td2A_eco_API_pocket_et_Webscraping_correction

May 15, 2019

1 2A.eco - Web Scrapping et API avec Pocket - correction

Le notebook revient sur le webscrapping et l'utilisation d'API avec pocket.

[1]:
```
from jyquickhelper import add_notebook_menu
add_notebook_menu()
```

[1]:
```
<IPython.core.display.HTML object>
```

1.1 Objectifs des prochaines séances

Connaissez-vous l'application Pocket ? C'est une application qui simplifie le bookmarking. Elle prend la forme d'une extension Chrome / Firefox. Quand on tombe sur un site intéressant, on peut le bookmaker, et ajouter, ou non, des tags pour "qualifier" le contenu. Cette application répond au besoin de conserver le contenu web pertinent et de le classer.
Au cours des prochaines séances, nous allons construire un outil de machine learning qui : - se connecte à un compte pocket - récupère les sites bookmarqués et les tags éventuels - à partir des articles taggés, prédit les meilleurs tags des sites non-taggés - tag les articles non taggés
Bref, nous allons concevoir un programme de classification automatique des articles !

1.2 Objectif de la séance

- Créer un compte Pocket
- S'authentifier auprès de l'API
- Populer le compte avec des données via l'API
- Récupérer les données via l'API
- Scraper les sites bookmarqués pour enrichir les données

1.3 Mais c'est quoi une API ?

On vous explique tout ici :
- Définition - Les API qui existent - Comment parler à une API ?
Donc un API est une interface permettant de *communiquer* avec une application. En général, on veut récupérer des données. Donc la communication consiste à envoyer une requete HTTP (le plus souvent GET ou POST) et à récupérer des données, souvent au format json. Ici nous voulons récupérer des données d'un compte utilisateur pocket.
Pour savoir comment on communique précisément avec l'API de pocket, il n'y a pas de secret : il faut lire la doc de ceux qui l'ont codée. On vous a simplifié un peu les étapes ci-après.

1.4 Création d'un compte Pocket

Créer un compte sur https://getpocket.com/signup?ep=4. Il n'y a pas de vérification d'email, donc vous pouvez mettre un faux mail.
Aller sur la *console developer* de pocket: https://getpocket.com/developer/apps/index.php
Cliquer sur CREATE AN APPLICATION
Compléter le formulaire comme suit (vous pouvez changer le nom de l'application et la description)
Cliquer sur CREATE APPLICATION

[2]:
```
import keyring
import os
CONSUMER_KEY = keyring.get_password("web", "ensae_teaching_cs,pocket")
```

Vous en aurez besoin pour vous connecter à l'API de pocket.

1.5 Authentification

D'abord il faut s'authentifier
Protocole utilisé ici : OAUTH2 (très classique).
6 étapes donc avant d'avoir le droit de récupérer les données. De temps en temps, il existe une librairie python. C'est notre cas : https://github.com/tapanpandita/pocket. On va s'en servir pour s'authentifier. Mais pas pour récupérer les données (elle n'est plus à jour pour faire ça).

1.5.1 Etape 1 : Obtenir un code d'authorisation => get_request_token

[3]:
```
import pocket
from pocket import Pocket

REDIRECT_URI = "http://localhost:8888/redirect"
# c'est l'url à laquelle vous allez rediriger l'utilisateur (ici, vous) après que␣
 ↪pocket a authentifié l'utilisateur (vous)
REQUEST_TOKEN = Pocket.get_request_token(consumer_key=CONSUMER_KEY,␣
 ↪redirect_uri=REDIRECT_URI)
```

1.5.2 Etape 2: Authoriser l'accès

Il faut le faire à chaque exécution du notebook.

[4]:
```
# Enlever les guillemets autour de REQUEST_TOKEN
url = "https://getpocket.com/auth/authorize?request_token={0}&redirect_uri={1}".
 ↪format("REQUEST_TOKEN", REDIRECT_URI)
print("Aller à l'url : \n" + url)
```

```
Aller à l'url :
https://getpocket.com/auth/authorize?request_token=REQUEST_TOKEN&redirect_uri=ht
tp://localhost:8888/redirect
```

Cliquer sur Autoriser.

1.5.3 Etape 3: Récupérer le token d'accès

[5]:
```
try:
    USER_CREDENTIALS = Pocket.get_credentials(consumer_key=CONSUMER_KEY,␣
 ↪code=REQUEST_TOKEN)
except Exception as e:
```

```
    print(e)
# print(USER_CREDENTIALS)
```

[6]:
```
ACCESS_TOKEN = USER_CREDENTIALS['access_token']
# print(ACCESS_TOKEN)
```

1.6 Chargement de données sur le nouveau compte

Comme vous venez de créer un compte, vous n'avez pas encore d'articles sauvegardés. On vous a préparé un peu moins de 500 articles (format json). Un tiers de ces articles sont taggés (catégorisés). Un article peut comprendre un ou plusieurs tags.
On vous rappelle que l'objectif à termes sera de prédire les meilleurs tags pour les articles non taggés, étant donnés les mots qui caractérisent ces articles (titre, résumé, et ensemble des mots présents dans le html de la page).

1.6.1 Chargement du fichier json en mémoire

[7]:
```
import json

with open('./images/data_pocket.json') as fp:
    data = json.load(fp)
```

On affiche le premier élément.

[8]:
```
from pprint import pprint
keys = list(data.keys())
pprint(data[keys[0]])
```

```
{'authors': {'2832761': {'author_id': '2832761',
                         'item_id': '1883956314',
                         'name': 'float',
                         'url': ''}},
 'excerpt': 'Il est impossible d'écrire un programme sans utiliser de '
            'variable. Ce terme désigne le fait d'attribuer un nom ou '
            'identificateur à des informations : en les nommant, on peut '
            'manipuler ces informations beaucoup plus facilement.',
 'favorite': '0',
 'given_title': 'Types et variables du langage python ¿  Programmation avec le '
                'langage Python',
 'given_url':
'http://www.xavierdupre.fr/app/teachpyx/helpsphinx/c_lang/types.html',
 'has_image': '1',
 'has_video': '0',
 'image': {'height': '0',
           'item_id': '1883956314',
           'src': 'http://www.xavierdupre.fr/app/teachpyx/helpsphinx/_images/mat
h/a283b6104f42fc7a8bf845790aa022ac525329f0.png',
           'width': '0'},
 'images': {'1': {'caption': '',
                  'credit': '',
                  'height': '0',
                  'image_id': '1',
                  'item_id': '1883956314',
                  'src': 'http://www.xavierdupre.fr/app/teachpyx/helpsphinx/_ima
```

```
ges/math/a283b6104f42fc7a8bf845790aa022ac525329f0.png',
                  'width': '0'},
            '2': {'caption': '',
                  'credit': '',
                  'height': '0',
                  'image_id': '2',
                  'item_id': '1883956314',
                  'src': 'http://www.xavierdupre.fr/app/teachpyx/helpsphinx/_ima
ges/math/d88bf614100dc9561ab4b951b22f965c77da355d.png',
                  'width': '0'},
            '3': {'caption': '',
                  'credit': '',
                  'height': '0',
                  'image_id': '3',
                  'item_id': '1883956314',
                  'src': 'http://www.xavierdupre.fr/app/teachpyx/helpsphinx/_ima
ges/math/e39ddf05c56c382e29aa91aaceca2d5de5c4bd29.png',
                  'width': '0'}},
 'is_article': '0',
 'is_index': '0',
 'item_id': '1883956314',
 'resolved_id': '1883956314',
 'resolved_title': 'Types et variables du langage pythonű',
 'resolved_url':
'http://www.xavierdupre.fr/app/teachpyx/helpsphinx/c_lang/types.html',
 'sort_id': 0,
 'status': '0',
 'tags': {'python': {'item_id': '1883956314', 'tag': 'python'}},
 'time_added': '1507548626',
 'time_favorited': '0',
 'time_read': '0',
 'time_updated': '1507548631',
 'word_count': '7921'}
```

1.6.2 Exercice 1

Chargement des données du json dans le compte nouvellement créé.
Pour communiquer avec une API, il faut envoyer des requêtes HTTP. Ici on veut ajouter les données du fichier json ("given_url" et "tags") dans le compte Pocket.
Que nous dit la doc ? Consulter https://getpocket.com/developer/docs/v3/add.
Par exemple, pour l'ajout d'un seul item, la doc nous donne l'url à laquelle il faut envoyer une requête (https://getpocket.com/v3/add), et la méthode qu'il faut employer. Ici, il s'agit d'une méthode POST.
Pour l'ajout de plusieurs items, il faut consulter https://getpocket.com/developer/docs/v3/modify. Il faut aussi une méthode POST.
Pour envoyer une requête en python, il y a plusieurs solutions. Un des plus simples consiste à utiliser la librairie requests.
A vous de jouer ! Commencez par un seul (par exemple http://docs.python-requests.org/en/master/user/quickstart/ avec les tags "python, requests"), puis si cela a fonctionné, vous pouvez uploader tout le json. Indice pour passer plusieurs items : penser à encoder les données de la requete en JSON, comme indiqué dans la doc. La documentation de requests indique comment faire ici : http://docs.python-requests.org/en/master/user/quickstart/#more-complicated-post-requests, json = ...
Pour voir si cela a marché, il suffit d'aller sur votre compte Pocket :)

1.6.3 Exercice 1 - correction

Ajouter un seul item

[9]:
```
import requests

data_test = {"consumer_key":CONSUMER_KEY,
"access_token":ACCESS_TOKEN,
"url":"http://docs.python-requests.org/en/master/user/quickstart/",
"tags": "python, requests"}

one_item = requests.post('https://getpocket.com/v3/add', data = data_test)
```

Ajouter l'ensemble des items

[10]:
```
list_action_add = []
for k,v in data.items():
    if  'tags' in v:
        list_action_add.append({'action':'add',
                                'url': v['given_url'],
                                'tags': list(v['tags'].keys())
                               })
list_action_add[0:10]
```

[10]:
```
[{'action': 'add',
  'tags': ['python'],
  'url': 'http://www.xavierdupre.fr/app/teachpyx/helpsphinx/c_lang/types.html'},
 {'action': 'add',
  'tags': ['boilerplate', 'lewagon', 'react', 'redux', 'router'],
  'url': 'https://github.com/lewagon/redux-router-boilerplate'},
 {'action': 'add',
  'tags': ['boilerplate', 'lewagon', 'react', 'redux'],
  'url': 'https://github.com/lewagon/redux-boilerplate'},
 {'action': 'add',
  'tags': ['boilerplate', 'lewagon', 'react'],
  'url': 'https://github.com/lewagon/react-boilerplate'},
 {'action': 'add',
  'tags': ['api', 'isomorphic'],
  'url': 'https://www.safaribooksonline.com/library/view/building-isomorphic-
javascript/9781491932926/ch04.html'},
 {'action': 'add',
  'tags': ['lewagon', 'react', 'setup'],
  'url': 'https://github.com/lewagon/react-redux-
challenges/tree/master/01-Tooling/01-Setup'},
 {'action': 'add',
  'tags': ['mailoop'],
  'url': 'https://console.cloud.google.com/apis/api/gmail.googleapis.com/overvie
w?project=parking-spot-161222'},
 {'action': 'add',
  'tags': ['as_json', 'rails'],
  'url': 'http://jonathanjulian.com/2010/04/rails-to_json-or-as_json/'},
 {'action': 'add',
  'tags': ['lewagon', 'react'],
  'url': 'https://github.com/lewagon/react-redux-challenges'},
 {'action': 'add',
```

```
  'tags': ['api', 'type of connections'],
  'url': 'https://www.codecademy.com/programs/ac3626878e13008ae7184c69089aab0c/i
tems/42276cdf573c1ac09591a0465853af34'}]
```

[11]:
```
payload = {'consumer_key':CONSUMER_KEY,
           'access_token':ACCESS_TOKEN,
           'actions': list_action_add}
```

[12]:
```
r = requests.post('https://getpocket.com/v3/send', json = payload)
r
```

[12]:
```
<Response [200]>
```

Vous pouvez aller voir votre compte, presque 500 items à analyser :)

1.7 Récupération des données disponibles dans l'API

1.7.1 Exercice 2

Récupérer les urls et les tags des items qui contiennent le tag "python".
C'est par ici : https://getpocket.com/developer/docs/v3/retrieve. A vous de jouer !

1.7.2 Exercice 2 - correction

[13]:
```
import requests
items = {"consumer_key":CONSUMER_KEY,
"access_token":ACCESS_TOKEN,
"tag": "python",
"detailType":"complete"}

exo2_items = requests.post('https://getpocket.com/v3/get', data = items)
exo2_list = [v for k,v in exo2_items.json()['list'].items()]
```

[14]:
```
[v['given_url'] for k,v in exo2_items.json()['list'].items()]
```

[14]:
```
['https://pythonprogramming.net/',
 'http://www.mikesboyle.com/post/117202964694/python-nltk-wtf-chapter-1-notes-
on-things-that',
 'https://jakevdp.github.io/blog/2013/06/15/numba-vs-cython-take-2/',
 'https://blog.rstudio.org/2016/03/29/feather/',
 'http://nbviewer.jupyter.org/github/ptwobrussell/Mining-the-Social-Web-2nd-
Edition/tree/master/ipynb/',
 'https://marcobonzanini.com/2015/06/16/mining-twitter-data-with-python-and-js-
part-7-geolocation-and-interactive-maps/',
 'http://www.nltk.org/book/ch03.html',
 'http://blog.fouadhamdi.com/introduction-a-nltk/',
 'https://qbox.io/blog/building-an-elasticsearch-index-with-python',
 'https://tryolabs.com/blog/2015/02/17/python-elasticsearch-first-steps/',
 'http://www.xavierdupre.fr/app/teachpyx/helpsphinx/c_lang/types.html',
 'http://docs.python-requests.org/en/master/user/quickstart/']
```

[15]:
```
[', '.join(list(v['tags'].keys())) for k,v in exo2_items.json()['list'].items()]
```

[15]:
```
['python',
 'nlp, python',
 'python',
 'python',
```

```
 'python',
 'python, tagerstreet',
 'nltk, python, tokenize',
 'french, nlp, nltk, python, tokenize, tokenizer',
 'elastic-search, python',
 'elastic-search, elasticsearch, python, tutorial',
 'python',
 'python, requests']
```

1.7.3 Exercice 3

Récupérer les urls et les titres des items qui contiennent le mot "python" dans le titre ou l'url

1.7.4 Exercice 3 - correction

[16]:
```
items = {"consumer_key":CONSUMER_KEY,
"access_token":ACCESS_TOKEN,
"search": "python",
"detailType":"complete"}

exo3_items = requests.post('https://getpocket.com/v3/get', data = items)
exo3_list = [v for k,v in exo3_items.json()['list'].items()]
```

