

Exercise 13:

.theme file and preprocess_page

Don't make yourself responsible for updating every site on January 1st. Let the computer do the work for you and automatically set items, like the copyright date, to the current year. We will begin to dive into some more advanced elements of Drupal theming, including preprocess functions and template variables.

What is a preprocess function? A preprocess function is a function that allows us to manipulate (add, edit, or remove) our data before it is processed by our template. Hence, the "pre" in preprocess. It's powered by **hook_preprocess()** function. **hook_preprocess_HOOK()** allows modules to preprocess theme variables, but for a specific theme hook, like node, page, or menu_link. For our example we will manipulate the data before it goes to the **page** twig template (`page.html.twig`), so we will use **hook_preprocess_page()**

Create our .theme file

1. Create a file at the root of the Acme Theme named **acme.theme** and open in your preferred code editor

```
$ cd MYDRUPAL
$ touch themes/acme/acme.theme
```

2. Write the following code in your new **acme.theme** file:

```
<?php
function acme_preprocess_page(&$variables) {
    $variables['copyright'] = "It works!";
}
```

Note that there is not a closing `?>` at the end, but there should be a line of whitespace.

3. Open up **page.html.twig** and near the closing footer element, print out our new variable using the following code:

```
{% if copyright %}
  <div class="copyright">
    {{ copyright }}
  </div>
{% endif %}
```

4. Clear cache and verify your copyright is on the page.

What year is it?

Great, it's printing our variable, but we're still not much better off than simply adding a block to say the copyright date manually. Lets start adding some logic in **acme.theme**.

1. Open **acme.theme** and add some php around our new variable so that it can print the current year instead of our filler message:

```
<?php
function acme_preprocess_page(&$variables) {
  $variables['copyright'] = t("Copyright @date",
    array('@date' => date('Y')))
};
}
```

Note that there is not a closing `?>` at the end, but there should be a line of whitespace.

Questions you may have...

- Where did you find that function name? I never would have guessed that.
- What would happen if I didn't check if a variable is set before printing it?
- Why don't I close the php tags?
- Why does the new variable have a `t()` function around it?
- What's the best way to add markup and styles to our variable?
- Could I have done all this in the twig template?

Done 😊
