

To Use XPath Expressions With Mozilla-Fire Fox:

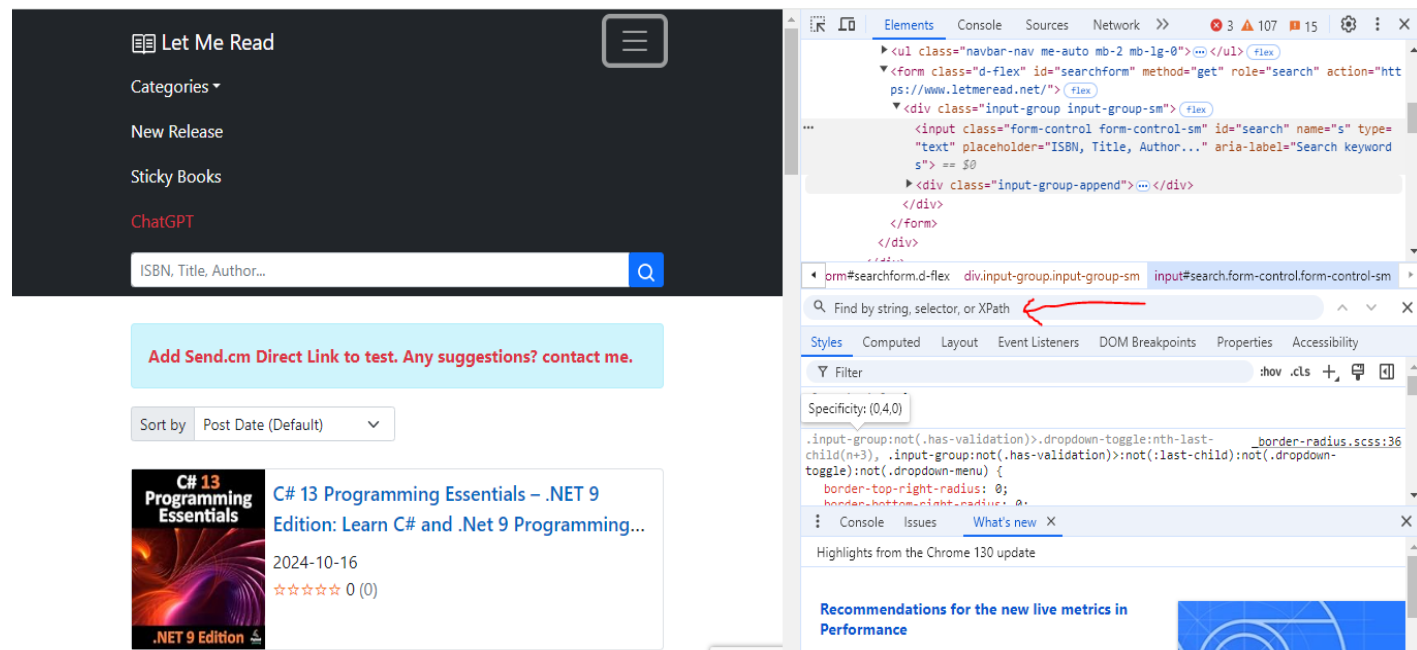
- From Terminal: \$x(... Expression Will Be Here ...)

\*\*\*\*\*

**Note:** When We Build Bots, To Automate The Web Pages Work, We Must Be Careful When Handle: iframe-Elements.

\*\*\*\*\*

**Note:** Using Chrome We Can Inspect Any Element, Then We Press CTRL + F:



\*\*\*\*\*

**Note:** For Chrome Browser, Also We Can Use \$x(... Expression Will Be Here ...)

```
> $x("//*[@class='visit']")
< (12) [a.visit, a.visit, a.visit, a.visit, a.visit, a.visit, a.visit, a.visit, a.visit, a.visit, a.visit, a.visit]
```

\*\*\*\*\*

**Note:** When We Choose Elements By Link Text OR Partial Link Text We Must Be Attention To Spaces Around Any Word.

\*\*\*\*\*

**Note:** When We Use CSS Selectors using the Formula, e.g. `input[class='class-name-here']`; it is exact matching, that means if the tag has multiple Classes We Must Set Them All, Otherwise: It Will Not Selected.

