

The Essential
Tutorial:

HOW TO CREATE THEME IN MAGENTO 2



Part 1

Whoever you are an extension or theme developer, you should spend time reading this blog post because you'll understand more about theme and template structure in **Magento 2**.

I will divide this tutorial into 2 parts. The part one is about **primary elements in Theme package**. Part 2 (that I will post next week) will reveal how to customize in **Magento 2**.

Elements

The fundamental elements in theme package in Magento 2

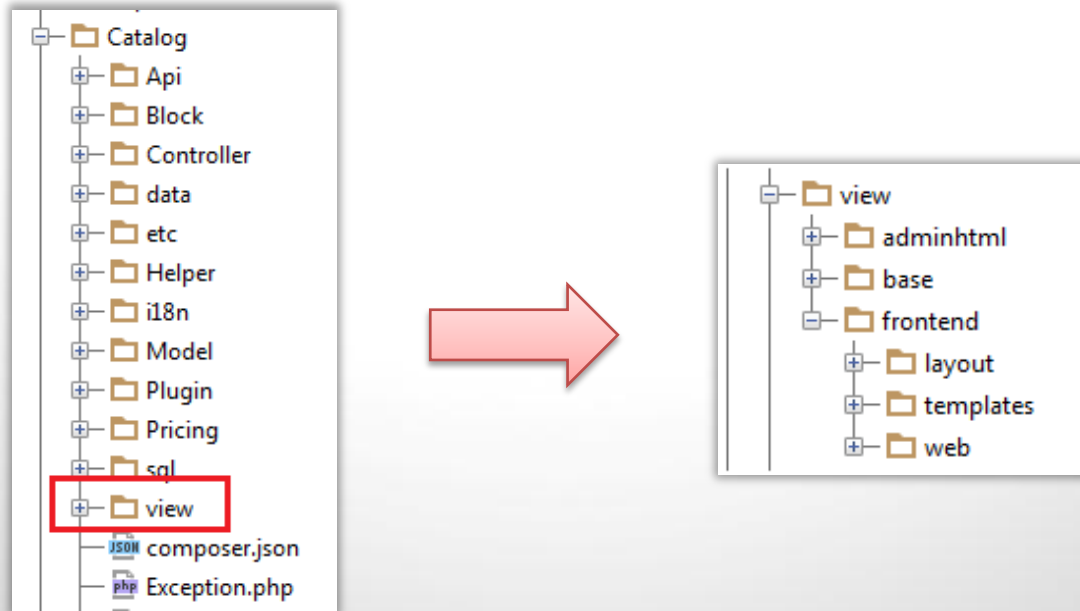
- a. Theme structure in Magento 2
- b. Layout
- c. New language style – LESS
- d. Mage_page element



a. Theme structure in Magento 2

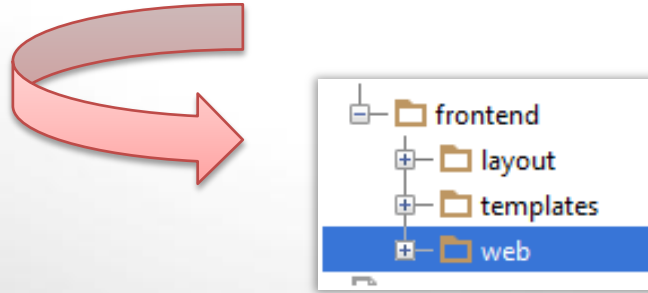
MVC structure in Magento 2 is clearer than in Magento 1.x. Modules in Magento 2 will also be added View element in the module folder structure. Catalog module: `app/code/Magento/Catalog/` is an example for you.

Please pay attention to View element which is in the same position with Controller, Model. In the **View** folder, there are below elements:



We can see that the inside structure has 3 elements, in which base element is moved out from <area>. The folder structure in <area> folder includes layout, templates, web.

