How to Debug HTML and JAVA Script and DOM, XPath

CIS 408 Internet Computing

Sunnie Chung

Google Chrome:

1. Open the URL / web page in the Chrome browser
2. Open the Web developer tools by pressing:
   - *Cmd + Alt + I* (on Mac)
   - or by clicking View -> Developer -> Developer tools
   - or *icon ... top right corner -> More tool -> Developer tools*
3. Click on the Elements tab in the Web developer tools

```
<table>
<thead>
<tr>
<th>CLEVELAND STATE UNIVERSITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Romeo &amp; Juliet, the Classic Tale of Star Crossed Lovers, Comes to CSU</td>
</tr>
<tr>
<td>Student production runs March 2-12 at Playhouse Square's Outlaw Theatre</td>
</tr>
</tbody>
</table>
```

OR
4. Click on the **Console** tab in the Web developer tools.

- The JavaScript function `console.log` is very useful for debugging. It takes a string argument, which it prints to the JavaScript log. You can display this log with "Tools -> JavaScript console" in the Chrome control menu. If you are having trouble figuring out what is happening in your JavaScript, sprinkle `console.log` statements around your code so you can see which code is being executed.

- You may also find the `debugger` statement useful in debugging. If the text `"debugger;"` is executed in a JavaScript program, it causes the program to drop into the JavaScript debugger. This is analogous to a breakpoint, except that your code can control when it triggers.

- If nothing seems to be happening in your JavaScript code, it’s possible that your code contains an error that is causing it to be aborted. To find out if this is happening, open the JavaScript console to see if there are any errors.

The JavaScript console and Chrome DevTools will be your best friends for the remaining projects. We encourage you to spend some time this project using and learning how these tools work.
MS Internet Explore:

Function 12 will bring Debug Mode on your window in HTML as below:
How To Debug XPATH

How to find the XPath in WebDriver test using Chrome browser?

While developing a test or when debugging a test you want to check to what element the XPath used in your test refers to. You will find this useful since you don’t need to install any other plugin or widget, you can use just built-in browser functionality.

For example, let’s imagine we have the following code in our UI Selenium test:

```java
@FindBy(xpath = "//div[contains(@class,'item--lowest')]//a[contains(@class,'summary__date')]")
public WebElement summaryDateElement;
```

To identify to which element this refers to just do the following:

1. Open the URL / web page in the Chrome browser
2. Open the Web developer tools by pressing:
   - Cmd + Alt + I (on Mac)
   - or by clicking View -> Developer -> Developer tools
   - or by Right-Click and Inspect Element
3. Click on the Console tab in the Web developer tools
4. Paste in the console the XPath from your test in the XPATH format:
5. Hovering with the mouse over the returned result will highlight the page element that the XPATH refers to.

You will see in the console what the XPath returns.

You can use $x in the Chrome javascript console. No extensions needed.

ex: $x("//img")

Also the search box in the web inspector will accept xpath
Google Chrome Xpath Helper

Extract, edit, and evaluate XPath queries with ease.
XPath Helper makes it easy to extract, edit, and evaluate XPath queries on any webpage.

IMPORTANT: After installing this extension, you must reload any existing tabs or restart Chrome for the extension to work.

Instructions:
1. Open a new tab and navigate to any webpage.
2. Hit Ctrl-Shift-X (or Command-Shift-X on OS X), or click the XPath Helper button in the toolbar, to open the XPath Helper console.
3. Hold down Shift as you mouse over elements on the page. The query box will continuously update to show the XPath query for the element below the mouse pointer, and the results box will show the results for the current query.
4. If desired, edit the XPath query directly in the console. The results box will immediately reflect your changes.
5. Repeat step (2) to close the console.

If the console gets in your way, hold down Shift and then move your mouse over it; it will move to the opposite side of the page.

One word of caution: When rendering HTML tables, Chrome inserts artificial <tbody> tags into the DOM, which will consequently show up in queries extracted by this extension.