Capgemini

**Css Selectors in Selenium - Tutorial**

**Information about this document**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project name** |  | | |
| **Authors** | | | |
| **Name** | **Role** | **E-mail** | |
| Cezary Siwiak | developer | [cezary.siwiak@capgemini.com](mailto:cezary.siwiak@capgemini.com) | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
| **History** | | | |
| **Name** | **Role** | **E-mail** | **Version** |
| Cezary Siwiak | developer | [cezary.siwiak@capgemini.com](mailto:cezary.siwiak@capgemini.com) | 1.0 – creation |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[Introduction 4](#_Toc419820374)

[HTML page that we will use for learning 5](#_Toc419820375)

[Select element with specific css class 6](#_Toc419820376)

[Example 1 6](#_Toc419820377)

[Example 2 7](#_Toc419820378)

[Select element by # 8](#_Toc419820379)

[Wildcards 9](#_Toc419820380)

[Starts with ^= 9](#_Toc419820381)

[Ends with $= 10](#_Toc419820382)

[Contains \*= 10](#_Toc419820383)

[Search by several css classes 11](#_Toc419820384)

[Search by several properties 12](#_Toc419820385)

[Find child 13](#_Toc419820386)

[Find one of several children’s - nth-child() 14](#_Toc419820387)

[Find one of several children’s with specific type- nth-of-type() 15](#_Toc419820388)

[How to find column number in table 16](#_Toc419820389)

[Dot notation 17](#_Toc419820390)

[Search first neighbor + 18](#_Toc419820391)

[Direct child > 19](#_Toc419820392)

[Limitations. 20](#_Toc419820393)

# Introduction

This document should help you in learning how to use css selectors in Selenium.

Also test application was created. Reader can use it as help. Reader can modify this application and use for self learning and testing.

Application is present in Capgemini network: [\\wrofs1.sdm.de\Transfer\csiwiak\selenium](file:///\\wrofs1.sdm.de\Transfer\csiwiak\selenium)

# HTML page that we will use for learning

|  |
| --- |
| <head>  <title>Page Title</title>  </head>  <body>    <h1> My First Heading </h1>  <div>logout</div>  <div>  <p> My first paragraph </p>  </div>  <div class=*"aaa"*> Second div </div>    <div class=*"bbb"*>  <input name=*"a\_search\_field\_1"* value=*""* size=*"15"* maxlength=*"128"* class=*"form-text"* type=*"text"*>  <input name=*"b\_search\_field\_2"* value=*""* size=*"15"* maxlength=*"128"* class=*"form-text"* type=*"text"*>  <input name=*"c\_search\_field\_3"* value=*""* size=*"15"* maxlength=*"128"* class=*"form-text"* type=*"text"*>  <input name=*"d\_search\_field\_4"* value=*""* size=*"15"* maxlength=*"128"* class=*"form-text"* type=*"text"*>  </div>  <div class=*"ccc ddd eee"*>  <p> paragraph0 </p>  <p class=*"efgh"*> paragraph1 </p>  <p> paragraph2 </p>  <p class=*"ijkl"*>  <b> bold </b>  <h1> header 1 </h1>  </p>  <h1> header 2</h1>  </div>    <div class=*"radios"*>  <div class=*"item"*>  <input id=*"item-all"* name=*"group1"* class=*"form-radio"* type=*"radio"* checked=*"checked"*>  <label class=*"option"* for=*"item-all"*>All</label>  </div>  <div class=*"item"*>  <input id=*"item-modules"* name=*"group1"* class=*"form-radio"* type=*"radio"*>  <label class=*"option"* for=*"item-modules"*>Modules</label>  </div>  <div class=*"item"*>  <input id=*"item-themes"* name=*"group1"* class=*"form-radio"* type=*"radio"*>  <label class=*"option"* for=*"item-themes"*>Themes</label>  </div>  <div class=*"item"*>  <input id=*"item-doc"* name=*"group1"* class=*"form-radio"* type=*"radio"*>  <label class=*"option doc-css"* for=*"item-doc"*>Documentation</label>  </div>  <div class=*"item"*>  <input id=*"item-summary"* name=*"group1"* class=*"form-radio"* type=*"radio"*>  <label class=*"option doc-css"* for=*"item-summary"*>Summary</label>  </div>  </div>  <div class=*"parent"*>  <div class=*"inside-1"*>  <div class=*"inside-2"*>  <ul class=*"menu"*>  <p>element 1 p </p>  <li>element 2 li </li>  <li>element 3 li </li>  <p>element 4 p </p>  </ul>  </div>  </div>  </div>  <table class=*"table-style-css"* style="width:*180px*;border: *1px solid black*;">  <tr>  <th>First\_Name</th>  <th>Last\_Name</th>  <th>Age</th>  </tr>  <tr>  <td>Jill</td>  <td>Smith</td>  <td>50</td>  </tr>  <tr>  <td>Eve</td>  <td>Jackson</td>  <td>94</td>  </tr>  </table>  </body> |

# Select element with specific css class

## Example 1

Selector starts by tag, next in square brackets we should write properties name and value.

**tag[propertiesName=’value’]**

We can alsow use dot notation if we like to search tag by class.

**Html**

|  |
| --- |
| <body>  **<div class="aaa"> some text </div>**  </body> |

**Css selector**

|  |
| --- |
| **div[class='aaa'] - search for div with properties class=’aaa’**  **div.aaa - the same search but with dot notation. After dot we put class name** |

**Java example**

|  |
| --- |
| **WebElement element = driver.findElement(By.cssSelector("div[class='aaa']"))**  **WebElement element = driver.findElement(By.cssSelector("div.aaa"))** |

## Example 2

We can find all tags we want with all properties. In next example we search for *input* tag with properties

*name="b\_search\_field\_2"*.

**Html**

|  |
| --- |
| <div class=*"bbb"*>  <input name=*"a\_search\_field\_1"* value=*""* size=*"15"* maxlength=*"128"* class=*"form-text"* type=*"text"*>  <*input name="b\_search\_field\_2"* value="" size="15" maxlength="128" class="form-text" type="text">  <input name=*"c\_search\_field\_3"* value=*""* size=*"15"* maxlength=*"128"* class=*"form-text"* type=*"text"*>  <input name=*"d\_search\_field\_4"* value=*""* size=*"15"* maxlength=*"128"* class=*"form-text"* type=*"text"*>  </div> |

**Css selector**

|  |
| --- |
| **input[name='b\_search\_field\_2'] – search tag input with properties name='b\_search\_field\_2'** |

**Java example**

|  |
| --- |
| **WebElement element = driver.findElement(By.cssSelector("input[name='b\_search\_field\_2']"));** |

# Select element by #

If we search by id we can also use #

**tag#someId**

**Html**

|  |
| --- |
| <body>  **<div id="someId"> some text </div>**  </body> |

**Css selector**

|  |
| --- |
| **div#someId - search for div with id someId** |

**Java example**

|  |
| --- |
| **WebElement element = driver.findElement(By.cssSelector("div#someId"))** |

# Wildcards

We can search tag by properties value that has specific chars at the beginning or end. Or contains specific chars

**Html**

|  |
| --- |
| <div class=*"bbb"*>  <input name=*"a\_search\_field\_1"* value=*""* size=*"15"* maxlength=*"128"* class=*"form-text"* type=*"text"*>  <input name="*b\_search\_field\_2*" value="" size="15" maxlength="128" class="form-text" type="text">  <input name=*"c\_search\_field\_3"* value=*""* size=*"15"* maxlength=*"128"* class=*"form-text"* type=*"text"*>  <input name=*"d\_search\_field\_4"* value=*""* size=*"15"* maxlength=*"128"* class=*"form-text"* type=*"text"*>  </div> |

## Starts with ^=

Use **^=** to find tag with properties value that starts with chars **'d\_search'**

**Css selector**

|  |
| --- |
| **input[name^='d\_search'] – search tag input with properties value starts by chars -'d\_search'** |

**Java example**

|  |
| --- |
| **WebElement element = driver.findElement(By.cssSelector("input[name^='d\_search']"));** |

## Ends with $=

Use **$=** to find tag with properties value that ends with **'field\_3'**

**Css selector**

|  |
| --- |
| **input[name$='field\_3'] – search input with properties ends with 'field\_3'** |

**Java example**

|  |
| --- |
| **WebElement element = driver.findElement(By.cssSelector("input[name$='field\_3']"));** |

## Contains \*=

Use **\*=** to find tag with properties value that contains chars - **'search'**

**Css selector**

|  |
| --- |
| **input[name\*='search'] – search input with properties that contains chars 'search'** |

**Java example**

|  |
| --- |
| **List<WebElement> elements = driver.findElements(By.cssSelector("input[name\*='search']"));** |

# Search by several css classes

We can search tags with several class values separated by space. All class values have to be enumerated in selector. To search only by one of classes you have use dot notation.

**Html**

|  |
| --- |
| <div class=*"radios"*>  <div class=*"item"*>  <input id=*"item-doc"* name=*"group1"* class=*"form-radio"* type=*"radio"*>  <label class="option doc-css" for=*"item-doc"*>Documentation</label>  </div>  </div> |

**Css selector**

|  |
| --- |
| **label[class='option doc-css']– search label with properties class with two values 'option’ and ‘doc-css'** |

**Java example**

|  |
| --- |
| **WebElement element = driver.findElement(By.cssSelector("label[class='option doc-css']"));** |

# Search by several properties

We can search tags by several properties. This example shows how to find *label* with properties *class* and *for*

**Html**

|  |
| --- |
| <div class=*"item"*>  <input id=*"item-summary"* name=*"group1"* class=*"form-radio"* type=*"radio"*>  <label class=*"option doc-css"* for=*"item-summary"*>Summary</label>  </div> |

**Css selector**

|  |
| --- |
| **label[class='option doc-css'][for='item-summary'] – search label with properties class and for** |

**Java example**

|  |
| --- |
| **WebElement element = driver.findElement(By.cssSelector("label[class='option doc-css'][for='item-summary']"));** |

# Find child

We can search child tag by writing all elements in html tree, separated by space. In example bellow we are looking for tag p which has parent *div* which has parent *body.*

**Html**

|  |
| --- |
| <body>  <div>  <p> My first paragraph </p>  </div>  </body> |

**Css selector**

|  |
| --- |
| **body div p – search body, then search child div, then search child p** |

**Java example**

|  |
| --- |
| **WebElement element = driver.findElement(By.cssSelector("body div p"));** |

# Find one of several children’s - nth-child()

In last example there was only one child. But parent can have several children’s. What we can do to select only correct child? In this example we search for second child.

**Html**

|  |
| --- |
| <div class=*"parent"*>  <div class=*"inside-1"*>  <div class=*"inside-2"*>  <ul class=*"menu"*>  <p>element 1 p </p>  <li>element 2 li </li>  <li>element 3 li </li>  <p>element 4 p </p>  </ul>  </div>  </div>  </div> |

**Css selector**

|  |
| --- |
| **ul[class='menu'] li:nth-child(2) – search ul with specific class, then search for second child, which should be li** |

**Java example**

|  |
| --- |
| **WebElement element = driver.findElement(By.cssSelector("ul[class='menu'] li:nth-child(2)"));** |

# Find one of several children’s with specific type- nth-of-type()

If we like to find second element of type li, then we can do it like bellow. Selected child is underlined.

**Html**

|  |
| --- |
| <div class=*"parent"*>  <div class=*"inside-1"*>  <div class=*"inside-2"*>  <ul class=*"menu"*>  <p>element 1 p </p>  <li>element 2 li </li>  <li>element 3 li </li>  <p>element 4 p </p>  </ul>  </div>  </div>  </div> |

**Css selector**

|  |
| --- |
| **ul[class='menu'] li:nth-of-type(2)– search ul with specific class, then search second li child** |

**Java example**

|  |
| --- |
| **WebElement element = driver.findElement(By.cssSelector("ul[class='menu'] li:nth-of-type(2)"));** |

# How to find column number in table

We like to find column number where header is ‘Age‘. First we will find all th tags. Next in java code we will iterate by those tags and search one with text ‘Age’

**Html**

|  |
| --- |
| <table class=*"table-style-css"* style="width:*180px*;border: *1px solid black*;">  <tr>  <th>First\_Name</th>  <th>Last\_Nazwisko</th>  <th>Age</th>  </tr>  <tr>  <td>Jill</td>  <td>Smith</td>  <td>50</td>  </tr>  <tr>  <td>Eve</td>  <td>Jackson</td>  <td>94</td>  </tr>  </table> |

**Css selector**

|  |
| --- |
| **table[class='table-style-css'] tr th – search table with specific class, then search tr, then select all th** |

**Java example**

|  |
| --- |
| **List<WebElement> elementsTh = driver.findElements(By.*cssSelector*("table[class='table-style-css'] tr th"));**  **int count=1;**  **int colum=-1;**  **for (WebElement webElement : elementsTh) {**  **if (webElement.getText().equals("Age")){**  **colum = count;**  **break;**  **}**  **count++;**  **}** |

# Dot notation

You can use dot notation instead square brackets. Advantage of dot notation is that you do not have to write all classes present in element. As you can see below you can search label only by one class even when it has two. Bellow you can see how you can write selector on different ways.

**Html**

|  |
| --- |
| <div class=*"radios"*>  <div class=*"item"*>  <input id=*"item-doc"* name=*"group1"* class=*"form-radio"* type=*"radio"*>  <label class="option doc-css" for=*"item-doc"*>Documentation</label>  </div>  </div> |

**Dot notation, two classes for label**

|  |
| --- |
| **div.radios div.item label.option.doc-css** |

**Dot notation, one class for label**

|  |
| --- |
| **div.radios div.item label.doc-css** |

**Bracket notation**

|  |
| --- |
| **div[class='radios'] div[class='item'] label[class='option doc-css']** |

**Dot and bracket notation mixed**

|  |
| --- |
| **div[class='radios'] div[class='item'] label.doc-css** |

# Search first neighbor +

If you like to search neighbor (but not children) of element then use + . In this example we are looking for h1 with text ‘header 2’

**Html**

|  |
| --- |
| <div class=*"ccc ddd eee"*>  <p class=*"efgh"*> paragraph2 </p>  <p class=*"ijkl"*>  <h1> header 1 </h1>  </p>  <h1> header 2</h1>  </div> |

**Css selector**

|  |
| --- |
| **div.eee p.ijkl + h1 – search div with class eee, then search p with class ijkl, then first neighbor h1** |

**Java example**

|  |
| --- |
| **WebElement element = driver.findElement(By.cssSelector("div.eee p.ijkl + h1"));** |

# Direct child >

To understand difference between child and direct child lets use example bellow. Second example will not work because b is not direct child of div.

**Html**

|  |
| --- |
| <div class=*"ccc ddd eee"*>  <p> paragraph2 </p>  <p class=*"ijkl"*>  <b> bold </b>  </p>  </div> |

**Css selector that works**

|  |
| --- |
| **div.ccc b – search div with class ccc, then search b (even if b is not direct child)** |

**Css selector that will failed**

|  |
| --- |
| **div.ccc > b – search div with class ccc, then search direct child b** |

**Css selector corrected.**

|  |
| --- |
| **div.ccc p > b – search div with class ccc, then search p, then search direct child b.** |

# Limitations.

Selectors have important limitations:

* It cannot search for parent.
* It is not possible to search element by inner text