Level up your layout

John Hughes

Head of Magento - Fisheye Media Ltd.

PROFESSIONAL SOLUTIONS PARTNER

:fisheye

About me.

- Head of Magento at Fisheye Media Ltd.
- Working with Magento for over 6 years
- Slightly obsessed with orange things...
- Passionate about helping developers learn Magento 2

MAGENTO 2 CERTIFIED Solution Specialist

MAGENTO 2 CERTIFIED Professional Developer

MAGENTO 2 CERTIFIED Associate Developer

MAGENTO 2 CERTIFIED Front End Developer

PROFESSIONAL SOLUTIONS PARTNER :fisheye

I am the barrier to your lunch!

Let’s get cracking then.

- Fast paced
- Don’t worry about taking in each slide
- Link to slides at the end

Introduction

Layout XML!!!

This talk is not about...

Magento 2 layout XML itself, e.g.

- Using blocks / containers
- Moving / removing elements
- Syntax

Or PWA!

#SorryNotSorry

So what is it actually about?

- Reusable components in Magento 2
  - Configuration over code
  - **DRY** (**D**on't **R**epeat **Y**ourself)
- Decoupling / separation of concerns
  - **S**OLID (**S**ingle Responsibility)

Layout XML is just the tool.

Overview.

- Arguments
- View models
- ~~Page Layouts~~

Arguments

What is an argument?

- Allows passing of data via layout XML to block / template
- Multiple data types
  - Scalar (string, number, boolean, array etc.)
  - Objects
  - Constants
  - And more...

layout/*.xml

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

Argument ‘name’

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

PROFESSIONAL SOLUTIONS PARTNER :fisheye

layout/*.xml

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

Can also use magic getter, if you must...

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

PROFESSIONAL SOLUTIONS PARTNER :fisheye

layout/*.xml

```
<block class="Magento\Framework\View\Element\Template"
       name="my.block"
       template="My_Module::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

Scenario.

- Display ‘**delivery and returns**’ information on the **product detail page**
- Content should be displayed in a **modal**, triggered by clicking / tapping on a **link**
- **Functionality to be used across multiple projects**

Basic module.

- **Fisheye_DeliveryReturns**
  - Layout XML
  - Block class / template file
  - jQuery modal widget

Modal widget

Edit this page on GitHub
Give us feedback

Overview

The Magento modal widget implements a secondary window that opens on top of the main window. It contains the overlay and modal content. The modal widget configuration enables the following:

- Configuring as popup or slide
- Controlling stack of modal widgets
- Setting buttons for action bar

The modal widget source is `<Magento_Ui_module_dir>/view/base/web/js/modal/modal.js`.

The widget uses the following templates:

- `<Magento_Ui_module_dir>/view/base/web/templates/modal/modal-popup.html` popup type template.
- `<Magento_Ui_module_dir>/view/base/web/templates/modal/modal-slide.html` slide type template.

The design patterns for the modal pop-up windows in the Admin are described in the Magento Admin Pattern Library, the Slide-out Panels, Modal Windows, and Overlays topic.

Initialize the modal widget

To initialize the widget in your script, use the following general notation:

```
$('#modal_content').modal({
    &lt;option1&gt;: &lt;value1&gt;,
    &lt;option2&gt;: &lt;value2&gt;,
    ...
});
```

For details about how to initialize the widget in a `.phtml` template, refer to the JavaScript initialization topic.

https://devdocs.magento.com/guides/v2.2/javascript-dev-guide/widgets/widget_modal.html

catalog_product_view.xml

```
<referenceContainer name="product.info.main">
    <block class="Fisheye\DeliveryReturns\Block\Modal"
           name="delivery.returns.info"
           template="Fisheye_DeliveryReturns::modal.phtml"
           after="product.info.extrahint"/>
</referenceContainer>
```

After product ‘extra’ actions (wishlist etc.)

PROFESSIONAL SOLUTIONS PARTNER :fisheye

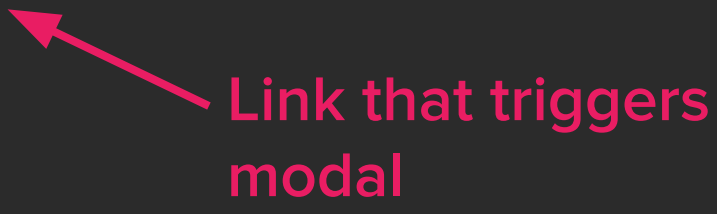
modal.phtml

```
<div data-mage-init='{"fisheye/delivery-returns-modal": {
    "title": "<?= __('Delivery & Returns Information') ?>",
    "modalClass": "delivery-returns-content"
  }}'>
    <a href="#" data-role="trigger" class="delivery-returns-link">
        <?= __('Delivery Info') ?>
    </a>
    <div data-role="target">
        <?= $block->getContent() ?>
    </div>
</div>
```

RequireJS declaration
+ jQuery modal widget
config

modal.phtml

```
<div data-mage-init='{"fisheye/delivery-returns-modal": {
     "title": "<?= __('Delivery & Returns Information') ?>",
     "modalClass": "delivery-returns-content"
  }}'>
    <a href="#" data-role="trigger" class="delivery-returns-link">
        <?= __('Delivery Info') ?>
    </a>
    <div data-role="target">
        <?= $block->getContent() ?>
    </div>
</div>
```



modal.phtml

```
<div data-mage-init='{"fisheye/delivery-returns-modal": {
     "title": "<?= __('Delivery & Returns Information') ?>",
     "modalClass": "delivery-returns-content"
  }}'>
    <a href="#" data-role="trigger" class="delivery-returns-link">
        <?= __('Delivery Info') ?>
    </a>
    <div data-role="target">
        <?= $block->getContent() ?>
    </div>
</div>
```

Modal content

PROFESSIONAL SOLUTIONS PARTNER :fisheye

modal.js

