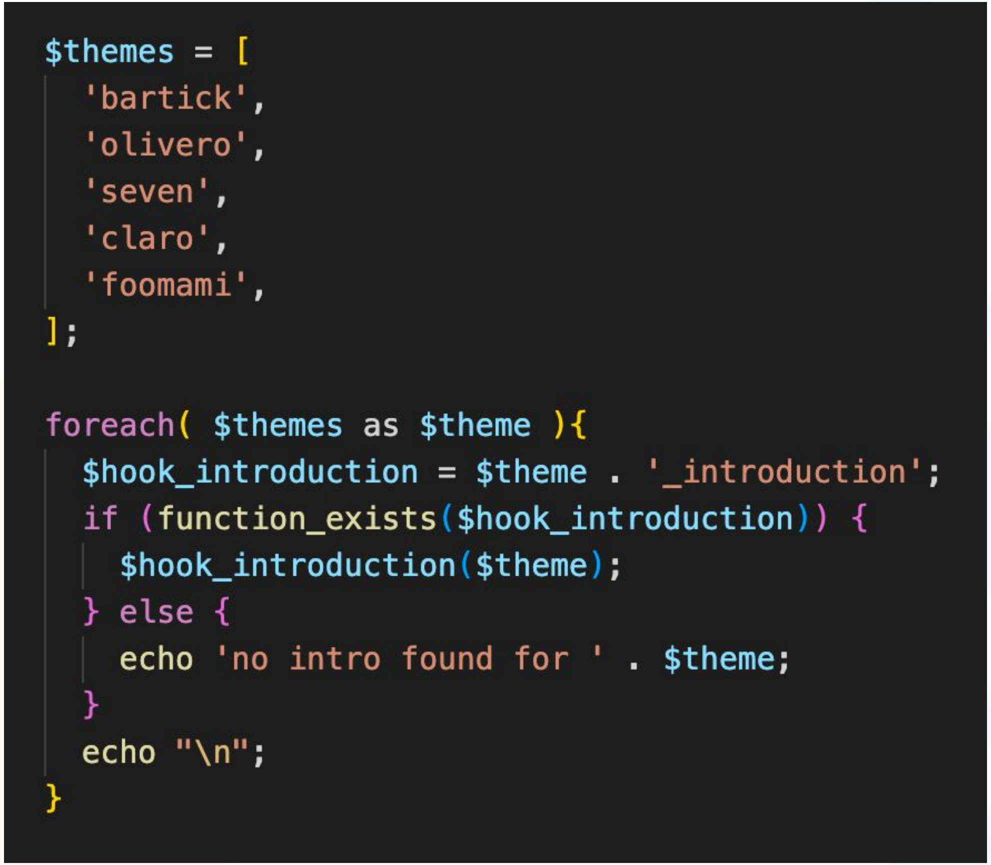
```
function foomami_introduction($theme_name) {
  echo 'hello world!';
}

function bartick_introduction($theme_name) {
  echo 'greetings and salutations.';
}
```



```
$themes = [
  'bartick',
  'olivero',
  'seven',
  'claro',
  'foomami',
];

foreach( $themes as $theme ){
  $hook_introduction = $theme . '_introduction';
  if (function_exists($hook_introduction)) {
    $hook_introduction($theme);
  } else {
    echo 'no intro found for ' . $theme;
  }
  echo "\n";
}
```

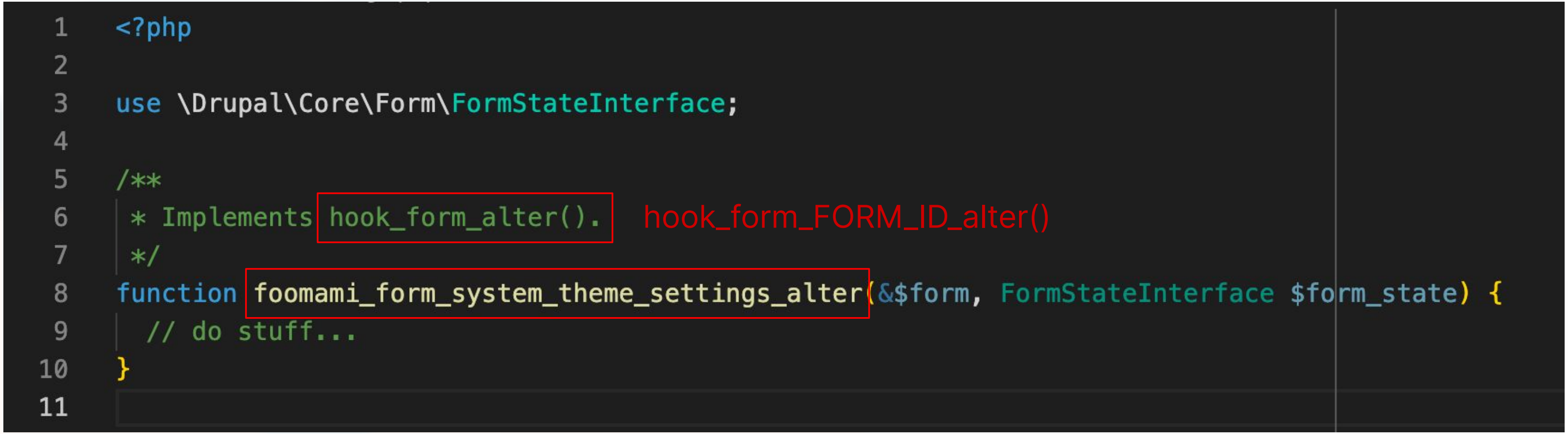
Hooks: Explained

- Theme hooks only run on the active theme and its base theme(s) if there are any, regardless of how many are installed
- Module hooks run in all themes for all installed modules
- Module hooks run first, then theme hooks
- Hooks run alphabetically
  - Unless we change that behavior



```
$themes = [
  'bartick',
  'olivero',
  'seven',
  'claro',
  'foomami',
];

foreach( $themes as $theme ){
  $hook_introduction = $theme . '_introduction';
  if (function_exists($hook_introduction)) {
    $hook_introduction($theme);
  } else {
    echo 'no intro found for ' . $theme;
  }
  echo "\n";
}
```

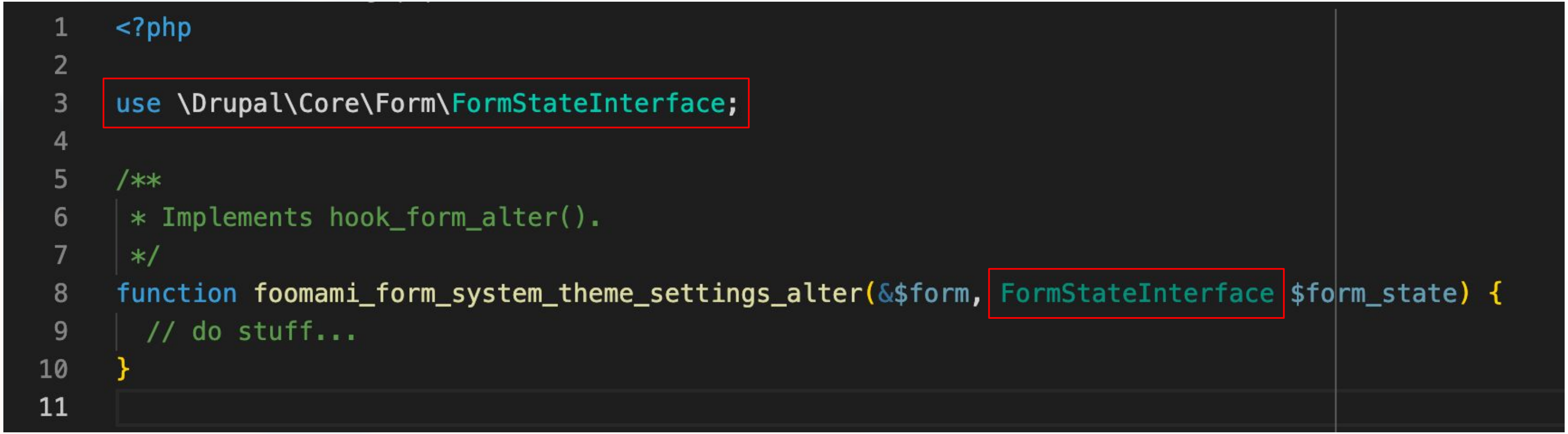
Back to our theme settings!



```php
<?php

use \Drupal\Core\Form\FormStateInterface;

/**
 * Implements hook_form_alter().
 */
function foomami_form_system_theme_settings_alter(&$form, FormStateInterface $form_state) {
  // do stuff...
}

```

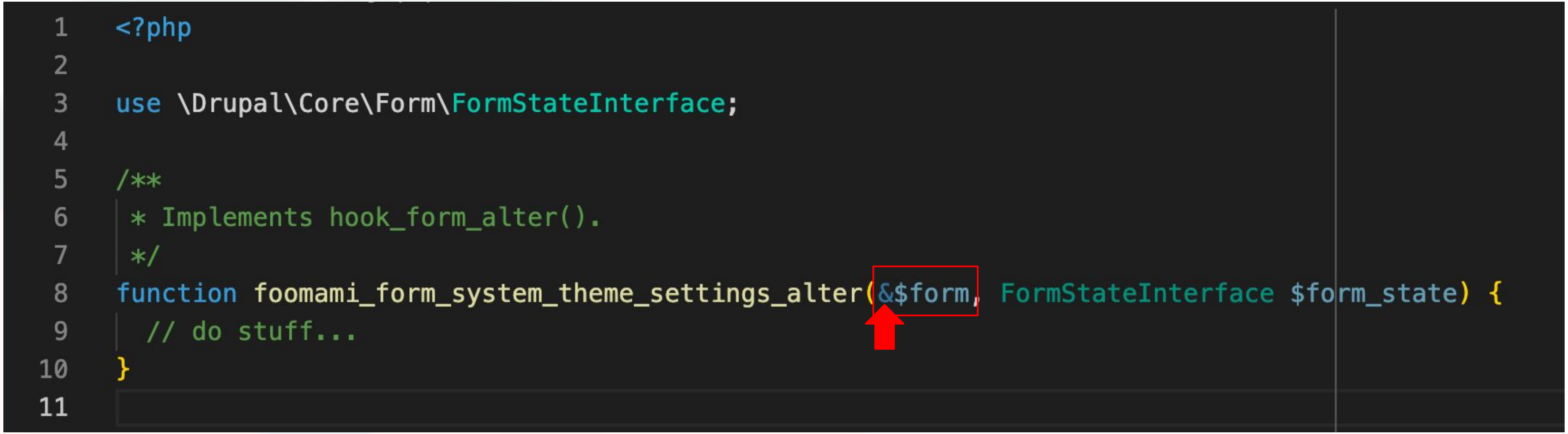
hook_form_FORM_ID_alter()

Back to our theme settings!



```php
<?php

use \Drupal\Core\Form\FormStateInterface;

/**
 * Implements hook_form_alter().
 */
function foomami_form_system_theme_settings_alter(&$form, FormStateInterface $form_state) {
  // do stuff...
}

```

Back to our theme settings!

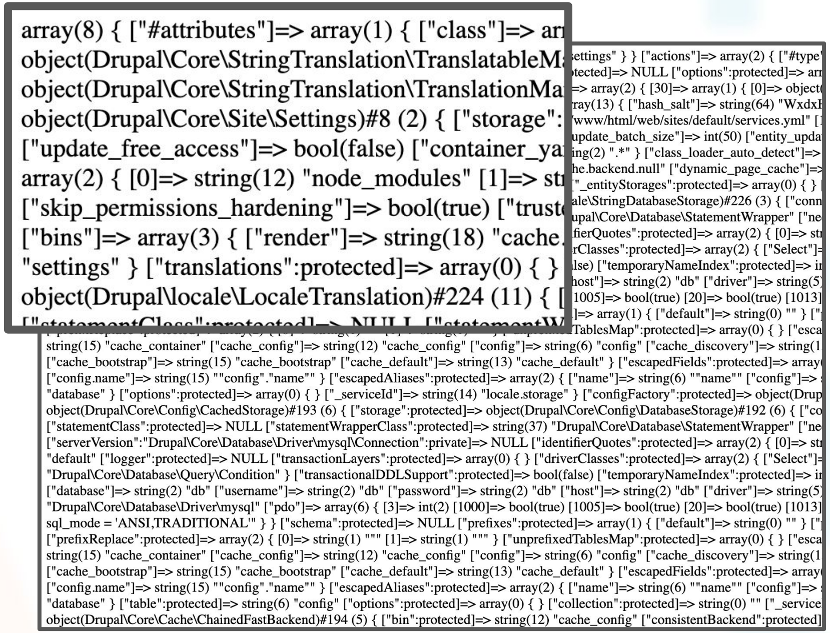


```php
<?php

use \Drupal\Core\Form\FormStateInterface;

/**
 * Implements hook_form_alter().
 */
function foomami_form_system_theme_settings_alter(&$form, FormStateInterface $form_state) {
  // do stuff...
}

```

What do we have access to?

- Lets find out what $form has in it.

- var_dump($form);
  - Baked into PHP
  - Looks better if you wrap with <pre> tags
  - If you have xDebug running (we don't yet) it will also improve the printout
  - A lot of variables we will work with are *massive*. Var_dump may easily run out of memory or take longer than your browser wants to wait on a request to complete
  - **Generally avoid**



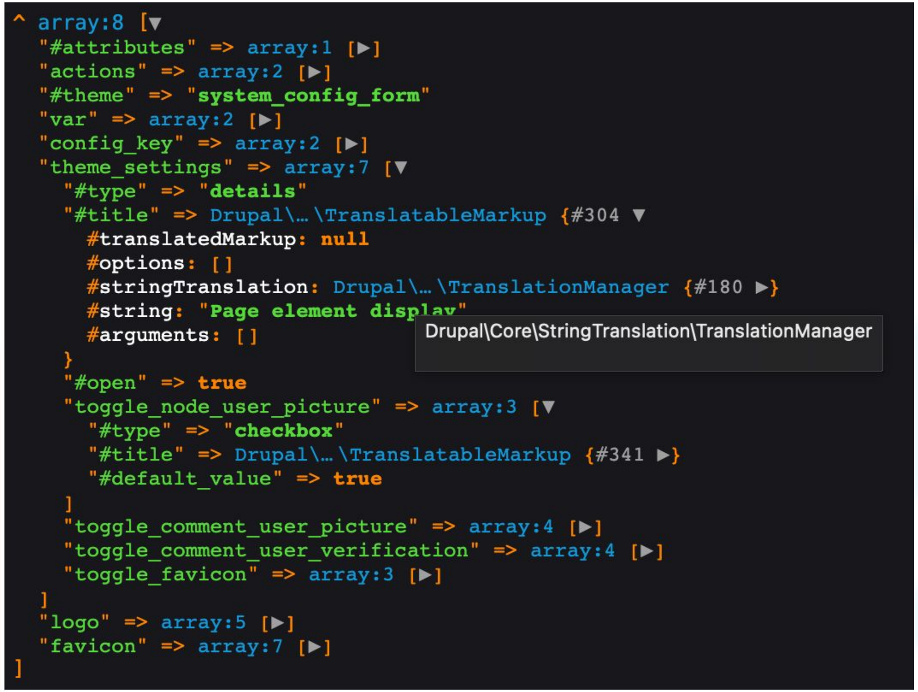
```
array(8) { ["#attributes"]=> array(1) { ["class"]=> ar
object(Drupal\Core\StringTranslation\TranslatableM
object(Drupal\Core\StringTranslation\TranslationMa
object(Drupal\Core\Site\Settings)#8 (2) { ["storage":
["update_free_access"]=> bool(false) ["container_ya
array(2) { [0]=> string(12) "node_modules" [1]=> str
["skip_permissions_hardening"]=> bool(true) ["truste
["bins"]=> array(3) { ["render"]=> string(18) "cache.
"settings" } ["translations":protected]=> array(0) { }
object(Drupal\locale\LocaleTranslation)#224 (11) { [
["statementClass":protected]=> NULL ["statementW
```

```
ettings" } } ["actions"]=> array(2) { ["#type"
tected]=> NULL ["options":protected]=> arr
=> array(2) { [30]=> array(1) { [0]=> object(
ray(13) { ["hash_salt"]=> string(64) "WxdxH
www/html/web/sites/default/services.yml" [
update_batch_size"]=> int(50) ["entity_upda
ing(2) ".*" } ["class_loader_auto_detect"]=>
he.backend.null" ["dynamic_page_cache"]=>
"_entityStorages":protected]=> array(0) { }
ale\StringDatabaseStorage)#226 (3) { ["conn
upal\Core\Database\StatementWrapper" ["ne
fierQuotes":protected]=> array(2) { [0]=> str
rClasses":protected]=> array(2) { ["Select"]=
lse) ["temporaryNameIndex":protected]=> in
host"]=> string(2) "db" ["driver"]=> string(5)
1005]=> bool(true) [20]=> bool(true) [1013]
]=> array(1) { ["default"]=> string(0) "" } ["
TablesMap":protected]=> array(0) { } ["esca
string(15) "cache_container" ["cache_config"]=> string(12) "cache_config" ["config"]=> string(6) "config" ["cache_discovery"]=> string(1
["cache_bootstrap"]=> string(15) "cache_bootstrap" ["cache_default"]=> string(13) "cache_default" } ["escapedFields":protected]=> array(
["config.name"]=> string(15) ""config"."name"" } ["escapedAliases":protected]=> array(2) { ["name"]=> string(6) ""name"" ["config"]=> s
"database" } ["options":protected]=> array(0) { } ["_serviceId"]=> string(14) "locale.storage" } ["configFactory":protected]=> object(Drup
object(Drupal\Core\Config\CachedStorage)#193 (6) { ["storage":protected]=> object(Drupal\Core\Config\DatabaseStorage)#192 (6) { ["co
["statementClass":protected]=> NULL ["statementWrapperClass":protected]=> string(37) "Drupal\Core\Database\StatementWrapper" ["ne
["serverVersion":"Drupal\Core\Database\Driver\mysql\Connection":private]=> NULL ["identifierQuotes":protected]=> array(2) { [0]=> str
"default" ["logger":protected]=> NULL ["transactionLayers":protected]=> array(0) { } ["driverClasses":protected]=> array(2) { ["Select"]=
"Drupal\Core\Database\Query\Condition" } ["transactionalDDLSupport":protected]=> bool(false) ["temporaryNameIndex":protected]=> in
["database"]=> string(2) "db" ["username"]=> string(2) "db" ["password"]=> string(2) "db" ["host"]=> string(2) "db" ["driver"]=> string(5)
"Drupal\Core\Database\Driver\mysql" ["pdo"]=> array(6) { [3]=> int(2) [1000]=> bool(true) [1005]=> bool(true) [20]=> bool(true) [1013]
sql_mode = 'ANSI,TRADITIONAL'" } } ["schema":protected]=> NULL ["prefixes":protected]=> array(1) { ["default"]=> string(0) "" } ["
["prefixReplace":protected]=> array(2) { [0]=> string(1) """ [1]=> string(1) """ } ["unprefixedTablesMap":protected]=> array(0) { } ["esca
string(15) "cache_container" ["cache_config"]=> string(12) "cache_config" ["config"]=> string(6) "config" ["cache_discovery"]=> string(1
["cache_bootstrap"]=> string(15) "cache_bootstrap" ["cache_default"]=> string(13) "cache_default" } ["escapedFields":protected]=> array(
["config.name"]=> string(15) ""config"."name"" } ["escapedAliases":protected]=> array(2) { ["name"]=> string(6) ""name"" ["config"]=> s
"database" } ["table":protected]=> string(6) "config" ["options":protected]=> array(0) { } ["collection":protected]=> string(0) "" ["_service
object(Drupal\Core\Cache\ChainedFastBackend)#194 (5) { ["bin":protected]=> string(12) "cache_config" ["consistentBackend":protected]
```

What do we have access to?

- Lets find out what $form has in it.

- dump($form);
  - Provided from Symfony
  - Arrays & objects are collapsed by default, making it easy to explore or find specific information
  - Color coding and types give additional context to the things you're looking at
  - Provides a considerable amount of depth without running out of memory
  - **Solid Choice for variable exploration**

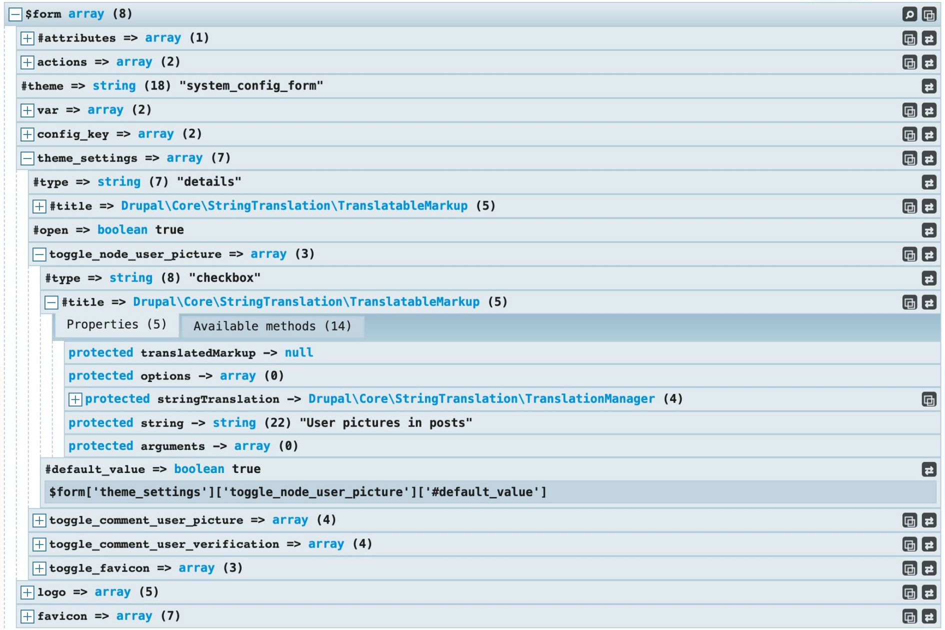


```
^ array:8 [▼
  "#attributes" => array:1 [▶]
  "actions" => array:2 [▶]
  "#theme" => "system_config_form"
  "var" => array:2 [▶]
  "config_key" => array:2 [▶]
  "theme_settings" => array:7 [▼
    "#type" => "details"
    "#title" => Drupal\…\TranslatableMarkup {#304 ▼
      #translatedMarkup: null
      #options: []
      #stringTranslation: Drupal\…\TranslationManager {#180 ▶}
      #string: "Page element display"
      #arguments: []
    }
    "#open" => true
    "toggle_node_user_picture" => array:3 [▼
      "#type" => "checkbox"
      "#title" => Drupal\…\TranslatableMarkup {#341 ▶}
      "#default_value" => true
    ]
    "toggle_comment_user_picture" => array:4 [▶]
    "toggle_comment_user_verification" => array:4 [▶]
    "toggle_favicon" => array:3 [▶]
  ]
  "logo" => array:5 [▶]
  "favicon" => array:7 [▶]
]
```

What do we have access to?

- Lets find out what $form has in it.

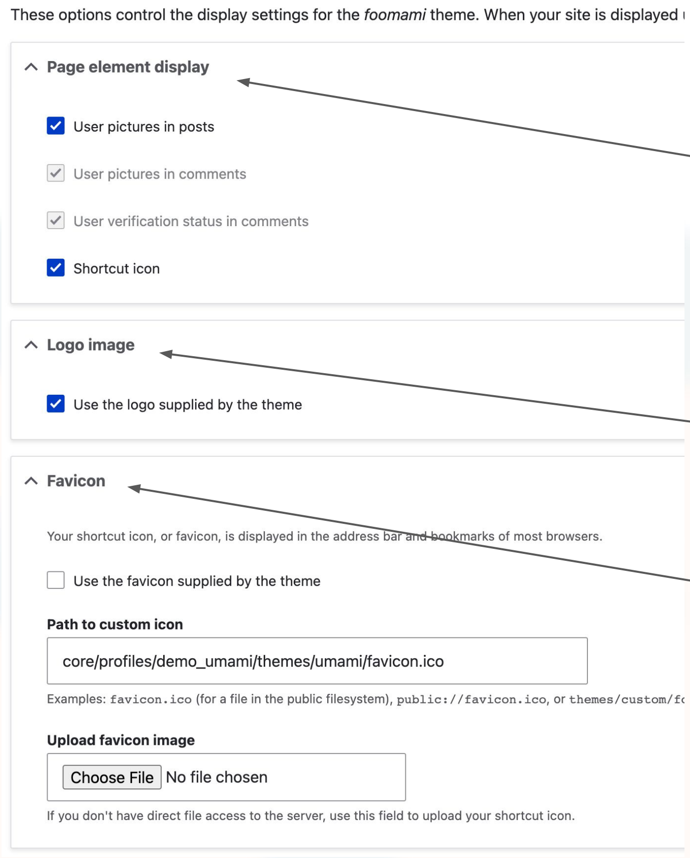
- kint($form);
  - Requires kint-php/kint
  - Similar to Symfony’s var dumper
  - Also provides type information
  - With the patch we added we can also see objects’ available methods
  - Buttons on right side add functionality
    - Copy access path
    - Open in new window
  - Can run out of memory, though configurable to limit depth



```
$form array (8)
 #attributes => array (1)
 actions => array (2)
 #theme => string (18) "system_config_form"
 var => array (2)
 config_key => array (2)
 theme_settings => array (7)
  #type => string (7) "details"
  #title => Drupal\Core\StringTranslation\TranslatableMarkup (5)
  #open => boolean true
  toggle_node_user_picture => array (3)
   #type => string (8) "checkbox"
   #title => Drupal\Core\StringTranslation\TranslatableMarkup (5)
    Properties (5)   Available methods (14)
    protected translatedMarkup -> null
    protected options -> array (0)
    protected stringTranslation -> Drupal\Core\StringTranslation\TranslationManager (4)
    protected string -> string (22) "User pictures in posts"
    protected arguments -> array (0)
   #default_value => boolean true
   $form['theme_settings']['toggle_node_user_picture']['#default_value']
  toggle_comment_user_picture => array (4)
  toggle_comment_user_verification => array (4)
  toggle_favicon => array (3)
 logo => array (5)
 favicon => array (7)
```

My personal favorite

What do we have access to?



These options control the display settings for the *foomami* theme. When your site is displayed

Page element display

☑ User pictures in posts

☑ User pictures in comments

☑ User verification status in comments

☑ Shortcut icon

Logo image

☑ Use the logo supplied by the theme

Favicon

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

☐ Use the favicon supplied by the theme

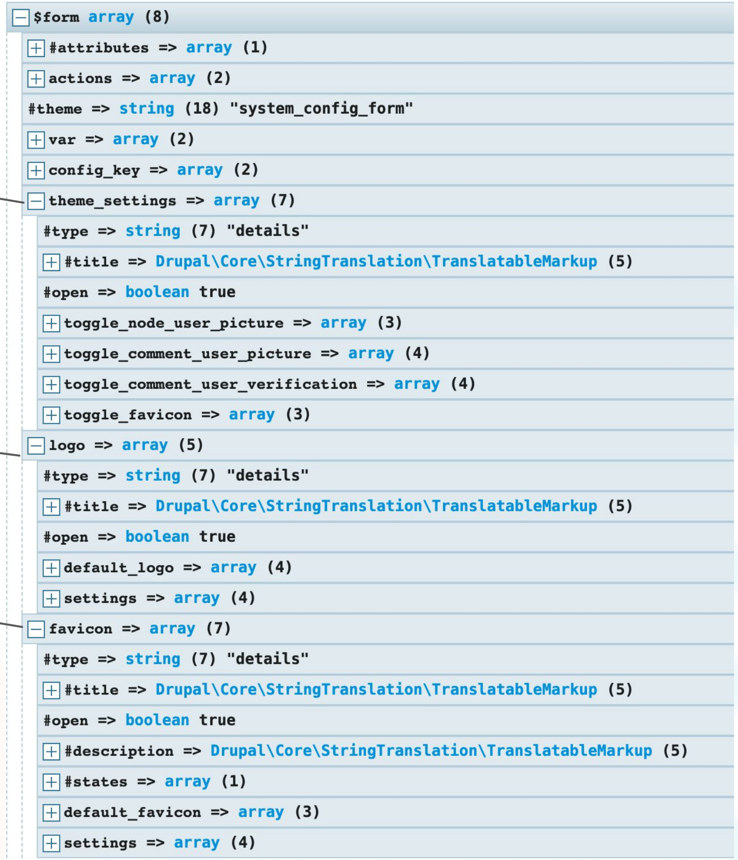
Path to custom icon

core/profiles/demo_umami/themes/umami/favicon.ico

Examples: `favicon.ico` (for a file in the public filesystem), `public://favicon.ico`, or `themes/custom/fc`

Upload favicon image

If you don't have direct file access to the server, use this field to upload your shortcut icon.



```
$form array (8)
  #attributes => array (1)
  actions => array (2)
  #theme => string (18) "system_config_form"
  var => array (2)
  config_key => array (2)
  theme_settings => array (7)
    #type => string (7) "details"
    #title => Drupal\Core\StringTranslation\TranslatableMarkup (5)
    #open => boolean true
    toggle_node_user_picture => array (3)
    toggle_comment_user_picture => array (4)
    toggle_comment_user_verification => array (4)
    toggle_favicon => array (3)
  logo => array (5)
    #type => string (7) "details"
    #title => Drupal\Core\StringTranslation\TranslatableMarkup (5)
    #open => boolean true
    default_logo => array (4)
    settings => array (4)
  favicon => array (7)
    #type => string (7) "details"
    #title => Drupal\Core\StringTranslation\TranslatableMarkup (5)
    #open => boolean true
    #description => Drupal\Core\StringTranslation\TranslatableMarkup (5)
    #states => array (1)
    default_favicon => array (3)
    settings => array (4)
```

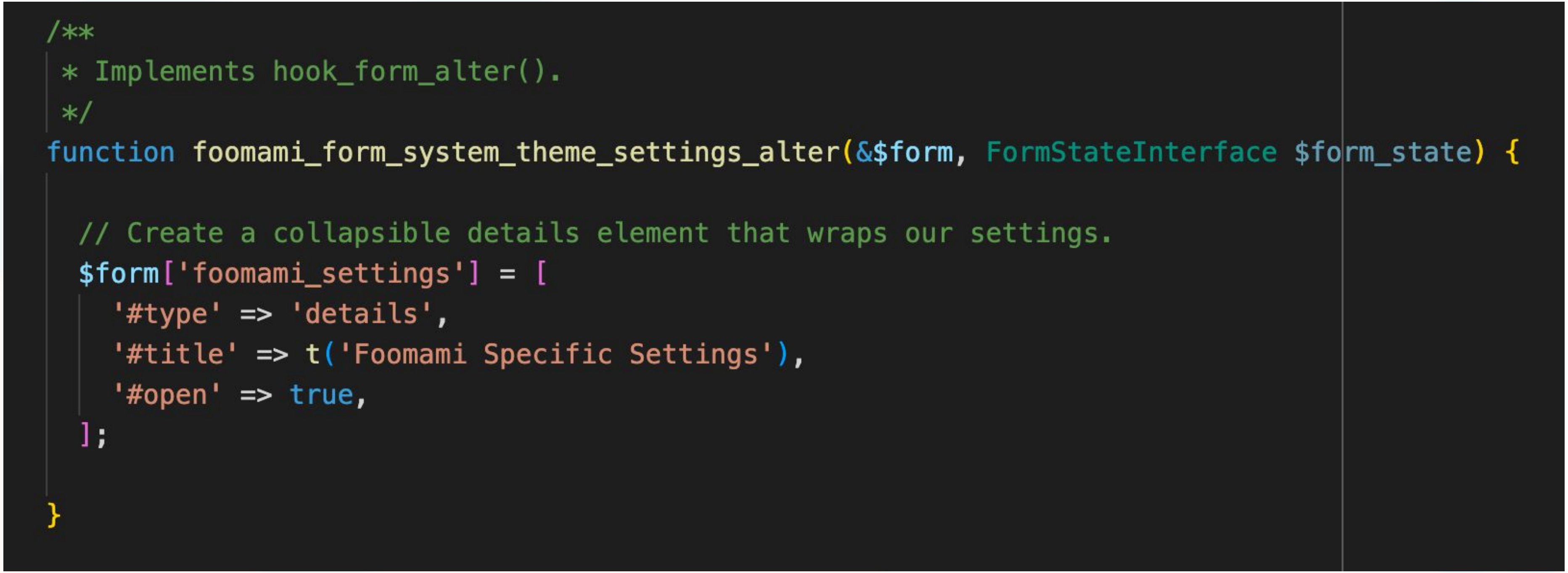
Let’s mimic what’s already there

- We need to add Form Render Elements
    - Descriptive documentation
    - Full List of all Elements

- Let’s start by making a collapsible panel to wrap all of the form fields we’ll add.
    - Details element
    - Call it “Foomami Specific Settings”
    - Open by default

theme-settings-01

Creating a wrapper for our fields

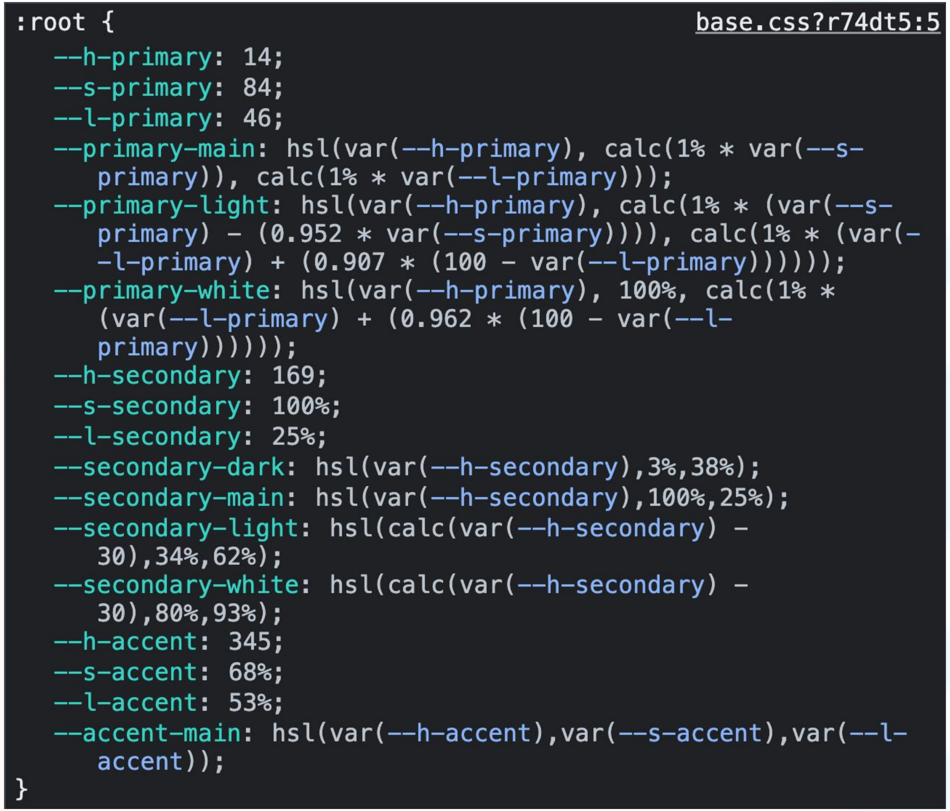


```php
/**
 * Implements hook_form_alter().
 */
function foomami_form_system_theme_settings_alter(&$form, FormStateInterface $form_state) {

  // Create a collapsible details element that wraps our settings.
  $form['foomami_settings'] = [
    '#type' => 'details',
    '#title' => t('Foomami Specific Settings'),
    '#open' => true,
  ];

}
```

What fields do we need?