\*\*\*\*\*

**Note 1:** With Single Slash We Look Only For Next Child Only (Immediate Child).

**Note 2:** With Double Slash We Look For All Nested Children.

\*\*\*\*\*

**Note:** We Must Understand The Using Of / And // To Build Effective XPATH-Expressions That Will Not Break When The Page Change Its Code.

\*\*\*\*\*

**Note:** We Can Use `text()` in XPATH For Finding Elements Depending On Their Text Content; e.g. `a[text()='Text-Will-Be-Here']`.

\*\*\*\*\*

**Note (To Remember):** If We Want To Search In Text OR In Attribute, We Can Use Contains; e.g. `a[contains(text, 'Text-Will-Be-Here')]`, `a[contains(@href, 'Link-Will-Be-Here')]`.

\*\*\*\*\*

**Note:** We Can Use Logical Operator Inside The Formula Of Tag; e.g. `a[contains(@id, 'Id-Will-Be-Here') and contains(@class, 'Class-Will-Be-Here')]`.

\*\*\*\*\*

**Note (To Remember):** If We Want To Select Depending On Tag Attribute; e.g. `a[@class='Class-Will-Be-Here']`. OR `div[@id='id-will-be-here']`.

\*\*\*\*\*

**Note:** If We Want To Select Element Depending On The Start Value Of Its Specific Attribute; We Can Use Start-With; e.g.: `div[starts-with(@class, ' ')]`

\*\*\*\*\*

**Note:** If We Want To Select The Parent Element Of Specific Element Using XPATH; We Can Use `xpath-expression-here//parent::tag-name-here`.

**Ex:** `a[@href='/url-will-be-here']//parent::ul`

\*\*\*\*\*

**Note:** If We Want To Select The Preceding-sibling Of Specific Element, We Can Use `preceding-sibling::tag-will-be-here`.

**Ex:** `a[@href='url-will-be-here']//parent::li//preceding-sibling::li`

\*\*\*\*\*

**Note:** If We Want To Find All Following Sibling Of Specific Element We Can Use: `following-sibling::tag-will-be-here`.

**Ex:** `a[@href='url-will-be-here']//parent::li//preceding-sibling::li//following-sibling::li`

**Ex:** If We Want To Select Specific Following Sibling We Can Use Indexing Of XPATH; In XPATH Indexing Starts From 1.

`a[@href='url-will-be-here']//parent::li//preceding-sibling::li//following-sibling::li[1]`

\*\*\*\*\*

**Note (To Remember):** If We Want To Select Element Depending On Text, We Can Use: `Tag-Name[text()='text-will-be-here']`

**Ex:** `a[text()='Text-Will-Be-Here']`

\*\*\*\*\*

Note: To Get The Current Title Of Page: *driver.title*

```
print("The Title Of Page is: ", driver.title);
*****
```

Note: To Get The Current URL Of Page: *driver.current\_url*

```
print("The Current URL Of Page Is: ", driver.current_url);
*****
```

Note: To Maximize The Window Of Driver: *driver.maximize\_window();*

```
driver.maximize_window();
*****
```

Note: To Refresh The Page Of Driver (ex: Exception Has Happened):

*driver.refresh();*

```
*****
```

Note: To Go One Step Forward In The History Of The Browser: *driver.forward();*

```
*****
```

Note: To Go One Step Backward In The History Of The Browser: *driver.back();*

```
*****
```

Note: To Get The Page Source Of The Page: *driver.page\_source*

```
*****
```

Note: The Difference Between *driver.close();* And *driver.quit();*

- If We Have Multiple Windows That Are Opened, Then *driver.quit();* Will Close All Of Them.
- If We Have Multiple Windows That Are Opened, Then *driver.close();* Will Close The Active One.

```
*****
```

