

*DataScience for Development and Social Change, 2015*

---

# Scraping Development Data

---

When it just isn't available in  
a handy CSV file...

---

# Don't scrape if you don't have to

---

- ❖ Small number of data points: type!
  - ❖ into a spreadsheet
- ❖ Larger datasets: search!
  - ❖ the website for file copies of the data
  - ❖ the website for an api (`http://api.theirsitename.com/`, `http://theirsitename.com/api`, Google “`site:theirsitename.com api`”)
  - ❖ connected sites
  - ❖ `scraperwiki.com`, `datahub.io` etc. for other peoples' scrapers

---

# Design first

---

- ❖ What do you need to scrape?
  - ❖ Which data values
  - ❖ From which formats (html table, excel, pdf etc)
- ❖ How often?
  - ❖ Is dataset regularly updated, or is once enough?
- ❖ Maintenance?
  - ❖ How will you make data available to other people?
  - ❖ How do you tell people if the data is updated?
  - ❖ Who could edit your code next year (if needed)?



You've got a file...

---

# A CSV file

---

- ❖ Lucky you: most packages read CSV.
- ❖ You still have to check for e.g. weird values (people aged -1 etc), and maybe do some cleaning, but your scraping is done.

---

# A file with lots of braces {}

---

- ❖ Probably a JSON file (.json)
- ❖ Not all packages will read json
- ❖ Converting JSON to CSV:
  - ❖ Use a conversion website (e.g. <http://www.convertcsv.com/json-to-csv.htm>)
  - ❖ Write some Python code
- ❖ Gotchas: JSON is structured and might not fit into CSV's row-by-column format without manipulation

```
{  
  "firstName": "John",  
  "lastName": "Smith",  
  "isAlive": true,  
  "age": 25,  
  "height_cm": 167.6,  
  "address": {  
    "streetAddress": "21 2nd Street",  
    "city": "New York",  
    "state": "NY",  
    "postalCode": "10021-3100"  
  },  
  "phoneNumbers": [  
    {  
      "type": "home",  
      "number": "212 555-1234"  
    },  
    {  
      "type": "cell",  
      "number": "212 555-1234"  
    }  
  ]  
}
```



---

# A file with lots of brackets <>

---

```
▼<note>  
  <to>Tove</to>  
  <from>Jani</from>  
  <heading>Reminder</heading>  
  <body>Don't forget me this weekend!</body>  
</note>
```

- ❖ Probably XML (might also be html: look for things like <body>)
- ❖ Converting XML to CSV:
  - ❖ Use a conversion website, e.g. <http://www.convertcsv.com/xml-to-csv.htm>
  - ❖ Write code
- ❖ Gotchas: also structured and might not fit easily into row-column form

---

# A PDF file

---

- ❖ Many PDF files \*can\* be scraped
  - ❖ Check by opening the file, then trying to select and cut text from it.
- ❖ Converting PDF to CSV (if you can select text):
  - ❖ Use a conversion website or program (e.g. Cometdocs or Pdftables)
  - ❖ Write code: the Tabula library is helpful.
- ❖ Converting PDF to CSV (if you can't select text):
  - ❖ You \*could\* try OCR (optical character recognition) software
  - ❖ But you'll probably have to either type text in, or turksource it



# Map files can contain data too

- ❖ File formats: shapefiles (.shp), geojson (.json), geotiffs (.tif) etc.
  - ❖ Often contain useful tabular data
- ❖ Converting map files to CSV:
  - ❖ 1) Open file in Quantum GIS (QGIS)
    - ❖ Save as CSV
  - ❖ 2) Use the ogr2ogr command, e.g.
    - ❖ `ogr2ogr -f CSV outfilename.csv infilename.shp -lco GEOMETRY=AS_XYZ`



WKT	ID_0	ISO	NAME_0	ID_1	NAME_1	VARNAM	NL_NAME	HASC_1	CC_1	TYPE_1	ENGTYPE	VALIDFR
POLYGON	229	TZA	Tanzania	3052	Arusha			TZ.AS		Mkoa	Region	between
MULTIPO	229	TZA	Tanzania	3053	Dar-Es-S	Dar es Salaam		TZ.DS		Mkoa	Region	9740101
POLYGON	229	TZA	Tanzania	3054	Dodoma			TZ.DO		Mkoa	Region	between
MULTIPO	229	TZA	Tanzania	3055	Iringa			TZ.IR		Mkoa	Region	between
POLYGON	229	TZA	Tanzania	3056	Kagera			TZ.KR		Mkoa	Region	between
MULTIPO	229	TZA	Tanzania	3057	Kaskazini	Pemba North		TZ.PN		Mkoa	Region	~1982
MULTIPO	229	TZA	Tanzania	3058	Kaskazini	Zanzibar North		TZ.ZN		Mkoa	Region	~1967

You've found an API...

---

# Using APIs without coding

---

- ❖ RESTful APIs:
  - ❖ API address(e.g. [api.worldbank.org](http://api.worldbank.org))
  - ❖ plus a description of what you want (e.g. countries / all / indicators / SP.RUR.TOTL.ZA)
  - ❖ plus some more details (e.g. date=2000:2015, format=csv)
- ❖ Gives you an address you can use to get your datafile:
  - ❖ <http://api.worldbank.org/countries/all/indicators/SP.RUR.TOTL.ZS?date=2000:2015&format=csv>
- ❖ format=xxx is a common detail in many APIs - try adding “&format=json” to the line above to get a JSON file instead!



---

# APIs with coding

---

- ❖ The Python Requests library is your friend...

```
import requests
worldbank_url = "http://api.worldbank.org/countries/all/indicators/SP.RUR.TOTL.ZS?date=2000:2015"
r = requests.get(worldbank_url)
```

or

```
import requests
r = requests.get('https://api.github.com/user', auth=('yourgithubname',
'yourgithubpassword'))
dataset = r.text
```

- ❖ If anything goes wrong, check `r.status_code`

No file, no API, just a website...

---

# Scraper Options

---

- ❖ 1) Use a conversion tool (see e.g. <http://www.capterra.com/data-extraction-software/>)
- ❖ 2) Write your own scraper code



---

# Here's your Python toolkit

---

- ❖ Webpage-grabbing libraries:

- ❖ requests
- ❖ cookielib
- ❖ mechanize

- ❖ Element-finding libraries:

- ❖ BeautifulSoup
- ❖ lxml.html
- ❖ cssselect

# The Webpage

← → ↻ cdc.khmer.biz/Reports/reports\_by\_sector\_list.asp?OtherSubSector=101&status=0 ☆ 🖨️ ☰



CAMBODIAN REHABILITATION AND DEVELOPMENT BOARD  
COUNCIL FOR THE DEVELOPMENT OF CAMBODIA

## THE CAMBODIA ODA DATABASE

Reporting Year: 2014



Kingdom  
of Cambodia


Welcome to Guest/ Exit

List of ProjectsQueryReportsAnnexUser's GuideAbout

🖨️ Print

### SUMMARY

- ▶ Donor
- ▶ Sector
- ▶ Province
- ▶ Duration
- ▶ Type of Assistance
- ▶ Term of Assistance
- ▶ Project Status
- ▶ Implementing Agency
- ▶ Thematic Marker
- ▶ Technical Working Group (TWG)
- ▶ Program-Based Approach
- ▶ Last Update



### Project Listing by Sector and Status

Sector/Sub-Sector:

Project Status:  🔍

1 - 2 - 3 - 4 - 5 - 6

No	Donor	Official Title	Program Number	Start Date	Completion Date	Budget	Project Status
▶ Health( 298 Projects)							
▶ Hospitals( 55 Projects )							
1	Belgium	Provision of Basic Health Services in the Province of Kampong Cham	N/A	1-Nov-2004	31-Oct-2008	7,130,000 EUR	Completed
2	Belgium	Provision of Basic Health Services in the Provinces of Siem Reap and Otdar Meanchey	NI18955/11	1-Jun-2004	30-May-2008	13,280,000 EUR	Completed
3	Denmark	UNDP Cambodia Climate Change Alliance Trust Fund	104.Cambodia.16	9-Dec-2009	9-Feb-2011	550,000 USD	Completed

### LISTING

- ▶ Donor
- ▶ Sector



# The Webpage's HTML Code

```
<td id="report_listing_by_sector" colspan="2" align="center" height="20">  
▶<table width="100%" border="0" cellpadding="2" cellspacing="0" bordercolor="#000000">...</table>  
▼<table width="100%" border="1" bordercolor="#CCCCCC" style="border-collapse: collapse"  
cellpadding="2" class="fontNormal" cellspacing="1">  
▼<tbody>  
▶<tr bgcolor="#006699" class="fontNormalBlue">...</tr>  
▶<tr bgcolor="#FFFFF0" valign="center" class="fontNormal" height="25">...</tr>  
▼<tr valign="center" class="fontNormal" height="25">  
▼<th colspan="8" align="left" bgcolor="#EFEFEF">  
    "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;"  
      
    <font color="black">Hospitals(  
  
        55 Projects  
    )</font>  
    </th>  
    </tr>  
▼<tr class="valign=top" height="20">  
    <td align="center">  
        1  
    </td>  
    <td>Belgium</td>  
▼<td>  
    <a target="_blank" href="../reports/additional_information/project_readonly_page01.asp?  
Record_Id=Belgium%2D2006%2DSVIM1722&DonorName=Belgium">Provision of Basic Health  
Services in the Province of Kampong Cham</a>  
    </td>  
    <td align="center">N/A</td>  
    <td align="center" nowrap>1-Nov-2004</td>  
    <td align="center" nowrap>31-Oct-2008</td>  
    <td align="right" nowrap>  
        7,130,000 EUR  
    </td>  
    <td align="center">Completed</td>  
    </tr>  
▶<tr class="bgGray" valign="top" height="20">...</tr>  
▶<tr class="valign=top" height="20">...</tr>
```



---

# Getting the Webpage's html code

---

- ❖ Mechanize
  - ❖ Extracts raw html code from a webpage
  - ❖ Needs lots of setup (see last slide), but useful when Requests fails (e.g. on the CDC website timeouts)
- ❖ CookieLib
  - ❖ gets and sets webpage cookies

---

# Getting the Webpage's html code

---

- ❖ `import mechanize`
- ❖ `import cookielib`
  
- ❖ `br = mechanize.Browser()`
- ❖ `cj = cookielib.LWPCookieJar()`
- ❖ `br.set_cookiejar(cj)`
- ❖ `br.set_handle_equiv(True)`
- ❖ `br.set_handle_gzip(True)`
- ❖ `br.set_handle_redirect(True)`
- ❖ `br.set_handle_referer(True)`
- ❖ `br.set_handle_robots(False)`
- ❖ `br.set_handle_refresh(mechanize._http.HTTPRefreshProcessor(), max_time=1)`
- ❖ `br.addheaders = [('User-agent', 'Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.1) Gecko/2008071615 Fedora/3.0.1-1.fc9 Firefox/3.0.1')]`
  
- ❖ `r = br.open("http://cdc.khmer.biz/index.asp?submit=Guest");` #have to do this, or get timeout message
- ❖ `r = br.open("http://cdc.khmer.biz/Reports/reports_by_sector_list.asp?OtherSubSector=101&status=0");`
- ❖ `htmlstring = r.read();`

---

# Getting the dataset out of the html

---

- ❖ lxml library:
- ❖ cssselect library:
  - ❖ Finds specific tags in a piece of html



# Getting the dataset out of the html

```
import lxml.html
import cssselect

#LXML version: find the list of pages and the data table
root = lxml.html.fromstring(htmlstring)
root.cssselect("td#report_listing_by_sector")
for el in root.cssselect("td#report_listing_by_sector tr"):
    print(el)
lbs = root.cssselect("td#report_listing_by_sector table");

#List of pages
for index,element in enumerate(lbs[0].cssselect("a")):
    print(element.text)
numpages = index+2; #first page doesn't have link, and index started at 0

#Data table - first row is header
datarows = lbs[1].cssselect("tr")
for datarow in range(1,len(datarows)):
    tds = datarow.cssselect("td");

    #Data structure is: header row, with subject row (text in td) then
    #subsubject row (text in th). Check for this.
    if len(tds) < 1:
        tds = datarow.cssselect("th");
        #Of course, it might just be a broken row...
        if len(tds) < 1:
            continue;

    if len(tds) == 1:
        #This is a subject row - save value, continue
        rowtext = tds[0].cssselect("img")[0].tail;
        print(rowtext);
```

---

# Writing that Scraper code

---

- ❖ Go to the webpage you're scraping.
  - ❖ Right-click on page, then "inspect element"
  - ❖ Use that to guide your scraper code
- ❖ Try things out: sometimes ".tail" will work when ".text" doesn't.
- ❖ Code incrementally: do a small bit at a time, not everything at once.
- ❖ And remember that this is going to take time. Keep yourself sane by referring back to your original design.

---

# Storing your dataset

---

- ❖ Python arrays, dictionaries and the CSV library are your friends: use them!