- I converted CSS colors from hex to HSL to allow site admins to customize colors
  - 3 colors/color families
    - primary, secondary, accent
  - 3 values per color
    - hue, saturation, lightness

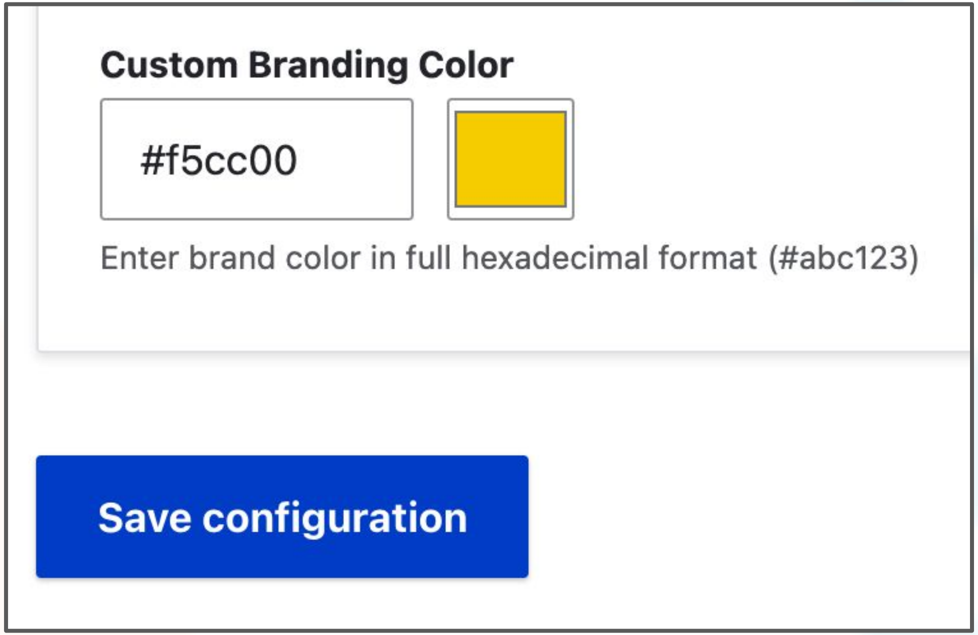
- How could we set this up?
  - *Pause for audience participation*



```
:root {                                                   base.css?r74dt5:5
   --h-primary: 14;
   --s-primary: 84;
   --l-primary: 46;
   --primary-main: hsl(var(--h-primary), calc(1% * var(--s-
      primary)), calc(1% * var(--l-primary)));
   --primary-light: hsl(var(--h-primary), calc(1% * (var(--s-
      primary) - (0.952 * var(--s-primary)))), calc(1% * (var(-
      -l-primary) + (0.907 * (100 - var(--l-primary))))));
   --primary-white: hsl(var(--h-primary), 100%, calc(1% *
      (var(--l-primary) + (0.962 * (100 - var(--l-
      primary))))));
   --h-secondary: 169;
   --s-secondary: 100%;
   --l-secondary: 25%;
   --secondary-dark: hsl(var(--h-secondary),3%,38%);
   --secondary-main: hsl(var(--h-secondary),100%,25%);
   --secondary-light: hsl(calc(var(--h-secondary) -
      30),34%,62%);
   --secondary-white: hsl(calc(var(--h-secondary) -
      30),80%,93%);
   --h-accent: 345;
   --s-accent: 68%;
   --l-accent: 53%;
   --accent-main: hsl(var(--h-accent),var(--s-accent),var(--l-
      accent));
}
```

How Olivero is doing this

- Add a plain text field
  - Should accept a full-length hexcode
  - Should validate the input
  - Highly accessible, but not great if you don’t have a color in mind
- Add a color input next to that
  - Add some event listeners to keep inputs in sync
  - Provides sighted users a better experience to experiment with colors
- Convert the hex value to HSL in PHP



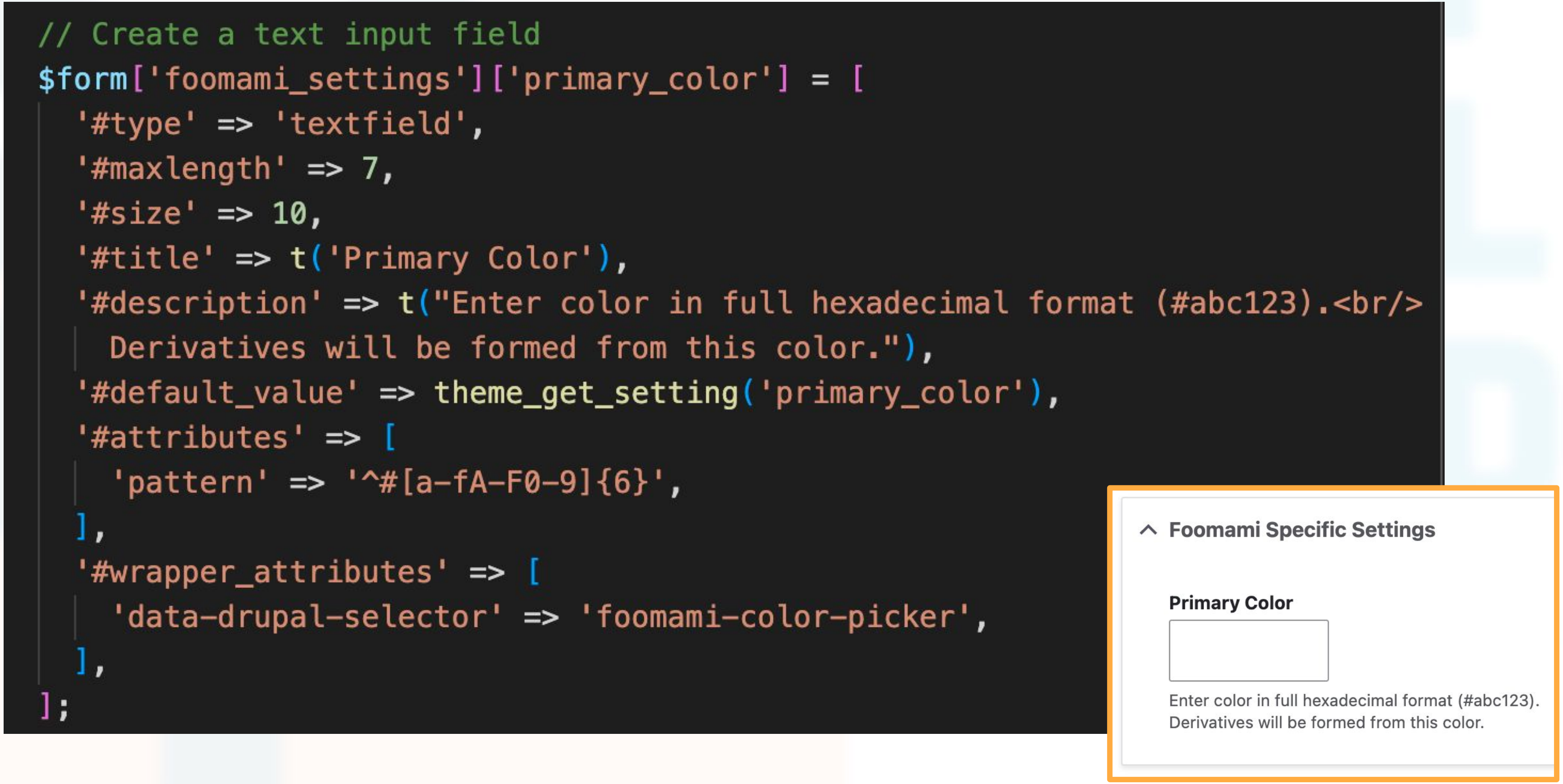
Custom Branding Color

#f5cc00

Enter brand color in full hexadecimal format (#abc123)

theme-settings-02

Create a new text input field



```
// Create a text input field
$form['foomami_settings']['primary_color'] = [
  '#type' => 'textfield',
  '#maxlength' => 7,
  '#size' => 10,
  '#title' => t('Primary Color'),
  '#description' => t("Enter color in full hexadecimal format (#abc123).<br/>
   Derivatives will be formed from this color."),
  '#default_value' => theme_get_setting('primary_color'),
  '#attributes' => [
    'pattern' => '^#[a-fA-F0-9]{6}',
  ],
  '#wrapper_attributes' => [
    'data-drupal-selector' => 'foomami-color-picker',
  ],
];
```

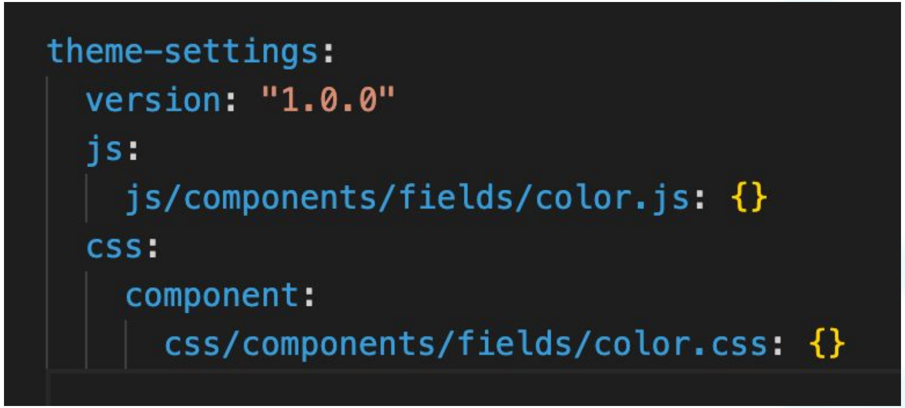
Primary Color

Enter color in full hexadecimal format (#abc123).
Derivatives will be formed from this color.

theme-settings-03

Let’s improve the UX!

- We have a text input
- We want to enhance it with JS & CSS
  - How? This isn’t our theme
- We can create a library in *our* theme and attach it to this form
  - Create new JS file
    - foomami/js/components/fields/color.js
  - Create new CSS file
    - foomami/css/components/fields/color.css
  - Define new libraries using these files
    - See image
  - Attach to our theme-settings forms



```
theme-settings:
  version: "1.0.0"
  js:
    js/components/fields/color.js: {}
  css:
    component:
      css/components/fields/color.css: {}
```

Quick Libraries Sidebar

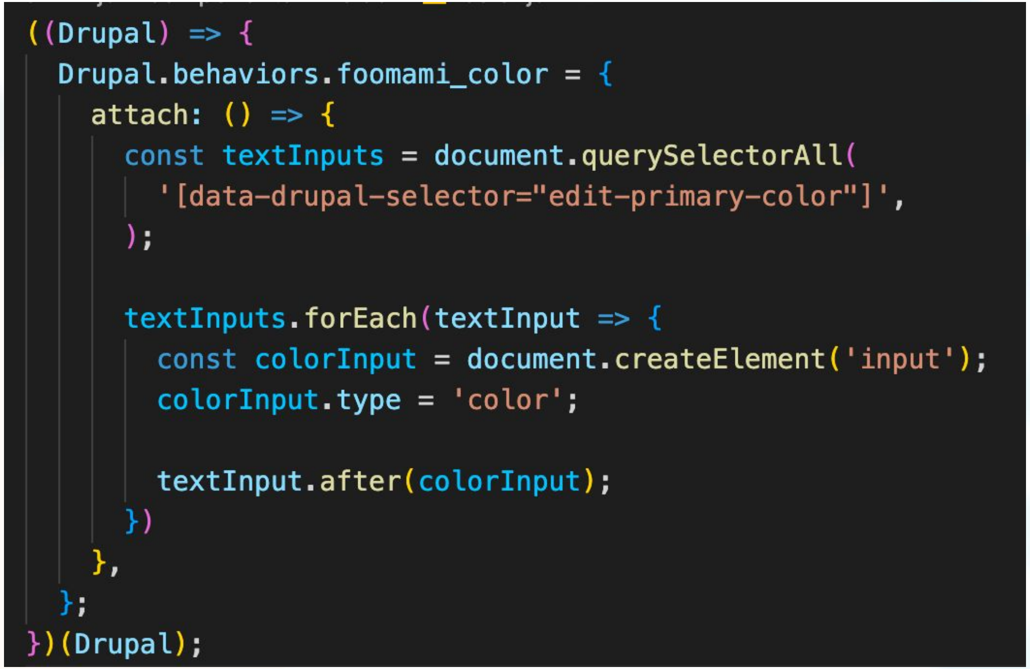
- Drupal’s library system is the way we incorporate CSS and JS
- A whole training could be done *just on the libraries API*
- There are lots of options that you can put on each resource
- Libraries can be extended or overridden in the info.yml file
  - Libraries-override lets you modify, replace, or remove individual resources or entire libraries
  - Libraries-extend lets you add additional resources onto other libraries
- Examples in Olivero
  - Override: replace core layout builder library with olivero-specific library
  - Extend: Add additional styles & scripts when the drupal.message library is used

theme-settings-04

Let’s improve the UX!

- Now we can inject a color input via JS
  - Use Drupal Behaviors

- What does this end up rendering?

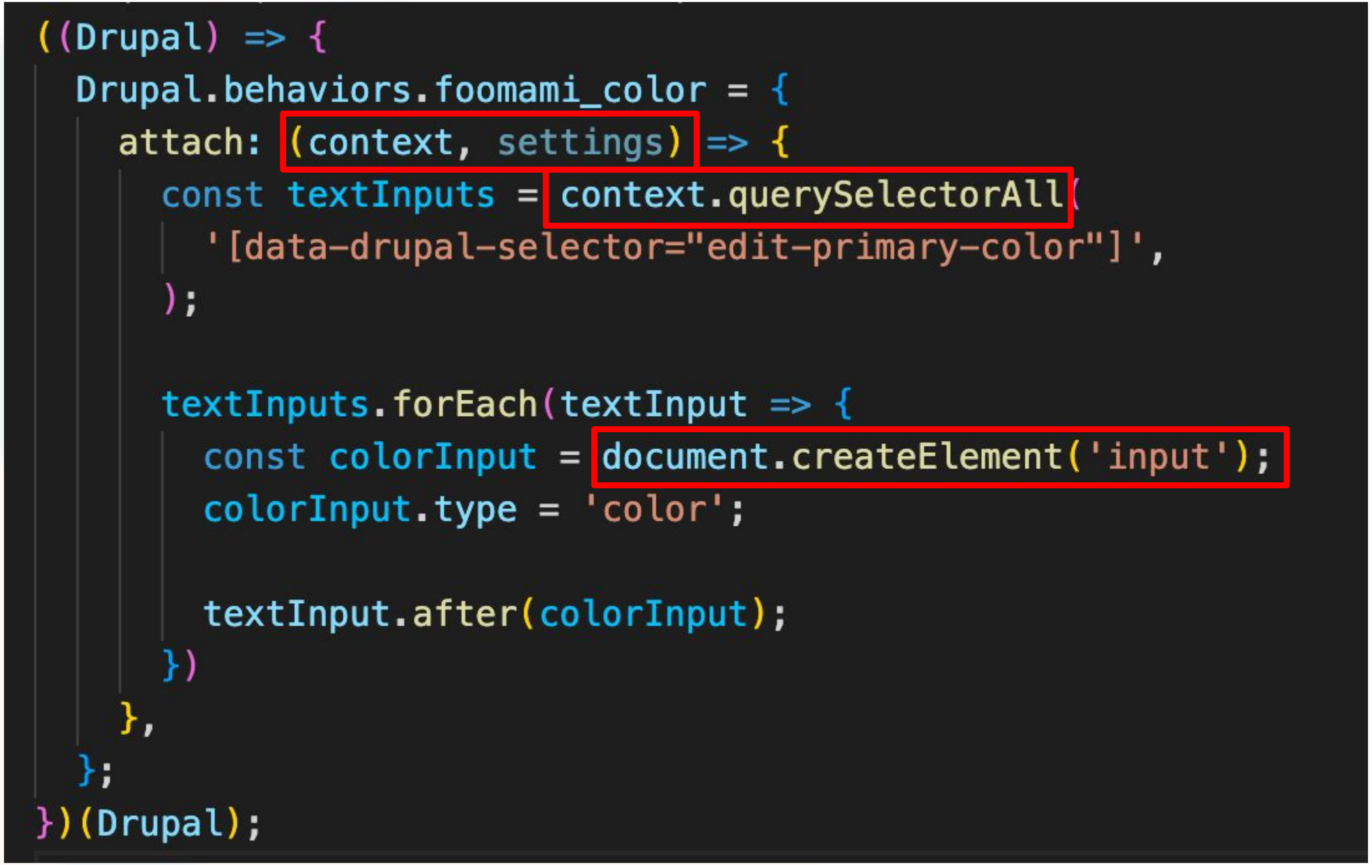


```
((Drupal) => {
  Drupal.behaviors.foomami_color = {
    attach: () => {
      const textInputs = document.querySelectorAll(
        '[data-drupal-selector="edit-primary-color"]',
      );

      textInputs.forEach(textInput => {
        const colorInput = document.createElement('input');
        colorInput.type = 'color';

        textInput.after(colorInput);
      })
    },
  };
})(Drupal);
```

How can we prevent multiple runs?

- Drupal behaviors can and often will run multiple times per page
    - Behaviors are run/attached every time Drupal updates the page
    - Ajax calls, BigPipe will cause this most often
- We can use the drupal.once library
    - Adds a class to elements once they’ve been selected, allowing us to avoid picking them again
- We can use `context`
    - context is equal to `document` on the first run
    - context is the DOM segment that’s injected after that

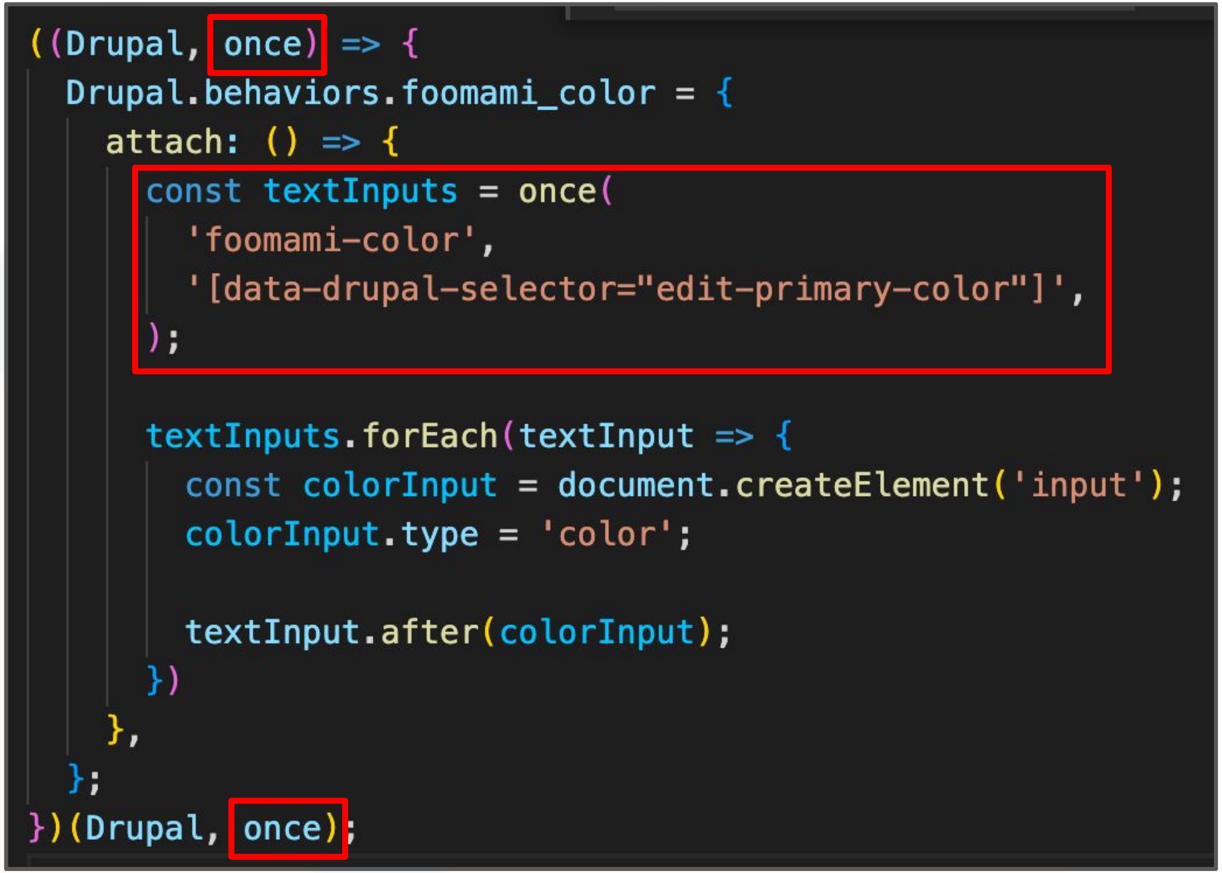
Limit selection with `context`



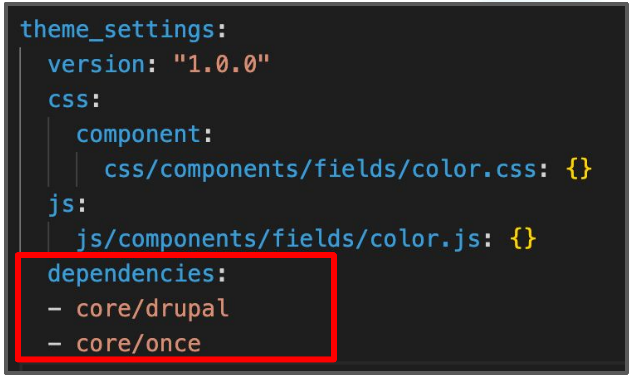
```
((Drupal) => {
  Drupal.behaviors.foomami_color = {
    attach: (context, settings) => {
      const textInputs = context.querySelectorAll(
        '[data-drupal-selector="edit-primary-color"]',
      );

      textInputs.forEach(textInput => {
        const colorInput = document.createElement('input');
        colorInput.type = 'color';

        textInput.after(colorInput);
      })
    },
  };
})(Drupal);
```

theme-settings-05

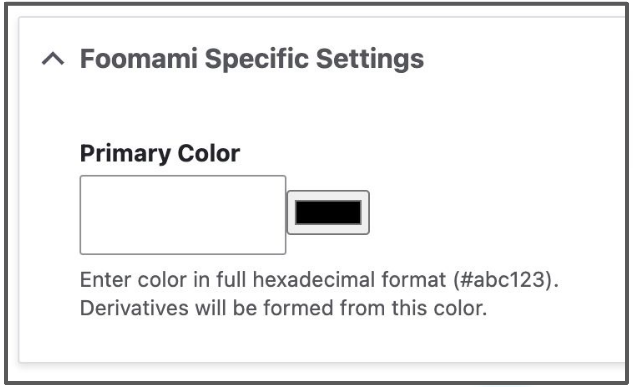
Limit selection with `once`



```
((Drupal, once) => {
  Drupal.behaviors.foomami_color = {
    attach: () => {
      const textInputs = once(
        'foomami-color',
        '[data-drupal-selector="edit-primary-color"]',
      );

      textInputs.forEach(textInput => {
        const colorInput = document.createElement('input');
        colorInput.type = 'color';

        textInput.after(colorInput);
      })
    },
  };
})(Drupal, once);
```



```
theme_settings:
  version: "1.0.0"
  css:
    component:
      css/components/fields/color.css: {}
  js:
    js/components/fields/color.js: {}
  dependencies:
  - core/drupal
  - core/once
```



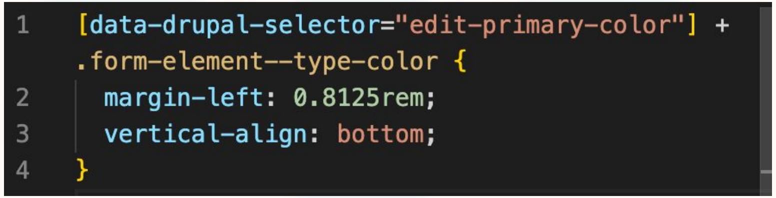
Primary Color

Enter color in full hexadecimal format (#abc123).
Derivatives will be formed from this color.

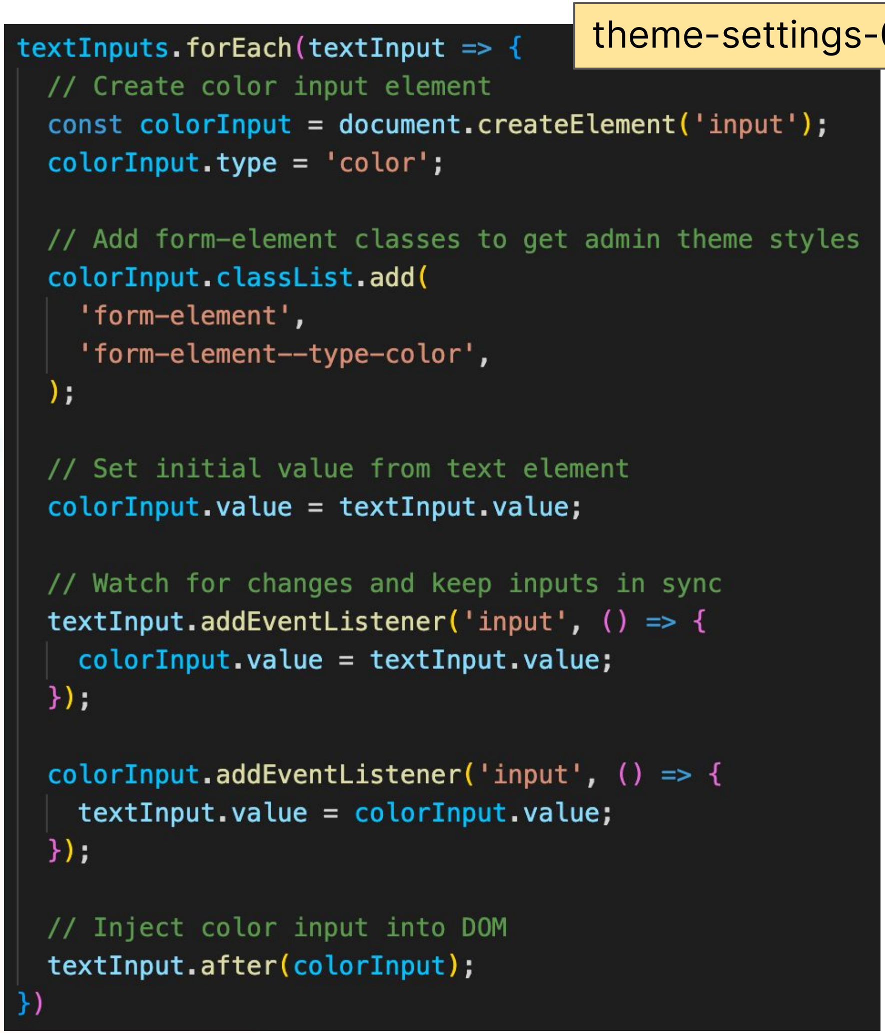
Wire it up

theme-settings-06

- Try it out!
  - ○ Change the text input
  - ○ Change the color input
    - ■ Changes should sync
  - ○ Save the form
    - ■ Changes should persist



```css
[data-drupal-selector="edit-primary-color"] + .form-element--type-color {
  margin-left: 0.8125rem;
  vertical-align: bottom;
}
```



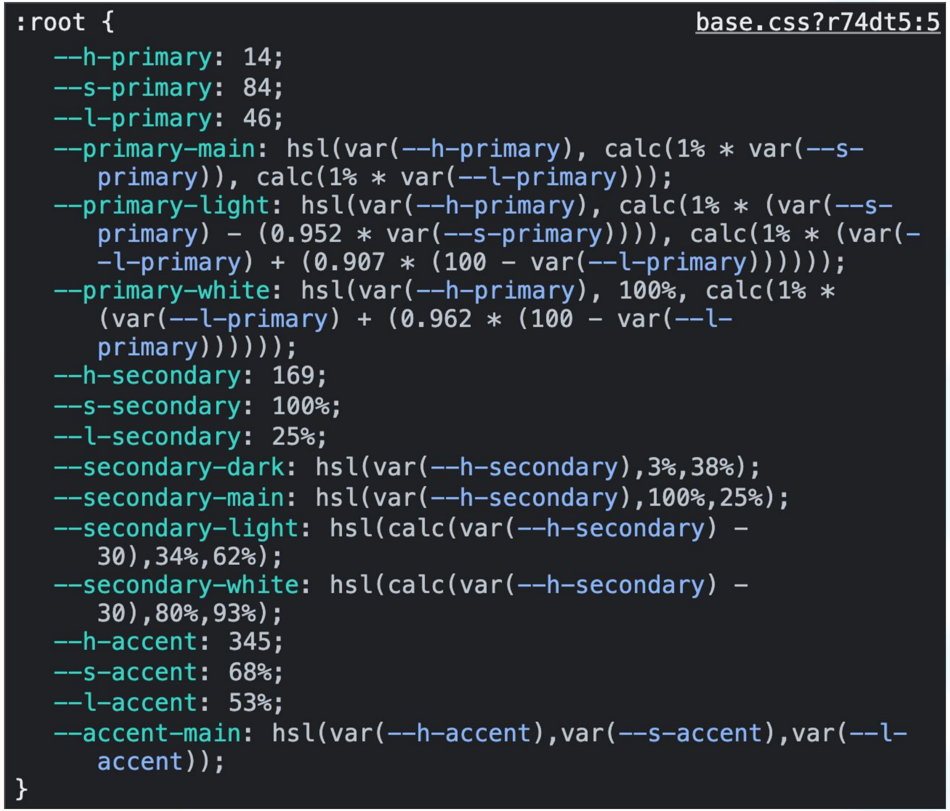
```javascript
textInputs.forEach(textInput => {
  // Create color input element
  const colorInput = document.createElement('input');
  colorInput.type = 'color';

  // Add form-element classes to get admin theme styles
  colorInput.classList.add(
    'form-element',
    'form-element--type-color',
  );

  // Set initial value from text element
  colorInput.value = textInput.value;

  // Watch for changes and keep inputs in sync
  textInput.addEventListener('input', () => {
    colorInput.value = textInput.value;
  });

  colorInput.addEventListener('input', () => {
    textInput.value = colorInput.value;
  });

  // Inject color input into DOM
  textInput.after(colorInput);
})
```

Don’t forget the other colors!

- We still need a field for:
  - Secondary color
  - Accent color

- We’ll need to modify our theme-settings
  - Make a template variable
  - Customize each individual field

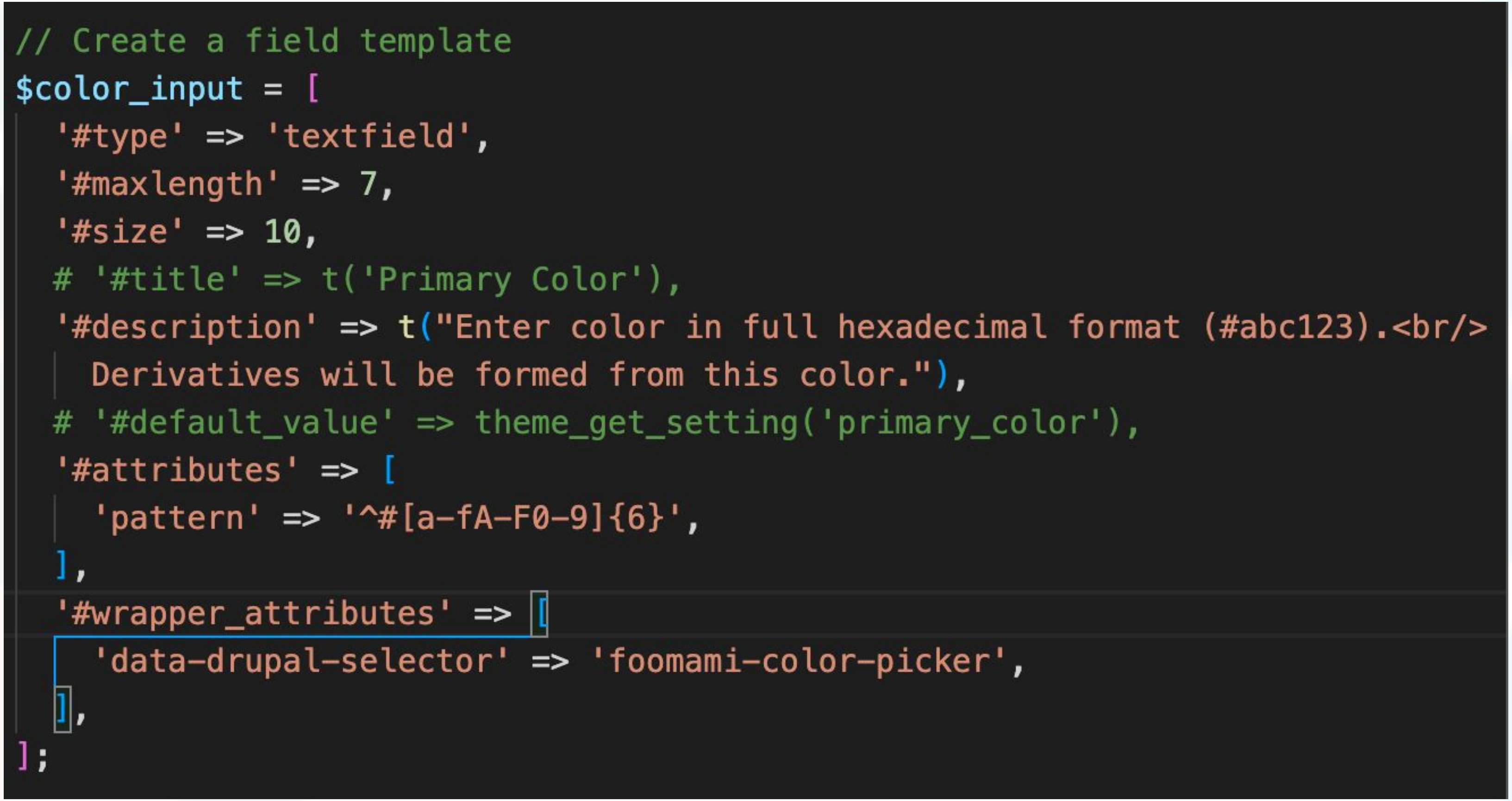
- We’ll need to generalize our JS/CSS



```
:root {                                                           base.css?r74dt5:5
    --h-primary: 14;
    --s-primary: 84;
    --l-primary: 46;
    --primary-main: hsl(var(--h-primary), calc(1% * var(--s-
       primary)), calc(1% * var(--l-primary)));
    --primary-light: hsl(var(--h-primary), calc(1% * (var(--s-
       primary) - (0.952 * var(--s-primary)))), calc(1% * (var(-
       -l-primary) + (0.907 * (100 - var(--l-primary))))));
    --primary-white: hsl(var(--h-primary), 100%, calc(1% *
       (var(--l-primary) + (0.962 * (100 - var(--l-
       primary))))));
    --h-secondary: 169;
    --s-secondary: 100%;
    --l-secondary: 25%;
    --secondary-dark: hsl(var(--h-secondary),3%,38%);
    --secondary-main: hsl(var(--h-secondary),100%,25%);
    --secondary-light: hsl(calc(var(--h-secondary) -
       30),34%,62%);
    --secondary-white: hsl(calc(var(--h-secondary) -
       30),80%,93%);
    --h-accent: 345;
    --s-accent: 68%;
    --l-accent: 53%;
    --accent-main: hsl(var(--h-accent),var(--s-accent),var(--l-
       accent));
}
```

theme-settings-07

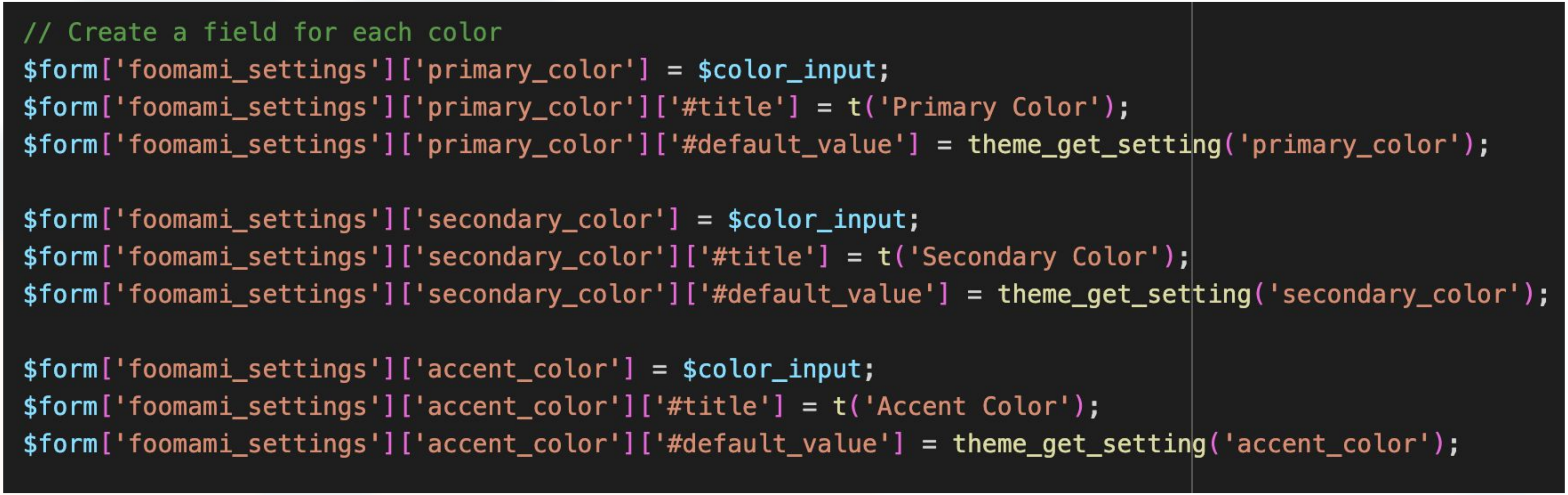
Add our other fields



```
// Create a field template
$color_input = [
  '#type' => 'textfield',
  '#maxlength' => 7,
  '#size' => 10,
  # '#title' => t('Primary Color'),
  '#description' => t("Enter color in full hexadecimal format (#abc123).<br/>
    Derivatives will be formed from this color."),
  # '#default_value' => theme_get_setting('primary_color'),
  '#attributes' => [
    'pattern' => '^#[a-fA-F0-9]{6}',
  ],
  '#wrapper_attributes' => [
    'data-drupal-selector' => 'foomami-color-picker',
  ],
];
```

theme-settings-07

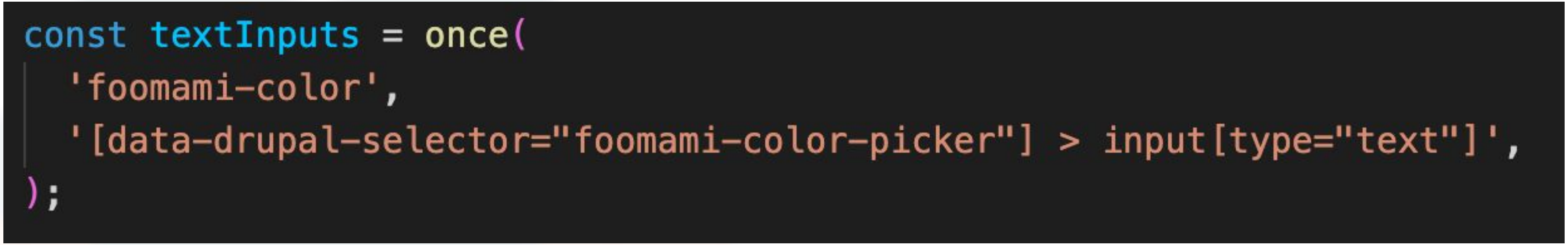
Add our other fields



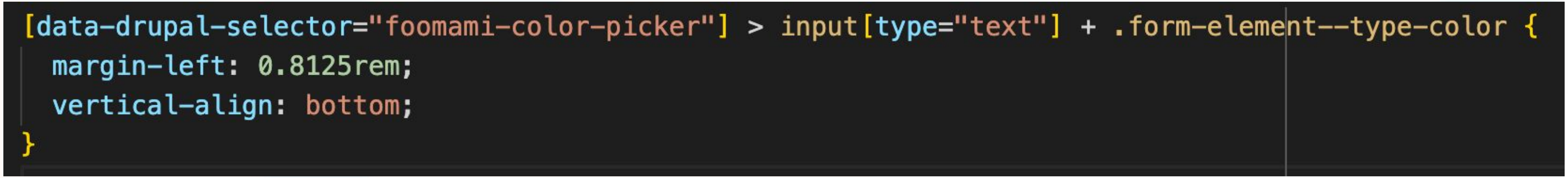
```
// Create a field for each color
$form['foomami_settings']['primary_color'] = $color_input;
$form['foomami_settings']['primary_color']['#title'] = t('Primary Color');
$form['foomami_settings']['primary_color']['#default_value'] = theme_get_setting('primary_color');

$form['foomami_settings']['secondary_color'] = $color_input;
$form['foomami_settings']['secondary_color']['#title'] = t('Secondary Color');
$form['foomami_settings']['secondary_color']['#default_value'] = theme_get_setting('secondary_color');

$form['foomami_settings']['accent_color'] = $color_input;
$form['foomami_settings']['accent_color']['#title'] = t('Accent Color');
$form['foomami_settings']['accent_color']['#default_value'] = theme_get_setting('accent_color');
```

theme-settings-07

Generalize our JS/CSS



```
const textInputs = once(
  'foomami-color',
  '[data-drupal-selector="foomami-color-picker"] > input[type="text"]',
);
```



```
[data-drupal-selector="foomami-color-picker"] > input[type="text"] + .form-element--type-color {
  margin-left: 0.8125rem;
  vertical-align: bottom;
}
```

Use the values from the form

- So far we’ve only stored the values. Now we have to use them too!