Note (To Remember): If We Want To Select Element With Attribute We Can use:

- Write: *a[@href='url-will-be-here']*

```
*****
```

**Note:** To Type Inside The Input Elements, We Can Use:

- We Write: *web\_driver\_element.send\_keys('Data-Will-Be-Here');*

```
email_input = driver.find_element(By.XPATH, '//input[@id="email"]');
password_input = driver.find_element(By.XPATH, '//input[@id="login-password"]');

email_input.send_keys("gaafer@loka.com");
password_input.send_keys("Test@1234567890");
*****
```

**Note:** If We Want To Clear The Data Of Input Field, After We Send Keys To It, We Can Use: *web\_driver\_element.clear();*

```
*****
```

**Note:** If We Want Our Driver To Wait N-Time-Of-Seconds For All The Session Of Browser, Then We Can Use: *driver.implicitly\_wait(30)*

```
*****
```

**Note (From Me):** May We Have Multiple Elements With Same Class, But In JS The Developer Make A Trick, To Save The Site, Then We Must Select All Of Them And Then Make Action.

```
*****
```

**Note:** In Web Driver Elements We Have:

- *First: is\_displayed().*
- *Second: is\_selected().*
- *Third: is\_disabled().*

```
img_element.is_displayed();
img_element.is_enabled();
img_element.is_selected();
*****
```

Note (To Remember): If We Want To Select Depending On Multiple Conditions, We Can Use Logical Operators:

*//input[contains(@type, 'radio') and contains(@name, 'cars')]*

\*\*\*\*\*

To Handle Select-Element:

```
from selenium.webdriver.support.select import Select;
```

```
element = driver.find_element(By.XPATH, '//select[@id="carselect"]');
```

```
select_element = Select(element);
```

```
select_element.select_by_value('honda');
```

```
# The Index Start From Zero And Index Can Be String.
```

```
# The Index Start From Zero.
```

```
select_element.select_by_index(1);
```

```
select_element.select_by_visible_text('BMW');
```

\*\*\*\*\*

To Get The Text Inside The Element That We Find It:

```
driver.maximize_window();
```

```
driver.get("https://www.letscodeit.com/practice");
```

```
WebDriverWait(driver=driver, timeout=15).until(
```

```
    EC.presence_of_element_located((
```

```
        By.XPATH, '//table[@id="product"]'
```

```
    ))
```

```
);
```

```
text_element = driver.find_element(
```

```
    By.XPATH,
```

```
    "//table[@id='product']/tbody/tr[2]/td[@class='course-name']"
```

```
);
```

```
print("The Text Element Is: ", text_element);
```

```
print("The Content Of Text Element Is: ", text_element.text);
```

```
driver.quit();
```

\*\*\*\*\*

*Note (From Me):* The *Until-Method* of Explicit Wait Return an Element, So We Don't Need To Use *driver.find\_element-Method* Again.

\*\*\*\*\*

*Note (From Me):* The **driver.find\_element** Also Return An Element That Can Use *find\_element()* && *find\_elements()*.

\*\*\*\*\*

*Note (From Testing):* The File Name Of Image Must Be End With **.png**

```
dest_file_name = str(time.time()) + "_j-l-screen-shot.png";
```

```
try:
```

```
    driver.save_screenshot(dest_file_name);
```

```
except Exception as e:
```

```
    print("The Error Is: ", e.__str__());
```

\*\*\*\*\*

In This Way, We Can Execute The Script That We Want.

