M2 CPFED
2. Layout XML / Templates

John Hughes

Head of Magento - Fisheye Media Ltd.

PROFESSIONAL SOLUTIONS PARTNER

:fisheye

Overview.

- Introduction
- Containers and blocks
- Layout handles
- Layout file structure
- Layout directives

Overview cont.

- Templates
- Arguments
- Overriding and replacing template files
- Escaping content
- Layout merging, overrides and processing

Not being able to read a players hand...

…is as frustrating as not being able to work out why your layout isn't working!

Introduction

Introduction.

- **What is layout XML?**
- What is a Block class?
- What are template files for?

Introduction.

What is layout XML?

- Defines the content on a given page
    - via 2 element types: containers and blocks
- Determine the page structure / hierarchy
    - Also known as the ‘element tree’

Introduction.

- What is layout XML?
- **What is a Block class?**
- What are template files for?

Introduction.

What is a Block class?

- A PHP class that collates and transforms data for presentation (output) on the frontend
    - In most cases in a template file

Introduction.

- What is layout XML?
- What is a Block class?
- **What are template files for?**

Introduction.

What are template files for?

- Outputting HTML markup and data passed from:
  - Block classes
  - Layout XML

Containers and blocks

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Containers.

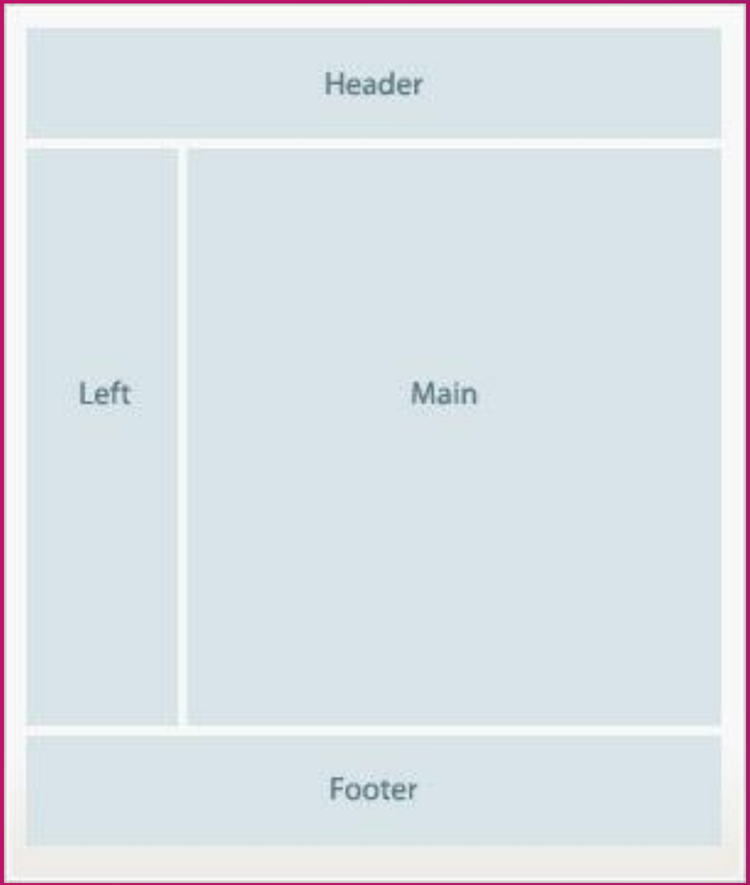
- **What are containers?**
- What attributes can containers have?

Containers.

What are containers?

- For assigning content structure on a given page
- As the name suggests they contain other elements:
    - Blocks as well as other containers
- Their contents can be wrapped in an HTML element

Containers.



- Examples:
    - Header
    - Left column
    - Main column
    - Footer

Blocks.

- **What are blocks?**
- What attributes can blocks have?