- We need to convert hex values to HSL

- We need to inject those values into the page so the CSS can use the overwritten variable values

theme-settings-08

Converting HEX to HSL

- Convert Hex → RGB
  - Convert trio of base-16 values to base-10
- Convert RGB → HSL
  - Math
- https://bit.ly/3HRsu8O
  - It’ll be easier to just copy & paste
  - Or you can update with git

Converting HEX to HSL

$$R' = R/255$$

$$G' = G/255$$

$$B' = B/255$$

$$Cmax = \max(R', G', B')$$

$$Cmin = \min(R', G', B')$$

$$\Delta = Cmax - Cmin$$

Hue calculation:

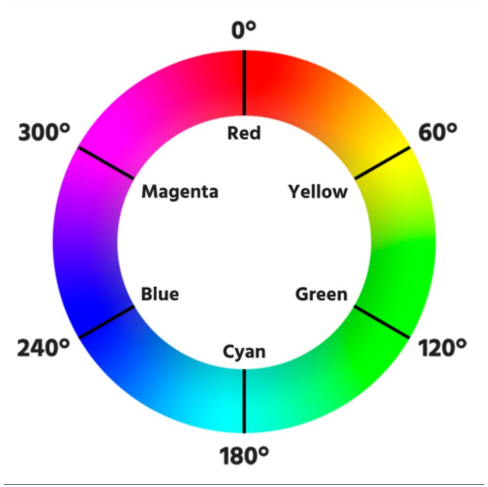
$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\frac{G'-B'}{\Delta} mod 6\right) & , C_{max} = R' \\ 60^\circ \times \left(\frac{B'-R'}{\Delta} + 2\right) & , C_{max} = G' \\ 60^\circ \times \left(\frac{R'-G'}{\Delta} + 4\right) & , C_{max} = B' \end{cases}$$

Saturation calculation:

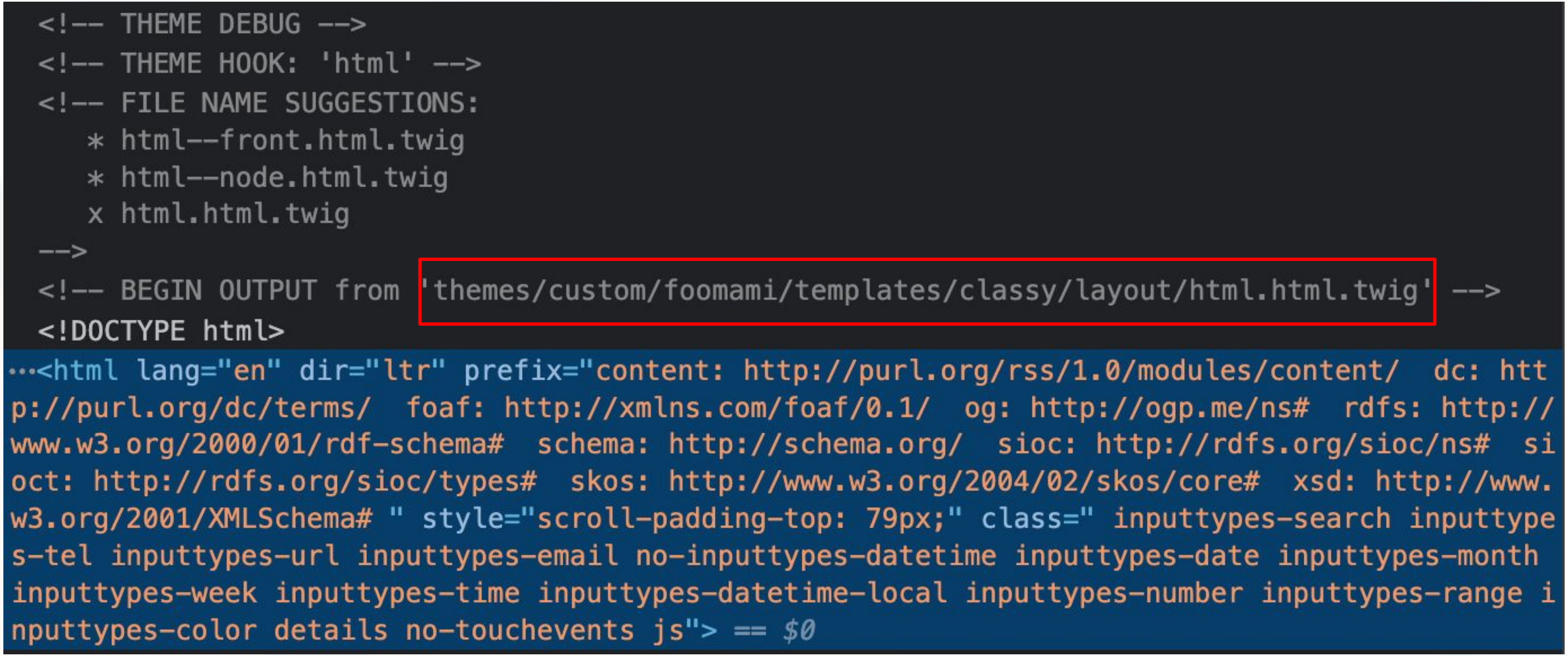
$$S = \begin{cases} 0 & , \Delta = 0 \\ \frac{\Delta}{1-|2L-1|} & , \Delta <> 0 \end{cases}$$

Lightness calculation:

$$L = (Cmax + Cmin) / 2$$

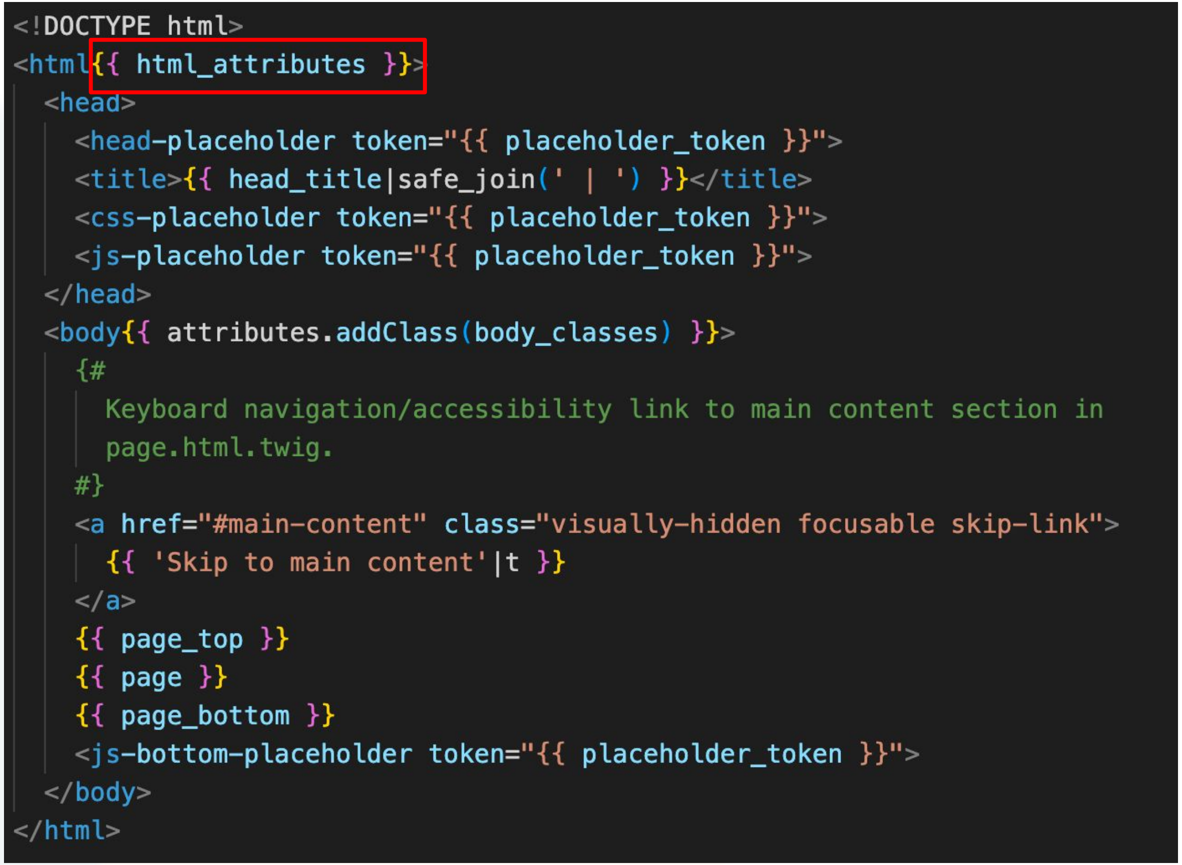


Use the values from the form



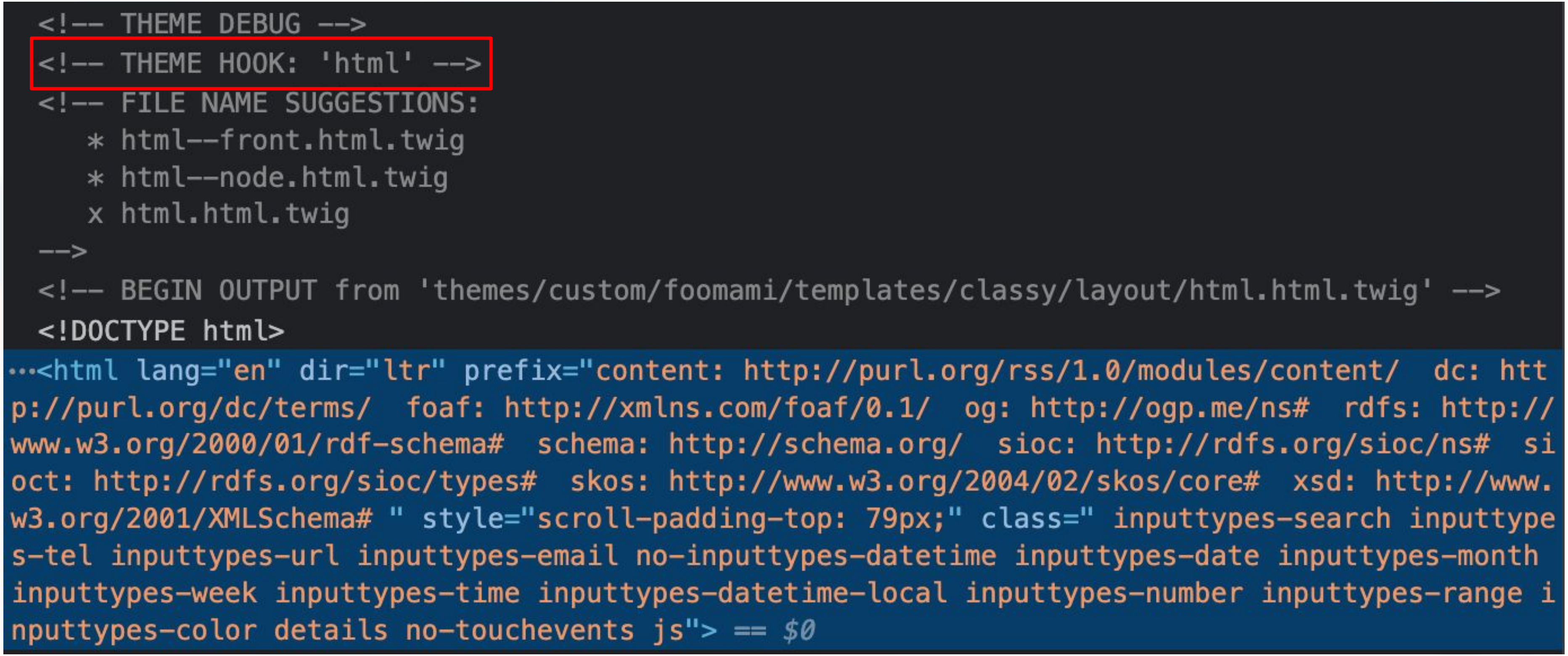
```
<!-- THEME DEBUG -->
<!-- THEME HOOK: 'html' -->
<!-- FILE NAME SUGGESTIONS:
   * html--front.html.twig
   * html--node.html.twig
   x html.html.twig
-->
<!-- BEGIN OUTPUT from 'themes/custom/foomami/templates/classy/layout/html.html.twig' -->
<!DOCTYPE html>
···<html lang="en" dir="ltr" prefix="content: http://purl.org/rss/1.0/modules/content/  dc: htt
p://purl.org/dc/terms/  foaf: http://xmlns.com/foaf/0.1/  og: http://ogp.me/ns#  rdfs: http://
www.w3.org/2000/01/rdf-schema#  schema: http://schema.org/  sioc: http://rdfs.org/sioc/ns#  si
oct: http://rdfs.org/sioc/types#  skos: http://www.w3.org/2004/02/skos/core#  xsd: http://www.
w3.org/2001/XMLSchema# " style="scroll-padding-top: 79px;" class=" inputtypes-search inputtype
s-tel inputtypes-url inputtypes-email no-inputtypes-datetime inputtypes-date inputtypes-month
inputtypes-week inputtypes-time inputtypes-datetime-local inputtypes-number inputtypes-range i
nputtypes-color details no-touchevents js"> == $0
```

Use the values from the form



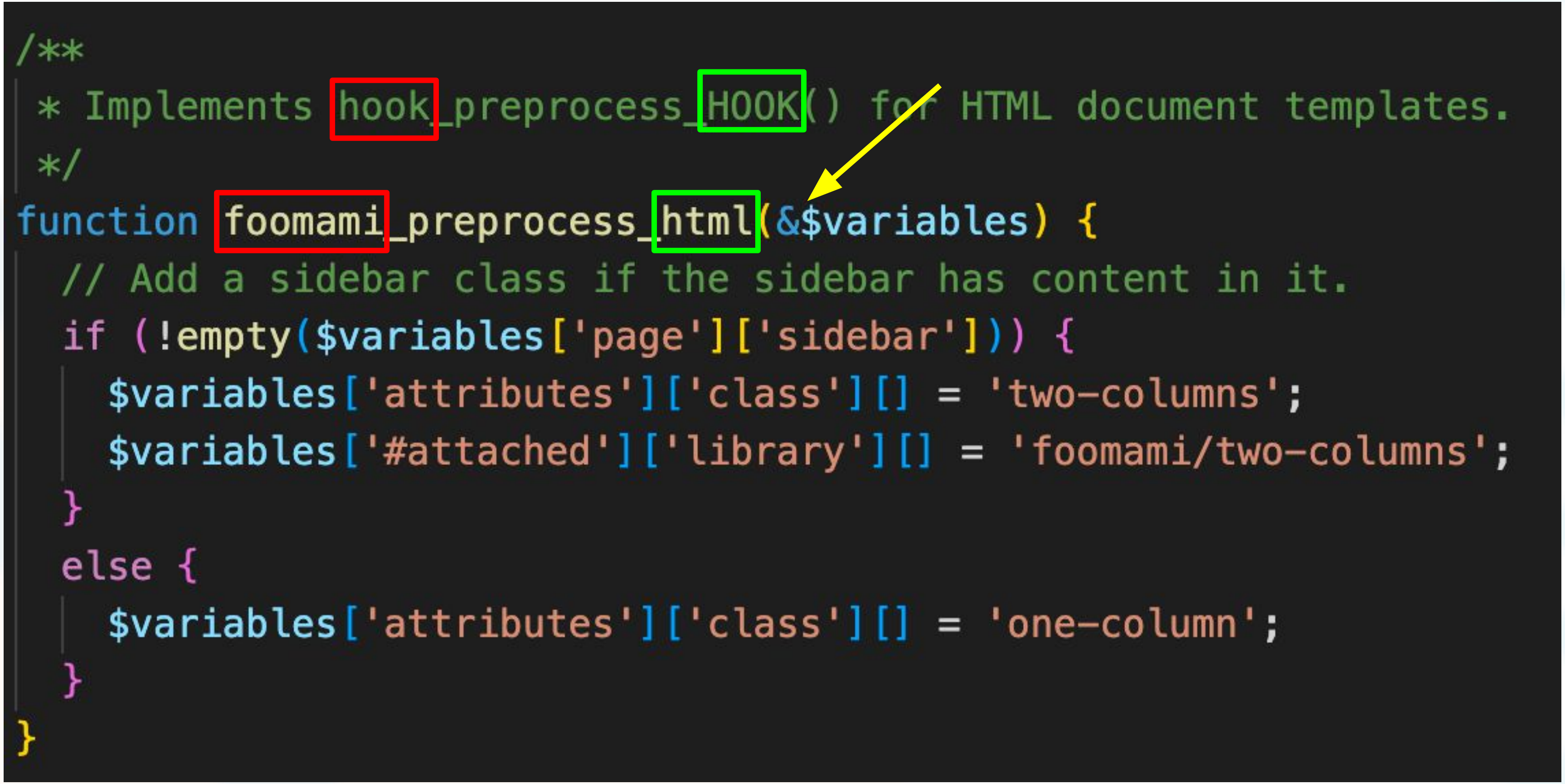
```
<!DOCTYPE html>
<html{{ html_attributes }}>
  <head>
    <head-placeholder token="{{ placeholder_token }}">
    <title>{{ head_title|safe_join(' | ') }}</title>
    <css-placeholder token="{{ placeholder_token }}">
    <js-placeholder token="{{ placeholder_token }}">
  </head>
  <body{{ attributes.addClass(body_classes)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

Back to the hooks!



```
<!-- THEME DEBUG -->
<!-- THEME HOOK: 'html' -->
<!-- FILE NAME SUGGESTIONS:
   * html--front.html.twig
   * html--node.html.twig
   x html.html.twig
-->
<!-- BEGIN OUTPUT from 'themes/custom/foomami/templates/classy/layout/html.html.twig' -->
<!DOCTYPE html>
…<html lang="en" dir="ltr" prefix="content: http://purl.org/rss/1.0/modules/content/  dc: htt
p://purl.org/dc/terms/  foaf: http://xmlns.com/foaf/0.1/  og: http://ogp.me/ns#  rdfs: http://
www.w3.org/2000/01/rdf-schema#  schema: http://schema.org/  sioc: http://rdfs.org/sioc/ns#  si
oct: http://rdfs.org/sioc/types#  skos: http://www.w3.org/2004/02/skos/core#  xsd: http://www.
w3.org/2001/XMLSchema# " style="scroll-padding-top: 79px;" class=" inputtypes-search inputtype
s-tel inputtypes-url inputtypes-email no-inputtypes-datetime inputtypes-date inputtypes-month
inputtypes-week inputtypes-time inputtypes-datetime-local inputtypes-number inputtypes-range i
nputtypes-color details no-touchevents js"> == $0
```

Back to the hooks!



```
/**
 * Implements hook_preprocess_HOOK() for HTML document templates.
 */
function foomami_preprocess_html(&$variables) {
  // Add a sidebar class if the sidebar has content in it.
  if (!empty($variables['page']['sidebar'])) {
    $variables['attributes']['class'][] = 'two-columns';
    $variables['#attached']['library'][] = 'foomami/two-columns';
  }
  else {
    $variables['attributes']['class'][] = 'one-column';
  }
}
```

Back to the hooks

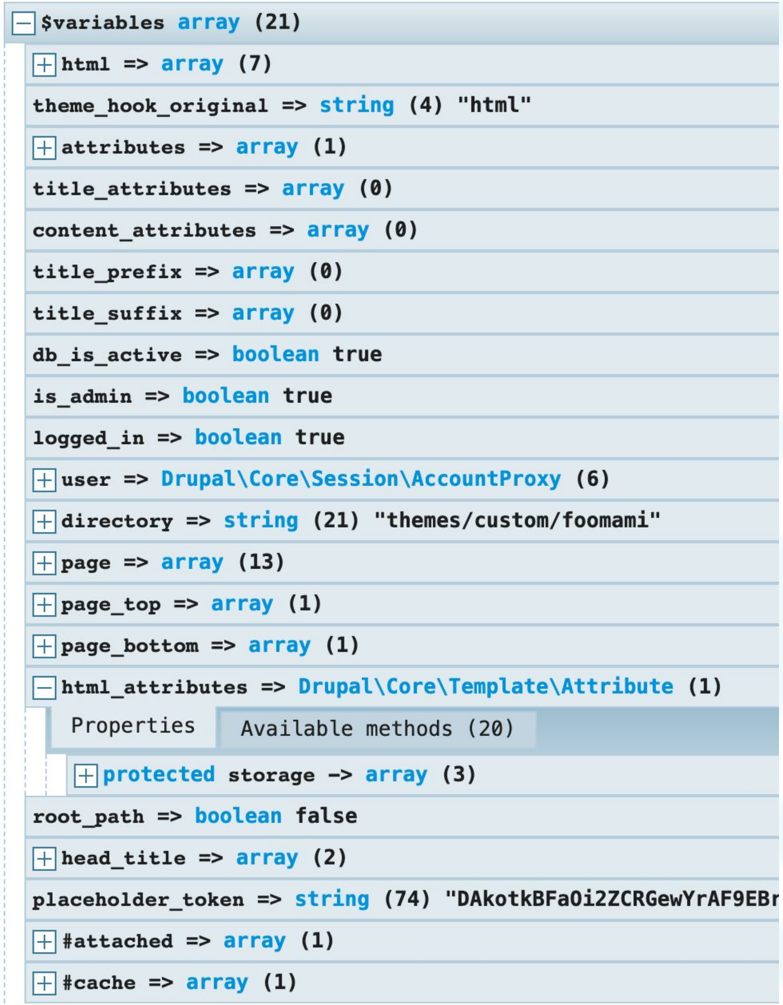
- We want to add in our H, S, & L values into the `style` attribute on our `html` tag

- We know we want to modify `html_attributes`

- How do we add our stuff?
- How do we avoid breaking this for other modules?
- Lets debug it!

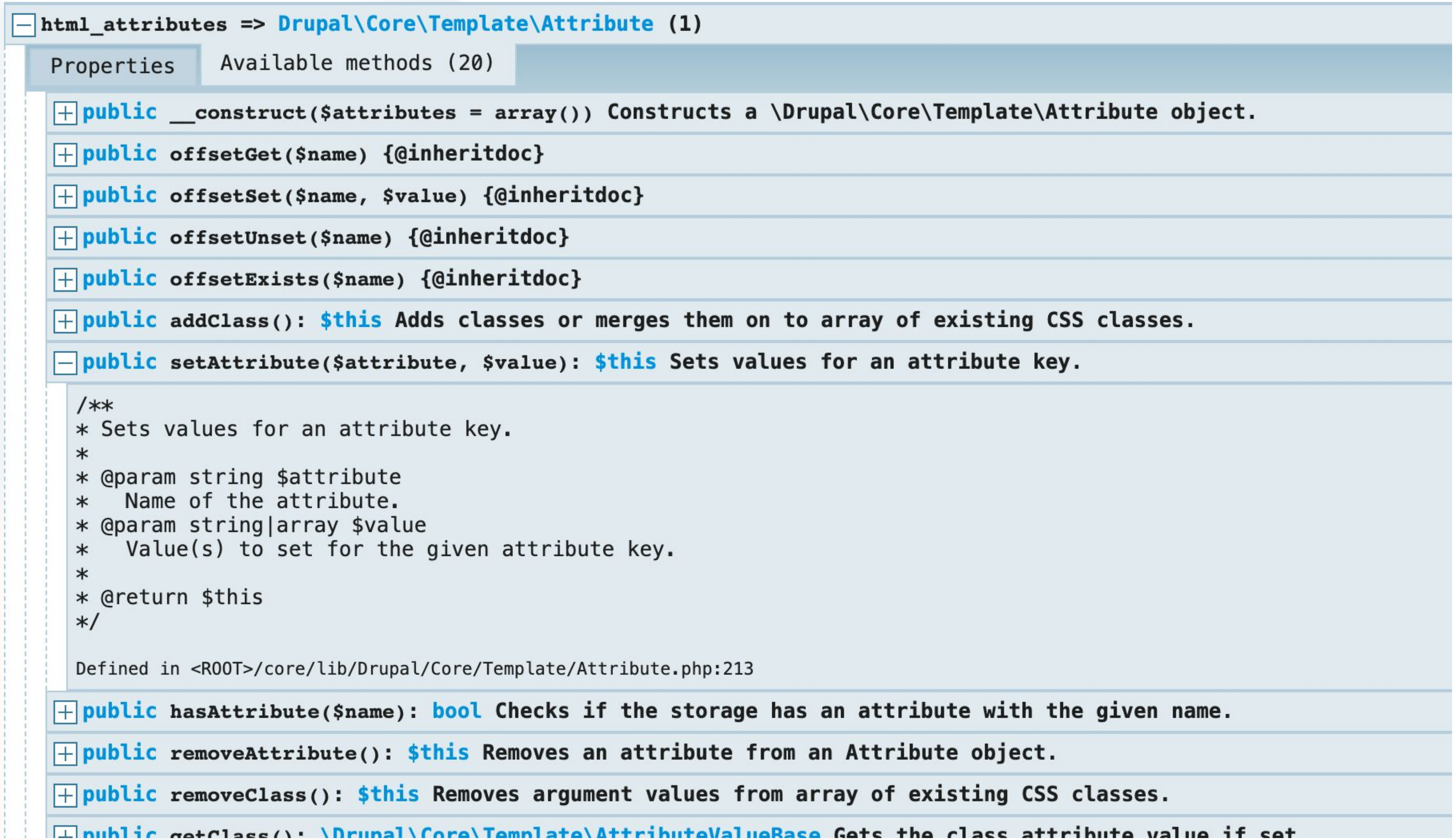
Back to the hooks

- kint($variables)
- html_attributes is an Attribute object
  - Has `protected` property `storage`
- What methods do we have access to?
  - Kint's methods tab
  - api.drupal.org
  - The PHP file the class comes from



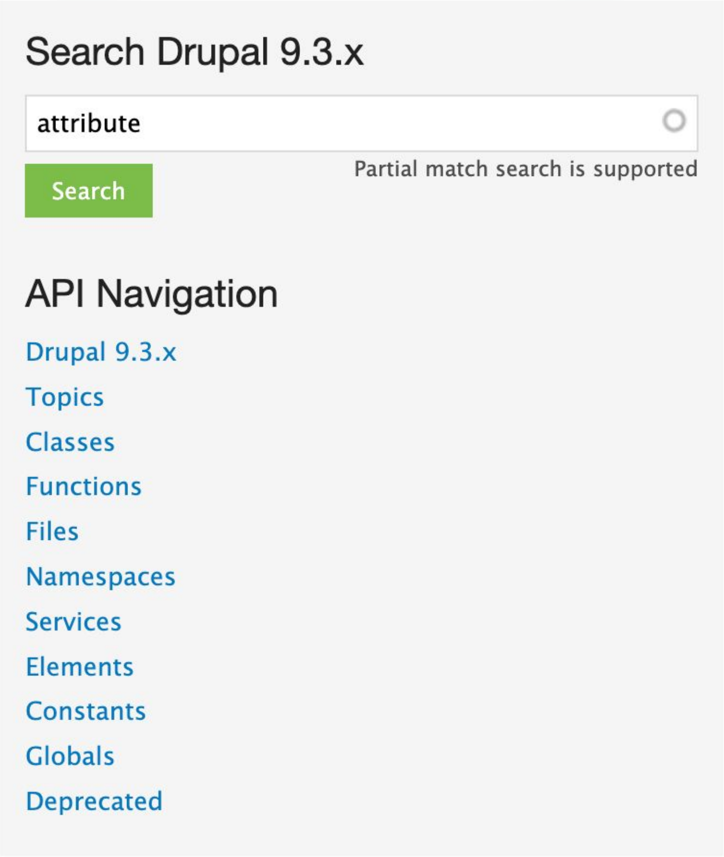
```
⊟ $variables array (21)
  ⊞ html => array (7)
  theme_hook_original => string (4) "html"
  ⊞ attributes => array (1)
  title_attributes => array (0)
  content_attributes => array (0)
  title_prefix => array (0)
  title_suffix => array (0)
  db_is_active => boolean true
  is_admin => boolean true
  logged_in => boolean true
  ⊞ user => Drupal\Core\Session\AccountProxy (6)
  ⊞ directory => string (21) "themes/custom/foomami"
  ⊞ page => array (13)
  ⊞ page_top => array (1)
  ⊞ page_bottom => array (1)
  ⊟ html_attributes => Drupal\Core\Template\Attribute (1)
      Properties    Available methods (20)
      ⊞ protected storage -> array (3)
  root_path => boolean false
  ⊞ head_title => array (2)
  placeholder_token => string (74) "DAkotkBFaOi2ZCRGewYrAF9EBr
  ⊞ #attached => array (1)
  ⊞ #cache => array (1)
```

Kint’s methods tab



```
⊟ html_attributes => Drupal\Core\Template\Attribute (1)
  Properties   Available methods (20)
  ⊞ public __construct($attributes = array()) Constructs a \Drupal\Core\Template\Attribute object.
  ⊞ public offsetGet($name) {@inheritdoc}
  ⊞ public offsetSet($name, $value) {@inheritdoc}
  ⊞ public offsetUnset($name) {@inheritdoc}
  ⊞ public offsetExists($name) {@inheritdoc}
  ⊞ public addClass(): $this Adds classes or merges them on to array of existing CSS classes.
  ⊟ public setAttribute($attribute, $value): $this Sets values for an attribute key.
      /**
      * Sets values for an attribute key.
      *
      * @param string $attribute
      *   Name of the attribute.
      * @param string|array $value
      *   Value(s) to set for the given attribute key.
      *
      * @return $this
      */

