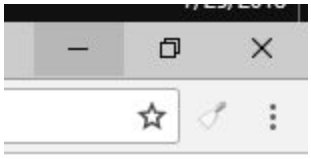
Web Scraping with the Chrome Scraper Extension

Set up the Scraper Extension

Install the Scraper extension in Chrome.

Test the Scraper extension

- [Highlight text and] **Right click** anywhere an choose Scrape similar…
- Highlight text and click the **icon in the toolbar** and choose Scrape similar... Instructions use right click, but you can always click the icon in the toolbar.



Check that the settings are correct

- Examine the window that opens, looking for **XPath** in the upper right. If it says "JQuery", click the click the Reset button at the bottom of the Scraper window and then restart Chrome.

Using the Developer Toolbar in Chrome

- To Inspect an element, it right click on it.
- You can also press Command+Shift+C (Mac) or Ctrl+Shift+C (PC) then find the element.
- Right click on a tag and follow the menus to copy the XPath.
- In the Developer Window you can press Ctrl-F (PC) or Command+F (Mac) to open the Find toolbar and test your XPaths.

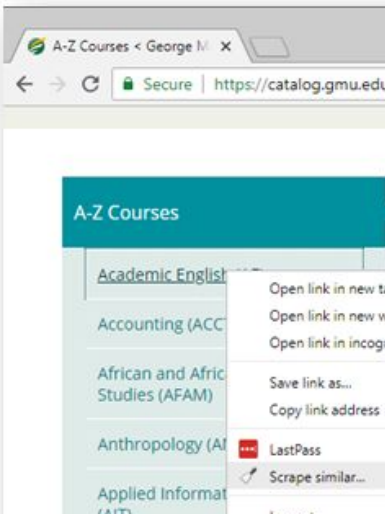
XPath Reference

HTML	**XPath** to locate **this**
this	p**/**span
anything	p**/**@class
this	p**[**@class**]**
this	p**[**@class="me"**]**
this	p**[**contains**(**@class, "me"**)]**
this	p**[**not**(**@class="me"**)]**
me	p**[**text**()**="me"**]/**@class
this	*[@id="me"]
Word**this**	div/text()
<hr>Word **this**	hr/following-sibling::p
<dl><dt>Word</dt><dd>**this**</dd></dl>	dl/dt[text()="Word"]/../dd

this div//b

Scrape a List

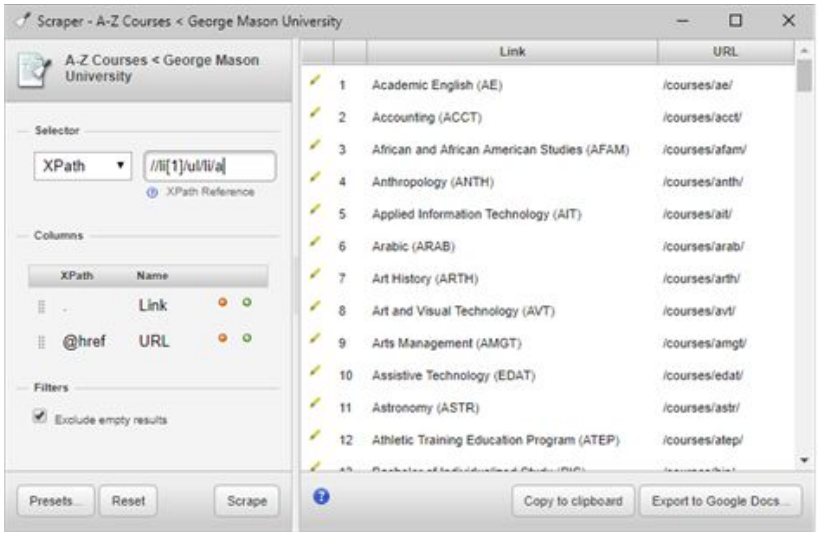
Obtain the List of Courses and URLs



Version A

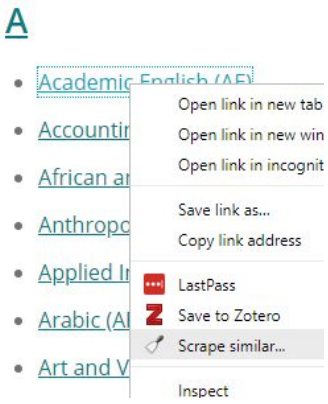
1. Go to https://catalog.gmu.edu/courses/
2. **Right Click** on the first link in the *left-side* list
3. Choose **Scrape similar...**
4. In the Scraper window, press **Copy to clipboard**
5. In Excel or Google Sheets, choose to **Paste** (ex Ctrl-V)

Extra Challenge: Change the Selector to include a tag ID instead of [1]



Version B

1. Go to https://catalog.gmu.edu/courses/
2. **Right Click** on the first link in the *middle* list
3. Choose **Scrape similar...**
4. In the Scraper window, **remove the [1]** to specify that you want all letter groupings, not just A.
5. Press **Scrape**
6. Press **Copy to clipboard**
7. In Excel or Google Sheets, choose to **Paste** (ex Ctrl-V)



Extra Challenge: Change the Selector to include a tag ID instead of [2]

Scrape a Table

1. Go to https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population
2. With your mouse, select a **whole row,** then right click and pick **Scrape Similar…**

Part 1

3. **Add a column** with the Green + button can click on the Name to label it Source URL
4. Click on the **XPath** area and type in the XPath you determine for the URL of the source.
    a. It should start with *[6] because that is the reference to the 6th column/td

Part 2

5. Tip: Use the red minus button to **remove all columns** *except* for the first two (*[1] and *[2])
6. Use the green plus button to add columns for **Country Name**, **Note**, **Country URL**, and **Flag URL.**
7. Click on the **XPath** area and type in the XPath you determine for the each element.
    a. The structure for Countries and Territories differ a bit, see below. Do Countries first.
    b. A few Territories have an extra <i>: #191 Artsakh, #219 Saint-Martin, #231 Saint Helena. Try to include them. Hint: You may want to use a function to specify which <i> to pick.

Country Listing:

```
<td align="left">
   <span class="flagicon" style="display:inline-block;width:25px;">
        <img alt="" src="[...]/23px-Flag_of_the_People%27s_Republic_of_China.svg.png"
        width="23" height="15" class="thumbborder" srcset=[...]  data-file-height="600">
    </span>
     <a href="/wiki/China" title="China">China</a>
    <sup id="cite_ref-5" class="reference"><a href="#cite_note-5">[Note 2]</a></sup>
</td>
```

Territory Listing:

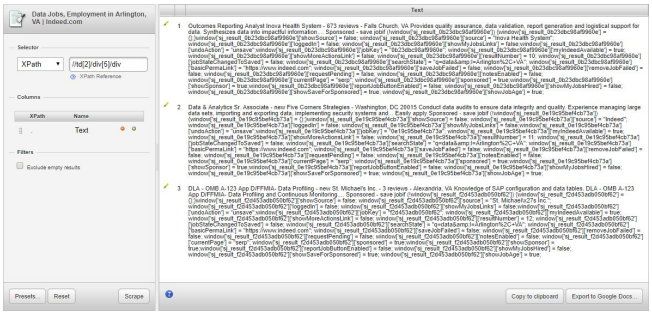
```
<td align="left">
    <i>
        <span class="flagicon" style="display:inline-block;width:25px;">
            <img alt="" src="[...]/23px-Flag_of_the_Pitcairn_Islands.svg.png" width="23"
            height="12" class="thumbborder" srcset=[...] data-file-height="600">
        </span>
        <a href="/wiki/Pitcairn_Islands" title="Pitcairn Islands">Pitcairn Islands</a>
    </i>
     <i>(<a href="/wiki/United_Kingdom" title="United Kingdom">UK</a>)</i>
</td>
```

Scrape Jobs from Indeed.com

1. **Go to https://www.indeed.com/** and search for jobs (I used “data”).

2. With your mouse, select the *whole* job listing, then right click and pick **Scrape Similar...**

   a. There should be a row for each job as in the screenshot. Ignore for now that only a few jobs appear.



3. **Identify an XPath** to obtain the **Job Title**

   a. To get you started, here is sample HTML for a title with some elements that you can use highlighted, see if you can use each one (e.g., make 4 different XPaths that work). Remember, classes are not separated, so use the contains() function to reference one.

```
<a target="_blank" id="sja1" data-tn-element="jobTitle"
class="jobtitle turnstileLink"
href="https://www.indeed.com/viewjob?jk=c74e528069aba
&amp;from=tp-serp&amp;tk=1c3gl72080j0f119&amp;tk=1c3gl0j0f119&amp;
jsa=6104" title="Digital Network Analyst" rel="noopener nofollow"
onmousedown="sjomd('sja1'); clk('sja1');"
onclick="setRefineByCookie([]);sjoc('sja1',0);
convCtr('SJ')">Digital Network Analyst</a>
```

4. **Add more columns** with the Green + button and identify XPaths to locate the other job elements, such as URL, Company, Location, and Summary

5. **Fix the XPath selector** to include all the jobs on the page by Inspecting the job listing as a whole and finding a more specific identifier for each individual job listing.

   a. The issue is that the Sponsored jobs are all in a special, extra div.

   b. **Note**: You may also have to fix some XPaths that previously worked because the structure of sponsored and unsponsored jobs also differs (uses div instead of span, etc)

6. **Display more job listings** by opening the Advanced Job Search (next to the Find button). Scroll to the bottom and have it display 50 results, find jobs and re-scrape.

7. **Remove extra spaces** in the Company and Summary columns (to see why, feel free to copy the data to Excel first)

   a. The XPath function to do this is called normalize-space(). Put the path inside the ().

   b. You could also use the TRIM function in Excel or Find-and-Replace

Get Country Flags from the State Dept using Google Sheets

In **Chrome**:

1. Go to https://www.state.gov/misc/list/
2. Right click one of the country names and click "**Scrape similar…**"
3. *If you want to practice XPath*, modify the Selector to
    a. Show all letters groups, not just A's
    b. Use another attribute to refer to the div instead of [2]
    c. Avoid including other links like the individual letters

    Note: you can use not() as in a[not(@id)] to find a link without an id attribute
4. Re-scrape if you made changes, then press **Copy to clipboard**
5. **Paste** into a Google Sheet so that the data is in columns A and B
6. Think ahead and create an XPATH to get the contents of the Flag image's **src** attribute
    a. Go to an individual country's page and **Inspect** the first image, which is of their flag. For more ideas, use the Scraper Extension and see what is put in the Selector box.
    b. Test it using the Developer Windows Find (Ctrl-F) or the Scraper Extension

In **Google Sheets**:

7. In column C, use the **IMPORTXML** function to fetch the image url for the flag.
    a. Refer to the URL of the country information
    b. Be sure to put the XPath you created in step 6 in quotes
8. In Column D, use **CONCAT** or **&** to add "http://www.state.gov" to the value in column C
    a. For a few countries, the URL already has the domain. If you wish, try to fix this. Hint: You can check if first/left character is a slash and thus needs the domain using LEFT
9. In Column E, use **IMAGE** to get the image from the URL in column D
    a. Tip: To make the images bigger, make the rows taller

“Answers” - Scrape a Table

7. Here are some ways to get the elements of the Country column:

`*[2]//@title`	Country or Territory
`*[2]//a[@title]`	Country or Territory (for sorting)
`*[2]/i[last()]/a`	County (if Dependent)
`*[2]/i[last()]/a/@title`	County (if Dependent) - Full
`*[2]//a[@title]/@href`	Country URL
`*[2]//sup`	Note
`*[2]//img/@src`	Flag URL

“Answers” - Getting Jobs from Indeed.com

3. Any of these will work:

`.//a`	Reference an element
`.//@title`	Reference an attribute
`.//*[@data-tn-element='jobTitle']`	Filters to tags with a specific attribute value
`.//*[contains(@class,"jobtitle")]`	Filters to tags with part of an attribute value

4. Here are some examples

```
.//@href
.//*[@class="company"]
.//*[@class="location"]
.//*[@class="summary"]
```

XPath	Name
.//@title	Title
.//@href	URL
.//*[@class="company"]	Company
.//*[@class="location"]	Location
.//*[@class="summary"]	Summary

5. Because the class for each job listing is “row result clickcard”, you must use the contains function:

```
//div[contains(@class,'clickcard')]
```

7a. Remove extra spaces with

```
normalize-space(.//*[@class="summary"])
```

7b. In Excel

```
=TRIM(C2)
```

“Answers” -- Get Country Flags

3a. `//div[2]/div/div/ul/li/a`

3b. `//div[@class="l-wrap"]/ul/li/a`

3c. To clean up the list of countries, here are some options for the Selector:

```
//div[@class="l-wrap"]/ul[@class="no-bullet"]/li/a
//div/ul[@class="no-bullet"]/li/a[not(@target)]
//div[@class="l-wrap"]/ul/li/a[contains(@href,"state.gov")]
```

6. To fetch the flag image’s URL, here are some options:

```
//p[1]/img/@src
//img[contains(@title,"Flag")]/@src
```

7. To Fetch the image src for the flag image in google sheets, here is an example:

```
=IMPORTXML(B2,"//p[1]/img/@src")
```

8. To do a straight concatenate, use one of the following in cell D2:

```
="http://www.state.gov”&C2
=CONCAT("http://www.state.gov", C2)
```

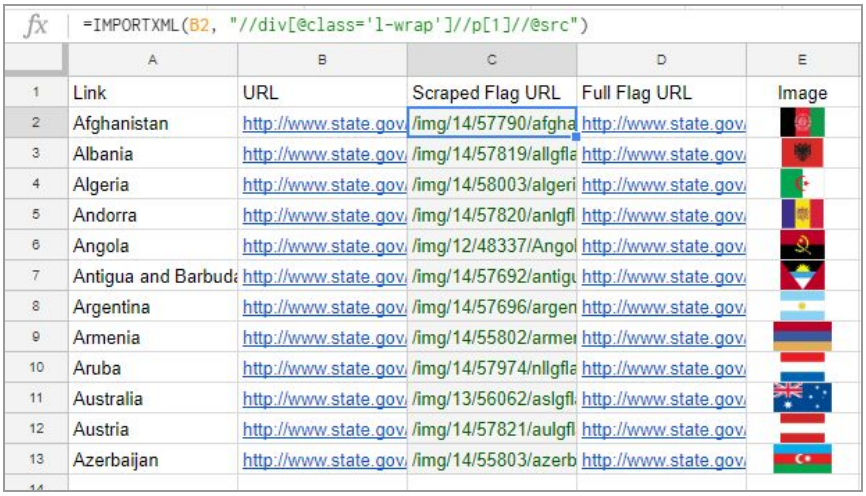
8a. To concatenate only if the first character is a slash, put this in cell D2

```
=IF(LEFT(C2, 1)="/", "http://www.state.gov"&C2, C2)
```

9. To Fetch the image at that URL put this in cell E2

```
=IMAGE(D2)
```

To see it in action, go to:
https://goo.gl/nBJJU3



fx =IMPORTXML(B2, "//div[@class='l-wrap']//p[1]//@src")

	A	B	C	D	E
1	Link	URL	Scraped Flag URL	Full Flag URL	Image
2	Afghanistan	http://www.state.gov	/img/14/57790/afgha	http://www.state.gov	
3	Albania	http://www.state.gov	/img/14/57819/allgfla	http://www.state.gov	
4	Algeria	http://www.state.gov	/img/14/58003/algeri	http://www.state.gov	
5	Andorra	http://www.state.gov	/img/14/57820/anlgfl	http://www.state.gov	
6	Angola	http://www.state.gov	/img/12/48337/Angol	http://www.state.gov	
7	Antigua and Barbuda	http://www.state.gov	/img/14/57692/antigu	http://www.state.gov	
8	Argentina	http://www.state.gov	/img/14/57696/argen	http://www.state.gov	
9	Armenia	http://www.state.gov	/img/14/55802/armer	http://www.state.gov	
10	Aruba	http://www.state.gov	/img/14/57974/nllgfla	http://www.state.gov	
11	Australia	http://www.state.gov	/img/13/56062/aslgfl	http://www.state.gov	
12	Austria	http://www.state.gov	/img/14/57821/aulgfl	http://www.state.gov	
13	Azerbaijan	http://www.state.gov	/img/14/55803/azerb	http://www.state.gov	

Useful Links

See also http://infoguides.gmu.edu/data-work/scraping

Selectors

XPath -- https://www.w3schools.com/xml/xpath_syntax.asp and https://www.w3.org/TR/xpath/all/
CSS -https://www.w3schools.com/cssref/css_selectors.asp and http://learnlayout.com/display.html
JQuery - https://www.w3schools.com/jquery/jquery_ref_selectors.asp
Regular Expressions - http://infoguides.gmu.edu/data-work/regex

JQuery vs XPath: https://www.ibm.com/developerworks/library/x-xpathjquery/index.html

https://genius.com/Mat-brown-xpath-is-actually-pretty-useful-once-it-stops-being-confusing-annotated
CSS vs XPath: https://en.wikibooks.org/wiki/XPath/CSS_Equivalents

XPath Hierarchies

http://dh.obdurodon.org/introduction-xpath.xhtml
http://dh.newtfire.org/explainXPath.html
https://dpastov.blogspot.com/2015/10/preceding-sibling-and-following-signling-xpath.html

Reference

Tester: https://extendsclass.com/xpath-tester.html or http://xpather.com/

Node / Path

element	div p a span	
attribute	@class @href	
text	text()	*this*
any *element*	*	
any *attribute*	@*	
either *element* or *text*	node()	

Axis / Direction

self::	**.**
parent::	**..**
child::	**/** (*also for attributes or text*)
descendant::	**//**
ancestor::	
attribute::	**@**
preceding-sibling::	
following-sibling::	
preceding::	
following::	

Predicates / Filter

div[**1**]
div[**b**]
div[@id = "me"]
div[@id **eq** "me"]
div[@class **!=** "me"]
div[@class **ne** "me"]
div[@class = ("you", "me")]
div[@class="you" **or** @class="me"]
div[@class="you"] **|** div[@class="me"]
div[@class="you" **and** @id="me"]

Functions

contains(@class, "me")
starts-with(@class, "me")
ends-with(@class, "me")
matches(@id, "[0-9][a-f]+", "i")
td[last()]
not(@id)
concat(" ",@class," ")
normalize-space(./div/p)