# Tutorial #1: Web Scraping treaties from the IEA Database

Alice Soldà

February 25, 2021

The IEA Database[1] currently contains over 1,300 MEAs, over 2,200 BEAs, 250 other environmental agreements, and over 90,000 individual country "membership actions" that are constantly being revised and updated.

Retrieving these texts manually can quickly become a hassle.

Luckily for us, the database has been modified in November 2020 in a way that enables to easily collect any treaty text included in the database using a simple Python code.

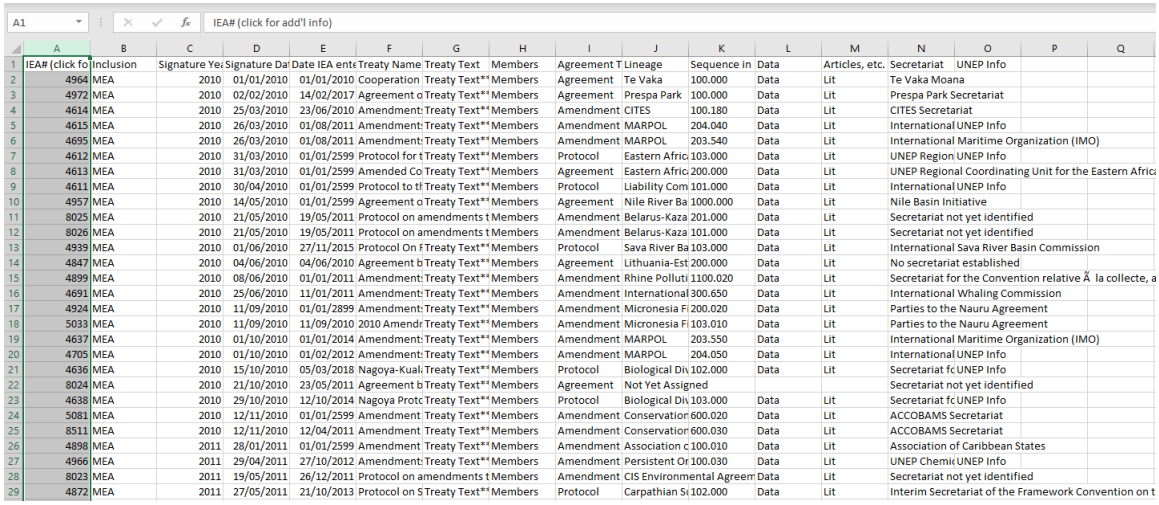
To do so, you will need to download and install Python (this tutorial was made using Python 3.7.3).

[1](c) Ronald B. Mitchell and the IEA Database Project, 2002-2020

# 1 Retrieving the relevant IDs

The first thing we need to do is to identify the IDs of our treaties of interest. Each treaty is given an ID (called Mitch ID) that links it to various sets of information (metadata, membership data, etc.). What matters to us here, is that a particular treaty text is linked to a unique treaty ID.

Identifying treaties IDs can be easily done by downloading the list of agreements in CSV format from the IEADB Agreement List and using filters on excel. You can already use some filters directly on the IEADB website. The treaties IDs are stored in the first column of the CSV file.