      Defined in <ROOT>/core/lib/Drupal/Core/Template/Attribute.php:213
  ⊞ public hasAttribute($name): bool Checks if the storage has an attribute with the given name.
  ⊞ public removeAttribute(): $this Removes an attribute from an Attribute object.
  ⊞ public removeClass(): $this Removes argument values from array of existing CSS classes.
  ⊞ public getClass(): \Drupal\Core\Template\AttributeValueBase Gets the class attribute value if set
```

api.drupal.org



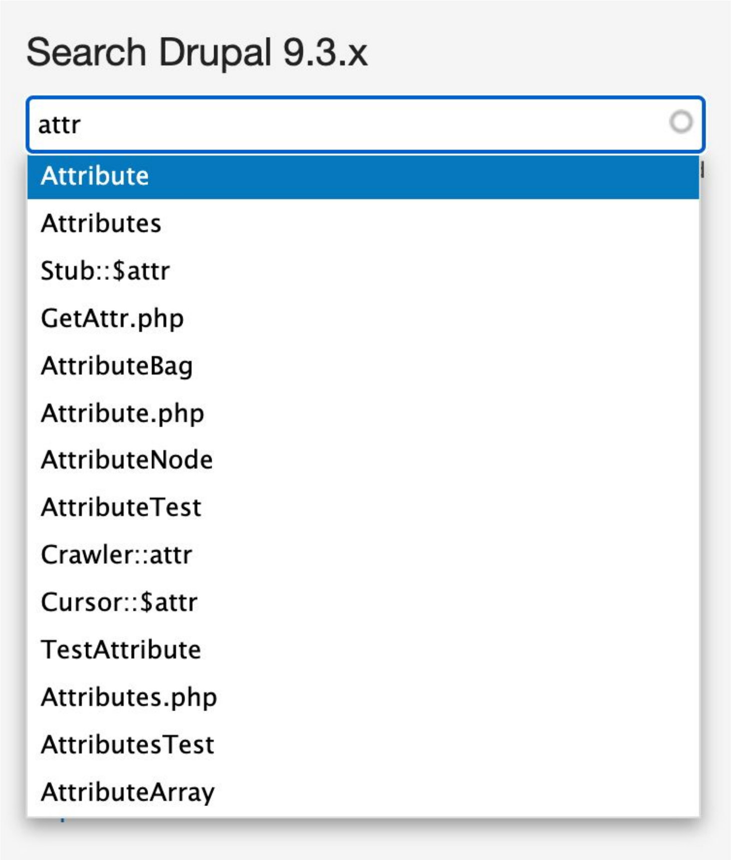
Search Drupal 9.3.x

attribute ○

Partial match search is supported

API Navigation

- Drupal 9.3.x
- Topics
- Classes
- Functions
- Files
- Namespaces
- Services
- Elements
- Constants
- Globals
- Deprecated

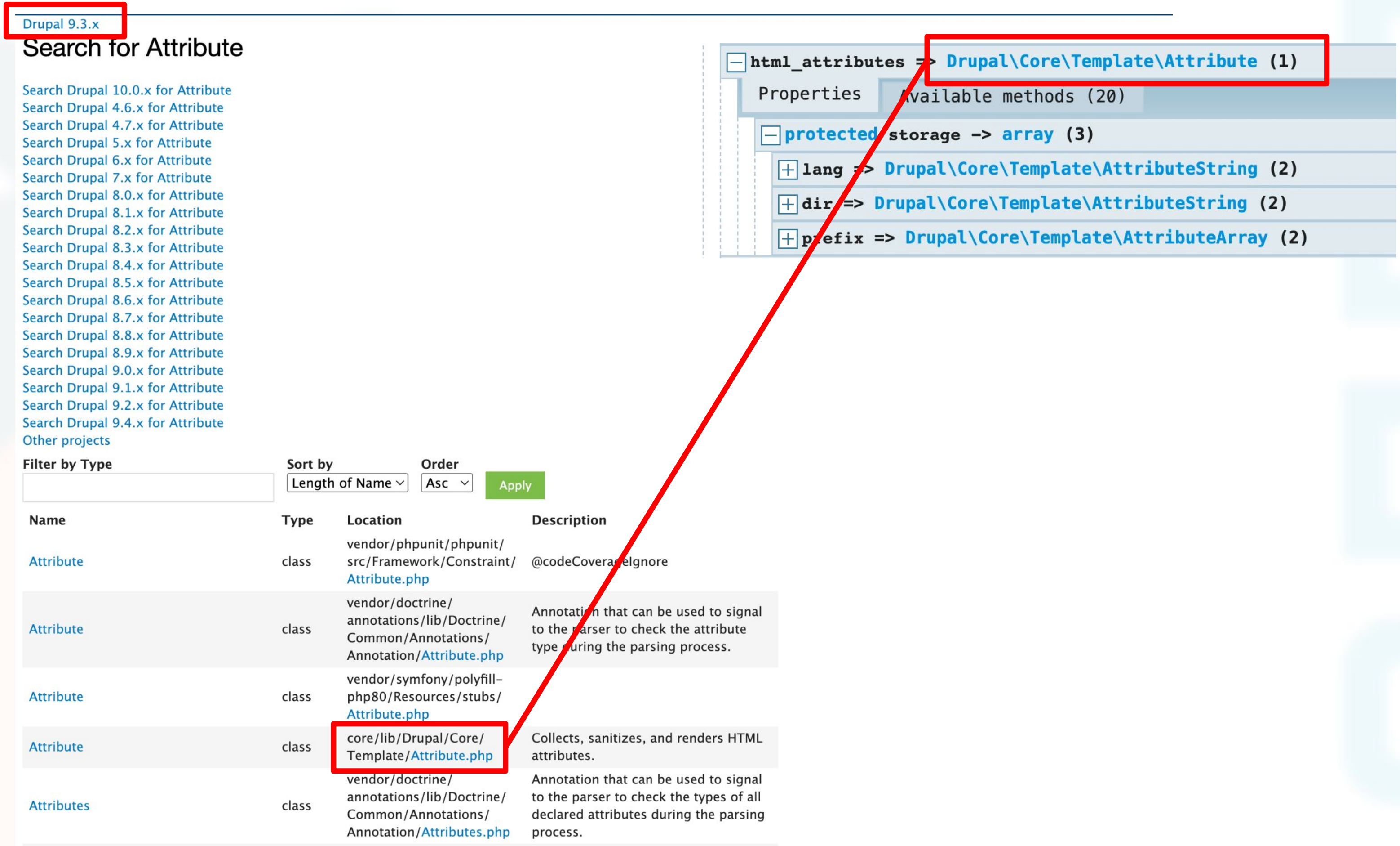


Search Drupal 9.3.x

attr ○

- Attribute
- Attributes
- Stub::$attr
- GetAttr.php
- AttributeBag
- Attribute.php
- AttributeNode
- AttributeTest
- Crawler::attr
- Cursor::$attr
- TestAttribute
- Attributes.php
- AttributesTest
- AttributeArray

api.drupal.org



Drupal 9.3.x

Search for Attribute

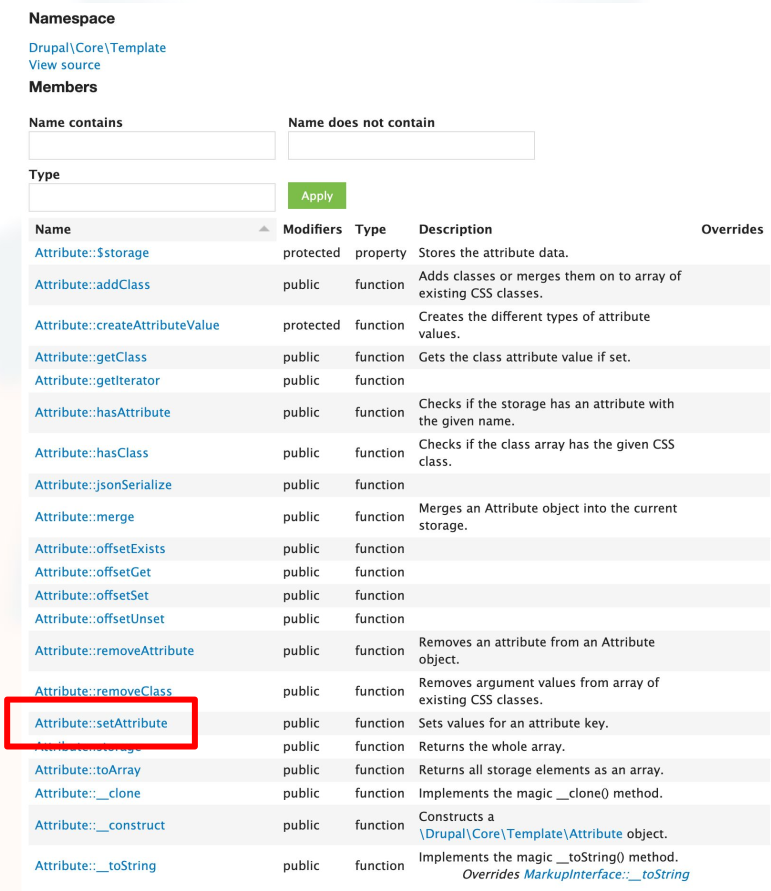
Search Drupal 10.0.x for Attribute
Search Drupal 4.6.x for Attribute
Search Drupal 4.7.x for Attribute
Search Drupal 5.x for Attribute
Search Drupal 6.x for Attribute
Search Drupal 7.x for Attribute
Search Drupal 8.0.x for Attribute
Search Drupal 8.1.x for Attribute
Search Drupal 8.2.x for Attribute
Search Drupal 8.3.x for Attribute
Search Drupal 8.4.x for Attribute
Search Drupal 8.5.x for Attribute
Search Drupal 8.6.x for Attribute
Search Drupal 8.7.x for Attribute
Search Drupal 8.8.x for Attribute
Search Drupal 8.9.x for Attribute
Search Drupal 9.0.x for Attribute
Search Drupal 9.1.x for Attribute
Search Drupal 9.2.x for Attribute
Search Drupal 9.4.x for Attribute
Other projects

Filter by Type:
Sort by: Length of Name
Order: Asc

Name	Type	Location	Description
Attribute	class	vendor/phpunit/phpunit/src/Framework/Constraint/Attribute.php	@codeCoverageIgnore
Attribute	class	vendor/doctrine/annotations/lib/Doctrine/Common/Annotations/Annotation/Attribute.php	Annotation that can be used to signal to the parser to check the attribute type during the parsing process.
Attribute	class	vendor/symfony/polyfill-php80/Resources/stubs/Attribute.php	
Attribute	class	core/lib/Drupal/Core/Template/Attribute.php	Collects, sanitizes, and renders HTML attributes.
Attributes	class	vendor/doctrine/annotations/lib/Doctrine/Common/Annotations/Annotation/Attributes.php	Annotation that can be used to signal to the parser to check the types of all declared attributes during the parsing process.

```
html_attributes => Drupal\Core\Template\Attribute (1)
  Properties    Available methods (20)
  protected storage -> array (3)
    lang => Drupal\Core\Template\AttributeString (2)
    dir => Drupal\Core\Template\AttributeString (2)
    prefix => Drupal\Core\Template\AttributeArray (2)
```

api.drupal.org

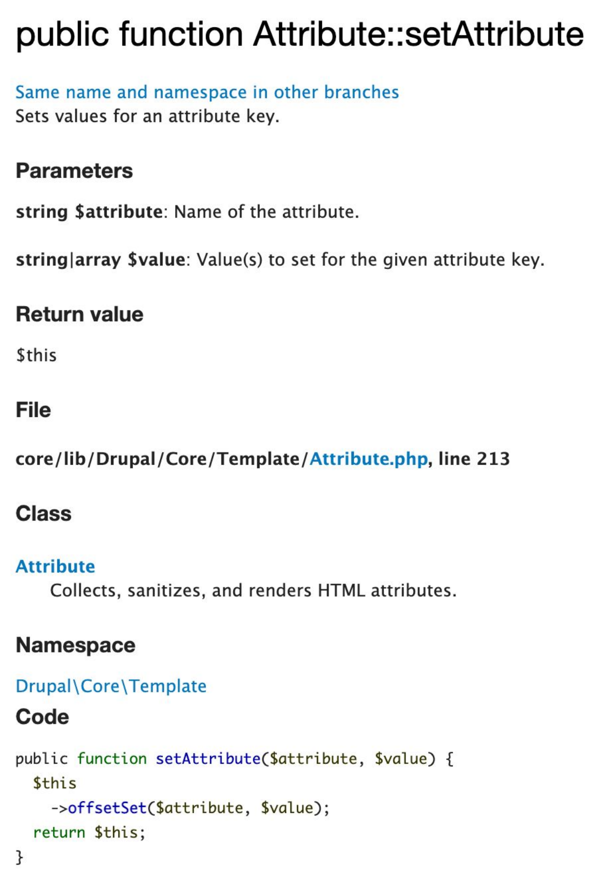


Namespace

Drupal\Core\Template
View source

Members

Name	Modifiers	Type	Description	Overrides
Attribute::$storage	protected	property	Stores the attribute data.	
Attribute::addClass	public	function	Adds classes or merges them on to array of existing CSS classes.	
Attribute::createAttributeValue	protected	function	Creates the different types of attribute values.	
Attribute::getClass	public	function	Gets the class attribute value if set.	
Attribute::getIterator	public	function		
Attribute::hasAttribute	public	function	Checks if the storage has an attribute with the given name.	
Attribute::hasClass	public	function	Checks if the class array has the given CSS class.	
Attribute::jsonSerialize	public	function		
Attribute::merge	public	function	Merges an Attribute object into the current storage.	
Attribute::offsetExists	public	function		
Attribute::offsetGet	public	function		
Attribute::offsetSet	public	function		
Attribute::offsetUnset	public	function		
Attribute::removeAttribute	public	function	Removes an attribute from an Attribute object.	
Attribute::removeClass	public	function	Removes argument values from array of existing CSS classes.	
Attribute::setAttribute	public	function	Sets values for an attribute key.	
Attribute::storage	public	function	Returns the whole array.	
Attribute::toArray	public	function	Returns all storage elements as an array.	
Attribute::__clone	public	function	Implements the magic __clone() method.	
Attribute::__construct	public	function	Constructs a \Drupal\Core\Template\Attribute object.	
Attribute::__toString	public	function	Implements the magic __toString() method. *Overrides MarkupInterface::__toString*	



public function Attribute::setAttribute

Same name and namespace in other branches
Sets values for an attribute key.

Parameters

string $attribute: Name of the attribute.

string|array $value: Value(s) to set for the given attribute key.

Return value

$this

File

core/lib/Drupal/Core/Template/Attribute.php, line 213

Class

Attribute
Collects, sanitizes, and renders HTML attributes.

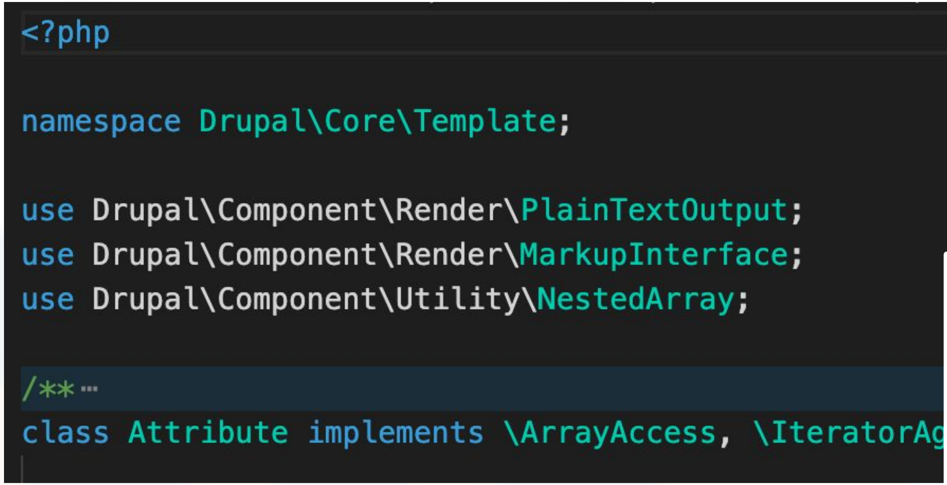
Namespace

Drupal\Core\Template

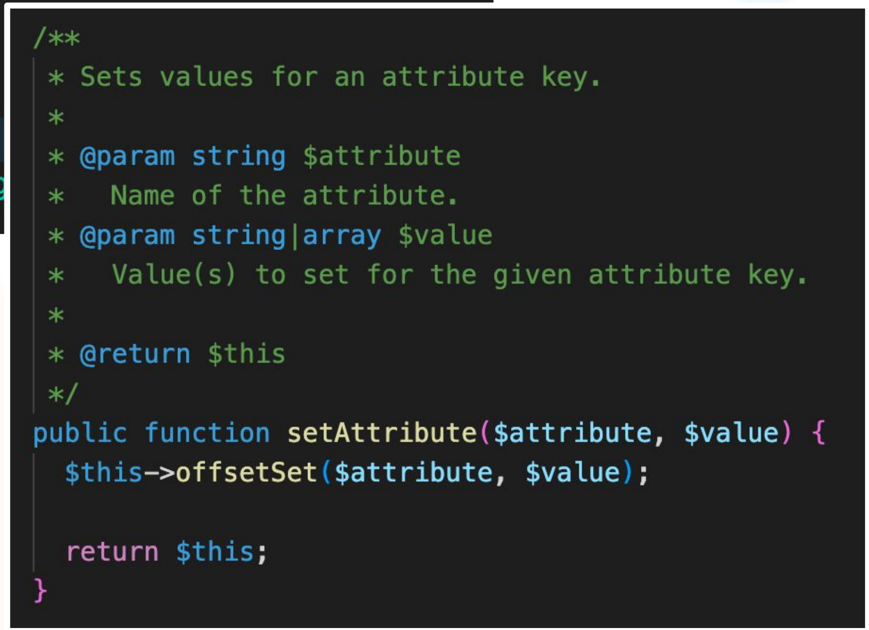
Code

```
public function setAttribute($attribute, $value) {
  $this
    ->offsetSet($attribute, $value);
  return $this;
}
```

The source file



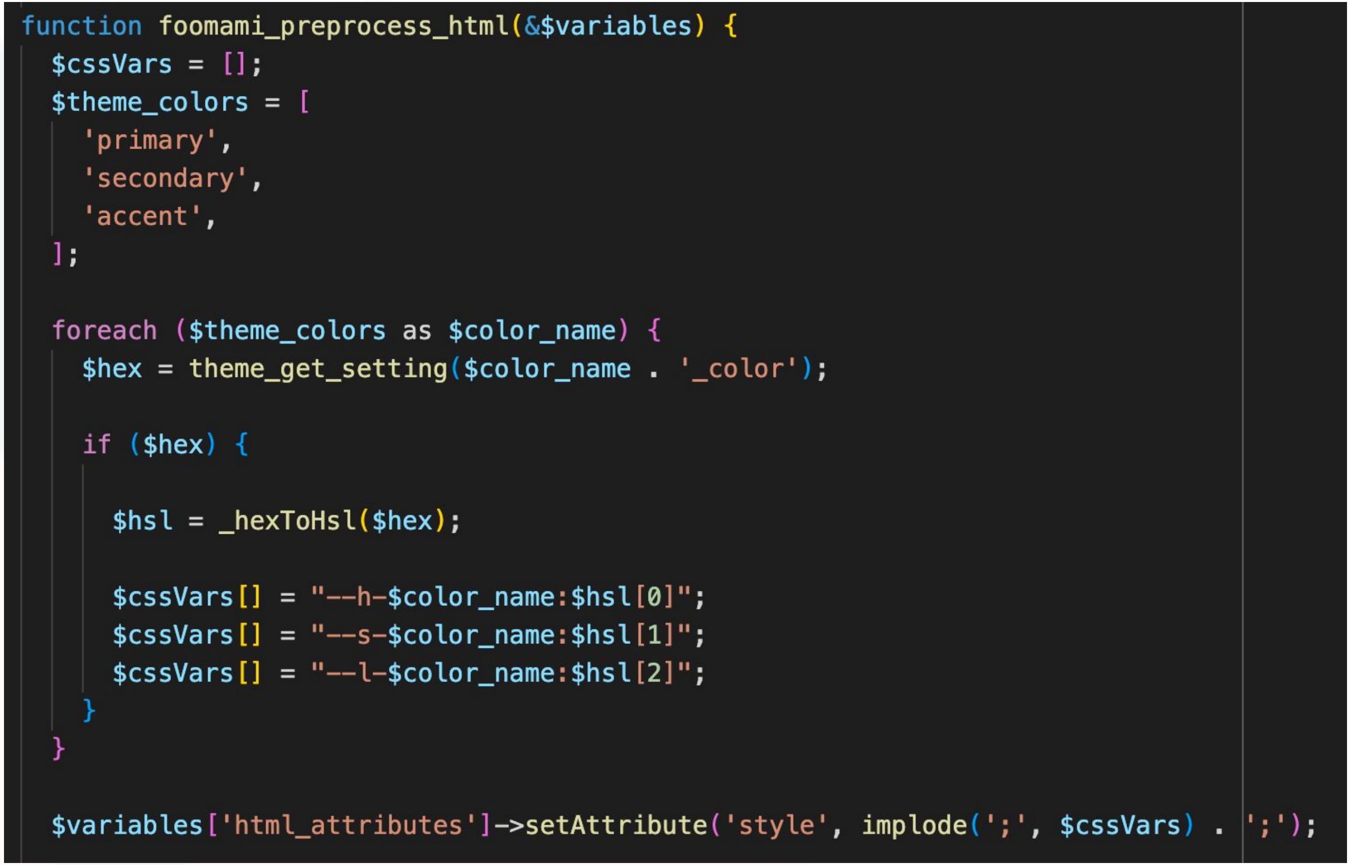
```
<?php

namespace Drupal\Core\Template;

use Drupal\Component\Render\PlainTextOutput;
use Drupal\Component\Render\MarkupInterface;
use Drupal\Component\Utility\NestedArray;

/**…
class Attribute implements \ArrayAccess, \IteratorAg
```



```
/**
 * Sets values for an attribute key.
 *
 * @param string $attribute
 *   Name of the attribute.
 * @param string|array $value
 *   Value(s) to set for the given attribute key.
 *
 * @return $this
 */
public function setAttribute($attribute, $value) {
  $this->offsetSet($attribute, $value);

  return $this;
}
```

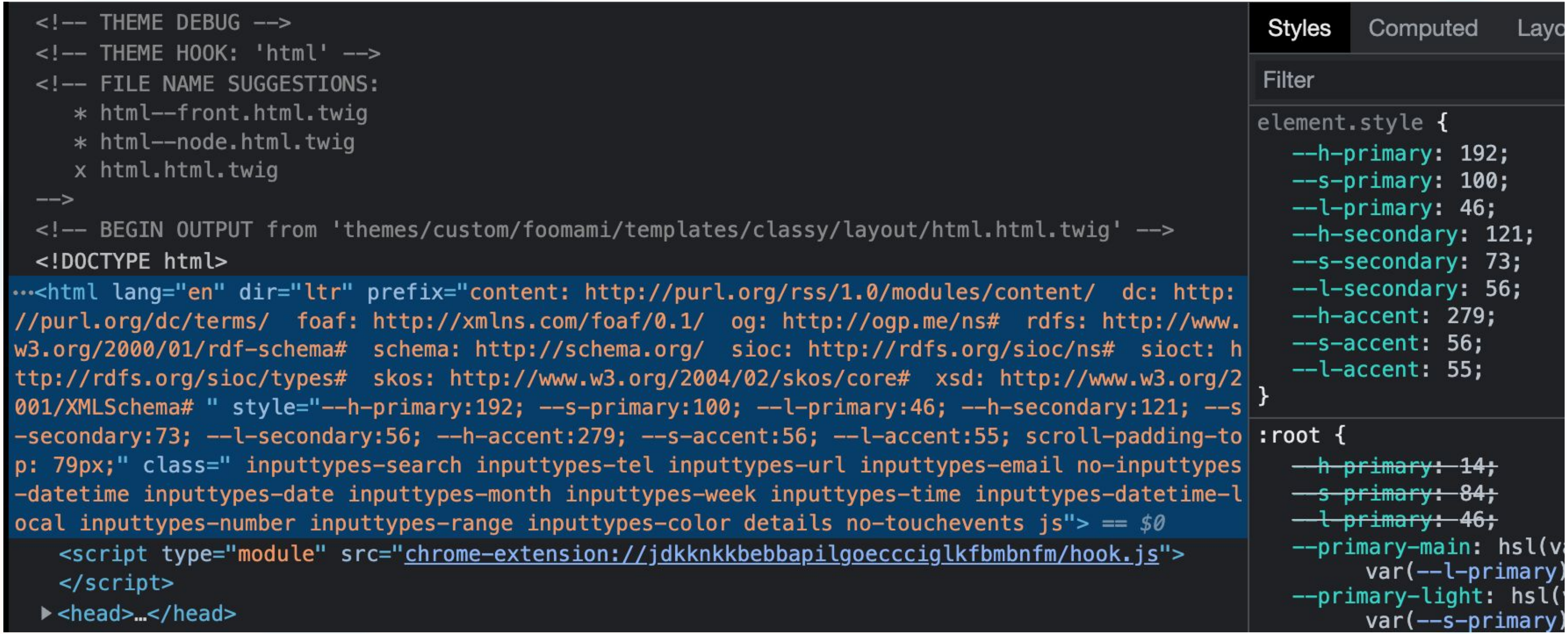
theme-settings-09

Back to the hooks



```php
function foomami_preprocess_html(&$variables) {
  $cssVars = [];
  $theme_colors = [
    'primary',
    'secondary',
    'accent',
  ];

  foreach ($theme_colors as $color_name) {
    $hex = theme_get_setting($color_name . '_color');

    if ($hex) {

      $hsl = _hexToHsl($hex);

      $cssVars[] = "--h-$color_name:$hsl[0]";
      $cssVars[] = "--s-$color_name:$hsl[1]";
      $cssVars[] = "--l-$color_name:$hsl[2]";
    }
  }

  $variables['html_attributes']->setAttribute('style', implode(';', $cssVars) . ';');
```

Back to the hooks



```
<!-- THEME DEBUG -->
<!-- THEME HOOK: 'html' -->
<!-- FILE NAME SUGGESTIONS:
   * html--front.html.twig
   * html--node.html.twig
   x html.html.twig
-->
<!-- BEGIN OUTPUT from 'themes/custom/foomami/templates/classy/layout/html.html.twig' -->
<!DOCTYPE html>
…<html lang="en" dir="ltr" prefix="content: http://purl.org/rss/1.0/modules/content/  dc: http:
//purl.org/dc/terms/  foaf: http://xmlns.com/foaf/0.1/  og: http://ogp.me/ns#  rdfs: http://www.
w3.org/2000/01/rdf-schema#  schema: http://schema.org/  sioc: http://rdfs.org/sioc/ns#  sioct: h
ttp://rdfs.org/sioc/types#  skos: http://www.w3.org/2004/02/skos/core#  xsd: http://www.w3.org/2
001/XMLSchema# " style="--h-primary:192; --s-primary:100; --l-primary:46; --h-secondary:121; --s
-secondary:73; --l-secondary:56; --h-accent:279; --s-accent:56; --l-accent:55; scroll-padding-to
p: 79px;" class=" inputtypes-search inputtypes-tel inputtypes-url inputtypes-email no-inputtypes
-datetime inputtypes-date inputtypes-month inputtypes-week inputtypes-time inputtypes-datetime-l
ocal inputtypes-number inputtypes-range inputtypes-color details no-touchevents js"> == $0
  <script type="module" src="chrome-extension://jdkknkkbebbapilgoecccciglkfbmbnfm/hook.js">
  </script>
 ▸<head>…</head>
```

```
element.style {
   --h-primary: 192;
   --s-primary: 100;
   --l-primary: 46;
   --h-secondary: 121;
   --s-secondary: 73;
   --l-secondary: 56;
   --h-accent: 279;
   --s-accent: 56;
   --l-accent: 55;
}
:root {
   ~~--h-primary: 14;~~
   ~~--s-primary: 84;~~
   ~~--l-primary: 46;~~
   --primary-main: hsl(v
      var(--l-primary)
   --primary-light: hsl(
      var(--s-primary)
```

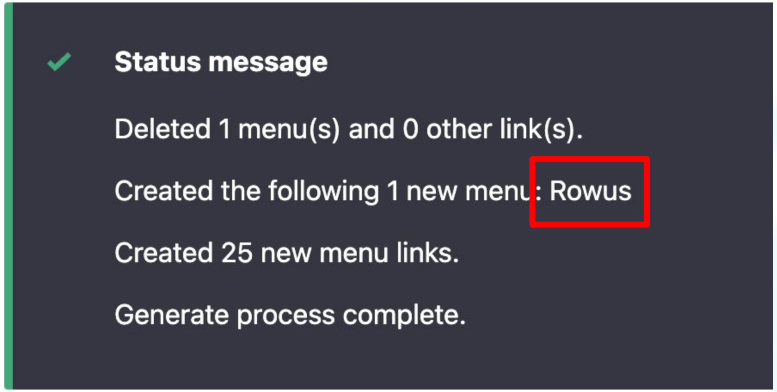
Menus

Create a new menu with depth

- Using devel_generate
  - `/admin/config/development/generate`

- 1 new menu
- 25 links
- Max length: 5
- Max depth: 4
- Max width: 4

- Note the name of the menu created



Status message

Deleted 1 menu(s) and 0 other link(s).

Created the following 1 new menu: Rowus

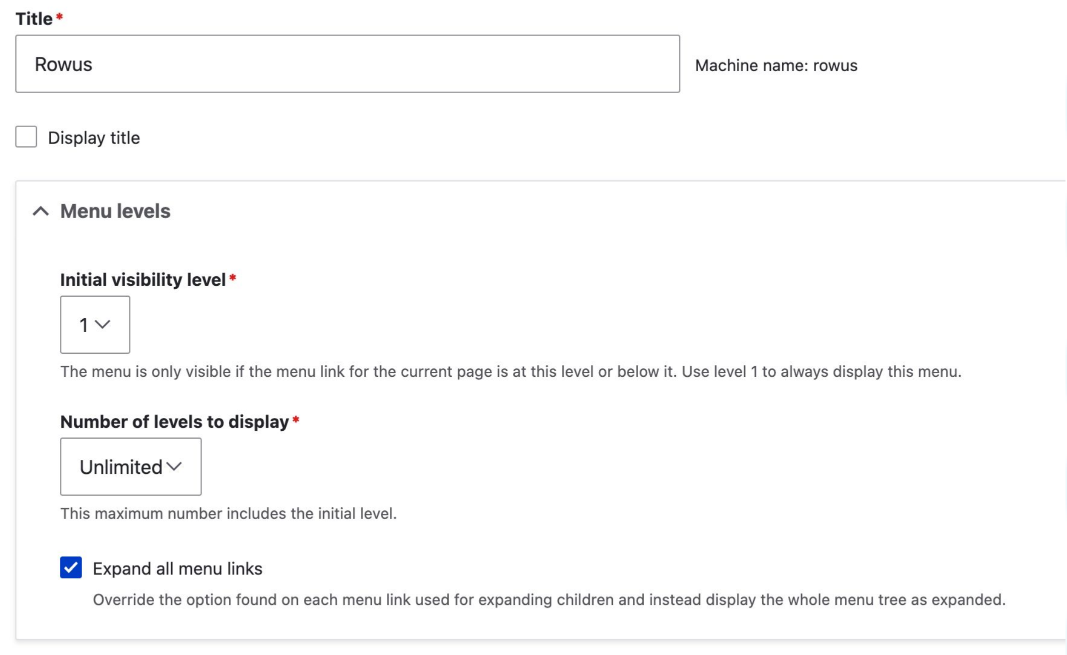
Created 25 new menu links.

Generate process complete.

Place your new menu

- Place new menu block directly above the main navigation menu block

- Set menu depth to unlimited
- Set expand all to true



Title *

Rowus

Machine name: rowus

□ Display title

Menu levels

Initial visibility level *

1

The menu is only visible if the menu link for the current page is at this level or below it. Use level 1 to always display this menu.

Number of levels to display *

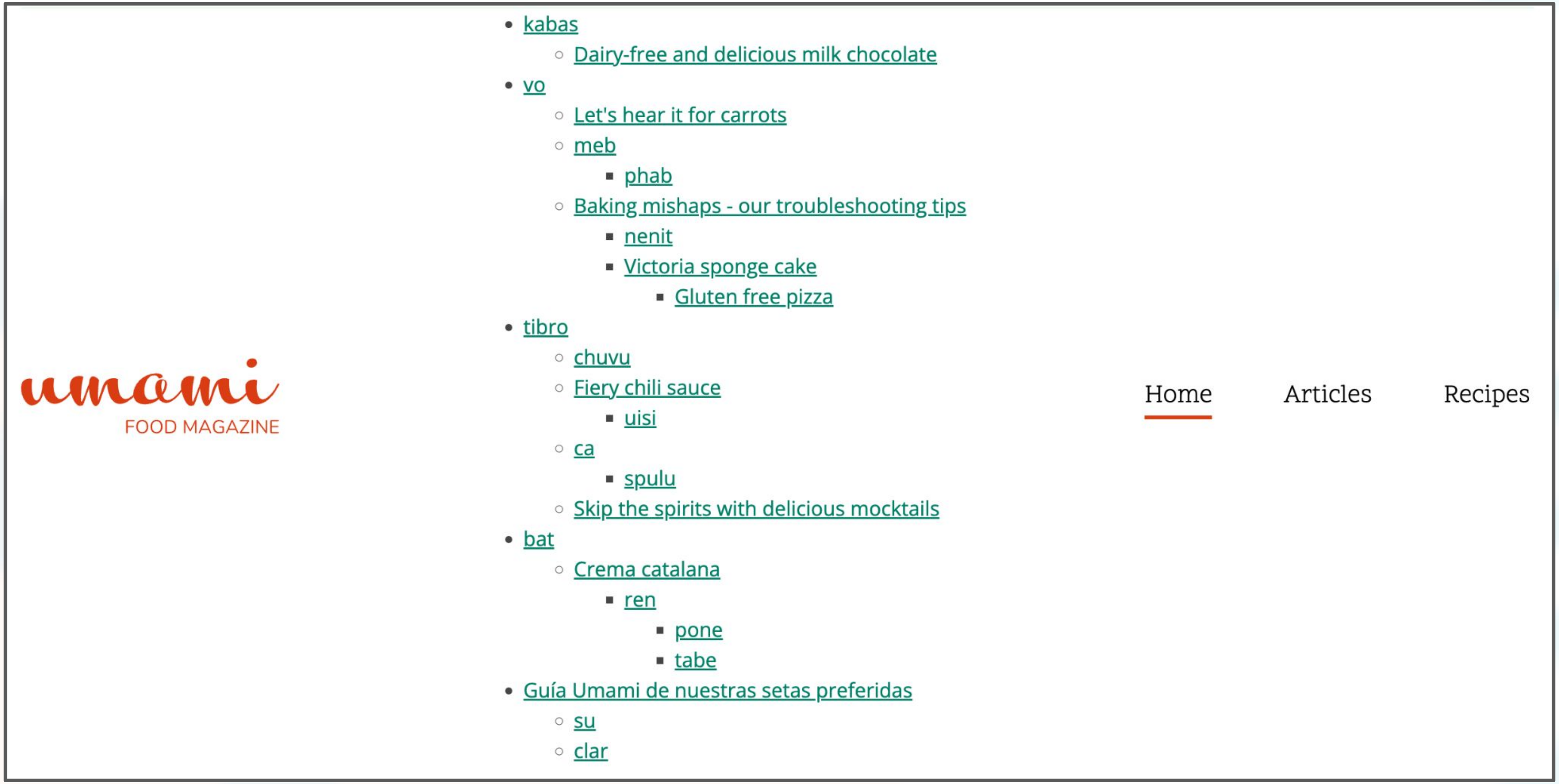
Unlimited

This maximum number includes the initial level.

☑ Expand all menu links

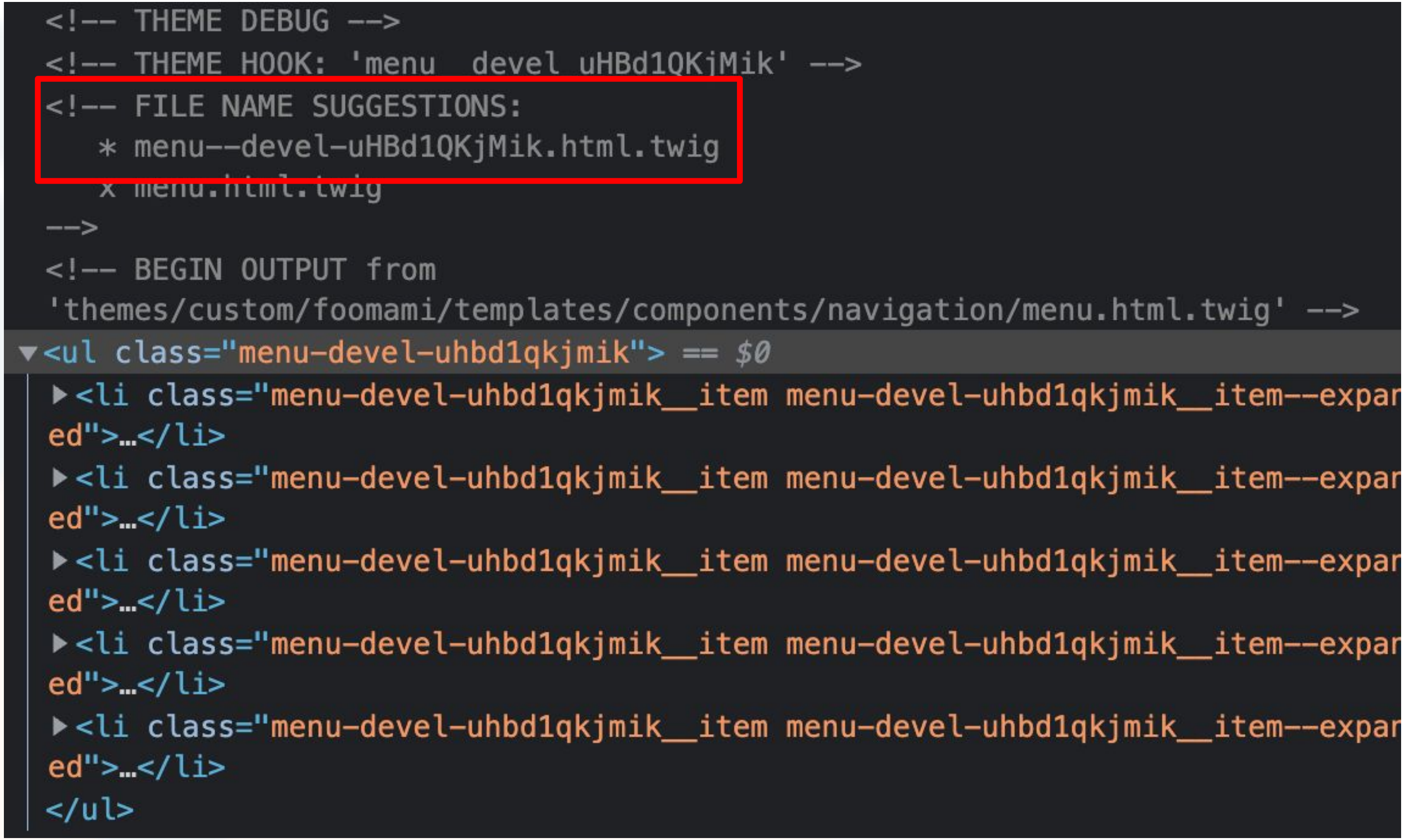
Override the option found on each menu link used for expanding children and instead display the whole menu tree as expanded.

Place your new menu



- kabas
  - Dairy-free and delicious milk chocolate
- vo
  - Let's hear it for carrots
  - meb
    - phab
  - Baking mishaps - our troubleshooting tips
    - nenit
    - Victoria sponge cake
      - Gluten free pizza
- tibro
  - chuvu
  - Fiery chili sauce
    - uisi
  - ca
    - spulu
  - Skip the spirits with delicious mocktails
- bat
  - Crema catalana
    - ren
      - pone
      - tabe
- Guía Umami de nuestras setas preferidas
  - su
  - clar

Place your new menu



