Contents

[Locators 1](#_Toc465333240)

[Identifier 1](#_Toc465333241)

[Id 1](#_Toc465333242)

[Name 1](#_Toc465333243)

[DOM (Document Object Mode) 1](#_Toc465333244)

[xPath 1](#_Toc465333245)

[CSS 1](#_Toc465333246)

[CSS vs xPath 2](#_Toc465333247)

[References 4](#_Toc465333248)

# Locators

## Identifier

Id or name (based on field), in source code example:

<div id=“login”>

<input name=“username” type=“text” />

<input name=“password” type=“password” />

<input name=“submit” type=“submit” value=“Login” />

</div>

Valid locators:

* Login;
* username;
* submit

## Id

Id, source code example:

<div id=“login”>

## Name

Name, source code example:

<input name=“username” type=“text” />

## xPath

xPath locators:

* //div[@id=login] – finds any div with id “login on page”
* //div/input[@name=’username’] – finds input with name “username”, which is under parent with tag div.

More examples below in xPath vs CSS table.

## CSS

css locators:

* div#login – finds any div with id “login on page”
* div > input[name=’username’] – finds input with name “username”, which is under parent with tag div.

More examples below in xPath vs CSS table.

## DOM (Document Object Mode)

JavaScript locator in document[[1]](#footnote-1):

* document.getElementById(‘login’) is 1 element with id login
* document.getElementsByName(‘username’) is list of all elements with name=”username”
* document.getElementsByTagName('input') is a list of all elements with tag input (the <input />)
* document.getElementsByClassName(‘style1’) is list of all elements with class=”style1’”
* document.querySelector(‘#login’) is 1 element, which is found via css selector (for more information on how css selector is written see above), in this case 1 element with id ‘login’.
* document.querySelectorAll('a') is all element, which is found via css selector (for more information on how css selector is written see above), in this case elements with a tag

# CSS vs xPath

|  |  |  |
| --- | --- | --- |
| Goal | CSS | xPath |
| All Elements | \* | //\* |
| All P Elements | p | //p |
| All Child Elements | p > \* | //p/\* |
| Direct child | div > a | //div/a |
| Child or sub child | div a | //div//a |
| Any parent of element (e.g. from <div><a></div>) | a:parent[[2]](#footnote-2) | //a/..  //a/parent::\* |
| Specific parent of element | span:has(a) [[3]](#footnote-3) | //a/parent::span |
| Element By ID | #foo | //\*[@id=’foo’] |
|  | div#example a | //div[@id='example']//a |
| Element By Class | .foo | //\*[contains(@class,’foo’)][[4]](#footnote-4) |
|  | div.example a | //div[@class='example']//a |
| Element With Attribute | \*[title] | //\*[@title] |
|  | form input[name='username'] | //form//input[name='username'] |
|  | a[href='url'] | //a[@href='url'] |
| … (name=”continue” and type=”button”) | input[name='continue'][type='button'] | //input[@name='continue' and @type=' button'] |
| First Child of All P | p > \*:first-child | //p/\*[0] |
| Nth-element (fourth li element in list) | ul#recordlist li:nth-of-type(4) | //ul[@id=’recordlist’]//li[4] |
| … (fourth element only if it is a li element) | ul#recordlist li:nth-child(4) |  |
| … (any kind of fourth element) | ul#recordlist \*:nth-child(4) |  |
| All P with an A child | *Not possible* | //p[a] |
| Next Element/sibling | p + \* | //p/following-sibling::\*[0] |
| Next Element/sibling (after input field “username”) | form input.username + input |  |
| A link with an “id” that starts with the text “id\_prefix\_” | a[id^='id\_prefix\_'] | //a[starts-with(@id, 'id\_prefix\_')] |
| A link with an “id” that ends with the text “\_id\_sufix” | a[id$='\_id\_sufix'] | Not supported in browsers |
| A link with an “id” that contains the text “id\_pattern” | a[id\*='id\_pattern'] | //a[contains(@id, 'id\_prefix\_')] |
| Matching by inner text (exact) |  | //a[.='Find me I have nothing in me!!']  //a[text()='Find me I have nothing in me!!'] |
| Matching by inner text | a:contains('Forgot')[[5]](#footnote-5) | //a[contains(text(),'Forgot')]  //a[contains(.,'Forgot')] |

# References

<https://newcircle.com/bookshelf/selenium_tutorial/locators>

<http://ejohn.org/blog/xpath-css-selectors/#class>

<https://saucelabs.com/resources/articles/selenium-tips-css-selectors>

<https://seleniumeasy.com/selenium-tutorials/examples-for-xpath-and-css-selectors>

<https://www.simple-talk.com/dotnet/.net-framework/xpath,-css,-dom-and-selenium-the-rosetta-stone/>

1. DOM unlike css and xpath, which can be checked via tab with html in browser debugger, can be checked via browser console. [↑](#footnote-ref-1)
2. Works in the IDE, but not in Selenium & JAVA [↑](#footnote-ref-2)
3. Works in the IDE, but not in Selenium & JAVA [↑](#footnote-ref-3)
4. This isn’t right due to the fact that it would match ‘foobar’ and ‘foo bar’, when only the second pattern is correct. The actual syntax would be far more complex and would probably require multiple expressions to get the job done. [↑](#footnote-ref-4)
5. Works in the IDE, but not in Selenium & JAVA [↑](#footnote-ref-5)