



# SELECTOR MADNESS: BUILDING “STABLE” SELECTORS

JOHN STENNITT

TEAM PANGOLINS

TWITTER: @JOHN\_STENNITT

# TODAY'S AGENDA

---

Browser Add-Ons

---

General Questions

---

Do's and Don'ts

---

CSS Overview

---

XPATH Overview

---

Replicating CSS in XPATH

---

Questions

---

Resources

# GENERAL WARNINGS

1

I'm not a "Master of Selectors".

- Alex Kolesnyk schools me in Xpath selectors often.

2

We are only scratching the surface of what you can do with selectors.

3

There are a lot of tables in this presentations. Sorry.



# BROWSER ADD-ONS

# BROWSER ADD-ONS



**Chrome**

- Live Preview of CSS/Less/Sass
- Xpath Helper



**FireFox**

- Live editor for CSS and LESS
- Try Xpath

# LIVE PREVIEW OF CSS/LESS/SASS

The screenshot shows the Magento Admin UI for editing a product. A modal window titled "css less sass or more" is open, displaying the CSS selector ".admin\_field-label label". The status bar at the top of the page indicates "Cache invalidated: Configuration, Page Cache. Please go to Cache Management and refresh cache types." A red arrow points to the status bar message.

**CSS Selector**: .admin\_field-label label

**Number of Elements that MATCH the current selector**: 14 matches

**Enable Product [website]**: Yes

**Attribute Set**: Default

**Product Name [store view]**: Simple product with category 1551498823

**SKU [global]**: simple\_product\_with\_category\_1551498823

**Elements that MATCH the current selector are outlined in RED\*\***

# LIVE PREVIEW OF CSS/LESS/SASS\*\*

The screenshot shows the Magento Admin Panel with the 'Enable Product [website]' switcher component highlighted. A red arrow points to the 'Yes' button of the switcher. The browser's developer tools Elements tab is open, showing the HTML structure of the switcher. The CSS selector `.admin__actions-switch label` is highlighted in the developer tools, and a yellow status bar message at the top right says 'Configuration, Page Cache. Please go to Cache Manager'.

```
<!-- ko ifnot: hasAddons() -->
<!-- ko template: elementTmpl -->
<div class="admin__actions-switch" data-role="switcher">
  <input type="checkbox" class="admin__actions-switch-checkbox" data-bind="attr: {id: uid, name: inputName}, hasFocus: focused, simpleChecked: checked" id="NG9EKIP" name="product[status]" value="1">
  <label class="admin__actions-switch-label" data-bind="attr: {for: uid}" for="NG9EKIP"></label>
</div>
```

The screenshot shows the same Magento Admin Panel component after the 'Yes' button has been clicked. A red arrow points to the now-unchecked 'Yes' button. The browser's developer tools Elements tab is open, showing the updated HTML structure where the checked attribute is removed from the input field. The CSS selector `.admin__actions-switch input` is highlighted in the developer tools, and the same yellow status bar message is present.

```
<!-- ko ifnot: hasAddons() -->
<!-- ko template: elementTmpl -->
<div class="admin__actions-switch" data-role="switcher">
  <input type="checkbox" class="admin__actions-switch-checkbox" data-bind="attr: {id: uid, name: inputName}, hasFocus: focused, simpleChecked: checked" id="NG9EKIP" name="product[status]" value="0">
  <label class="admin__actions-switch-label" data-bind="attr: {for: uid}" for="NG9EKIP"></label>
</div>
```

# XPATH HELPER

One or more of the Cache Types are invalidated: Configuration, Page Cache, Please go to Cache Management and refresh.

RESULTS (14)

Simple product with category 1551498823

QUERY  
`//*[contains(@class, "admin__field-label")]/label`

Enable Product

Attribute Set

Number of Elements that MATCH the current selector

← Back Add Attribute Save

**XPath Selector**

Store View: All Store Views ?

Elements that MATCH the current selector are outlined in YELLOW\*\*

Enable Product [website]  Yes

Attribute Set Default

Product Name \* Simple product with category 1551498823

Simple product with category 1551498823

NAVIGATION

\$ SALES

CATALOG

CUSTOMERS

MARKETING

CONTENT

# XPATH HELPER\*\*

QUERY  
//\*[contains(@class, "admin\_\_actions-switch")]/label

RESULTS (1)

Simple product with category 1551498823

Store View: All Store Views

← Back

Enable Product [website]   
`div.admin__actions-switch | 68.69x32`

Attribute Set Default

Elements Console Sources Network Performance Memory Application Security Audits AdBlock Adblock P...

```
<div class="admin__field-control" data-bind="css: {'_with-tooltip': $data.tooltip, '_with-reset': $data.showFallbackReset && $data.isDifferedFromDefault}">
  <!-- ko ifnot: hasAddons() -->
  <!-- ko template: elementTemplate -->
<div class="admin__actions-switch" data-role="switcher"> == $0
  <input type="checkbox" class="admin__actions-switch-checkbox" data-bind="attr: {id: uid, name: inputName}, value: value, disable: disabled, hasFocus: focused, simpleChecked: checked" id="BYN9SCP" name="product[status]" value="1">
  <label class="admin__actions-switch-label xh-highlight" data-bind="attr: {for: uid}" for="BYN9SCP">...
```

