XPATH COMPENDIUM AND PRACTICE

XPath is the address of an HTML element.
There are examples of writing XPath in multiple ways in this document.

Two types of XPath:

- Absolute XPath
- Relative XPath

How to write an XPath?

1. single forward slash /
2. double forward slash //
3. XPath functions
4. Relative XPath using axes
5. Relative XPath without using axes

XPath syntax and Functions

- /tagName
- //tagName[@attributeName=”attribute Value”]
- //tagName[@name,”subStringOfValue”)]
- AND, OR functions
- text()
- //tagname[text()=’value’]
- //tagname[contains(text(),’value’)]
- normalize-space()
- //tagname[normalize-space()=’value’]
- Index, position, last, etc.

What is the right platform to write and verify XPath

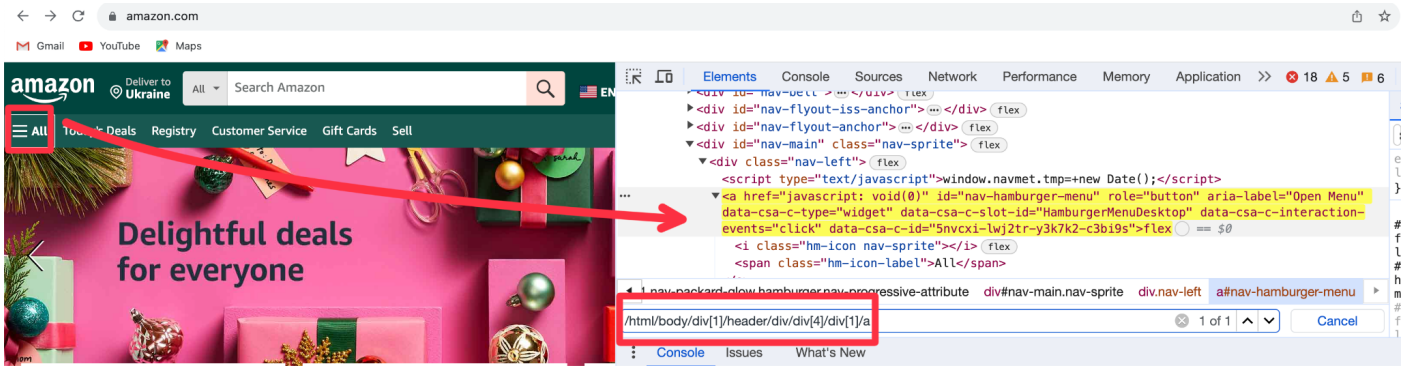
1. Chrome DevTools
2. Console Sx
3. Using SelectorsHub

XPath cheetsheet

Write XPath using XPath functions

single forward slash /

Website: https://www.amazon.com/ Element: hamburger menu button



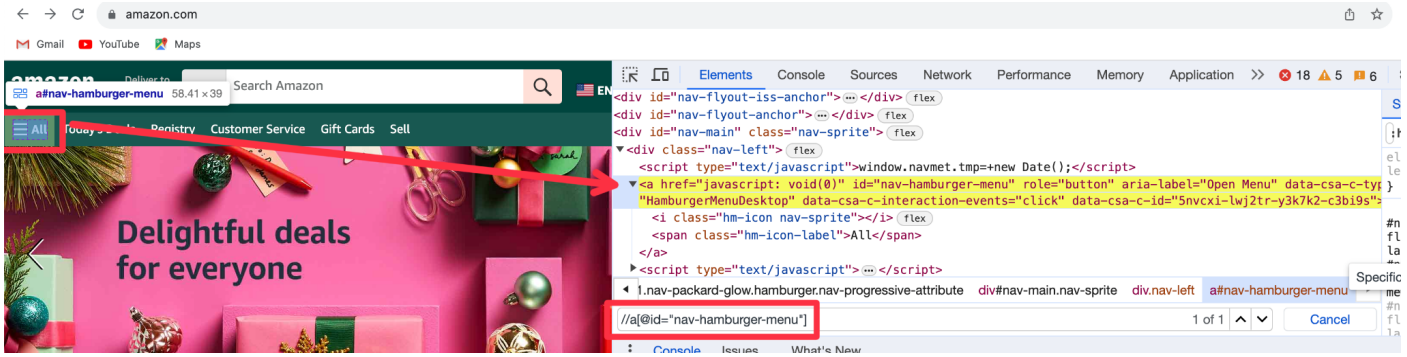
Absolute XPath:
/html/body/div[1]/header/div/div[4]/div[1]/a | /tagName

(not recommended to use because if any of the above elements changes the XPath will break)

---

double forward slash //

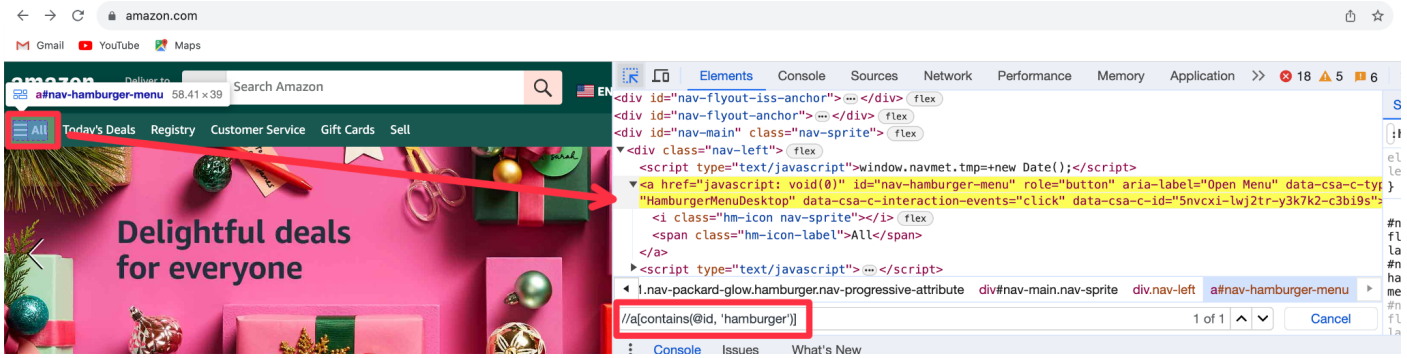
Website: https://www.amazon.com/ Element: hamburger menu button



Relative XPath:
//a[@id="nav-hamburger-menu"] | //tagName[@attributeName=”attribute Value”]

XPath functions

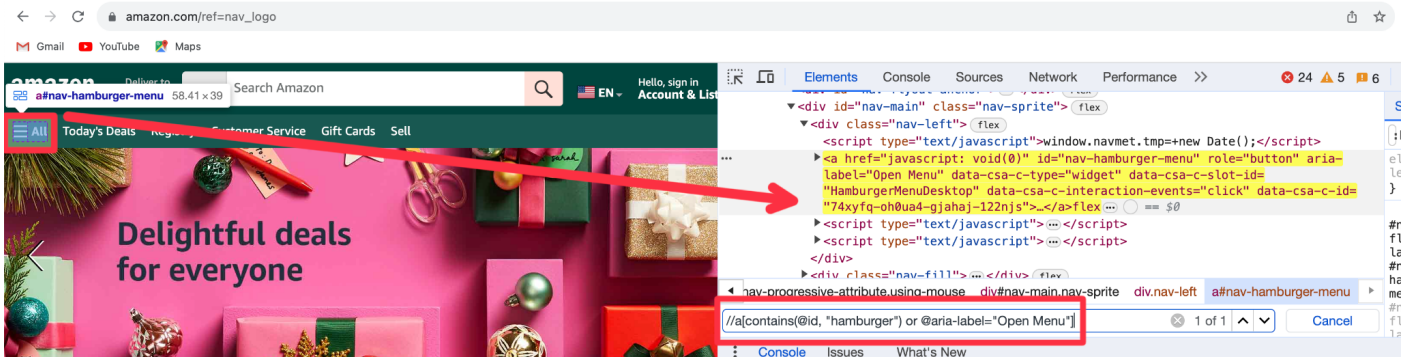
Website: https://www.amazon.com/ Element: hamburger menu button



Using ‘contains’ method to shorten the XPath:
//a[contains(@id,'hamburger')] | //tagName[@name,”subStringOfValue”)

AND, OR functions

Website: https://www.amazon.com/ Element: hamburger menu button



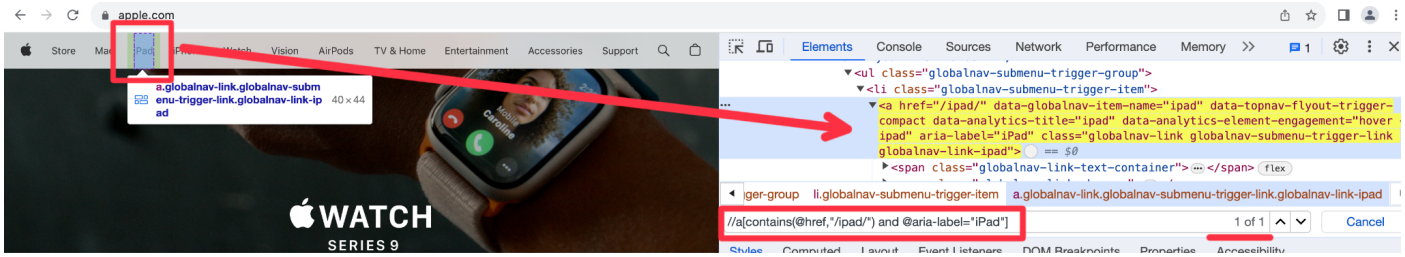
Add OR in case the first XPath won't work:
//a[contains(@id,"hamburger") or @aria-label="Open Menu"]

Add AND to make the XPath more robust:
//a[contains(@id,"hamburger") and @aria-label="Open Menu"]

Website: https://www.google.com/ Element: Google logo

//img[contains(@class,"lnXdpd") or @alt="Google"]

Website: https://www.google.com/ Element: Google logo



//a[contains(@href,"/ipad/") and @aria-label="iPad"]

(if not to use AND function in this case, the XPath selects 8 elements including those in the dropdown submenu of iPad category)

---

text() functions

Website: https://www.amazon.com/

Element: hamburger menu button with text

//a/span[text()='All'] | //tagname[text()='value']

//a/span[text()='All' and @class="hm-icon-label"]

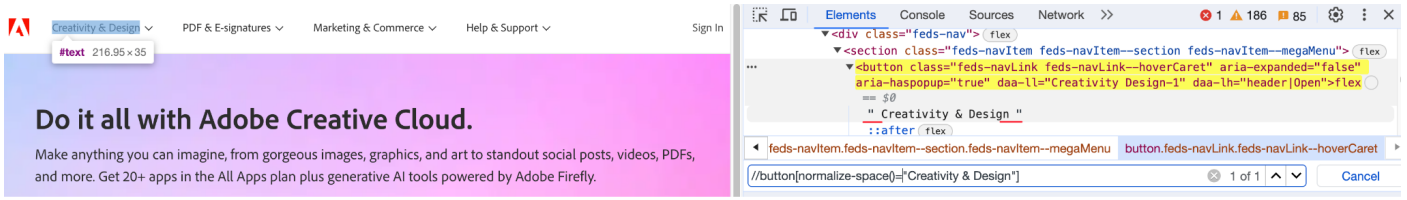
//a/span[contains(text(), 'All')] | //tagname[contains(text(),'value')]

---

normalize-space()

Website: https://www.adobe.com/

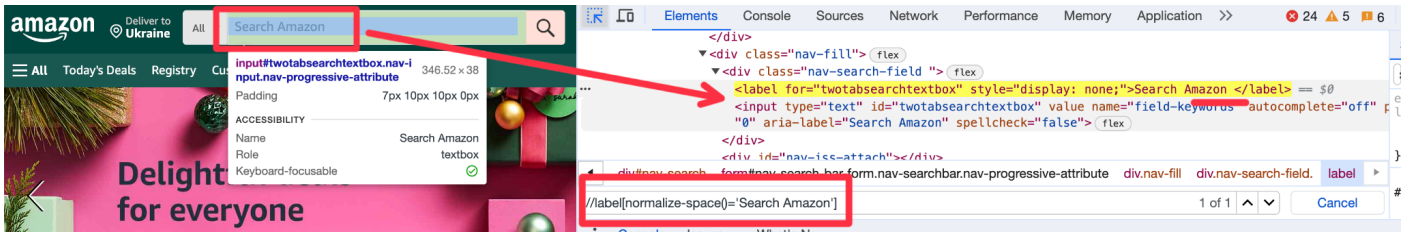
Element: menu item text



The 'normalize-space' function removes an unnecessary space in the text:
//div[section[button[normalize-space()="Creativity & Design"]]]//button[normalize-space()="PDF & E-signatures"]

Website: https://www.amazon.com/

Element: Search Amazon text inside the Search Field



The 'normalize-space' function removes an unnecessary space in the text:
//label[normalize-space()='Search Amazon'] | //tagname[normalize-space()='value']

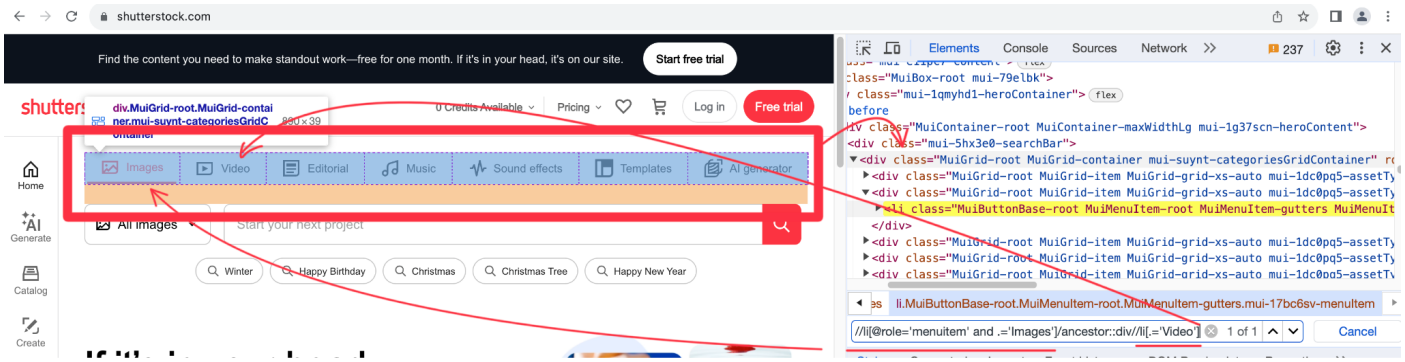
//label[contains(normalize-space(),'Search Amazon')]

//label[contains(normalize-space(),'Search Amazon') and @style='display: none;']

(of course Amazon.com don't have unnecessary spaces in their code, or I didn't find one. So I had to modify HTML a little for this example)

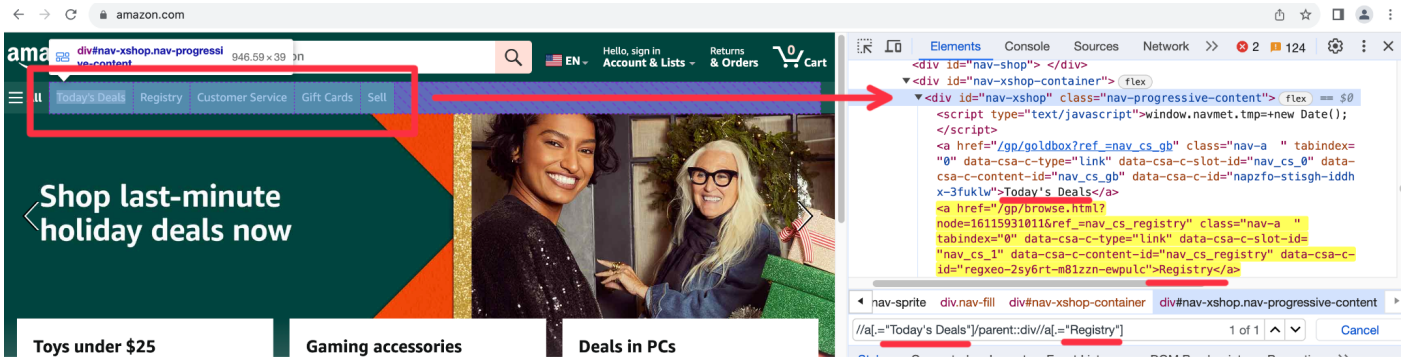
Relative XPath using axes

Website: https://www.shutterstock.com/ Element: menu items



Find the 'Video' menu item by going from the 'Images' menu item to their common ancestor:
//li[@role='menuitem' and .='Images']/ancestor::div//li[.='Video']

Website: https://www.amazon.com/ Element: menu items

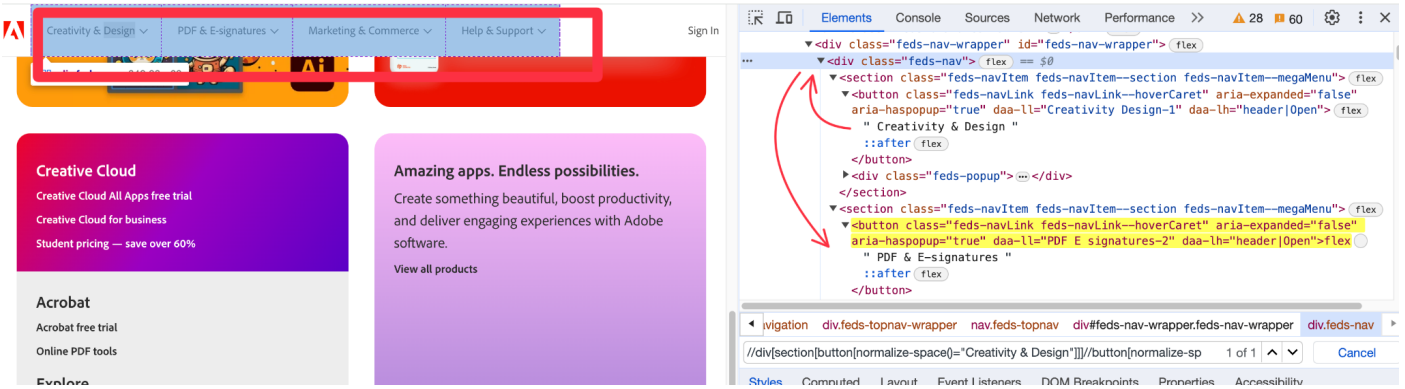


Find the 'Registry' menu item by going from the 'Today's Deals' menu item to their common parent:
//a[.="Today's Deals"]/parent::div//a[.="Registry"]

Relative XPath without axes

Website: https://www.adobe.com/

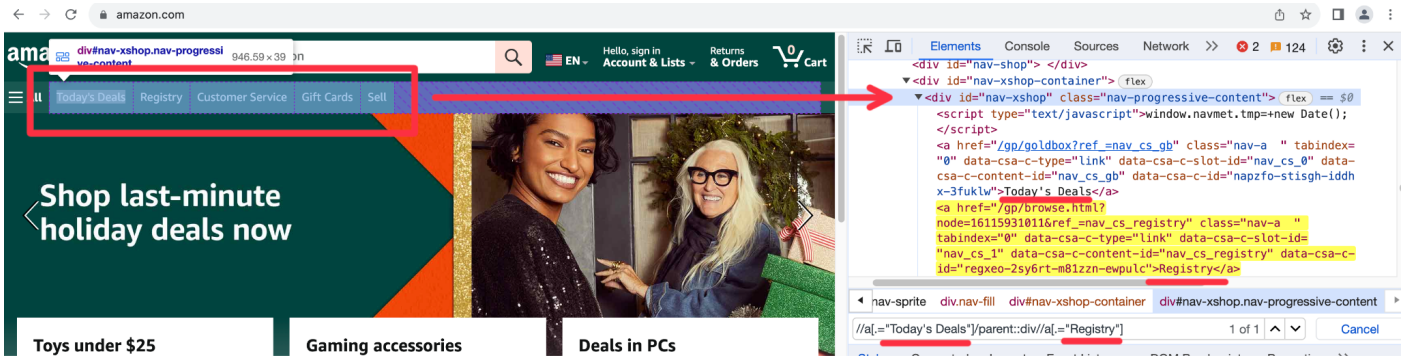
Element: menu items



Using '[ ]' to find the parent of the menu items elements and jump to the next button:
//div[section[button[normalize-space()="Creativity & Design"]]]//button[normalize-space()="PDF & E-signatures"]

Website: https://www.adobe.com/

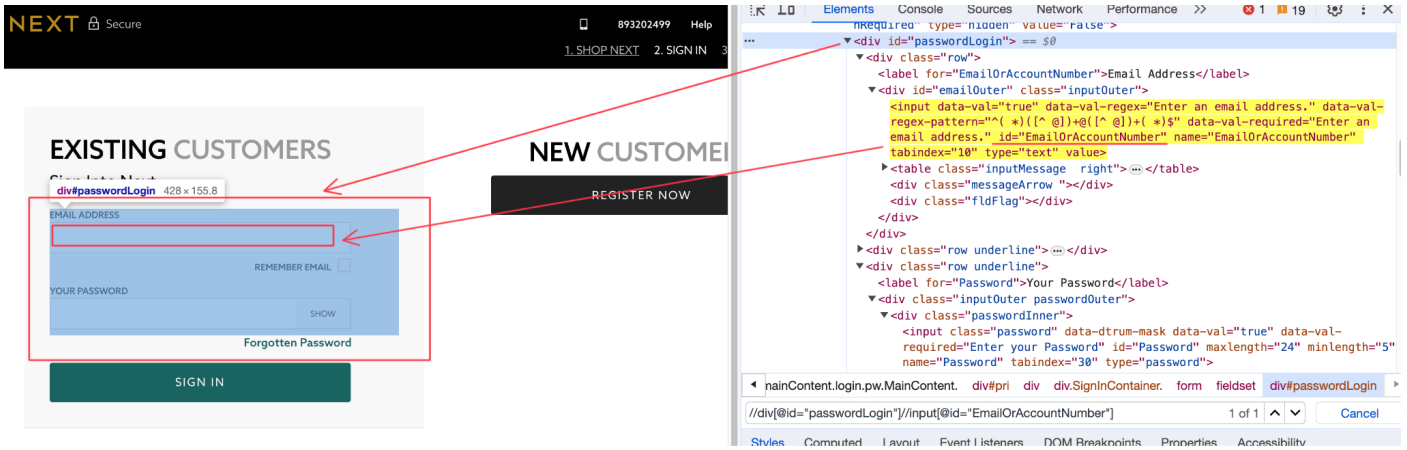
Element: list items



Using '[ ]' to find the parent of the list items elements and jump to the next item:
//div[ul[li[a[.="Creative Cloud"]]]]//a[.="Adobe Express"]

Website: https://account.next.ua/en/Login

Element: LogIn fields input



One more way to write more unique parent XPath:
//div[@id="passwordLogin"]//input[@id="EmailOrAccountNumber"]