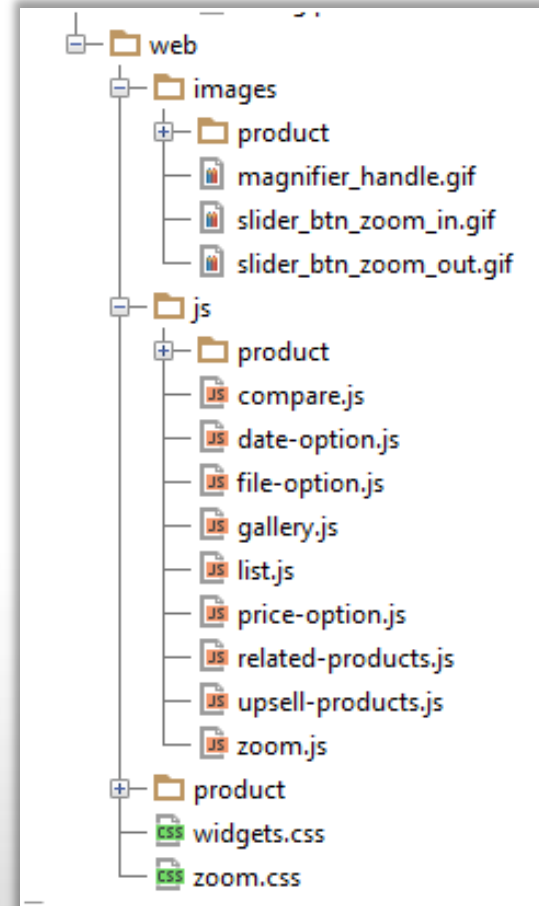
For example, frontend has 3 fundamental folders



- + **Layout folder** contains all the layout file of module (similar to layout file of Magento 1.x which is contained in all layout folders of theme). The code of these layouts, of course, have different structure. I will write in details in the specific part for layout file below.
- + **Template folder** has all template.phtml file, which is file rendered into html as we know in Magento 1.x, inside code is not very different, just php code mixes with html code.
- + **Web folder** is a totally new folder in Magento 2. You can see the image in the next slide

New **Web folder** in Magento 2

+ The folders: **css – stylesheet, js, media – images** are grouped. It replaces for 3 folders: **css, images, js** in previous skin/frontend/base/default. In other words, in Magento 2, the old skin folder is removed and divided into elements in each module.

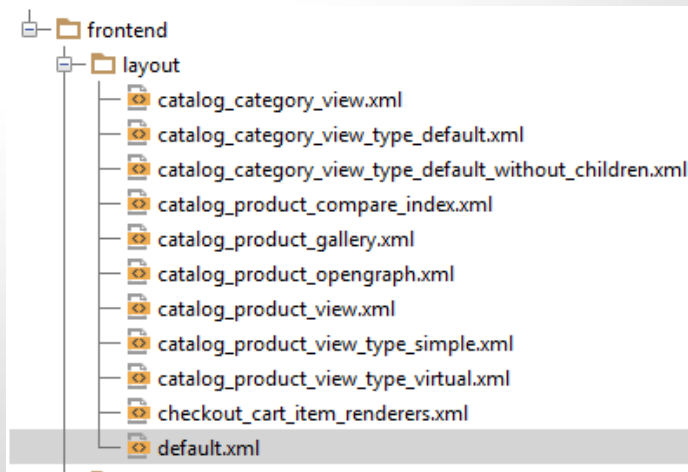


b. Layout

In Magento 2, each module has a default layout which can be overwritten and expanded by another layout.

Eg: `app/code/Magento/Catalog/view/frontend/layout/default.xml`

Magento 2 separates layout files into particular handle



Handle declaration is nearly the same as the previous rule:

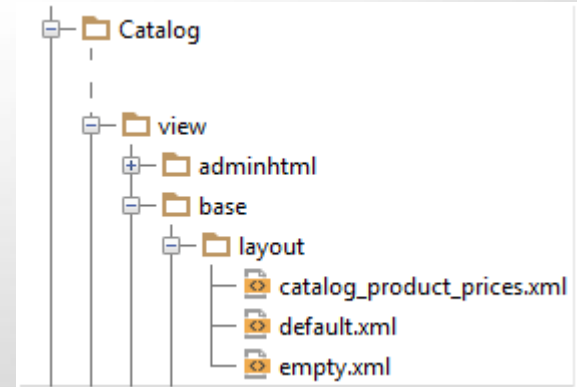
catalog_product_view – is a handle according to module and controller action

