

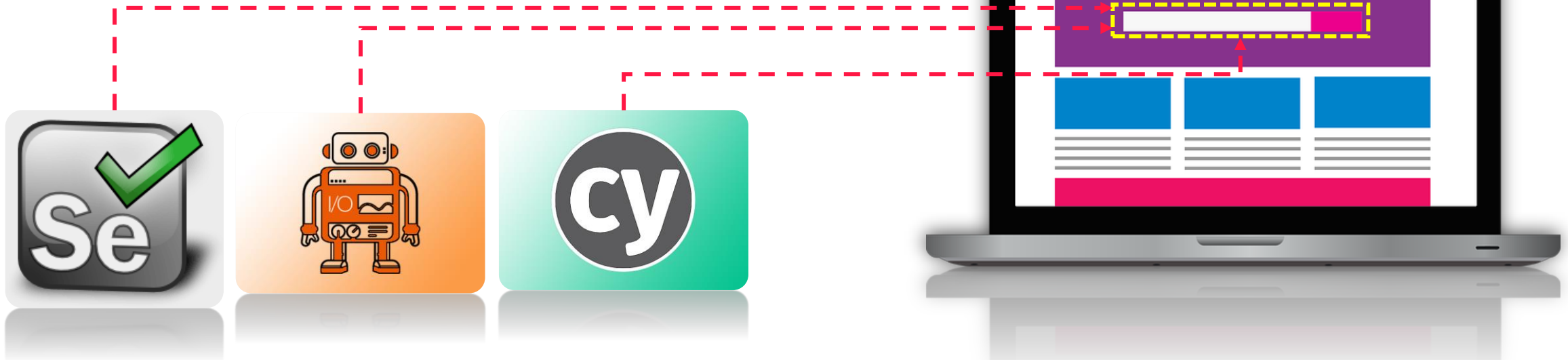
# **AUTOMATION TESTING FRAMEWORK(S) & SELECTORS**

# WHY STUDY SELECTORS?

- Major skill / component required when building effective automation test(s) and test automation framework(s).
- Improve the efficiency and robustness of your tests.
- We require a means of interacting with the System Under Test (SUT); selectors enable us to locate whilst interact with given web element(s) etc.

# SELECTORS & TEST AUTOMATION FRAMEWORKS

- Many of the Popular Test Automation Frameworks require selectors to interact with the target system (SUT):
  - **Selenium WebDriver** (Java, Python etc)
  - **Cypress**
  - **WebdriverIO** etc.

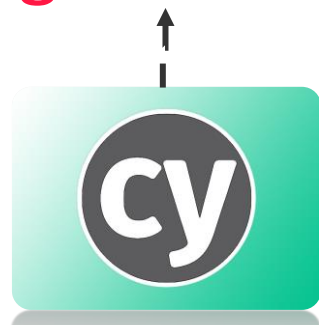
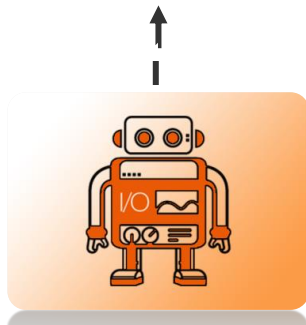
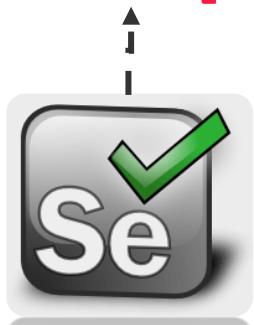


# PRACTICAL EXAMPLE

- We need an automated test which interacts with the following 'Send' button.

- The following CSS selector will locate the 'Send' button:

`button[id='sendMessageButton']`



Phone Number

---

Message

---

Send

```
<div class="form-group">  
  <button class="btn btn-primary btn-xl" id="sendMessageButton"  
    type="submit">Send</button> == $0  
</div>  
</form>
```

# SELECTORS & AUTOMATION TESTING TECHNOLOGIES

## Selenium WebDriver

```
driver.findElement(By.cssSelector,"button[id='sendMessageButton']").click();  
driver.findElement(By.id,"sendMessageButton").click();  
By.name  
By.className  
By.tagName  
By.linkText  
By.partialLinkText
```

## WebdriverIO

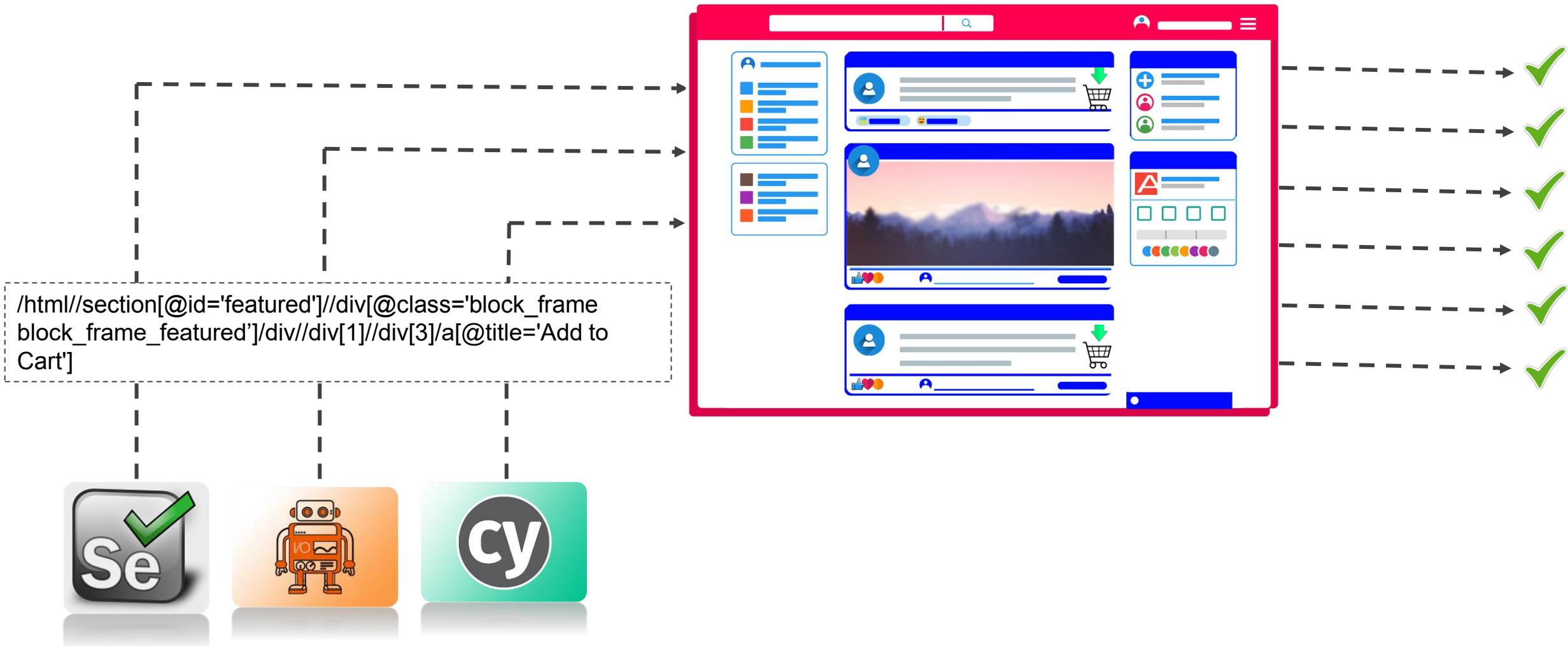
```
const elem = $(selector)  
  
const elem = $('button[id='sendMessageButton']')  
elem.click()  
  
elem.getText()
```