[17]:
```
[v['given_url'] for k,v in exo3_items.json()['list'].items()]
```

[17]:
```
['http://stackoverflow.com/questions/9663918/how-can-i-tag-and-chunk-french-
text-using-nltk-and-python',
 'https://jakevdp.github.io/blog/2015/08/14/out-of-core-dataframes-in-python/',
 'https://pythonprogramming.net/',
 'https://pypi.python.org/pypi/munkres/',
 'http://www.mikesboyle.com/post/117202964694/python-nltk-wtf-chapter-1-notes-
on-things-that',
 'https://blog.rstudio.org/2016/03/29/feather/',
 'https://realpython.com/blog/python/setting-up-sublime-text-3-for-full-stack-
python-development/',
 'https://marcobonzanini.com/2015/06/16/mining-twitter-data-with-python-and-js-
part-7-geolocation-and-interactive-maps/',
 'http://okomestudio.net/biboroku/?p=2375',
 'https://qbox.io/blog/building-an-elasticsearch-index-with-python',
 'https://tryolabs.com/blog/2015/02/17/python-elasticsearch-first-steps/',
 'https://blog.dominodatalab.com/ab-testing-with-hierarchical-models-in-
python/',
 'http://www.xavierdupre.fr/app/teachpyx/helpsphinx/c_lang/types.html',
 'http://docs.python-requests.org/en/master/user/quickstart/']
```

[18]:
```
[v['resolved_title'] for k,v in exo3_items.json()['list'].items()]
```

[18]:
```
['How can I tag and chunk French text using NLTK and Python?',
 'Out-of-Core Dataframes in Python: Dask and OpenStreetMap',
 'Python Programming Tutorials',
 'munkres 1.0.9',
 'Python NLTK WTF, Chapter 1: Notes on things that don't work right',
 'Feather: A Fast On-Disk Format for Data Frames for R and Python, powered by
Apache Arrow',
 'Setting Up Sublime Text 3 for Full Stack Python Development',
```

```
 'Mining Twitter Data with Python (and JS) ¿  Part 7: Geolocation and Interactive
Maps',
 'Interpreting A/B Test using Python',
 'Build an Elasticsearch Index with Python¿ Machine Learning Series, Part 1',
 'Python + Elasticsearch. First steps.',
 'A/B Testing with Hierarchical Models in Python',
 'Types et variables du langage pythonű',
 'Quickstartű']
```

Dans le fichier qui servira à la catégorisation automatique, nous allons avoir besoin : de l'url, du titre, de l'extrait, des tags et du contenu.
L'api ne permet pas d'accéder au contenu des sites épinglés. Mais nous pouvons le récupérer grâce à l'url.

1.7.5 Exercice 4

Constituer d'abord un DataFrame avec les champs resolved_url, resolved_title, excerpt, et les tags des items comprenant le terme python.

1.7.6 Exercice 4 - correction

[19]:
```
items = {"consumer_key":CONSUMER_KEY,
"access_token":ACCESS_TOKEN,
"detailType":"complete"}

exo4_items = requests.post('https://getpocket.com/v3/get', data = items)
len(exo4_items.json()['list'])
```

[19]:
```
296
```

[20]:
```
exo4_list = [v for k,v in exo4_items.json()['list'].items() if 'Python' in␣
 ↪v['excerpt']]
```

[21]:
```
import pandas as p

df2 = p.
 ↪DataFrame(exo2_list)[['item_id','resolved_url','resolved_title','excerpt','tags']]
df3 = p.
 ↪DataFrame(exo3_list)[['item_id','resolved_url','resolved_title','excerpt','tags']]
df4 = p.
 ↪DataFrame(exo4_list)[['item_id','resolved_url','resolved_title','excerpt','tags']]
```

[22]:
```
import numpy as np
df = p.merge(df2, df3, on='item_id', how='outer', suffixes=('', '_df3'))
df = p.merge(df,  df4, on='item_id', how='outer', suffixes=('', '_df4'))

def complete_data(x,y,z):
    if x != x:
        if y != y:
            return z
        else:
            return y
    else:
        return x

df['url'] = np.vectorize(complete_data)(df['resolved_url'], df['resolved_url_df3'],␣
 ↪df['resolved_url_df4'])
```

```
df['title'] = np.vectorize(complete_data)(df['resolved_title'],␣
 ↪df['resolved_title_df3'], df['resolved_title_df4'])
df['excerpt'] = np.vectorize(complete_data)(df['excerpt'], df['excerpt_df3'],␣
 ↪df['excerpt_df4'])
df['tags'] = np.vectorize(complete_data)(df['tags'], df['tags_df3'], df['tags_df4'])
df = df[['item_id','url','title','excerpt', 'tags']]
df['tags'] = df['tags'].apply(lambda x: ','.join(list(x.keys())) if x==x else x)
df
```

[22]:

	item_id	url	title	excerpt	tags
0	680797791	https://pythonprogramming.net/	Python Programming Tutorials		
1	1072497525	http://www.mikesboyle.com/post/117202964694/py...	Python NLTK WTF, Chapter 1: Notes on things th...		
2	378831480	https://jakevdp.github.io/blog/2013/06/15/numb...	Numba vs. Cython: Take 2		
3	1240472260	https://blog.rstudio.org/2016/03/29/feather/	Feather: A Fast On-Disk Format for Data Frames...		
4	1056127688	http://nbviewer.jupyter.org/github/ptwobrussel...	Jupyter Notebook Viewer		
5	957402029	https://marcobonzanini.com/2015/06/16/mining-t...	Mining Twitter Data with Python (and JS) ¿ Par...		
6	241420475	http://www.nltk.org/book/ch03.html	3 Processing Raw Text		
7	1057169119	http://blog.fouadhamdi.com/introduction-a-nltk/	Introduction à l'analyse de texte avec nltk - ...		
8	1064802343	https://qbox.io/blog/building-an-elasticsearch...	Build an Elasticsearch Index with Python¿ Machi...		
9	1395948696	https://tryolabs.com/blog/2015/02/17/python-el...	Python + Elasticsearch. First steps.		
10	1883956314	http://www.xavierdupre.fr/app/teachpyx/helpsph...	Types et variables du langage pythonű		
11	427014788	http://docs.python-requests.org/en/master/user...	Quickstartű		
12	295141437	http://stackoverflow.com/questions/9663918/how...	How can I tag and chunk French text using NLTK...		
13	1011618630	https://jakevdp.github.io/blog/2015/08/14/out-...	Out-of-Core Dataframes in Python: Dask and Ope...		
14	833287792	https://pypi.python.org/pypi/munkres/	munkres 1.0.9		
15	687517654	https://realpython.com/blog/python/setting-up-...	Setting Up Sublime Text 3 for Full Stack Pytho...		
16	829788447	http://okomestudio.net/biboroku/?p=2375	Interpreting A/B Test using Python		
17	1014684096	https://blog.dominodatalab.com/ab-testing-with...	A/B Testing with Hierarchical Models in Python		
18	241420549	http://www.nltk.org/book/ch04.html			
19	952475069	https://people.duke.edu/~ccc14/sta-663/Optimiz...	Optimization bake-offű		
20	163291612	https://blog.miguelgrinberg.com/post/the-flask...			
21	415947413	http://xavierdupre.fr/			

```
20     The Flask Mega-Tutorial, Part I: Hello, World!
21                 Xavier Dupré, ENSAE, Microsoft Bing

                                                 excerpt  \
0   Learn how to use Python with Pandas, Matplotli...
1   If you're reading this post, you're probably a...
2   Last summer I wrote a post comparing the perfo...
3   This past January, we (Hadley and Wes) met and...
4   Delivered by Fastly, Rendered by Rackspace nbv...
5   Geolocation is the process of identifying the ...
6   The most important source of texts is undoubte...
7   nltk est une librairie python très utile pour ...
8   In this first article, we're going to set up s...
9   Lately, here at Tryolabs, we started gaining i...
10  Il est impossible d'écrire un programme sans u...
11  Eager to get started? This page gives a good i...
12  If I were to do (1), I imagine I would need to...
13  In recent months, a host of new tools and pack...
14  The Munkres module provides an implementation ...
15  Sublime Text 3 (ST3) is lightweight, cross-pla...
16  Suppose we ran an A/B test with two different ...
17  In this post, I discuss a method for A/B testi...
18  By now you will have a sense of the capabiliti...
19  Python is a high-level interpreted language, w...
20  This is the first article in a series where I ...
21  Blog  Enseignements ENSAE / Teachings More mat...

                                          tags
0                                       python
1                                   nlp,python
2                                       python
3                                       python
4                                       python
5                           python,tagerstreet
6                         nltk,python,tokenize
7     french,nlp,nltk,python,tokenize,tokenizer
8                         elastic-search,python
9   elastic-search,elasticsearch,python,tutorial
10                                      python
11                              python,requests
12                                          nlp
13                                 data science
14                                 data science
15                                      sublime
16                                       abtest
17                                       abtest
18                                          nlp
19                                          nlp
20                                      web dev
21                                          NaN
```

1.8 Compléter les données avec du webscraping

1.8.1 Mais c'est quoi le webscraping ?

On vous explique tout ici :
- Définition - Un détour par le Web : comment fonctionne un site ? - Scrapper avec python

1.8.2 Beautiful Soup

Reprenons notre liste de sites sur python et analysons le contenu HTML.

[23]:
```
from bs4 import BeautifulSoup
```

[24]:
```
from pprint import pprint

#première url de la liste
url = df['url'].iloc[0]
print(url)

# récupération du contenu (même librairie que pour les requetes api =>
#requests est une librairie qui permet de faire des requetes http, que cela soit pour␣
 ↪des api ou du webscraping)

r = requests.get(url)

#pprint : librairie pour "pretty print" (essayer sans : on voit pas grand chose)
text = r.text if len(r.text) < 1000 else r.text[:1000] + "\n..."
pprint(text)
```

```
https://pythonprogramming.net/
('<html>\n'
 '\t<head>\n'
 '\t\t\n'
 '\t\t<!-- \n'
 '\t\tpalette:\n'
 '\t\tdark blue: #003F72\n'
 '\t\tyellow: #FFD166\n'
 '\t\tsalmon: #EF476F\n'
 '\t\toffwhite: #e7d7d7\n'
 '\t\tLight Blue: #118AB2\n'
 '\t\tLight green: #7DDF64\n'
 '\t\t-->\n'
 '\n'
 '\t\t<meta name="viewport" content = "width=device-width, '
 'initial-scale=1.0">\n'
 '\t\t<title>Python Programming Tutorials</title>\n'
 '\n'
 '\t\t<meta name="description" content="Python Programming tutorials from '
 'beginner to advanced on a massive variety of topics. All video and text '
 'tutorials are free.">\n'
 '\n'
 '\t\t<link rel="shortcut icon" href="/static/favicon.ico">\n'
 '\t\t<link rel="stylesheet" href="/static/css/materialize.min.css">\n'
 '        <link href="https://fonts.googleapis.com/icon?family=Material+Icons" '
 'rel="stylesheet">\n'
 '        <meta name="google-site-verification" '
```

```
'content="3fLok05gk5gGtWd_VSXbSSSH27F2kr1QqcxYz9vYq2k" />\n'
'        <link rel="stylesheet" type="text/css" '
'href="/static/css/bootstrap.css">\n'
'\n'
'\t\t\n'
'\t\t  <!-- Compiled and minified CSS -->\n'
'\n'
'\t\t<!-- Compiled and minified JavaScript -->\n'
'\n'
'\t\t<script src="https://code.jquery.com/jquery-2.1.4.min.js\n'
'...')
```

[25]:
```
# on stock le contenu html dans la variable html
html = r.text

# on "parse" le html grâce à la librairie beautiful soup
soup = BeautifulSoup(html, "html5lib")

# c'est plus "joli" encore que pprint et surtout,
# il y a plein de méthodes pour extraire les informations que l'on souhaite
soup
```

[25]:
```
<html><head>

                <!--
                palette:
                dark blue: #003F72
                yellow: #FFD166
                salmon: #EF476F
                offwhite: #e7d7d7
                Light Blue: #118AB2
                Light green: #7DDF64
                -->

                <meta content="width=device-width, initial-scale=1.0"
name="viewport"/>
                <title>Python Programming Tutorials</title>

                <meta content="Python Programming tutorials from beginner to
advanced on a massive variety of topics. All video and text tutorials are free."
name="description"/>

                <link href="/static/favicon.ico" rel="shortcut icon"/>
                <link href="/static/css/materialize.min.css" rel="stylesheet"/>
        <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet"/>
        <meta content="3fLok05gk5gGtWd_VSXbSSSH27F2kr1QqcxYz9vYq2k"
name="google-site-verification"/>
        <link href="/static/css/bootstrap.css" rel="stylesheet"
type="text/css"/>


                  <!-- Compiled and minified CSS -->
```

```
                <!-- Compiled and minified JavaScript -->

                <script
src="https://code.jquery.com/jquery-2.1.4.min.js"></script>
                <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/
0.97.3/js/materialize.min.js"></script>



        <style>
                @media (min-width:992px) {
                #aside {
                        width:250px;
                }
                pre { tab-size: 4;}
                .btn {background-color:#FFD166;
                          color:#000;
                      height:auto;
                          font-color:#000;
                          }
                .btn:hover {background-color:#FFD166;
                                    }


                </style>

                <!-- Google Tracking -->
                <script>
(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
                        (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new
Date();a=s.createElement(o),
m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
                        })(window,document,'script','//www.google-
analytics.com/analytics.js','ga');

                        ga('create', 'UA-51891827-1', 'auto');
                        ga('send', 'pageview');

                </script>




        </head>

        <body>

                <div class="navbar-fixed">
                        <nav style="background-color:#003F72">
                                <div class="nav-wrapper container">
                                        <a class="brand-logo" href="/"><img
class="img-responsive" src="/static/images/mainlogowhitethick.jpg"
style="width:50px; height;50px; margin-top:5px"/></a>
```

```
                                        <a class="button-collapse" data-
activates="navsidebar" href="#"><i class="mdi-navigation-menu"></i></a>
                                        <ul class="right hide-on-med-and-down">
                                        <li><i class="material-
icons">search</i></li>
                                        <li>
                                                <form action="/search/?l=hi"
id="searchform" method="get" style="height:50px; padding-top:10px">
                                                        <input id="search"
name="q" placeholder="search" style="font-size:16px" type="search"/>
                                                </form>
                                        </li>

                        <li><a href="/">Home</a></li>
                        <li><a class="tooltipped" data-delay="50" data-
position="bottom" data-tooltip="sudo apt-get upgrade" href="/+=1/">+=1</a></li>
                        <li><a href="/store/python-hoodie/">Store</a></li>

                        <li><a href="/community/">Community</a></li>
                                <li><a href="/login/">Log in</a></li>
                <li><a href="/register/">Sign up</a></li>


                                        </ul>
                                        <ul class="side-nav" id="navsidebar">
                                        <li>
                                                <form action="/search/?l=hi"
id="searchform" method="get" style="height:50px; padding-top:10px">
                                                        <input id="search"
name="q" placeholder="search" style="font-size:16px" type="search"/>
                                                </form>
                                        </li>

                        <li><a href="/">Home</a></li>
                        <li><a class="tooltipped" data-delay="50" data-
position="bottom" data-tooltip="sudo apt-get upgrade" href="/+=1/">+=1</a></li>
                        <li><a href="/store/python-hoodie/">Store</a></li>

                        <li><a href="/community/">Community</a></li>
                                <li><a href="/login/">Log in</a></li>
                <li><a href="/register/">Sign up</a></li>


                                        </ul>
                                </div>
                        </nav>
                </div>

                <!-- main content -->
                <div class="container" style="max-width:1500px; min-
height:100%">
```

```
        <div class="row">
        <!-- size 4 on a large screen, 12 on a small, 6 on a med -->
                <div class="col l4 s12 m6">
                        <a class="waves-effect waves-light" href="/data-
analysis-tutorials/" style="color:#000">
                        </a><div class="card-panel hoverable"
style="height:360px; background-color:#FFFFFF;"><a class="waves-effect waves-
light" href="/data-analysis-tutorials/" style="color:#000">
                                <div class="card-image">
                                  <img class="responsive-img"
src="/static/images/categories/3D-Matplotlib.png" style="width:250px"/>
                                  <br/><span class="card-title"><strong>Data
Analysis</strong></span>
                                </div>
                                <div class="card-content">
                                  <p>Learn how to use Python with Pandas,
Matplotlib, and other modules to gather insights from and about your data.</p>
                                </div>
                                </a><div class="card-action right-align"><a
class="waves-effect waves-light" href="/data-analysis-tutorials/"
style="color:#000">
                                  </a><a class="waves-effect waves-light btn"
href="/data-analysis-tutorials/" style="color:#000000; background-
color:#FFD166">View</a>
                                </div>
            </div>

                </div>

                <div class="col l4 s12 m6">
                        <a class="waves-effect waves-light" href="/robotics-
tutorials/" style="color:#000">
                        </a><div class="card-panel hoverable"
style="height:360px; background-color:#FFFFFF;"><a class="waves-effect waves-
light" href="/robotics-tutorials/" style="color:#000">
                                <div class="card-image">
                                  <img class="responsive-img"
src="/static/images/categories/robotics_with_raspberry_pi_gopigo.jpg"
style="width:250px"/>
                                  <br/><span class="card-
title"><strong>Robotics</strong></span>
                                </div>
                                <div class="card-content">
                                  <p>Control hardware with Python programming
and the Raspberry Pi.</p>
                                </div>
                                </a><div class="card-action right-align"><a
class="waves-effect waves-light" href="/robotics-tutorials/" style="color:#000">
                                  </a><a class="waves-effect waves-light btn"
```

```
href="/robotics-tutorials/" style="color:#000000; background-
color:#FFD166">View</a>
                                </div>
            </div>

                </div>

                <div class="col l4 s12 m6">
                        <a class="waves-effect waves-light" href="/web-
development-tutorials/" style="color:#000">
                        </a><div class="card-panel hoverable"
style="height:360px; background-color:#FFFFFF;"><a class="waves-effect waves-
light" href="/web-development-tutorials/" style="color:#000">
                                <div class="card-image">
                                  <img class="responsive-img"
src="/static/images/categories/webdev.png" style="width:250px"/>
                                  <br/><span class="card-title"><strong>Web
Development</strong></span>
                                </div>
                                <div class="card-content">
                                  <p>How to develop websites with either the
Flask or Django frameworks for Python.</p>
                                </div>
                                </a><div class="card-action right-align"><a
class="waves-effect waves-light" href="/web-development-tutorials/"
style="color:#000">
                                  </a><a class="waves-effect waves-light btn"
href="/web-development-tutorials/" style="color:#000000; background-
color:#FFD166">View</a>
                                </div>
            </div>

                </div>



                <div class="col l4 s12 m6">
                        <a class="waves-effect waves-light" href="/game-
development-tutorials/" style="color:#000">
                        </a><div class="card-panel hoverable"
style="height:360px; background-color:#FFFFFF;"><a class="waves-effect waves-
light" href="/game-development-tutorials/" style="color:#000">
                                <div class="card-image">
                                  <img class="responsive-img"
src="/static/images/categories/pygame-tutorials.png" style="width:250px"/>
                                  <br/><span class="card-title"><strong>Game
Development</strong></span>
                                </div>
                                <div class="card-content">
                                  <p>Create your own games with Python's PyGame
library, or check out the multi-platform Kivy.</p>
                                </div>
                                </a><div class="card-action right-align"><a
```

```
class="waves-effect waves-light" href="/game-development-tutorials/"
style="color:#000">
                                  </a><a class="waves-effect waves-light btn"
href="/game-development-tutorials/" style="color:#000000; background-
color:#FFD166">View</a>
                              </div>
            </div>

                </div>



                <div class="col l4 s12 m6">
                        <a class="waves-effect waves-light" href="/python-
fundamental-tutorials/" style="color:#000">