```
<!-- THEME DEBUG -->
<!-- THEME HOOK: 'menu__devel_uHBd1QKjMik' -->
<!-- FILE NAME SUGGESTIONS:
   * menu--devel-uHBd1QKjMik.html.twig
   x menu.html.twig
-->
<!-- BEGIN OUTPUT from
'themes/custom/foomami/templates/components/navigation/menu.html.twig' -->
▼<ul class="menu-devel-uhbd1qkjmik"> == $0
  ▶<li class="menu-devel-uhbd1qkjmik__item menu-devel-uhbd1qkjmik__item--expan
  ed">…</li>
  ▶<li class="menu-devel-uhbd1qkjmik__item menu-devel-uhbd1qkjmik__item--expan
  ed">…</li>
  ▶<li class="menu-devel-uhbd1qkjmik__item menu-devel-uhbd1qkjmik__item--expan
  ed">…</li>
  ▶<li class="menu-devel-uhbd1qkjmik__item menu-devel-uhbd1qkjmik__item--expan
  ed">…</li>
  ▶<li class="menu-devel-uhbd1qkjmik__item menu-devel-uhbd1qkjmik__item--expan
  ed">…</li>
</ul>
```

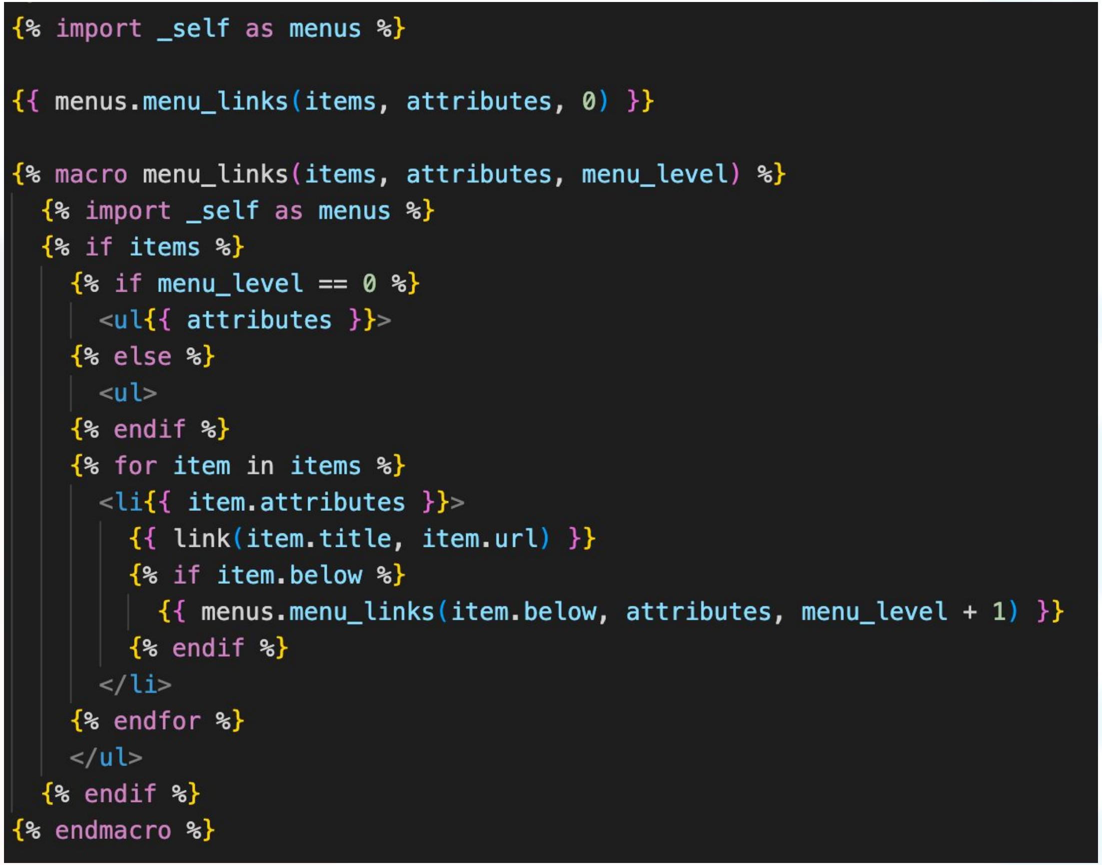
Make our template file

- Create a new file with your suggested template name
- foomami/templates/components/navigation/menu...html.twig
  - This follows the pattern that already exists
  - Drupal will find the template anywhere inside the templates folder
- Contents copied from the stable9 theme
  - It’s the most bare-bones theme, so it’s where I like to get my starting point when I make a new template
  - web/core/themes/stable9/templates/navigation/menu.html.twig

menu-01

Make our template file

- We define a new macro
  - Function
  - Typically recursive
- We import macros from `_self`
  - This file
- Alias all macros to `menus`
- Call the menu_links macro with starting conditions
- **The template I added to the repo will almost certainly not work for your site. It's reliant on the devel-menu id that's auto-generated.**

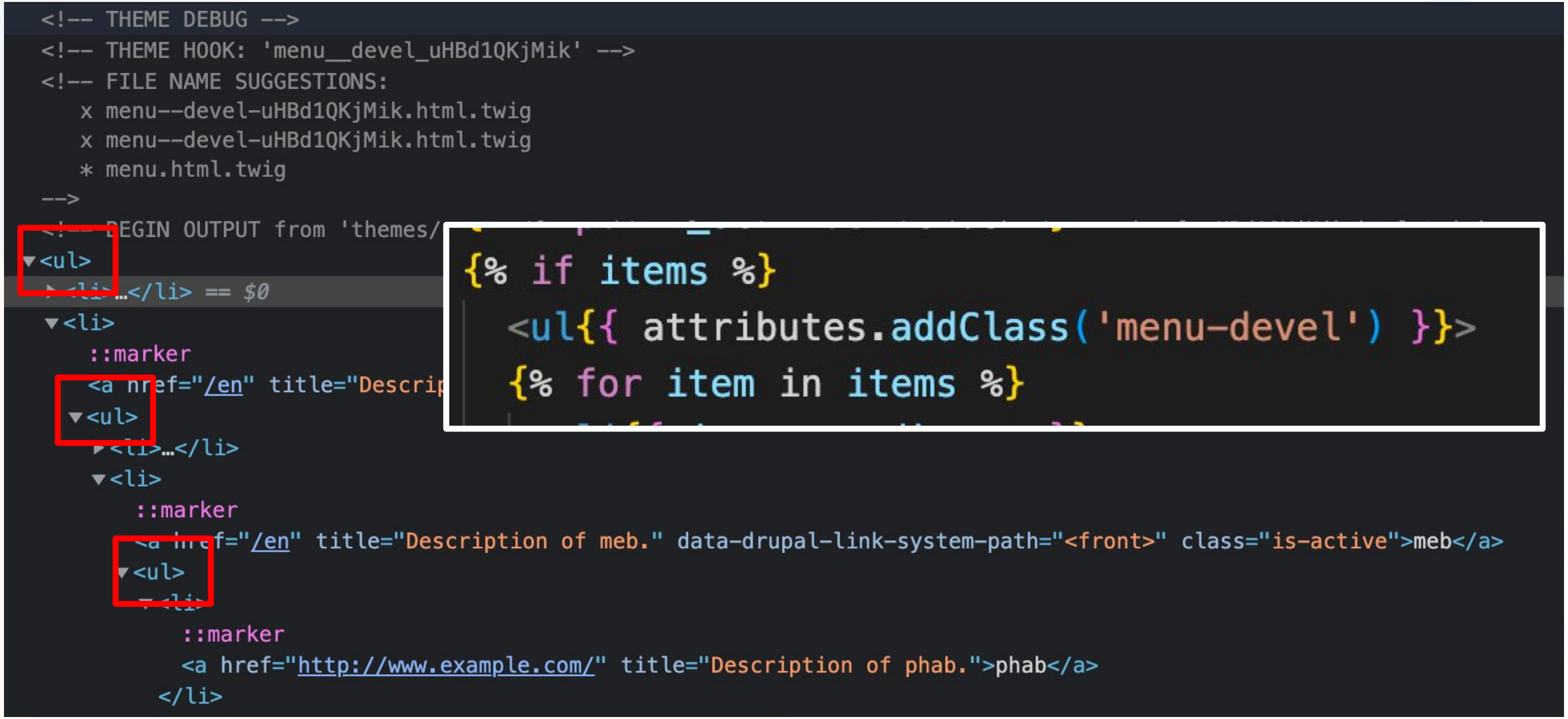


```
{% import _self as menus %}

{{ menus.menu_links(items, attributes, 0) }}

{% macro menu_links(items, attributes, menu_level) %}
  {% import _self as menus %}
  {% if items %}
    {% if menu_level == 0 %}
      <ul{{ attributes }}>
    {% else %}
      <ul>
    {% endif %}
    {% for item in items %}
      <li{{ item.attributes }}>
        {{ link(item.title, item.url) }}
        {% if item.below %}
          {{ menus.menu_links(item.below, attributes, menu_level + 1) }}
        {% endif %}
      </li>
    {% endfor %}
    </ul>
  {% endif %}
{% endmacro %}
```

menu-02

Make our template file

- No difference between ul elements
  - We can simplify



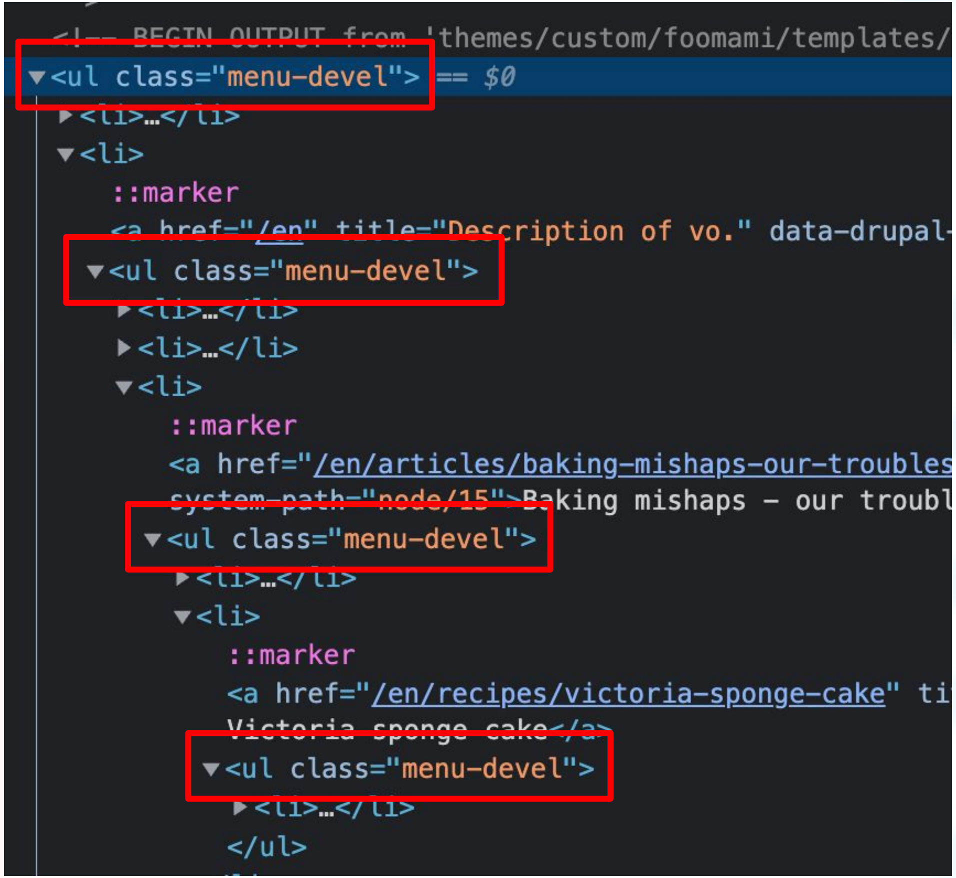
```
<!-- THEME DEBUG -->
<!-- THEME HOOK: 'menu__devel_uHBd1QKjMik' -->
<!-- FILE NAME SUGGESTIONS:
   x menu--devel-uHBd1QKjMik.html.twig
   x menu--devel-uHBd1QKjMik.html.twig
   * menu.html.twig
-->
<!-- BEGIN OUTPUT from 'themes/
▼<ul>
  ▶<li>…</li> == $0
  ▼<li>
      ::marker
      <a href="/en" title="Descrip
    ▼<ul>
      ▶<li>…</li>
      ▼<li>
          ::marker
          <a href="/en" title="Description of meb." data-drupal-link-system-path="<front>" class="is-active">meb</a>
        ▼<ul>
          ▼<li>
              ::marker
              <a href="http://www.example.com/" title="Description of phab.">phab</a>
            </li>
```

```
{% if items %}
  <ul{{ attributes.addClass('menu-devel') }}>
  {% for item in items %}
```

Lessons learned

- Attributes is mutable
  - We'll need to remove any changes we make before kicking off the recursion

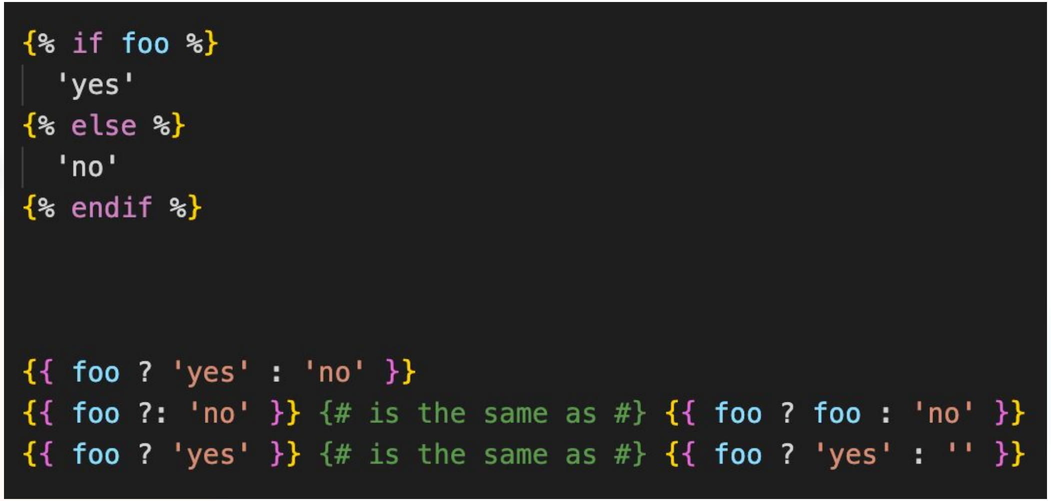
- To make this easier, we'll put classes in an array and make them conditional with ternary null coalescing operators



```
<!-- BEGIN OUTPUT from 'themes/custom/foomami/templates/
▼<ul class="menu-devel"> == $0
  ▶<li>…</li>
  ▼<li>
     ::marker
     <a href="/en" title="Description of vo." data-drupal-
    ▼<ul class="menu-devel">
      ▶<li>…</li>
      ▶<li>…</li>
      ▼<li>
         ::marker
         <a href="/en/articles/baking-mishaps-our-troubles
         system-path="node/15">Baking mishaps - our troubl
        ▼<ul class="menu-devel">
          ▶<li>…</li>
          ▼<li>
             ::marker
             <a href="/en/recipes/victoria-sponge-cake" ti
             Victoria sponge cake</a>
            ▼<ul class="menu-devel">
              ▶<li>…</li>
             </ul>
```

Ternary Operator

- Useful when you want to use a boolean value as a switch
- Can be used within:
  - print statements {{ }}
  - Set statements {% set %}

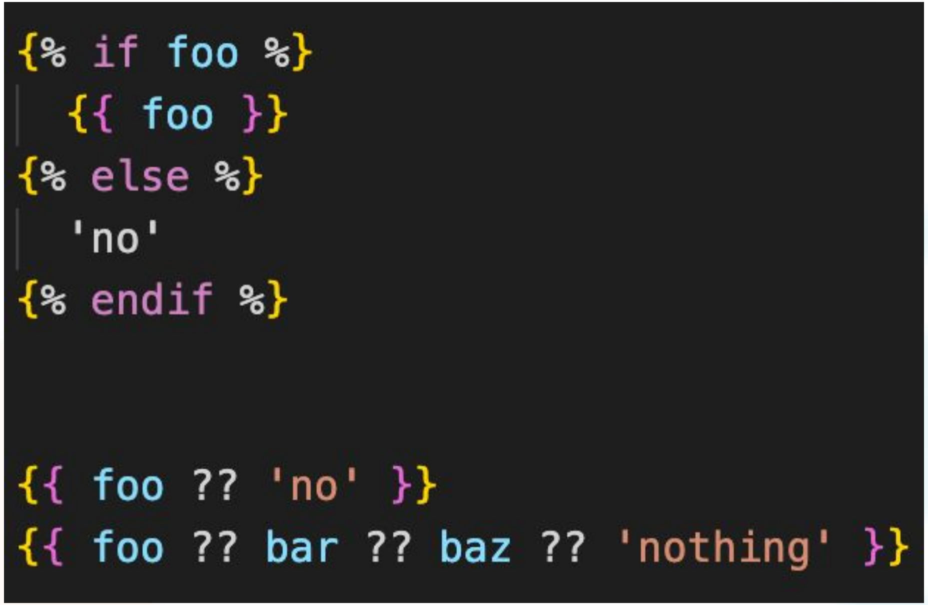


```
{% if foo %}
  'yes'
{% else %}
  'no'
{% endif %}



{{ foo ? 'yes' : 'no' }}
{{ foo ?: 'no' }} {# is the same as #} {{ foo ? foo : 'no' }}
{{ foo ? 'yes' }} {# is the same as #} {{ foo ? 'yes' : '' }}
```

Null Coalescing Operator

- Useful when you want to use a value with one or more fallbacks

- Can be used within:
  - print statements {{ }}
  - Set statements {% set %}



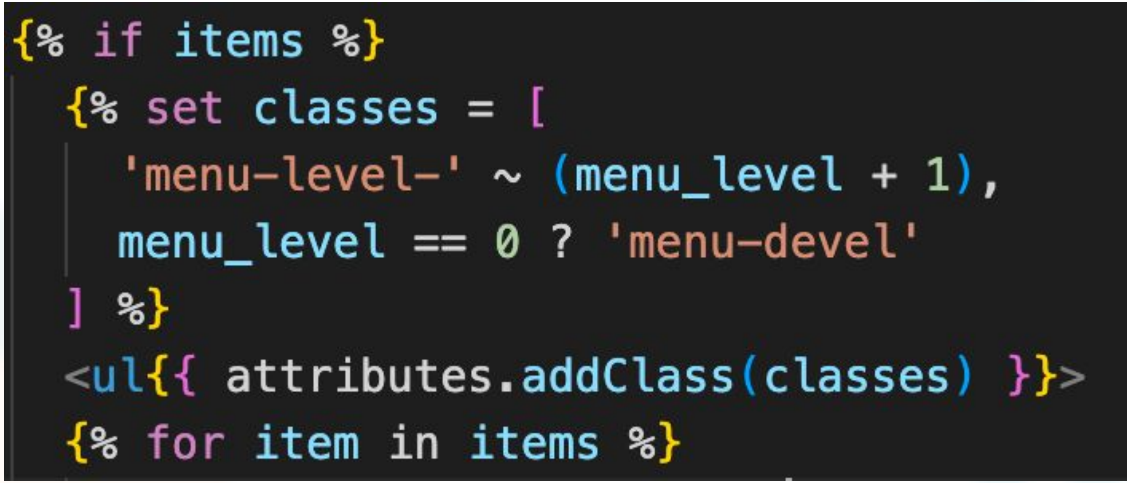
```
{% if foo %}
  {{ foo }}
{% else %}
  'no'
{% endif %}


{{ foo ?? 'no' }}
{{ foo ?? bar ?? baz ?? 'nothing' }}
```

menu-03

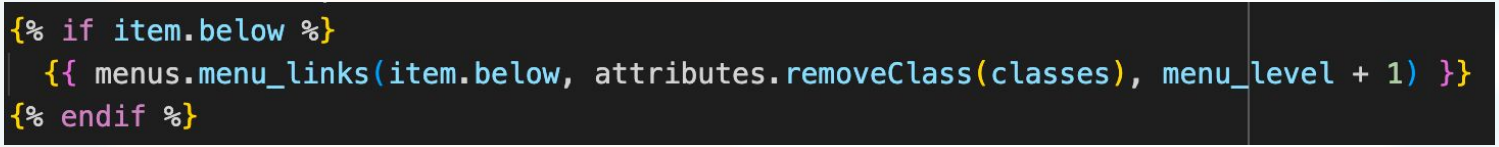
Make our template file

- Create classes array
- Add dynamic classes
- Add conditional classes

- Remove classes from attributes object before cycling deeper

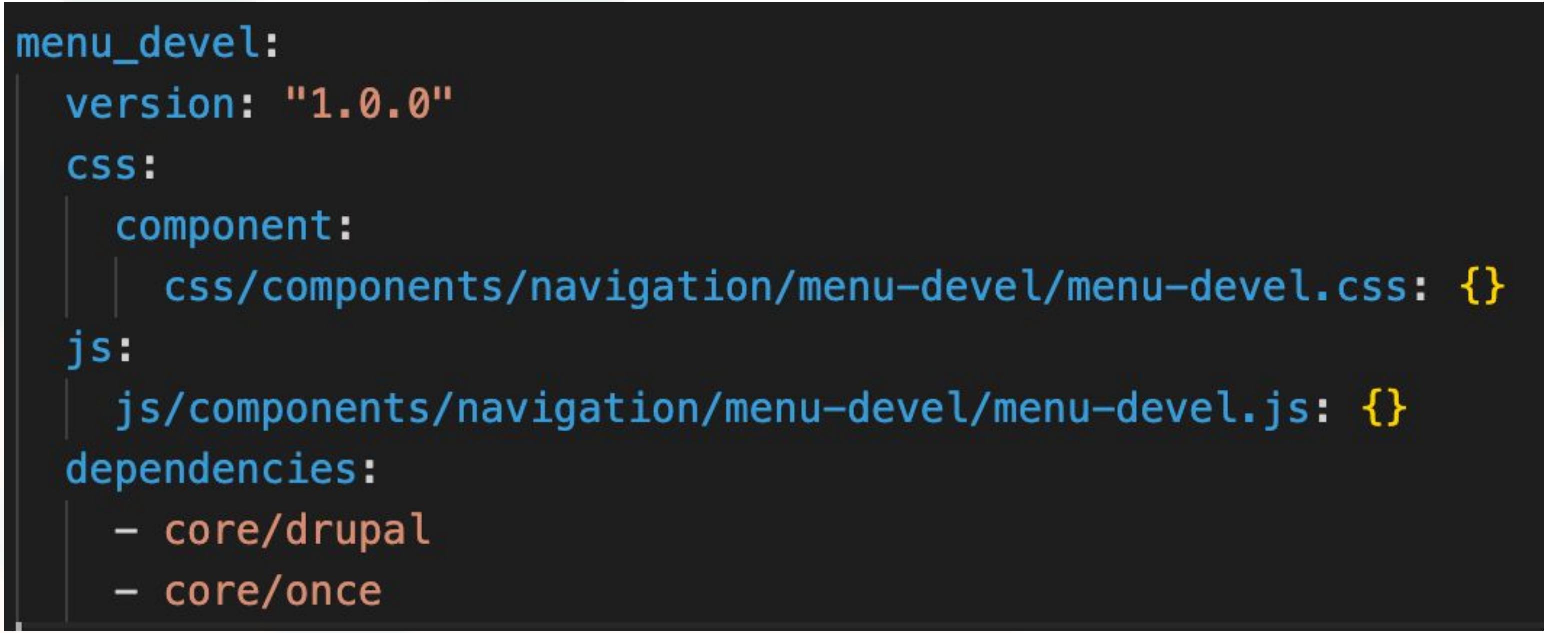


```
{% if items %}
  {% set classes = [
    'menu-level-' ~ (menu_level + 1),
    menu_level == 0 ? 'menu-devel'
  ] %}
  <ul{{ attributes.addClass(classes) }}>
  {% for item in items %}
```



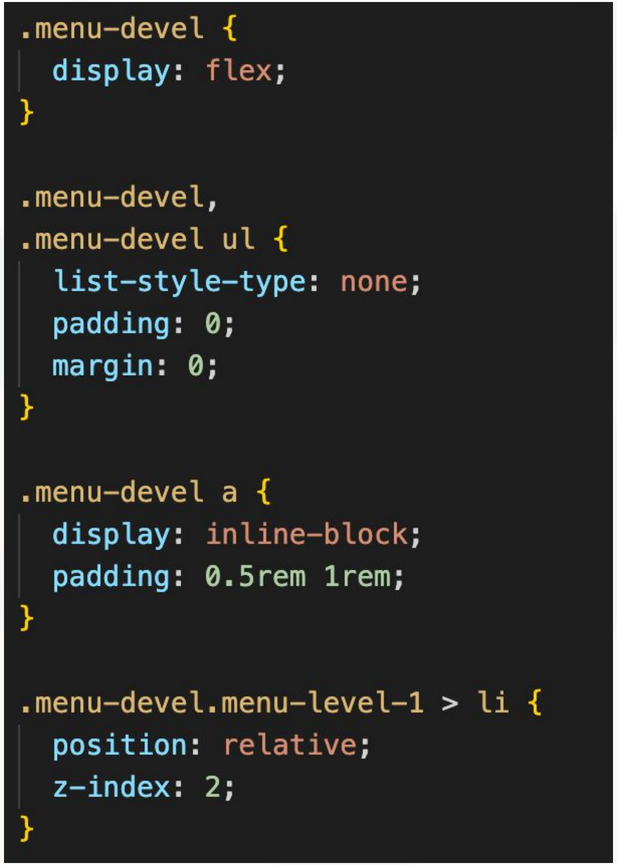
```
{% if item.below %}
  {{ menus.menu_links(item.below, attributes.removeClass(classes), menu_level + 1) }}
{% endif %}
```

Lets polish it up!

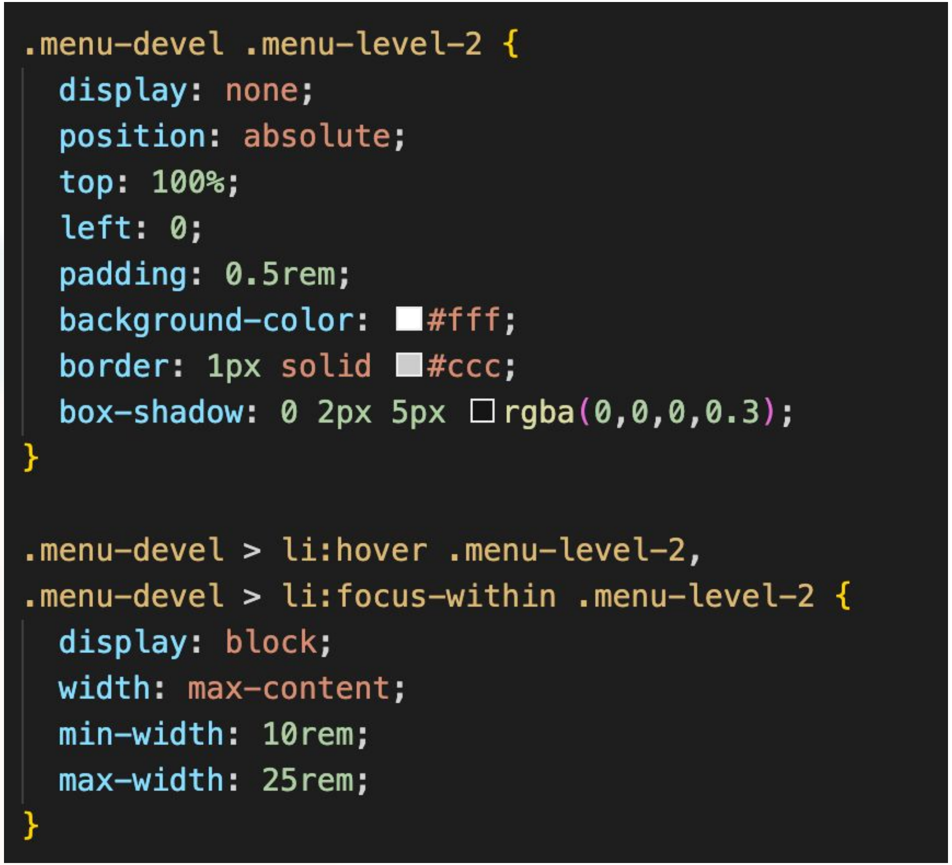


```yaml
menu_devel:
  version: "1.0.0"
  css:
    component:
      css/components/navigation/menu-devel/menu-devel.css: {}
  js:
    js/components/navigation/menu-devel/menu-devel.js: {}
  dependencies:
    - core/drupal
    - core/once
```

Lets polish it up!