## Cypress

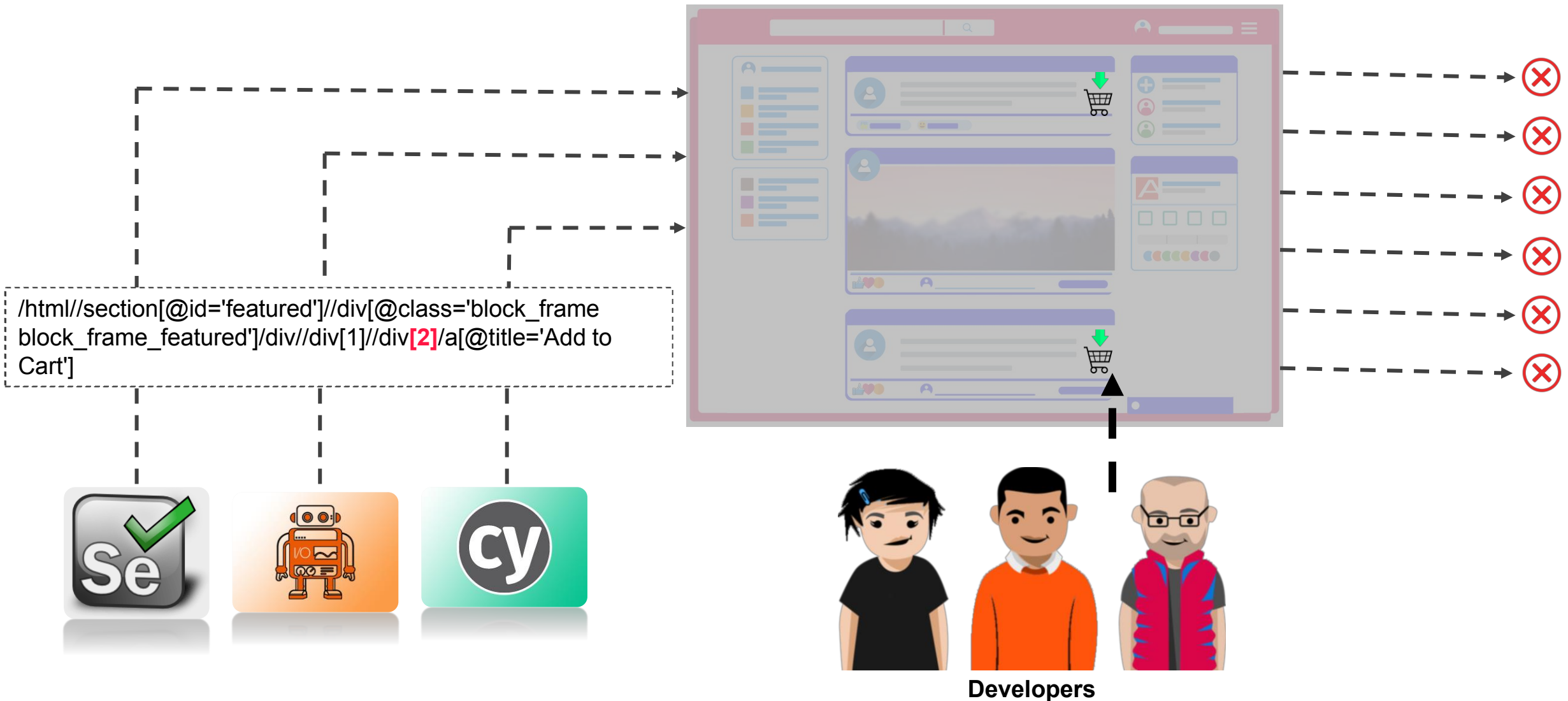
```
cy.get(selector)  
  
cy.get('button[id='sendMessageButton']").click()  
  
cy.get('#sendMessageButton').click()
```

# THE IMPORTANCE OF SELECTORS

# WHY ARE SELECTORS SO IMPORTANT?



# WHY ARE SELECTORS SO IMPORTANT?





# **HTML – DOM & ELEMENTS**

# HTML DOM - (DOCUMENT OBJECT MODEL)

- **HTML DOM** is an Object Model for HTML.
- What is **HTML**?
  - **Hypertext Mark-up Language** is the standard mark-up language for documents designed to be displayed in a web browser.
- What is **DOM**?
  - **Document** just like a file; html file, word file, text file etc.
  - **Object** can be anything you put inside the HTML document; <head> tags, <h1> tags etc.
  - **Model** how you layout the structure.

# HTML DOM - (DOCUMENT OBJECT MODEL)

## EXAMPLE WEBSITE

Not secure | webdriveruniversity.com/Contact-Us/contactus.html

webdriverUniversity.com (New Approach To Learning)

### CONTACT US

Document: index.html

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
```

```
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
</head>
```

```
<body>
```

```
<!-- Navigation -->
<nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
</nav>
```

```
<!-- Page Content -->
```

```
<div class="container">
```

```
<form action="contact_us.php" method="post" id="contact_form">
```

```
<input name="first_name" type="text" class="feedback-input" placeholder="First Name" />
```

```
<input name="last_name" type="text" class="feedback-input" placeholder="Last Name" />
```

```
<input name="email" type="text" class="feedback-input" placeholder="Email Address" />
```

```
<textarea name="message" class="feedback-input" placeholder="Comments"></textarea>
```

```
<div id="form_buttons" class="text-center">
```

```
<input class="contact_button" type="reset" value="RESET" />
```

```
<input class="contact_button" type="submit" value="SUBMIT" />
```

```
</div>
```

```
</form>
```

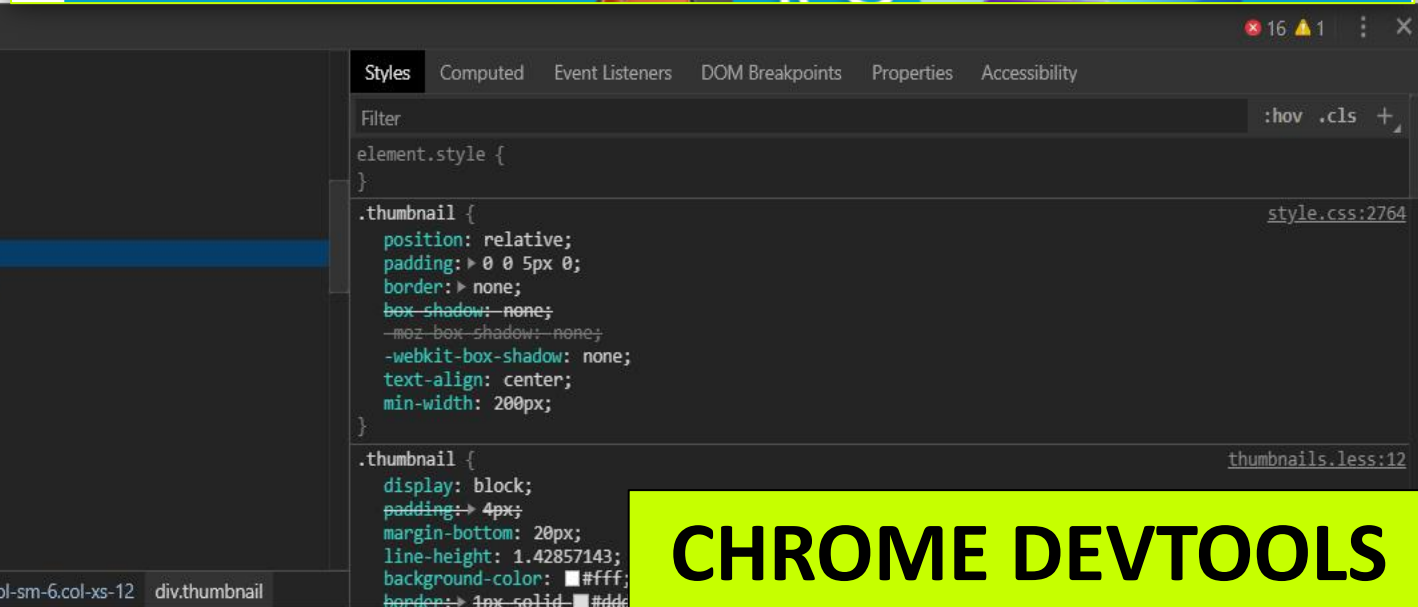
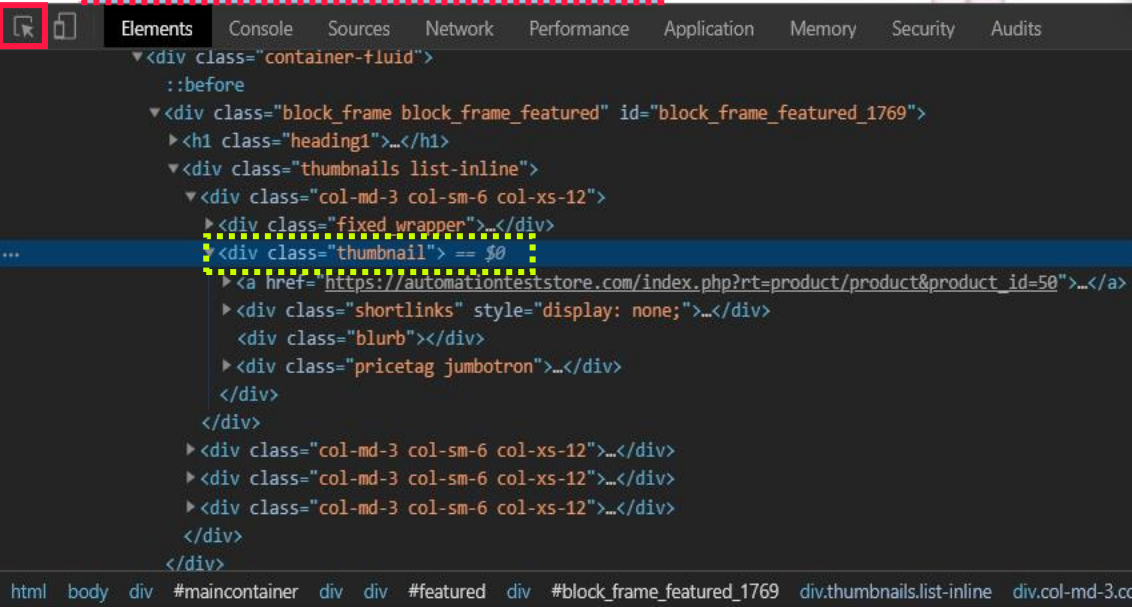
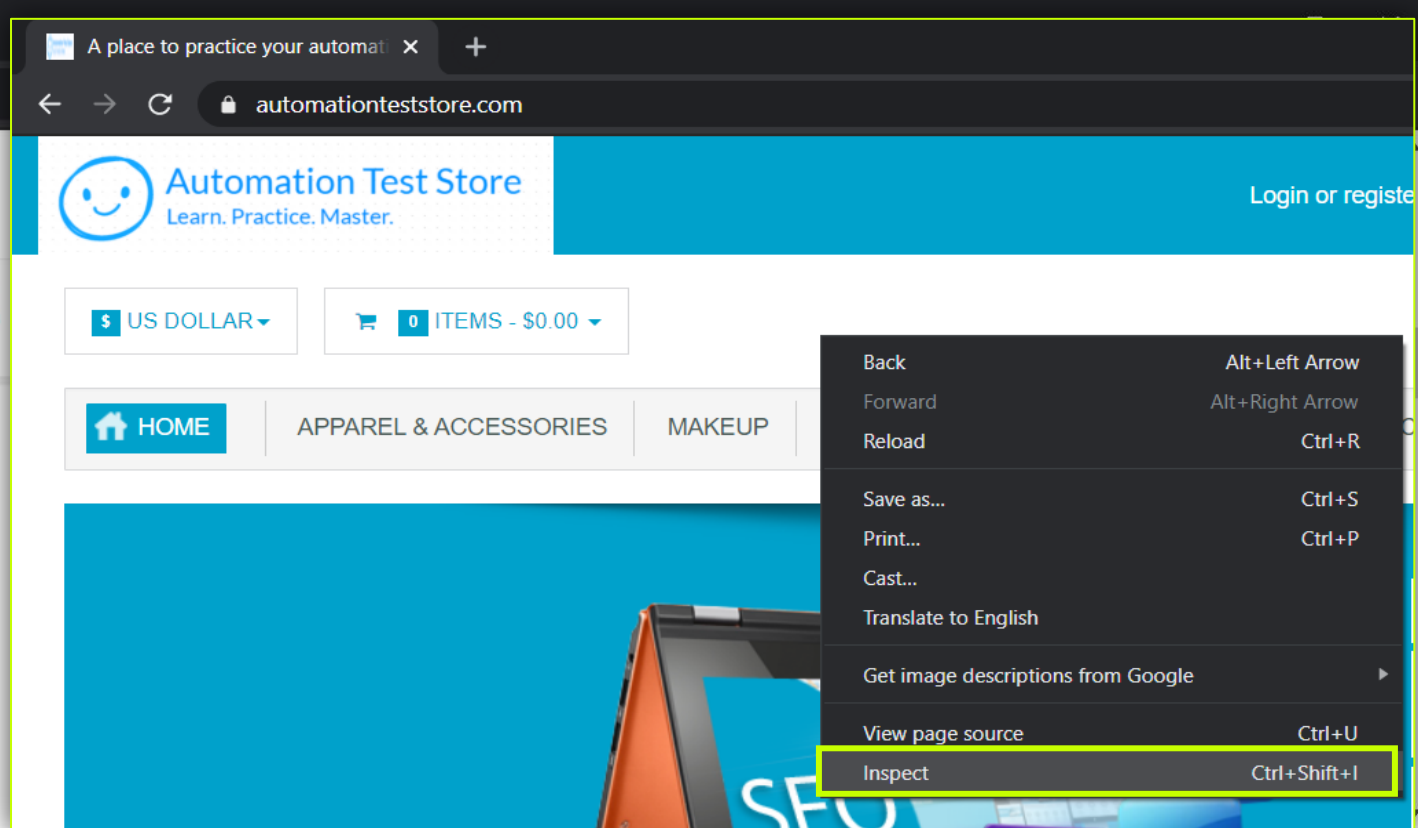
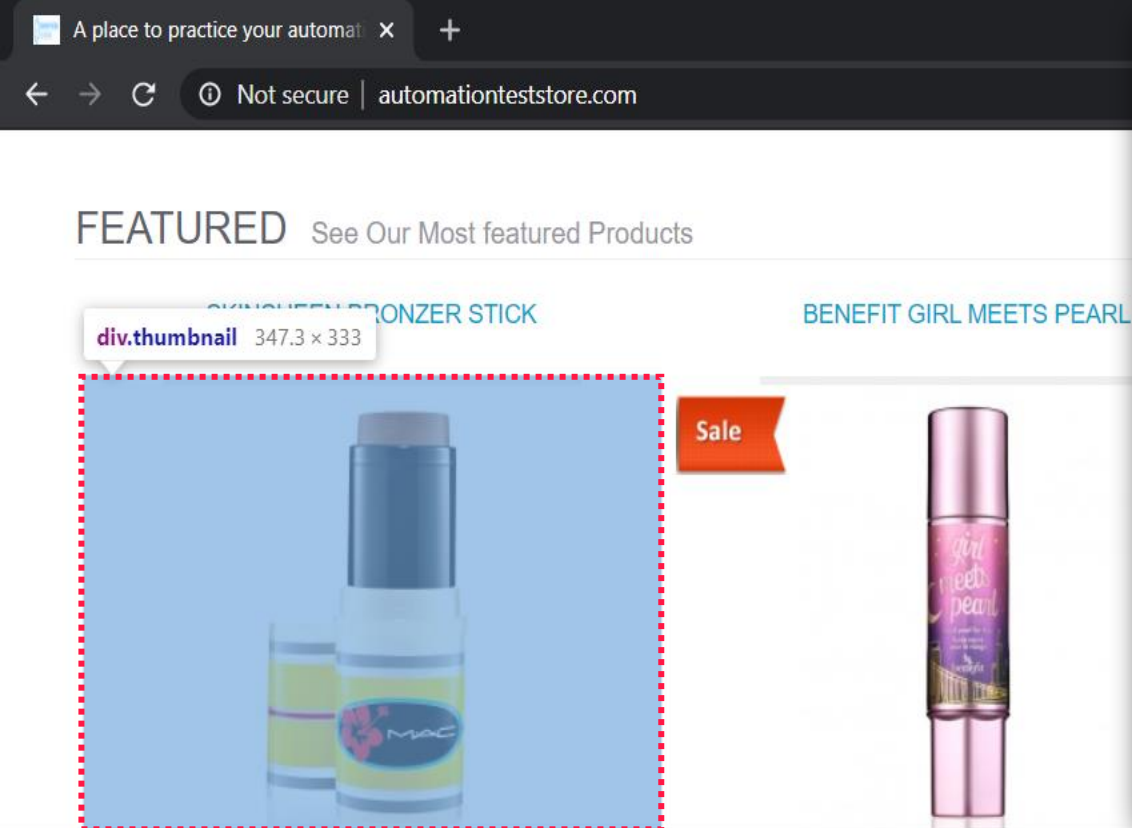
```
</div>
```

