Adventures in Babysitting: Introduction to Web Scraping in Python

Julia Piaskowski 2020/02/09 https://github.com/jpiaskowski/pycas2020_web_scraping

But, Who Actually Reads These A to Z?

(spoiler: not me)



me and my programming books

The Main Things to Know in a Web Scraping Project:

- Is it worth the trouble?
- Is it ethical?
- \cdot Tools available in <code>BeautifulSoup</code> and <code>requests</code>
- \cdot What to look for in html code
- Parsing json objects with j son
- Rudimentary pandas skills
- <pro-tip> All you need to know about html is how tags work </pro-tip>

What to Look for in a Scraping Project:

- A sizeable amount of structured data with a regular repeatable format.
- Identical formating is not required, but the more edge cases present, the more complicated the scraping will be.



Ethics in Scraping



Accessing vast troves of information can be intoxicating.

Just because it's possible doesn't mean it should be done

Legal Considerations

(note: I have zero legal training - this is not legal advice!)

- Are you scraping copyrighted material?
- Will your scraping activity compromise individual privacy?
- Are you making a large number of request that may overload or damage a server?
- Is it possible the scraping will expose intellectual property you do not own?
- Are there terms of service governing use of the website and are you following those?
- Will your scraping activities diminish the value of the original data?

Dollar Stores are Taking Over the World!



Store in Cascade, Idaho

Goal: Extract addresses for all Family Dollar stores in Idaho.

The Starting Point:

https://locations.familydollar.com/id/

•••	Family Dollar Store Locations in × +			
$\leftarrow \rightarrow$	C â familydollar.com/locations/id/			
	FIND A STORE Your Store: 12978 Hwy 12		Sign	n In or C
	FAMILY® DOLLAR.	What can we help you find ${f Q}$		Get St
	OUR DEPARTMENTS V Explore Mor	re 🗸	Shop 🖉	🆗 DOI

Select a state > Idaho (ID)

Family Dollar Store Locations in in Idaho

Ci	ties			
A				
	Aberdeen	American Falls	Arco	Ashton
B				
	Bellevue	Blackfoot	Boise	Buhl

Step 1: Load the Libraries

import requests # for making standard html requests
from bs4 import BeautifulSoup # magical tool for parsing html data
import json # for parsing data
from pandas import DataFrame as df # data organization

Step 2: Grab Some Data from Target Web Address

```
page = requests.get("https://locations.familydollar.com/id/")
soup = BeautifulSoup(page.text, 'html.parser')
```

Beautiful Soup will take html or xml content and transform it into a complex tree of objects. Here are several common types:

- BeautifulSoup the soup (the parsed content)
- Tag main type of bs4 element you will encounter
- \cdot NavigableString string within a tag
- Comment special type of NavigableString

Step 3: Determine How to Extract Relevant Content from bs4 Soup

This process can be frustrating.



Step 3: Finding Content...

- \cdot Start with one representative example and then scale up
- \cdot Viewing the page's html source code is essential
 - Run at your own risk:

print(soup.prettify())

Step 3: Finding Content...

9358

9372

٠

• It is usually easiest to browse via "View Page Source":

<#3 style="font-size:16px; font-weight:bold; padding-bottom:10px; margin-bottom:20px; border-bottom:1px solid #cdcdcd;">Cities</#3>
<h4 id="aAnchor">A</h4>
<h4 id="aAnchor">A</h4 id= aAnchor">A</h4 id= aAnchor" id= aAn

What attribute or tag sets your content apart from the rest?

Step 3: Finding Content by Searching

Searching for 'href' does not work.

```
dollar_tree_list = soup.find_all('href')
dollar_tree_list
```

[]

But searching on a specific class is often successful:

```
dollar_tree_list = soup.find_all(class_ = 'itemlist')
for i in dollar_tree_list[:2]:
    print(i)
```

<div class="itemlist"><a dta-linktrack="City index page - Aberdeen" href="https://locations
<div class="itemlist"><a dta-linktrack="City index page - American Falls" href="https://locations")</pre>

Step 3: Finding Target Content by Using 'contents'

```
type(dollar_tree_list)
## <class 'bs4.element.ResultSet'>
len(dollar_tree_list)
## 48
```

Next, extract contents from this BeautifulSoup "ResultSet".

```
example = dollar_tree_list[2] # Arco, ID (single representative example)
example_content = example.contents
print(example content)
```

[<a dta-linktrack="City index page - Arco" href="https://locations.familydollar.com/id/arco,</pre>

Step 3: Finding Content in Attributes

Find out what attributes are present in the contents:

Note: contents usually return a list of exactly one item, so the first step is to index that item.

```
example_content = example.contents[0]
example_content.attrs
```

{'dta-linktrack': 'City index page - Arco', 'href': 'https://locations.familydollar.com/id/

Extract the relevant attribute:

```
example_href = example_content['href']
print(example_href)
```

https://locations.familydollar.com/id/arco/

Step 4: Extract the Relevant Content

```
city_hrefs = [] # initialise empty list
```

```
for i in dollar_tree_list:
    cont = i.contents[0]
    href = cont['href']
    city_hrefs.append(href)
```

```
# check to be sure all went well
for i in city_hrefs[:2]:
    print(i)
```

```
## https://locations.familydollar.com/id/aberdeen/
## https://locations.familydollar.com/id/american-falls/
```

Result: a list of URL's of Family Dollar stores in Idaho to scrape

Repeat Steps 1-4 for the City URLs

page2 = requests.get(city_hrefs[2]) # representative example soup2 = BeautifulSoup(page2.text, 'html.parser')

Getting Started	$\bigcirc \ \ \bigcirc \ \ \square \ \ \bigcirc \ \ \bigcirc \ \ \square \ \ \bigcirc \ \ $
https://www.mozilla.org/en-US/firefox/central/	Q Search HTML + 🗡
FREE Shipping to Your Store: 12978 Hwy 12	<pre><!DOCTYPE html> <html lang="en"> event scroll </html></pre>
	<pre><meta content="text/html; charset=utf-8" http-equiv="content-type"/> <title>Family Dollar Store Locations in Arco, ID</title> <meta content="width=device-width, initial-scale=1" name="viewport"/> <meta content="Find nearby Family Dollar Store</pre></th></tr><tr><th>SMART C@UPONS Get Started & Save! >
Select a state > Idaho (ID) > Arco
Family Dollar Store Locations in Arco, ID</th><th rowspan=2 colspan=6><pre><meta name= description content= rind nearby ramity bottar store
locations in Arco, ID to shop for groceries, housewares, toys, pet
supplies, and more." name="description"/></pre>
Click on Store Details for Hours and More Information Family Dollar #9143 157 W Grand Avenue Arco, ID 83213 US PHONE: 208-881-5738 View Store Details	<pre>{ "@context":"https://schema.org", "@type":"Schema Business Type", "name": "Family Dollar #9143", "address": { "@type":"PostalAddress", "streetAddress":"157 W Grand Avenue", "addressLocality":"Arco", "addressRegion":"ID", "postalCode":"83213", "addressCountry":"US" } "containedIn":"", "branchOf": { "name":"Family Dollar", "url": "https://www.familydollar.com/" }, "url":"https://locations.familydollar.com/id/arco/29143/", "telephone":"208-881-5738", "image": "//hosted.where2getit.com /familydollarstore/images/storefront.png" } </pre>
	<pre>> <script> 20/33 <!from client's header/footer file> 20/33 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"> <meta charset="utf-8"></pre></th></tr></tbody></table></script></pre>

Extract Address Information

```
from type="application/ld+json"
```

```
arco = soup2.find_all(type="application/ld+json")
print(arco[1])
```

```
## <script type="application/ld+json">
##
    {
##
      "@context":"https://schema.org",
##
      "@type":"Schema Business Type",
      "name": "Family Dollar #9143",
##
      "address":{
##
##
        "@type":"PostalAddress",
        "streetAddress":"157 W Grand Avenue",
##
##
        "addressLocality":"Arco",
        "addressRegion":"ID",
##
        "postalCode":"83213",
##
        "addressCountry":"US"
##
##
      },
```

```
## "containedIn":"",
```