```css
.menu-devel {
  display: flex;
}

.menu-devel,
.menu-devel ul {
  list-style-type: none;
  padding: 0;
  margin: 0;
}

.menu-devel a {
  display: inline-block;
  padding: 0.5rem 1rem;
}

.menu-devel.menu-level-1 > li {
  position: relative;
  z-index: 2;
}
```



```css
.menu-devel .menu-level-2 {
  display: none;
  position: absolute;
  top: 100%;
  left: 0;
  padding: 0.5rem;
  background-color: #fff;
  border: 1px solid #ccc;
  box-shadow: 0 2px 5px rgba(0,0,0,0.3);
}

.menu-devel > li:hover .menu-level-2,
.menu-devel > li:focus-within .menu-level-2 {
  display: block;
  width: max-content;
  min-width: 10rem;
  max-width: 25rem;
}
```

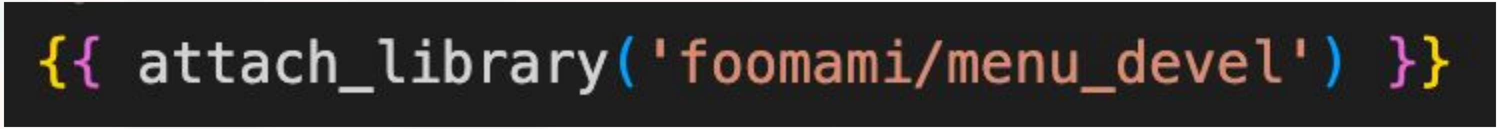
menu-04

Why isn’t it showing up?

- We created a library and registered our stylesheet

- We didn’t attach it yet!

- Let’s go ahead and remove the old menu block while we’re here



```
{{ attach_library('foomami/menu_devel') }}
```

Template Customization

The last hook type

- We’ve already seen form alters and preprocesses
- The last hook type we have the a theme suggestion alter
  - Provides new template names that we can use to customize markup based on content or configuration
- The ideal use of these is when entities with identical data structures need different markup based on their content or config.

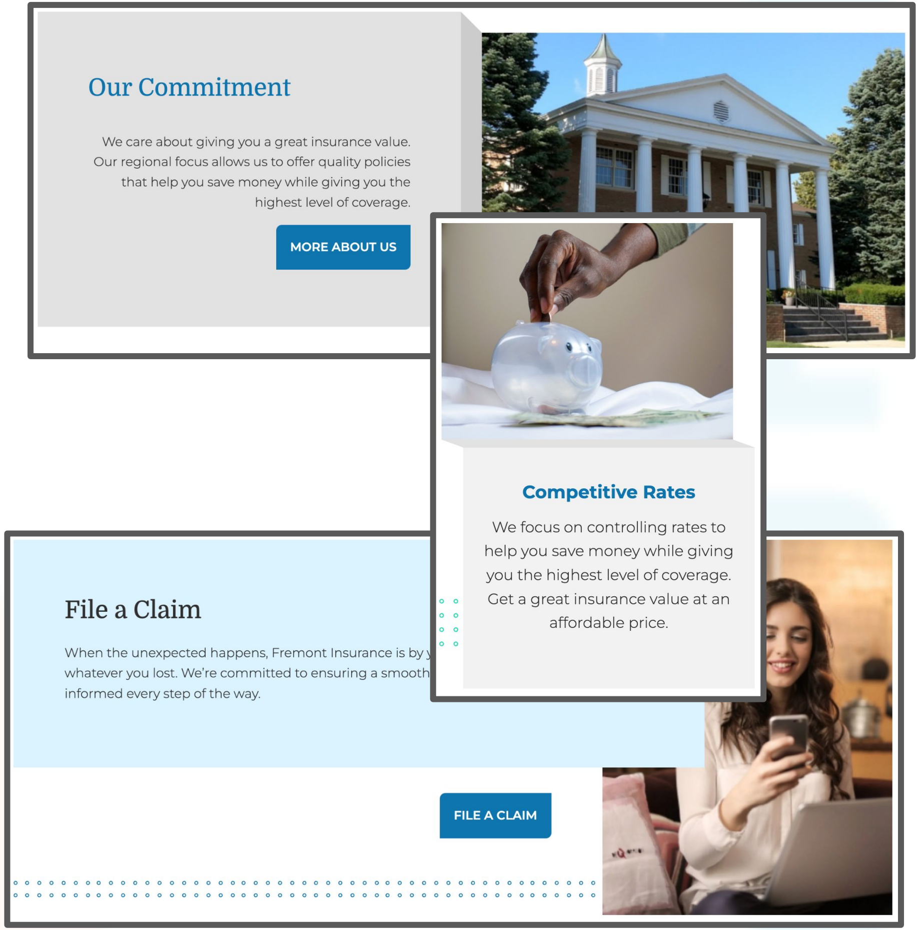
The last hook type

- We’ve already seen form alters and preprocesses

- The last hook type we have the a theme suggestion alter
  - Provides new template names that we can use to customize markup based on content or configuration

- The ideal use of these is when entities with identical data structures need different markup based on their content or config.

- Example: Fremont Insurance Company
  - fmic.com
  - Gravity Works



Our Commitment

We care about giving you a great insurance value. Our regional focus allows us to offer quality policies that help you save money while giving you the highest level of coverage.

MORE ABOUT US

Competitive Rates

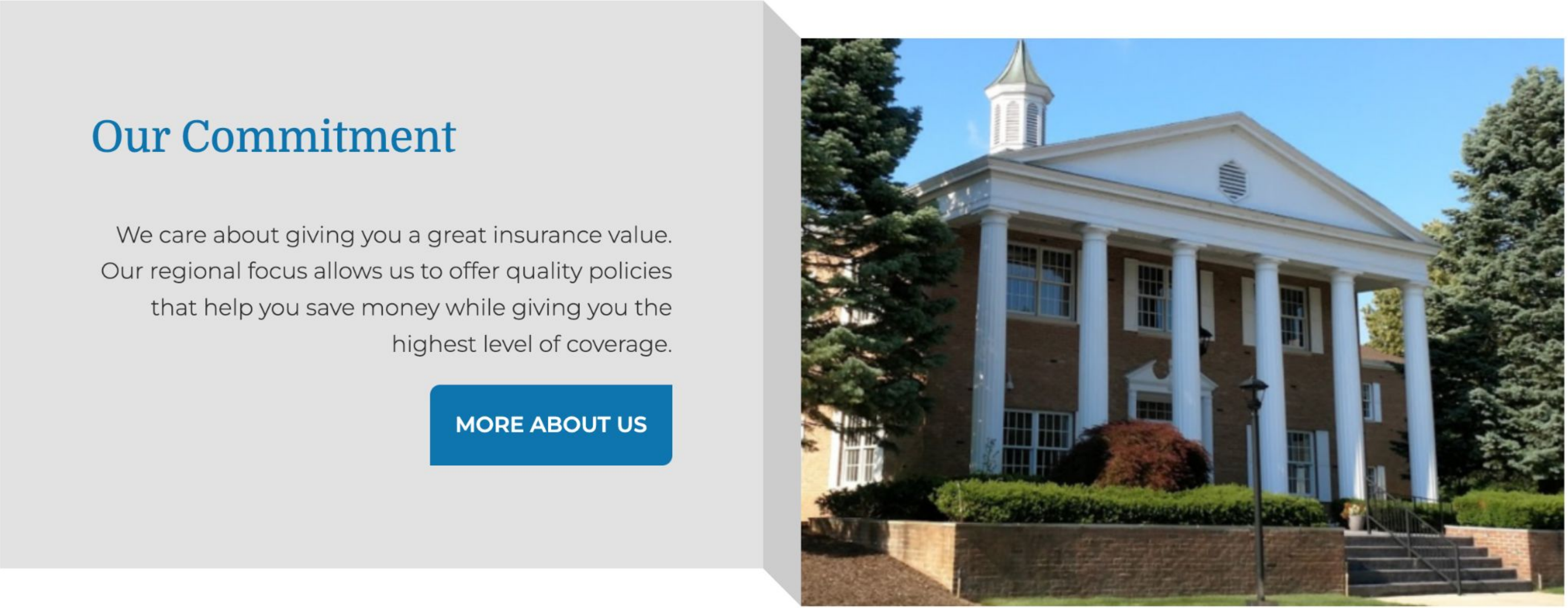
We focus on controlling rates to help you save money while giving you the highest level of coverage. Get a great insurance value at an affordable price.

File a Claim

When the unexpected happens, Fremont Insurance is by y
whatever you lost. We’re committed to ensuring a smooth
informed every step of the way.

FILE A CLAIM

FMIC Paragraph Templates



Our Commitment

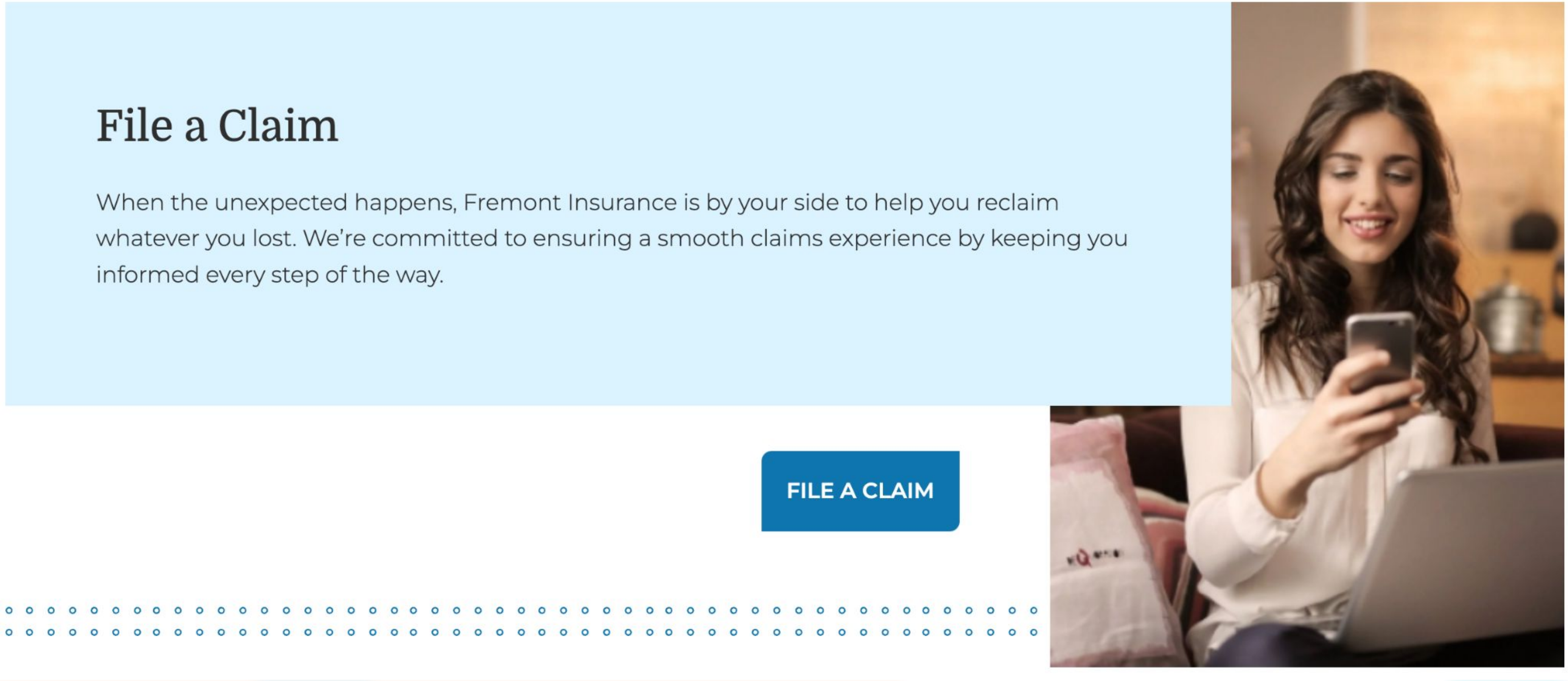
We care about giving you a great insurance value. Our regional focus allows us to offer quality policies that help you save money while giving you the highest level of coverage.

FMIC Paragraph Templates

Competitive Rates

We focus on controlling rates to help you save money while giving you the highest level of coverage. Get a great insurance value at an affordable price.

FMIC Paragraph Templates

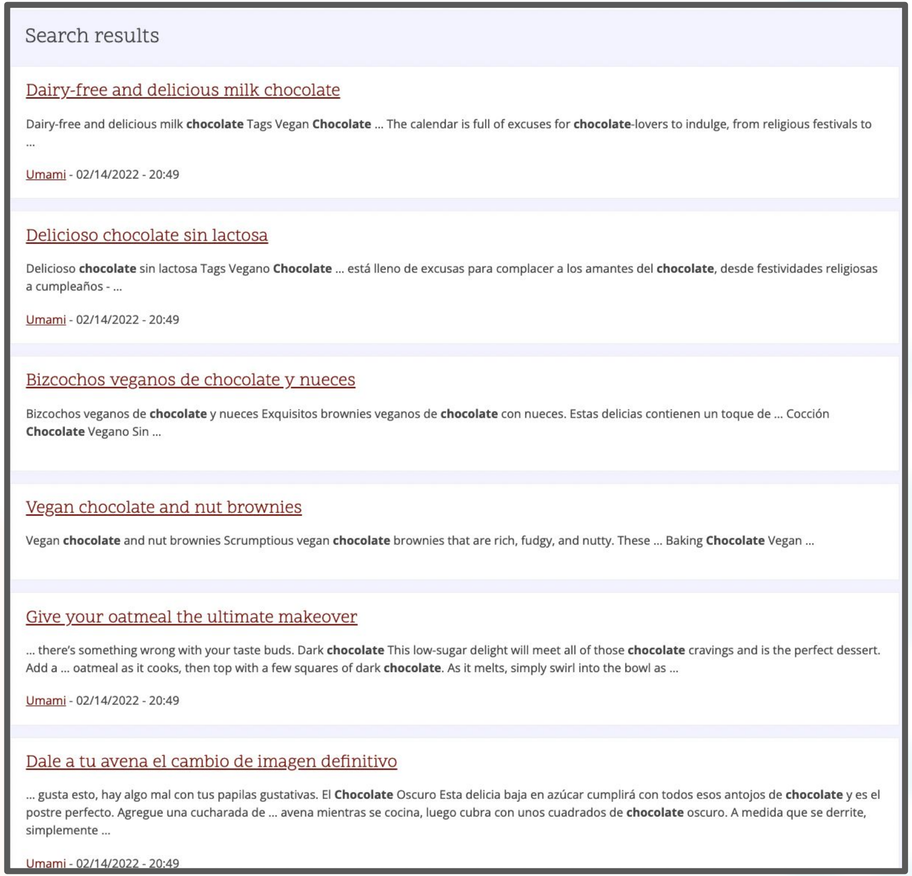


File a Claim

When the unexpected happens, Fremont Insurance is by your side to help you reclaim whatever you lost. We're committed to ensuring a smooth claims experience by keeping you informed every step of the way.

Search Results

- Site header search combines all node types
- Articles & Recipes are difficult to tell apart
- Let's make some new template suggestions so we can visually differentiate for our users
- What kind of content/config differences can we leverage?



Search results

Dairy-free and delicious milk chocolate

Dairy-free and delicious milk **chocolate** Tags Vegan **Chocolate** ... The calendar is full of excuses for **chocolate**-lovers to indulge, from religious festivals to ...

Umami - 02/14/2022 - 20:49

Delicioso chocolate sin lactosa

Delicioso **chocolate** sin lactosa Tags Vegano **Chocolate** ... está lleno de excusas para complacer a los amantes del **chocolate**, desde festividades religiosas a cumpleaños - ...

Umami - 02/14/2022 - 20:49

Bizcochos veganos de chocolate y nueces

Bizcochos veganos de **chocolate** y nueces Exquisitos brownies veganos de **chocolate** con nueces. Estas delicias contienen un toque de ... Cocción **Chocolate** Vegano Sin ...

Vegan chocolate and nut brownies

Vegan **chocolate** and nut brownies Scrumptious vegan **chocolate** brownies that are rich, fudgy, and nutty. These ... Baking **Chocolate** Vegan ...

Give your oatmeal the ultimate makeover

... there's something wrong with your taste buds. Dark **chocolate** This low-sugar delight will meet all of those **chocolate** cravings and is the perfect dessert. Add a ... oatmeal as it cooks, then top with a few squares of dark **chocolate**. As it melts, simply swirl into the bowl as ...

Umami - 02/14/2022 - 20:49

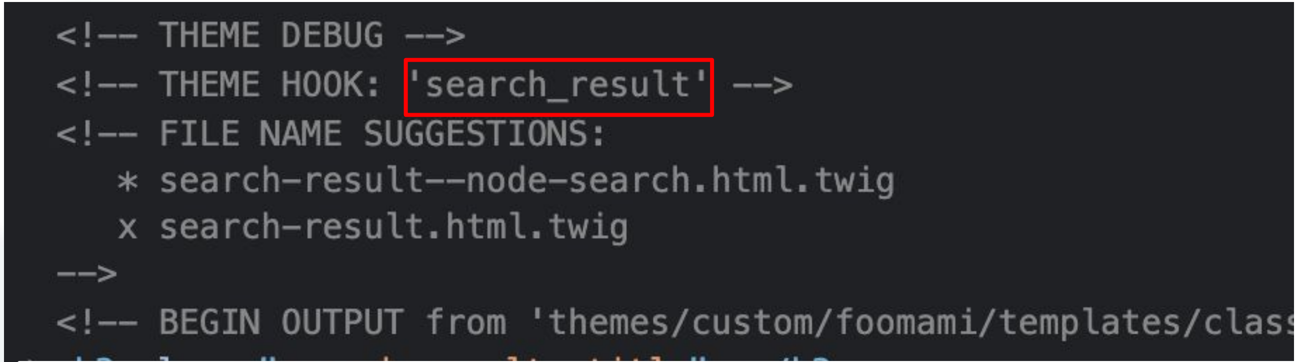
Dale a tu avena el cambio de imagen definitivo

... gusta esto, hay algo mal con tus papilas gustativas. El **Chocolate** Oscuro Esta delicia baja en azúcar cumplirá con todos esos antojos de **chocolate** y es el postre perfecto. Agregue una cucharada de ... avena mientras se cocina, luego cubra con unos cuadrados de **chocolate** oscuro. A medida que se derrite, simplemente ...

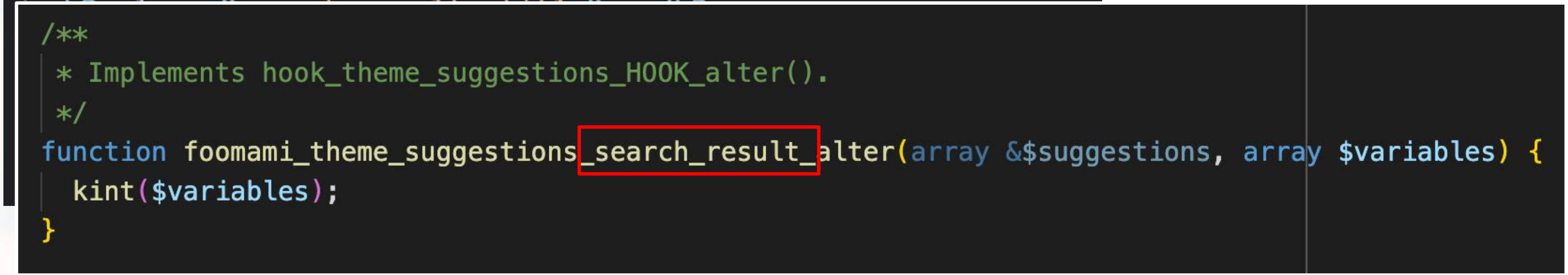
Umami - 02/14/2022 - 20:49

templates-01

Search Results

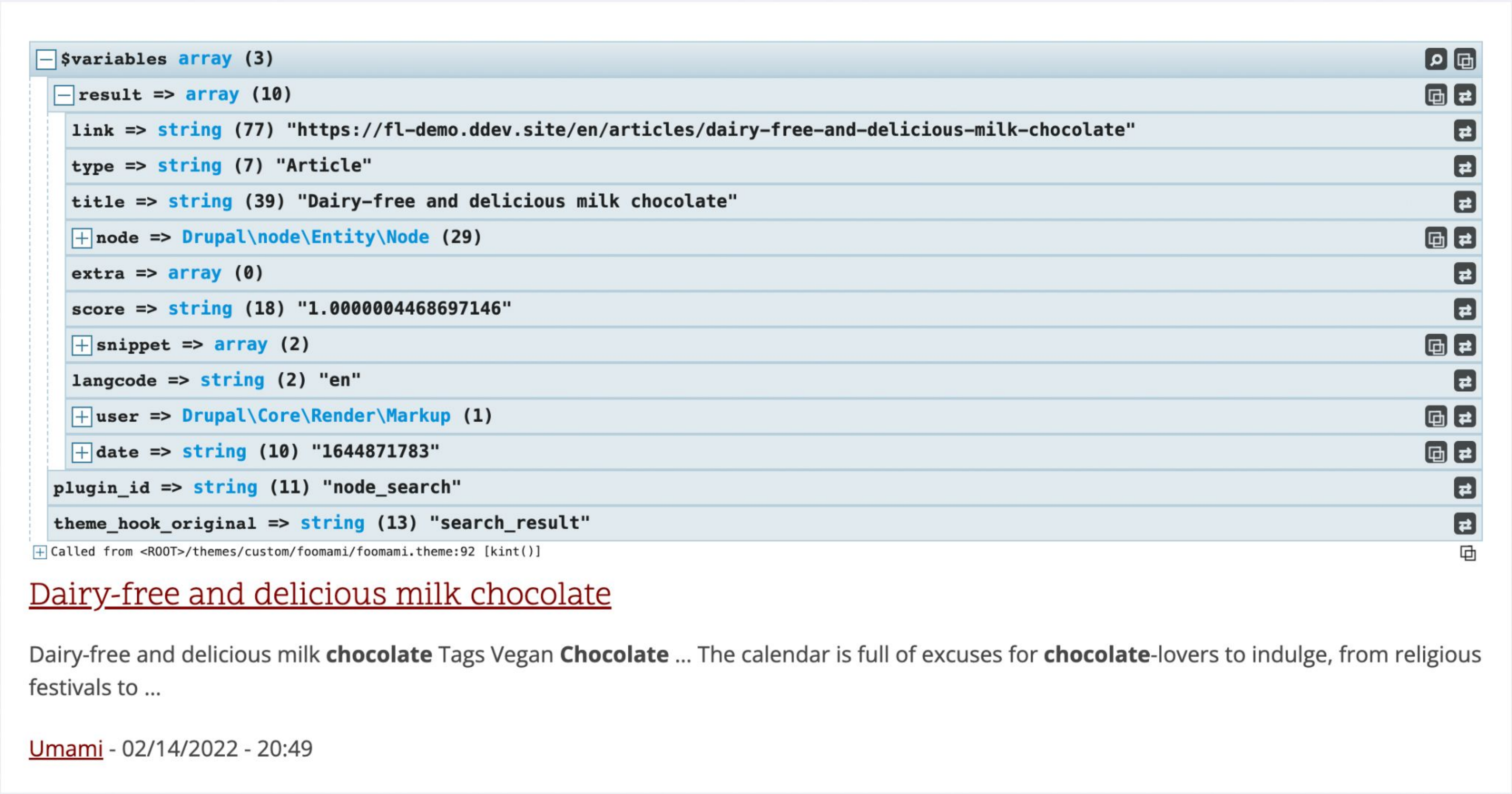


```
<!-- THEME DEBUG -->
<!-- THEME HOOK: 'search_result' -->
<!-- FILE NAME SUGGESTIONS:
   * search-result--node-search.html.twig
   x search-result.html.twig
-->
<!-- BEGIN OUTPUT from 'themes/custom/foomami/templates/class
```



```
/**
 * Implements hook_theme_suggestions_HOOK_alter().
 */
function foomami_theme_suggestions_search_result_alter(array &$suggestions, array $variables) {
  kint($variables);
}
```

Search Results



```
⊟ $variables array (3)
  ⊟ result => array (10)
      link => string (77) "https://fl-demo.ddev.site/en/articles/dairy-free-and-delicious-milk-chocolate"
      type => string (7) "Article"
      title => string (39) "Dairy-free and delicious milk chocolate"
    ⊞ node => Drupal\node\Entity\Node (29)
      extra => array (0)
      score => string (18) "1.0000004468697146"
    ⊞ snippet => array (2)
      langcode => string (2) "en"
    ⊞ user => Drupal\Core\Render\Markup (1)
    ⊞ date => string (10) "1644871783"
    plugin_id => string (11) "node_search"
    theme_hook_original => string (13) "search_result"
⊞ Called from <ROOT>/themes/custom/foomami/foomami.theme:92 [kint()]
```

Dairy-free and delicious milk chocolate

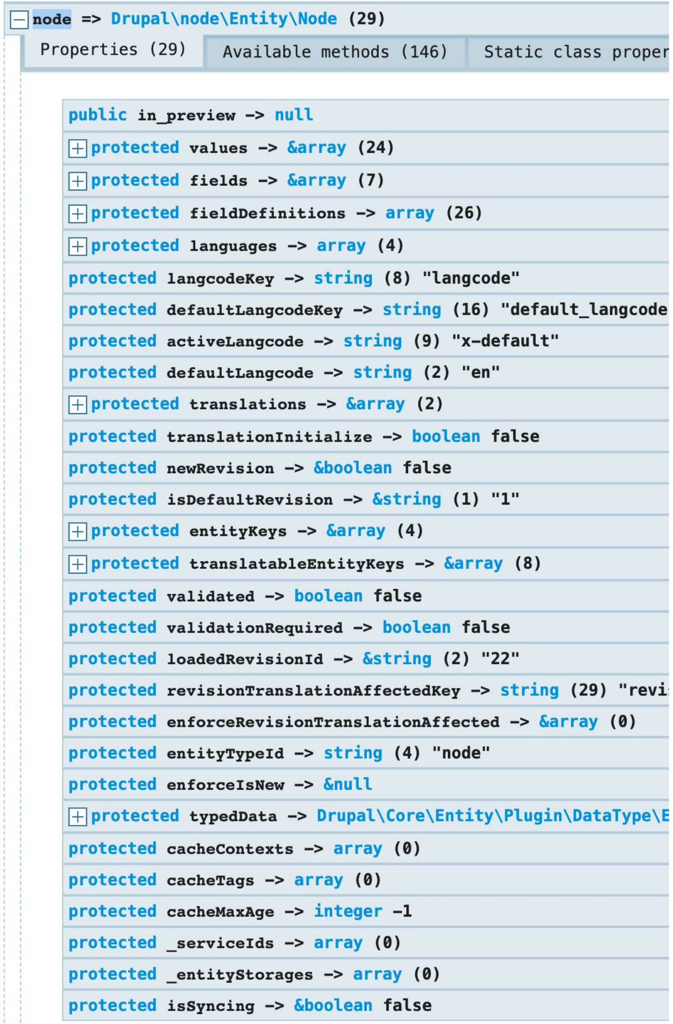
Dairy-free and delicious milk **chocolate** Tags Vegan **Chocolate** ... The calendar is full of excuses for **chocolate**-lovers to indulge, from religious festivals to ...

Umami - 02/14/2022 - 20:49

Search Results

- Obvious choice is ‘type’
  - Note that it’s capitalized. This is the human-readable name. Do not use!

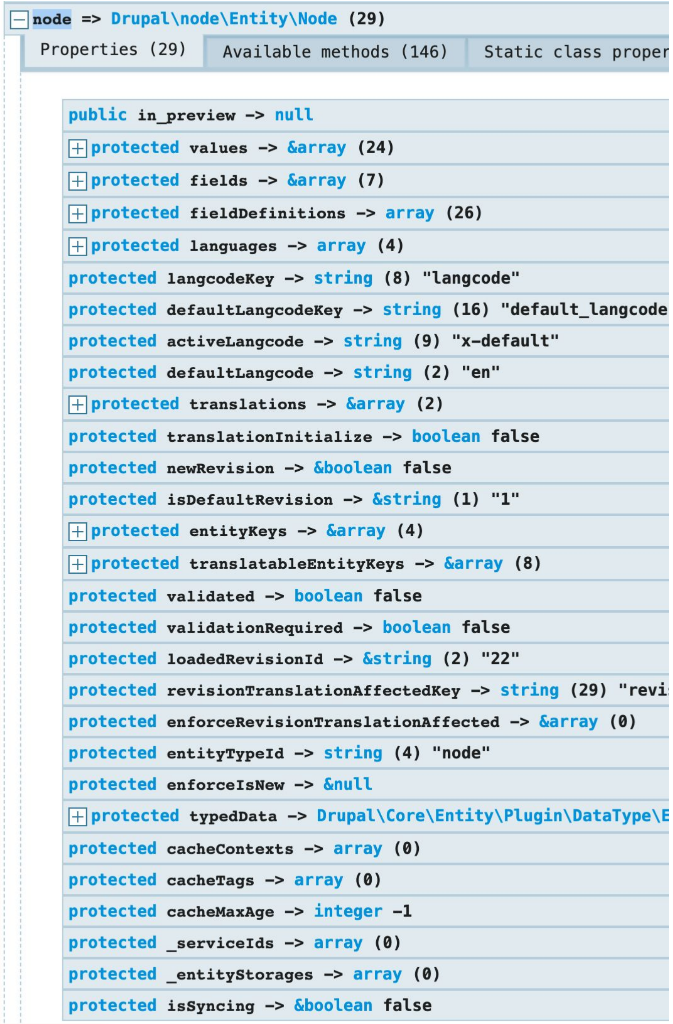
- Node object provides access to all field values and entity properties
  - We can use stored field values directly
  - We can also modify them if we want!



```
⊟node => Drupal\node\Entity\Node (29)
  Properties (29)   Available methods (146)   Static class proper

    public in_preview -> null
    ⊞protected values -> &array (24)
    ⊞protected fields -> &array (7)
    ⊞protected fieldDefinitions -> array (26)
    ⊞protected languages -> array (4)
    protected langcodeKey -> string (8) "langcode"
    protected defaultLangcodeKey -> string (16) "default_langcode
    protected activeLangcode -> string (9) "x-default"
    protected defaultLangcode -> string (2) "en"
    ⊞protected translations -> &array (2)
    protected translationInitialize -> boolean false
    protected newRevision -> &boolean false
    protected isDefaultRevision -> &string (1) "1"
    ⊞protected entityKeys -> &array (4)
    ⊞protected translatableEntityKeys -> &array (8)
    protected validated -> boolean false
    protected validationRequired -> boolean false
    protected loadedRevisionId -> &string (2) "22"
    protected revisionTranslationAffectedKey -> string (29) "revi
    protected enforceRevisionTranslationAffected -> &array (0)
    protected entityTypeId -> string (4) "node"
    protected enforceIsNew -> &null
    ⊞protected typedData -> Drupal\Core\Entity\Plugin\DataType\E
    protected cacheContexts -> array (0)
    protected cacheTags -> array (0)
    protected cacheMaxAge -> integer -1
    protected _serviceIds -> array (0)
    protected _entityStorages -> array (0)
    protected isSyncing -> &boolean false
```

Search Results

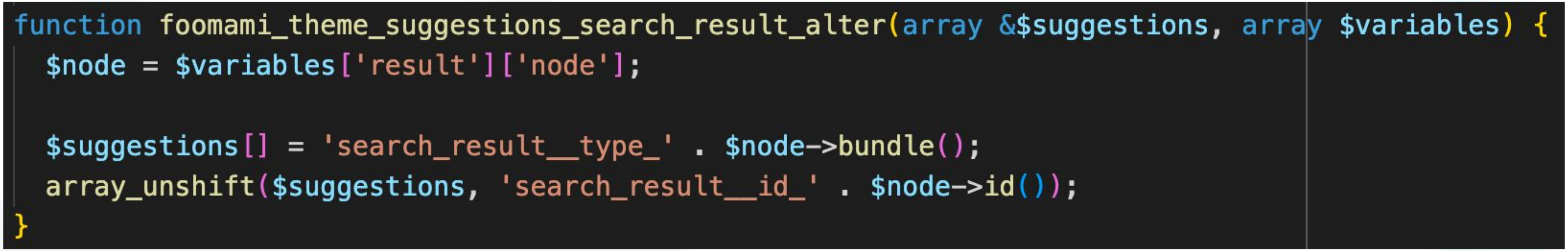
- We’ll use:
  - Drupal-controlled information
    - Node bundle
    - Node ID
    - Node Age
  - Author-controlled information
    - Field values
    - Published status
    - Author info
- This is obviously overkill. Only make the suggestions you’ll actually need.



```
⊟ node => Drupal\node\Entity\Node (29)
  Properties (29)   Available methods (146)   Static class proper

    public in_preview -> null
  ⊞ protected values -> &array (24)
  ⊞ protected fields -> &array (7)
  ⊞ protected fieldDefinitions -> array (26)
  ⊞ protected languages -> array (4)
    protected langcodeKey -> string (8) "langcode"
    protected defaultLangcodeKey -> string (16) "default_langcode
    protected activeLangcode -> string (9) "x-default"
    protected defaultLangcode -> string (2) "en"
  ⊞ protected translations -> &array (2)
    protected translationInitialize -> boolean false
    protected newRevision -> &boolean false
    protected isDefaultRevision -> &string (1) "1"
  ⊞ protected entityKeys -> &array (4)
  ⊞ protected translatableEntityKeys -> &array (8)
    protected validated -> boolean false
    protected validationRequired -> boolean false
    protected loadedRevisionId -> &string (2) "22"
    protected revisionTranslationAffectedKey -> string (29) "revi
    protected enforceRevisionTranslationAffected -> &array (0)
    protected entityTypeId -> string (4) "node"
    protected enforceIsNew -> &null
  ⊞ protected typedData -> Drupal\Core\Entity\Plugin\DataType\E
    protected cacheContexts -> array (0)
    protected cacheTags -> array (0)
    protected cacheMaxAge -> integer -1
    protected _serviceIds -> array (0)
    protected _entityStorages -> array (0)
    protected isSyncing -> &boolean false
```

templates-02

Search Results

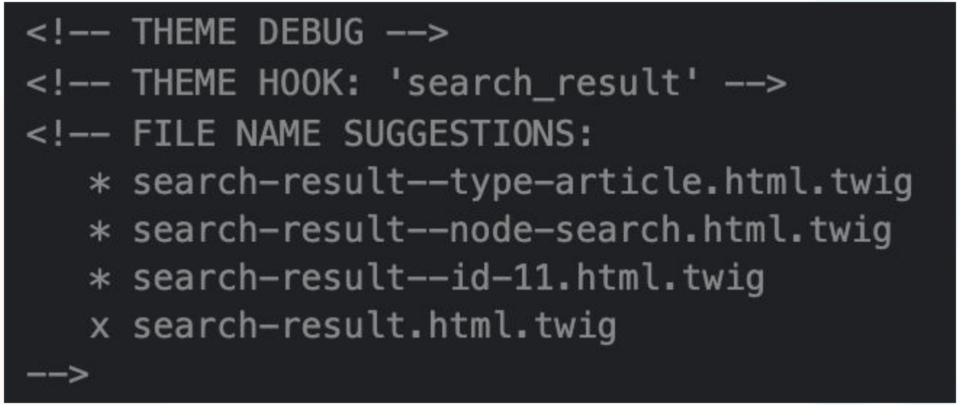


```php
function foomami_theme_suggestions_search_result_alter(array &$suggestions, array $variables) {
  $node = $variables['result']['node'];

  $suggestions[] = 'search_result__type_' . $node->bundle();
  array_unshift($suggestions, 'search_result__id_' . $node->id());
}
```

- Can append or prepend
  - Later in the array = higher priority

- `type-article` overrides `all others`

- search-result.html.twig will always be the fallback.
  - Will fallback to base theme if present, else `stable`



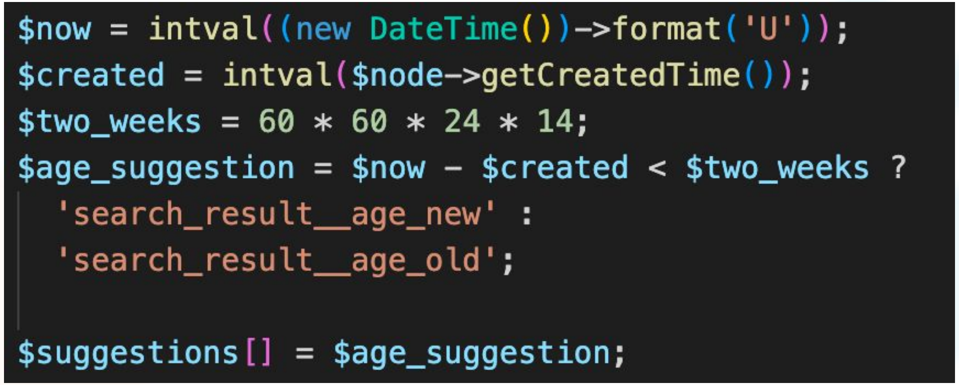
```
<!-- THEME DEBUG -->
<!-- THEME HOOK: 'search_result' -->
<!-- FILE NAME SUGGESTIONS:
   * search-result--type-article.html.twig
   * search-result--node-search.html.twig
   * search-result--id-11.html.twig
   x search-result.html.twig
-->
```

templates-03

Search Results

- We don’t have to use the node’s values directly.

- We can build templates around logic based on node’s values

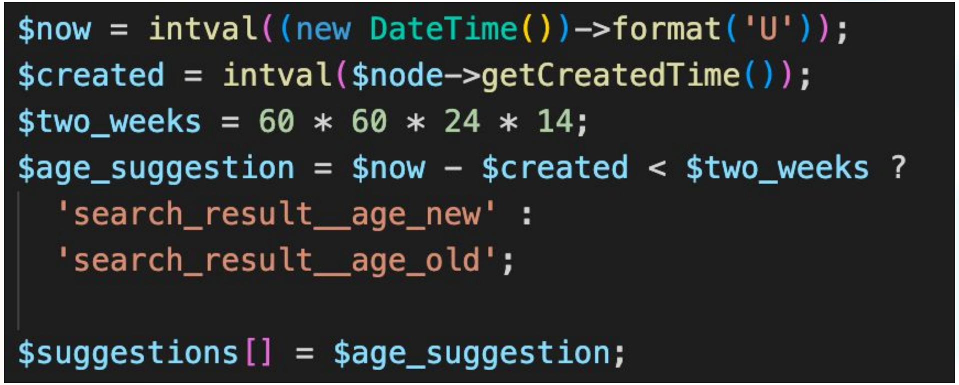


```
$now = intval((new DateTime())->format('U'));
$created = intval($node->getCreatedTime());
$two_weeks = 60 * 60 * 24 * 14;
$age_suggestion = $now - $created < $two_weeks ?
  'search_result__age_new' :
  'search_result__age_old';

$suggestions[] = $age_suggestion;
```

templates-03

Search Results

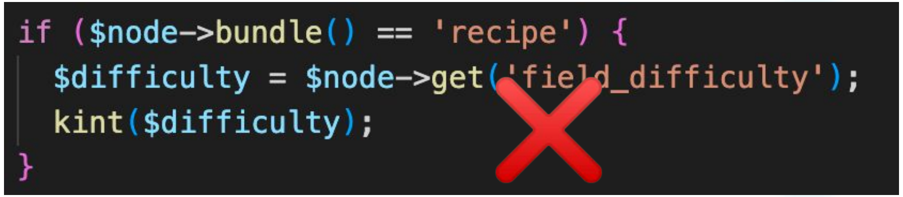
- We don’t have to use the node’s values directly.
- We can build templates around logic based on node’s values



```
$now = intval((new DateTime())->format('U'));
$created = intval($node->getCreatedTime());
$two_weeks = 60 * 60 * 24 * 14;
$age_suggestion = $now - $created < $two_weeks ?
  'search_result__age_new' :
  'search_result__age_old';

$suggestions[] = $age_suggestion;
```

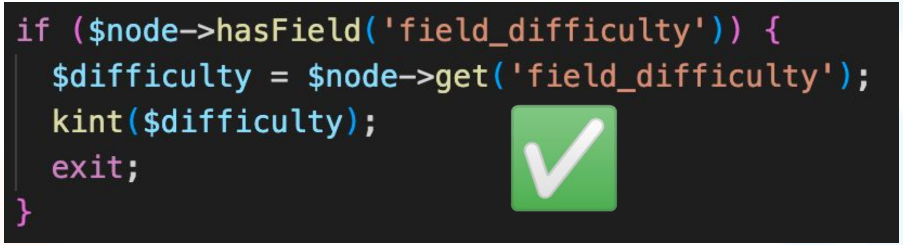
Search Results

- Field values can be a little trickier to get
- Remember that we’re working with two different node types.
  - Fields on one type *might not exist on the other.*
- Remember that fields can be removed and data structures can be renamed.
  - Be sure your checks are what you *really* want.
- Field values often require more than one method/property combo to acquire.