```
</body>
```

```
</html>
```

Objects: tags

Model



CHROME DEVTOOLS

# HTML TAGS - Opening Tags: **<div>** & **<a>**

```
<div class="thumbnail"> == $0
  <a href="https://automationteststore.com/index.php?rt=product/product&product_id=50">
    
  </a>
  <div class="shortlinks" style="display: none;">...</div>
  <div class="blurb"></div>
  <div class="pricetag jumbotron">...</div>
</div>
</div>
<div class="col-md-3 col-sm-6 col-xs-12">...</div>
<div class="col-md-3 col-sm-6 col-xs-12">...</div>
<div class="col-md-3 col-sm-6 col-xs-12">...</div>
</div>
```

```
<\q?A>
  <q?A CJ922=„coJ-wq-3 coJ-2w-e coJ-x2-J5„>“<\q?A>
  <q?A CJ922=„coJ-wq-3 coJ-2w-e coJ-x2-J5„>“<\q?A>
  <q?A CJ922=„coJ-wq-3 coJ-2w-e coJ-x2-J5„>“<\q?A>
  <\q?A>
```

# HTML TAGS - Closing Tags: `</div>` & `</a>`

```
<div class="thumbnail"> == $0
  <a href="https://automationteststore.com/index.php?rt=product/product&product_id=50">
    
  </a>
  <div class="shortlinks" style="display: none;">...</div>
  <div class="blurb"></div>
  <div class="pricetag jumbotron">...</div>
</div>
<div class="col-md-3 col-sm-6 col-xs-12">...</div>
<div class="col-md-3 col-sm-6 col-xs-12">...</div>
<div class="col-md-3 col-sm-6 col-xs-12">...</div>
</div>
```

```
<\q?A>
> <q?A CJ922=„coJ-wq-3 coJ-2w-e coJ-x2-J5„>“<\q?A>
> <q?A CJ922=„coJ-wq-3 coJ-2w-e coJ-x2-J5„>“<\q?A>
> <q?A CJ922=„coJ-wq-3 coJ-2w-e coJ-x2-J5„>“<\q?A>
<\q?A>
```

# CLASS ATTRIBUTE

```
<!DOCTYPE html>
<html>
<head>
<style>
.cities {
  background-color: tomato;
  color: black;
  margin: 20px;
  padding: 20px;
}
</style>
</head>
<body>

<div class="cities">
<h2>London</h2>
<p>London is the capital of England.</p>
</div>

<div class="cities">
<h2>Rome</h2>
<p>Rome is the capital of Italy.</p>
</div>
</body>
</html>
```

## London

London is the capital of England.

## Rome

Rome is the capital of Italy.

The same class **.cities** is applied to multiple elements.

# ID ATTRIBUTE

```
<!DOCTYPE html>
<html>
<head>
<style>
#myHeader {
  background-color: #99ff99;
  color: black;
  padding: 40px;
  text-align: center;
}
</style>
</head>
<body>

<h2>Example id Attribute</h2>
<p>Use CSS to style an element with the id of "myHeader":</p>

<h1 id="myHeader">Cypress</h1>
</body>
</html>
```

## Example id Attribute

Use CSS to style an element with the id of "myHeader":

Cypress

HTML ID's use the # tag and can also be applied to given element(s).  
ID **#myHeader** has been applied to the h1 header tag.



```

<div class="col-md-3 col-sm-6 col-xs-12"> == $0
  <div class="fixed_wrapper">...</div>
  <div class="thumbnail">
    <a href="https://automationteststore.com/index.php?rt=product/product&product_id=50">
      
    </a>
    <div class="shortlinks" style="display: none;">...</div>
    <div class="blurb"></div>
    <div class="pricetag jumbotron">...</div>
  </div>
  <div class="col-md-3 col-sm-6 col-xs-12">...</div>
  <div class="col-md-3 col-sm-6 col-xs-12">...</div>
  <div class="col-md-3 col-sm-6 col-xs-12">...</div>
</div>
::after
</div>
::after
</section>
<div class="sep"></div>
<section id="latest" class="row mt20">...</section>
<div class="sep"></div>
<section id="bestseller" class="row mt20">...</section>
<div class="sep"></div>
<section id="special" class="row mt20">...</section>

```

Multiple **classes** applied to highlighted html tags.

HTML tags can only contain one **id**.

**ID & CLASS ATTRIBUTES**

# CLASS vs ID

- A **class** selector is a name preceded by a full stop (“.”)  
For example: `.subcategories`
- An **ID** selector is a name preceded by a hash character (“#”)  
For example: `#homepageHeader`
- The **difference between** an **ID** and a **Class** is that an **ID** can be used to identify one element, whereas a **Class** can be used to identify more than one.

Firstname	Lastname
John	Smith

Elements

Console

Sources

Network

Performance

Application

Memory

Security

Audits

```
<!doctype html>
<html lang="en">
  <head>...</head>
  <body>
    <!-- Navigation -->
    <nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">...</nav>
    <!-- Page Content -->
    <div class="container">
      ::before
      <div class="col-sm-12 text-center" id="main-header">...</div>
      <div class="col-sm-12">
        <div class="thumbnail">
          <table id="t01" style="width:100%">
            <tbody>
              <tr>
                <th>Firstname</th>
                <th>Lastname</th>
                <th>Age</th>
              </tr>
              <tr>
                <td>John</td>
                <td>Smith</td>
                <td>45</td>
              </tr>
            </tbody>
          </table>
        </div>
      </div>
    </div>
  </body>
</html>
```

Parent Tag / Element

Child Elements

PARENT & CHILD ELEMENTS

Firstname	Lastname
John	Smith

Elements

Console

Sources

Network

Performance

Application

Memory

Security

Audits

```
<!doctype html>
<html lang="en">
  <head>...</head>
  <body>
    <!-- Navigation -->
    <nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">...</nav>
    <!-- Page Content -->
    <div class="container">
      ::before
      <div class="col-sm-12 text-center" id="main-header">...</div>
      <div class="col-sm-12">
        <div class="thumbnail">
          <table id="t01" style="width:100%">
            <tbody>
              <tr>
                <th>Firstname</th>
                <th>Lastname</th>
                <th>Age</th>
              </tr>
              <tr>
                <td>John</td> == $0
                <td>Smith</td>
                <td>45</td>
              </tr>
            </tbody>
          </table>
        </div>
      </div>
    </div>
  </body>
</html>
```

Parent Tag / Element

Sibling Elements

SIBLING ELEMENTS

# CSS *vs* XPATH

# CSS vs XPATH

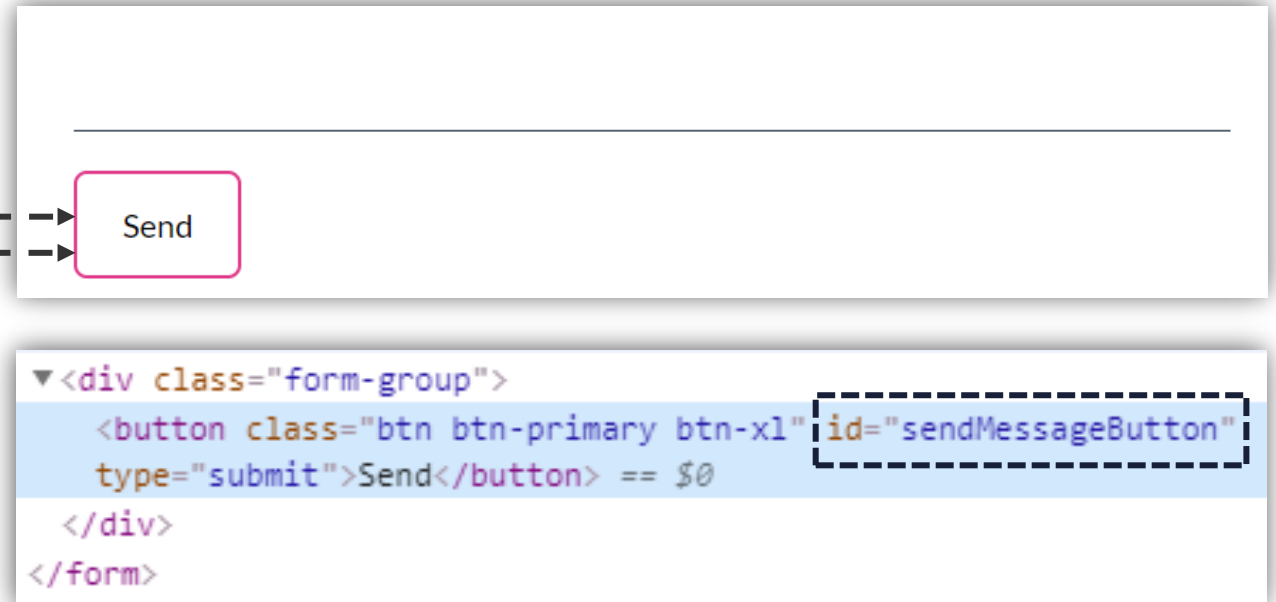
- **CSS & XPATH** are two methods which can be used to create selectors in order to locate web element(s) on a given webpage (DOM) in turn enabling us interact with them and perform specific actions.

## CSS Selector:

`button[id='sendMessageButton']`

## XPATH Selector:

`//button[@id='sendMessageButton']`



# CSS vs XPATH - *SOME* DIFFERENCES - PART 1/3

- **CSS Selectors** only move forwards.
- **XPath Selectors** have the ability to traverse both forwards and backwards.
- It is noted that CSS Selectors perform better than XPath Selectors; refer to: <http://elementalselenium.com/tips/34-xpath-vs-css-revisited-2> .
- XPath Selectors have some advantages and disadvantages over CSS Selectors.

# CSS vs XPATH - *SOME* DIFFERENCES - PART 2/3

## **XPATH *SOME* DISADVANTAGES:**

- XPath engines are different in each browser, making them inconsistent; for example an XPath's index number may vary between Chrome, Firefox and IE browsers etc. In which case a given XPath Selector may work for Chrome browser but not work for Firefox browser etc.
- XPath Selectors can become complex; in turn making them hard to read; for example:

`/html//section[@id='playground']//a[@role='button']/parent::div`

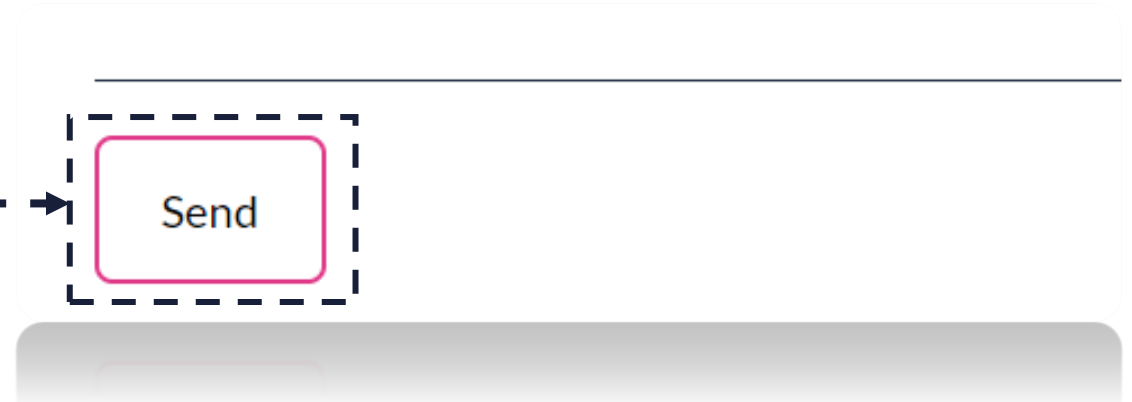


# CSS vs XPATH - *SOME* DIFFERENCES - PART 3/3

## XPATH *SOME* ADVANTAGES:

- XPath Selectors have the ability to search for Parent, Sibling elements etc.
- With XPath Selectors you can perform unique actions, for example you can create an XPath Selector which locates an element based upon text:

`//*[text()='Send']` ----->



# CSS SELECTORS

# CSS - BASIC INTRO

METHOD	EXAMPLE	REAL WORLD EXAMPLE
<b>Absolute CSS selector</b>	<b>Ranorex Selocity:</b> Right mouse click over an element > Hover over 'Selector actions' > Click on 'Copy css'.	.dashed.js-scroll-trigger.nav-link.px-0.px-lg-3.py-3
<b>Element</b>	div	a
<b>Class</b>	.class	.nav-link
<b>ID – <i>Optimal Solution</i></b>	#id	#sendMessageButton

# **CSS SELECTORS – PSEUDO-CLASSES**

# COMBINATORS – PSEUDO-CLASSES – PART 2/3

METHOD	EXAMPLE	SUMMARY
:nth-child(3)	li:nth-child(3)	Selects all <b>third</b> list item(s).
:nth-child(2)	li:nth-child(2)	Selects all <b>second</b> list item(s).
:nth-child(odd) or :nth-child(2n+1)	li:nth-child(odd)	Selects all <b>odd</b> list item(s).
:nth-child(even) or :nth-child(2n)	li:nth-child(even)	Selects all <b>even</b> list item(s).
:nth-last-child(1)	li:nth-last-child(1)	Selects all <b>last</b> list item(s).
:nth-of-type(odd)	button:nth-of-type(odd)	Matches elements of a given tag type i.e. <b>button</b> .

# **CSS SELECTORS – PSEUDO-CLASSES**

# COMBINATORS – PSEUDO-CLASSES – PART 1/3

METHOD	EXAMPLE
:disabled	input[name="vegetable"]:disabled
:enabled	input[name="vegetable"]:enabled
:checked	input[name="vegetable"]:checked

# **CSS SELECTORS – ATTRIBUTE SELECTORS**



# COMBINATORS – ATTRIBUTE SELECTORS

METHOD	EXAMPLE	REAL WORLD EXAMPLE
<b>= Exact</b>	<code>[role="dialog"]</code>	<code>[disabled]</code> <code>[value='option-1']</code>
<b>\$= Ends in</b>	<code>[href\$.doc"]</code>	<code>[value\$='1']</code>
<b>^= Begins with</b>	<code>[href^="/index"]</code>	<code>[value^='opt']</code>
<b>*= Contains</b>	<code>[class*="-is-"]</code>	<code>[value*='tion']</code>

# **CSS SELECTORS – COMBINATORS**

# COMBINATORS – CONTAINS BOTH CLASSES

METHOD	EXAMPLE	REAL WORLD EXAMPLE