The definition for page we want to apply the layout is more specifically by declaring

catalog_product_view_type_simple_id_128

Or we can call to include other handles

Here is the example about layout file position of Catalog module:



Handle declaration is nearly the same as the previous rule:

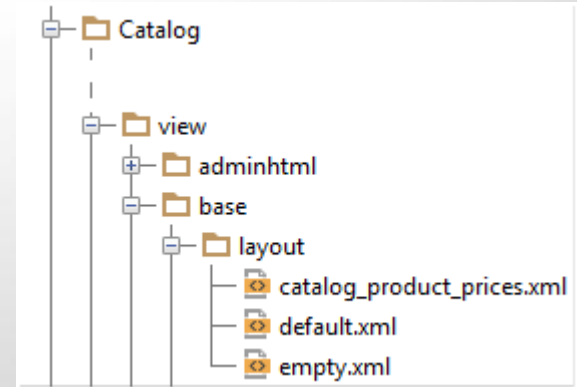
catalog_product_view – is a handle according to module and controller action

The definition for page we want to apply the layout is more specifically by declaring

catalog_product_view_type_simple_id_128

Or we can call to include other handles

Here is the example about layout file position of Catalog module:



Base Layout

+ **Base layout** is a default layout area of each module. It's not recommended to edit in these layout files if it's not your custom module.

Eg:

app/code/Magento/Checkout/view/frontend/layout/checkout_cart_item_renderers.xml

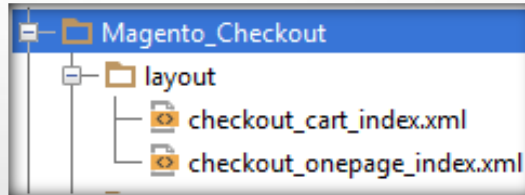
__app/code/<Namespace>/<Module> | __/view | __/<area> | __/layout | -
<layout_file1>.xml | - <layout_file2>.xml

Theme Layout

+ **Theme layouts** is the theme outside the module, allowing customizing default layout of module by reporting corresponding `<Namespace>_<Module>` in theme folder

Eg:

app/design/frontend/Magento/blank/Magento_Checkout/layout/



To create a layout file, follow the rules:

- + Each layout file calls for one handle and others called
- + Name of layout file is the name of handle layout. Eg: checkout_cart_index
- + Layout file called is in layout folder
- + Eg:

```
<layout xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<update handle="page_one_column" />
  <referenceContainer name="content">
    <!-- ... -->
  </referenceContainer>
</layout>
```