| IEA# (click for add'l info) | Inclusion | Signature Yea | Signature Dat | Date IEA ente | Treaty Name | Treaty Text | Members | Agreement T | Lineage | Sequence in | Data | Articles, etc. | Secretariat | UNEP Info |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4964 | MEA | 2010 | 01/01/2010 | 01/01/2010 | Cooperation | Treaty Text** | Members | Agreement | Te Vaka | 100.000 | Data | Lit | Te Vaka Moana | |
| 4972 | MEA | 2010 | 02/02/2010 | 14/02/2017 | Agreement o | Treaty Text** | Members | Agreement | Prespa Park | 100.000 | Data | Lit | Prespa Park Secretariat | |
| 4614 | MEA | 2010 | 25/03/2010 | 23/06/2010 | Amendment: | Treaty Text** | Members | Amendment | CITES | 100.180 | Data | Lit | CITES Secretariat | |
| 4615 | MEA | 2010 | 26/03/2010 | 01/08/2011 | Amendment: | Treaty Text** | Members | Amendment | MARPOL | 204.040 | Data | Lit | International | UNEP Info |
| 4695 | MEA | 2010 | 26/03/2010 | 01/08/2011 | Amendment: | Treaty Text** | Members | Amendment | MARPOL | 203.540 | Data | Lit | International Maritime Organization (IMO) | |
| 4612 | MEA | 2010 | 31/03/2010 | 01/01/2599 | Protocol for t | Treaty Text** | Members | Protocol | Eastern Africa | 103.000 | Data | Lit | UNEP Region | UNEP Info |
| 4613 | MEA | 2010 | 31/03/2010 | 01/01/2599 | Amended Co | Treaty Text** | Members | Agreement | Eastern Africa | 200.000 | Data | Lit | UNEP Regional Coordinating Unit for the Eastern Africa | |
| 4611 | MEA | 2010 | 30/04/2010 | 01/01/2599 | Protocol to th | Treaty Text** | Members | Protocol | Liability Com | 101.000 | Data | Lit | International | UNEP Info |
| 4957 | MEA | 2010 | 14/05/2010 | 01/01/2599 | Agreement o | Treaty Text** | Members | Agreement | Nile River Ba | 1000.000 | Data | Lit | Nile Basin Initiative | |
| 8025 | MEA | 2010 | 21/05/2010 | 19/05/2011 | Protocol on amendments t | Members | Amendment | Belarus-Kaza | 201.000 | Data | Lit | Secretariat not yet identified | | |
| 8026 | MEA | 2010 | 21/05/2010 | 19/05/2011 | Protocol on amendments t | Members | Amendment | Belarus-Kaza | 101.000 | Data | Lit | Secretariat not yet identified | | |
| 4939 | MEA | 2010 | 01/06/2010 | 27/11/2015 | Protocol On F | Treaty Text** | Members | Protocol | Sava River Ba | 103.000 | Data | Lit | International Sava River Basin Commission | |
| 4847 | MEA | 2010 | 04/06/2010 | 04/06/2010 | Agreement b | Treaty Text** | Members | Agreement | Lithuania-Est | 200.000 | Data | Lit | No secretariat established | |
| 4899 | MEA | 2010 | 08/06/2010 | 01/01/2011 | Amendment: | Treaty Text** | Members | Amendment | Rhine Polluti | 1100.020 | Data | Lit | Secretariat for the Convention relative Ã  la collecte, a | |
| 4691 | MEA | 2010 | 25/06/2010 | 11/01/2011 | Amendment: | Treaty Text** | Members | Amendment | International | 300.650 | Data | Lit | International Whaling Commission | |
| 4924 | MEA | 2010 | 11/09/2010 | 01/01/2899 | Amendment: | Treaty Text** | Members | Amendment | Micronesia Fi | 200.020 | Data | Lit | Parties to the Nauru Agreement | |
| 5033 | MEA | 2010 | 11/09/2010 | 11/09/2010 | 2010 Amendr | Treaty Text** | Members | Amendment | Micronesia Fi | 103.010 | Data | Lit | Parties to the Nauru Agreement | |
| 4637 | MEA | 2010 | 01/10/2010 | 01/01/2014 | Amendment: | Treaty Text** | Members | Amendment | MARPOL | 203.550 | Data | Lit | International Maritime Organization (IMO) | |
| 4705 | MEA | 2010 | 01/10/2010 | 01/02/2012 | Amendment: | Treaty Text** | Members | Amendment | MARPOL | 204.050 | Data | Lit | International | UNEP Info |
| 4636 | MEA | 2010 | 15/10/2010 | 05/03/2018 | Nagoya-Kuala | Treaty Text** | Members | Protocol | Biological Div | 102.000 | Data | Lit | Secretariat fo | UNEP Info |
| 8024 | MEA | 2010 | 21/10/2010 | 23/05/2011 | Agreement b | Treaty Text** | Members | Agreement | Not Yet Assigned | | | | Secretariat not yet identified | |
| 4638 | MEA | 2010 | 29/10/2010 | 12/10/2014 | Nagoya Protc | Treaty Text** | Members | Protocol | Biological Div | 103.000 | Data | Lit | Secretariat fo | UNEP Info |
| 5081 | MEA | 2010 | 12/11/2010 | 01/01/2599 | Amendment | Treaty Text** | Members | Amendment | Conservation | 600.020 | Data | Lit | ACCOBAMS Secretariat | |
| 8511 | MEA | 2010 | 12/11/2010 | 12/04/2011 | Amendment | Treaty Text** | Members | Amendment | Conservation | 600.030 | Data | Lit | ACCOBAMS Secretariat | |
| 4898 | MEA | 2011 | 28/01/2011 | 01/01/2599 | Amendment | Treaty Text** | Members | Amendment | Association c | 100.010 | Data | Lit | Association of Caribbean States | |
| 4966 | MEA | 2011 | 29/04/2011 | 27/10/2012 | Amendment: | Treaty Text** | Members | Amendment | Persistent Or | 100.030 | Data | Lit | UNEP Chemi | UNEP Info |
| 8023 | MEA | 2011 | 19/05/2011 | 26/12/2011 | Protocol on amendments t | Members | Amendment | CIS Environmental Agreem | Data | Lit | Secretariat not yet identified | | | |
| 4872 | MEA | 2011 | 27/05/2011 | 21/10/2013 | Protocol on S | Treaty Text** | Members | Protocol | Carpathian Si | 102.000 | Data | Lit | Interim Secretariat of the Framework Convention on t | |

# 2 Retrieving the relevant texts

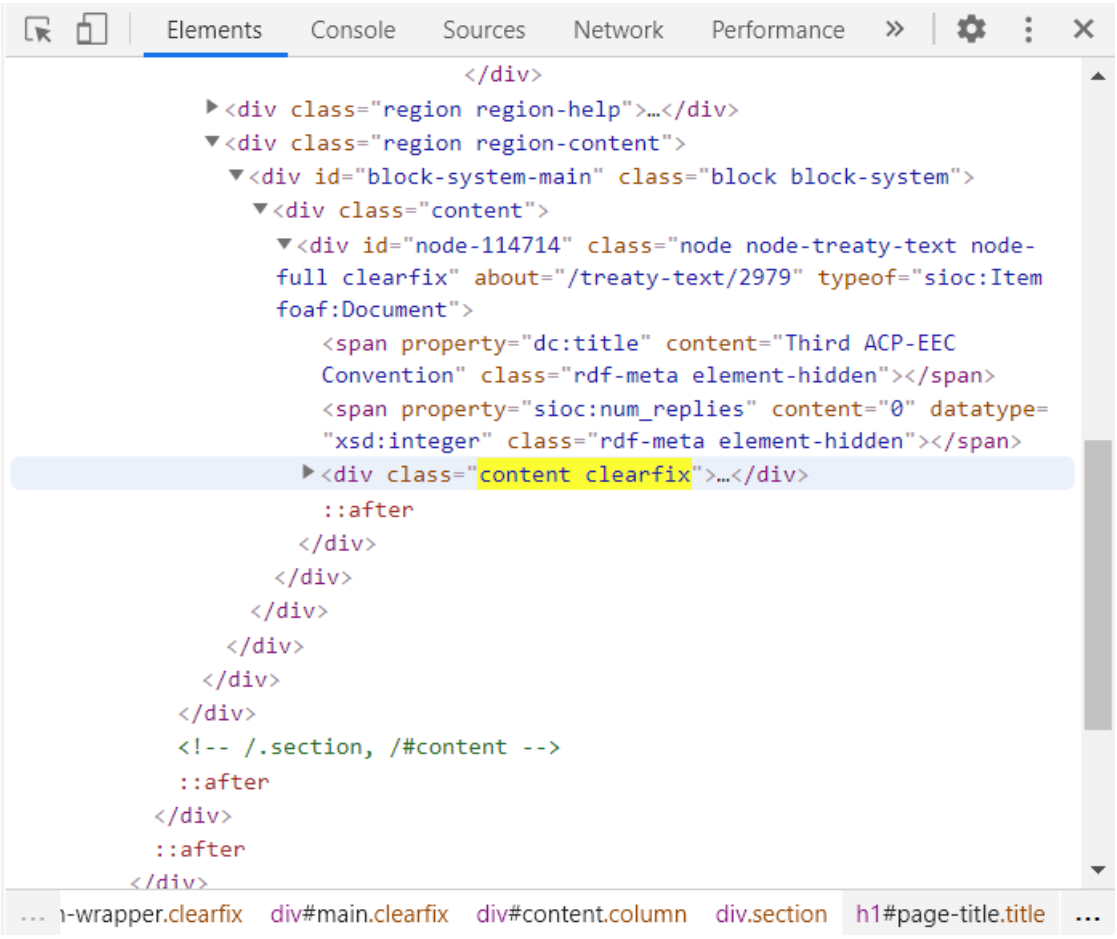
## 2.1 Inspecting the website

The second thing we need to do is to figure out where our treaties texts are stored in the IEADB website. The updated version of the database makes things easier for us. The URLs we are looking for are now all of the form https://iea.uoregon.edu/treaty-text/[treaty ID]. This will be helpful to code our Python program.

In addition, we need to tell our python program what we need from the page. More specifically, we need to figure out what are the HTML tags that contain the data we want to retrieve. Simply put, there is a lot of code on a website page and we want to find the relevant pieces of code that contains our data. You can learn more about HTML tags here.

To save time, I have already identified the tag (`<div>`) and the class (`"content clearfix"`) that contains our text. Importantly, the class of this tag is unique within the page and is

the same for all treaties. If you want to know more about how to identify relevant HTML tags, you can check out Julia Kho's tutorial on web scraping with Python here.




## 2.2 Python Code

Now that we have the CSV file with our IDs and we identified where to locate the treaties texts, the last thing we need before diving into the code is a folder in which we will store the treaties texts. For this tutorial, I have created a folder named "treaties" on my desktop.

All set? Now open your Python IDLE and let the fun begin!

We start by importing the relevant libraries.

```
import requests
from bs4 import BeautifulSoup
```

We use the requests library to tell our python program where to look for our data and the Beautiful Soup library for pulling information from web pages.

Next, we create a list that contains our treaties IDs. However, we cannot just copy-paste the first column from our CSV file into our python code because each element of a Python list

needs to be separated by a comma. Luckily, we can easily do this in excel with the function CONCAT. If you need some help with this step, check here. We can now copy-paste our concatenated IDs into our list of treaties.

```
list_treaty = [
3059,
8501,
3100,
5006]
```

Let's say we want to scrap the text of the first treaty in the list. We need to set the URL to the web page where the text is and access the page with our requests library.

```
url_string='https://iea.uoregon.edu/treaty-text/'+str(list_treaty[0])
website_url=requests.get(url_string)
```

As mentioned earlier, the treaties texts URL are all of the form: https://iea.uoregon.edu/treaty-text/[treaty Mitch ID]. Because we want the first treaty in the list, we tell python to replace [treaty Mitch ID] by `list_text_id[0]` (i.e. the first element of our list).

To check that the URL is correct, we can print the URL in the Python console.

```
print(url_string)
```

If the request is successful, you should see the following output.

```
>>> https://iea.uoregon.edu/treaty-text/3059
```

Next, we parse the HTML with BeautifulSoup so that we can work with a nicer, nested BeautifulSoup data structure.

```
soup = BeautifulSoup(website_url.text,'lxml')
```

We use the method `.find` to locate our text based on the HTML tag and class we identified earlier.

```
text_treaty = soup.find('div', {'class':'content clearfix'}).get_text()
```

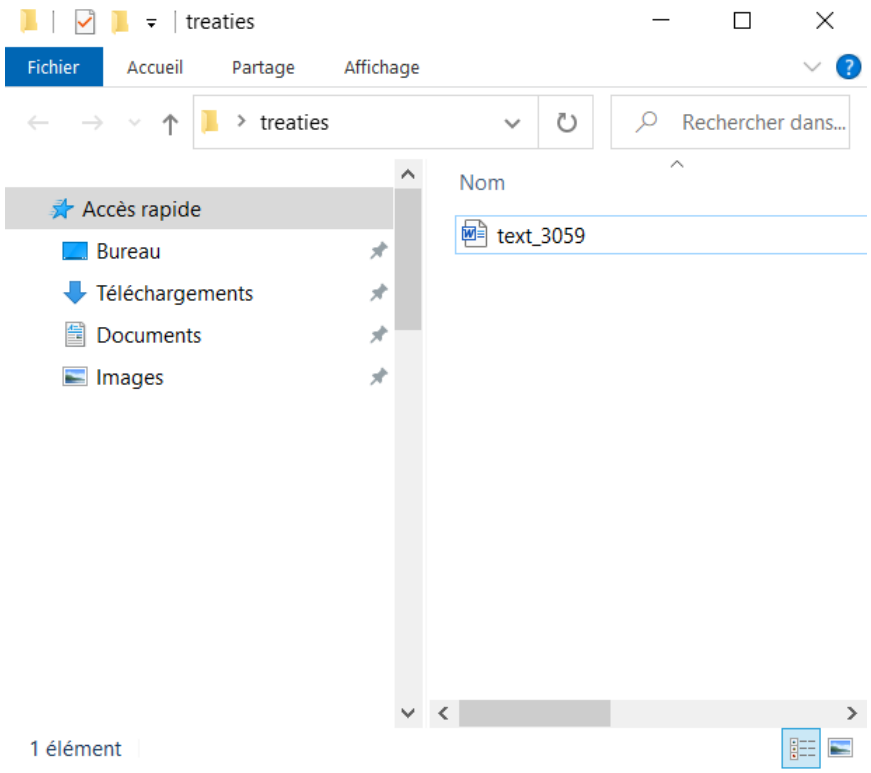
Finally, we create the word document in which we are going to save our treaty text. I name the file text_ + the treaty ID (which, as before, is given by the first element of our list of treaties) and I tell the program that I want the word document inside the folder "treaties" I have created at the beginning of this section. Of course, you need to replace `"C:/Users/alice/Desktop/treaties/"` by the path to your own folder. If you need help with this step, check here.

```
filename = "text_"+str(list_text_id[0])
location = "C:/Users/alice/Desktop/treaties/" + filename + ".doc"
```

We then tell the program to open the word document and write our treaty text in it.

```
file = open(location, "w", encoding="utf-8")
file.write(str(text_treaty))
file.close()
```

Go to your "treaties" folder. You now have a word document in there.




Open the word document. If word asks you to choose the encoding to use for this document choose "Unicode (UTF-8). You just webscraped your first treaty text. Well done!

Note that if a text you are looking for is not available on the IAE Database, your word document will contain the following:

```
Apologies but that page does not exist.
We migrated the IEADB website to a new Drupal platform in September 2016
and a few links may have gotten broken in the process.  You may be able
to find the old link by changing the URL from iea.uoregon.edu to iea-
archive.uoregon.edu
e.g., change
http://iea.uoregon.edu/page.php?file=home.htm&amp;query=static  to
http://iea-archive.uoregon.edu/page.php?file=home.htm&amp;query=static
We would greatly appreciate it if you cut and pasted the URL that is not
working and sent it to the PI at rmitchel@uoregon.edu so I can repair it.
Thanks,
Ronald Mitchell
```

Now that we understand how to download a treaty text, we can download the entire list of treaties texts using this code.

# 3 Alternative option

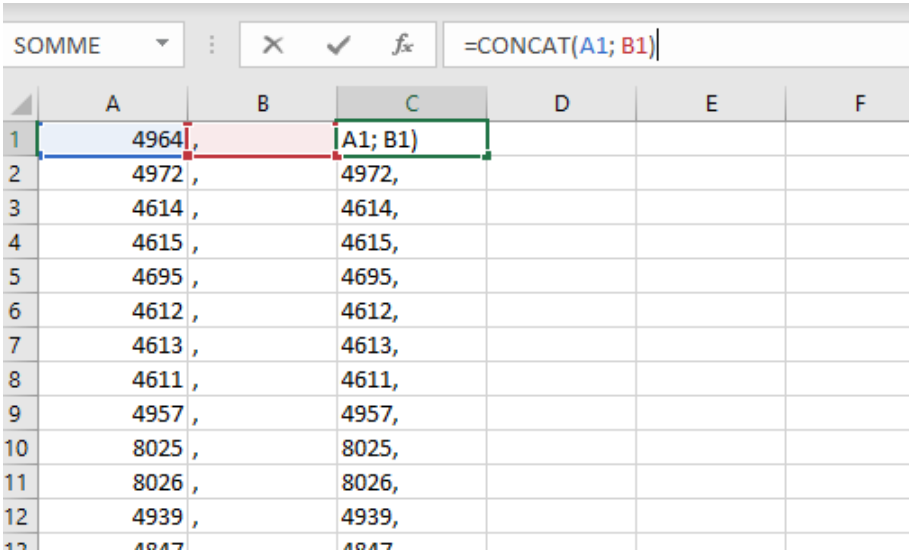
The code we have just seen is perfect if you plan on using automated text analyses techniques. If you need to read through the text manually instead, the lack of format can make the exercise tedious.

One quick way to go around this is to parse the whole HTML page that contain the treaties text you are looking for. Doing so will also save the style of the page (e.g. the font, the spacing, the colors, etc.). This will also save some irrelevant information (e.g. the different tabs) but you will get nicer word documents. The code for this is available here.

# 4 Extra Help

## 4.1 CONCAT in Excel

What we need is a column with the treaties IDs and one column filled with commas. Then we use the CONCAT function to add the comma to each ID. (Note that if you are using the ENGLISH version of excel, you need to use "," instead of ";" in the CONCAT function).




We can now copy-paste the content from the last column into our list of treaties in our python code!