hillel компьютерная школа

Lection 22

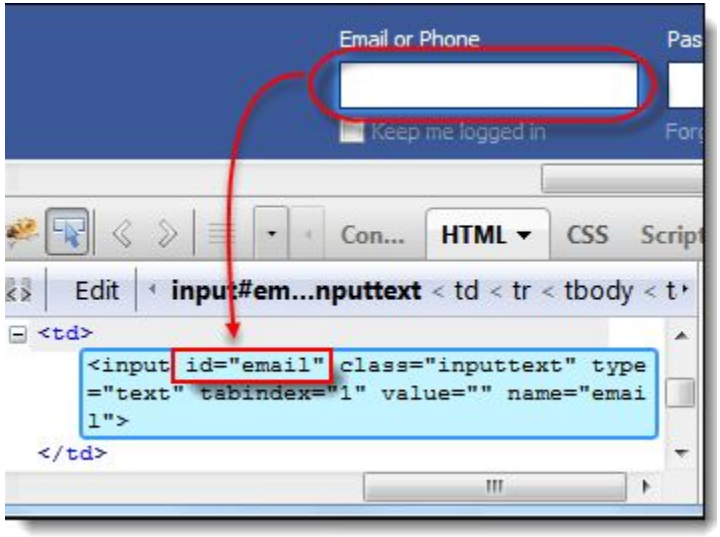
1. Selectors
2. Xpath & CSS

Element selection

Locator	Description
class name	Locates elements whose class name contains the search value (compound class names are not permitted)
css selector	Locates elements matching a CSS selector
id	Locates elements whose ID attribute matches the search value
name	Locates elements whose NAME attribute matches the search value
link text	Locates anchor elements whose visible text matches the search value
partial link text	Locates anchor elements whose visible text contains the search value. If multiple elements are matching, only the first one will be selected.
tag name	Locates elements whose tag name matches the search value
xpath	Locates elements matching an XPath expression

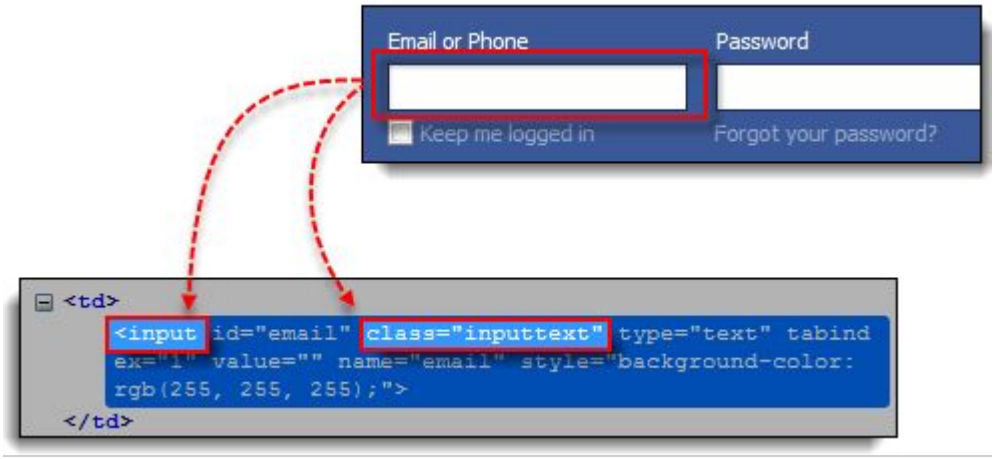
```
WebElement cheese = driver.findElement(By.id("cheese"));
```

Locating by ID



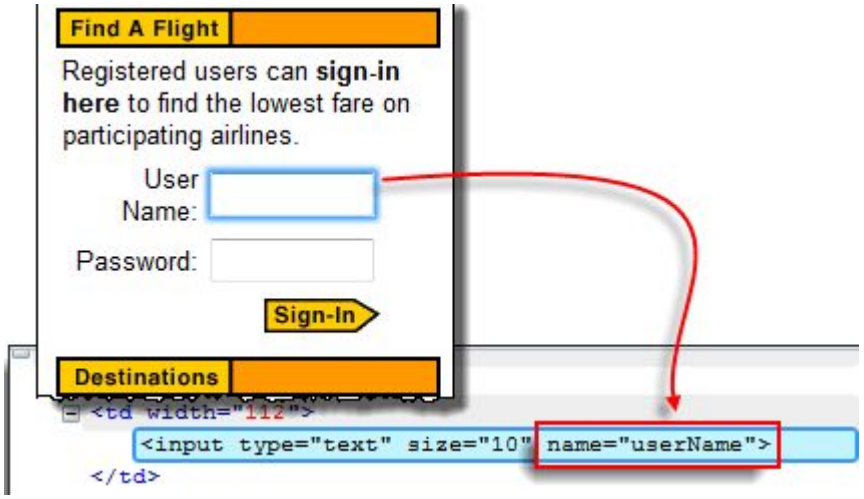

```
WebElement elem = driver.findElement(By.id("email"));
```

Locating by ClassName



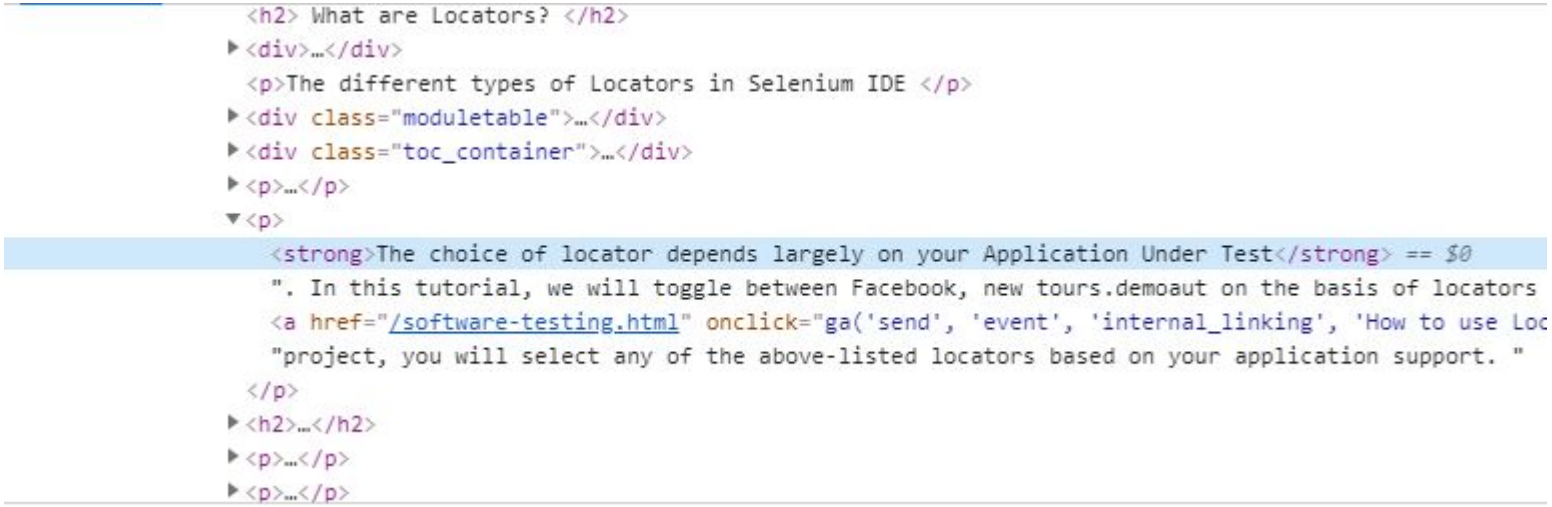

```
WebElement elem = driver.findElement(By.className("inputtext"));
```

Locating by Name



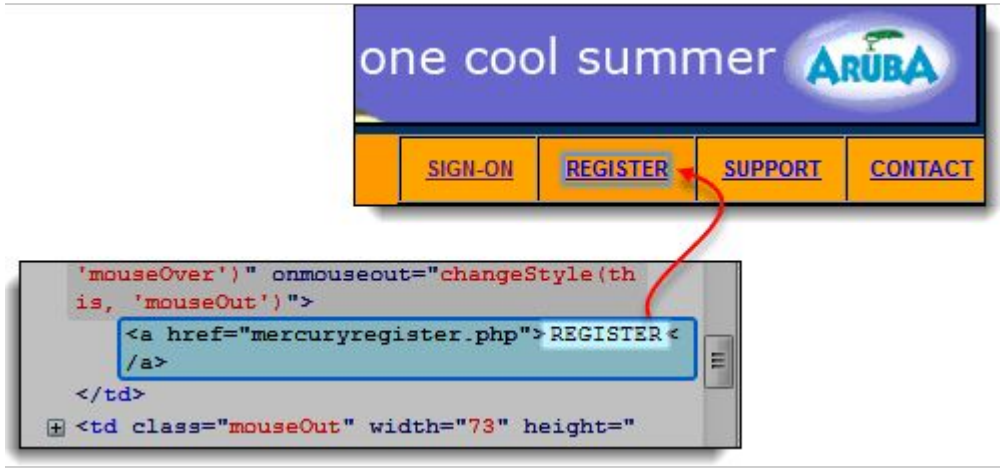

```
WebElement elem = driver.findElement(By.name("userName"));
```

Locating by TagName



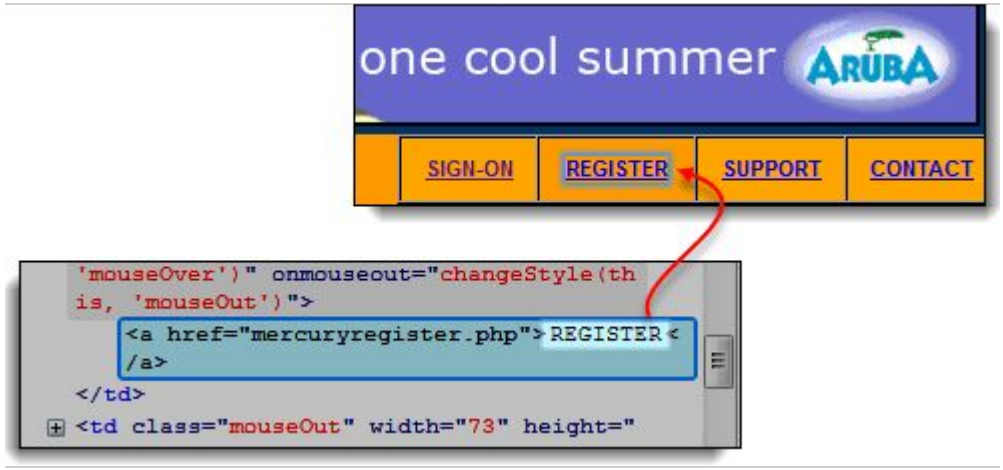

```
WebElement elem = driver.findElement(By.tagName("strong"));
```

Locating by Link text




```
WebElement elem = driver.findElement(By.linkText("REGISTER"));
```

Locating by Partial link text

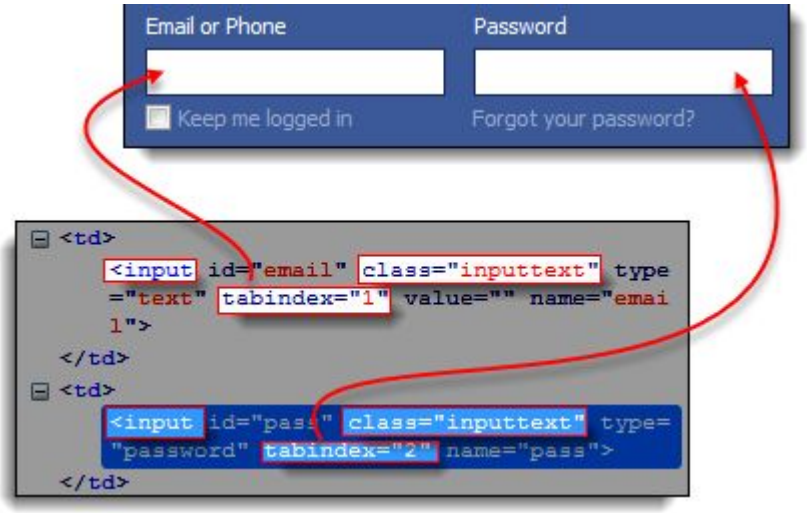



```
WebElement elem = driver.findElement(By.partialLinkText("STER"));
```

Locating by xpath


```
WebElement elem = driver.findElement(By.xpath("//img[@src='/images/hdr_right.gif']"));
```

Locating by cssSelector




```
WebElement elem = driver.findElement(By.cssSelector("input.inputtext[tabindex=1]"));
```

hillel компьютерная школа

Xpath

Xpath

Xpath - It is a syntax or language for finding any element on the web page using XML path expression. XPath is used to find the location of any element on a webpage using HTML DOM structure

```
Xpath=//tagname[@attribute='value']
```

- **//** : Select current node.
- **Tagname:** Tagname of the particular node.
- **@:** Select attribute.
- **Attribute:** Attribute name of the node.
- **Value:** Value of the attribute.

Syntax

Path Expression	Result
bookstore	Selects all nodes with the name "bookstore"
/bookstore	Selects the root element bookstore Note: If the path starts with a slash (/) it always represents an absolute path to an element!
bookstore/book	Selects all book elements that are children of bookstore
//book	Selects all book elements no matter where they are in the document
bookstore//book	Selects all book elements that are descendant of the bookstore element, no matter where they are under the bookstore element
//@lang	Selects all attributes that are named lang

Types of X-path

1) Absolute XPath

```
html/body/div[1]/section/div[1]/div/div/div/div[1]/div/div/div/div/div[3]/div[1]/div/h4[1]/b
```

2) Relative XPath. For Relative Xpath the path starts from the middle of the HTML DOM structure

```
Relative xpath: //*[@class='featured-box']//*[text()='Testing']
```

1) Basic XPath:


```
Xpath=//input[@name='uid']
Xpath=//input[@type='text']
Xpath= //label[@id='message23']
Xpath= //input[@value='RESET']
Xpath=//*[@class='barone']
Xpath=//a[@href='http://demo.guru99.com/']
Xpath= //img[@src='//cdn.guru99.com/images/home/java.png']
```

2) Contains():

```
Xpath=//*[contains(@type,'sub')]
Xpath=//*[contains(@name,'btn')]
Xpath=//*[contains(@id,'message')]
Xpath=//*[contains(text(),'here')]
Xpath=//*[contains(@href,'guru99.com')]
```

3) Using OR & AND:


```
Xpath=//*[@type='submit' or @name='btnReset']
Xpath=//input[@type='submit' and @name='btnLogin']
```

4) Starts-with function:

Id=" message12"

Id=" message345"

Id=" message0873"

Id=" message8769"

```
Xpath=//label[starts-with(@id,'message')]
```

5) Text():

```
Xpath=//td[text()='UserID']
```

6) XPath axes methods:

a) Following - Selects all elements in the document of the current node

```
Xpath=//*[@type='text']//following::input
Xpath=//*[@type='text']//following::input[1]
```

b) Ancestor - selects all ancestors element (grandparent, parent, etc.) of the current node

```
Xpath=//*[text()='Enterprise Testing']//ancestor::div
```

c) Child - Selects all children elements of the current node

```
Xpath=//*[@id='java_technologies']//child::li
```

d) Preceding - Select all nodes that come before the current node

```
Xpath=//*[@type='submit']//preceding::input
```

e) Following-sibling - Select the following siblings of the context node

```
xpath=//*[@type='submit']//following-sibling::input
```

f) Parent - Selects the parent of the current node

```
Xpath=//*[@id='rt-feature']//parent::div
```

g) Self - Selects the current node or 'self' means it indicates the node itself

```
Xpath =//*[@type='password']//self::input
```

h) Descendant - Selects the descendants of the current node

```
Xpath=//*[@id='rt-feature']//descendant::a
```

nodes: https://www.w3schools.com/xml/xpath_nodes.asp

Predicates

Path Expression	Result
/bookstore/book[1]	Selects the first book element that is the child of the bookstore element. Note: In IE 5,6,7,8,9 first node is[0], but according to W3C, it is [1]. To solve this problem in IE, set the SelectionLanguage to XPath: *In JavaScript: xml*.setProperty("SelectionLanguage","XPath");
/bookstore/book[last()]	Selects the last book element that is the child of the bookstore element
/bookstore/book[last()-1]	Selects the last but one book element that is the child of the bookstore element
/bookstore/book[position()<3]	Selects the first two book elements that are children of the bookstore element
//title[@lang]	Selects all the title elements that have an attribute named lang
//title[@lang='en']	Selects all the title elements that have a "lang" attribute with a value of "en"
/bookstore/book[price>35.00]	Selects all the book elements of the bookstore element that have a price element with a value greater than 35.00
/bookstore/book[price>35.00]/title	Selects all the title elements of the book elements of the bookstore element that have a price element with a value greater than 35.00

hillel компьютерная школа

CSS

Css selectors

When we don't have an option to choose Id or Name, we should prefer using CSS locators as the best alternative.

- CSS has more Advantage than Xpath
- CSS is much more faster and simpler than the Xpath.
- Looks shorter

```
tagName[attributename=attributeValue]

Example 1: input[id=email]
Example 2: input[name=email][type=text]
Example 3: input#email
Example 4: input[class=menu4]
Example 5: input.menu4
```

Css selector and classes


```
<button class="submit btn primary-btn flex-table-btn js-submit"
type="submit" style="background-color: rgb(85, 172, 238);">
 Log in
 </button>

WebElement ele1 = driver.findElement(By.cssSelector(".primary-btn"));
WebElement ele2 = driver.findElement(By.cssSelector(".btn.primary-btn"));
WebElement ele3 = driver.findElement(By.cssSelector(".submit.primary-btn"));
```

Special characters

1. '^' symbol, represents the starting text in a string.
2. '$' symbol represents the ending text in a string.
3. '*' symbol represents contains text in a string.

```
css=input[id^='ema']
```

```
css=input[id$='mail']
```

```
css=input[id*='mai']
```

Selector	Example	Example description
.class	.intro	Selects all elements with class="intro"
.class1.class2	.name1.name2	All elements with both *name1* and *name2* set within its class attribute
.class1 .class2	.name1 .name2	All elements with *name2* that is a descendant of element with *name1*
#id	#firstname	Selects the element with id="firstname"
*	*	Selects all elements
element	p	Selects all <p> elements
element.class	p.intro	Selects all <p> elements with class="intro"
element,element	div, p	Selects all elements and all <p> elements
element element	div p	Selects all <p> elements inside elements
element>element	div > p	Selects all <p> elements where the parent is a element
element+element	div + p	All <p> elements that are placed immediately after elements
element1~element2	p ~ ul	Selects every element that are preceded by a <p> element
[*attribute*]	[target]	Selects all elements with a target attribute
[*attribute=value*]	[target=_blank]	Selects all elements with target="_blank"
[*attribute~=value*]	[title~=flower]	Selects all elements with a title attribute containing the word "flower"
[*attribute|=value*]	[lang|=en]	Selects all elements with a lang attribute value starting with "en"
[*attribute^=value*]	a[href^="https"]	Every <a> element whose href attribute value begins with "https"
[*attribute$=value*]	a[href$=".pdf"]	Selects every <a> element whose href attribute value ends with ".pdf"
[*attribute*=value*]	a[href*="ols"]	Every <a> element whose href value contains the substring "ols"

:active	a:active	Selects the active link
::after	p::after	Insert something after the content of each <p> element
::before	p::before	Insert something before the content of each <p> element
:checked	input:checked	Selects every checked <input> element
:default	input:default	Selects the default <input> element
:disabled	input:disabled	Selects every disabled <input> element
:empty	p:empty	Selects every <p> element that has no children (including text nodes)
:enabled	input:enabled	Selects every enabled <input> element
:first-child	p:first-child	Selects every <p> element that is the first child of its parent
::first-letter	p::first-letter	Selects the first letter of every <p> element
::first-line	p::first-line	Selects the first line of every <p> element
:first-of-type	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
:focus	input:focus	Selects the input element which has focus
:hover	a:hover	Selects links on mouse over
:in-range	input:in-range	Selects input elements with a value within a specified range
:indeterminate	input:indeterminate	Selects input elements that are in an indeterminate state
:invalid	input:invalid	Selects all input elements with an invalid value
:lang(*language*)	p:lang(it)	Selects every <p> element with a lang attribute equal to "it" (Italian)
:last-child	p:last-child	Selects every <p> element that is the last child of its parent
:last-of-type	p:last-of-type	Selects every <p> element that is the last <p> element of its parent

:link	a:link	Selects all unvisited links
:not(*selector*)	:not(p)	Selects every element that is not a <p> element
:nth-child(*n*)	p:nth-child(2)	Selects every <p> element that is the second child of its parent
:nth-last-child(*n*)	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
:nth-last-of-type(*n*)	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
:nth-of-type(*n*)	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
:only-of-type	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
:only-child	p:only-child	Selects every <p> element that is the only child of its parent
:optional	input:optional	Selects input elements with no "required" attribute
:out-of-range	input:out-of-range	Selects input elements with a value outside a specified range
::placeholder	input::placeholder	Selects input elements with the "placeholder" attribute specified
:read-only	input:read-only	Selects input elements with the "readonly" attribute specified
:read-write	input:read-write	Selects input elements with the "readonly" attribute NOT specified
:required	input:required	Selects input elements with the "required" attribute specified
:root	:root	Selects the document's root element
::selection	::selection	Selects the portion of an element that is selected by a user
:target	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)
:valid	input:valid	Selects all input elements with a valid value
:visited	a:visited	Selects all visited links

xpath or css

hillel компьютерная школа

Description	Xpath	CSS Path
Direct Child	//div/a	div > a
Child or Subchild	//div//a	div a
Id	//div[@id='idValue']//a	div#idValue a
Class	//div[@class='classValue']//a	div.classValue a
Following Sibling	//ul/li[@class='first']/following-siblin	ul>li.first + li
Attribute	//form/input[@name='username']	form input[name='username']
Multiple Attributes	//input[@name='continue' and @typ	input[name='continue'][type='button'
nth Child	//ul[@id='list']/li[4]	ul#list li:nth-child(4)
First Child	//ul[@id='list']/li[1]	ul#list li:first-child
Last Child	//ul[@id='list']/li[last()]	ul#list li:last-child
Attribute Contains	//div[contains(@title,'Title')]	div[title*='Title"]
Attribute Starts With	//input[starts-with(@name,'user')]	input[name^='user"]
Attribute Ends With	//input[ends-with(@name,'name')]	input[name$='name"]
Element with Attribute	//div[@title]	div[title]

```
{
 table_header_id_and_class: {
  css: "table#large-table thead .column-50",
  xpath: "//table[@id='large-table']//thead//*[@class='column-50']"
 },
 table_header_id_class_and_direct_desc: {
  css: "table#large-table > thead .column-50",
  xpath: "//table[@id='large-table']/thead//*[@class='column-50']"
 },
 table_header_traversing: {
  css: "table#large-table thead tr th:nth-of-type(50)",
  xpath: "//table[@id='large-table']//thead//tr//th[50]"
 },
 table_header_traversing_and_direct_desc: {
  css: "table#large-table > thead > tr > th:nth-of-type(50)",
  xpath: "//table[@id='large-table']/thead/tr/th[50]"
 },
 table_cell_id_and_class: {
  css: "table#large-table tbody .column-50",
  xpath: "//table[@id='large-table']//tbody//*[@class='column-50']"
 },
 table_cell_id_class_and_direct_desc: {
  css: "table#large-table > tbody .column-50",
  xpath: "//table[@id='large-table']/tbody//*[@class='column-50']"
 },
 table_cell_traversing: {
  css: "table#large-table tbody tr td:nth-of-type(50)",
  xpath: "//table[@id='large-table']//tbody//tr//td[50]"
 },
 table_cell_traversing_and_direct_desc: {
  css: "table#large-table > tbody > tr > td:nth-of-type(50)",
  xpath: "//table[@id='large-table']/tbody/tr/td[50]"
 }
}
```

Xpath vs Css

XPath we can traverse both forward and backward

Any set of conditions for the nodes in the path

Queries return any number of results, including zero

Xpath engines are different in each browser, hence make them inconsistent

Easy to read and write

Faster

hillel компьютерная школа

Practice