*Note:* Also; We Have Method To Execute Async Scripts, Like: **setTimeout**, **setInterval**.

```
title = driver.execute_script("return document.title");
```

```
print("The Title Of Document Is: ", title);
```

\*\*\*\*\*

To Get The Size Of Window That We Use It For Run Tests, We May Use JS.

```
height = driver.execute_script("return window.innerHeight");
```

```
width = driver.execute_script("return window.innerWidth");
```

```
print("The Height Of Window Is: " + str(height));
```

```
print("The Width Of Window Is: " + str(width));
```

\*\*\*\*\*

Note: To Scroll Down Using JS: **window.scrollTo(x, y);**

- **Ex:** *window.scrollTo(0, 800);*
  - This Will Scroll Down By **800 pixels**.
  - **X** is For Horizontal Scrolling, **Y** is For Vertical Scrolling.
- **Note:** If We Set Negative Value For Scrolling, It Will Scroll Up.

\*\*\*\*\*

To Pass Arguments To `execute_script()`:

- Run The Command:

*`driver.execute_script("arguments[0].scrollIntoView(true)", argument_here);`*

- Note: The argument Can Be: *element* of *`driver.find_element()`*;

\*\*\*\*\*

```
t1 = driver.find_element(By.LINK_TEXT, "Cookie Policy");
location = t1.location_once_scrolled_into_view;

for key, value in location.items():
    print("The Key Is: ", key, ", The Value Is: ", value);

time.sleep(2);
```

\*\*\*\*\*

The Best Way To Handle The Element Not Intractable Is Using XPATH:

```
t2 = WebDriverWait(driver=driver, timeout=25, poll_frequency=1).until(
    EC.presence_of_element_located((By.XPATH, '//input[@id="search"]'))
)
```

\*\*\*\*\*

To Get The Current Window Handle:

```
w1 = driver.current_window_handle;
```

\*\*\*\*\*

To Get List Of Window Handles, When Open Multiple Windows:

```
w2 = driver.window_handles;
```

\*\*\*\*\*

To Switch From Window Handle To Window Handle:

```
for w in w2:
    if w not in w1:
        driver.switch_to.window(w);
        print("Switch To Window: ", w);
        break;
```

\*\*\*\*\*



To Switch To iFrame By ID:

```
driver.switch_to.frame(driver.find_element(By.ID, 'courses-iframe'));
*****
```

To Return To iFrame Parent:

```
driver.switch_to.parent_frame();
OR:
```

```
driver.switch_to.default_content();
*****
```

*Note 1:* We Can Also Switch To iFrame Using *name-Attribute*.

*Note 2:* We Can Also Switch To iFrame Using Numbers, Where The First iFrame Is **0**.

```
*****
```

*Note:* Also For Alert We Can Authenticate The User, Using: username, and password.

```
*****
```

In This Way We Can Switch To Alert And Confirm Pop-Up:

```
t4 = driver.switch_to.alert;
```

And In This Way We Confirm The Pop-Up:

```
t4.accept();
```

And In This Way We Can Cancel The Confirm:

```
t5.dismiss();
```

```
*****
```

To Perform Mouse Hover, Drag And Drop, Click The Element ...etc. We Need To Use ActionChains:

```
from selenium import webdriver;
from selenium.webdriver import ActionChains;
from selenium.webdriver.chrome.service import Service;
from selenium.webdriver.common.by import By;
import time;

service = Service(executable_path='chromedriver.exe');
driver = webdriver.Chrome(service=service);

driver.maximize_window();

driver.get("https://www.letscodeit.com/practice");

driver.execute_script("window.scrollTo(0, 700)");

time.sleep(1);

t1 = driver.find_element(By.XPATH, '//div[@class="mouse-hover"]');

try:
    actionChains = ActionChains(driver=driver);

    actionChains.move_to_element(to_element=t1).perform();

    time.sleep(3);

    t2 = driver.find_element(By.XPATH, '//div[@class="mouse-
hover"]//a[text()="Top"]');
    t3 = driver.find_element(By.XPATH, '//div[@class="mouse-
hover"]//a[text()="Reload"]');

    # t2.click();

    # Second Way
    actionChains.move_to_element(to_element=t2).click().perform();

    time.sleep(3);

except Exception as e:
    print("The Exception Is: ", e.__str__());
*****
```