Contains Both Classes	<p><b>.class1.class2</b></p> <p>‘In the following example a Web Element must contain two unique classes’.</p>	<p><b>.btn.btn-primary</b></p>

# **CSS SELECTORS – COMBINATORS**

# COMBINATORS – ADJACENT & FAR SIBLING

METHOD	EXAMPLE	REAL WORLD EXAMPLE
Adjacent sibling	<p><b>.child + .sibling</b></p> <p>‘The + combinator selects adjacent siblings. This means that the second element directly follows the first, and both share the same parent.’</p> <p>‘Sibling elements must have the same parent element, and <b>adjacent</b> means <i>immediately following</i>.’</p>	<p><b>div + img</b></p>
Far sibling	<p><b>.child ~ .sibling</b></p> <p><b><i>Element1 ~ element2</i></b></p> <p>‘Element 1 &amp; 2 must have the same parent, but element2 does not have to be immediately preceded by element1.’</p>	<p><b>div ~ p</b></p>

# **CSS SELECTORS – COMBINATORS**

# COMBINATORS – DESCENDANT & DIRECT DESCENDANT

METHOD	EXAMPLE	REAL WORLD EXAMPLE
Descendant	<p><b>.parent .child</b></p> <p>‘The descendant selector <b>matches all elements</b> that are <b>descendants</b> of a specified element.’</p>	<p><b>ul li</b></p> <p><b>ul li[id='fruits']</b></p>
Direct Descendant	<p><b>.parent &gt; child</b></p> <p>‘The child selector selects all elements that are the children of a specified element.’</p>	<p><b>div &gt; input</b></p> <p>‘Searches for direct children of a given element; <b>only looks one level down</b> the mark-up structure and does not search deeper. ‘</p>

# **XPATH SELECTORS – ATTRIBUTE SELECTORS**



# XPATH – ATTRIBUTE SELECTORS – PART 3/3

METHOD	EXAMPLE	SUMMARY
<b>^ Starts with</b>	<code>//a[starts-with(@href, 'http://www.webdriver')]</code>	Locate all <b>a</b> tagged elements who's <b>href</b> attribute starts with specific text.
<b>contains</b>	<code>//a[contains(@href,'university.com')]</code>	Locate all <b>a</b> tagged elements who's <b>href</b> attribute contains specific text.

# **XPATH SELECTORS – ATTRIBUTE SELECTORS**

# XPATH – ATTRIBUTE SELECTORS – PART 2/3

EXAMPLE	SUMMARY
<code>//input[@required='required']</code>	Locate <b>input</b> elements which contain an attribute of <b>required</b> and that attribute has a value of <b>required</b> .
<code>//input[@required='required'][@id='name']</code>	Locate <b>input</b> elements which contain an attribute of <b>required</b> and that attribute has a value of <b>required</b> and the element must also have an <b>id</b> attribute which has a value of <b>name</b> .
<code>//*[ @required]</code>	Locate <b>all</b> elements which contain an attribute of <b>required</b> .

# **XPATH SELECTORS – ATTRIBUTE SELECTORS**

# XPATH – ATTRIBUTE SELECTORS – PART 1/3

METHOD	EXAMPLE	SUMMARY
#id	<code>//*[<b>@id</b>='sendMessageButton']</code>	Locate <b>all</b> elements which contain an <b>id</b> of 'sendMessageButton'
#id	<code>//button[<b>@id</b>='sendMessageButton']</code>	Locate all <b>button</b> elements which contain an <b>id</b> of 'sendMessageButton'
.class	<code>//*[<b>@class</b>='form-control']</code>	Locate <b>all</b> elements which contain a <b>class</b> of 'form-control'

# **XPATH – DESCENDANT SELECTORS**

# XPATH SELECTORS – DESCENDANT SELECTORS

EXAMPLE	SUMMARY
<code>//h1</code>	Locates all elements which use a <b>h1</b> tag within the html DOM.
<code>//div//p</code>	Locates all elements which use a <b>p</b> tag which are housed within a <b>div</b> tagged element.
<code>//div//ul/li</code>	Locates all elements which use a <b>li</b> tag which are housed within a <b>ul</b> tagged element and the <b>ul</b> elements are housed within a <b>div</b> tagged element.
<code>//div//ul/li/a</code>	Locates all <b>a</b> tag elements which are a child of a <b>li</b> tagged element. The <b>li</b> tagged element is also a child of the <b>ul</b> tagged elements etc.
<code>//tr//td</code>	Locates all <b>td</b> tag elements which are descendants to <b>tr</b> tagged elements.
<code>//tr//*</code>	Locates all tagged elements which are descendants to a <b>tr</b> tagged elements.

# **XPATH SELECTORS – INTRODUCTION**



# XPATH SELECTORS – INTRODUCTION - PART 1/2

METHOD	EXAMPLE	SUMMARY
<code>\$x("selector")</code>	<code>\$x("//div")</code>	Enables you to locate XPath selectors via the browser console window (Chrome & Firefox).
Absolute XPath	<code>/html/body/div[1]/div[3]/form/input</code>	A direct way to locate element(s) and can be very brittle and fragile. Absolute selectors start with a <code>/</code> because the search starts at the root of the html DOM.
Relative XPath	<code>//button[@id='sendMessageButton']</code>	Less brittle and fragile, starts with <code>//</code> because Relative XPaths can search anywhere within the html DOM.

# XPATH SELECTORS – INTRODUCTION - PART 2/2

METHOD	EXAMPLE	SUMMARY
/	//ul/li/a	Locates a child element.
//	//*[@id="list"]//a	Locates a descendant element.

# **CSS SELECTORS – PSEUDO-CLASSES**

# COMBINATORS – PSEUDO-CLASSES – PART 3/3

METHOD	EXAMPLE	SUMMARY
:first-child	[class*='btn']:first-child	Locate the <b>first child</b> element which contains 'btn' within the <b>class attribute</b> .
:last-child	[class*='btn']:last-child	Locate the <b>last child</b> element which contains 'btn' within the <b>class attribute</b> .

# **XPATH SELECTORS – INDEXING SELECTORS**

# XPATH – INDEXING SELECTORS

EXAMPLE	SUMMARY
<code>//li[position()=2]</code>	Locate all <b>li</b> tagged elements which have a <b>positioning</b> of <b>2</b> .
<code>//li[position()&gt;2]</code>	Locate all <b>li</b> tagged elements which have a <b>positioning</b> of <b>greater</b> than <b>2</b> .
<code>//li[2]</code>	Locate all <b>li</b> tagged elements which have a <b>positioning</b> of <b>2</b> .

# **XPATH SELECTORS – OTHER THINGS**

# XPATH SELECTORS – OTHER THINGS - PART 3/3

METHOD	EXAMPLE	SUMMARY
Or logic	<code>//*[<b>@href</b> <b>or</b> <b>@aria-valuemax</b>]</code>	Locate <b>all elements</b> which contain the <b>attribute</b> of <b>href</b> or <b>aria-valuemax</b> .
Union (Joining results)	<code><b>//a</b> <b> </b> <b>//td</b></code>	Locate <b>all elements</b> which use the <b>a</b> tag or <b>td</b> tag.



# **XPATH SELECTORS – OTHER THINGS**

# XPATH SELECTORS – OTHER THINGS - PART 2/3

METHOD	EXAMPLE	SUMMARY
Arithmetic	<code>//div[@aria-valuenow &gt;74.00]</code>	Locate all <b>div</b> elements which contain an attribute of <b>aria-valuenow</b> and the attribute value should be <b>greater</b> than <b>74.00</b> .
Has children	<code>//ul[*]</code>	Locate <b>ul</b> tagged elements which have children.
Has children (specific)	<code>//ul[li]</code>	Locate <b>ul</b> tagged elements which have children which use a <b>li</b> tag.

# **XPATH SELECTORS – OTHER THINGS**

# XPATH SELECTORS – OTHER THINGS - PART 1/3

METHOD	EXAMPLE	SUMMARY
not	<code>//section[not(@id='portfolio')]</code>	Locate all <b>section</b> elements apart from the <b>section</b> element which has an <b>id</b> of <b>portfolio</b> .
Text match	<code>//a[text()='Contact']</code>	Locate all <b>a</b> tagged elements which have the text of ' <b>Contact</b> '.
Text match (substring)	<code>//a[contains(text(),'Play')]</code>	Locate all <b>a</b> tagged elements which contain the text of ' <b>Play</b> '.

# **XPATH SELECTORS – SIBLING(S) SELECTORS**

# XPATH - SIBLING(S) SELECTORS

EXAMPLE	SUMMARY
<code>//header/following-sibling::section</code>	Locate all <b>section</b> elements which are a <b>following sibling</b> to <b>header</b> element(s).
<code>//header/following-sibling::section[1]</code>	Locate the <b>first section</b> element(s) which are a <b>following sibling</b> to <b>header</b> element(s).
<code>//header/following-sibling::section[@id='portfolio']</code>	Locate all <b>section</b> elements which are a <b>following sibling</b> to <b>header</b> element(s) and the section element(s) must have an <b>id</b> attribute which has the value of <b>portfolio</b> .

# **XPATH SELECTORS – ORDER SELECTORS**

# XPATH – ORDER SELECTORS

EXAMPLE	SUMMARY
<code>//ul/li[1]</code>	Locate all of the <b>first list (li)</b> item elements.
<code>//ul/li[2]</code>	Locate all of the <b>second list (li)</b> items elements.
<code>//ul/li[last()]</code>	Locate all of the <b>last list (li)</b> items elements.
<code>//td[last()]</code>	Locate all of the <b>last td</b> elements.
<code>//div[@class="progress"]</code> <code>//div[@class="progress"][1]</code> <code>//div[@class="progress"][2]</code>	Locates all elements based upon tag type ( <b>div</b> ), attribute ( <b>class</b> ) value and index number(s).
<code>//td[1]</code>	Locate all of the <b>td</b> elements which have an <b>index</b> of <b>one</b> .