                        </a><div class="card-panel hoverable"
style="height:360px; background-color:#FFFFFF;"><a class="waves-effect waves-
light" href="/python-fundamental-tutorials/" style="color:#000">
                              <div class="card-image">
                                <img class="responsive-img"
src="/static/images/categories/python-3-basics.png" style="width:250px"/>
                                <br/><span class="card-title"><strong>Python
Fundamentals</strong></span>
                              </div>
                              <div class="card-content">
                                <p>Learn the basic and intermediate Python
fundamentals.</p>
                              </div>
                              </a><div class="card-action right-align"><a
class="waves-effect waves-light" href="/python-fundamental-tutorials/"
style="color:#000">
                                  </a><a class="waves-effect waves-light btn"
href="/python-fundamental-tutorials/" style="color:#000000; background-
color:#FFD166">View</a>
                              </div>
            </div>

                </div>



                <!--<div class="col l4 s12 m6">

                        <div class="card-panel hoverable" style="height:360px;
background-color:#FFFFFF;">
                        <a href="/introduction-to-python-programming/"
class="waves-effect waves-light" style="color:#000">
                              <div class="card-image">
                                <img class="responsive-img"
src="/static/images/categories/python-3-basics.png" style="width:250px">
                                <br><span class="card-
title"><strong>Basics</strong></span>
```

```
                                </div>
                                <div class="card-content">
                                  <p>Just getting started?</p>
                                  <p>Not a problem, learn the basics of
programming with Python 3 here!</p>
                                </div>
                                <div class="card-action right-align">
                                  <a href="/introduction-to-python-programming/"
class="waves-effect waves-light btn" style="color:#000000; background-
color:#FFD166">Start</a>
                                </div>
                                </a>
            </div>


                </div>-->

                <div class="col l4 s12 m6">
                        <a class="waves-effect waves-light" href="/gui-
development-tutorials/" style="color:#000">
                        </a><div class="card-panel hoverable"
style="height:360px; background-color:#FFFFFF;"><a class="waves-effect waves-
light" href="/gui-development-tutorials/" style="color:#000">
                                <div class="card-image">
                                  <img class="responsive-img"
src="/static/images/categories/tkinter-gui-tutorial-with-python-3.png"
style="width:250px"/>
                                  <br/><span class="card-
title"><strong>GUIs</strong></span>
                                </div>
                                <div class="card-content">
                                  <p>Create software with a user interface using
Tkinter, PyQt, or Kivy.</p>
                                </div>
                                </a><div class="card-action right-align"><a
class="waves-effect waves-light" href="/gui-development-tutorials/"
style="color:#000">
                                  </a><a class="waves-effect waves-light btn"
href="/gui-development-tutorials/" style="color:#000000; background-
color:#FFD166">View</a>
                                </div>
            </div>

                </div>
        </div>


                        </div>


                <!--login modal-->

<div class="modal" id="modalLogin">
```

```
                        <div class="modal-content">
                                <h2 class="center-align">Login</h2>
                                <div class="center-align">
                                        <div class="divider"></div>

                                        <form class="col s12">
                                                <div class="row center-align">
                                                        <div class="input-field
col m10">
                                                                <input
class="validate" id="icon_prefix" type="text"/>
                                                                <label
for="icon_prefix">Username</label>
                                                        </div>
                                                        <div class="input-field
col m10">
                                                                <input
class="validate" id="icon_password" type="password"/>
                                                                <label
for="icon_password">Password</label>
                                                        </div>
                                                </div>
                                        </form>


                                        <div class="divider"></div>
                                        <p>
                                                <a class="btn btn-flat white
modal-close" href="#">Cancel</a>
                                                <a class="waves-effect waves-
blue blue btn btn-flat modal-action modal-close" href="#">Login</a>
                                        </p>


                                </div>
                        </div>
                </div>



                <!--Register modal-->

                <div class="modal" id="modalRegister">
                        <div class="modal-content">
                                <h2 class="center-align">Sign up</h2>
                                <div class="center-align">
                                        <div class="divider"></div>

                                        <form action="/register/" method="post">
                                                <div class="input-field
col">
                                                        <input
class="validate" id="username" type="text"/>
```

```
                                                        <label
for="username">Username</label>
                                                </div>
                                                <div class="input-field
col">
                                                        <input
class="validate" id="password" type="password"/>
                                                        <label
for="password">Password</label>
                                                </div>
                                                <div class="input-field
col">
                                                        <input
class="validate" id="confirm" type="password"/>
                                                        <label
for="confirm">Repeat Password</label>
                                                </div>

                                                <div class="input-field
col">
                                                        <input
class="validate" id="email" type="text"/>
                                                        <label
for="email">Email</label>
                                                </div>

                                                <div
class="divider"></div>
                                                <p>
                                                        <a class="btn
btn-flat white modal-close" href="#">Cancel</a>
                                                        <button
class="btn" type="submit" value="Register">Sign Up</button>
                                                </p>

                                        </form>



                                </div>
                        </div>
                </div>




                <script>
                        $(document).ready(function(){
                                $(".button-collapse").sideNav();
                                $('.modal-trigger').leanModal();
                                $('.collapsible').collapsible({
                                  accordion : false // A setting that changes
the collapsible behavior to expandable instead of the default accordion style
```

```
                                });
                                $('#aside').pushpin({ top:110, bottom:500 });
                  });
                </script>
                <script>
                  $(document).ready(function(){
                        $('.collapsible').collapsible({
                          accordion : false // A setting that changes the
collapsible behavior to expandable instead of the default accordion style
                        });
                        $('select').material_select();
                  });
          </script>
          <script>
                function goBack() {
                        window.history.back()
                }
        </script>
                <script src="/static/js/run_prettify.js"
type="text/javascript"></script>



<footer class="page-footer">
          <div class="container">
            <div class="row">
              <div class="col l6 s12">
                <h5 class="white-text">You've reached the end!</h5>

                <p class="grey-text text-lighten-4">Contact:
Harrison@pythonprogramming.net.</p>
                <ul>
                  <li><a class="grey-text text-lighten-3" href="/support-
donate/">Support this Website!</a></li>
                  <li><a class="grey-text text-lighten-3"
href="/consulting/">Hire me</a></li>
                  <li><a class="grey-text text-lighten-3"
href="https://www.facebook.com/pythonprogramming.net/">Facebook</a></li>
                  <li><a class="grey-text text-lighten-3"
href="https://twitter.com/sentdex">Twitter</a></li>
                  <li><a class="grey-text text-lighten-3"
href="https://plus.google.com/+sentdex">Google+</a></li>
                </ul>
              </div>
              <div class="col l4 offset-l2 s12">
                <h6 class="white-text">Legal stuff:</h6>
                <ul>
                  <li><a class="grey-text text-lighten-3"
href="/about/tos/">Terms and Conditions</a></li>
                  <li><a class="grey-text text-lighten-3" href="/about/privacy-
policy/">Privacy Policy</a></li>

                </ul>
```

```
            </div>
          </div>
        </div>
        <a href="https://xkcd.com/353/" target="blank"><p class="grey-text
right" style="padding-right:10px">Programming is a superpower.</p></a>
        <div class="footer-copyright">
          <div class="container">
          I OVER 9000! PythonProgramming.net

          </div>

        </div>
      </footer>
```

```
</body></html>
```

[26]:
```
type(soup)
```

[26]:
```
bs4.BeautifulSoup
```

soup est une instance de la classe BeautifulSoup (que l'on a importé de la librairie bs4). C'est une représentation du code html avec des méthodes pratiques, telles que **find**. **find** permet de trouver la première occurence d'une balise html qu'on lui passe. Par exemple le 1er lien de la page :

[27]:
```
soup.find("a")
```

[27]:
```
<a class="brand-logo" href="/"><img class="img-responsive"
src="/static/images/mainlogowhitethick.jpg" style="width:50px; height;50px;
margin-top:5px"/></a>
```

Si on veut tous les liens, il faut utiliser **findAll**. Cela renvoie une objet qui se comporte comme une liste. On peut itérer dessus, ou facilement le transformer en objet "list" (en faisant list())

[28]:
```
print(type(soup.findAll("a")))
soup.findAll("a")
```

```
<class 'bs4.element.ResultSet'>
```

[28]:
```
[<a class="brand-logo" href="/"><img class="img-responsive"
src="/static/images/mainlogowhitethick.jpg" style="width:50px; height;50px;
margin-top:5px"/></a>,
 <a class="button-collapse" data-activates="navsidebar" href="#"><i class="mdi-
```

```
navigation-menu"></i></a>,
 <a href="/">Home</a>,
 <a class="tooltipped" data-delay="50" data-position="bottom" data-tooltip="sudo
apt-get upgrade" href="/+=1/">+=1</a>,
 <a href="/store/python-hoodie/">Store</a>,
 <a href="/community/">Community</a>,
 <a href="/login/">Log in</a>,
 <a href="/register/">Sign up</a>,
 <a href="/">Home</a>,
 <a class="tooltipped" data-delay="50" data-position="bottom" data-tooltip="sudo
apt-get upgrade" href="/+=1/">+=1</a>,
 <a href="/store/python-hoodie/">Store</a>,
 <a href="/community/">Community</a>,
 <a href="/login/">Log in</a>,
 <a href="/register/">Sign up</a>,
 <a class="waves-effect waves-light" href="/data-analysis-tutorials/"
style="color:#000">
                        </a>,
 <a class="waves-effect waves-light" href="/data-analysis-tutorials/"
style="color:#000">
                                <div class="card-image">
                                  <img class="responsive-img"
src="/static/images/categories/3D-Matplotlib.png" style="width:250px"/>
                                  <br/><span class="card-title"><strong>Data
Analysis</strong></span>
                                </div>
                                <div class="card-content">
                                  <p>Learn how to use Python with Pandas,
Matplotlib, and other modules to gather insights from and about your data.</p>
                                </div>
                                </a>,
 <a class="waves-effect waves-light" href="/data-analysis-tutorials/"
style="color:#000">
                                  </a>,
 <a class="waves-effect waves-light btn" href="/data-analysis-tutorials/"
style="color:#000000; background-color:#FFD166">View</a>,
 <a class="waves-effect waves-light" href="/robotics-tutorials/"
style="color:#000">
                        </a>,
 <a class="waves-effect waves-light" href="/robotics-tutorials/"
style="color:#000">
                                <div class="card-image">
                                  <img class="responsive-img"
src="/static/images/categories/robotics_with_raspberry_pi_gopigo.jpg"
style="width:250px"/>
                                  <br/><span class="card-
title"><strong>Robotics</strong></span>
                                </div>
                                <div class="card-content">
                                  <p>Control hardware with Python programming
and the Raspberry Pi.</p>
                                </div>
                                </a>,
```

```
 <a class="waves-effect waves-light" href="/robotics-tutorials/"
style="color:#000">
                                  </a>,
 <a class="waves-effect waves-light btn" href="/robotics-tutorials/"
style="color:#000000; background-color:#FFD166">View</a>,
 <a class="waves-effect waves-light" href="/web-development-tutorials/"
style="color:#000">
                          </a>,
 <a class="waves-effect waves-light" href="/web-development-tutorials/"
style="color:#000">
                                <div class="card-image">
                                  <img class="responsive-img"
src="/static/images/categories/webdev.png" style="width:250px"/>
                                  <br/><span class="card-title"><strong>Web
Development</strong></span>
                                </div>
                                <div class="card-content">
                                  <p>How to develop websites with either the
Flask or Django frameworks for Python.</p>
                                </div>
                                </a>,
 <a class="waves-effect waves-light" href="/web-development-tutorials/"
style="color:#000">
                                  </a>,
 <a class="waves-effect waves-light btn" href="/web-development-tutorials/"
style="color:#000000; background-color:#FFD166">View</a>,
 <a class="waves-effect waves-light" href="/game-development-tutorials/"
style="color:#000">
                          </a>,
 <a class="waves-effect waves-light" href="/game-development-tutorials/"
style="color:#000">
                                <div class="card-image">
                                  <img class="responsive-img"
src="/static/images/categories/pygame-tutorials.png" style="width:250px"/>
                                  <br/><span class="card-title"><strong>Game
Development</strong></span>
                                </div>
                                <div class="card-content">
                                  <p>Create your own games with Python's PyGame
library, or check out the multi-platform Kivy.</p>
                                </div>
                                </a>,
 <a class="waves-effect waves-light" href="/game-development-tutorials/"
style="color:#000">
                                  </a>,
 <a class="waves-effect waves-light btn" href="/game-development-tutorials/"
style="color:#000000; background-color:#FFD166">View</a>,
 <a class="waves-effect waves-light" href="/python-fundamental-tutorials/"
style="color:#000">
                          </a>,
 <a class="waves-effect waves-light" href="/python-fundamental-tutorials/"
style="color:#000">
                                <div class="card-image">
```

```
                                  <img class="responsive-img"
src="/static/images/categories/python-3-basics.png" style="width:250px"/>
                                  <br/><span class="card-title"><strong>Python
Fundamentals</strong></span>
                                </div>
                                <div class="card-content">
                                  <p>Learn the basic and intermediate Python
fundamentals.</p>
                                </div>
                                </a>,
 <a class="waves-effect waves-light" href="/python-fundamental-tutorials/"
style="color:#000">
                                  </a>,
 <a class="waves-effect waves-light btn" href="/python-fundamental-tutorials/"
style="color:#000000; background-color:#FFD166">View</a>,
 <a class="waves-effect waves-light" href="/gui-development-tutorials/"
style="color:#000">
                            </a>,
 <a class="waves-effect waves-light" href="/gui-development-tutorials/"
style="color:#000">
                                <div class="card-image">
                                  <img class="responsive-img"
src="/static/images/categories/tkinter-gui-tutorial-with-python-3.png"
style="width:250px"/>
                                  <br/><span class="card-
title"><strong>GUIs</strong></span>
                                </div>
                                <div class="card-content">
                                  <p>Create software with a user interface using
Tkinter, PyQt, or Kivy.</p>
                                </div>
                                </a>,
 <a class="waves-effect waves-light" href="/gui-development-tutorials/"
style="color:#000">
                                  </a>,
 <a class="waves-effect waves-light btn" href="/gui-development-tutorials/"
style="color:#000000; background-color:#FFD166">View</a>,
 <a class="btn btn-flat white modal-close" href="#">Cancel</a>,
 <a class="waves-effect waves-blue blue btn btn-flat modal-action modal-close"
href="#">Login</a>,
 <a class="btn btn-flat white modal-close" href="#">Cancel</a>,
 <a class="grey-text text-lighten-3" href="/support-donate/">Support this
Website!</a>,
 <a class="grey-text text-lighten-3" href="/consulting/">Hire me</a>,
 <a class="grey-text text-lighten-3"
href="https://www.facebook.com/pythonprogramming.net/">Facebook</a>,
 <a class="grey-text text-lighten-3"
href="https://twitter.com/sentdex">Twitter</a>,
 <a class="grey-text text-lighten-3"
href="https://plus.google.com/+sentdex">Google+</a>,
 <a class="grey-text text-lighten-3" href="/about/tos/">Terms and
Conditions</a>,
 <a class="grey-text text-lighten-3" href="/about/privacy-policy/">Privacy
```

```
Policy</a>,
 <a href="https://xkcd.com/353/" target="blank"><p class="grey-text right"
style="padding-right:10px">Programming is a superpower.</p></a>]
```

On peut sélectionner des tags qui ont certains attributs css. Par exemple ici, on peut vouloir seulement les liens internes du site. Ils ont la classe “internal”. Les autres “external”.

[29]:
```
list(soup.findAll("a", {"class": "internal"}))
```

[29]:
```
[]
```

Ce n’est pas toujours comme ça, la plupart du temps, il faudra la forme de la valeur de l’attribut href. les liens externes pourront etre identifiés parce qu’ils sont absolus (on indique l’ensemble du lien comme href=“https://docs.python.org/3/reference/expressions.html etc. et non pas un lien relatif comme href=”../c_exception/exception.html (‘..’ signifie "le répertoire parent).
On peut lister toutes les balises h1 et h2 de cette page, en une ligne.

[30]:
```
list(soup.findAll({"h1", "h2"}))
```

[30]:
```
[<h2 class="center-align">Login</h2>, <h2 class="center-align">Sign up</h2>]
```

Naviguer vers les enfants

[31]:
```
child = soup.find("h2").a
child
```

Retrouver le parent

[32]:
```
parent = child.parent if child else None
parent
```

Récupérer tous les enfants

[33]:
```
children = soup.find("h2").findAll("a")
children
```

[33]:
```
[]
```

Récupérer les attributs des éléments html

[34]:
```
[c.attrs for c in children]
```

[34]:
```
[]
```

Accéder à la valeur d’un attribut en particulier

[35]:
```
[c.attrs['href'] for c in children]
```

[35]:
```
[]
```

Jusqu’ici, on passe systématiquement par une balise html. Comment faire si on veut récupérer les élements qui ont une class css, quelle que soit la balise html ?

[36]:
```
internals = soup.findAll("", {"class": "internal"})
internals
```

[36]:
```
[]
```

On a presque fini de passer en revue la librairie. Il reste la notion de “sibling”, pratique pour parcourir un tableau. nextSibling : permet de récupérer l’élément html suivant et “de même niveau” dans l’arbre (le prochain frère), previousSibling, le précédent. findChildren permet de trouver tous les enfants.
Enfin la méthode “get_text()” pour récupérer le contenu des balises html. A appliquer en dernier ! En effet, une fois appliquer, on perd toute trace à l’arbre html. SI vous voulez aller plus loin, la doc est bien faite : https://www.crummy.com/software/BeautifulSoup/bs4/doc/

1.8.3 Exercice 5

Récupérer le contenu des code snippets de la page : http://www.xavierdupre.fr/app/teachpyx/helpsphinx/c_lang/types.h

1.8.4 Exercice 5 - correction

[37]:
```
code_snippets = soup.findAll('pre')
if code_snippets:
    t = code_snippets[0].get_text()
else:
    t = ""
t
```

[37]:
```
''
```

1.8.5 Exercice 6

Récupérer la 3ème colonne (intitulée "exemples") du tableau des opérateurs : http://www.xavierdupre.fr/app/teachpyx/helpsphinx/c_lang/types.html
Astuce : explorer les CSS selectors : https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

1.8.6 Exercice 6 - correction

[38]:
```
table = soup.find('tbody')
if table is not None:
    rows = table.findAll('tr')
    res = [r.select_one("td:nth-of-type(3)").get_text() for r in rows]
else:
    res = "rien du tout"
res
```

[38]:
```
'rien du tout'
```

1.8.7 Exercice 7

Récupérer le contenu des titres (h1, h2, h3, etc.) et des balise p de l'ensemble des liens de notre liste et compter le nombre d'occurences du mot python, et la porportion que cela représente dans l'ensemble des mots

1.8.8 Exercice 7 - correction

[39]:
```
def nb_words(url):
    print(url)
    try:
        html = requests.get(url).text
        soup = BeautifulSoup(html, "html5lib")
        nb_python = 0
        nb_words = 0
        for e in soup.findAll({'h1', 'h2', 'h3', 'h4', 'h5', 'h6', 'p'}):
            text = e.get_text()
            words = text.split(' ')
            nb_words += len(words)
            python_words = [w for w in words if "python" in w.lower()]
            nb_python += len(python_words)
```

```
        return (nb_python, nb_words)
    except:
        return "scraper banned"
```

[40]:
```
df['python_occ'] = df['url'].apply(lambda x: nb_words(x))
```

```
https://pythonprogramming.net/
http://www.mikesboyle.com/post/117202964694/python-nltk-wtf-chapter-1-notes-on-
things-that
https://jakevdp.github.io/blog/2013/06/15/numba-vs-cython-take-2/
https://blog.rstudio.org/2016/03/29/feather/
http://nbviewer.jupyter.org/github/ptwobrussell/Mining-the-Social-Web-2nd-
Edition/tree/master/ipynb/
https://marcobonzanini.com/2015/06/16/mining-twitter-data-with-python-and-js-
part-7-geolocation-and-interactive-maps/
http://www.nltk.org/book/ch03.html
http://blog.fouadhamdi.com/introduction-a-nltk/
https://qbox.io/blog/building-an-elasticsearch-index-with-python
https://tryolabs.com/blog/2015/02/17/python-elasticsearch-first-steps/
http://www.xavierdupre.fr/app/teachpyx/helpsphinx/c_lang/types.html
http://docs.python-requests.org/en/master/user/quickstart/
http://stackoverflow.com/questions/9663918/how-can-i-tag-and-chunk-french-text-
using-nltk-and-python
https://jakevdp.github.io/blog/2015/08/14/out-of-core-dataframes-in-python/
https://pypi.python.org/pypi/munkres/
https://realpython.com/blog/python/setting-up-sublime-text-3-for-full-stack-
python-development
http://okomestudio.net/biboroku/?p=2375
https://blog.dominodatalab.com/ab-testing-with-hierarchical-models-in-python/
http://www.nltk.org/book/ch04.html
https://people.duke.edu/~ccc14/sta-663/Optimization_Bakeoff.html
https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world
http://xavierdupre.fr/
```

[41]:
```
df
```

[41]:

	item_id	url \
0	680797791	https://pythonprogramming.net/
1	1072497525	http://www.mikesboyle.com/post/117202964694/py...
2	378831480	https://jakevdp.github.io/blog/2013/06/15/numb...
3	1240472260	https://blog.rstudio.org/2016/03/29/feather/
4	1056127688	http://nbviewer.jupyter.org/github/ptwobrussel...
5	957402029	https://marcobonzanini.com/2015/06/16/mining-t...
6	241420475	http://www.nltk.org/book/ch03.html
7	1057169119	http://blog.fouadhamdi.com/introduction-a-nltk/
8	1064802343	https://qbox.io/blog/building-an-elasticsearch...
9	1395948696	https://tryolabs.com/blog/2015/02/17/python-el...
10	1883956314	http://www.xavierdupre.fr/app/teachpyx/helpsph...
11	427014788	http://docs.python-requests.org/en/master/user...
12	295141437	http://stackoverflow.com/questions/9663918/how...
13	1011618630	https://jakevdp.github.io/blog/2015/08/14/out-...
14	833287792	https://pypi.python.org/pypi/munkres/
15	687517654	https://realpython.com/blog/python/setting-up-...
16	829788447	http://okomestudio.net/biboroku/?p=2375

```
17  1014684096  https://blog.dominodatalab.com/ab-testing-with...
18   241420549                http://www.nltk.org/book/ch04.html
19   952475069  https://people.duke.edu/~ccc14/sta-663/Optimiz...
20   163291612  https://blog.miguelgrinberg.com/post/the-flask...
21   415947413                             http://xavierdupre.fr/

                                                 title  \
0                         Python Programming Tutorials
1   Python NLTK WTF, Chapter 1: Notes on things th...
2                             Numba vs. Cython: Take 2
3   Feather: A Fast On-Disk Format for Data Frames...
4                              Jupyter Notebook Viewer
5   Mining Twitter Data with Python (and JS) ¿  Par...
6                                3 Processing Raw Text
7   Introduction à l'analyse de texte avec nltk - ...
8   Build an Elasticsearch Index with Python¿ Machi...
9                 Python + Elasticsearch. First steps.
10               Types et variables du langage pythonű
11                                        Quickstartű
12  How can I tag and chunk French text using NLTK...
13  Out-of-Core Dataframes in Python: Dask and Ope...
14                                      munkres 1.0.9
15  Setting Up Sublime Text 3 for Full Stack Pytho...
16                  Interpreting A/B Test using Python
17     A/B Testing with Hierarchical Models in Python
18
19                               Optimization bake-offű
20     The Flask Mega-Tutorial, Part I: Hello, World!
21                  Xavier Dupré, ENSAE, Microsoft Bing

                                                excerpt  \
0   Learn how to use Python with Pandas, Matplotli...
1   If you're reading this post, you're probably a...
2   Last summer I wrote a post comparing the perfo...
3   This past January, we (Hadley and Wes) met and...
4   Delivered by Fastly, Rendered by Rackspace nbv...
5   Geolocation is the process of identifying the ...
6   The most important source of texts is undoubte...
7   nltk est une librairie python très utile pour ...
8   In this first article, we're going to set up s...
9   Lately, here at Tryolabs, we started gaining i...
10  Il est impossible d'écrire un programme sans u...
11  Eager to get started? This page gives a good i...
12  If I were to do (1), I imagine I would need to...
13  In recent months, a host of new tools and pack...
14  The Munkres module provides an implementation ...
15  Sublime Text 3 (ST3) is lightweight, cross-pla...
16  Suppose we ran an A/B test with two different ...
17  In this post, I discuss a method for A/B testi...
18  By now you will have a sense of the capabiliti...
19  Python is a high-level interpreted language, w...
20  This is the first article in a series where I ...
21  Blog  Enseignements ENSAE / Teachings More mat...
```

	tags	python_occ
0	python	(6, 93)
1	nlp,python	(21, 1475)
2	python	(21, 1183)
3	python	(13, 771)
4	python	(0, 195)
5	python,tagerstreet	(10, 2435)
6	nltk,python,tokenize	(58, 11800)
7	french,nlp,nltk,python,tokenize,tokenizer	scraper banned
8	elastic-search,python	(11, 1901)
9	elastic-search,elasticsearch,python,tutorial	(11, 1509)
10	python	(29, 5164)
11	python,requests	(2, 1594)
12	nlp	(5, 1102)
13	data science	(7, 1758)
14	data science	(0, 42)
15	sublime	(18, 2210)
16	abtest	(2, 813)
17	abtest	(4, 3938)
18	nlp	(87, 12724)
19	nlp	(11, 523)
20	web dev	(38, 4358)
21	NaN	(0, 33)

[42]:

[43]: