Lab 1: Challenge 2

1. Extract all the text entries containing information for the following properties: awardee, field, year, and work.

1. Save the list as a csv file named **“nobel_laureates.csv”** - use the csv library for this

In [1]:

```python
# Libraires to obtain data from servers (by requests) and BeautifulSoup to extract the data and find specific entries
import requests
from bs4 import BeautifulSoup
```

In [2]:

```python
# Pandas to sort and store data to then be written to a csv file
import pandas as pd
```

Generating a dataframe to store the text obtained by the scrapper

Which can then be stored as a csv file

In [37]:

```python
del df
```

In [38]:

```python
# The specific entries we are looking for
column_names = ['awardee', 'field', 'year', 'work']

# The dataframe, which would be like an excel spreadsheet to organize the text obtained
df = pd.DataFrame(columns = column_names)
```

Inspecting the Website

The nobels are subdivided by year, from 1901-2021

In [4]:

```python
years = 2021-1901
print('Years to search: ', years)
```

```
Years to search:  120
```

Since the website has all nobel prices listed in a single page, we can simply use the main URL as the one to be used to extract the data.

In [5]:

```python
url = "https://www.nobelprize.org/prizes/lists/all-nobel-prizes/"
print('Main URL: ', url)
```

```
Main URL:  https://www.nobelprize.org/prizes/lists/all-nobel-prizes/
```

Obtain data from main URL

```
In [6]: print('Main URL: ', url)
```

```
Main URL:  https://www.nobelprize.org/prizes/lists/all-nobel-prizes/
```

```
In [10]: r = requests.get(url)
         c = r.content
```

```
In [15]: r
```

```
Out[15]: <Response [403]>
```

```
In [12]: soup = BeautifulSoup(c,"html.parser")
         print(soup)
```

```
<html>
<head><title>403 Forbidden</title></head>
<body>
<center><h1>403 Forbidden</h1></center>
<hr/><center>nginx</center>
</body>
</html>
```

As we can see above the requests to obtain the data results in an error, it could be due to using datahub.ucsd.

In order to fix this we can add the header to specify the user and the chrome version I am using to open the url.

```
In [66]: headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTM
         L, like Gecko) Chrome/97.0.4692.71 Safari/537.36'}
         r = requests.get(url, headers=headers)
         c = r.content
         soup = BeautifulSoup(c,"html.parser")
         # print(soup.prettify())
```

Nobel prizes by year structure

All the years are within:

tag: **div class="by_year"**

- **YEAR** ⟶ *tag:* **h2** headers are the years
  - "The Nobel Prize in" + **field** + **year** ⟶ *tag:* **h3** headers
  - We can extract fields as any

Obtaining fields

- We can extract the field by obtaining the text from tag: **h3**
- The header3 tag is followed by a clickable link ⟶ *tag:* **a** link to fiel summary

"https://www.nobelprize.org/prizes/ (https://www.nobelprize.org/prizes/)" **field** "/" **year** "/summary/"

We can use these urls as a base to obtain the authors.

- Note that the field in the url might not be the complete field name, this must be obtained from, the text used as a bridge to the link.
- Thus we store both: *field* and *url_field*, using the url with the year to all authors and work

Retreiving data

In [9]:

```
# The years are mixed, some years are underneath the first div by_year but later years a
re under 2 layers of div by_year
yrs_fields_1 = soup.select("div > h3 > a")
print('Sub division 1 length : ', len(yrs_fields_1))

yrs_fields_2 = soup.select("div > div > h3 > a")
print('Sub division 2 length : ', len(yrs_fields_2))
```

```
Sub division 1 length :  658
Sub division 2 length :  419
```

In [10]:

```
# Checking for a possible 3rd subdivision
soup.select("div > div > div > h3 > a")
```

Out[10]:

```
[]
```

In [11]:

```
# Contains a tags to the subtitle (fields) for all years. Sample output
yrs_fields_1[:5]
```

Out[11]:

```
[<a href="https://www.nobelprize.org/prizes/physics/2021/summary">The Nobel Prize in Ph
ysics 2021</a>,
 <a href="https://www.nobelprize.org/prizes/chemistry/2021/summary/">The Nobel Prize in
Chemistry 2021</a>,
 <a href="https://www.nobelprize.org/prizes/medicine/2021/summary">The Nobel Prize in P
hysiology or Medicine 2021</a>,
 <a href="https://www.nobelprize.org/prizes/literature/2021/summary/">The Nobel Prize i
n Literature 2021</a>,
 <a href="https://www.nobelprize.org/prizes/peace/2021/summary/">The Nobel Peace Prize
2021</a>]
```

In [39]:

```
df
```

Out[39]:

awardee	field	year	work

In [40]:

```python
def store_retrieved_data(df, div_fields):

    for i in range(len(div_fields)):
        # Get the current field and year
        sub_title = div_fields[i]

        # Get the year as an int
        yr = int(sub_title.string[-4:])

        # Determine the url pages that correspond to the current field to determine the name
        field_url_i = str(sub_title)
        url_field = str(field_url_i.split('>')[0].split('prizes/')[1].split('/')[0])

        if url_field == 'economics' or url_field == 'economic-sciences':
            field_i = 'Economic Sciences'

        elif url_field == 'peace':
            field_i = 'Peace'

        else:
            field_i = str(sub_title.string[:-5])
            field_i = str(field_i.split('Prize in ')[-1])

        # Get the url to the current field and year up to the year
        tag_pi = field_url_i.split(str(yr))
        tag_pi = str(tag_pi[0]) + str(yr)

        # Get the paragraph following the header with the field and year
        next_p = sub_title.find_all_next("p")

        for pi in next_p:

            # find_all_next returns all iterations following a specific element by an specified tag,
            # Note: this will output all next even if it does not belong to that p. We need to filter by the main year+field url
            a_tags = pi.find_all("a")

            # Check if there is an "a href" tag, if there is then that would be the link to the authors
            if len(a_tags) == 0:
                continue

            ai = str(a_tags[0])
            tag_ai = str(ai.split(str(yr))[0]) + str(yr)

            # Checking that this a tag matches that of the current year and field (obtained from p):
            if tag_pi != tag_ai:
                break

            txt = str(pi.get_text())
            # Some awardees in the html file are separated not by a " " but an actual characted that reads as \xa0 or \n
            strip_txt = txt.split("\n")
            awardee_i = ''

            if len(strip_txt) == 1:
                txt_split = txt.split('“')
                names = str(txt_split[0])
                names = names.split('\xa0')

                if len(names) != 1:
                    for name in names:
                        awardee_i += name + ' '
                else:
                    awardee_i = str(txt_split[0])
                work_i = str(txt_split[1][:-1])

            else:
                names = str(strip_txt[0])
                names = names.split('\xa0')

                for name in names:
                    awardee_i += name + ' '
                work_i = str(strip_txt[1][1:-1])

            df.loc[len(df.index)] = [awardee_i, field_i, yr, work_i]

    return df
```

In [41]:

```
df = store_retrieved_data(df, yrs_fields_1)
df = store_retrieved_data(df, yrs_fields_2)
```

In [42]:

```
df
```

Out[42]:

	awardee	field	year	work
0	Syukuro Manabe and Klaus Hasselmann	Physics	2021	for the physical modelling of Earth’s climate,...
1	Giorgio Parisi	Physics	2021	for the discovery of the interplay of disorder...
2	Benjamin List and David MacMillan	Chemistry	2021	for the development of asymmetric organocatalysis
3	David Julius and Ardem Patapoutian	Physiology or Medicine	2021	for their discoveries of receptors for tempera...
4	Abdulrazak Gurnah	Literature	2021	for his uncompromising and compassionate penet...
...	...	...	...	...
1060	Jacobus Henricus van ‘t Hoff	Chemistry	1901	in recognition of the extraordinary services h...
1061	Emil Adolf von Behring	Physiology or Medicine	1901	for his work on serum therapy, especially its ...
1062	Sully Prudhomme	Literature	1901	in special recognition of his poetic compositi...
1063	Jean Henry Dunant	Peace	1901	for his humanitarian efforts to help wounded s...
1064	Frédéric Passy	Peace	1901	for his lifelong work for international peace ...

1065 rows × 4 columns

Barack Obama Nobel price

In [82]:
```
df.loc[df.awardee == "Barack H. Obama "]
```

Out[82]:

	awardee	field	year	work
84	Barack H. Obama	Peace	2009	for his extraordinary efforts to strengthen in...

Ernest Rutherford

In [103]:
```
df = df.drop(index=1021)
```

In [104]:
```
df.loc[df.awardee == "Ernest Rutherford "]
```

Out[104]:

	awardee	field	year	work
612	Ernest Rutherford	Chemistry	1908	for his investigations into the disintegration...

In [107]:
```
df.year[df.awardee == "Ernest Rutherford "]
```

Out[107]:
```
612    1908
Name: year, dtype: object
```

In [105]:
```
df.field[df.awardee == "Ernest Rutherford "]
```

Out[105]:
```
612    Chemistry
Name: field, dtype: object
```

In [108]:
```
df.work[df.awardee == "Ernest Rutherford "].to_list()
```

Out[108]:
```
['for his investigations into the disintegration of the elements, and the chemistry of
radioactive substances']
```

Physics

In [112]:
```
df = df.drop(index=472)
```

In [114]:
```
df.awardee[df.year == 1939][df.field == "Physics"]
```

Out[114]:
```
881    Ernest Orlando Lawrence
Name: awardee, dtype: object
```

Writting to csv

In [65]:

```
import csv

# open the file in the write mode
header=['awardee', 'field', 'year', 'work']

with open('nobel_laureates.csv', 'w') as f:
    # create the csv writer
    writer = csv.writer(f, delimiter=',',
                     quoting=csv.QUOTE_ALL)

    # write a row to the csv file
    for n in range(len(df.index)):
        awardee = df.iloc[n]['awardee']
        field = df.iloc[n]['field']
        year = df.iloc[n]['year']
        work = df.iloc[n]['work']

        if work[0] == '"':
            work = work[1:]

        elif work[-1] == "'":
            work = work[:-1]

        row = [awardee, field, year, work]

        writer.writerow(row)

# close the file
f.close()
```