Chemistry Programming with Python (Part 3) –Web Scraping Wikipedia For Chemical Identifiers

Andrew P. Cornell[1], Robert E. Belford[1]

[1]Department of Chemistry, University of Arkansas at Little Rock, 2801 S. University Avenue, Little Rock, AR 72204, USA

Abstract

Many individual chemicals have a specific page on Wikipedia that will give information about the use, manufacture and properties of that chemical. The properties that are displayed off to the side include the relevant chemical identifiers along with alternate names and reaction information. There are several different identifier formats displayed within the properties box that include InChI (International Chemical Identifier), SMILES (Simplified Molecular-Input Line-Entry System) and various registration numbers. This lesson will explain how Python can be used to web scrape Wikipedia and retrieve the InChI after the user inputs a chemical name. Web scraping is a process for extracting the contents of a web page. This is often useful for working with online sources that do not offer an API (Application Programming Interface) for certain types of data. Wikipedia does have API's for a lot of the information published, however this tutorial would like to look at the technique of web scraping with Python as an alternate method.

This program will work by importing a few helper modules that will allow the Python program to go onto the web, grab an HTML file and then parse the file specifically for the InChI string. Retrieving a valid result means that the user must input a chemical name that has a page designated on Wikipedia. Many chemicals have multiple names, so Wikipedia handles this through making the most commonly used name to be expressed in the URL (Uniform Resource Locator). All other naming formats will redirect to the URL that uses the chemicals common name.

Learning Objectives

- Import Python Libraries and Modules
- Create and Define Functions
- Parse HTML Text
- Display Results

Recommended Reading

- Internet of Chemistry Things Activity 1 (Page that explains basic Instructions for setting up Python on a computer. The Python Activities listed in the sidebar may also help to explain some of the background information.)
- Spring ChemInformatics OLCC Course (This site provides lots of information on working with chemical data.)
- Python Documentation (Python 3 documentation that correlates to the version used within this tutorial.)
- Beautiful Soup Module (Documentation on the installation and use of this module with Python.)

Methods

The Python File used in this tutorial can be located within the following GitHub Page along with a DOI (Digital Object Identifier) on FigShare.[1] Python will run on many different operating systems, however this tutorial uses the Thonny IDE (Integrated Development Environment) to design, run and test the code.[2] The following code will take a chemical name and insert this into a preformatted URL that will pull all of the html from a corresponding Wikipedia page. The code will then parse and separate out everything in the html from the InChI identifier displaying the results.

Python 3 has been used for all code in this tutorial so make sure to consult the correct version documentation if additional reference is needed. Should the syntax or format change with future updates to the Python Language, it may be necessary to approach the task in a different way. The steps are broken down into sections which should be placed into the file one after the other from top to bottom.

Step 1

Starting with the libraries and modules that need to be declared, enter the code in step 1. The first line will import a function called “urlopen” from a library module called “urllib.request”. This will be responsible for allowing the program to fetch URL’s. The second line will import a library called “BeautifulSoup” from the package “bs4”. This module will be responsible for isolating the html text that we would like to retrieve as a result. The last module that will be imported is called “re” and this will be used to make some regular expressions that look for the pattern defined in the programs code which will contain the results.

The Python documentation recommended may be helpful with getting a deeper understanding of how importation of libraries into the program works. Be sure that the following code in step 1 is placed at the top of the file.

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re
```

Step 2

After making the imports, add the following code which will define the first variable stored by Python. The name of the variable will be called “chemical” and it will store the value given through the text input displayed to the user. The variable stored should be a type of chemical identified by either its common name or systematic chemical name.

```
chemical = input("Put in the name of the chemical you want the InChI for: ")
```

Step 3

Two more variables will be necessary in setting up the preformatted URL structure needed to find corresponding chemical page on Wikipedia. The following code will set “html” as the variable and it will be assigned a full URL that is a combination of a preformatted section that does not change along with a piece that takes the user input defined in step 2. The URL will be pieced together into a single string matching where the chemical page is located on Wikipedia. In programming, this process is often called concatenation. The command “urlopen” will serve as the function or assignment to that page defining how to use the variable when called. After the “html” variable has been stored, a second variable called “wikiExtract” will store the text retrieved from this webpage. The first piece in the parenthesis will define what variable to call for the URL assignment followed by the format and the command for what should be done. The command “get_text” will then store everything on the page to the variable defined.

```
html = urlopen("https://en.wikipedia.org/wiki/" + chemical)
wikiExtract = BeautifulSoup(html, "lxml").get_text()
```

Step 4

After the html of the webpage has been retrieved, the next few lines will search, isolate and store the InChI value independently of the other text from the webpage. The first line will perform a search of the html for the pattern “InChI=.*” and put this into memory as a value. The star and dot in the pattern will tell the program to grab everything found after “InChI=” as a parameter. Once the value has been found, the second line of code will then break off all text that follows the InChI string and store only the string to a new variable. The last function will provide the most refinement in isolating the InChI string by removing any added space or unwanted characters. The variable “inchiFinal” will be sent to the users display as the result of the search.

```
inchiMatch = re.findall("InChI=.*", wikiExtract)
inchiClean = inchiMatch[0].split('H\\')
inchiFinal = inchiClean[0].split()
```

Step 5

Before the user receives the results, the following code can be inserted to give a nice little message that is followed by the actual InChI string. This will help to keep things looking nice and clean.

```
print("\n" + "Wikipedia says the InChI is:" + '\n')
print(inchiFinal[0])
```

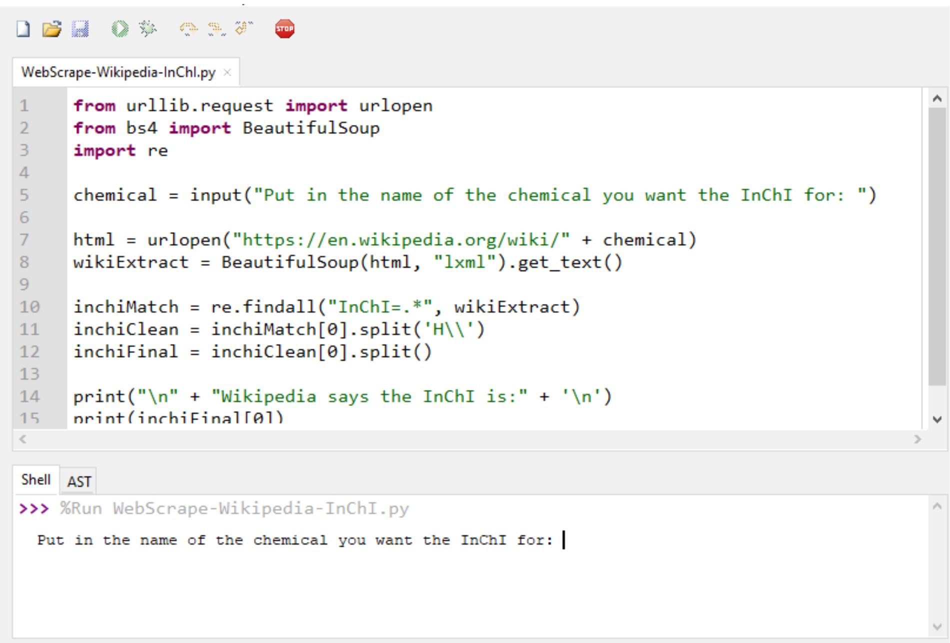
If you would like to just copy the entire program in sequence, below is the completed code containing everything that is needed to perform retrieving an InChI from Wikipedia.

Completed Code Example

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re

chemical = input("Put in the name of the chemical you want the InChI for: ")

html = urlopen("https://en.wikipedia.org/wiki/" + chemical)
wikiExtract = BeautifulSoup(html, "lxml").get_text()

inchiMatch = re.findall("InChI=.*", wikiExtract)
inchiClean = inchiMatch[0].split('H\\')
inchiFinal = inchiClean[0].split()

print("\n" + "Wikipedia says the InChI is:" + '\n')
print(inchiFinal[0])
```

Program Demonstration

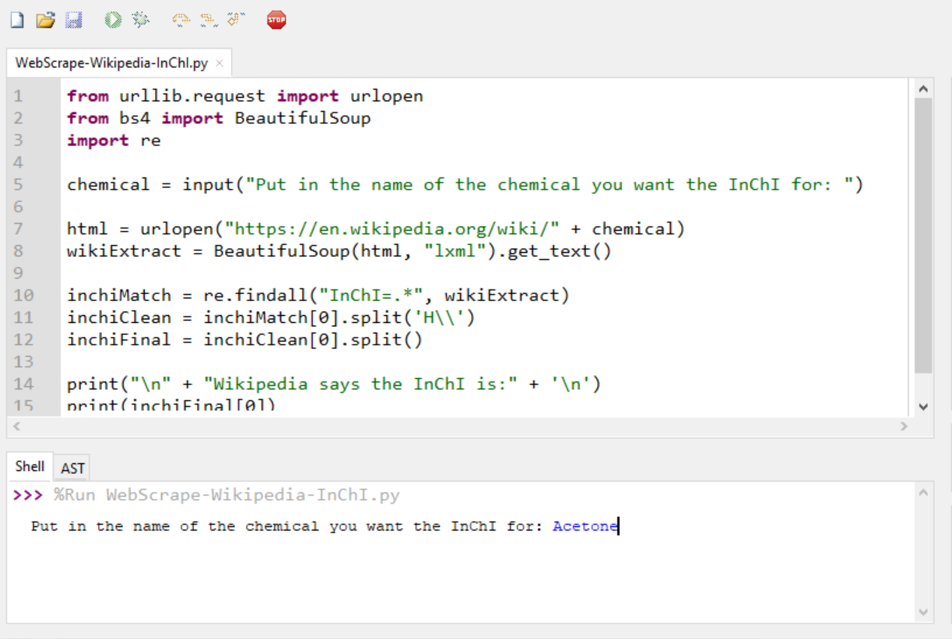
An interactive demo of this program is provided by Trinket in the online publication.[5] The trinket can be visited at this location. The stored and printable copies will only contain screenshots below with descriptions as they cannot display the trinket program in a live environment.



```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re

chemical = input("Put in the name of the chemical you want the InChI for: ")

html = urlopen("https://en.wikipedia.org/wiki/" + chemical)
wikiExtract = BeautifulSoup(html, "lxml").get_text()

inchiMatch = re.findall("InChI=.*", wikiExtract)
inchiClean = inchiMatch[0].split('H\\')
inchiFinal = inchiClean[0].split()

print("\n" + "Wikipedia says the InChI is:" + '\n')
print(inchiFinal[0])
```

```
>>> %Run WebScrape-Wikipedia-InChI.py

  Put in the name of the chemical you want the InChI for: 
```

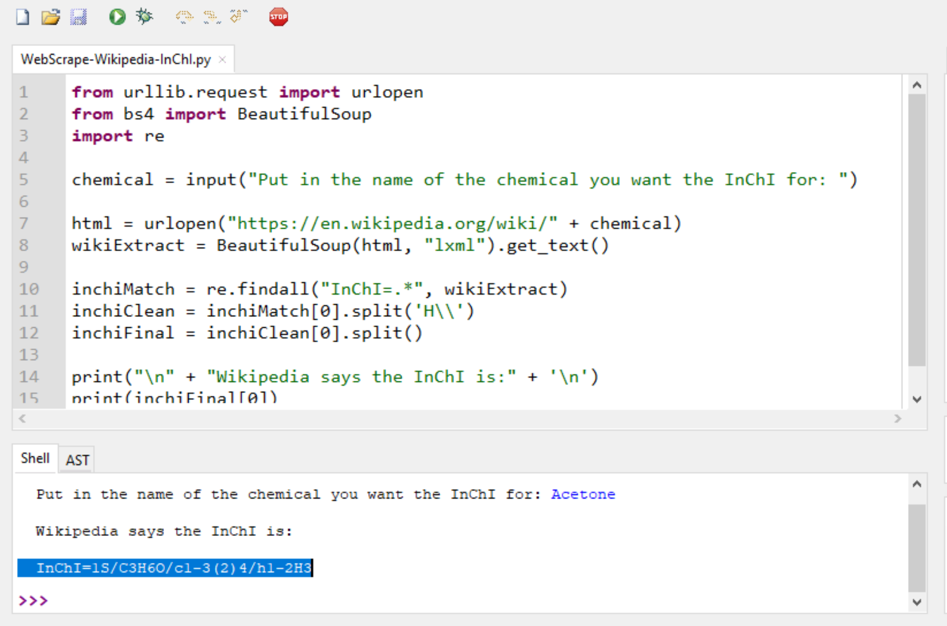
Figure 1 The above image shows the interpreter requesting a chemical name from the user



```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re

chemical = input("Put in the name of the chemical you want the InChI for: ")

html = urlopen("https://en.wikipedia.org/wiki/" + chemical)
wikiExtract = BeautifulSoup(html, "lxml").get_text()

inchiMatch = re.findall("InChI=.*", wikiExtract)
inchiClean = inchiMatch[0].split('H\\')
inchiFinal = inchiClean[0].split()

print("\n" + "Wikipedia says the InChI is:" + '\n')
print(inchiFinal[0])
```

```
>>> %Run WebScrape-Wikipedia-InChI.py

 Put in the name of the chemical you want the InChI for: Acetone
```

Figure 2 The above image shows the user inserting the chemical acetone into the request.



```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re

chemical = input("Put in the name of the chemical you want the InChI for: ")

html = urlopen("https://en.wikipedia.org/wiki/" + chemical)
wikiExtract = BeautifulSoup(html, "lxml").get_text()

inchiMatch = re.findall("InChI=.*", wikiExtract)
inchiClean = inchiMatch[0].split('H\\')
inchiFinal = inchiClean[0].split()

print("\n" + "Wikipedia says the InChI is:" + '\n')
print(inchiFinal[0])
```

```
 Put in the name of the chemical you want the InChI for: Acetone

 Wikipedia says the InChI is:

 InChI=1S/C3H6O/c1-3(2)4/h1-2H3

>>>
```

Figure 3 The above image shows the Python program displaying the resulting InChI after web scraping wikipedia for the answer.

References

(1) Cornell, A. Cheminformatics-Python. Figshare 2018. https://doi.org/10.6084/m9.figshare.7255901.

(2) Annamaa, A. Introducing Thonny, a Python IDE for Learning Programming. In *Proceedings of the 15th Koli Calling Conference on Computing Education Research - Koli Calling ’15*; ACM Press: Koli, Finland, 2015; pp 117–121. https://doi.org/10.1145/2828959.2828969.