Use 'contents' to Find Address Information

Extract the contents (from the second list item) and index the first (and only) list item:

```
arco_contents = arco[1].contents[0]
arco_contents
```

```
## '\n\t{\n\t "@context":"https://schema.org",\n\t "@type":"Schema Business Type",\n\t "name
```

Next, convert to a json object: (these are way easier to work with)

arco_json = json.loads(arco_contents)

Extract Content from a json Object

A json object is a dictionary:

type(arco_json)

<class 'dict'>

print(arco_json)

{'@context': 'https://schema.org', '@type': 'Schema Business Type', 'name':
'Family Dollar #9143', 'address': {'@type': 'PostalAddress', 'streetAddress': '157 W
Grand Avenue', 'addressLocality': 'Arco', 'addressRegion': 'ID', 'postalCode':
'83213', 'addressCountry': 'US'}, 'containedIn': '', 'branchOf': {'name': 'Family
Dollar', 'url': 'https://www.familydollar.com/'}, 'url':
'https://locations.familydollar.com/id/arco/29143/', 'telephone': '208-881-5738',
'image': '//hosted.where2getit.com/familydollarstore/images/storefront.png'}

Extract Content from a json Object

```
arco_address = arco_json['address']
arco_address
```

{'@type': 'PostalAddress', 'streetAddress': '157 W Grand Avenue', 'addressLocality': 'Arco', 'addressRegion': 'ID', 'postalCode': '83213', 'addressCountry': 'US'}

Step 5: Put It All Together

Iterate over the list store URLs in Idaho:

```
locs_dict = [] # initialise empty list
for link in city_hrefs:
    locpage = requests.get(link)  # request page info
    locsoup = BeautifulSoup(locpage.text, 'html.parser')
        # parse the page's content
    locinfo = locsoup.find_all(type="application/ld+json")
        # extract specific element
    loccont = locinfo[1].contents[0]
        # get contents from the bs4 element set
    locjson = json.loads(loccont)  # convert to json
    locaddr = locjson['address'] # get address
    locs_dict.append(locaddr) # add address to list
```

Step 6: Finalise Data

```
locs_df = df.from_records(locs_dict)
locs_df.drop(['@type', 'addressCountry'], axis = 1, inplace = True)
locs_df.head(n = 5)
```

##	streetAddress	addressLocality	addressRegion	postalCode
## 0	111 N Main Street	Aberdeen	ID	83210
## 1	253 Harrison St	American Falls	ID	83211
## 2	157 W Grand Avenue	Arco	ID	83213
## 3	177 Main Street	Ashton	ID	83420
## 4	747 N. Main St.	Bellevue	ID	83313

Results!!



df.to_csv(locs_df, "family_dollar_ID_locations.csv", sep = ",", index = False)

A Few Words on Selenium

"Inspect Element" provides the code for what is displayed in a browser.

NORTH BEND, OK					
0	Q Portland) 1 of 2 + ,			
ONTARIO, OR	<pre>>> <div> </div> >> <div> </div></pre>				
OREGON CITY, OR	▶ <div> </div> ▶ <div> </div>				
	▼ <div></div>				
P	<pre>▶ <div id="P"></div></pre>				
	<pre></pre>				
PORTLAND, OR	▼ 				
FUNILAND, UN	::marker				
R	<pre><a href="/storelocator</pre></td><td></td></tr><tr><td></td><td>/find.jsp?requestType=locator&state=OR&city=PORTLAND&from= event event</td><td>localSearch" name="PORTLAND" title="PORTLAND"></pre>				
REDMOND, OR	event event react-text: 282 PORTLAND /react-text /react-text				
ROSEBURG, OR					
S	/react-text				
5	react-text: 284				
	OR /react-text				
SAINT HELENS, OR					
SALEM, OR					

A Few Words on Selenium

"View Page Source" - provides the code for what requests will obtain

ightarrow $ ightarrow$ $ ig$	⊘ ☆	<u>↓</u> III/		э 🚽	Z
1					
2 <html lang="en"></html>					
<pre>3</pre>)===b dialogargu	onte)rot	urn		
navigator.cookieEnabled;document.cookie=" dTCookie=1";yar a=-11==document.cookie.indexOf(" dTCookie");doc					
					u, 01-
Jan-1970 00:00:01 GMT"; return a) if (window.dT_) window.console&&window.console.log("Duplicate agent injection	n detected, turn	.ng orr re	aundant	2	
initConfig.");else{var b="undefined"!==typeof window;window;self;a()&&(window.dT_ (window.dT_=					
{cfg:"domain=walgreens.com reportUrl=https://www.walgreens.com/dthandler					
/ uam=1 app=0eed2717dafcc06d cors=1 featureHash=A2SVfhqru rdnt=1 uxrgce=1 srms=1,1,,, uxrgcm=100,25,300,3;					
100,25,300,3 dpvc=1 md=1=cwag sid,2=cwrp dt tracker lastModification=1578928765992 dtVersion=1018119119154	4660 tp=500,50,0	1 uxdcw=	1500 agr	entUri=	
/dthandler/ruxitagentjs A2SVfhgru 10181191119154660.js auto=1 domain=walgreens.com rid=RID 1638716348 rpid=	=-829757625 app=	eed2717d	afcc06d'	· ,	
4 iCE:a}))}})();}).call(this);					
6 (function(){(function(){function Mb(){return ta?new ta:ua?new ua("MSXML2.XMLHTTP.3.0"):d.XMLHttpRequest?new	d XMLHttpReque	t · new			
d.ActiveXObject("MSXML2.XMLHTP.3.0") function Nb() {ua-ta=void 0} function u() {var a=0;	" uninneepheque	C.IIC.			
try(a-d.performance.tming.navigationStart+Math.floor(d.performance.now())}catch(b){return 0>=a isNaN(a)	Ligzinite(a)2(act mim.		unatio
ba(a,b){function c(){delete la[g];a.apply(this,arguments)}for(var e=[],I=2;I <arguments.length;i++)e[i-2]=arguments< td=""><td>rguments[1];var</td><td>; appiy i</td><td>in varg=</td><td><pre>sva.app.</pre></td><td>ıy(a,</td></arguments.length;i++)e[i-2]=arguments<>	rguments[1];var	; appiy i	in varg=	<pre>sva.app.</pre>	ıy(a,
7 [c,b].concat(e)):g=va(c,b);la[g]=!0;return g}function \$a(a){delete la[a];"apply"in ca?ca.call(d,a):ca(a)}fu					
<pre>c<arguments.length;c++)b[c-1]=arguments[c];a.push.apply(a,b)}function ab(a){k(da,a)}function="" ob(a){for(var<="" pre=""></arguments.length;c++)b[c-1]=arguments[c];a.push.apply(a,b)}function></pre>					
break}function Pb(){return da}function Qb(a,b){return bb(a,b)}function Rb(a,b){a=new Sb([a],{type:b});retu	urn Tb(a)}functi	n Ub(a,b)){returr	1 cb?net	w
cb(a,b):void 0}function Vb(a){"function"===typeof a&&k(db,a)}function Wb(){return db}					
<pre>8 function Xb(){return Da}function eb(a){return function(){for(var b=[],c=0;c<arguments.length;c++)b[c]=argum< pre=""></arguments.length;c++)b[c]=argum<></pre>					
b[0] !la[b[0]])try{return a.apply(this,b)}catch(e){return a(b[0])}}function Yb(){ma&&(d.clearTimeout=ca,c	d.clearInterval=	a,ma=!1)'	functio	on ea(a	,b)
{return isNaN(a) isNaN(b)?0:Math.floor(Math.random()*(b-a+1))+a}function v(a,b){return parseInt(a,b 10)}	function p(a,b,c	{void 0=	==C&& (C=	=0);var	e=-1;
b&&a&&a.indexOf&&(e=a.indexOf(b,c));return e}function fb(a){return document.getElementsByTagName(a)}					
9 function qb(a) {var b=a.length; if("number"===typeof b)a=b; else{for(var b=0,c=2048;a[c-1];)b=c,c=c; for(var e	e=7:1 <c-b:)e=(c+< td=""><td>)/2,a[e-</td><td>11?b=e:</td><td>с=е;</td><td></td></c-b:)e=(c+<>)/2,a[e-	11?b=e:	с=е;	
a=a[e]?c:b}return a)function Zb(a) {a=encodeURIComponent(a);var b=[];if(a) for(var c=0;c <a.length;c++) e="</td" {var=""><td></td><td></td><td></td><td></td><td></td></a.length;c++)>					
b.join("") function R(a) {if (!a) return"; var b=d.crypto d.msCrypto; if (b) {var c=new Int8Array(a); b.getRandom					
b+)c.push(ea(0,32));a=[];for(b=0;b <c.length;b++){var e="Math.abs(c[b]\$32);a.push(String.fromCharCode(e+</td"><td>araideb(0))erbe</td><td>01(0 [])</td><td>, 0,2-4,</td><td>(</td><td></td></c.length;b++){var>	araideb(0))erbe	01(0 [])	, 0,2-4,	(
() (9>=e?48:55)))return a.join("))function hb(){return((1d.console) 1d.console.log)}function ac(){try{bc.app	alw(d parant arg	montalla	at ab (a)	()) fund	tion
cc){try{dc.apply(d.top,arguments)}catch(a){}function ec(a){var b=Array.prototype.slice.cal(arguments,1);					CION
cc(){try{dc.appry(d.top,arguments);catch(a){};runction ec(a){var b=Array.protocype.sites.car(arguments,r); c(a){var b=Array.prototype.site.call(arguments,1);try{hc.appry(a,b){catch(c){}} function J(){return d.dT					- 4
	function ic(){r	curn A}IU	inction]C(){I	eturn
ib}function kc(){return jb}function lc(){return wa}function kb(){return"dtAdk"}					
<pre>L function mc(){return fa}function lb(a){-1<d.dtio(a,"^")&&< pre=""></d.dtio(a,"^")&&<></pre>					
(a=a.split("^^").join("^"),a=a.split("^dq").join(""),a=a.split("^tb").join(">"),a=a.split("^1b").join("					
<"),a=a.split("^p").join(" "),a=a.split("^e").join("="),a=a.split("^s").join(";"),a=a.split("^c").join(",")					
a}function nc(){return S}function oc(a){S=a}function mb(a){var b=d.dT_,c=b.scv("rid"),b=b.scv("rpid");c&{(a)}	a.rid=c);b&&(a.r	id=b)}fur	action r	1b(a){i	f(a.xh
{a=lb(a.xb);try{S=new RegExp(a)}catch(b){}}					
2 function $ob(a)$ {var $b=$ }; a=a.split(" "); for(var c=0; c <a.length; 2="=e.length&&(b[e=a])" b="}" c++)="" e='a[c].split("=");' td="" {var="" {var<=""><td>e[0]]=decodeURIC</td><td>mponent(</td><td>≥[1].rep</td><td>place(/</td><td>\+/g,'</td></a.length;>	e[0]]=decodeURIC	mponent(≥[1].rep	place(/	\+/g,'
")))}return b}function Fa(){var a=n("csu");return(a.indexOf("dbg")===a.length-3?a.substr(0,a.length-3):a)+'	" "+n("app")+" S	ore"}fund	ction xa	a(a,b,c) {b=b
{};a=a.split(" ");for(var e=0;e <a.length;e++){var d='a[e],g=p(a[e],"=");-1===g?b[d]="1":(d=a[e].substring(0,</td'><td>g),b[d]=a[e],su</td><td>string(g</td><td>+1,a[e]</td><td>.length</td><td>))}!c8</td></a.length;e++){var>	g),b[d]=a[e],su	string(g	+1,a[e]	.length))}!c8
(c=b,a=c.spc)&&(e=document.createElement("textarea"),		5.05		5	
3 e.innerHTML=a,c.spc=e.value);return b}function T(a){return a in f?f[a]:ya[a]}function 1(a){a=T(a);return "fa	alse"===a "0"==	a?!1:!!a	functio	on K(a)	{var
b=v(T(a)); isNaN(b)&&(b=ya[a]); return b}function n(a) {return String(T(a)] ["")} function pc(a,b) f[a]=b}function					
b=location.hostname;return b&&a?b===a -1!==b.indexOf("."+a,b.length-("."+a).length):10}function	F=(=)(100u1n	, runo.	q	, (vur	
$G_{(a)} \{f[a]=0 > p(f[a], "#"+a, to UpperCase())?f[a]: "} function Ha(a) \{var b=a.aqentUri; b\&-1$	41	([a=7A-7	0-91*1	10-91-	
antarial of partial a second percase()) and a second secon		.(La-2A-2	TA - 2 1) T	-10-21+	
Q Portland N V Highlight All Match Case Whole Words Phrase not found					

