



Writing CSS Selectors: the easy way

Corina Pip

@imalittletester

<https://imalittletester.com/>

<https://github.com/iamalittletester>

The principle

- Getting a handle to HTML elements --> used for interacting with the page
- Find unique path to the HTML element
- Find the shortest such path

@FindBy

- @FindBy – used for storing selector result to
 - WebElement (variable)
 - List<WebElement> (list of variables)
- Stored in separate class: PageObject class
 - Allows modelling of pages or modules

```
package tutorialsolution.pages;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import java.util.List;

public class BasicPage {
    @FindBy(css = "#idModuleImage") public WebElement elementWithId;
    @FindBy(css = ".w3-btn") public WebElement elementWithSingleClass;
    @FindBy(css = "h2") public WebElement h2Element;
    @FindBy(css = "#relativeModule
button") public List<WebElement> allButtonsInRelativeModule;
... }
```

The test

```
public class CSSSelectorsTest {  
    private BrowserGetter browserGetter = new BrowserGetter();  
    private BasicPage page;  
    private WebDriver driver;  
  
    @BeforeAll  
    public void beforeAll() {  
        driver = browserGetter.getChromeDriver();  
        page = PageFactory.initElements(driver,  
BasicPage.class);  
        ...  
    }  
}
```




SELECTOR TYPES

ID, CLASS, TAG, ATTRIBUTE

HTML elements

`<tagName attribute=value attribute ...>`

Button

`<button id='buttonId' ...>`

Button

`<button class='buttonClass' ...>`

Link

``

Image

``

Id – should be unique on page (should)

- HTML:

```

```

- Selector: #idModuleImage

- WebElement:

```
@FindBy(css = "#idModuleImage") public WebElement elementWithId;
```


Class – one class gives uniqueness; several can be present

- HTML:

```
<button class="w3-btn">One class</button>
```

- Selector: .w3-btn

- WebElement – first (or only) item with specified selector:

```
@FindBy(css = ".w3-btn") public WebElement elementWithClass;
```

- List<WebElement> - all elements with specified selector:

```
@FindBy(css = ".w3-btn") public List<WebElement>  
allElementsWithSingleClass;
```

Combination of classes is unique

- HTML:

```
<button class="w3-btn w3-padding w3-border w3-sand">Many  
classes</button>
```

- Selector: `.w3-btn.w3-padding.w3-border.w3-sand`

- WebElement – first (or only) item with specified selector:

```
@FindBy(css = ".w3-btn.w3-padding.w3-border.w3-sand")  
public WebElement elementWithMultipleClasses;
```

Tag name

- HTML:

```
<h2>This is a H2 text</h2>
```

- Selector: h2

- WebElement – first (or only) item with specified selector:

```
@FindBy(css = "h2") public WebElement h2Element;
```

Attribute value equals

- HTML:

```

```

- Selector: `[width='189']`
- WebElement – first (or only) item with specified selector:
`@FindBy(css = "[width='189']") public WebElement
imageWithSpecifiedWidth;`

Attribute value contains

- HTML:

```

```

- Selector: `[src*='1878']`
- WebElement – first (or only) item with specified selector:
`@FindBy(css = "[src*='1878']") public WebElement
imageWithSrcContains;`



SELECTOR STRATEGY

Siblings

CHARLIE
(BRAVO's parent)

BRAVO
(ALPHA's parent)

SIBLING

ALPHA

SIBLING

SIBLING

`<a ...>`

`<h3>...`

`
`

ALPHA

`<div>...`

``

`<div ...>`

`<div>...`

`<div>`

ALPHA

`<div>...`

`</div>`

2 levels, no siblings

```
<div id="level2">  
  <h3>ALPHA</h3>  
</div>
```

```
@FindBy(css = "#level2 h3")  
public WebElement level2;
```

2 levels, different tag siblings

```
<div id="level2SiblingsDiffTags">  
  <h4>Sibling</h4>  
  <h3>ALPHA</h3>  
  <div>Sibling</div>  
  <br>  
</div>
```

```
@FindBy(css = "#level2SiblingsDiffTags h3")  
public WebElement level2SiblingsDiffTags;
```


2 levels, same tag siblings

```
<div id="level2SiblingsSameTag">  
  <h3>Sibling</h3>  
  <h3>ALPHA</h3>  
  <h3>Sibling</h3>  
</div>
```

```
@FindBy(css="#level2SiblingsSameTag h3")  
public List<WebElement> level2SiblingsSameTag;  
level2SiblingsSameTag.get(1)
```


2 levels, same tag siblings, ALPHA has class

```
<div id="level2SiblingsSameTagAndClass">  
  <div>Sibling</div>  
  <div class="someClass">ALPHA</div>  
  <div>Sibling</div>  
</div>
```

```
@FindBy(css = "#level2SiblingsSameTagAndClass .someClass")  
public WebElement level2SiblingsSameTagAndAlphaHasClass;
```

Particular 2 level cases – unordered list (UL)

```
<ul id="BRAVOUL">  
  <li>ZULU</li>  
  <li>YANKEE</li>  
  <li>ALPHA</li>  
</ul>
```

```
@FindBy(css = "#BRAVOUL li")  
public List<WebElement> unorderedList;  
unorderedList.get(2)
```

Particular 2 level cases – ordered list (OL)

```
<ol id="BRAVOOL">  
  <li>ZULU</li>  
  <li>YANKEE</li>  
  <li>ALPHA</li>  
</ol>
```

```
@FindBy(css = "#BRAVOOL li")  
public List<WebElement> orderedList;  
orderedList.get(2)
```

3 levels – ALPHA != BRAVO

```
<div id="level3">  
  <div>  
    <h3>ALPHA</h3>  
  </div>  
</div>
```

```
@FindBy(css = "#level3 h3") public WebElement level3;
```

3 levels – ALPHA != BRAVO

```
<div id="level3">  
  <div>  
    <h3>ALPHA</h3>  
  </div>  
</div>
```

```
@FindBy(css = "#level3 h3") public WebElement level3;
```


3 levels – ALPHA = BRAVO, ALPHA has class

```
<div id="level3AEqualsBAHasClass" class="w3-blue-gray">  
  <div>  
    BRAVO  
    <div class="w3-red">ALPHA</div>  
  </div>  
</div>
```

```
@FindBy(css = "#level3AEqualsBAHasClass .w3-red")  
public WebElement level3AlphaEqualsBravoAlphaHasClass;
```

3 levels – one BRAVO, ALPHA has siblings

```
<div id="level3OneBravoSeveralAlphas">
```

```
  <div>
```

```
    <h3>ZULU</h3>
```

```
    <h3>ALPHA</h3>
```

```
    <h3>YANKEE</h3>
```

```
  </div>
```

```
</div>
```

```
@FindBy(css = "#level3OneBravoSeveralAlphas h3")
```

```
public List<WebElement> level3OneBravoSeveralAlphas;
```

```
level3OneBravoSeveralAlphas.get(1)
```

3 levels – BRAVO siblings equal, BRAVO children equal to ALPHA

```
<div id="level3BravoldenticalSiblings">  
  <div>  
    <h3>ZULU</h3></div>  
  <div>  
    <h3>ALPHA</h3></div>  
  <div>  
    <h3>YANKEE</h3></div>  
</div>
```

3 levels – BRAVO siblings equal, BRAVO children equal to ALPHA

```
@FindBy(css = "#level3BravoldenticalSiblings h3")  
public List<WebElement> level3BravoldenticalSiblings;
```

level3BravoldenticalSiblings.get(1) - ALPHA

level3BravoldenticalSiblings.get(0) - ZULU

level3BravoldenticalSiblings.get(2) - YANKEE

Level3BravoldenticalSiblings.size() - 3

3 levels – BRAVO siblings equal, BRAVO children equal to ALPHA

```
<div id="level3AllIdenticalSiblings">  
  <div>  
    <div>ZULU</div></div>  
  <div>  
    <div>ALPHA</div></div>  
  <div>  
    <div>YANKEE</div></div>  
</div>
```


3 levels – BRAVO siblings equal, BRAVO children equal to BRAVO, ALPHA=BRAVO

```
@FindBy(css = "#level3AllIdenticalSiblings div div")
```

```
public List<WebElement> level3AllIdentical;
```

```
level3AllIdentical.get(1) - ALPHA
```

```
level3AllIdentical.get(0) - ZULU
```

```
level3AllIdentical.get(2) - YANKEE
```

```
level3AllIdentical.size() - 3
```

Let's go to N levels

```
<div id="levelN">
```

```
  <div>
```

```
    <div>
```

```
      <div>
```

```
        <div>
```

```
.....
```

```
          <h3>ALPHA</h3>
```

```
        </div></div></div></div>...
```

```
@FindBy(css= "#levelN h3")  
public WebElement levelNH3;
```

Let's go to N levels

```
<div id="levelN">  
  <div>  
    <div>  
      <div>  
        ....  
        <span> (or <div class=...>  
          <div>ALPHA</div>  
</div></div></div></div>...
```

```
@FindBy(css= "#levelN span div")  
public WebElement levelNdiv;
```

Element in list – with random position

- Element is part of a list. We don't know the position of the desired WebElement in the list
- We know the text of the element
- A `List<WebElement>` is used and it is iterated
- If, in the iteration, the current WebElement has the desired text (`getText()`), we return it and use it

Element in list – with random position

```
@FindBy(css = "...") public List<WebElement> elements;

for (WebElement element : elements) {
    if (element.getText().equals(expectedText)
        { ... }
}
```

Or for this

```
<div>ZULU</div>  
<div>YANKEE</div>  
<div>CHARLIE</div>  
<div>BRAVO</div>  
<div>ALPHA</div>
```

```
@FindBy(css = "div")  
public List<WebElement> elements;  
for (WebElement element : elements) {  
    if (element.getText().equals("ALPHA"))  
        { ... }  
}
```




Twitter: **@imalittletester**

Blog: <https://imalittletester.com/>

GitHub: <https://github.com/iamalittletester>

Comics: <https://imalittletester.com/category/comics>

Examples from this webinar can be found here:

- the tests: <https://bit.ly/31w6fAA>
- the HTML: <https://bit.ly/2XgLaLc>