```php
if ($node->bundle() == 'recipe') {
  $difficulty = $node->get('field_difficulty');
  kint($difficulty);
}
```

❌



```php
if ($node->hasField('field_difficulty')) {
  $difficulty = $node->get('field_difficulty');
  kint($difficulty);
  exit;
}
```

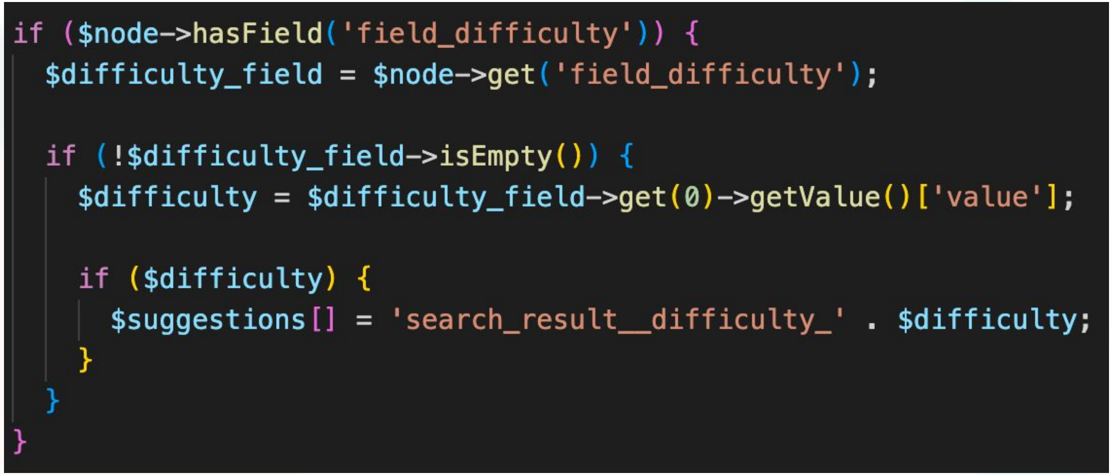
✅

templates-04

Search Results

- Fields **always** store a list.
    - Even if you only allow one item
    - Even if there’s nothing in it
    - It’s always always a list.

- Before you attempt to use a field value:
    - Check if the entity has the field
    - Check if the list is empty
    - Try to access your list item
    - Make sure you have a value that matches your expectations
        - Truthy/Falsey
        - In/Equivalence



```php
if ($node->hasField('field_difficulty')) {
  $difficulty_field = $node->get('field_difficulty');

  if (!$difficulty_field->isEmpty()) {
    $difficulty = $difficulty_field->get(0)->getValue()['value'];

    if ($difficulty) {
      $suggestions[] = 'search_result__difficulty_' . $difficulty;
    }
  }
}
```

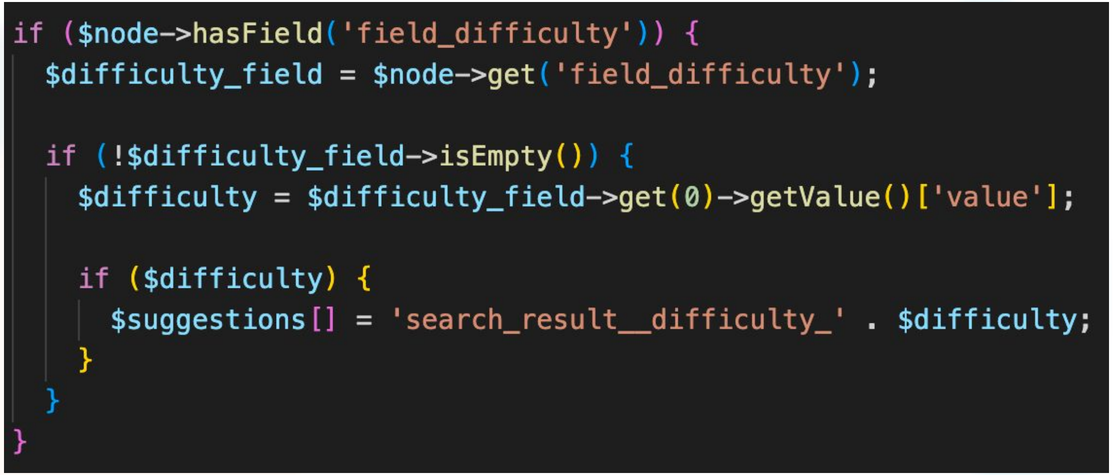
Search Results

- There are lots of ways to get what you want

- $field→getValue()
  - Gives you all items in field list

- $field→get(n)
  - Gives you nth value in list
  - 0-indexed

- $field→first()
  - Gives you first item
  - Same as $field→get(0)



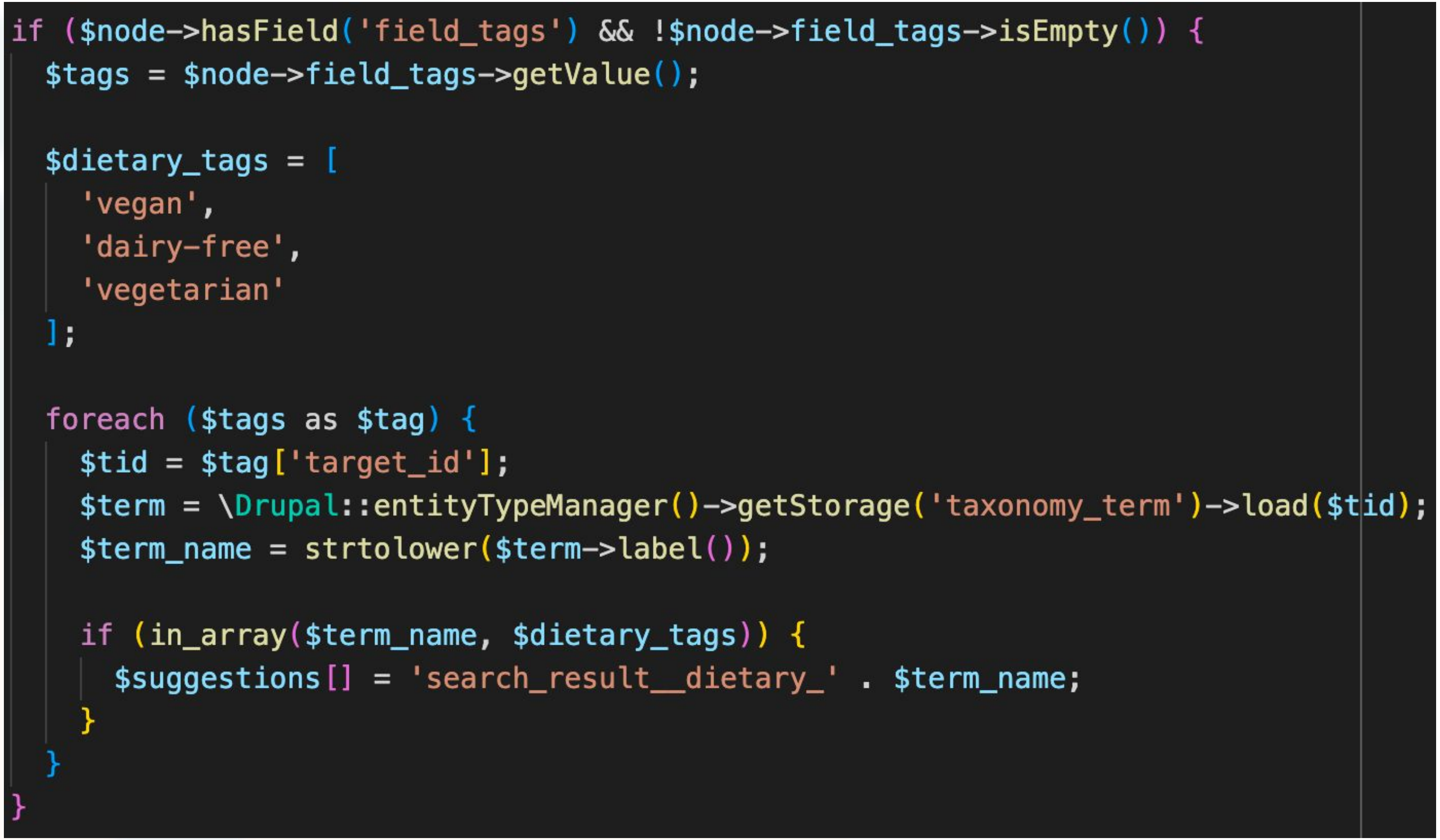
```php
if ($node->hasField('field_difficulty')) {
  $difficulty_field = $node->get('field_difficulty');

  if (!$difficulty_field->isEmpty()) {
    $difficulty = $difficulty_field->get(0)->getValue()['value'];

    if ($difficulty) {
      $suggestions[] = 'search_result__difficulty_' . $difficulty;
    }
  }
}
```

Search Results

- Some fields store ‘useful’ data
  - Lists, strings, numbers, etc
- Some fields store references
  - Taxonomy terms, media items, files
- You can access the referenced entity and use it’s info too, just be careful to build in failure checks

templates-05

Search Results



```php
if ($node->hasField('field_tags') && !$node->field_tags->isEmpty()) {
  $tags = $node->field_tags->getValue();

  $dietary_tags = [
    'vegan',
    'dairy-free',
    'vegetarian'
  ];

  foreach ($tags as $tag) {
    $tid = $tag['target_id'];
    $term = \Drupal::entityTypeManager()->getStorage('taxonomy_term')->load($tid);
    $term_name = strtolower($term->label());

    if (in_array($term_name, $dietary_tags)) {
      $suggestions[] = 'search_result__dietary_' . $term_name;
    }
  }
}
```

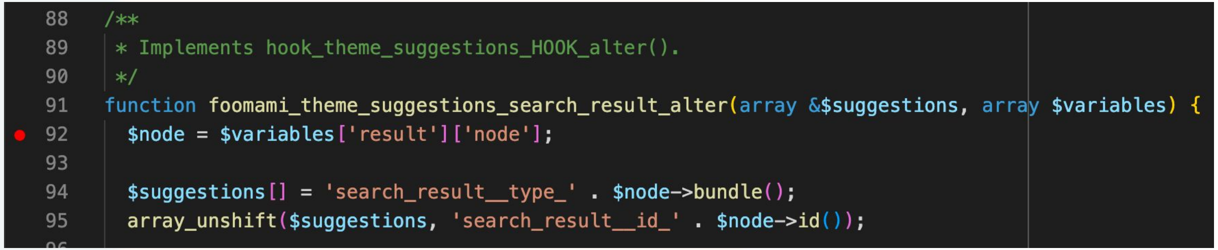
Views Templates

- Views templates already have really good patterns in place, but don’t tell you what they are in the twig debug statements
  - There are issues open to make this happen

- View template base names are always first
  - views-view.html.twig
  - views-view--view-name.html.twig
  - views-view--display-mode.html.twig
  - views-view--view-name--display-mode.html.twig

Let’s Try Step-Debugging

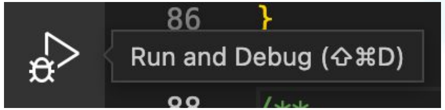
- This is our first good opportunity to attempt using xDebug.
- xDebug lets us peek into our code mid-execution
- I find it to be the most useful when trying to track logic flows.
    - If I want to explore variables/arrays/objects I typically stick to Kint
- To start, enable xDebug in DDEV
    - `ddev xdebug enable`
- Place a breakpoint where we define $node

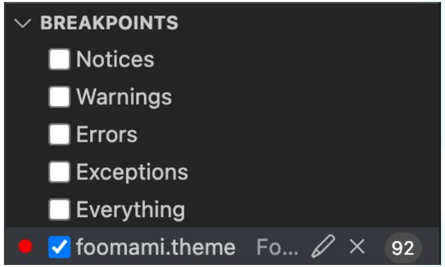


```php
/**
 * Implements hook_theme_suggestions_HOOK_alter().
 */
function foomami_theme_suggestions_search_result_alter(array &$suggestions, array $variables) {
  $node = $variables['result']['node'];

  $suggestions[] = 'search_result__type_' . $node->bundle();
  array_unshift($suggestions, 'search_result__id_' . $node->id());
```

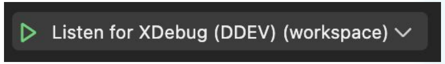
Let’s Try Step-Debugging

- Open the “Run and Debug” sidebar in VS Code
- xDebug will pause execution on all “breakpoints”
  - Where we set them
  - PHP notices/warnings/errors/exceptions (if desired)
- Start the VS code debug listener
- Refresh the page

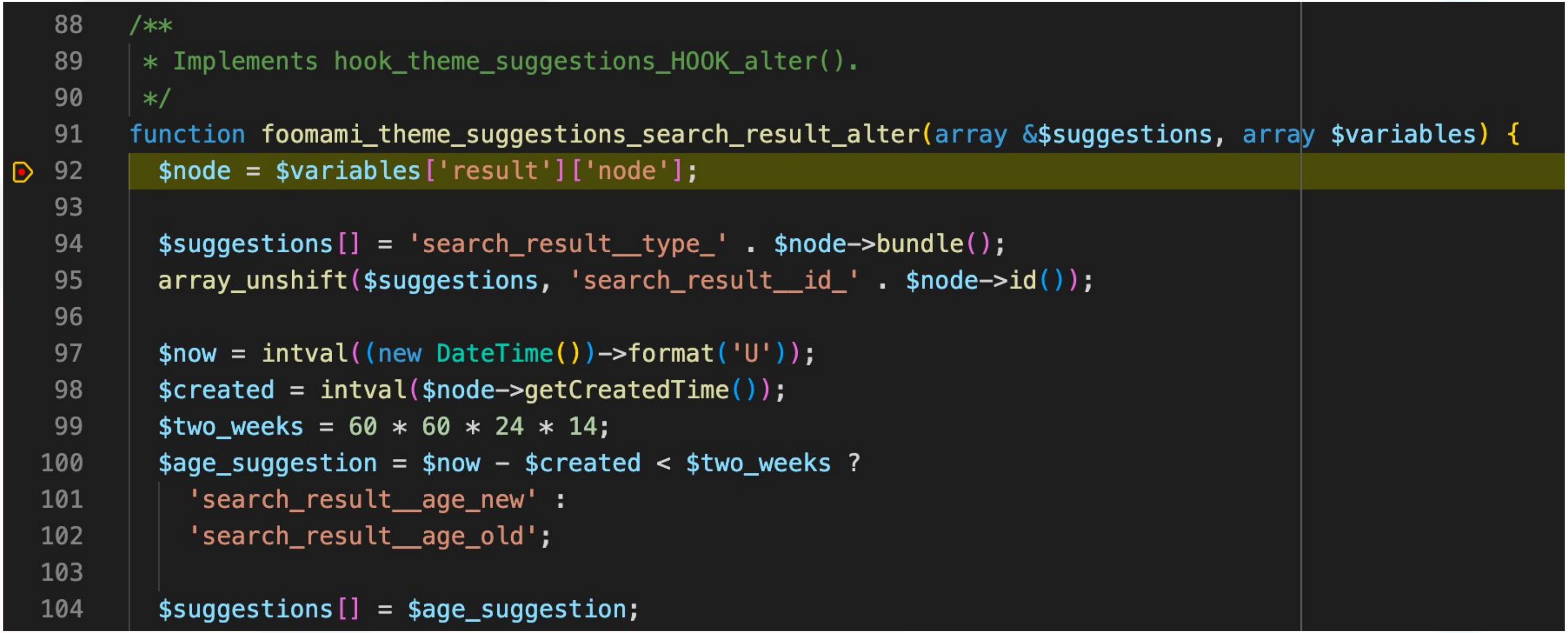



BREAKPOINTS

- □ Notices
- □ Warnings
- □ Errors
- □ Exceptions
- □ Everything
- ● ☑ foomami.theme Fo... 92

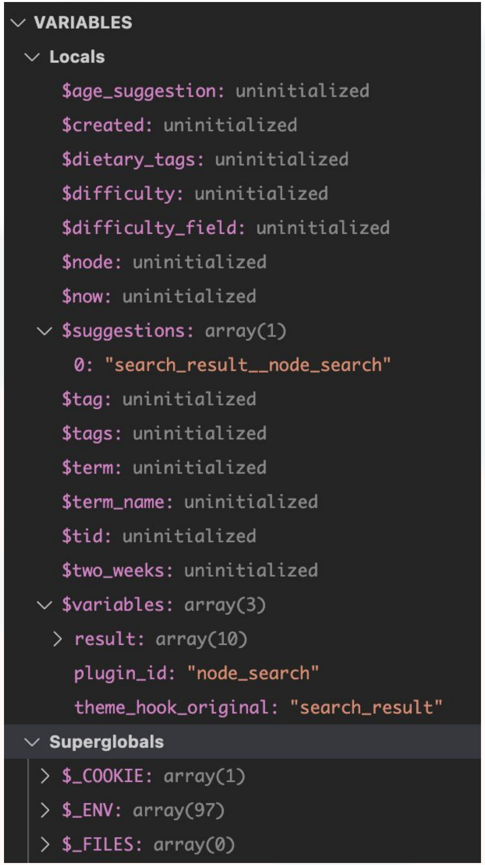


Let’s Try Step-Debugging

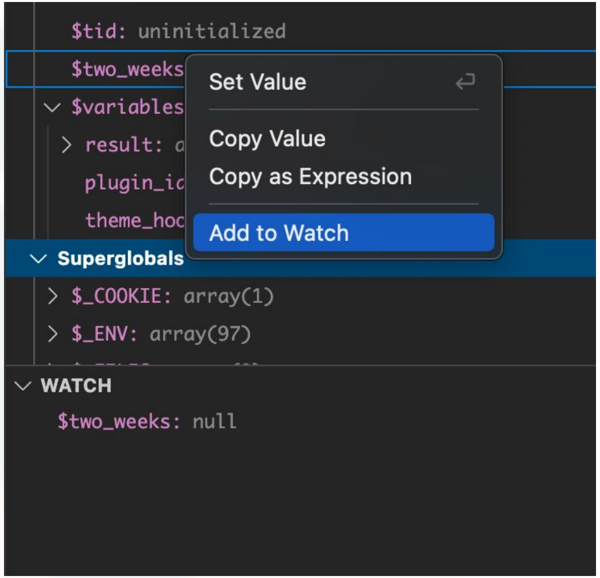


```
/**
 * Implements hook_theme_suggestions_HOOK_alter().
 */
function foomami_theme_suggestions_search_result_alter(array &$suggestions, array $variables) {
  $node = $variables['result']['node'];

  $suggestions[] = 'search_result__type_' . $node->bundle();
  array_unshift($suggestions, 'search_result__id_' . $node->id());

  $now = intval((new DateTime())->format('U'));
  $created = intval($node->getCreatedTime());
  $two_weeks = 60 * 60 * 24 * 14;
  $age_suggestion = $now - $created < $two_weeks ?
    'search_result__age_new' :
    'search_result__age_old';

  $suggestions[] = $age_suggestion;
```

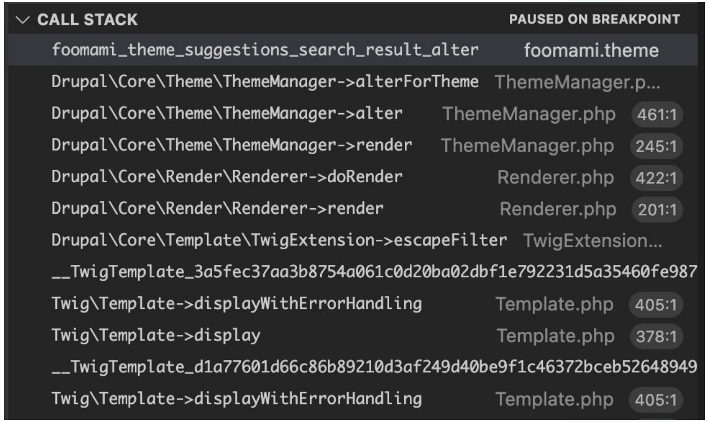
Let’s Try Step-Debugging



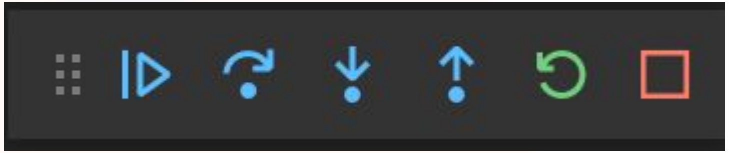
```
VARIABLES
  Locals
    $age_suggestion: uninitialized
    $created: uninitialized
    $dietary_tags: uninitialized
    $difficulty: uninitialized
    $difficulty_field: uninitialized
    $node: uninitialized
    $now: uninitialized
    $suggestions: array(1)
      0: "search_result__node_search"
    $tag: uninitialized
    $tags: uninitialized
    $term: uninitialized
    $term_name: uninitialized
    $tid: uninitialized
    $two_weeks: uninitialized
    $variables: array(3)
      result: array(10)
      plugin_id: "node_search"
      theme_hook_original: "search_result"
  Superglobals
    $_COOKIE: array(1)
    $_ENV: array(97)
    $_FILES: array(0)
```



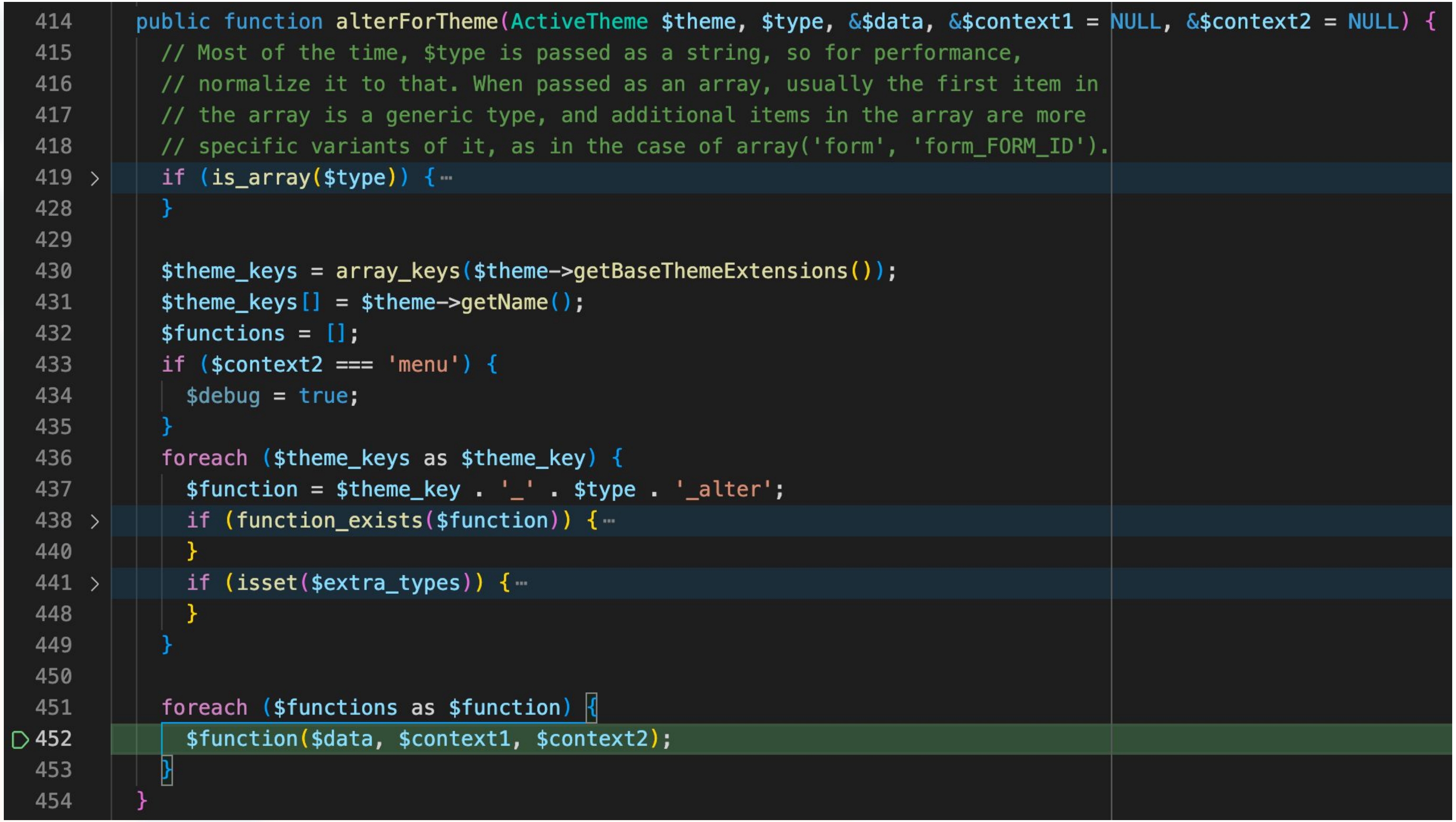
```
    $tid: uninitialized
    $two_weeks
    $variables
      result: a
      plugin_ic
      theme_hoc
  Superglobals
    $_COOKIE: array(1)
    $_ENV: array(97)
WATCH
  $two_weeks: null
```



```
CALL STACK                                                    PAUSED ON BREAKPOINT
  foomami_theme_suggestions_search_result_alter        foomami.theme
  Drupal\Core\Theme\ThemeManager->alterForTheme  ThemeManager.p...
  Drupal\Core\Theme\ThemeManager->alter         ThemeManager.php  461:1
  Drupal\Core\Theme\ThemeManager->render        ThemeManager.php  245:1
  Drupal\Core\Render\Renderer->doRender             Renderer.php  422:1
  Drupal\Core\Render\Renderer->render               Renderer.php  201:1
  Drupal\Core\Template\TwigExtension->escapeFilter  TwigExtension...
  __TwigTemplate_3a5fec37aa3b8754a061c0d20ba02dbf1e792231d5a35460fe987
  Twig\Template->displayWithErrorHandling           Template.php  405:1
  Twig\Template->display                            Template.php  378:1
  __TwigTemplate_d1a77601d66c86b89210d3af249d40be9f1c46372bceb52648949
  Twig\Template->displayWithErrorHandling           Template.php  405:1
```



Let’s Try Step-Debugging



```php
public function alterForTheme(ActiveTheme $theme, $type, &$data, &$context1 = NULL, &$context2 = NULL) {
  // Most of the time, $type is passed as a string, so for performance,
  // normalize it to that. When passed as an array, usually the first item in
  // the array is a generic type, and additional items in the array are more
  // specific variants of it, as in the case of array('form', 'form_FORM_ID').
  if (is_array($type)) {⋯
  }

  $theme_keys = array_keys($theme->getBaseThemeExtensions());
  $theme_keys[] = $theme->getName();
  $functions = [];
  if ($context2 === 'menu') {
    $debug = true;
  }
  foreach ($theme_keys as $theme_key) {
    $function = $theme_key . '_' . $type . '_alter';
    if (function_exists($function)) {⋯
    }
    if (isset($extra_types)) {⋯
    }
  }

  foreach ($functions as $function) {
    $function($data, $context1, $context2);
  }
}
```

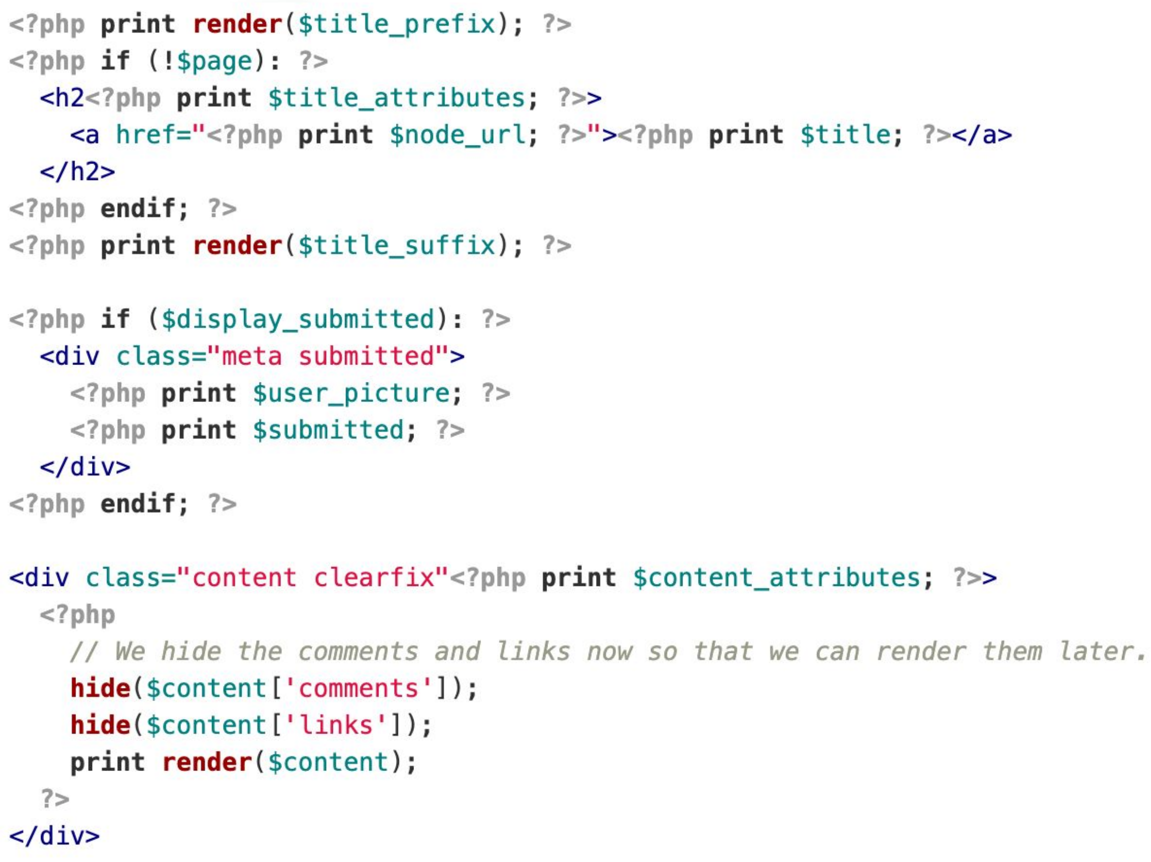
Hard Stuff in Templates

- Twig templates are pretty easy to get started in
  - Mainly just HTML
  - {{ variable }} to print variable
  - Some additional programming with {% set %}, {% if %}, {% for %}
  - The documentation is pretty good

- They can get complex fast when you want to get specific data out of arrays or objects

- So what is twig?
  - It’s “The flexible, fast, and secure template engine for PHP”
  - It renders to PHP similar to what templates looked like in Drupal 7 and prior

Hard Stuff in Templates



```
<?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; ?>

<div class="content clearfix"<?php print $content_attributes; ?>>
  <?php
    // We hide the comments and links now so that we can render them later.
    hide($content['comments']);
    hide($content['links']);
    print render($content);
  ?>
</div>
```

Primary Benefits

- Easier to read and write
  - No more <?php print(‘inline php’); ?>
- More secure
  - We can limit what twig can do
  - No direct database connections in templates
  - Twig print ({{ }}) sanitizes all strings by default
- Highly extensible
  - We can create our own twig filters and functions
  - **We can bundle those filters and functions into modules and share it with a community of open source developers to make everyone’s lives easier**

templates-06

Handy Modules for Twig

- twig_tweak
  - Provides functions that do drupal-specific things
    - {{ drupal_entity('node', 123, 'teaser') }}
    - Cheat sheet with lots of examples

- twig_field_value
  - Useful for getting partial values from fields instead of cutting up `content.field_name` on your own
  - Also capable of getting raw values from the field, bypassing the field processing that would normally happen

- Kint (not really a module)
  - You can use kint to dump out twig's context or individual variables from your template
  - `ddev composer require kint-php/kint-twig`
  - Generally best to avoid putting more than one kint dump on a page

Accessing Data in Twig

- Let’s mimic some of the data finding we did in our theme_suggestion_HOOK_alter function
    - We won’t be editing these templates beyond just trying to get to the data via Kint()

- Get basic node info (bundle, id)
- Set a variable based on some info
- Get a string out of a field
- Get an entity referenced from a field

- node.html.twig

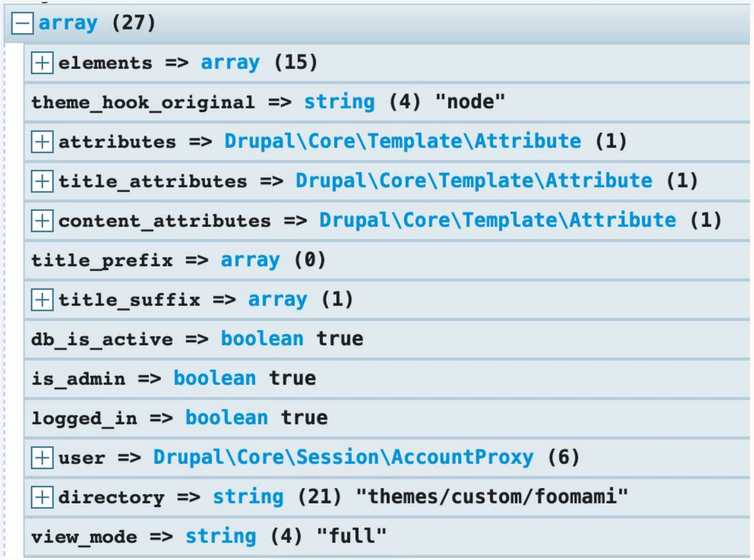
- Go to an individual recipe page

templates-07

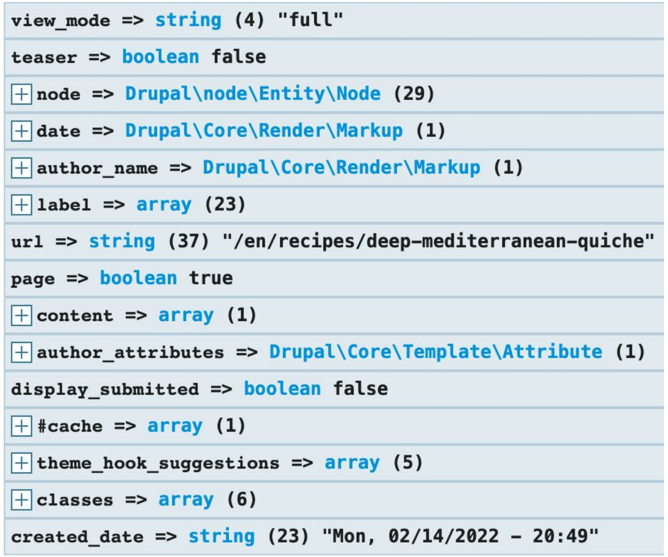
Get basic node info

- kint(_context|keys)
  - Prints out array of names of all variables we have access to

- kint() or kint(_context)
  - Prints out array of all variables (with contents)



```
⊟array (27)
  ⊞elements => array (15)
  theme_hook_original => string (4) "node"
  ⊞attributes => Drupal\Core\Template\Attribute (1)
  ⊞title_attributes => Drupal\Core\Template\Attribute (1)
  ⊞content_attributes => Drupal\Core\Template\Attribute (1)
  title_prefix => array (0)
  ⊞title_suffix => array (1)
  db_is_active => boolean true
  is_admin => boolean true
  logged_in => boolean true
  ⊞user => Drupal\Core\Session\AccountProxy (6)
  ⊞directory => string (21) "themes/custom/foomami"
  view_mode => string (4) "full"
```



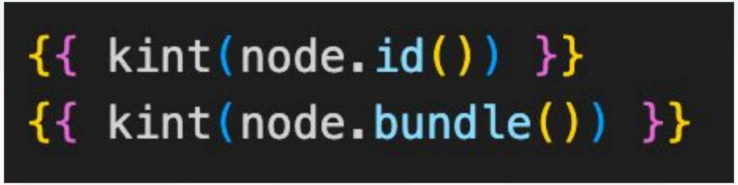
```
view_mode => string (4) "full"
teaser => boolean false
⊞node => Drupal\node\Entity\Node (29)
⊞date => Drupal\Core\Render\Markup (1)
⊞author_name => Drupal\Core\Render\Markup (1)
⊞label => array (23)
url => string (37) "/en/recipes/deep-mediterranean-quiche"
page => boolean true
⊞content => array (1)
⊞author_attributes => Drupal\Core\Template\Attribute (1)
display_submitted => boolean false
⊞#cache => array (1)
⊞theme_hook_suggestions => array (5)
⊞classes => array (6)
created_date => string (23) "Mon, 02/14/2022 - 20:49"
```

templates-08

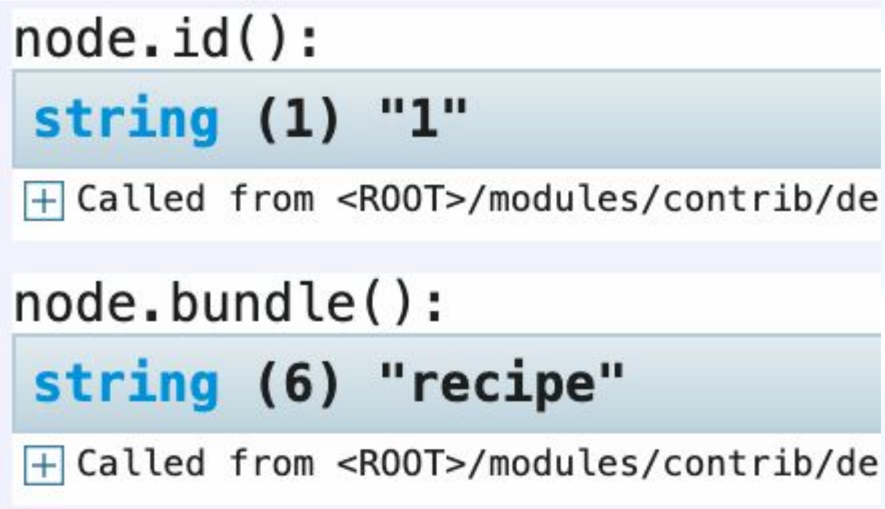
Get basic node info

- We can use some methods on objects using dot-notation
  - isPublished(), hasField(), getCreatedTime(), other common methods

- Object.methodName()



```
{{ kint(node.id()) }}
{{ kint(node.bundle()) }}
```

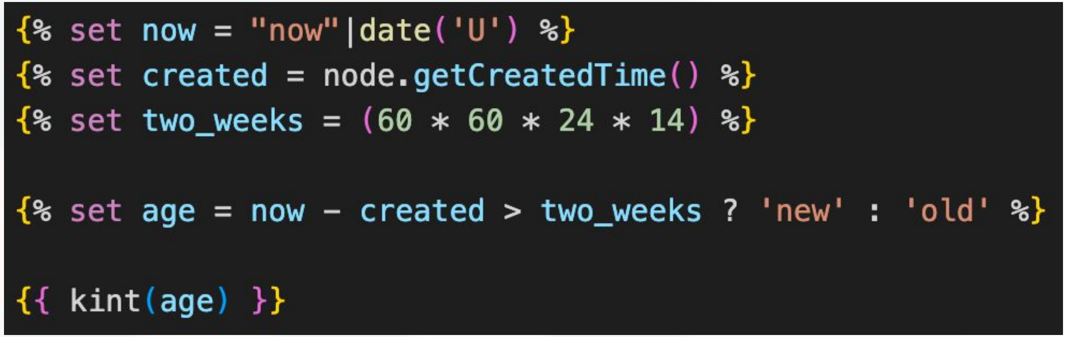


```
node.id():
string (1) "1"
⊞ Called from <ROOT>/modules/contrib/de

node.bundle():
string (6) "recipe"
⊞ Called from <ROOT>/modules/contrib/de
```

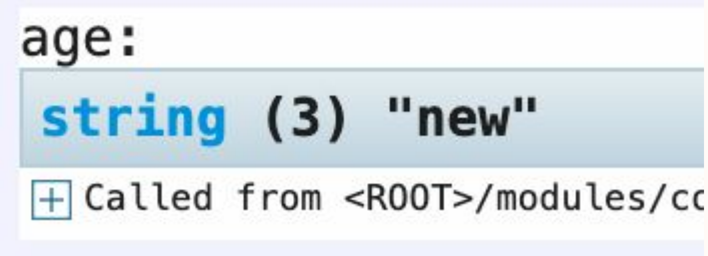
templates-09

Set a variable based on info

- For some reason kint won’t print out number values
- Note that we switch types from string to numbers and do math



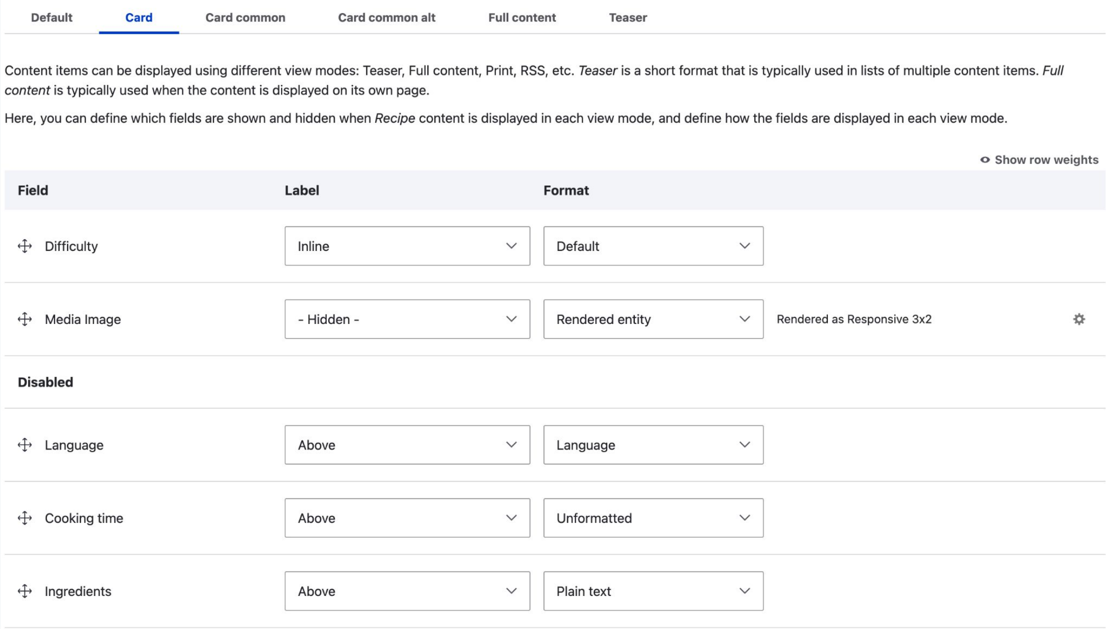
```
{% set now = "now"|date('U') %}
{% set created = node.getCreatedTime() %}
{% set two_weeks = (60 * 60 * 24 * 14) %}

{% set age = now - created > two_weeks ? 'new' : 'old' %}

{{ kint(age) }}
```



```
age:
string (3) "new"
⊞ Called from <ROOT>/modules/cc
```

Getting a string out of a field

- content vs. node
  - “content” is the field values *after* going through field processors
  - “node” has the field values as stored in the database



Content items can be displayed using different view modes: Teaser, Full content, Print, RSS, etc. *Teaser* is a short format that is typically used in lists of multiple content items. *Full content* is typically used when the content is displayed on its own page.

Here, you can define which fields are shown and hidden when *Recipe* content is displayed in each view mode, and define how the fields are displayed in each view mode.

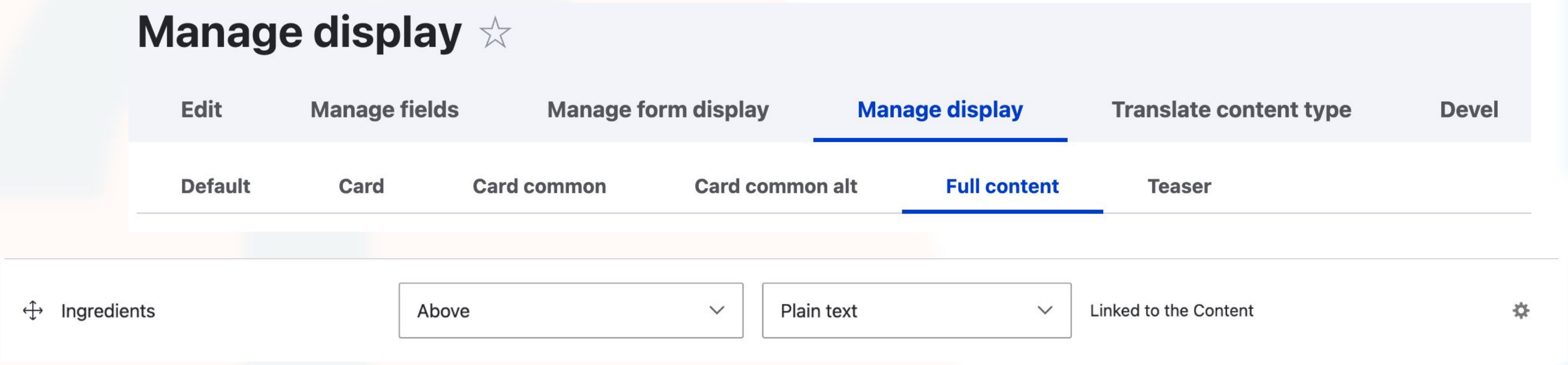
Field	Label	Format	
Difficulty	Inline	Default	
Media Image	- Hidden -	Rendered entity	Rendered as Responsive 3x2
Disabled			
Language	Above	Language	
Cooking time	Above	Unformatted	
Ingredients	Above	Plain text	

Getting a string out of a field

- Umami has nodes configured to use layout builder
  - I’ve disabled that to show the differences

- Let’s look at a field that has multiple values like ‘ingredients’

- I’ve also changed the field processor to “link to content”



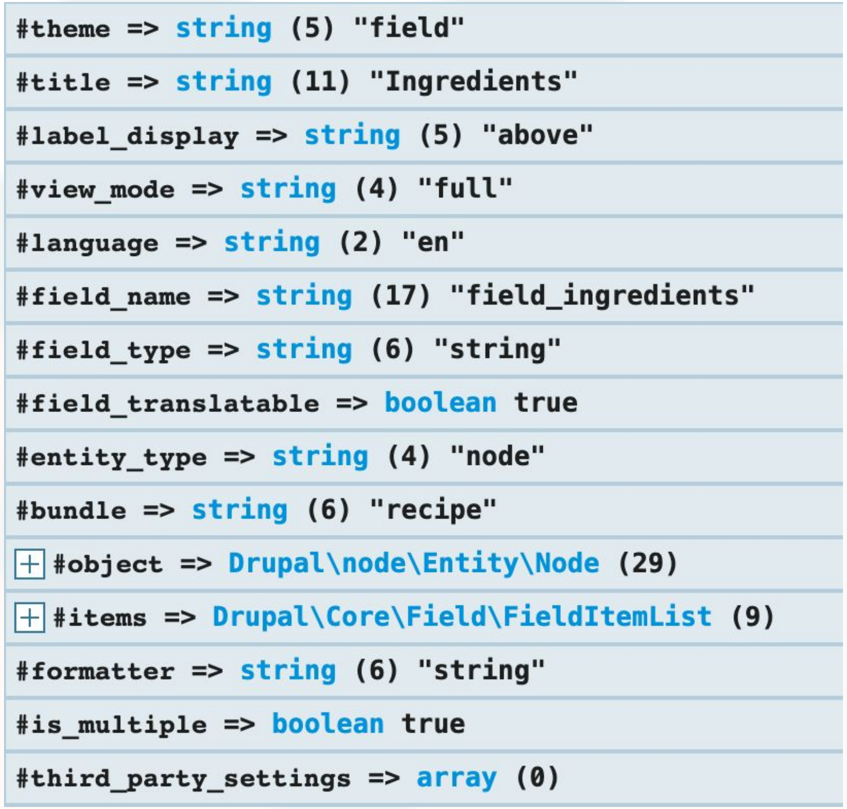
Manage display ☆

Edit | Manage fields | Manage form display | **Manage display** | Translate content type | Devel

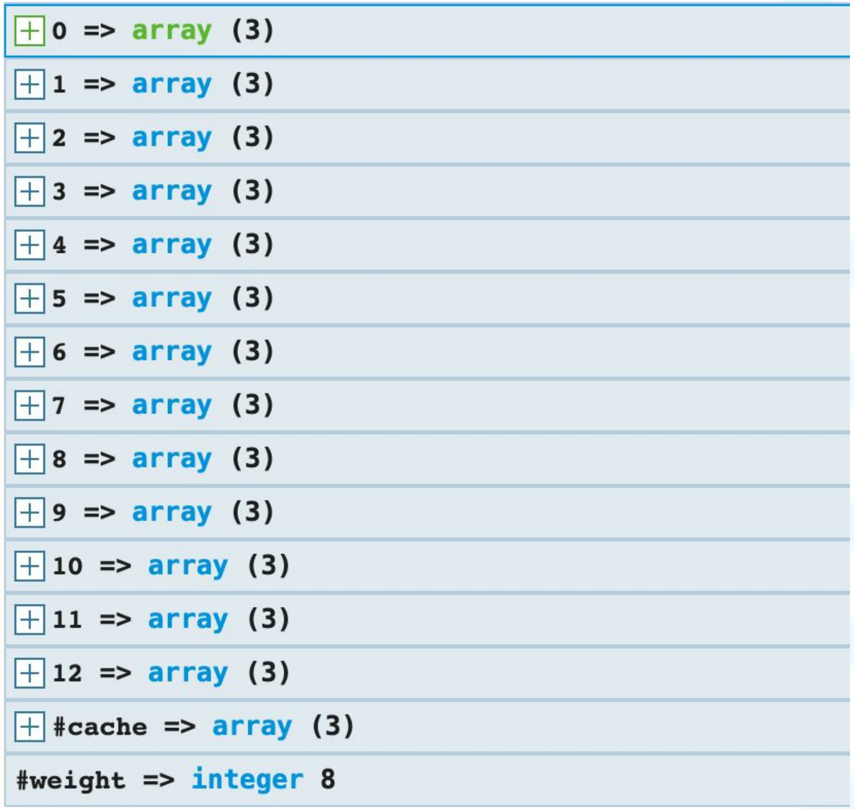
Default | Card | Card common | Card common alt | **Full content** | Teaser

Ingredients	Above	Plain text	Linked to the Content

Getting a string out of a field (contents)

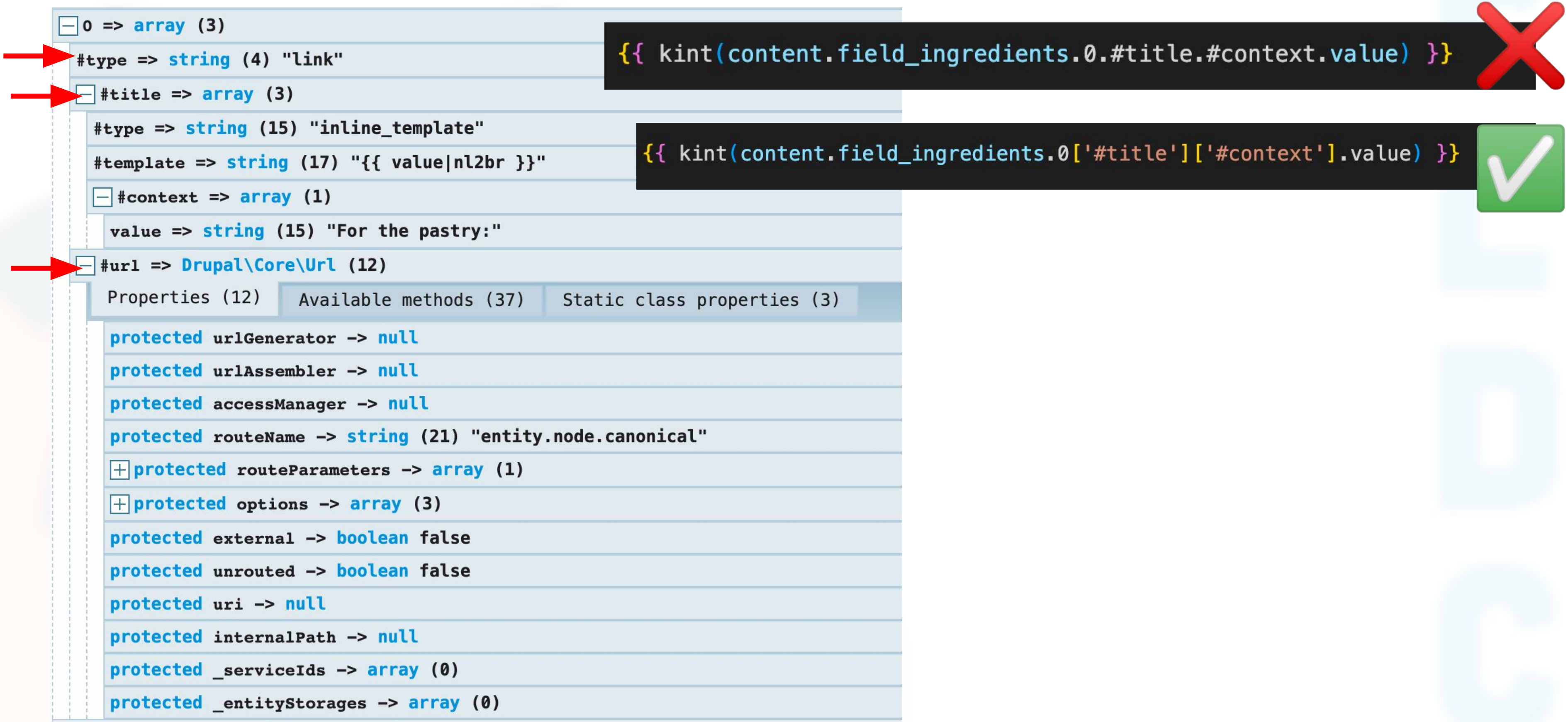


```
#theme => string (5) "field"
#title => string (11) "Ingredients"
#label_display => string (5) "above"
#view_mode => string (4) "full"
#language => string (2) "en"
#field_name => string (17) "field_ingredients"
#field_type => string (6) "string"
#field_translatable => boolean true
#entity_type => string (4) "node"
#bundle => string (6) "recipe"
⊞ #object => Drupal\node\Entity\Node (29)
⊞ #items => Drupal\Core\Field\FieldItemList (9)
#formatter => string (6) "string"
#is_multiple => boolean true
#third_party_settings => array (0)
```



```
⊞ 0 => array (3)
⊞ 1 => array (3)
⊞ 2 => array (3)
⊞ 3 => array (3)
⊞ 4 => array (3)
⊞ 5 => array (3)
⊞ 6 => array (3)
⊞ 7 => array (3)
⊞ 8 => array (3)
⊞ 9 => array (3)
⊞ 10 => array (3)
⊞ 11 => array (3)
⊞ 12 => array (3)
⊞ #cache => array (3)
#weight => integer 8
```

Getting a string out of a field (contents)



```
0 => array (3)
 #type => string (4) "link"
 #title => array (3)
  #type => string (15) "inline_template"
  #template => string (17) "{{ value|nl2br }}"
  #context => array (1)
   value => string (15) "For the pastry:"
 #url => Drupal\Core\Url (12)
  Properties (12)   Available methods (37)   Static class properties (3)
   protected urlGenerator -> null
   protected urlAssembler -> null
   protected accessManager -> null
   protected routeName -> string (21) "entity.node.canonical"
   protected routeParameters -> array (1)
   protected options -> array (3)
   protected external -> boolean false
   protected unrouted -> boolean false
   protected uri -> null
   protected internalPath -> null
   protected _serviceIds -> array (0)
   protected _entityStorages -> array (0)
```

```
{{ kint(content.field_ingredients.0.#title.#context.value) }}
```

❌

```
{{ kint(content.field_ingredients.0['#title']['#context'].value) }}
```

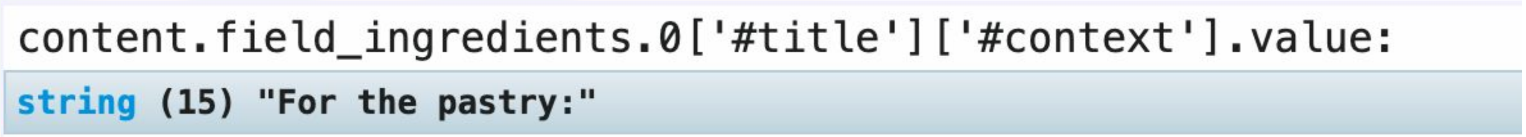
✅

templates-10

Getting a string out of a field (contents)

- Accessing array values normally uses dot-notation
  - contents.field_ingredients.0

- Accessing array values can also use bracket-notation
  - contents[‘field_ingredients’][‘0’]
    - Zero can be a string or int here
  - Bracket notation required for keys that start with a #



```
content.field_ingredients.0['#title']['#context'].value:
string (15) "For the pastry:"
```

templates-10

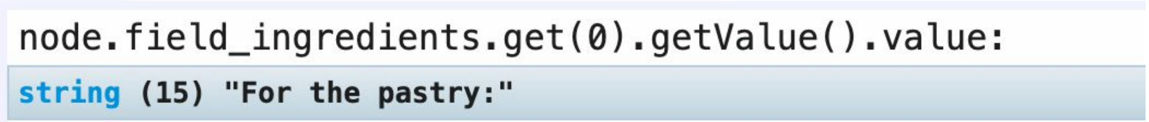
Getting a string out of a field (entity)

- Node is an object and the properties are all protected
  - Need to access values through methods or iterator

- Iterator allows access to common values we might want to expose that would otherwise require complex data traversal
  - node.getIterator()|keys shows all values we can access!
  - See comparison on next slide

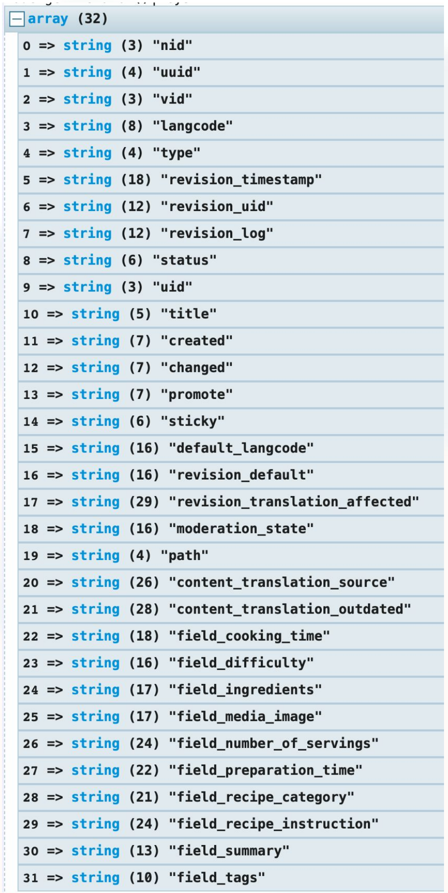
- node.getField(‘field_ingredients’) or node.field_ingredients both work!

- Some of the methods we used in php won’t work in twig
  - .first(), for example

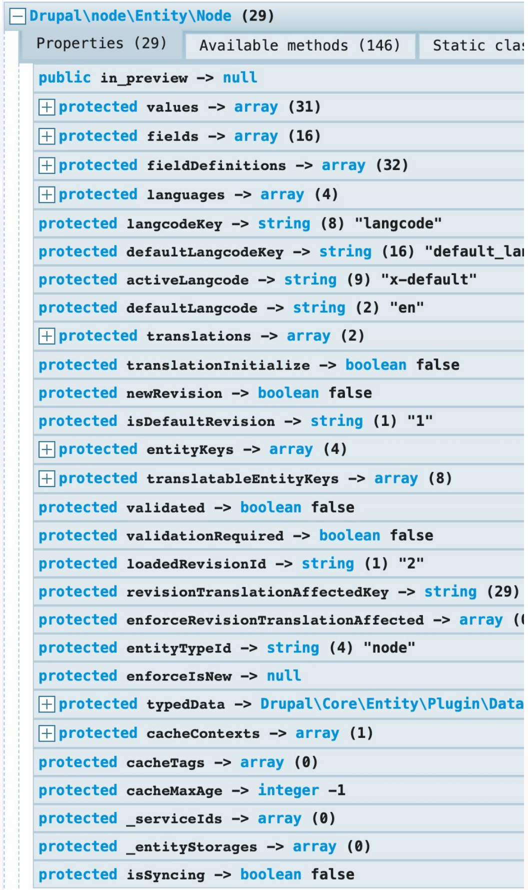


```
node.field_ingredients.get(0).getValue().value:
string (15) "For the pastry:"
```

Getting a string out of a field (entity)



```
⊟ array (32)
0 => string (3) "nid"
1 => string (4) "uuid"
2 => string (3) "vid"
3 => string (8) "langcode"
4 => string (4) "type"
5 => string (18) "revision_timestamp"
6 => string (12) "revision_uid"
7 => string (12) "revision_log"
8 => string (6) "status"
9 => string (3) "uid"
10 => string (5) "title"
11 => string (7) "created"
12 => string (7) "changed"
13 => string (7) "promote"
14 => string (6) "sticky"
15 => string (16) "default_langcode"
16 => string (16) "revision_default"
17 => string (29) "revision_translation_affected"
18 => string (16) "moderation_state"
19 => string (4) "path"
20 => string (26) "content_translation_source"
21 => string (28) "content_translation_outdated"
22 => string (18) "field_cooking_time"
23 => string (16) "field_difficulty"
24 => string (17) "field_ingredients"
25 => string (17) "field_media_image"
26 => string (24) "field_number_of_servings"
27 => string (22) "field_preparation_time"
28 => string (21) "field_recipe_category"
29 => string (24) "field_recipe_instruction"
30 => string (13) "field_summary"
31 => string (10) "field_tags"
```



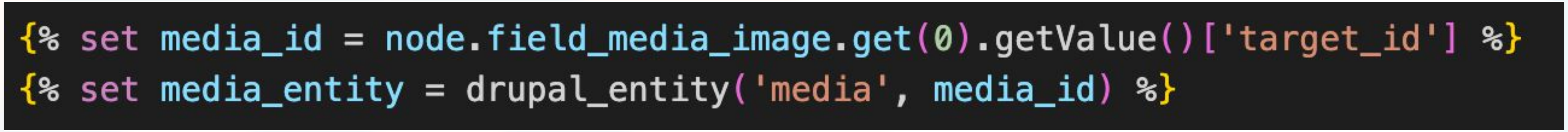
```
⊟ Drupal\node\Entity\Node (29)
  Properties (29)   Available methods (146)   Static clas
  public in_preview -> null
  ⊞ protected values -> array (31)
  ⊞ protected fields -> array (16)
  ⊞ protected fieldDefinitions -> array (32)
  ⊞ protected languages -> array (4)
  protected langcodeKey -> string (8) "langcode"
  protected defaultLangcodeKey -> string (16) "default_lan
  protected activeLangcode -> string (9) "x-default"
  protected defaultLangcode -> string (2) "en"
  ⊞ protected translations -> array (2)
  protected translationInitialize -> boolean false
  protected newRevision -> boolean false
  protected isDefaultRevision -> string (1) "1"
  ⊞ protected entityKeys -> array (4)
  ⊞ protected translatableEntityKeys -> array (8)
  protected validated -> boolean false
  protected validationRequired -> boolean false
  protected loadedRevisionId -> string (1) "2"
  protected revisionTranslationAffectedKey -> string (29)
  protected enforceRevisionTranslationAffected -> array (
  protected entityTypeId -> string (4) "node"
  protected enforceIsNew -> null
  ⊞ protected typedData -> Drupal\Core\Entity\Plugin\Data
  ⊞ protected cacheContexts -> array (1)
  protected cacheTags -> array (0)
  protected cacheMaxAge -> integer -1
  protected _serviceIds -> array (0)
  protected _entityStorages -> array (0)
  protected isSyncing -> boolean false
```

templates-11

Get a referenced entity

- In PHP, you can directly access the referenced entity
  - node→field_media_image→referencedEntities()
  - Not allowed in twig!

- To get access to the referenced entities we have some options
  - Get the target id and use twig_tweak
    - Requires the module (we haven’t installed it here)

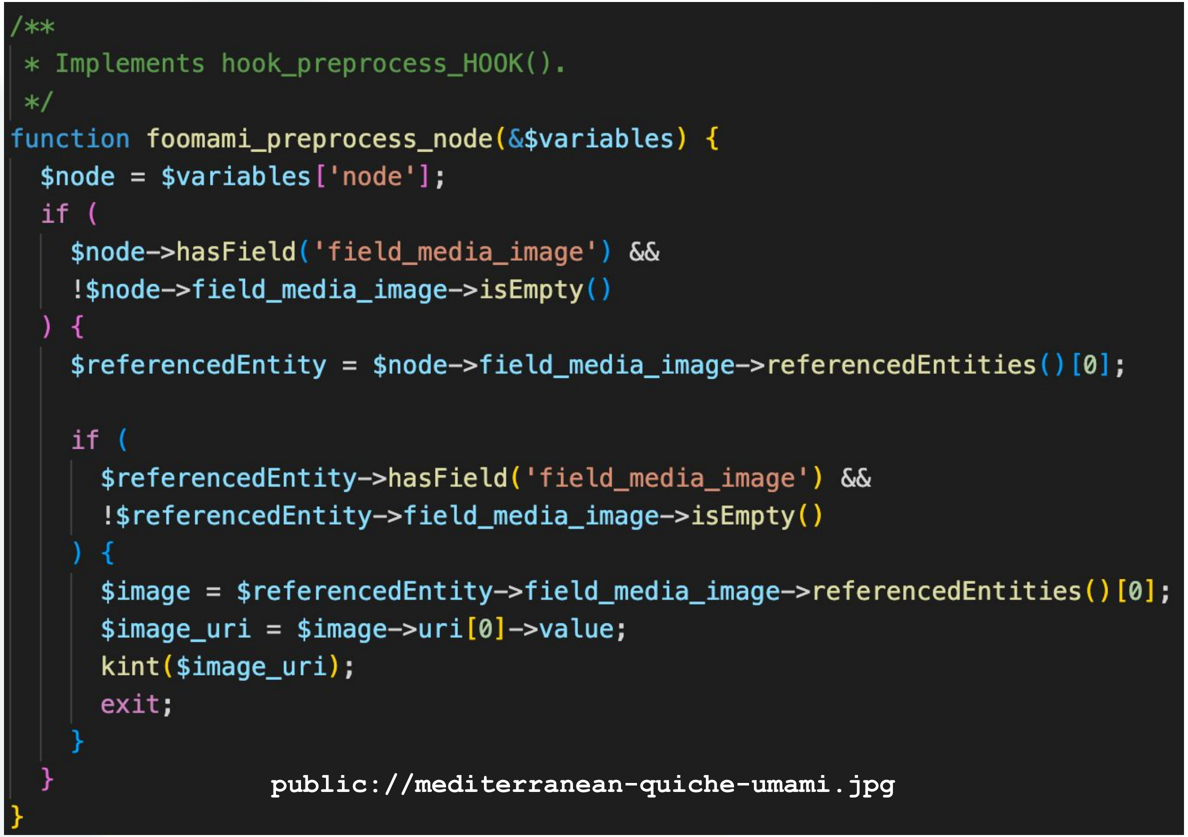


```
{% set media_id = node.field_media_image.get(0).getValue()['target_id'] %}
{% set media_entity = drupal_entity('media', media_id) %}
```

  - Create a new variable and pass it into our twig template
    - Requires going back to our preprocess hooks

templates-12

Get a referenced entity



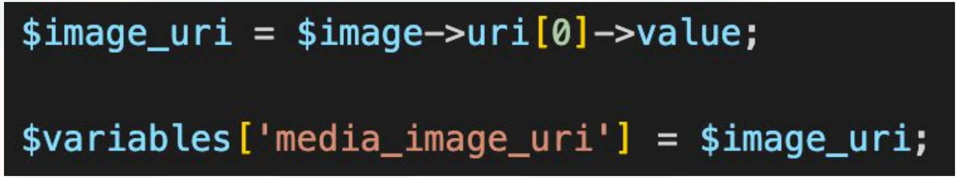
```php
/**
 * Implements hook_preprocess_HOOK().
 */
function foomami_preprocess_node(&$variables) {
  $node = $variables['node'];
  if (
    $node->hasField('field_media_image') &&
    !$node->field_media_image->isEmpty()
  ) {
    $referencedEntity = $node->field_media_image->referencedEntities()[0];

    if (
      $referencedEntity->hasField('field_media_image') &&
      !$referencedEntity->field_media_image->isEmpty()
    ) {
      $image = $referencedEntity->field_media_image->referencedEntities()[0];
      $image_uri = $image->uri[0]->value;
      kint($image_uri);
      exit;
    }
  }
}
```

```
public://mediterranean-quiche-umami.jpg
```

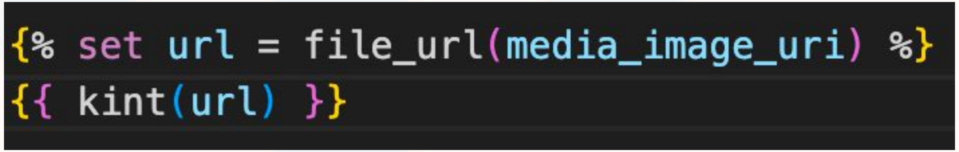
templates-13

Get a referenced entity

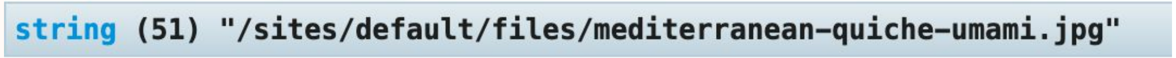
- From here we have a couple options
- Send the URI string to the template and use one of Drupal’s “file_url” twig function
  - Drupal twig filters
  - Drupal twig functions



```php
$image_uri = $image->uri[0]->value;

$variables['media_image_uri'] = $image_uri;
```



```twig
{% set url = file_url(media_image_uri) %}
{{ kint(url) }}
```



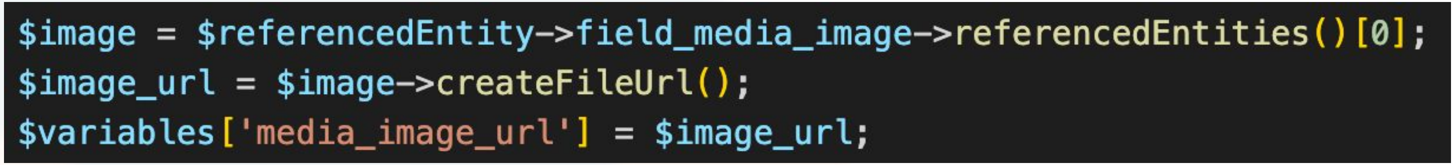
```
string (51) "/sites/default/files/mediterranean-quiche-umami.jpg"
```

templates-14

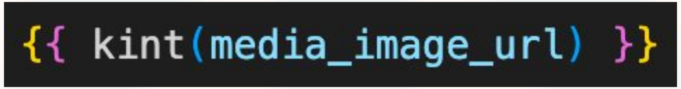
Get a referenced entity

- From here we have a couple options

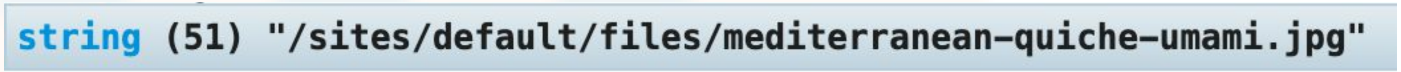
- Convert the URI to a URL in PHP and then send it to twig



```
$image = $referencedEntity->field_media_image->referencedEntities()[0];
$image_url = $image->createFileUrl();
$variables['media_image_url'] = $image_url;
```



```
{{ kint(media_image_url) }}
```



```
string (51) "/sites/default/files/mediterranean-quiche-umami.jpg"
```

Questions/Open Floor