QUERY  
//\*[contains(@class, "admin\_\_actions-switch")]/input

RESULTS (1)

Simple product with category 1551498823

Store View: All Store Views

← Back

Enable Product [website]   
`input#BYN9SCP.admin__actions-switch-checkbox.xh-highlight | 1x1`

Attribute Set Default

Elements Console Sources Network Performance Memory Application Security Audits AdBlock Adblock P...

```
<div class="admin__field-control" data-bind="css: {'_with-tooltip': $data.tooltip, '_with-reset': $data.showFallbackReset && $data.isDifferedFromDefault}">
  <!-- ko ifnot: hasAddons() -->
  <!-- ko template: elementTemplate -->
<div class="admin__actions-switch" data-role="switcher">
  <input type="checkbox" class="admin__actions-switch-checkbox xh-highlight" data-bind="attr: {id: uid, name: inputName}, value: value, disable: disabled, hasFocus: focused, simpleChecked: checked" id="BYN9SCP" name="product[status]" value="1">
  <label class="admin__actions-switch-label" data-bind="attr: {for: uid}" for="BYN9SCP">...
```

# GENERAL QUESTIONS

# WHY ARE WE TALKING ABOUT SELECTORS?

While reviewing Tests & Section Objects I've ran across selectors that make me...



# WHICH SELECTORS?

#1

```
div[data-role="banner"] .pagebuilder-mobile-hidden[style*="40px calc(50% + 40px) 40px 40px"]  
[style*="magento"]
```

#2

```
(//div[contains(@class,"pagebuilder-buttons")])[{{arg1}}][@style="border: {{arg2}}; border-radius: {{arg3}}px;  
margin: {{arg4}}px; padding: {{arg5}}px;"]
```

#3

```
//div[contains(@style,"padding: {{arg1}}")]
```

#4

```
//div[contains(@class, "admin__data-grid-action-bookmarks")]/ul/li/div/a[text() = "{{label}}"]
```

# REMEMBER, WHEN IT COMES TO SELECTORS...

- “Less is more” [most of the time]

- Robert Browning

- “Keep It Simple, Stupid”

- U.S. Navy 1960's

# *WHICH IS BETTER CSS OR XPATH?*

- CSS is more stable, easier to read/write\* and maintain (typically).
- Xpath provides several powerful tools and it's been around the longest so it's well documented.
- Xpath can be less stable and potentially unsupported by certain actions in Selenium.

# *WHAT IS THE PRIORITY FOR SELECTORS?*

- 1<sup>st</sup> – ID, Class, Name or anything else that is unique
- 2<sup>nd</sup> – Complex CSS selectors
- 3<sup>rd</sup> – Xpath selectors
- n<sup>th</sup> – Ask the Developer to add a unique ID or Class

# WHY ARE WE STILL TALKING ABOUT XPATH?

*Unsupported CSS Pseudo Selectors*

*in the latest version of Selenium (v3.141.5)\*:*

- `:first`
- `:contains`
- `:has`

*in previous versions of Selenium\*\*:*

- `:first-of-type`
- `:last-of-type`
- `:nth-of-type(n)`
- `:only-of-type`
- `:first-child`
- `:last-child`
- `:nth-child(n)`
- `:only-child`
- `:empty`

*Unsupported CSS Pseudo Selectors*

# WHAT HAPPENS IF YOU USE THEM?

The screenshot shows a terminal window with the following output:

```
Terminal
+ Local Local (1)
x -----
1) ExperimentTestCest: Experiment test
   Test tests/functional/Magento/FunctionalTest/_generated/default/ExperimentTestCest.php:ExperimentTest
      [MalformedLocatorException] CSS or XPath locator is malformed: a:contains('Dashboard')
```

Below the terminal, a code editor pane displays a file with the following content:

```
a:contains("Dashboard") (1 match)
css less sass + ✎ ⌂ ⌂
a:contains("Dashboard")
```

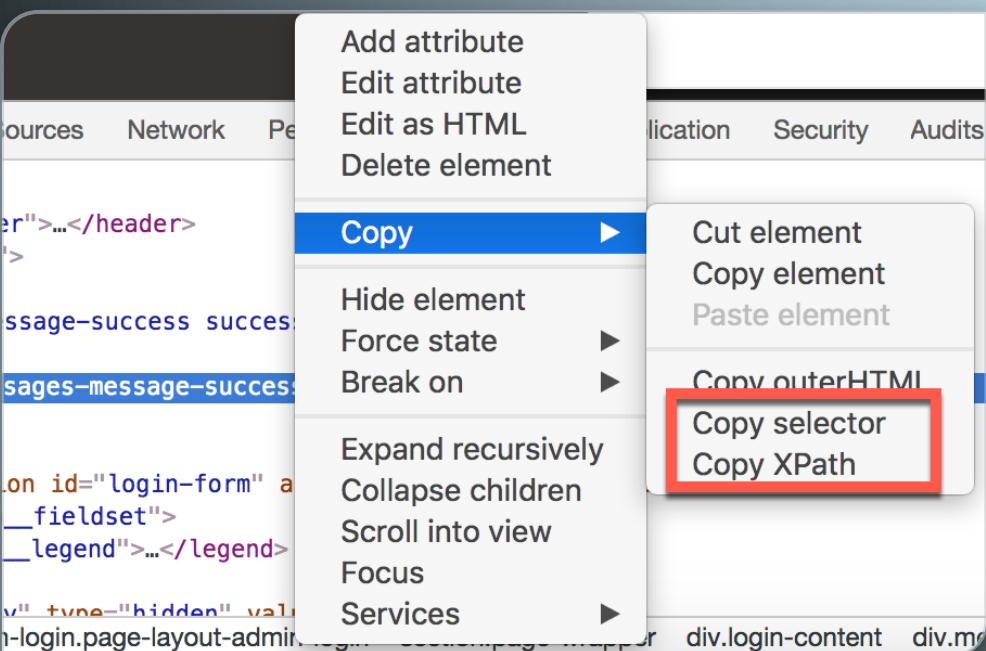
On the right side of the interface, there are several panels:

- Structure: Shows a tree view of the project structure.
- Favorites: Shows a list of favorite files.
- 6: TODO
- 9: Version Control
- Terminal



# *DO'S AND DON'TS*

# DON'T #1



- DO NOT right click on an Element in the “Browser Inspector” and select ‘Copy selector’ or ‘Copy Xpath’.

- ‘Copy selector’ results:
  - `#html-body > section > div > div > div > div`
- ‘Copy Xpath’ results:
  - `//*[@id="html-body"]/section/div/div/div/div`

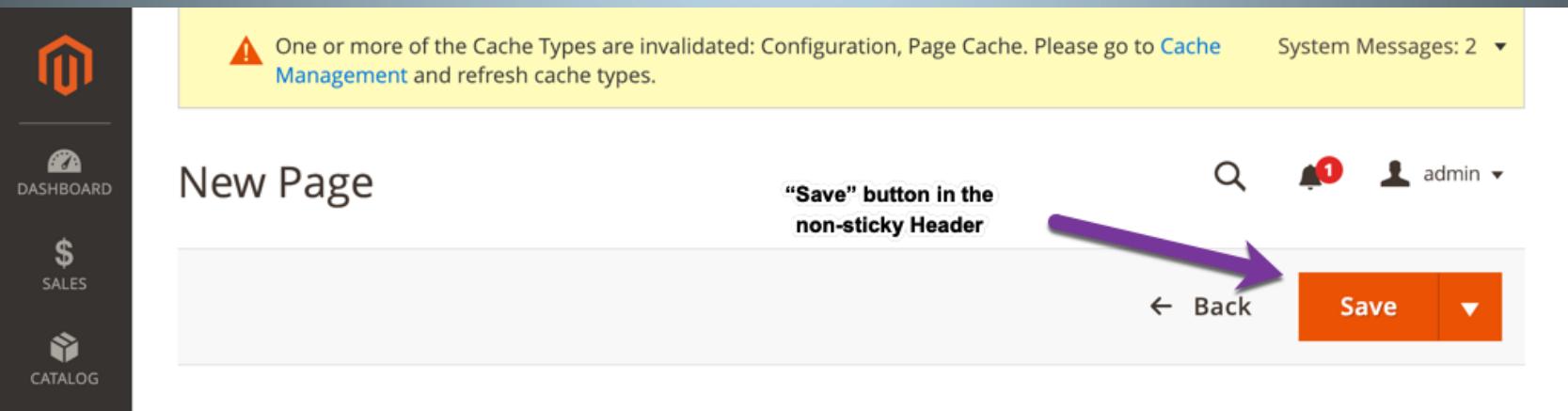
## DON'T #2

- Don't make your selectors too generic.

### Examples

- `input[name*="firstname"]`
- `.add`
- `.save`

# DON'T #2 EXAMPLE



## DON'T #3

- Don't make your selectors too specific either.

### Examples

- `#container .dashboard-advanced-reports .dashboard-advanced-reports-description .dashboard-advanced-reports-title`
- `//*[@id='container']/*[@class='dashboard-advanced-reports']/*[@class='dashboard-advanced-reports-description']/*[@class='dashboard-advanced-reports-title']`

# DON'T #3 EXAMPLE

The screenshot shows a Magento Business Intelligence dashboard. At the top, a yellow banner displays a warning: "⚠ Failed to synchronize data to the Magento Business Intelligence service. [Retry Synchronization](#)". The main area is titled "Dashboard" and includes a "Store View" dropdown set to "All Store Views". A red dashed box highlights the "Advanced Reporting" section, which contains the text: "Gain new insights and take command of your business data." Below this, a blue box shows a snippet of CSS code: ".description .dashboard-advanced-reports-title". A small CSS editor interface is overlaid on the right side of the reporting section.

⚠ Failed to synchronize data to the Magento Business Intelligence service. [Retry Synchronization](#)

## Dashboard

Store View: All Store Views ?

**Advanced Reporting**

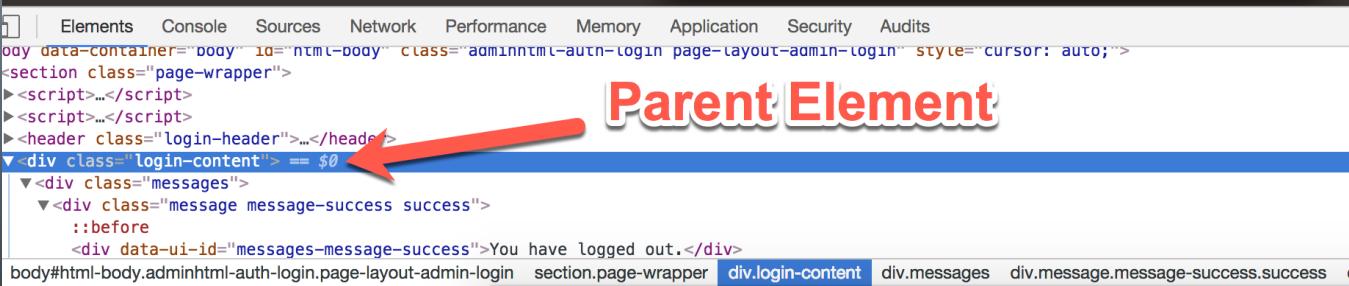
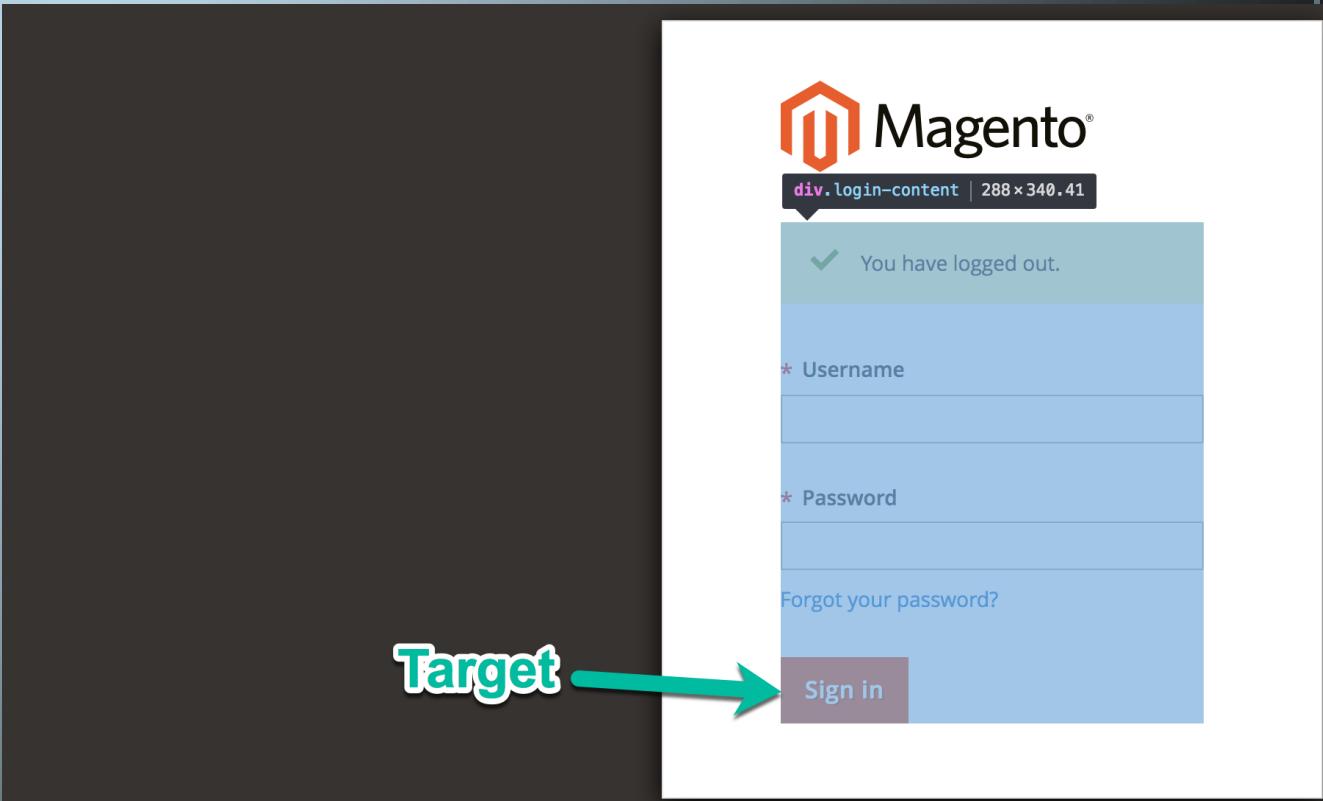
Gain new insights and take command of your business data.

```
css less sass + ⚡ ✎ 🤔 🔍 x
.description .dashboard-advanced-reports-title
```

# DO #1 – ISOLATION

Think "Isolation":

- Isolate the parent element 1<sup>st</sup>
- Then target the child element under the parent element
- This reduces the amount of DOM that you need to worry about



# DO #2 – REVERSE ISOLATION

- IF you need to interact with the parent element but it's too generic, AND the internal contents are unique:
  - Target the unique internal content 1<sup>st</sup>
  - Then Jump to the parent element

The screenshot shows a web application interface titled "Products". On the left, there is a sidebar with icons for CATALOG, CUSTOMERS, MARKETING, CONTENT, REPORTS, STORES, and SYSTEM. The main area displays a table with one record found. The table columns are ID,Thumbnail,Name,Type,Attribute Set,SKU,Price,Quantity per Source, and Salable Quantity. The single record is for a "Virtual MSI Product" with SKU "Virtual\_MSI\_Product\_Safeffba0f82", Price \$100.00, and Quantity per Source 1. A tooltip with the text "Then jump to the parent" has a purple arrow pointing to the parent element in the DOM tree. Another tooltip with the text "Target unique content 1st" has a green arrow pointing to the unique content element.

ID	Thumbnail	Name	Type	Attribute Set	SKU	Price	Quantity per Source	Salable Quantity
113		Virtual MSI Product Safeffba0f82	Virtual Product	Default	Virtual_MSI_Product_5afeffba0f82	\$100.00	Test Source 1 Safeffba8c724242 :100	Basic N 1 Safeff :99

```
<!-- ko repeat: {foreach: rows, item: '$row'} -->
  <tr class="data-row" data-bind="css: {'_odd-row': $index % 2}" data-repeat-index="0">
    <!-- ko fastForEach: {data: getVisible(), as: 'col'} -->
      <td data-bind="css: getFieldClass($row()), click: getFieldHandler($row()), template: getBody()" class="data-grid-checkbox-cell">...</td>
      <td data-bind="css: getFieldClass($row()), click: getFieldHandler($row()), template: getBody()" class="data-grid-thumbnail-cell">...</td>
      <td data-bind="css: getFieldClass($row()), click: getFieldHandler($row()), template: getBody()"/>Virtual MSI Product Safeffba0f82</td>
    </td>
    <td data-bind="css: getFieldClass($row()), click: getFieldHandler($row()), template: getBody()"/>...</td>
    <td data-bind="css: getFieldClass($row()), click: getFieldHandler($row()), template: getBody()"/>...</td>
```

# CSS OVERVIEW

# GENERAL CSS SELECTORS

TYPE	HTML	SELECTOR
ID's	<code>&lt;div id="idname"&gt;/&lt;/div&gt;</code>	<code>#idname {}</code>
Classes	<code>&lt;div class="classname"&gt;/&lt;/div&gt;</code>	<code>.classname {}</code>
HTML Tags	<code>&lt;div&gt;/&lt;/div&gt;</code>	<code>div {}</code>
HTML Tag & ID	<code>&lt;div id="idname"&gt;/&lt;/div&gt;</code>	<code>div#idname {}</code>
HTML Tag & Class	<code>&lt;div class="classname"&gt;/&lt;/div&gt;</code>	<code>div.classname {}</code>
ID & Class	<code>&lt;div id="idname" class="classname"&gt;/&lt;/div&gt;</code>	<code>#idname.classname {}</code>
HTML Tag & ID & Class	<code>&lt;div id="idname" class="classname"&gt;/&lt;/div&gt;</code>	<code>div#idname.classname {}</code>

# CSS SELECTOR OPERATORS

SYMBOL	NAME	PURPOSE	SELECTOR
*	Universal Selector	Allows you to select ALL ELEMENTS on the Page. Wild Card.	* {}
Whitespace	Descendant Combinator	Allows you to combine 2 or more selectors.	#idname .classname {}
>	Child Combinator	Allows you to select the <b>top-level <u>elements</u></b> THAT FOLLOWS another specified <u>element</u> .	#idname > .classname {}
+	Adjacent Sibling Combinator	Allows you to select an <u>element</u> THAT FOLLOWS DIRECTLY AFTER another specified <u>element</u> .	#idname + .classname {}
~	General Sibling Combinator	Allows you to select an <u>element</u> THAT FOLLOWS (directly or indirectly) another specified <u>element</u> .	#idname ~ .classname {}

# CSS ATTRIBUTE SELECTOR OPERATORS

SYMBOL	PURPOSE	EXAMPLE
=	Returns all elements that CONTAIN the EXACT string in the <u>value</u> .	[attribute="value"]
=	Returns all elements that CONTAINS or STARTS WITH the substring in the <u>value</u> .	[attribute = "value"]
*=	Returns all elements that CONTAINS the substring in the <u>value</u> .	[attribute*= "value"]
~=	Returns all elements that CONTAINS the <i>given words delimited by spaces</i> in the <u>value</u> .	[attribute~= "value"]
\$=	Returns all elements that ENDS WITH the substring in the <u>value</u> .	[attribute\$= "value"]
^=	Returns all elements that BEGIN EXACTLY WITH the substring in the <u>value</u> .	[attribute^= "value"]
!=	Returns all elements that either DOESN'T HAVE the given <u>attribute</u> <b>or</b> the <u>value</u> of the <u>attribute</u> is NOT EQUAL to the <u>value</u> .	[attribute!="value"]

# XPATH OVERVIEW

/ VS //

NAME	SYMBOL	SELECTOR
Absolute XPath Selectors	/	/html/body/div[2]/div/div[2]/div[1]/div[2]/form/div/input
Relative XPath Selectors	//	//div[@class="form-group"]//input[@id="user-message"]

NAME	SYMBOL	HTML	SELECTOR
Parent Selector	..	<pre>&lt;tr&gt;   &lt;td&gt;     &lt;div&gt;Unique Value&lt;/div&gt;   &lt;/td&gt; &lt;/tr&gt;</pre>	<code>//*[text()="Unique Value"]/..../..</code>
Parent Selector	..	<pre>&lt;tr&gt;   &lt;td&gt;     &lt;a href="#"&gt;Edit&lt;/a&gt;   &lt;/td&gt;   &lt;td&gt;     &lt;div&gt;Unique Value&lt;/div&gt;   &lt;/td&gt; &lt;/tr&gt;</pre>	<code>//div[text()="Unique Value"]/..../..//a</code>

# ATTRIBUTE SELECTORS

- Allows you to select an element that contains a specific attribute or a value in a specific attribute.

## SYNTAX

- `//*[@attribute]`
- `//*[@attribute="value"]`

# ATTRIBUTE SELECTORS EXAMPLES

ATTRIBUTE	HTML	SELECTOR
ID	<div id="idname"/>	//*[@id="idname"]
Class	<div class="classname"/>	//*[@class="classname"]
Type	<button type="submit"/>	//*[@type="submit"]
Value	<input value="value"/>	//*[@value="value"]
Href	<a href="https://google.com"/>	//*[@href="https://google.com"]
Src		//*[@src="/img.png"]

# CONTAINS() SELECTOR

- Allows you to select an element that contains a specific attribute or a value in a specific attribute.

## SYNTAX

- `//*[contains(@attribute, "")]`
- `//*[contains(@attribute, "value")]`

# CONTAINS() SELECTOR EXAMPLES

ATTRIBUTE	HTML	EXAMPLE
<code>text()</code>	<code>&lt;p&gt;Hello World!&lt;/p&gt;</code>	<code>//*[contains(text(), "Hello")]</code>
<code>@id</code>	<code>&lt;div id="idname1234abcd"/&gt;</code>	<code>//*[contains(@id, "idname")]</code>
<code>@class</code>	<code>&lt;div class="classname1 classname2"/&gt;</code>	<code>//*[contains(@class, "classname1")]</code>
<code>@name</code>	<code>&lt;input name='inputname'/&gt;</code>	<code>//*[contains(@name, "name")]</code>
<code>@value</code>	<code>&lt;input value="value"/&gt;</code>	<code>//*[contains(@value, "value")]</code>
<code>@href</code>	<code>&lt;a href="https://google.com"/&gt;</code>	<code>//*[contains(@href, "google.com")]</code>

# TEXT() SELECTOR

- Allows you to select an element that contains a specific *string*.

## SYNTAX

- `//*[text()="exact string"]`
- `//*[contains(text(), "substring")]`

# TEXT() SELECTOR EXAMPLES

TYPE	HTML	EXAMPLE
Exact String Matching	<p>CSS Selectors are great!!!</p> <p>CSS Selectors are great.</p>	//p[text()="CSS Selectors are great!!!"]
Substring Matching	<p>CSS Selectors are great!!!</p> <p>CSS Selectors are great.</p>	//p[contains(text(), "CSS Selectors")]

# **STARTS-WITH() SELECTOR**

- Allows you to select an element that STARTS WITH the substring or contains text that STARTS WITH the substring.

## SYNTAX

- `//*[starts-with(@attribute, "substring")]`
- `//*[starts-with(text(), "substring")]`

# STARTS-WITH() SELECTOR EXAMPLES

ATTRIBUTE	HTML	EXAMPLE
@id	<div id="unique_id_abcd1234"/>	//*[starts-with(@id, "unique_id")]
@class	<div class="unique_class_abcd1234"/>	//*[starts-with(@class, "unique_class")]
@href	<a href="https://www.google.com/">/>	//a[starts-with(@href, "https://")]
text()	<p>Hello World!</p>	//p[starts-with(text(), "Hello")]

# *ENDS-WITH() SELECTOR*

- Allows you to select an element that ENDS WITH the substring or contains text that ENDS WITH the substring.

## SYNTAX

- `//*[ends-with(@attribute, "substring")]`
- `//*[ends-with(text(), "substring")]`

# ENDS-WITH() SELECTOR EXAMPLES

ATTRIBUTE	HTML	EXAMPLE
@id	<div id="abcd1234_unique_id"/>	//*[ends-with(@id, "unique_id")]
@class	<div class="abcd1234_unique_class"/>	//*[ends-with(@class, "unique_class")]
@href	<a href="https://www.google.com/">/</a>	//a[ends-with(@href, "google.com/")]
text()	<p>Hello World!</p>	//p[ends-with(text(), "World!")]

# ANCESTOR:: SELECTOR

- Allows you to select an element that CONTAINS a specific element UNDER IT.

## SYNTAX

- `//target_element[ancestor::unique_element[contains(., "value")]]`

# ANCESTOR:: SELECTOR EXAMPLES

TARGET	HTML	EXAMPLE
“Edit” Button	SEE SCREENSHOT BELOW	//a[ancestor::tr[contains(., “1551498823”)]]

## QUERY

```
//a[ancestor::tr[contains(., '1551498823')]]
```

## RESULTS (1)

Edit

Add Product ▾

Filters

Default View ▾

Columns ▾

## DASHBOARD

Search by keyword



Actions ▾

5 records found

20 ▾

per page



1 of 1



Attribute Set	SKU	Price	Quantity	Visibility	Status	Website	Action
Default	simple_product_with_category_1551498823	\$100.00	777.0000	Catalog, Search	Enabled	Main Website	<a href="#">Edit</a>

Elements Console Sources Network Performance Memory Application Security Audits Adblock Plus AdBlock

```

<!-- ko repeat: {foreach: rows, item: '$row'} -->
<tr class="data-row" data-bind="css: {'_odd-row': $index % 2}" data-repeat-index="0">
    <!-- ko fastForEach: {data: getVisible(), as: '$col'} -->
    ><td data-bind="css: getFieldClass($row()), click: getFieldHandler($row()), template: getBody()" class="data-grid-cell">
    ><td data-bind="css: getFieldClass($row()), click: getFieldHandler($row()), template: getBody()>...</td>
    ><td data-bind="css: getFieldClass($row()), click: getFieldHandler($row()), template: getBody()>
        <div class="data-grid-cell-content" data-bind="text: $col.getLabel($row())">
            simple_product_with_category_1551498823</div> == $0
        </td>
    ><td data-bind="css: getFieldClass($row()), click: getFieldHandler($row()), template: getBody()>...</td>
    ><td data-bind="css: getFieldClass($row()), click: getFieldHandler($row()), template: getBody()" class="data-grid-actions-cell">
        <!-- ko if: $col.isSingle($row()._rowIndex) -->
        <!-- ko repeat: {foreach: $col.getVisibleActions($row()._rowIndex), item: '$action'} -->
        <a class="action-menu-item xh-highlight" data-bind="attr: {target: $col.getTarget($action()), href: $action().href}, text: $action().label, click: $col.getActionHandler($action())" data-repeat-index="0"
            target="_self" href="http://magento2ce.local/magento2ce/admin/catalog/product/edit/id/1/">Edit</a>
        </td>
    ></tr>

```

# *REPLICATING CSS IN XPATH*

# REPLICATING GENERAL CSS SELECTORS IN XPATH

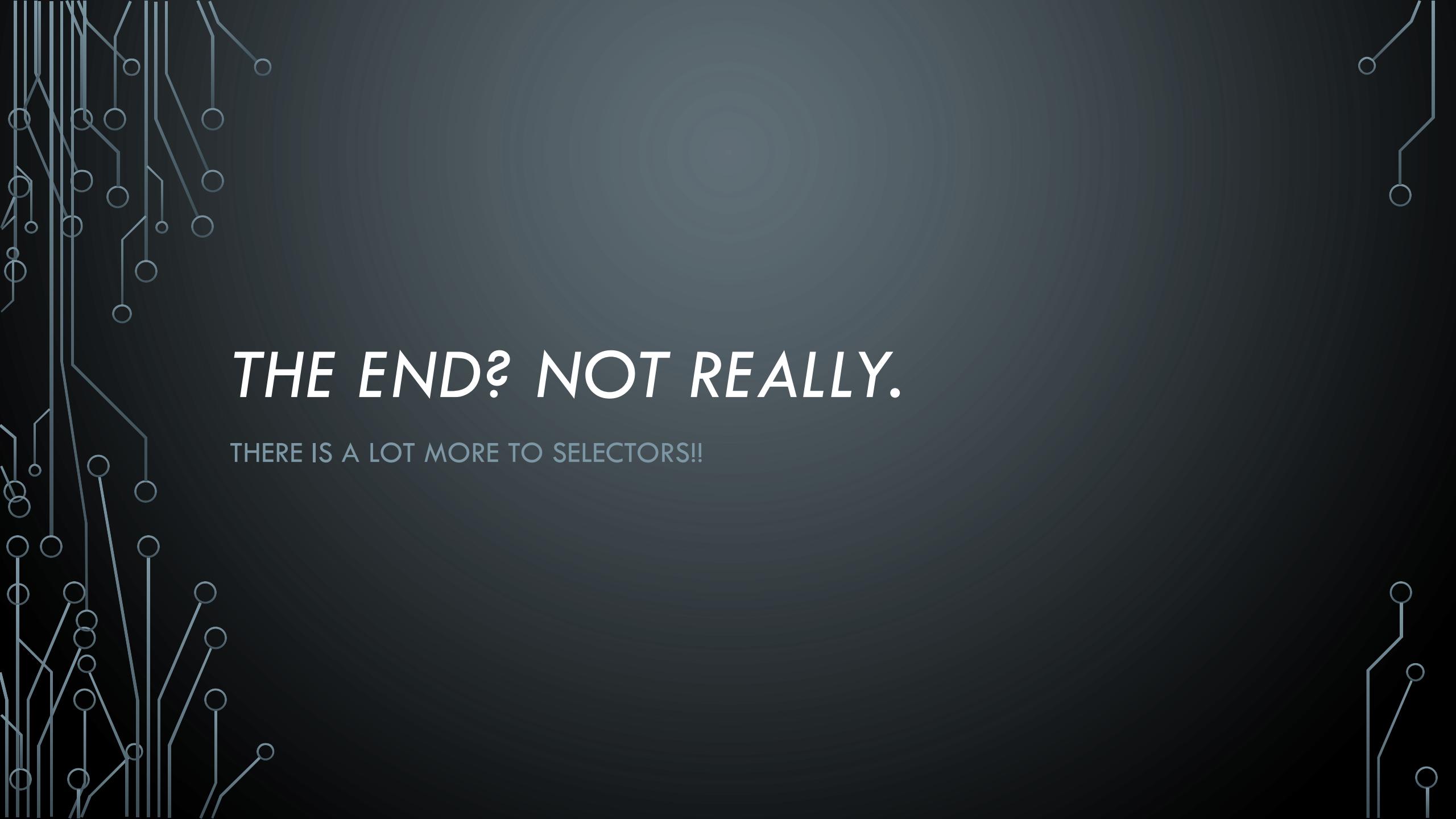
TYPE	CSS	XPATH
ID's	#idname {	//*[@@id="idname"]
Classes	.classname {	//*[@@class="classname"]
HTML Tags	div {	//div
HTML Tag & ID	div#idname {	//div[@id="idname"]
HTML Tag & Class	div.classname {	//div[@class="classname"]
ID & Class (?)	#idname.classname {	//*[@@id="idname"][@class="classname"]

# REPLICATING SELECTOR OPERATORS IN XPATH

NAME	CSS	XPATH
Universal Selector	* {}	//*[.]
Descendant Combinator	#idname .classname	//*[@@id="idname"]//*[@@class="classname"]
Child Combinator	#idname > .classname	//*[@@id="idname"]/*[@class="classname"]

# REPLICATING ATTRIBUTE SELECTOR OPERATORS

SYMBOL	CSS	XPATH
=	a[href="/"]	//a[@href="/"]
*=	a[href*="://"]	//a[contains(@href, "://")]
~=	input[title~="help"]	//input[contains(@title, "help")]
\$=	a[href\$="pdf"]	//a[ends-with(@href, "pdf")]
^=	a[href^="/"]	//a[starts-with(@href, "/")]



*THE END? NOT REALLY.*

THERE IS A LOT MORE TO SELECTORS!!



# QUESTIONS?

# *RESOURCES*

- CSS
  - <https://devhints.io/css>
  - <https://www.w3schools.com/css/>
- XPATH
  - <https://devhints.io/xpath>
  - [https://www.w3schools.com/xml/xpath\\_intro.asp](https://www.w3schools.com/xml/xpath_intro.asp)
- CONFLUENCE
  - <https://wiki.corp.magento.com/display/QMT>Selectors>