```
initModal: function(config, element) {

    var target = $(element).find('[data-role="target"]'),
        trigger = $(element).find('[data-role="trigger"]')

    target.modal({
        autoOpen: false,
        clickableOverlay: true,
        title: config.title,
        modalClass: config.modalClass
    });

    $(trigger).click(function(event) {
        event.preventDefault();
        target.modal('openModal');
    });
}
```

Modal initialisation with config from template

modal.js

```
initModal: function(config, element) {

    var target = $(element).find('[data-role="target"]'),
        trigger = $(element).find('[data-role="trigger"]')

    target.modal({
        autoOpen: false,
        clickableOverlay: true,
        title: config.title,
        modalClass: config.modalClass
    });

    $(trigger).click(function(event) {
        event.preventDefault();
        target.modal('openModal');
    });
}
```

Trigger modal on link click / tap

Default welcome msg! Sign In or Create an Account

LUMA

Search entire store here...

What's New Women Men Gear Training Sale

Home > Joust Duffle Bag

Image

Joust Duffle Bag

2 Reviews Some others string

£34.00

IN STOCK

SKU#: 24-MB01

Qty

1

Add to Cart

ADD TO WISH LIST ADD TO COMPARE EMAIL

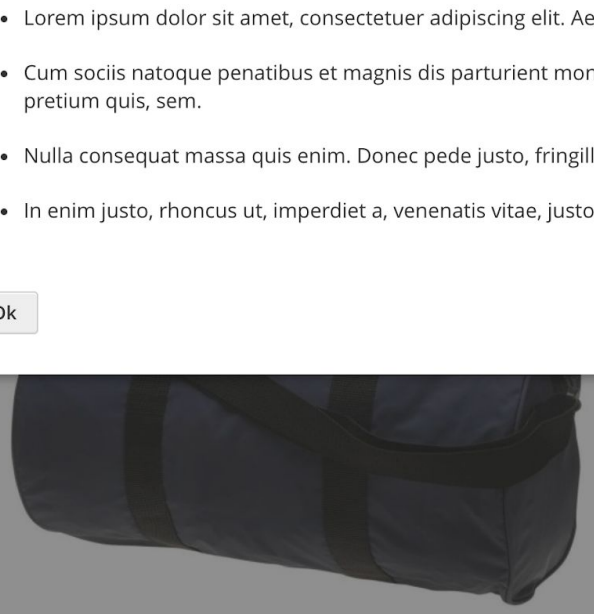
Delivery Info

Details More Information Reviews (2)

Delivery & Returns Information

- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa.
- Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem.
- Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu.
- In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt.

Ok



Image

My work here is done.

Right...?


So what's the problem?
PROFESSIONAL
SOLUTIONS PARTNER
:fisheye

template.

/ˈtɛmpleɪt,ˈtɛmplət/

NOUN

“Something that serves as a model for others to copy.”

Computing:

A preset format for a document or file.

Source: Oxford Dictionary

It’s not really a template.

- Most code is ‘boilerplate’ (template)
- But also ‘implementation’ code
- ‘Implementation’ also in the template
- **They are tightly coupled to one another**

Template issues.

- Template needs to be overridden (duplicated) to
  - Change link or title text*
  - Add or modify CSS classes
  - Add or modify modal configuration

*translations could be used, but not ideal / best approach (e.g. generic phrase)

Example use case.

- Template is overridden (duplicated) in multiple projects
- Later we realise we should've escaped the text output
- Now we have to manually update multiple copies of the same template file **:-(**


So what
can we do?
PROFESSIONAL
SOLUTIONS PARTNER
:fisheye

Arguments to the rescue.

modal.phtml

```
<div data-mage-init='{"fisheye/delivery-returns-modal": {
     "title": "<?= __('Delivery & Returns Information') ?>",
     "modalClass": "delivery-returns-content"
  }}'>
    <a href="#" data-role="trigger" class="delivery-returns-link">
        <?= __('Delivery Info') ?>
    </a>
    <div data-role="target">
        <?= $block->getContent() ?>
    </div>
</div>
```

Let's remove specific text content

PROFESSIONAL SOLUTIONS PARTNER :fisheye

modal.phtml

```
<div data-mage-init='{"fisheye/delivery-returns-modal": {
     "title": "<?= $block->getData('modalTitle') ?>",
     "modalClass": "<?= $block->getData('modalClass') ?>"
  }}'>
    <a ... class="<?= $block->getData('linkClass') ?>">
        <?= $block->getData('linkText') ?>
    </a>
    <div data-role="target">
        <?= $block->getContent() ?>
    </div>
</div>
```

And replace with arguments

PROFESSIONAL SOLUTIONS PARTNER :fisheye

catalog_product_view.xml

```
<block … name="delivery.returns.info" … >
    <arguments>
        <argument name="linkClass" xsi:type="string">
            delivery-returns-link
        </argument>
        <argument name="linkText" translate="true" xsi:type="string">
            Delivery Info
        </argument>
        <argument name="modalClass" xsi:type="string">
            delivery-returns-content
        </argument>
        <argument name="modalTitle" translate="true" xsi:type="string">
            Delivery &amp;amp; Returns Information
        </argument>
    </arguments>
</block>
```

catalog_product_view.xml

```
<referenceBlock name="delivery.returns.info">
    <arguments>
        <argument name="linkText" translate="true" xsi:type="string">
            Shipping Details
        </argument>
        <argument name="modalClass" xsi:type="string">
            delivery-returns-content some-other-custom-class
        </argument>
    </arguments>
</referenceBlock>
```

Now making changes requires no modification / override of our template

Much better!

modal.phtml

```
<div data-mage-init='{"fisheye/delivery-returns-modal": {
    "title": "<?= $block->getData('modalTitle') ?>",
    "modalClass": "<?= $block->getData('modalClass') ?>",
    "type": "<?= $block->getData('modalType') ?>"
  }}'>
    <a ... class="<?= $block->getData('linkClass') ?>">
        <?= $block->getData('linkText') ?>
    </a>
    <div data-role="target">
        <?= $block->getContent() ?>
    </div>
</div>
```

We can even add more modal widget config

catalog_product_view.xml

```
<block … name="delivery.returns.info" … >
    <arguments>
        ...
        <argument name="modalType" xsi:type="string">
            slide
        </argument>
        ...
    </arguments>
</block>
```

So long as we pass in the values via arguments

modal.js

```
initModal: function(config, element) {
    ...
    target.modal({
        ...
        type: config.type
    });
    ...
}
```

And map them
in the widget

More issues...

PROFESSIONAL SOLUTIONS PARTNER

:fisheye

Javascript issues.

- Modal JS component locked to **Fisheye_DeliveryReturns**
  - If we want a similar modal we either
    - Depend on this module
    - Duplicate code and add another JS file / request

Hint: both options are bad!

modal.phtml

```
<div data-mage-init='{"fisheye/delivery-returns-modal": {
      "title": "<?= $block->getData('modalTitle') ?>",
      "modalClass": "<?= $block->getData('modalClass') ?>",
      "type": "<?= $block->getData('modalType') ?>"
   }}'>
    <a ... class="<?= $block->getData('linkClass') ?>">
        <?= $block->getData('linkText') ?>
    </a>
    <div data-role="target">
        <?= $block->getContent() ?>
    </div>
</div>
```

The JS module is too specific

PROFESSIONAL SOLUTIONS PARTNER :fisheye

modal.phtml

```
<div data-mage-init='{"fisheye/modal": {
      "title": "<?= $block->getData('modalTitle') ?>",
      "modalClass": "<?= $block->getData('modalClass') ?>",
      "type": "<?= $block->getData('modalType') ?>"
  }}'>
    <a ... class="<?= $block->getData('linkClass') ?>">
        <?= $block->getData('linkText') ?>
    </a>
    <div data-role="target">
        <?= $block->getContent() ?>
    </div>
</div>
```

So let's abstract the modal JS module

catalog_product_view.xml

```
<referenceContainer name="product.info.main">
    <block class="Fisheye\DeliveryReturns\Block\Modal"
           name="delivery.returns.info"
           template="Fisheye_DeliveryReturns::modal.phtml"
           after="product.info.extrahint">
        <arguments>
            <argument ...>
                ...
            </argument>
        </arguments>
    </block>
</referenceContainer>
```

The module name no longer makes much sense

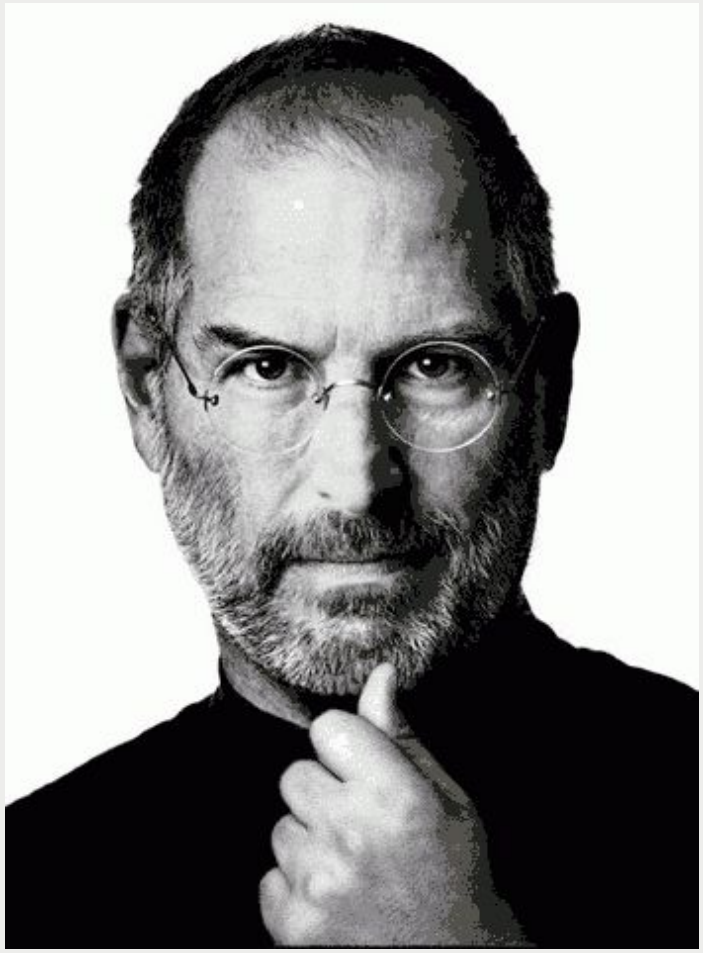
PROFESSIONAL SOLUTIONS PARTNER :fisheye

catalog_product_view.xml

```
<referenceContainer name="product.info.main">
    <block class="Fisheye\Modal\Block\Modal"
           name="delivery.returns.info"
           template="Fisheye_Modal::modal.phtml"
           after="product.info.extrahint">
        <arguments>
            <argument ...>
                ...
            </argument>
        </arguments>
    </block>
</referenceContainer>
```

This layout should now belong outside the ‘Fisheye_Modal’ module

One more thing.



PROFESSIONAL SOLUTIONS PARTNER :fisheye

modal.phtml

```
<div data-mage-init='{"fisheye/modal": { ... }}'>
    <a ... class="<?= $block->getData('linkClass') ?>">
        <?= $block->getData('linkText') ?>
    </a>
    <div data-role="target">
        <?= $block->getContent() ?>
    </div>
</div>
```

The content is still coupled to block / template

modal.phtml

```
<div data-mage-init='{"fisheye/modal": { ... }}'>
    <a ... class="<?= $block->getData('linkClass') ?>">
        <?= $block->getData('linkText') ?>
    </a>
    <div data-role="target">
        <?= $block->getChildHtml() ?>
    </div>
</div>
```

But the content is of no concern to the modal, let's decouple it

catalog_product_view.xml

```
<block ... name="delivery.returns.info" ... >
    <arguments>...</arguments>
    <block class="Magento\Cms\Block\Block"
           name="delivery.returns.info.cms">
        <arguments>
            <argument name="block_id" xsi:type="string">
                delivery_returns_info
            </argument>
        </arguments>
    </block>
</block>
```

Now we can add any content via a child block e.g. a CMS (static) block

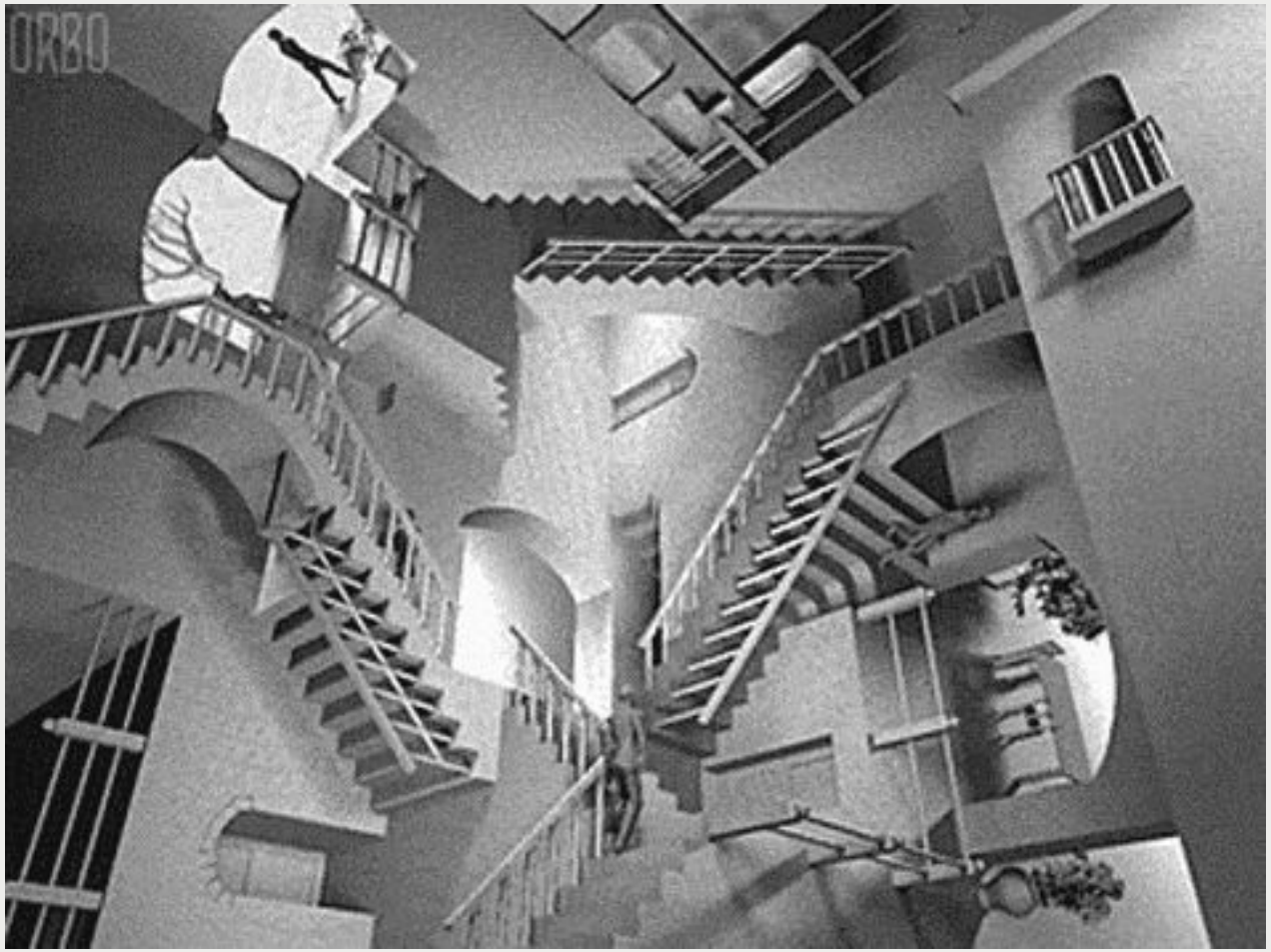
catalog_product_view.xml

```
<block ... name="delivery.returns.info" ... >
    <arguments>...</arguments>
</block>
```

```
<move element="some.custom.block"
      destination="delivery.returns.info"/>
```

Or even move one or more
existing blocks into our modal

The possibilities are endless!



The possibilities are endless!

- Now we can create unlimited modals with
  - any content
  - any config
  - in any location
- From the same component
- With minimal upgrade / maintenance concerns

It’s all great in theory...

- It turns out this module wasn’t just something I invented for this talk...
- Who’d have guessed **;-)**

STAR WARS DEATH STAR PET CAVE

☆☆☆☆☆ Write A Review

In Stock

£29.99 — 1 + BUY NOW

FREE UK DELIVERY ON ORDERS OVER £30

Add To Wishlist | Delivery & Returns

- ✓ Bring a touch of geekery and h[illegible]our into your home with this Star Wars-inspired design
- ✓ Super soft Velboa fabric construction [illegible]akes this a luxurious crash pad for any animal
- ✓ Machine washable with waterproof interior for easy cleaning

DELIVERY & RETURNS

UK DELIVERY

Standard - 3 to 5 Working Days - **£1**

Free for orders over £30

Next Day - 1 Working Day - **£5**

Our Next Day service guarantees next working day delivery for orders placed before 3pm, Monday to Friday, for UK mainland addresses. For full details, please visit our Delivery Page.

INTERNATIONAL DELIVERY

Our shipping rates to your country are simple. One flat rate, no matter the size of your order:

£4.99 - France, Germany, Ireland and The Netherlands.

£6.99 - Australia, Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Latvia, Lithuania, Luxembourg, Poland, Portugal, Slovakia, Slovenia, Sweden and The United States.

£9.99 - Canada, Iceland, Italy, New Zealand, Norway, South Korea, Spain and Switzerland.

Delivery to the European Union - 5 to 10 working days.

Delivery to the Rest of the World - 7 to 15 working days.

For full details, please visit our Delivery Page.

PRODUCT INFORMATION

This essential set is perfect for Joe's lean in 15 meals! Batch cook meals such as pasta or chilli in the large 26cm all rounder pan. The cast aluminum 28cm square grill pan is the perfect way to grill meat, fish or veggies. Both pans heat up rapidly and evenly to give great results and the easy-lift non-stick system ensures the food just slides out so they're easy to clean too.
Complimenting the grill pan, the 12" lock & serve elevated tongs are convenient and practical and feature an elevated design preventing the heads from laying on the work surface, keeping the area clean.

Easy Release Aluminium is designed to give all hob types a good work out including induction, and is dishwasher safe (excluding grill pan).

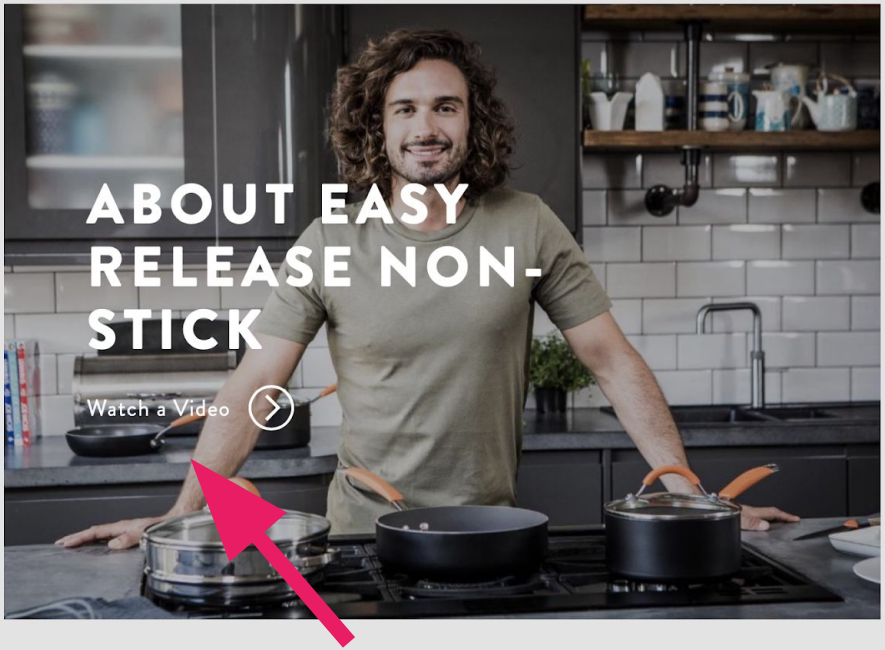
EASY LIFT NON-STICK SYSTEM

Long lasting non-stick interior not only heats rapidly and evenly, but also allows food to slide off making clean-up quick and easy

GRIPPY HANDLES

Colourful silicon handle adds a pop of colour and provides a comfortable slip free grip. Double-riveted for added durability and oven safe to 180°C / Gas Mark 4.

GLASS LID (ALL ROUNDER PAN ONLY)



Back to THE BODYCOACH

joe WICKS

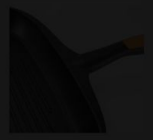
Cookware Ovenware Food Prep Hydration Storage

Close ×

PRODUCT IN

This essential set is perfe
in the large 26cm all rou
way to grill meat, fish or
and the easy-lift non-sti
too.
Complimenting the grill
and feature an elevated
the area clean.

Eas
wor
grill



EASY LIFT NON-STICK SYSTEM

Long lasting non-stick interior not only heats rapidly and evenly, but also allows food to slide off making clean-up quick and easy

GRIPPY HANDLES

Colourful silicon handle adds a pop of colour and provides a comfortable slip free grip. Double-riveted for added durability and oven safe to 180°C / Gas Mark 4.

GLASS LID (ALL ROUNDER PAN ONLY)

— DESCRIPTION

Vintage 80s Adidas track top. Zipped front with two hand pockets and ribbed cuffs and waist

+ DETAILS

+ CONDITION

INFO & CARE DELIVERY & RETURNS SIZE GUIDE

VINTAGE 80S ADIDAS TRACK TOP - M

£50.00

SIZE - M

ADD TO BAG

Free Delivery on Orders over £50

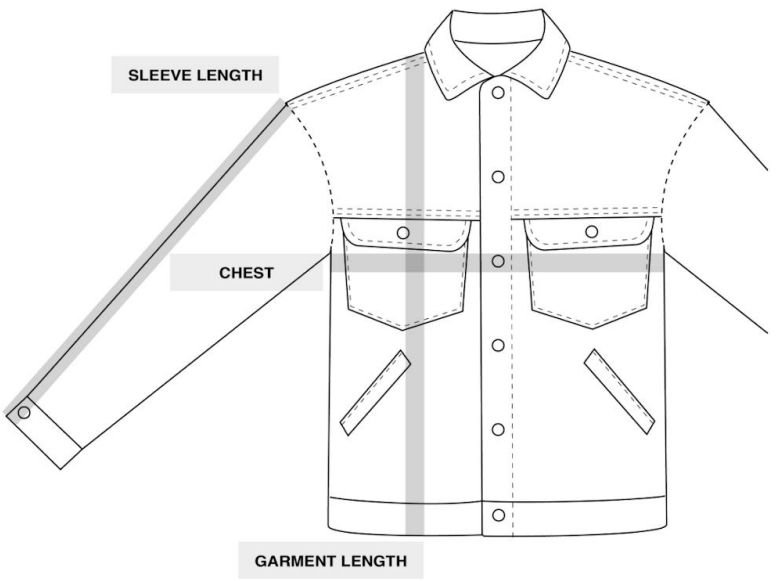
WISHLIST

SIZE GUIDE

Label sizes of vintage garments are different to today's standardised clothing sizes and are sometimes not labelled at all. Through general wear and washing garment sizes do alter so we measure all our clothes to give our customers the most accurate indication of size. To help you work out which clothing size is right for you we have included examples of how we measure our garments.

Size chart

INCHES CHEST	UK	APPROX WOMEN'S SIZE
34	XS	S
36	S	M
38-40	M	L
42-44	L	XL
46-48	XL	XXL
50-52	XXL	XXXL
54-56	XXXL	XXXXL



Licensed Lamborghini Aventador Roadster SV 12V Children's Ride On Car - Orange

SKU: BDM0913-ORANGE

☆☆☆☆☆ Write a review

Age Group **3+ Years**

~~RRP £319.95~~

£229.95 | **SAVE 28%**

PayPal CREDIT From only **£11.04 per month** (14.9% APR) ⓘ

Select Options & Buy >

Get It Delivered By **Monday**

Can We Help You? Click Here To Contact Us

Hassle Free Returns. Click Here For More Info

PayPal CREDIT 0% Finance | Next Day Delivery on orders before 3pm | 5* Reviews Reviews.io

OutdoorToys

Start typing to search...

Currency | Account | Basket

Ride On Toys | Trampolines | Shop By Brand | Clearance

Home > Licensed Lamborghini Aventador Roadster S

Close ×

Personalised Number Plate (Sent Attached to Outside of Box)

Enter up to 7 characters... + £4.95

+ Add Accessories & Assembly

Assembly Service +£99.95 - 0 +

Spare Battery +£29.95 - 0 +

Ride-On Car or Jeep Outdoor Weather Cover - 3 Sizes +£29.95 - 0 +

£229.95 Add to Basket >

...borghini Aventador ...12V Children's Ride On Car -

...NGE

...a review Age Group 3+ Years

SAVE 28%

...n only £11.04 per month (14.9% APR)

Select Options & Buy >

...d By Monday

Can We Help You? Click Here To Contact Us

Hassle Free Returns. Click Here For More Info

Argument considerations.

- Account for arguments not being passed in your code
  - Set default values
- Ensure relevant argument ***xsi:types*** are used
  - Add error output for developers if wrong type is used
- Document the arguments (and their values) that your component can / must utilise

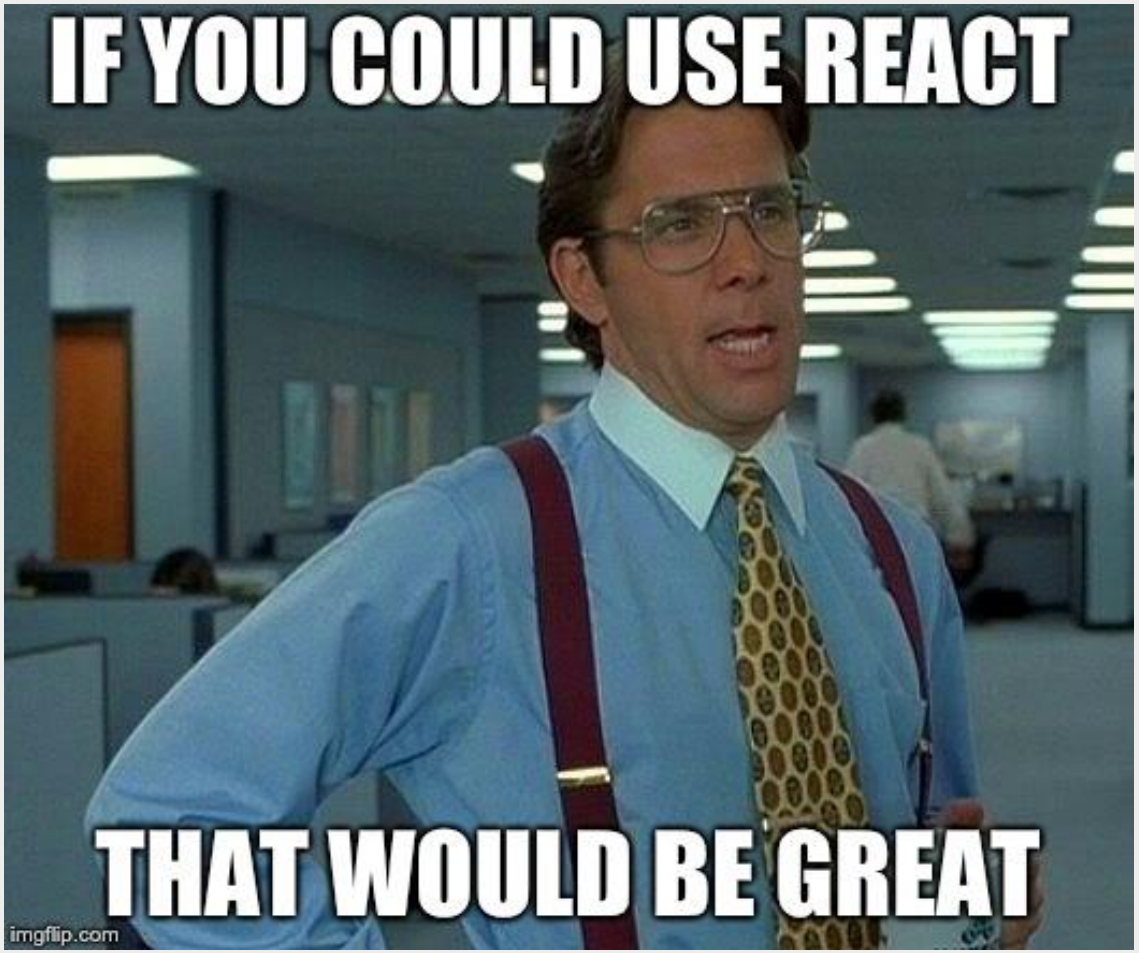
Reference.

https://github.com/fisheyehq/module-modal

Summary.

- Reuse code where you can as components
  - Provide flexibility by passing arguments
- Write less code! (especially boilerplate)
  - Make code maintainable / upgrade friendly
- **It's not about the tool, but the approach**

The framework doesn't matter



Look ma, React JS!

```
function Modal(props) {
    return (
        <div className="modal-wrapper">
            <a href="#" className={props.linkClass}>
                {props.linkText}
            </a>
            <div className={props.modalClass}>
                {props.modalContent}
            </div>
        </div>
    );
}
```

Look ma, React JS!

```
var modal = <Modal linkClass="delivery-returns-link"
                   linkText="Delivery Info"
                   modalClass="delivery-returns-content"
                   modalContent="Content goes here"/>;

ReactDOM.render(
    modal,
    document.getElementById('delivery-returns-container')
);
```

Reference.

- https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/layouts/xml-instructions.html
- https://devdocs.magento.com/guides/v2.2/frontend-dev-guide/layouts/xml-manage.html
- Plenty of examples in Magento core!

catalog_product_view.xml

```
<block class="Magento\Catalog\Block\Product\View\Description"
       name="product.info.sku"
       template="Magento_Catalog::product/view/attribute.phtml"
       after="product.info.type">
   <arguments>
       <argument name="at_call" xsi:type="string">
           getSku
       </argument>
       <argument name="at_code" xsi:type="string">
           sku
       </argument>
       <argument name="css_class" xsi:type="string">
           sku
       </argument>
       ...
   </arguments>
</block>
```

View models

Food is coming, stay strong!

What is a view model?

- A class you can use in addition to a block
- To decouple custom presentation logic

Why?

- Magento's default block classes* handle block / template related functions, such as:
  - ***toHtml()***
  - ***getChildHtml()***
  - ***escapeHtml()***

*e.g. ***Magento\Framework\View\Element\Template***

Magento\Framework\View\Element\AbstractBlock

Why?

- **S**ingle Responsibility Principle (from **S**OLID)
  - Tells us classes should only have one purpose
- So if blocks handle block / template specific code
  - We should use another class for our custom logic

Benefits.

- Separation of Magento and custom code
- No inheritance - only inject what you need
- Easier to write testable code
  - No parent dependencies to mock
- Easier to understand / maintain / upgrade

Limitations.

- Magento 2.2+ only*
- Still need to extend block for custom block caching (e.g. cache keys / tags etc.)

*__Note:__ workarounds are available


So how do
I use them?
PROFESSIONAL
SOLUTIONS PARTNER
:fisheye

Arguments of course!

Scenario.

- Display key ‘**store information**’ in the footer
- Phone number, opening hours etc.
- e.g. a quick reference ‘**store help**’ block

Basic module.

- **Fisheye_StoreHelp**
  - Layout XML
  - Template file
  - View model (no custom block class)

default.xml

```
<referenceContainer name="footer">
    <block class="Magento\Framework\View\Element\Template"
           name="store.help"
           template="Fisheye_StoreHelp::store-help.phtml"
           after="footer_links">
        <arguments>
            <argument name="storeInformation" xsi:type="object">
                Fisheye\StoreHelp\ViewModel\StoreInformation
            </argument>
        </arguments>
    </block>
</referenceContainer>
```

‘xsi:type’ must be ‘object’
and ‘value’ the class

PROFESSIONAL SOLUTIONS PARTNER :fisheye

default.xml

```
<referenceContainer name="footer">
    <block class="Magento\Framework\View\Element\Template"
           name="store.help"
           template="Fisheye_StoreHelp::store-help.phtml"
           after="footer_links">
        <arguments>
            <argument name="storeInformation" xsi:type="object">
                Fisheye\StoreHelp\ViewModel\StoreInformation
            </argument>
        </arguments>
    </block>
</referenceContainer>
```

‘class’ attribute is no longer needed as Template used as default

PROFESSIONAL SOLUTIONS PARTNER :fisheye

StoreInformation.php

```
class StoreInformation implements ArgumentInterface
{
    public function __construct(
        Information $storeInformation,
        StoreManagerInterface $storeManager
    ) {
        $this->storeInformation = $storeInformation;
        $this->storeManager = $storeManager;
    }
}
```

To get current store and formatted store data object

StoreInformation.php

```
class StoreInformation implements ArgumentInterface
{
    public function __construct(
        Information $storeInformation,
        StoreManagerInterface $storeManager
    ) {
        $this->storeInformation = $storeInformation;
        $this->storeManager = $storeManager;
    }
}
```

ArgumentInterface?

What is ArgumentInterface?

- Every view model **MUST** implement:

  Magento\Framework\View\Element\Block\ArgumentInterface

- This is an empty interface, but is used to ensure only classes intended to be view models are passed to blocks

- A conscious decision has to made on whether a class is to be used as a view model, not just any random class

StoreInformation.php

```
private function getStoreData(): DataObject
{
    return $this->storeInformation->getStoreInformationObject(
        $this->storeManager->getStore());
}

public function getStorePhone(): ?string
{
    return $this->getStoreData()->getData('phone');
}

public function getStoreHours(): ?string
{
    return $this->getStoreData()->getData('hours');
}
```

store-help.phtml

```php
<?php
/** @var Fisheye\StoreHelp\ViewModel\StoreInformation $storeInformation */
$storeInformation = $block->getData('storeInformation');
?>

<p class="store-phone">
    <?= $block->escapeHtml($storeInformation->getPhone()) ?>
</p>
<p class="store-hours">
    <?= $block->escapeHtml($storeInformation->getHours()) ?>
</p>
```

About us

With more than 230 stores spanning 43 states and growing, Luma is a nationally recognized active wear manufacturer and retailer. We're passionate about active lifestyles – and it goes way beyond apparel.

At Luma, wellness is a way of life. We don't believe age, gender or past actions define you, only your ambition and desire for wholeness... today.

We differentiate ourselves through a combination of unique designs and styles merged with unequaled standards of quality and authenticity. Our founders have deep roots in yoga and health communities and our selections serve amateur practitioners and professional athletes alike.

Contact Luma

Customer Service

Luma Privacy Policy

Shop Luma

Slightly more expanded example

Need Help?

Call +44 123 456 789
(Our lines are open Mon - Fri, 9am - 5pm)

or click here to email us!

Enter your email address Subscribe

Multiple view models.

Multiple view models

- You can pass in any number of view models via layout
- e.g. add config to allow optionally displaying store address
- Our **StoreInformation** responsibility is getting store info
- So display config should be handled by separate class again

default.xml

```
<referenceContainer name="footer">
    <block name="store.help" ...>
        <arguments>
            <argument name="store_information" xsi:type="object">
                Fisheye\StoreHelp\ViewModel\StoreInformation
            </argument>
            <argument name="config" xsi:type="object">
                Fisheye\StoreHelp\ViewModel\Config
            </argument>
        </arguments>
    </block>
</referenceContainer>
```

Let’s add an additional view model

Config.php

```
class Config implements ArgumentInterface
{
    const XML_PATH_DESIGN_STORE_HELP_DISPLAY_ADDRESS = 'design/store_help
/display_address';
    private $scopeConfig;

    public function __construct(ScopeConfigInterface $scopeConfig)
    {
        $this->scopeConfig = $scopeConfig;
    }

    public function displayStoreAddress(): bool
    {
        return $this->scopeConfig->isSetFlag(
            self::XML_PATH_DESIGN_STORE_HELP_DISPLAY_ADDRESS,
            ScopeInterface::SCOPE_STORE
        );
    }
}
```

Add our config check

PROFESSIONAL SOLUTIONS PARTNER :fisheye

StoreInformation.php

```
class StoreInformation implements ArgumentInterface
{
    public function getStoreAddress(): ?string
    {
        return $this->storeInformation->getFormattedAddress(
            $this->storeManager->getStore()
        );
    }
}
```

And retrieve the
formatted store address

PROFESSIONAL SOLUTIONS PARTNER :fisheye

store-help.phtml

```php
<?php
/** @var Fisheye\StoreHelp\ViewModel\StoreInformation $storeInformation */
$storeInformation = $block->getData('store_information');
/** @var Fisheye\StoreHelp\ViewModel\Config $config */
$config = $block->getData('config');
?>

...
<?php if ($config->displayStoreAddress()): ?>
    <div class="store-address">
        <?= $storeInformation->getStoreAddress() ?>
    </div>
<?php endif ?>
```

About us

With more than 230 stores spanning 43 states and growing, Luma is a nationally recognized active wear manufacturer and retailer. We're passionate about active lifestyles – and it goes way beyond apparel.

At Luma, wellness is a way of life. We don't believe age, gender or past actions define you, only your ambition and desire for wholeness... today.

We differentiate ourselves through a combination of unique designs and styles merged with unequaled standards of quality and authenticity. Our founders have deep roots in yoga and health communities and our selections serve amateur practitioners and professional athletes alike.

Contact Luma

Customer Service

Luma Privacy Policy

Shop Luma

Need Help?

Call +44 123 456 789
(Our lines are open Mon - Fr[illegible] - 5pm)
or click here to email us!

My Awesome Store
Awesome Street
Awesome City, AB1 2CD,
United Kingdom

Done.

PROFESSIONAL SOLUTIONS PARTNER :fisheye

Reference.

https://github.com/fisheyehq/module-store-help

View model considerations.

- Don’t name view model arguments *‘view_model’*
  - **Use a descriptive name**
- Using multiple view models per block **is good**
- Don’t add too many view models to one block
  - **If over 3 or 4 consider splitting up your block**

View model considerations.

- Consider how you'll handle passing data between block and view model
  - **Try to avoid passing in the block to a method**
- Create / re-use view models for common uses
  - **e.g. store data, product, category, config etc.**

Credit / further reading.

https://www.yireo.com/blog/2017-08-12-viewmodels-in-magento-2

https://firegento.com/blog/2017/12/07/better-blocks-magento-2-php-view-models/

https://www.integer-net.com/view-models-in-magento-1-and-2

Thanks Jisse / Vinai / Fabian!

Wrap up

PROFESSIONAL SOLUTIONS PARTNER

:fisheye

Think twice.

- Use what works for you and the scenario
- Don’t get ‘hung up’ on abstracting everything
- Apply some common sense
- Remember these were simple examples

Lunch!!!

Thank you!

Email: **johnh@fisheyehq.com**

Twitter: **@JohnHughes1984**

Slides: **https://tiny.cc/level-up-layout**