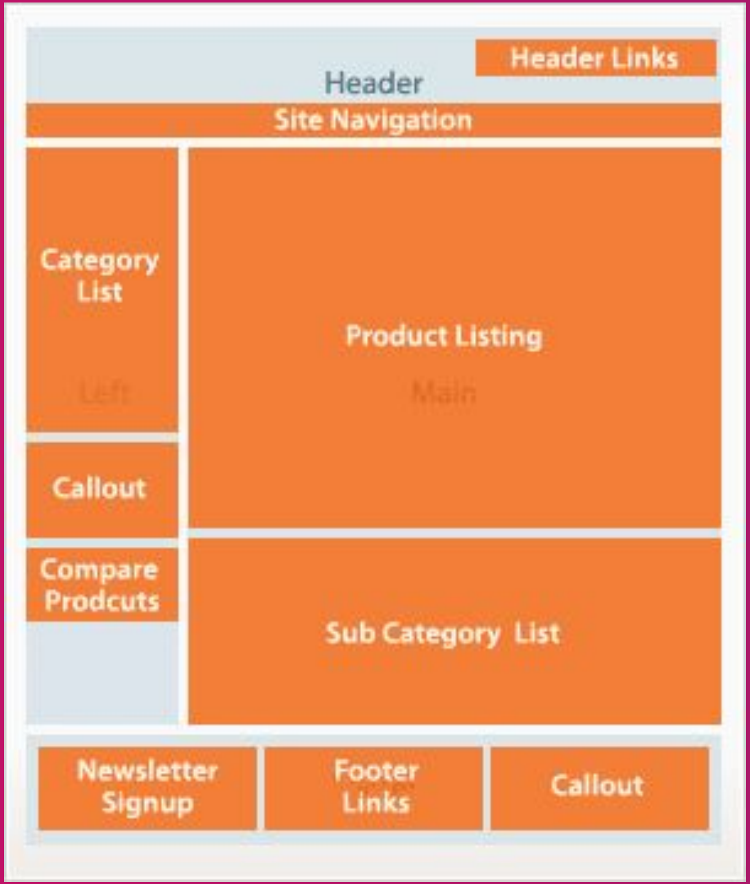
Blocks.

What are blocks?

- An individual feature or view component on a page
- Their content is generally output in a template (**.phtml** file) and data passed from a Block class

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Blocks.



- Examples:
  - Navigation menu
  - Product list
  - Media gallery
  - Newsletter signup

Layout files and handles

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Layout files.

- **What components can layout files be found in?**
- What is the file structure of layout files?

Layout files.

What components can layout files be found in?

- Modules
- Themes

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Layout files.

- What components can layout files be found in?
- **What is the file structure of layout files?**

Layout files.

What is the file structure of layout files?

- **Modules**
  - view/**<area>**/layout/**<layout_handle>**.xml
- **Themes**
  - **<Vendor_Module>**/layout/**<layout_handle>**.xml

Layout handles.

- **What is a layout handle?**
- How are handles linked to routing?
- What are common layout handles?
- How many handles can be on one page?

Layout handles.

What is a layout handle?

- A string that associates a layout XML file with a specific page or group of layout instructions
- e.g. the **customer_account_login** handle relates to the file of the same name: **customer_account_login.xml**

Layout handles.

- What is a layout handle?
- **How are handles linked to routing?**
- What are common layout handles?
- How many handles can be on one page?

Layout handles.

How are handles linked to routing?

- Whilst handles can be programmatically added to a page
- Most are generated based on the current page route / url

Layout handles.

How are handles linked to routing?

- http://www.myawesomestore.com/**customer/account/login**
- **customer_account_login**.xml

Layout handles.

- What is a layout handle?
- How are handles linked to routing?
- **What are common layout handles?**
- How many handles can be on one page?

Layout handles.

Handle	Usage
default	All pages
cms_index_index	Homepage
catalog_category_view	Category pages
catalog_product_view	Product detail pages
catalog_product_view_type_<type> e.g **catalog_product_view_type_simple**	Specific product pages e.g. simple, configurable

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Layout handles.

Handle	Usage
contact_index_index	Contact page
checkout_cart_index	Basket page
checkout_index_index	Checkout
customer_account	All account area pages
customer_account_index	Account dashboard
cms_page_view	CMS pages

Layout handles.

