**What is a Web Table in Selenium?**

A **Web Table** in Selenium is a WebElement used for the tabular representation of data or information. The data or information displayed can be either static or dynamic. Web table and its elements can be accessed using WebElement functions and locators in Selenium. A typical example of a web table would be product specifications displayed on an eCommerce platform.

**Reading a HTML Web Table**

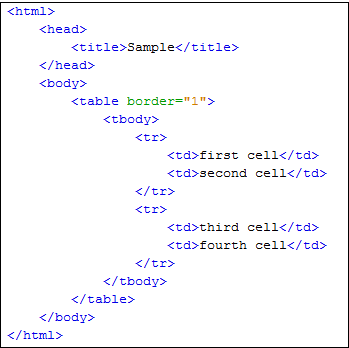
There are times when we need to access elements (usually texts) that are within HTML tables. However, it is very seldom for a web designer to provide an id or name attribute to a certain cell in the table. Therefore, we cannot use the usual methods such as “By.id()”, “By.name()”, or “By.cssSelector()”. In this case, the most reliable option is to access them using the “By.xpath()” method.

In This Tutorial for How to handle web table in Selenium WebDriver, you will learn-

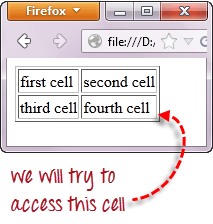
* [What is a Web Table in Selenium?](https://www.guru99.com/selenium-webtable.html#1)
* [How to write XPath for Table in Selenium](https://www.guru99.com/selenium-webtable.html#10)
* [Accessing Nested Tables](https://www.guru99.com/selenium-webtable.html#11)
* [Using Attributes as Predicates](https://www.guru99.com/selenium-webtable.html#12)
* [Shortcut: Use Inspect Element for Accessing Tables in Selenium](https://www.guru99.com/selenium-webtable.html#13)

**How to write XPath for Table in Selenium**

Consider the HTML code below for handling web tables in Selenium.



We will use[XPath](https://www.guru99.com/xpath-selenium.html)to get the inner text of the cell containing the text “fourth cell.”



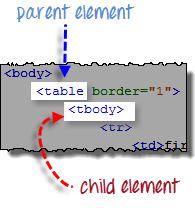
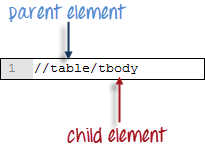
**Step 1 – Set the Parent Element (table)**

**XPath locators in WebDriver always start with a double forward slash “//” and then followed by the parent element**. Since we are dealing with web tables in Selenium, the parent element should always be the <table> tag. The first portion of our Selenium XPath table locator should, therefore, start with “//table”.

Selenium Web Table Example

**Step 2 – Add the child elements**

The element immediately under <table> is <tbody> so we can say that <tbody> is the “child” of <table>. And also, <table> is the “parent” of <tbody>. All child elements in XPath are placed to the right of their parent element, separated with one forward slash “/” like the code shown below.

**Step 3 – Add Predicates**

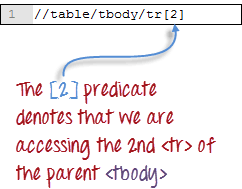
The <tbody> element contains two <tr> tags. We can now say that these two <tr> tags are “children” of <tbody>. Consequently, we can say that <tbody> is the parent of both the <tr> elements.

Another thing we can conclude is that the two <tr> elements are siblings. **Siblings refer to child elements having the same parent**.

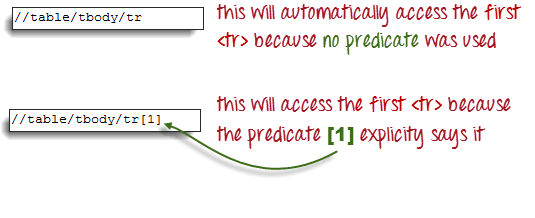
To get to the <td> we wish to access (the one with the text “fourth cell”), we must first access the **second** <tr> and not the first. If we simply write “//table/tbody/tr”, then we will be accessing the first <tr> tag.

So, how do we access the second <tr> then? The answer to this is to use **Predicates**.

**Predicates are numbers or HTML attributes enclosed in a pair of square brackets “[ ]” that distinguish a child element from its siblings**. Since the <tr> we need to access is the second one, we shall use “[2]” as the predicate.



If we won’t use any predicate, XPath will access the first sibling. Therefore, we can access the first <tr> using either of these XPath codes.

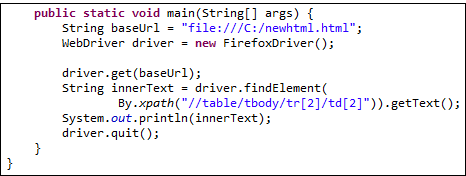


**Step 4 – Add the Succeeding Child Elements Using the Appropriate Predicates**

The next element we need to access is the second <td>. Applying the principles we have learned from steps 2 and 3, we will finalize our XPath code to be like the one shown below.

Web Table in Selenium using Xpath

Now that we have the correct XPath locator, we can already access the cell that we wanted to and obtain its inner text using the code below. It assumes that you have saved the HTML code above as “newhtml.html” within your C Drive.



public static void main(String[] args) {

String baseUrl = "http://demo.guru99.com/test/write-xpath-table.html";

WebDriver driver = new FirefoxDriver();

driver.get(baseUrl);

String innerText = driver.findElement(

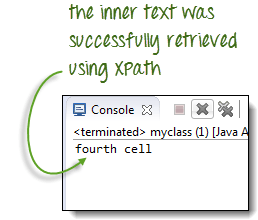
By.xpath("//table/tbody/tr[2]/td[2]")).getText();

System.out.println(innerText);

driver.quit();

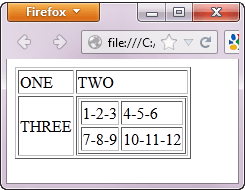
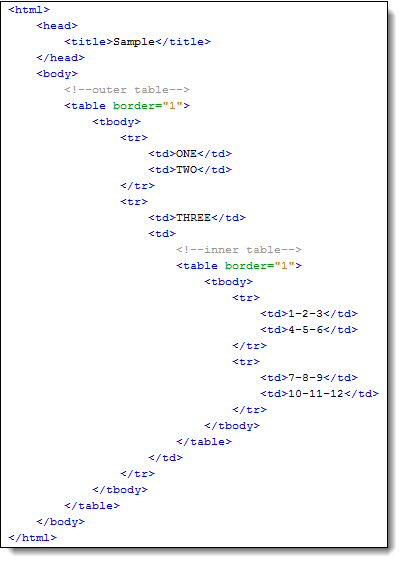
}

}

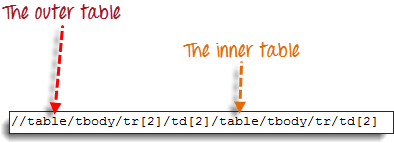


**Accessing Nested Tables**

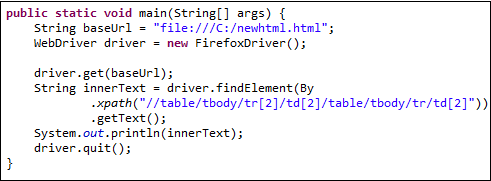
The same principles discussed above applies to nested tables. **Nested tables are tables located within another table**. An example is shown below.



To access the cell with the text “4-5-6” using the “//parent/child” and predicate concepts from the previous section, we should be able to come up with the XPath code below.



The WebDriver code below should be able to retrieve the inner text of the cell which we are accessing.



public static void main(String[] args) {

String baseUrl = "http://demo.guru99.com/test/accessing-nested-table.html";

WebDriver driver = new FirefoxDriver();

driver.get(baseUrl);

String innerText = driver.findElement(

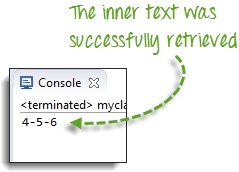
By.xpath("//table/tbody/tr[2]/td[2]/table/tbody/tr/td[2]")).getText();

System.out.println(innerText);

driver.quit();

}

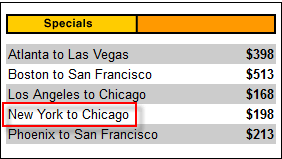
The output below confirms that the inner table was successfully accessed.

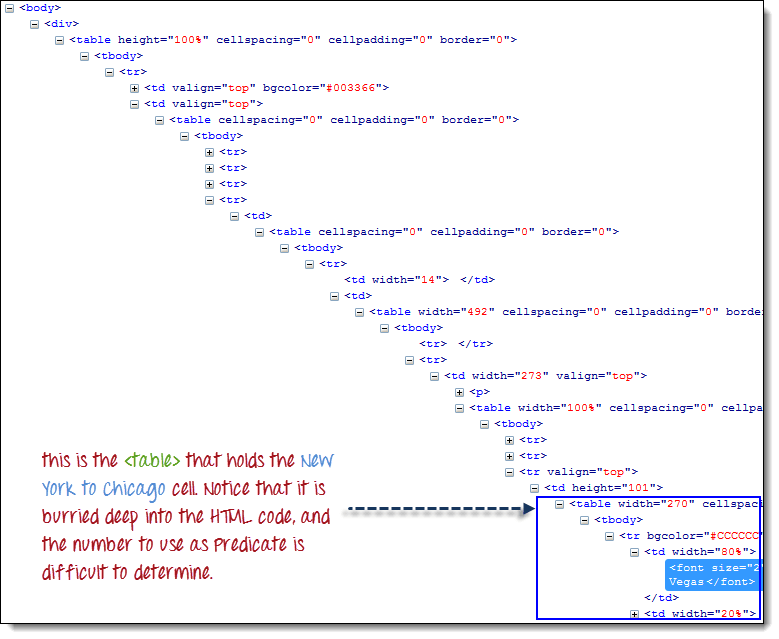


**Using Attributes as Predicates**

If the element is written deep within the HTML code such that the number to use for the predicate is very difficult to determine, we can use that element’s unique attribute instead.

In the example below, the “New York to Chicago” cell is located deep into Mercury Tours homepage’s HTML code.

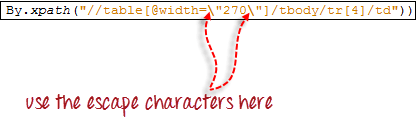




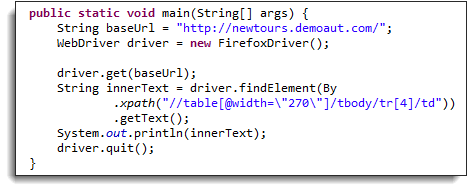
In this case, we can use the table’s unique attribute (width=”270″) as the predicate. **Attributes are used as predicates by prefixing them with the @ symbol**. In the example above, the “New York to Chicago” cell is located in the first <td> of the fourth <tr>, and so our XPath should be as shown below.

Selenium Web table example with Attributes as Predicates

Remember that when we put the XPath code in Java, we should use the escape character backward slash “\” for the double quotation marks on both sides of “270” so that the string argument of By.xpath() will not be terminated prematurely.



We are now ready to access that cell using the code below.



public static void main(String[] args) {

String baseUrl = "http://demo.guru99.com/test/newtours/";

WebDriver driver = new FirefoxDriver();

driver.get(baseUrl);

String innerText = driver.findElement(By

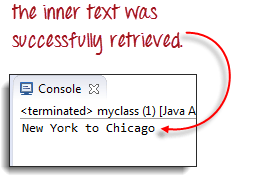
.xpath("//table[@width=\"270\"]/tbody/tr[4]/td"))

.getText();

System.out.println(innerText);

driver.quit();

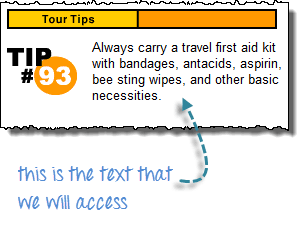
}



**Shortcut: Use Inspect Element for Accessing Tables in Selenium**

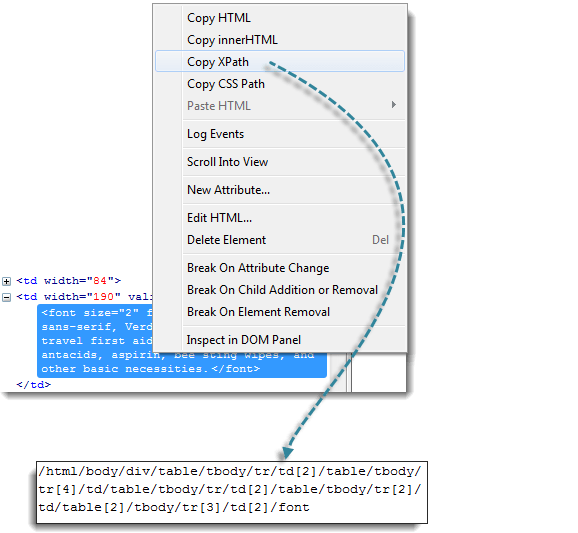
If the number or attribute of an element is extremely difficult or impossible to obtain, the quickest way to generate the XPath code is using Inspect Element.

Consider the example below from Mercury Tours homepage.



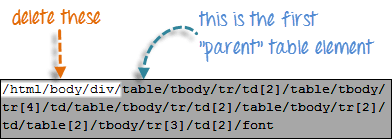
**Step 1**

Use Firebug to obtain the XPath code.



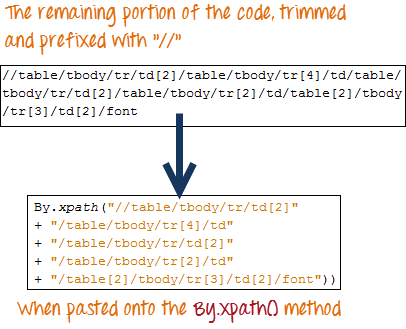
**Step 2**

Look for the first “table” parent element and delete everything to the left of it.

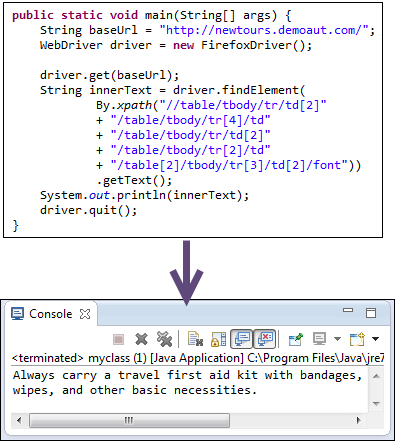


**Step 3**

Prefix the remaining portion of the code with double forward slash “//” and copy it over to your WebDriver code.



The WebDriver code below will be able to successfully retrieve the inner text of the element we are accessing.



public static void main(String[] args) {

String baseUrl = "http://demo.guru99.com/test/newtours/";

WebDriver driver = new FirefoxDriver();

driver.get(baseUrl);

String innerText = driver.findElement(By

.xpath("//table/tbody/tr/td[2]"

+ "//table/tbody/tr[4]/td/"

+ "table/tbody/tr/td[2]/"

+ "table/tbody/tr[2]/td[1]/"

+ "table[2]/tbody/tr[3]/td[2]/font"))

.getText();

System.out.println(innerText);

driver.quit();

}