

# M2 CPFED

# 3. Page Layouts

# / LESS / Deploy

**John Hughes**

Head of Magento - Fisheye Media Ltd.

# Overview.

- Page layouts
- etc/view.xml configuration
- LESS basics
- Magento 2 LESS usage
- Magento 2 LESS UI library
- Deployment modes and developer process

**BRACE YOURSELF**



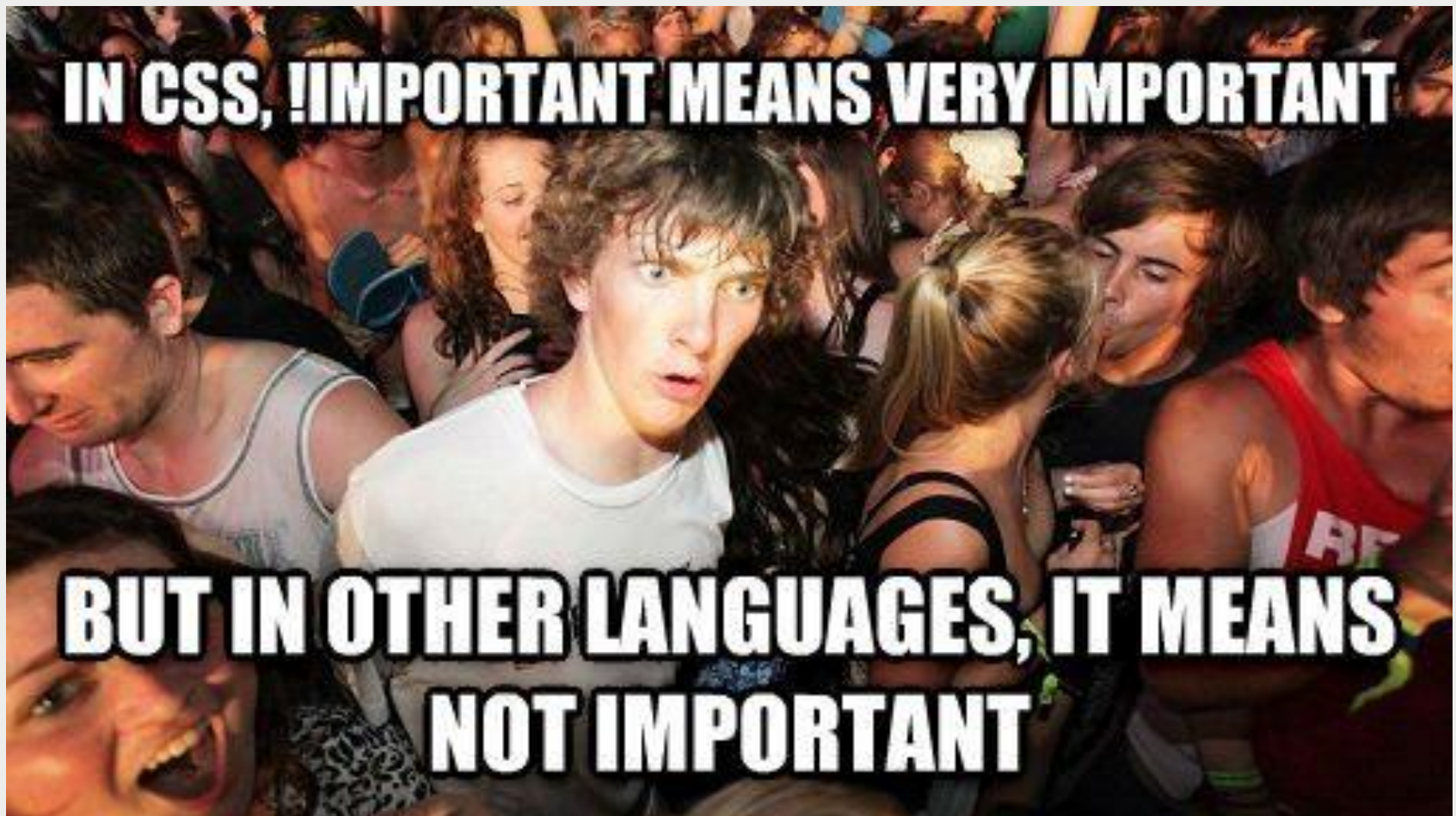
## Humanity's victories:



**land probe perfectly on a comet 310 million miles away, using science**



**get stuff on a web page to align properly  
using CSS**



**IN CSS, !IMPORTANT MEANS VERY IMPORTANT**

**BUT IN OTHER LANGUAGES, IT MEANS  
NOT IMPORTANT**

# Page layouts

# Page layouts overview.

- **What are page layouts?**
- How do you set a page layout in XML?
- Where can you override page layouts in the admin?

# Page layouts overview.

## What are page layouts?

- To allow for creation / reuse of common structural elements for a page e.g. defining:
  - the header, footer and main content areas
  - the number of columns within the main content area
- **No features (blocks), just structure (containers) allowed**

# Page layouts overview.

- What are page layouts?
- **How do you set a page layout in XML?**
- Where can you override page layouts in the admin?

# Page layouts overview.

## How do you set a page layout in XML?

- Within standard layout XML files
- In the `<page/>` node using the **layout** attribute

# <some\_handle>.xml

```
<?xml version="1.0"?>
<page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      layout="1column"
      xsi:noNamespaceSchemaLocation="...">
  <body>
    ...
  </body>
</page>
```

# Page layouts overview.

- What are page layouts?
- How do you set a page layout in XML?
- **Where can you override page layouts in the admin?**

# Page layouts overview.

Where can you override page layouts in the admin?

- CMS pages
- Categories
- Products

Note: under the 'Design' tab



# Joust Duffle Bag

← Back

Add Attribute

Save



Customizable Options



Product in Websites



Design



Layout  
[store view]

✓ No layout updates

Empty

1 column

2 columns with left bar

2 columns with right bar

3 columns

Display Product Options In  
[store view]

Layout Update XML  
[store view]

Schedule Design Update



Gift Options



Downloadable Information



DASHBOARD

SALES

CATALOG

CUSTOMERS

MARKETING

CONTENT

REPORTS

STORES

SYSTEM

FIND PARTNERS  
& EXTENSIONS

# Existing page layouts.

- **What are the standard page layout types?**
- Where would you find them?
- What is the 'empty' page layout?

# Existing page layouts.

What are the standard page layout types?

- empty
- 1column
- 2column-left
- 2column-right
- 3column

# Existing page layouts.

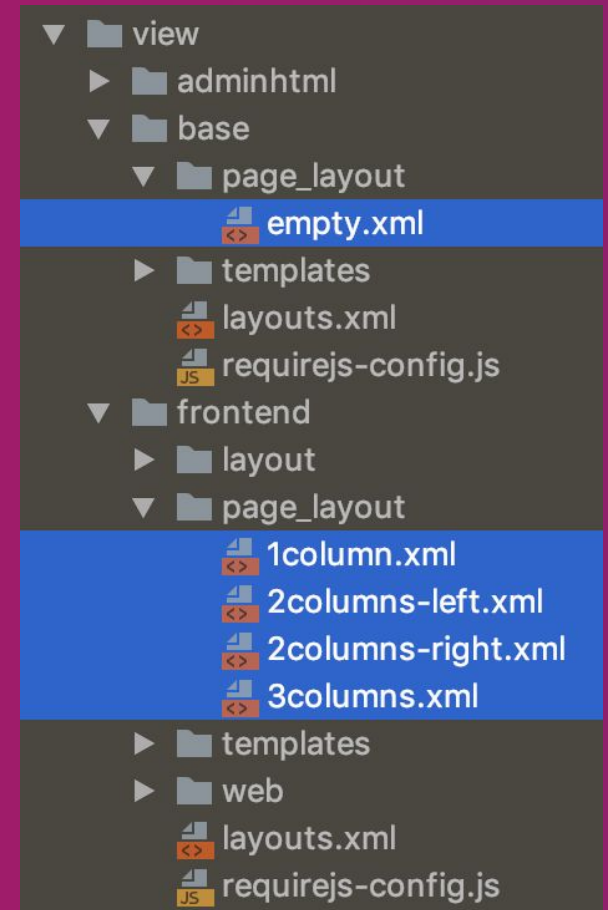
- What are the standard page layout types?
- **Where would you find them?**
- What is the 'empty' page layout?

# Existing page layouts.

Where would you find them?

- In **Magento\_Theme** module under:
  - **view/frontend/page\_layout/**\*

\*'empty' is found under the **base** area



# Existing page layouts.

- What are the standard page layout types?
- Where would you find them?
- **What is the ‘empty’ page layout?**

# Existing page layouts.

## What is the 'empty' page layout?

- A completely blank page
- No header / footer
- The base all other page layouts inherit from

# Creating page layouts.

- **What files are required to create a page layout?**
- What is the structure of a page layout file?
- How do you register a page layout file?
- How do you extend from other page layouts?

# Creating page layouts.

What files are required to create a page layout?

- **page\_layout/<file-name>.xml**
- **layouts.xml**

# Creating page layouts.

- What files are required to create a page layout?
- **What is the structure of a page layout file?**
- How do you register a page layout file?
- How do you extend from other page layouts?

# my\_page\_layout.xml

```
<?xml version="1.0"?>
```

```
<layout xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="urn:magento:framework:
        View/Layout/etc/page_layout.xsd">
```

```
    <container name="my.container">
```

```
        <container name="my.sub.container"/>
```

```
    </container>
```

```
</layout>
```

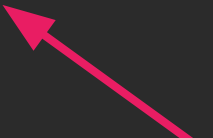
Specific XSD file

<layout> not <page>

# my\_page\_layout.xml

```
<?xml version="1.0"?>
<layout xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="urn:magento:framework:
        View/Layout/etc/page_layout.xsd">

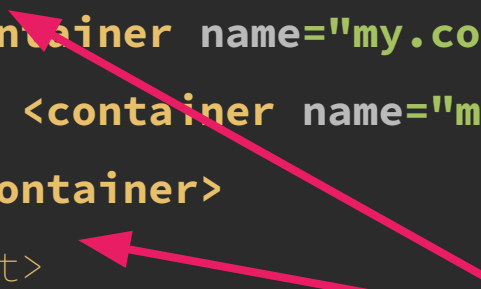
    <container name="my.container">
        <container name="my.sub.container"/>
    </container>
</layout>
```



**<container/> and <move/> only, no <block/>**

# my\_page\_layout.xml

```
<?xml version="1.0"?>
<layout xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="urn:magento:framework:
        View/Layout/etc/page_layout.xsd">
    <container name="my.container">
        <container name="my.sub.container"/>
    </container>
</layout>
```



No need for **<body/>**, **<layout/>** is part of the page

# Creating page layouts.

- What files are required to create a page layout?
- What is the structure of a page layout file?
- **How do you register a page layout file?**
- How do you extend from other page layouts?

# Creating page layouts.

How do you register a page layout file?

- In **layouts.xml**

# layouts.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<page_layouts xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:noNamespaceSchemaLocation="urn:magento:framework:
              View/PageLayout/etc/layouts.xsd">
  <layout id="my_page_layout">
    <label translate="true">My Page Layout</label>
  </layout>
</page_layouts>
```

Specific XSD file



<page\_layouts>



# layouts.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<page_layouts xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="urn:magento:framework:
    View/PageLayout/etc/layouts.xsd">
    <layout id="my_page_layout">
        <label translate="true">My Page Layout</label>
    </layout>
</page_layouts>
```



**For admin use**

**id must match page layout file name**

# Creating page layouts.

- What files are required to create a page layout?
- What is the structure of a page layout file?
- How do you register a page layout file?
- **How do you extend from other page layouts?**

# Creating page layouts.

How do you extend from other page layouts?

- Use `<handle/>`
- Same principle as with standard layout files

# layouts.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<page_layouts xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="urn:magento:framework:
    View/PageLayout/etc/layouts.xsd">
    <update handle="1column">
        <layout id="my_page_layout">
            <label translate="true">My Page Layout</label>
        </layout>
    </page_layouts>
```

Specify page layout id

**etc/view.xml  
configuration**



PROFESSIONAL  
SOLUTIONS PARTNER

**:fisheye**

# etc/view.xml configuration.

- **What is etc/view.xml for?**
- How do you customise etc/view.xml?
- How can you retrieve values set in etc/view.xml?
- How does inheritance for etc/view.xml work?

# etc/view.xml configuration.

## What is etc/view.xml for?

- Theme configuration values
  - Image sizes
  - Product gallery settings
  - JS bundling exclusions (files and folders)

# etc/view.xml configuration.

- What is etc/view.xml for?
- **How do you customise etc/view.xml?**
- How can you retrieve values set in etc/view.xml?
- How does inheritance for etc/view.xml work?

# etc/view.xml configuration.

## How do you customise etc/view.xml?

- Copy the file to your current theme\*
- Change the values required

\*You can also add **etc/view.xml** in modules

# etc/view.xml configuration.

- What is etc/view.xml for?
- How do you customise etc/view.xml?
- **How can you retrieve values set in etc/view.xml?**
- How does inheritance for etc/view.xml work?

# etc/view.xml configuration.

How can you retrieve values set in etc/view.xml?

- Use **`$block->getVar()`**

# <some-file>.phtml

```
$block->getVar($name, $module = null)
```

```
$block->getVar('some_var', 'Vendor_Module')
```

# etc/view.xml configuration.

- What is etc/view.xml for?
- How do you customise etc/view.xml?
- How can you retrieve values set in etc/view.xml?
- **How does inheritance for etc/view.xml work?**

# etc/view.xml configuration.

## How does inheritance for etc/view.xml work?

- Same as standard theme inheritance
  - Child theme -> Parent theme(s) -> Module(s)
- Full file contents must be copied
  - XML is **NOT** merged like layout files

# LESS basics

# LESS overview.

- **What is LESS?**
- What are the main CSS files in Magento 2?
- How are LESS/CSS files included on a page?

# LESS overview.

## What is LESS?

- **LE**aner **S**tyle **S**heets
- CSS Preprocessor
- Provides features such as variables, nesting, mathematical operations and mixins (functions)

# LESS overview.

- What is LESS?
- **What are the main CSS files in Magento 2?**
- How are LESS/CSS files included on a page?

# LESS overview.

File	Purpose
<b>styles-m.css</b>	Mobile viewport styles
<b>styles-l.css</b>	Desktop viewport styles (for 768px and greater)
<b>print.css</b>	Print styles

# LESS overview.

## What are the main CSS files in Magento 2?

- These main files are compiled from multiple '**component**' LESS files across modules and themes
- '**Component**' LESS files usually begin with an underscore to denote they are not individually output

# LESS overview.

- What is LESS?
- What are the main CSS files in Magento 2?
- **How are LESS/CSS files included on a page?**

# LESS overview.

How are LESS/CSS files included on a page?

- Via layout XML in the **<head/>** section using **.css**
- Can use **<css/>** or **<script/>** tag
- Normally included in **default\_head\_blocks.xml** handle
- **.less** files will automatically be compiled to **.css**

# default\_head\_blocks.xml

```
<?xml version="1.0"?>
<page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="...">
  <head>
    <css src="css/local-theme-styles.css" >
      <css
src="FisheyeAcademy_Training::css/my-styles.css" >
        <css
src="https://www.example.com/external-styles.css" >
      </head>
    </page>
```

**Local files must be placed within a web/  
subdirectory, whether in a theme or module**

# LESS core concepts.

- **@import** directive
- Variables
- Mixins
- Nesting
- Calculations

# LESS core concepts.

## @import directive

- To include other LESS files
- Relative to current file

# styles-m.less

```
@import 'source/_reset.less';  
@import '_styles.less';  
@import (reference) 'source/_extends.less';
```

# LESS core concepts.

- @import directive
- **Variables**
- Mixins
- Nesting
- Calculations

# LESS core concepts.

## Variables

- Assign values for future use

# <\_my-file>.less

```
// Assign variable
```

```
@my-favourite-colour: orange;
```

```
// Use variable
```

```
body {
```

```
    background-color: @my-favourite-colour;
```

```
}
```

# LESS core concepts.

- @import directive
- Variables
- **Mixins**
- Nesting
- Calculations

# LESS core concepts.

## Mixins

- Like a method or function
- **DRY** principle (write it once, use it multiple times)
- Written in plain CSS (or LESS)
  - Then called by adding ()
  - Can also pass parameters

# <\_my-file>.less

```
// Create mixin (standard CSS)
.mixin-class {
    background-color: orange;
}
```

```
// Use mixin elsewhere
.another-class {
    .mixin-class();
}
```

```
// Plain CSS output of above
.another-class {
    background-color: orange;
}
```

# <\_my-file>.less

```
// Create mixin (with parameter)
.mixin-class(@color) {
    background-color: @color;
}
```

```
// Use mixin and pass value for parameter
.mixin-class {
    .some-class(pink);
}
```

```
// Plain CSS output of above
.another-class {
    background-color: pink;
}
```

# LESS core concepts.

- @import directive
- Variables
- Mixins
- **Nesting**
- Calculations

# LESS core concepts.

## Nesting

- Allows writing styles for adjacent or child elements in a nested sections
- No limit to how many levels of nesting can be used, but best practice is to go no more than 3 levels deep

# <\_my-file>.less

// Child elements - Plain CSS

```
.element {  
    color: orange;  
}
```

```
.element .child-element {  
    color: pink;  
}
```

# <\_my-file>.less

```
// Child elements - LESS nesting
.element {
    color: orange;

    .child-element {
        color: pink;
    }
}
```

# <\_my-file>.less

// Concatenation - Plain CSS

```
.element {  
    color: orange;  
}
```

```
.element.active {  
    color: pink;  
}
```

# <\_my-file>.less

```
// Concatenation - LESS nesting
.element {
    color: orange;

    &.active {
        color: pink;
    }
}
```

# <\_my-file>.less

```
// Media queries - plain CSS
```

```
.element {  
    color: orange;  
}
```

```
@media screen and (min-width: 768px) {  
    .element {  
        color: pink;  
    }  
}
```



PROFESSIONAL  
SOLUTIONS PARTNER

:fisheye

# <\_my-file>.less

```
// Media queries - LESS
```

```
.element {
```

```
    color: orange;
```

```
    @media screen and (min-width: 768px) {
```

```
        color: pink;
```

```
    }
```

```
}
```

# LESS core concepts.

- @import directive
- Variables
- Mixins
- Nesting
- **Calculations**

# LESS core concepts.

## Calculations

- Allows using mathematical operators
- Powerful when combined with variable

# <\_my-file>.less

```
// Plain CSS
```

```
.element1 {  
    border-width: 1px;  
}
```

```
.element2 {  
    border-width: 2px;  
}
```

```
.element3 {  
    border-width: 3px;  
}
```

# <\_my-file>.less

```
// LESS
```

```
@border-width__one: 1px;
```

```
@border-width__two: 2px;
```

```
.element1 {
```

```
    border-width: @border-width__one;
```

```
}
```

```
.element2 {
```

```
    border-width: @border-width__two;
```

```
}
```

```
.element3 {
```

```
    border-width: @border-width__one + @border-width__two;
```

```
}
```

# Magento 2 LESS usage

# Magento 2 LESS overview.

- What is the **@magento\_import** directive?
- Where can Magento's implementation be found?
- What is the custom Magento processing step?
- What is the developer workflow for LESS files?

# Magento 2 LESS overview.

What is the **@magento\_import** directive?

- Customised version of the @import directive
- Allows scanning for specified file paths across predefined directories (modules)

# styles-m.less

```
//@magento_import 'source/_my-styles.less';
```

# Magento 2 LESS overview.

## What is the **@magento\_import** directive?

- Must be commented out (//) to avoid conflict with actual LESS directives / compilation
- Example Looks for '**source/\_my-styles.less**' in every module and theme
  - Module: **view/<area>/web/css/source/\_my-styles.less**
  - Theme: **Magento\_Catalog/web/css/source/\_my-styles.less**
  - Theme: **web/css/source/\_my-styles.less**

# Magento 2 LESS overview.

- What is the @magento\_import directive?
- **Where can Magento's implementation be found?**
- What is the custom Magento processing step?
- What is the developer workflow for LESS files?

# Magento 2 LESS overview.

Where can Magento's implementation be found?

- Classes: `\Magento\Framework\Css\...`
- Filepath: `vendor/magento/framework/Css/*`

# Magento 2 LESS overview.

- What is the @magento\_import directive?
- Where can Magento's implementation be found?
- **What is the custom Magento processing step?**
- What is the developer workflow for LESS files?

# Magento 2 LESS overview.

## What is the `@magento_import` directive?

- When Magento looks for all `.less` files to convert to `.css` it also resolves all the `//@magento_import` directives to actual files
- Files are then compiled into the final `styles-m.css`, `styles-l.css` and `print.css` files

# Magento 2 LESS overview.

- What is the @magento\_import directive?
- Where can Magento's implementation be found?
- What is the custom Magento processing step?
- **What is the developer workflow for LESS files?**

# Magento 2 LESS overview.

## What is the developer workflow for LESS files?

- New files, run: **bin/magento dev:source-theme:deploy**
- Existing files: refresh your browser
- Ideally use inbuilt support for **grunt** task runner
- Note: all files are symlinked in **developer** mode

# Working with M2 LESS.

- **What are `_module.less`, `_widgets.less`, `_extend.less`?**
- What are they for?
- When / how should they be overridden?
- What is `_extends.less`?

# Working with M2 LESS.

What are `_module.less`, `_widgets.less`, `_extend.less`?

- Default files included via **`//@magento_import directive`**
  - i.e. files that are available to be placed in any module or theme (see previous slides) under the **`web/css/source`** directory
- Then resolved via Magento's custom LESS processing

# Working with M2 LESS.

- What are `_module.less`, `_widgets.less`, `_extend.less`?
- **What are they for?**
- When / how should they be overridden?
- What is `_extends.less`?

# Working with M2 LESS.

File	Purpose
<b>_module.less</b>	Core module styles
<b>_widgets.less</b>	Styles for Magento widgets
<b>_extend.less</b>	To make minor changes to a modules styles (loaded after _module.less)

# Working with M2 LESS.

- What are `_module.less`, `_widgets.less`, `_extend.less`?
- What are they for?
- **When / how should they be overridden?**
- What is `_extends.less`?

# Working with M2 LESS.

File	Overridden
<b>_module.less</b>	Only if you want to completely replace / change the styles for a module
<b>_widgets.less</b>	If you want to change / replace the widget styles
<b>_extend.less</b>	To make minor changes to a modules styles (loaded after <b>_module.less</b> )

# Working with M2 LESS.

When / how should they be overridden?

- Files can be overridden by placing them in the same location in the currently active theme
  - Within the relevant module or theme **web/css** directory
- i.e. see theme fallback from workshop 1

# Working with M2 LESS.

- What are `_module.less`, `_widgets.less`, `_extend.less`?
- What are they for?
- When / how should they be overridden?
- **What is `_extends.less`?**

# Working with M2 LESS.

What is `_extends.less`?

- Does **NOT** use `//@magento_import`
- Contains a list of abstract **selectors** that can be 'extended' from as well as **mixins**

# Magento 2 LESS UI library

# Magento 2 LESS UI library.

- **What is the Magento 2 UI library?**
- Where is the UI library located?
- How can you use / extend / customise it?

# Magento 2 LESS UI library.

## What is the Magento 2 UI library?

- A library of mixins for reuse to simplify theming
- <https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/css-topics/theme-ui-lib.html>



PROFESSIONAL  
SOLUTIONS PARTNER

**fish**eye

# Magento 2 LESS UI library.

- What is the Magento 2 UI library?
- **Where is the UI library located?**
- How can you use / extend / customise it?

# Magento 2 LESS UI library.

Where is the UI library located?

- **lib/web/css/source/lib**

# Magento 2 LESS UI library.

- What is the Magento 2 UI library?
- Where is the UI library located?
- **How can you use / extend / customise it?**

# Magento 2 LESS UI library.

How can you use / extend / customise it?

- New variables should be placed in **lib/web/css**
- Overrides of existing variables should be declared in:
  - **<theme\_dir>/web/css/ source/\_theme.less**

# Magento 2 LESS UI library.

How can you use / extend / customise it?

- You can also copy files into your theme to override them
  - Under **lib/web/css** e.g. **lib/web/css/\_dropdowns.less**
- Use sparingly!

# Deployment modes and developer process

# Magento modes.

- **What are the 3 modes Magento 2 can be run in?**
- How does each mode differ / when should they be used?
- How do you view / switch between modes?

# Magento modes.

What are the 3 modes Magento 2 can be run in?

- **default**
- **developer**
- **production**

# Magento modes.

- What are the 3 modes Magento 2 can be run in?
- **How does each mode differ / when should they be used?**
- How do you view / switch between modes?

# Magento modes.

Mode	Characteristics
<b>default</b>	Static view files symlinked and cached Errors / exceptions logged
<b>developer</b>	Static view files symlinked, not cached Verbose errors / logging / debugging Code (PHP) compiled 'on the fly'
<b>production</b>	Static view files copied (do not change) Errors / exceptions logged Code (PHP) compiled and cached

\*There is also '**maintenance**' mode (503 holding page)

# Magento modes.

Mode	Use case
<b>default</b>	<b>NEVER</b> (shared hosting / DIY merchants)
<b>developer</b>	When developing Magento extensions / themes
<b>production</b>	On production (live) servers for maximum performance

# Magento modes.

- What are the 3 modes Magento 2 can be run in?
- How does each mode differ / when should they be used?
- **How do you view / switch between modes?**

# Magento modes.

How do you view / switch between modes?

- Via CLI (command line interface)
  - Or editing value in **app/etc/env.php** (not advised)
- Using **bin/magento**
- You can't switch back to **default** mode

# Terminal

# see the current mode

```
bin/magento deploy:mode:show
```

# switch to different mode

```
bin/magento deploy:mode:set <mode>
```

```
bin/magento deploy:mode:set developer
```

```
bin/magento deploy:mode:set production
```

# Cache management.

- **How can you manage cache in the admin panel?**
- How can you manage cache on the CLI?

# Cache management.

How can you manage cache in the admin panel?

- **SYSTEM > Tools > Cache Management**
- **Note:** cache can't be disabled in the admin panel when in production mode

# Cache management.

- How can you manage cache in the admin panel?
- **How can you manage cache on the CLI?**

# Terminal

```
# see which caches types enabled / disabled  
bin/magento cache:status
```

# Terminal

```
# enable cache type(s)
```

```
bin/magento cache:enable <type>
```

```
bin/magento cache:enable block_html
```

```
# enable all cache types
```

```
bin/magento cache:enable
```

# Terminal

```
# disable cache type(s)
```

```
bin/magento cache:disable <type>
```

```
bin/magento cache:disable config layout
```

```
# disable all cache types
```

```
bin/magento cache:disable
```

# Terminal

```
# clean cache type(s)
```

```
bin/magento cache:clean <type>
```

```
bin/magento cache:clean full_page
```

```
bin/magento cache:clean block_html layout
```

```
# clean all cache types
```

```
bin/magento cache:clean
```

# Terminal

```
# flush entire cache storage  
bin/magento cache:flush
```

# Static assets.

- **What are static assets and where would you find them?**
- How and when would you deploy static assets?
- Explain asset minification, merging and static signing
- What common issues can occur?
- How do the LESS compilation modes work / differ?

# Static assets.

What are static assets and where would you find them?

- Files requested in the browser (client side)
  - e.g. **stylesheets, javascript, images, fonts**
- Static assets are compiled to:
  - **pub/static/<area>/<theme-package>/<theme-name>/<locale>**
  - e.g. **pub/static/frontend/FisheyeAcademy/training/en\_GB**

# Static assets.

- What are static assets and where would you find them?
- **How and when would you deploy static assets?**
- Explain asset minification, merging and static signing
- What common issues can occur?
- How do the LESS compilation modes work / differ?

# Static assets.

How and when would you deploy static assets?

- In **production** mode **ONLY** to copy / compile files
  - In developer mode these are symlinked / auto updated
- Static assets are deployed using **bin/magento** CLI tool

# Terminal

```
# deploy static content
```

```
bin/magento setup:static-content:deploy
```

```
bin/magento s:s:d # shorthand version
```

# Terminal

# deploy for specific theme

```
bin/magento s:s:d --theme[=THEME]
```

```
bin/magento s:s:d --theme=FisheyeAcademy/training
```

# deploy for specific locale (language)

```
bin/magento s:s:d --language[=LANGUAGE]
```

```
bin/magento s:s:d --language=en_GB
```

# Static assets.

- What are static assets and where would you find them?
- How and when would you deploy static assets?
- **Explain asset minification, merging and static signing**
- What common issues can occur?
- How do the LESS compilation modes work / differ?

# Static assets.

Explain asset minification, merging and static signing

- Minification reduces file size and therefore download time
- Merging combines all files together reducing the number of requests, but increasing overall download size

# Static assets.

Explain asset minification, merging and static signing

- Static file signing adds a versioning number as a pseudo directory to all static asset requests
- This version is updated every time static files are deployed
- This forces assets to be redownloaded in users browsers (i.e. not be returned from local browser cache)

# Static assets.

Explain minification and merging of assets

- Settings can be managed in the admin panel
- **STORES > Settings > Configuration**
  - **ADVANCED > Developer**
    - **Javascript Settings**
    - **CSS Settings**
    - **Static Files Settings**

# Static assets.

- What are static assets and where would you find them?
- How and when would you deploy static assets?
- Explain asset minification, merging and static signing
- **What common issues can occur?**
- How do the LESS compilation modes work / differ?

# Static assets.

## What common issues can occur?

- Not using an automated deployment tool (e.g. capistrano)
  - To mitigate / reduce downtime
- Wrong theme or locale deployed (or missed)
- Not using static file signing (to version files)
- Not using minification or merging where relevant
- Running **bin/magento s:s:d** in developer mode

# Static assets.

- What are static assets and where would you find them?
- How and when would you deploy static assets?
- Explain asset minification, merging and static signing
- What common issues can occur?
- **How do the LESS compilation modes work / differ?**

# Static assets.

How do the LESS compilation modes work / differ?

- Server side (PHP) and client side modes
- Server side compiled using PHP LESS library
- Client side compiled in browser on every page load
  - **VERY VERY SLOW!**

# Static assets.

How do the LESS compilation modes work / differ?

- Mode can be set in the admin panel
- **STORES > Settings > Configuration**
  - **ADVANCED > Developer > Frontend Developer Workflow**

# Wrap up / resources

# Code challenge 1.

- Add a new page layout for product pages that adds an additional container
- Add another block to your new page layout container
- Remove the message on configurable product pages only

# Code challenge 2.

- Modify gallery settings in **etc/view.xml** and review impact on product page
- Add a new variable to **etc/view.xml** and output in a template file

# Code challenge 3.

- Add a new CSS file via layout XML
- Create a LESS file that matches the file added in layout
- Change body colour using a variable (and a mixin)
- Implement nesting and calculations to change the colour at a different breakpoint (screen size)

# Code challenge 4.

- Override a modules **\_module.less** and modify the styles
  - Try removing bulk of styles to see how bad the page looks
- Revert above and use **\_extend.less** to make minor changes

# Code challenge 5.

- Read up on Magento LESS UI library
- Override variables using **\_theme.less**
- Override a default component from the library and modify

# Credit / Resources / Thanks.

- **Official Magento Study Guide**
- **Swift Otter Study Guide**
- **Magento DevDocs**