- What is a layout handle?
- How are handles linked to routing?
- What are common layout handles?
- **How many handles can be on one page?**

Layout handles.

How many handles can be on one page?

- Multiple, no limit
- At a minimum: **default** and the **current route handle**

Layout file structure

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Layout file structure.

- **What are the main layout file nodes?**
- What do they contain?

Layout file structure.

What are the main layout file nodes?

- <update/>
- <head/>
- <body/>

Layout file structure.

- What are the main layout file nodes?
- **What do they contain?**

<some_handle>.xml

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

    <update handle="some_other_handle"/>

    <head>
        <!-- Meta data, file includes (e.g. CSS/JS), removal etc.-->
    </head>

    <body>
        <!-- Container and block declarations -->
    </body>
</page>
```

Layout directives

PROFESSIONAL
SOLUTIONS PARTNER

:fisheye

Layout directives.

- **What are the main layout directives?**
- What are they for?

Layout directives.

What are the main layout directives?

- <container/>
- <block/>
- <referenceContainer/>
- <referenceBlock/>
- <move/>

Layout directives.

- What are the main layout directives?
- **What are they for?**

Layout directives.

Directive	Usage
<container/>	Add new containers
<block/>	Add new blocks
<referenceContainer/>	Update / add child elements to existing containers
<referenceBlock/>	Update / add child elements to existing blocks
<move>	Move existing blocks / containers

<container/>.

- What attributes can containers have?

<container/>.

What attributes can containers have?

- name
- htmlTag
- htmlClass
- before/after
- and more...

<some_handle>.xml

```
<?xml version="1.0"?>
<page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="...">
   <body>
       <container name="my.container"
                  htmlTag="div"
                  htmlClass="my-container">
           <!-- Add child elements here-->
       </container>
   </body>
</page>
```

<block/>.

- What attributes can blocks have?

<block/>.

What attributes can blocks have?

- name
- class
- template
- before/after
- cacheable (disables FPC for entire page if set to true!)
- and more...

<some_handle>.xml

```
<?xml version="1.0"?>
<page xmlns:xsi="..." xsi:noNamespaceSchemaLocation="...">
   <body>
       <container name="my.container"
                  htmlTag="div"
                  htmlClass="my-container">
           <block name="my.block"
                  class="Magento\Framework\View\Element\Template"
                  template="Magento_Theme::my-template.phtml"/>
       </container>
   </body>
</page>
```

Default class

PROFESSIONAL SOLUTIONS PARTNER :fisheye

<referenceContainer/>.

- What attributes can reference containers have?

<referenceContainer/>.

What attributes can reference containers have?

- name
- remove
- display

PROFESSIONAL SOLUTIONS PARTNER :fisheye

<some_handle>.xml

```
<?xml version="1.0"?>
<page xmlns:xsi="..." xsi:noNamespaceSchemaLocation="...">
   <body>
       <referenceContainer name="footer">
           <block name="my.block"
                  class="Magento\Framework\View\Element\Template"
                  template="Magento_Theme::my-template.phtml"/>
       </referenceContainer>
   </body>
</page>
```

PROFESSIONAL SOLUTIONS PARTNER :fisheye

<some_handle>.xml

```xml
<?xml version="1.0"?>
<page xmlns:xsi="..." xsi:noNamespaceSchemaLocation="...">
    <body>
        <referenceContainer name="footer" remove="true"/>
    </body>
</page>
```

PROFESSIONAL SOLUTIONS PARTNER :fisheye

<some_handle>.xml

```
<?xml version="1.0"?>
<page xmlns:xsi="..." xsi:noNamespaceSchemaLocation="...">
   <body>
       <referenceContainer name="footer" display="false"/>
   </body>
</page>
```

PROFESSIONAL SOLUTIONS PARTNER :fisheye

<referenceBlock/>.

- **What attributes can reference blocks have?**

<referenceBlock/>.

What attributes can reference blocks have?

- name
- template
- remove
- display

<some_handle>.xml

```
<?xml version="1.0"?>
<page xmlns:xsi="..." xsi:noNamespaceSchemaLocation="...">
    <body>
        <referenceBlock name="my.block"
                        template="Magento_Theme::some/other.phtml"/>
    </body>
</page>
```

PROFESSIONAL SOLUTIONS PARTNER :fisheye

<some_handle>.xml

```
<?xml version="1.0"?>
<page xmlns:xsi="..." xsi:noNamespaceSchemaLocation="...">
   <body>
       <referenceBlock name="my.block" remove="true"/>
   </body>
</page>
```

<some_handle>.xml

```xml
<?xml version="1.0"?>
<page xmlns:xsi="..." xsi:noNamespaceSchemaLocation="...">
   <body>
       <referenceBlock name="my.block" display="false"/>
   </body>
</page>
```

before and after.

- **What are “before” and “after” for?**
- How can elements be placed first / last within a parent element?

before and after.

What are “before” and “after” for?

- To allow elements (blocks / containers) to be placed before or after others

<some_handle>.xml

```
<?xml version="1.0"?>
<page xmlns:xsi="..." xsi:noNamespaceSchemaLocation="...">
   <body>
       <referenceContainer name="my.container">
           <block name="my.block"
                  class="Magento\Framework\View\Element\Template"
                  template="Magento_Theme::my-template.phtml"
                  before="some.other.block"/>
       </container>
   </body>
</page>
```

PROFESSIONAL
SOLUTIONS PARTNER
:fisheye

before and after.

- What are “before” and “after” for?
- **How can elements be placed first / last within a parent element?**

before and after.

How can elements be placed first / last within a parent element?

- Using **before="-"** / **after="-"**

<some_handle>.xml

```
<?xml version="1.0"?>
<page xmlns:xsi="..." xsi:noNamespaceSchemaLocation="...">
   <body>
       <referenceContainer name="my.container">
           <block name="my.block"
                  class="Magento\Framework\View\Element\Template"
                  template="Magento_Theme::my-template.phtml"
                  after="-"/>
       </container>
   </body>
</page>
```

<move/>.

- What attributes can move elements have?

PROFESSIONAL SOLUTIONS PARTNER :fisheye

<move/>.

What attributes can move elements have?

- element
- destination
- before/after

PROFESSIONAL SOLUTIONS PARTNER :fisheye

<some_handle>.xml

```
<?xml version="1.0"?>
<page xmlns:xsi="..." xsi:noNamespaceSchemaLocation="...">
    <body>
        <move element="my.block" destination="footer" before="-"/>
    </body>
</page>
```

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Templates

Templates.

- **What components can templates be found in?**
- What is the file structure of template files?

Templates.

What components can templates be found in?

- Modules
- Themes

Templates.

- What components can templates be found in?
- **What is the file structure of template files?**

Templates.

What is the file structure of template files?

- **Modules**
  - view/**<area>**/templates/some/path/**file.phtml**
- **Themes**
  - **<Vendor_Module>**/templates/some/path/**file.phtml**

Block classes.

- **What is the default block class for use with templates?**
- What block class do all others extend from?

Block classes.

What is the default block class for use with templates?

- Magento\Framework\View\Element\Template

Block classes.

- What is the default block class for use with templates?
- **What block class do all others extend from?**

Block classes.

What block class do all others extend from?

- Magento\Framework\View\Element\AbstractBlock

Block data.

- **How do you access the block from a template?**
- How do you call block methods or get data?

Block data.

How do you access a block from a template?

- Using the **$block** variable
    - **$this** should not be used

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Block data.

- How do you access a block from a template?
- **How do you call block methods or get data?**

<some-file>.phtml

Always use type hint the $block variable to enable IDE autocompletion

```
<?php
/** @var Magento\Framework\View\Element\Template $block */
?>

<?= $block->callSomeMethod() ?>

<?= $block->getData('title') ?>
```

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Block methods.

- **What are common block methods?**
- How do child blocks work?
- How do you output a group of child blocks?

<some-file>.phtml

```
$block->getUrl('checkout/cart/index')

$block->getBaseUrl()

$block->getMediaDirectory()

$block->getChildBlock('block.name')

$block->getChildHtml('block.name')

$block->getChildChildHtml('block.name')

$block->getBlockHtml('block.name')

$block->escapeHtml()

$block->escapeUrl()
```

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Block methods.

- What are common block methods?
- **How do child blocks work?**
- How do you output a group of child blocks?

<some_handle>.xml

name (or as) used to reference child in parent template

```
...
<referenceContainer name="main">
    <block name="my.block"
           class="Magento\Framework\View\Element\Template"
           template="Magento_Theme::my-template.phtml">

        <block name="my.child.block"
               as="childBlock"
               class="Magento\Framework\View\Element\Template"
               template="Magento_Theme::my-child-template.phtml">
        </block>
    </block>
</referenceContainer>
...
```

PROFESSIONAL SOLUTIONS PARTNER :fisheye

my-template.phtml

Child block name in layout

```
<?php
/** @var Magento\Framework\View\Element\Template $block */
?>

<?= $block->getChildHtml('my.child.block') ?>
```

PROFESSIONAL SOLUTIONS PARTNER :fisheye

<some-file>.phtml

Can also use as

```
<?php
/** @var Magento\Framework\View\Element\Template $block */
?>

<?= $block->getChildHtml('childBlock') ?>
```

PROFESSIONAL SOLUTIONS PARTNER :fisheye

<some-file>.phtml

If you have multiple child blocks and want to output them all, don't pass a block name as a parameter

```
<?php
/** @var Magento\Framework\View\Element\Template $block */
?>

<?= $block->getChildHtml() ?>
```

Block methods.

- What are common block methods?
- How do child blocks work?
- **How do you output a group of child blocks?**

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Block methods.

How do you output a group of child blocks?

- Using the **group** attribute in layout and the **getGroupChildNames** method in the parent template
- Requires custom implementation
- Product detail page tabs are best example in Magento core codebase

<some_handle>.xml

```
...
<referenceContainer name="main">
   <block name="my.block"
          template="Magento_Theme::group.phtml">
       <block name="child.block.one"
              template="Magento_Theme::child-template-one.phtml"
              group="my.group.name"/>
       <block name="child.block.two"
              template="Magento_Theme::child-template-one.phtml"
              group="my.group.name"/>
       </block>
   </block>
</referenceContainer>
...
```

PROFESSIONAL SOLUTIONS PARTNER :fisheye

group.phtml

```
<?php
/** @var Magento\Framework\View\Element\Template $block */
?>

<?= $groupChildren = $block->getGroupChildNames('my.group.name') ?>

<?php foreach ($groupChildren as $groupChild) : ?>

   <!-- Do something -->

<?php endforeach ?>
```

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Best practice.

- **What should / shouldn’t templates contain?**
- What is <?=?
- What is the recommended format for PHP control structures in templates? (ifs, loops etc.)

Best practice.

What should / shouldn’t templates contain?

- **Should:**
  - View / presentation markup
  - Data and basic conditions / constructs (if / foreach)
  - Translated strings
- **Shouldn’t:**
  - Business logic
  - Excessive file contents

Best practice.

- What should / shouldn’t templates contain?
- **What is <?=?**
- What is the recommended format for PHP control structures in templates? (ifs, loops etc.)

Best practice.

What is <?=?

- PHP short echo tag
    - Equivalent to <?php echo
- Magento uses <?=

Best practice.

- What should / shouldn’t templates contain?
- What is <?=?
- **What is the recommended format for PHP control structures in templates? (ifs, loops etc.)**

Best practice.

What is the recommended format for PHP control structures in templates? (ifs, loops etc.)

- Standard PHP control structures should **NOT** be used
  - e.g. **ifs** or **loops** with curly braces
- Use alternative control structures at **ALL TIMES**
  - **http://php.net/manual/en/control-structures.alternative-syntax.php**

<some-file>.phtml

```
<?php
   if ($something === true) {

       echo '<div>' . $something . '</div>';

   } else {

       echo '<div>' . __('Nothing') . '</div>';

   }
?>
```

Bad - hard to read,
HTML inside PHP

PROFESSIONAL SOLUTIONS PARTNER :fisheye

<some-file>.phtml

```
<?php if ($something === true): ?>

    <div><?= $something ?></div>

<?php else: ?>

    <div><?= __('Nothing') ?></div>

<?php endif ?>
```

Good - readable and HTML separated from PHP

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Arguments

Arguments.

- **What are arguments?**
- What data types are allowed?
- How do you utilise them?
- Why would you use them?

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Arguments.

What are arguments?

- They allow passing of data via layout XML to a block
- This data can then be accessed from the template file

Arguments.

- What are arguments?
- **What data types are allowed?**
- How do you utilise them?
- Why would you use them?

Arguments.

What data types are allowed?

- strings
- numbers
- booleans
- arrays
- objects and more...

Arguments.

- What are arguments?
- What data types are allowed?
- **How do you utilise them?**
- Why would you use them?

<some_handle>.xml

```
<block class="Magento\Framework\View\Element\Template"
       name="my.block"
       template="Magento_Theme::my-template.phtml">
    <arguments>
        <argument name="title"
                  translate="true"
                  xsi:type="string">My Title</argument>
    </arguments>
</block>
```

my-template.phtml

Argument name

```
<?php
/** @var Magento\Framework\View\Element\Template $block */
?>

<?= $block->getData('title') ?>
```

PROFESSIONAL SOLUTIONS PARTNER :fisheye

<some_handle>.xml

```
<block class="Magento\Framework\View\Element\Template"
       name="my.block"
       template="Magento_Theme::my-template.phtml">
    <arguments>
        <argument name="my_array" xsi:type="array">
            <item name="array_key_one" xsi:type="string">
                Some value
            </item>
            <item name="array_key_two" xsi:type="string">
                Some other value
            </item>
        </argument>
    </arguments>
</block>
```

Array example using ‘<item>’

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Arguments.

- What are arguments?
- What data types are allowed?
- How do you utilise them?
- **Why would you use them?**

Arguments.

Why would you use them?

- To remove (decouple) content from templates
- To improve reusability of template
- To improve maintainability / upgradeability

Overriding and replacing template files

Overriding template files.

- **How can template files be overridden in a theme?**
- How does that impact upgradeability?
- How can you determine which template is output on the frontend?

Overriding template files.

How can template files be overridden in a theme?

- By placing the relevant file in the **same location** in the child theme
- Matching the template location in the module after view/**<area>** in the theme under relevant **<Vendor_Module>** directory

*See workshop 1 slides!

Overriding template files.

- How can template files be overridden in a theme?
- **How does that impact upgradeability?**
- How can you determine which template is output on the frontend?

Overriding template files.

How does that impact upgradeability?

- Likely very minimal chance of breaking upgradeability
- Potential concerns
  - Security vulnerabilities patched not in overridden file
  - New features / fixes added not in overridden file

Overriding template files.

- How can template files be overridden in a theme?
- How does that impact upgradeability?
- **How can you determine which template is output on the frontend?**

Overriding template files.

How can you determine which template is output on the frontend?

- **Using template hints**
  - **CLI**
    - **bin/magento dev:template-hints:enable**
  - **Admin**
    - STORES > Settings > Configuration
    - Advanced > Developer
    - Debug > Enabled Template Path Hints for Storefront > Yes

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Replacing template files.

- **What are the 2 methods you can use to change the template for a block?**
- What problems can arise from this method?
- What precautions can be taken?

Replacing template files.

What are the 2 methods you can use to change the template for a block?

- Using the **template="..."** attribute
- Using an **<argument/>**

<some_handle>.xml

```xml
<?xml version="1.0"?>
<page xmlns:xsi="..." xsi:noNamespaceSchemaLocation="...">
   <body>
       <referenceBlock name="page.main.title"
                       template="Vendor_Module::path/to/template.phtml"/>
   </body>
</page>
```

<some_handle>.xml

```xml
<?xml version="1.0"?>
<page xmlns:xsi="..." xsi:noNamespaceSchemaLocation="...">
    <body>
        <referenceBlock name="page.main.title">
            <arguments>
                <argument name="template" xsi:type="string">
                    Vendor_Module::path/to/template.phtml
                </argument>
            </arguments>
        </referenceBlock>
    </body>
</page>
```

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Replacing template files.

- What are the 2 methods you can use to change the template for a block?
- **What problems can arise from this method?**
- What precautions can be taken?

Replacing template files.

What problems can arise from this method?

- The original template file fallback path is broken
- If the original template file was overridden in a theme further down the chain, this will no longer be picked up

Replacing template files.

- What are the 2 methods you can use to change the template for a block?
- What problems can arise from this method?
- **What precautions can be taken?**

Replacing template files.

What precautions can be taken?

- Use a plugin to only override when needed (backend dev)
- Avoid overriding templates
- Always check overridden files when upgrading Magento

Escaping content

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Escaping content.

- **What is escaping and why should it be used?**
- What are the escaping types?
- How are they used?

Escaping content.

What is escaping and why should it be used?

- Security
- To ensure all output is sanitised
  - i.e. doesn’t contain malicious content

Escaping content.

- What is escaping and why should it be used?
- **What are the escaping types?**
- How are they used?

Escaping content.

What are the escaping types?

- HTML
- HTML attributes
- Javascript
- URLs

Escaping content.

- What is escaping and why should it be used?
- What are the escaping types?
- **How are they used?**

<some-file>.phtml

```php
$block->escapeHtml('value', $allowedTags);

$block->escapeHtmlAttr('value', $escapeSingleQuote);

$block->escapeJs('value');

$block->escapeUrl($url);
```

Layout merging, overrides and processing

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Layout merging / overriding.

- **What is layout merging?**
- What is layout overriding?
- How can you override layout?

Layout merging / overriding.

What is layout merging?

- Collating layout instructions between files
- For files of the same handle
- e.g. across multiple themes / modules

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Layout merging / overriding.

- What is layout merging?
- **What is layout overriding?**
- How can you override layout?

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Layout merging / overriding.

What is layout overriding?

- Overriding a specified layout file
- Original file is removed from merging process
- Only used in niche use cases

Layout merging / overriding.

- What is layout merging?
- What is layout overriding?
- **How can you override layout?**

Layout merging / overriding.

How can you override layout?

- See DevDocs:
  - https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/layouts/layout-override.html

Layout processing.

- **How is layout XML combined / processed?**
- What determines the order?
- What are common issues?

Layout processing.

How is layout XML combined / processed?

- All layout XML files are combined into one final document
- Based on the current pages layout handle(s)
- And any overrides

Layout processing.

- How is layout XML combined / processed?
- **What determines the order?**
- What are common issues?

Layout processing.

What determines the order?

- Modules load order - **app/etc/config.php**
- Modules areas (**base** then **frontend** or **adminhtml**)
- Theme inheritance e.g. child themes takes precedence
- **default.xml** is always loaded first before other handles

Layout processing.

- How is layout XML combined / processed?
- What determines the order?
- **What are common issues?**

Layout processing.

What are common issues?

- **Accidental overrides**
  - Where you specify **<block name="...">** which happens to be the same name as another block
- **Wishful overrides**
  - Where you want to override another block, but you specify a different name and the block is injected twice

Wrap up / resources

PROFESSIONAL SOLUTIONS PARTNER

:fisheye

Useful links.

Magento DevDocs

- Layout instructions
- Layout file types
- Common layout customisation tasks
- Template basic concepts
- Template customisation walkthrough

Code challenge 1.

- Add a container to the footer on all pages
- Add a block to the container that outputs a message
- Move the message on category pages to the header

Code challenge 2.

- Add a child block to one of your previously created blocks and output further content from another template
- Use arguments to pass and output data to the new child block
- Escape the output in your templates

Code challenge 3.

- Override a layout file:
  - From a theme
  - From a module

Bonus challenge.

- How can you add a block to multiple pages (but not all) without duplicating the layout instructions?

Credit / Resources / Thanks.

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