Homework 2

October 13, 2016

In this notebook we will guide you through our solution for the second homework of ADA.

1 Part 1: Extracting the options from the dropdown menu

We start with the URL from the homepage where you can the select the different years/semesters etc..

```
In [1]: import requests
        from bs4 import BeautifulSoup

        urlHome = 'http://isa.epfl.ch/imoniteur_ISAP/\
        %21gedpublicreports.htm?ww_i_reportmodel=133685247'
```

We then use requests and BeautifulSoup to import the HTML data from the homepage.

```
In [3]: home = requests.get(urlHome)
        homeSoup = BeautifulSoup(home.content,'lxml')
        homeSoup
```

```
Out[3]: <html>
        <head>
        <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
        <title>IS-Academia</title>
        <link href="gedpublicreports.css?ww_x_path=Gestac.Moniteur.Style" rel="styl
        <link href="!ewxslsource.css?ww_x_path=Eqnx.Ajax.PageNL_css" rel="styleshee
        </head>
        <frameset border="0" frameborder="0" rows="113,*">
        <frame name="entete" noresize="" scrolling="no" src="!GEDPUBLICREPORTS.ente
        <frameset border="0" cols="25%,*" frameborder="0">
        <frame name="toc" src="!GEDPUBLICREPORTS.filter?ww_i_reportModel=133685247"
        <frame name="principal" src="GEDPUBLICREPORTS.empty"/>
        </frameset>
        </frameset>
        </html>
        <!-- OpenXml:0.01s  agent ctrl:0.00s  xml:0.00s  xsl clob before parse:0.00
```

The insepction tool of our browser helped learn that we were to access a #document that is in the fram of one of the framesets of the HTML page, the document is sourced from another page

of IS-Academia that we have to access. To access it, we scroll through the framesets and create a list of all the frames.

```
In [107]: iframe = []
          for frame in homeSoup.frameset.find_all('frame'):
              iframe.append(frame)
          print (iframe)
          urlOptions = iframe[1]['src']
          #urlOptions (homeSoup.frameset.frameset.frame['src']) this can be more op
          print(urlOptions)
```

```
[<frame name="entete" noresize="" scrolling="no" src="!GEDPUBLICREPORTS.entete?ww_i
!GEDPUBLICREPORTS.filter?ww_i_reportModel=133685247
```

The url of the source page is found, we can use it to get the other HTML page

```
In [672]: iframe = []
          for frame in homeSoup.frameset.find_all('frame'):
              iframe.append(frame)
          #print (iframe)
          urlOptions = iframe[1]['src']
          #homeSoup.frameset.frameset.frame['src'] this can be more optimal.....
          #print(urlOptions)
          urlOptions = 'http://isa.epfl.ch/imoniteur_ISAP/' + urlOptions
          options = requests.get(urlOptions)
          soupOptions = BeautifulSoup(options.content,'lxml')

          ##########################################################################
          #soupOptions.body.find_all('div')
          #By inspection we have two divs. Its the second one that has the filters
          #divOptions=soupOptions.find_all('div', {'class':'filtres'})# the div wit
          #from the options ablove let us get the cotents
          # Especially it contains  3 tables we extract those and filter out the pa


          ##########################################################################
          #### Part 1 name extraction

          unitStorage=[]

          model1 = homeSoup.frameset.frameset.frame['src']
          formater = model1[25:41]
          unitStorage.append(formater) ### appending formater

          # return the model formatter   and append to unit

          soupOptions.body.find_all('table')
          tab0= soupOptions.find(lambda tag: tag.name=='table' and  tag['id']=="for
```

```
        formatSpec=  tab0.input['name'] # this brings out the name . We also need
        unitStorage.append(formatSpec)

        ####for the other option selectors
        for l in  tableaux.find_all('select'):
                unitStorage.append(l['name'])


        print("the unit list contains names from option tag as below:")
        print(unitStorage)
        # the unit generated here will be used for the mappings later
```

```
the unit list contains names from option tag as below:
['ww_i_reportModel', 'ww_i_reportModelXsl', 'ww_x_UNITE_ACAD', 'ww_x_PERIODE_ACAD',
```

2 Getting Parameter values

```
In [673]: ### PArt 2 : we want to get parameter values in the  options. we make a c
          # this uses table 1
          # values for formatters
          print("We first generate the values for model and format generated above:
          value =   model1[42:53];
          print(value)
          formatVal = tab0.input['value'];
          print(formatVal)
          tableaux= soupOptions.find(lambda tag: tag.name=='table' and  tag['id']==
          array1 = [];# array 1 to be zipped with array 2
          array2 = [];


          for el in tableaux.find_all('option'):
              array1.append(el.string)
              array2.append(el['value'])

          lexicon ={};
          lexicon = dict(zip(array1,array2));
          print('\n')
          print("lexicon is our dictionary of parameters and its contains key/value
          print(lexicon)

          ##########################################################################
          print('\n')
          ### Part 3:  from lexicon we extract the values we observed in postman ar
          # we create a new list for each of the parameters we care about.

          parameter_1 = (lexicon['Informatique']) # put the value for informatique
```

```
print(parameter_1)

# for the semesters we have
parameter_2=[]

parameter_2.append(lexicon["Semestre d'automne"])
parameter_2.append(lexicon["Semestre printemps"])

print(parameter_2) # values for the two semesters available


#generate third parameter for years and their values
parameter_4={}; #this is dictionary

for k,v in lexicon.items():
    if k is not None: #checking for none because it exist in the keys. Wi
        if k.startswith('2'):
            parameter_4[k]=v
print(parameter_4)
########Ordering the dictionary above###################

import collections
OrderedDict = { '2007-2008': '978181','2008-2009': '978187','2009-2010':
order= collections.OrderedDict(sorted(parameter_4.items()))

###############end of ordering#######################
```

```
We first generate the values for model and format generated above:
133685247
133685270


lexicon is our dictionary of parameters and its contains key/value pairs below:
{'Stage printemps master': '2227132', 'Mise à niveau': '2063602308', 'Architecture'


249847
['2936286', '2754553']
{'2011-2012': '123455150', '2007-2008': '978181', '2014-2015': '213637922', '2012-2
```

3 Parameter Generation continues

```
In [674]: ###creating list for the semesters#######
          parameter_4_list=[];
          for i,v in order.items():
              parameter_4_list.append(v)
```

```
print(parameter_4_list )

######the final part for the periode pedagogique

test = ["Ba", "Ma","Min","Mis","Pro","Sem","Sta"]
parameter_5 = {};
for k,v in lexicon.items():
    if k is not None: #checking for none because it exist in the keys. Wi
        if k.startswith(tuple(test)):
            parameter_5[k]=v
parameter_5
```

```
['978181', '978187', '978195', '39486325', '123455150', '123456101', '213637754', '
```

```
Out[674]: {'Bachelor semestre 1': '249108',
 'Bachelor semestre 2': '249114',
 'Bachelor semestre 3': '942155',
 'Bachelor semestre 4': '942163',
 'Bachelor semestre 5': '942120',
 'Bachelor semestre 5b': '2226768',
 'Bachelor semestre 6': '942175',
 'Bachelor semestre 6b': '2226785',
 'Management de la technologie': '946882',
 'Master semestre 1': '2230106',
 'Master semestre 2': '942192',
 'Master semestre 3': '2230128',
 'Master semestre 4': '2230140',
 'Mathématiques': '944590',
 'Mineur semestre 1': '2335667',
 'Mineur semestre 2': '2335676',
 'Mise à niveau': '2063602308',
 'Projet Master automne': '249127',
 'Projet Master printemps': '3781783',
 'Semestre automne': '953159',
 "Semestre d'automne": '2936286',
 'Semestre de printemps': '2936295',
 'Semestre printemps': '2754553',
 'Stage automne 3ème année': '953137',
 'Stage automne 4ème année': '2226616',
 'Stage printemps 3ème année': '983606',
 'Stage printemps 4ème année': '2226626',
 'Stage printemps master': '2227132'}
```

4 Parameter aggregation

```
In [675]:  #####bringing all together

           print(unitStorage)
           print('\n')
           parameter = [value,formatVal,parameter_1,parameter_2,parameter_4_list,par
           print (parameter)

           # lets parametr 5 be list and return only values
           semesterList=[]

           for i,v in parameter_5.items():
               semesterList.append(v)
           print(semesterList)

           parameter = [value,formatVal,parameter_1,parameter_4_list,semesterList,pa
           print('\n')
           parameter
```

```
['ww_i_reportModel', 'ww_i_reportModelXsl', 'ww_x_UNITE_ACAD', 'ww_x_PERIODE_ACAD',


['133685247', '133685270', '249847', ['2936286', '2754553'], ['978181', '978187', '
['2227132', '2063602308', '2230128', '2226626', '3781783', '249114', '2230140', '29
```

```
Out[675]: ['133685247',
           '133685270',
           '249847',
           ['978181',
            '978187',
            '978195',
            '39486325',
            '123455150',
            '123456101',
            '213637754',
            '213637922',
            '213638028',
            '355925344'],
           ['2227132',
            '2063602308',
            '2230128',
            '2226626',
            '3781783',
            '249114',
```

```
            '2230140',
            '2936295',
            '2226768',
            '2754553',
            '953137',
            '942192',
            '2335667',
            '942175',
            '983606',
            '2335676',
            '953159',
            '249108',
            '942155',
            '2226616',
            '944590',
            '946882',
            '2230106',
            '2226785',
            '2936286',
            '942120',
            '942163',
            '249127'],
           ['2936286', '2754553']]
```

5 Create mappings

```
In [683]: # I have changed names so that it becomes obvious
          from collections import OrderedDict
          keys = unitStorage
          Values = parameter
          formParameter= dict(zip(keys, Values))
          print(formParameter)
          # force dictionary to be ordered
          OrderedDict = {'ww_i_reportModel', 'ww_i_reportModelXsl', 'ww_x_UNITE_ACA

          formParameters =  collections.OrderedDict(sorted(formParameter.items()))
          formParameters

{'ww_x_HIVERETE': ['2936286', '2754553'], 'ww_x_PERIODE_ACAD': ['978181', '978187',


Out[683]: OrderedDict([('ww_i_reportModel', '133685247'),
                       ('ww_i_reportModelXsl', '133685270'),
                       ('ww_x_HIVERETE', ['2936286', '2754553']),
                       ('ww_x_PERIODE_ACAD',
                        ['978181',
                         '978187',
                         '978195',
```

```
                          '39486325',
                          '123455150',
                          '123456101',
                          '213637754',
                          '213637922',
                          '213638028',
                          '355925344']),
                         ('ww_x_PERIODE_PEDAGO',
                          ['2227132',
                          '2063602308',
                          '2230128',
                          '2226626',
                          '3781783',
                          '249114',
                          '2230140',
                          '2936295',
                          '2226768',
                          '2754553',
                          '953137',
                          '942192',
                          '2335667',
                          '942175',
                          '983606',
                          '2335676',
                          '953159',
                          '249108',
                          '942155',
                          '2226616',
                          '944590',
                          '946882',
                          '2230106',
                          '2226785',
                          '2936286',
                          '942120',
                          '942163',
                          '249127']),
                         ('ww_x_UNITE_ACAD', '249847')])
```

In [ ]:
```
####################TO_DO

## ordered colection syntex to be corrected on the dictionary for parameter

### We have to include GPS

### ok
```

```
        ###################### from here downwards this is jsut for crosschecking

        periode_acad = [978181,978187,978195,39486325,123455150,123456101,213637754
            # 2007-2008 to 2016-2017 with S1 and s2
            GPS = [(71297531,19561998),
                   (109576936,39494897),
                   (213617925,71297626),
                   (357704486,109577031),
                   (736308968,213618020),
                   (1378362092,357704613),
                   (1650771864,736309090),
                   (1744377893,1378362238),
                   (1897032870,1650772010),
                   (2021043255,1744378039)]
            periode_pedago = [249108,942175] # S1-S2
            hiverete = [2936286,2936295]

            dic_param = {'ww_x_GPS':GPS[period][semester],
                         'ww_i_reportModel':133685247,
                         'ww_i_reportModelXsl':133685270,
                         'ww_x_UNITE_ACAD':249847,
                         'ww_x_PERIODE_ACAD':periode_acad[period],
                         'ww_x_PERIODE_PEDAGO':periode_pedago[semester],# Value (S1 o
                         'ww_x_HIVERETE':hiverete[semester]}# Value (Automne or print

In [ ]:

In [662]: tableaux.find_all('option') # this is just for the group. It is to be del

Out[662]: [<option value="null"></option>,
           <option value="942293">Architecture</option>,
           <option value="246696">Chimie et génie chimique</option>,
           <option value="943282">Cours de mathématiques spéciales</option>,
           <option value="637841336">EME (EPFL Middle East)</option>,
           <option value="942623">Génie civil</option>,
           <option value="944263">Génie mécanique</option>,
           <option value="943936">Génie électrique et électronique </option>,
           <option value="2054839157">Humanités digitales</option>,
           <option value="249847">Informatique</option>,
           <option value="120623110">Ingénierie financière</option>,
           <option value="946882">Management de la technologie</option>,
           <option value="944590">Mathématiques</option>,
           <option value="945244">Microtechnique</option>,
           <option value="945571">Physique</option>,
           <option value="944917">Science et génie des matériaux</option>,
           <option value="942953">Sciences et ingénierie de l'environnement</option
```

```
            <option value="945901">Sciences et technologies du vivant</option>,
            <option value="1574548993">Section FCUE</option>,
            <option value="946228">Systèmes de communication</option>,
            <option value="null"></option>,
            <option value="355925344">2016-2017</option>,
            <option value="213638028">2015-2016</option>,
            <option value="213637922">2014-2015</option>,
            <option value="213637754">2013-2014</option>,
            <option value="123456101">2012-2013</option>,
            <option value="123455150">2011-2012</option>,
            <option value="39486325">2010-2011</option>,
            <option value="978195">2009-2010</option>,
            <option value="978187">2008-2009</option>,
            <option value="978181">2007-2008</option>,
            <option value="null"></option>,
            <option value="249108">Bachelor semestre 1</option>,
            <option value="249114">Bachelor semestre 2</option>,
            <option value="942155">Bachelor semestre 3</option>,
            <option value="942163">Bachelor semestre 4</option>,
            <option value="942120">Bachelor semestre 5</option>,
            <option value="2226768">Bachelor semestre 5b</option>,
            <option value="942175">Bachelor semestre 6</option>,
            <option value="2226785">Bachelor semestre 6b</option>,
            <option value="2230106">Master semestre 1</option>,
            <option value="942192">Master semestre 2</option>,
            <option value="2230128">Master semestre 3</option>,
            <option value="2230140">Master semestre 4</option>,
            <option value="2335667">Mineur semestre 1</option>,
            <option value="2335676">Mineur semestre 2</option>,
            <option value="2063602308">Mise à niveau</option>,
            <option value="249127">Projet Master automne</option>,
            <option value="3781783">Projet Master printemps</option>,
            <option value="953159">Semestre automne</option>,
            <option value="2754553">Semestre printemps</option>,
            <option value="953137">Stage automne 3ème année</option>,
            <option value="2226616">Stage automne 4ème année</option>,
            <option value="983606">Stage printemps 3ème année</option>,
            <option value="2226626">Stage printemps 4ème année</option>,
            <option value="2227132">Stage printemps master</option>,
            <option value="null"></option>,
            <option value="2936286">Semestre d'automne</option>,
            <option value="2936295">Semestre de printemps</option>]

In [663]:

Out[663]: {'Stage printemps master': '2227132',
           'Mise à niveau': '2063602308',
           'Architecture': '942293',
```

```
'Semestre de printemps': '2936295',
'Stage printemps 4ème année': '2226626',
'2016-2017': '355925344',
'Science et génie des matériaux': '944917',
'EME (EPFL Middle East)': '637841336',
None: 'null',
'Informatique': '249847',
'Microtechnique': '945244',
'Management de la technologie': '946882',
'2012-2013': '123456101',
'Stage printemps 3ème année': '983606',
'Mineur semestre 2': '2335676',
'Semestre automne': '953159',
'Mathématiques': '944590',
'Stage automne 4ème année': '2226616',
'Systèmes de communication': '946228',
'Ingénierie financière': '120623110',
'Chimie et génie chimique': '246696',
'2014-2015': '213637922',
'Bachelor semestre 4': '942163',
'2010-2011': '39486325',
'Projet Master automne': '249127',
'2009-2010': '978195',
'Projet Master printemps': '3781783',
'Master semestre 4': '2230140',
'Master semestre 3': '2230128',
'Bachelor semestre 2': '249114',
'Sciences et technologies du vivant': '945901',
'2007-2008': '978181',
'Génie mécanique': '944263',
'Génie électrique et électronique ': '943936',
'Humanités digitales': '2054839157',
'Semestre printemps': '2754553',
'2015-2016': '213638028',
'Stage automne 3ème année': '953137',
'2008-2009': '978187',
'Génie civil': '942623',
'Master semestre 2': '942192',
'Bachelor semestre 6': '942175',
'Section FCUE': '1574548993',
'Bachelor semestre 1': '249108',
'Bachelor semestre 5b': '2226768',
'Bachelor semestre 3': '942155',
"Sciences et ingénierie de l'environnement": '942953',
'Physique': '945571',
'Master semestre 1': '2230106',
'2013-2014': '213637754',
'Bachelor semestre 6b': '2226785',
```

```
            "Semestre d'automne": '2936286',
            'Bachelor semestre 5': '942120',
            'Mineur semestre 1': '2335667',
            '2011-2012': '123455150',
            'Cours de mathématiques spéciales': '943282'}
```

In [630]:
```
model1[42:53]
```

Out[630]:
```
'133685247'
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: