Exercise 4:

The .info.yml file

To create a new theme for Drupal 8, the only real requirement is to make sure you have implemented the .info.yml file (YaML, like Camel). Drupal 8 runs off of YaML files much the way that Drupal 7 ran off of .info files.

As long as you have your .info.yml file in place with a few keys in place, you have a theme (although if thats all you have in place, it's not going to be a pretty theme).

Create the theme folder and "YaML" file

1. Locate the */themes* folder in your Drupal installation. In Drupal 8, Contrib and Custom modules and themes, are saved at the root level */modules* and */themes* folders.

2. Create a new folder and call it `acme`

```
$ cd MYDRUPAL
$ mkdir themes/acme
```

1. In that folder create a new file. It should be called *acme.info.yml*

```
$ touch themes/acme/acme.info.yml
```

1. Add the following lines to that file

```
name: Acme
description: My first Drupal 8 theme.
type: theme
base theme: classy
core: 8.x
version: VERSION

regions:
  header: Header
  primary_menu: 'Primary menu'
  secondary_menu: 'Secondary menu'
  breadcrumb: Breadcrumb
  help: Help
  hightlighted: Highlighted
  content: Content
  sidebar_first: 'Left sidebar'
  sidebar_second: 'Right Sidebar'
  footer: Footer
  page_top: 'Page top'
  page_bottom: 'Page Bottom'
```

2. Go to /admin/appearance and under the Acme theme choose "Install and set as default".

Things we might find in a .info.yml file

The following keys are items that we will often find in a *.info.yml file. Some are optional, some are requiered. These keys provide meta-data about your theme, and define some of the basic functionality.

1. `name` **Required** The human readable name will appear on the Appearance page, where you can activate your theme.

2. `description` **Required** The description is displayed on the Appearance page.

3. `type` **Required** The type key indicates the type of extension, e.g. module, theme or profile. For themes this should always be set to "theme".

4. `base theme` The theme can inherit the resources from another theme by defining it as a base theme. Not declaring this, will default to using "Stable" as the base theme.

5. `core` **Required** The core key specifies the version of Drupal core that your theme is compatible with.

6. `version` For modules hosted on drupal.org, the version number will be filled in by the packaging script. You should not specify it manually, but leave out the version line entirely.

7. `regions` Regions are declared as children of the regions key. You are required to have a content region. The regions we just declared are also the default regions that are enabled by core if you do not declare any regions in your .info.yml file.

Other items

1. `regions_hidden` will allow you to remove default regions from the output if you don't specifically declare any regions.

2. `screenshot: IMAGE_NAME.png` With the screenshot key you define a screenshot that is shown on the Appearance page. If you do not define this key then Drupal will look for a file named 'screenshot.png' or 'screenshot.svg' in the theme folder to display.

3. The `libraries` key can be used to add asset libraries to all pages where the theme is active.

   ```
   libraries:
     - THEMENAME/global-styling
     - THEMENAME/LIBRARY-NAME
   ```

   Theme libraries contain css and/or javascript. Theme libraries are declared in another type of YaML file (THEME.libraries.yml). We will go over libraries in later exercises.

4. The `libraries-override` and `libraries-extend` keys can be used to take control over the components of a library, remove items or the complete library, or add additional elements to a library.

5. The `stylesheets-remove` key is used to stop the addition of css components for core and contrib modules. However, it is better to do this with `libraries-override`.

   ```
   stylesheets-remove:
     - core/assets/vendor/normalize-css/normalize.css
     - '@classy/css/components/tabs.css'
   ```

6. `ckeditor_stylesheets` will allow you to have custom ckeditor styles attached to your theme

7. `quickedit_stylesheets` will allow you to have custom styles and javascript attached to the quickedit functionality

Questions you may have...

- What happens if I don't declare any regions?

- Why are some regions in `' '` and others not?

Done ☺