There are plugins modifying the source code - so, it should be accessed *after* the page has loaded in a browser.

A Few Words on Selenium

- Requires a webdriver to retrieve the content
- $\cdot\;$ It actually opens a web browser, and this info is collected
- Selenium is powerful it can interact with loaded content in many ways
- After getting data, continue to use BeautifulSoup as before

```
url = "https://www.walgreens.com/storelistings/storesbycity.jsp?requestType=locator&state=ID"
driver = webdriver.Firefox(executable_path = 'mypath/geckodriver.exe')
driver.get(url)
soup_ID = BeautifulSoup(driver.page_source, 'html.parser')
store_link_soup = soup_ID.find_all(class_ = 'col-xl-4 col-lg-4 col-md-4')
```

The Penultimate Slide

Read the Manuals

- https://beautifulsoup-4.readthedocs.io/en/latest/
- https://selenium.dev/

This talk available at:

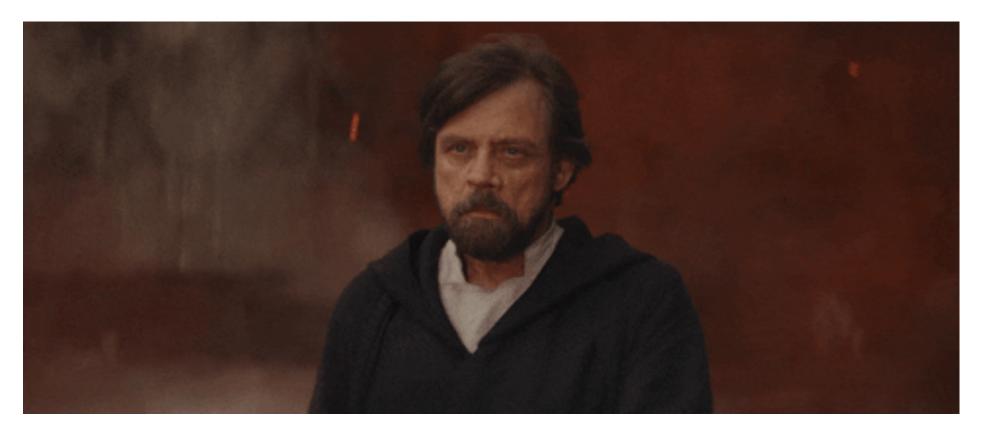
https://github.com/jpiaskowski/pycas2020_web_scraping



Perservere

~ After Becoming a Web Scraping Master ~

https://github.com/jpiaskowski/pycas2020_web_scraping



Bonus Slide!