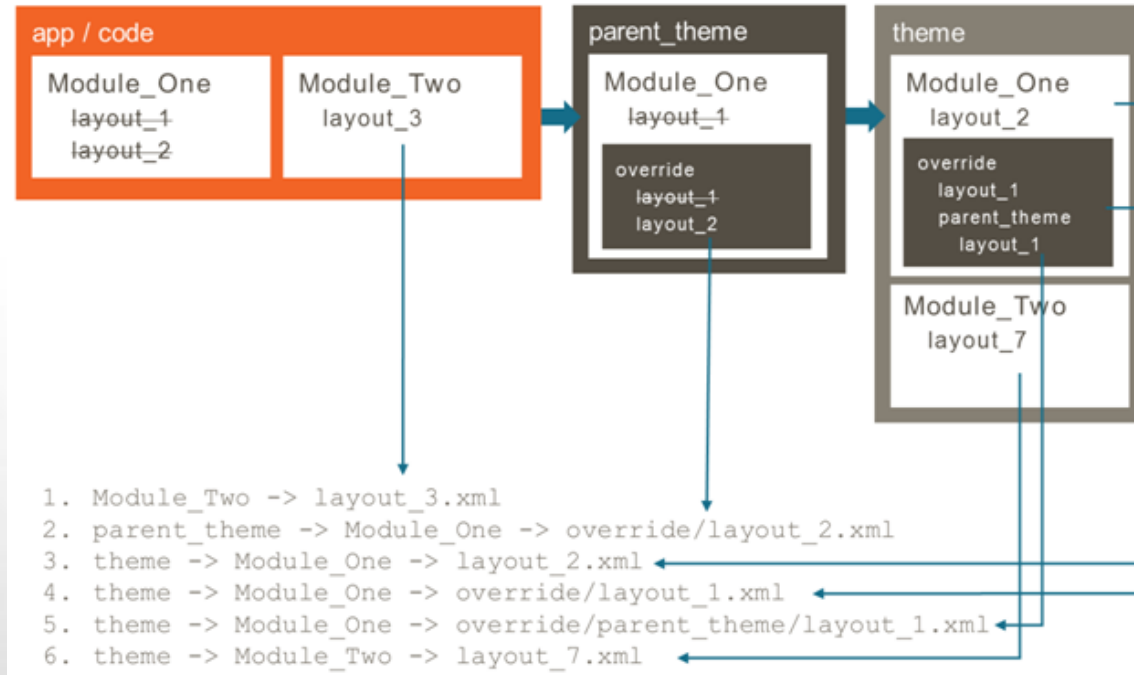
The order to read and process layout files is described as below:

- + If layout file belongs to different module, the order to run will be: independent module, dependent module and then alphabetical order of file name.
- + If file belongs to the same module, the order will be alphabetical order of file name

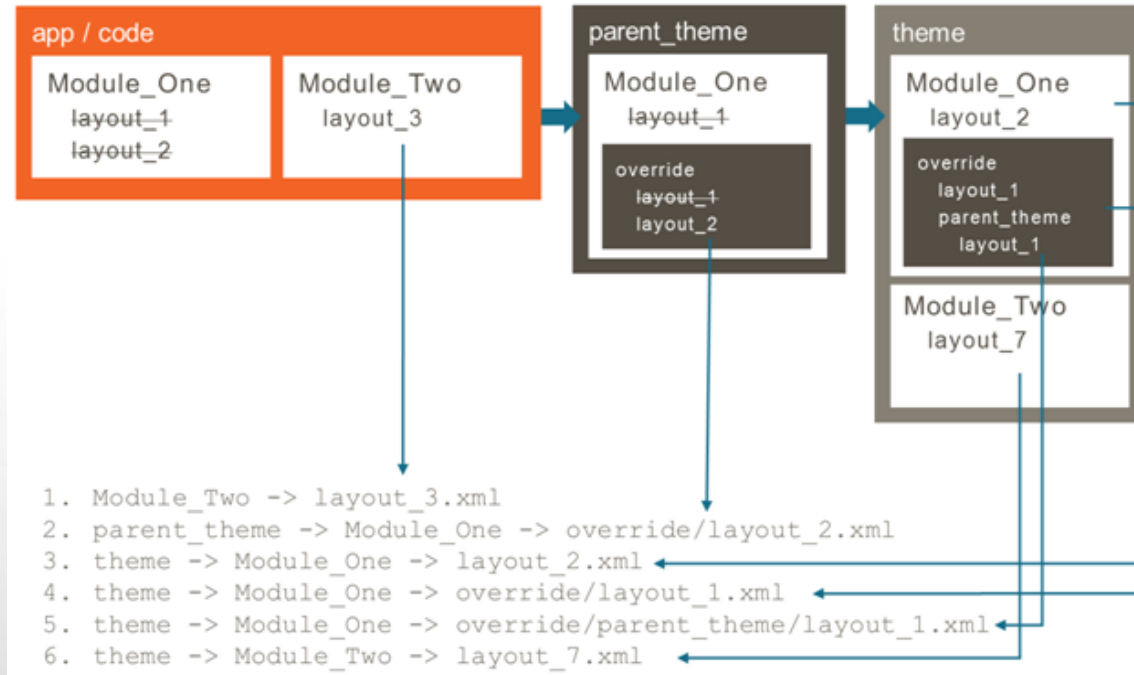
The steps to proceed layout of the system:

- + Read the group of all default layout files (base), including dependent ones
- + Determine the order of inherited or overwritten files
- + Add all expanded layout, replace layout in base, get in parent layout, replace parent layout files which are overwritten by the child ones

Here is the model of processing layout

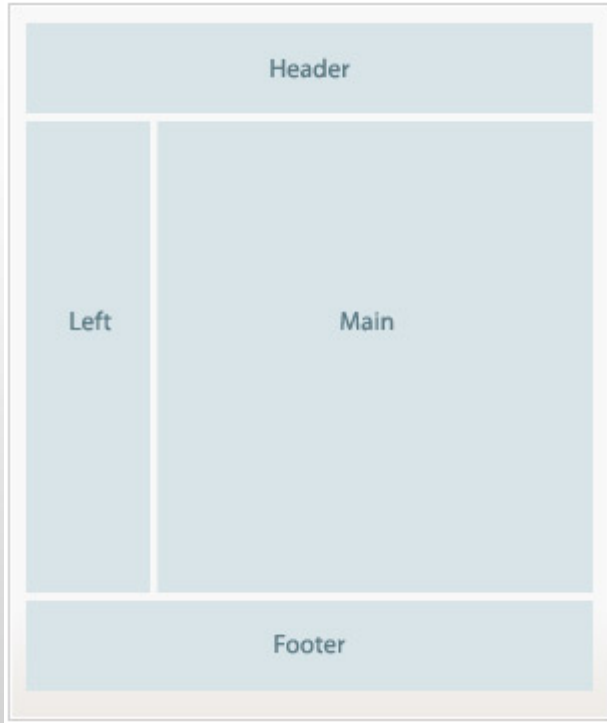


Here is the model of processing layout



Tip: Read [this blog post](#) to see detailed explanation

Layout/Block Area



+ **Container:** The area to distribute page elements



Layout/Block Area



+ **Block:** the area where default block elements of Magento are in container



The sample of layout content

```
<container name="container.1" label="Container 1"
as="container_1" output="1" htmlTag="div"
htmlId="container-1"
htmlClass="container">
<block class="Magento\Module\Block\Class" name="block.1">
<container name="child.container" label="Child Container"
as="child">
<block type="Magento\Module\Block\Class" name="block.2">
</container>
<block class="Magento\Module\Block\Class"
name="block.3"/>
</container>
<container name="container.2" as="container_2"
htmlTag="div"
htmlId="container-2" htmlClass="container"/>
```

c. New language style - LESS

LESS is added as **higher-level than CSS**. It is translated into CSS thanks to LESS library in php. We're still allowed to add CSS URL in the source to use directly.

Use via LESS:

The pre-process LESS language in Magento 2 is through library in
`lib/internal/Magento/Css/PreProcesso`

c. New language style - LESS

LESS is added as **higher-level than CSS**. It is translated into CSS thanks to LESS library in php. We're still allowed to add CSS URL in the source to use directly.

Use via LESS:

The pre-process LESS language in Magento 2 is through library in `lib/internal/Magento/Css/PreProcesso`

Tip: Want to know more about LESS. Go to [this tutorial](#) and find an example

d. Mage_page element

The Mage_page element in the previous Magento version is replaced by Magento_Theme module



A young woman with long brown hair and black-rimmed glasses is smiling and talking on a blue and orange flip phone. She is wearing a black sleeveless top with a colorful floral pattern. The background is a blurred outdoor setting with trees and a brick building.

So, we're done for the 1st part. Have any questions? Tell us



Download PDF File

Visit Magestore Blog to
download pdf file of this
tutorial & update latest
tutorials about Magento 2

Download PDF File



Visit Magestore Blog to download pdf file of this tutorial & update latest tutorials about Magento 2

[Magestore Blog: How to Create theme in Magento 2](#)