Importer les bibliothèques numpy et pandas et importer functools de reduce

Entrée [9]:
```
# Importer les bibliothèques nécessaires
```

PART I

Explorer la dataset `BL-Flickr-Images-Book.csv`

NB: faites attentionsau chemin

1 Exploration du dataset

Entrée [7]:
```
# Exporter BL-Flickr-Images-Book.csv dans un data frame nomé df


# Afficher les 5 premières lignes

```

Out[7]:

	Identifier	Edition Statement	Place of Publication	Date of Publication	Publisher	Title	Author	Contributors	Corporate Author	Corporate Contributors	Former owner	Engraver	Issuance type	
0	206	NaN	London	1879 [1878]	S. Tinsley & Co.	Walter Forbes. [A novel.] By A. A	A. A.	FORBES, Walter.	NaN	NaN	NaN	NaN	monographic	http://www.flickr.com
1	216	NaN	London; Virtue & Yorston	1868	Virtue & Co.	All for Greed. [A novel. The dedication signed...	A., A. A.	BLAZE DE BURY, Marie Pauline Rose - Baroness	NaN	NaN	NaN	NaN	monographic	http://www.flickr.com
2	218	NaN	London	1869	Bradbury, Evans & Co.	Love the Avenger. By the author of “All for Gr...	A., A. A.	BLAZE DE BURY, Marie Pauline Rose - Baroness	NaN	NaN	NaN	NaN	monographic	http://www.flickr.com
3	472	NaN	London	1851	James Darling	Welsh Sketches, chiefly ecclesiastical, to the...	A., E. S.	Appleyard, Ernest Silvanus.	NaN	NaN	NaN	NaN	monographic	http://www.flickr.com
4	480	A new edition, revised, etc.	London	1857	Wertheim & Macintosh	[The World in which I live, and my place in it...	A., E. S.	BROOME, John Henry.	NaN	NaN	NaN	NaN	monographic	http://www.flickr.com

2 Jeter un coup d'oeil dans le dataset

Entrée [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8287 entries, 0 to 8286
Data columns (total 15 columns):
Identifier                8287 non-null int64
Edition Statement         773 non-null object
Place of Publication      8287 non-null object
Date of Publication       8106 non-null object
Publisher                 4092 non-null object
Title                     8287 non-null object
Author                    6509 non-null object
Contributors              8287 non-null object
Corporate Author          0 non-null float64
Corporate Contributors    0 non-null float64
Former owner              1 non-null object
Engraver                  0 non-null float64
Issuance type             8287 non-null object
Flickr URL                8287 non-null object
Shelfmarks                8287 non-null object
dtypes: float64(3), int64(1), object(11)
memory usage: 971.3+ KB
```

3 Qu'est ce que vous remarquez?

Entrée [25]:
```
# Rédiger votre réponse
```

4 Dropping unnecessary columns

Nous allons restreindre l'étude à seulement 7 colonnes pour cela nous allons supprimer 7 colonnes que nous jugeons unitiles por le moment :

```
to_drop = ['Edition Statement',
           'Corporate Author',
           'Corporate Contributors',
           'Former owner',
           'Engraver',
           'Contributors',
           'Issuance type',
           'Shelfmarks']
```

Entrée [4]:

```
to_drop = ['Edition Statement',
           'Corporate Author',
           'Corporate Contributors',
           'Former owner',
           'Engraver',
           'Contributors',
           'Issuance type',
           'Shelfmarks']

#Eliminer les colonnes dans la liste de to_drop du dataframe df


#Afficherle résultat

```

Out[4]:

	Identifier	Place of Publication	Date of Publication	Publisher	Title	Author	Flickr URL
0	206	London	1879 [1878]	S. Tinsley & Co.	Walter Forbes. [A novel.] By A. A	A. A.	http://www.flickr.com/photos/britishlibrary/ta...
1	216	London; Virtue & Yorston	1868	Virtue & Co.	All for Greed. [A novel. The dedication signed...	A., A. A.	http://www.flickr.com/photos/britishlibrary/ta...
2	218	London	1869	Bradbury, Evans & Co.	Love the Avenger. By the author of “All for Gr...	A., A. A.	http://www.flickr.com/photos/britishlibrary/ta...
3	472	London	1851	James Darling	Welsh Sketches, chiefly ecclesiastical, to the...	A., E. S.	http://www.flickr.com/photos/britishlibrary/ta...
4	480	London	1857	Wertheim & Macintosh	[The World in which I live, and my place in it...	A., E. S.	http://www.flickr.com/photos/britishlibrary/ta...

5 Setting the index of the dataset

Changer l'index pourqu'ille prenne la valeur de l'identifiant `Identifier`

Entrée [5]:

```
###Changerl'index


###Afficher le résultat

```

Out[5]:

Identifier	Place of Publication	Date of Publication	Publisher	Title	Author	Flickr URL
206	London	1879 [1878]	S. Tinsley & Co.	Walter Forbes. [A novel.] By A. A	A. A.	http://www.flickr.com/photos/britishlibrary/ta...
216	London; Virtue & Yorston	1868	Virtue & Co.	All for Greed. [A novel. The dedication signed...	A., A. A.	http://www.flickr.com/photos/britishlibrary/ta...
218	London	1869	Bradbury, Evans & Co.	Love the Avenger. By the author of “All for Gr...	A., A. A.	http://www.flickr.com/photos/britishlibrary/ta...
472	London	1851	James Darling	Welsh Sketches, chiefly ecclesiastical, to the...	A., E. S.	http://www.flickr.com/photos/britishlibrary/ta...
480	London	1857	Wertheim & Macintosh	[The World in which I live, and my place in it...	A., E. S.	http://www.flickr.com/photos/britishlibrary/ta...

6 Donner un appérçu sur la colonne `Date of Publication`

Entrée [6]:

```
### Afficher  les 25 premières valeurs de Date of Publication

```

Out[6]:

```
Identifier
206            1879 [1878]
216                   1868
218                   1869
472                   1851
480                   1857
481                   1875
519                   1872
667                    NaN
874                   1676
1143                  1679
1280                  1802
1808                  1859
1905                  1888
1929           1839, 38-54
2836                  1897
2854                  1865
2956               1860-63
2957                  1873
3017                  1866
3131                  1899
4598                  1814
4884                  1820
4976                  1800
5382    1847, 48 [1846-48]
5385               [1897?]
Name: Date of Publication, dtype: object
```

> Remarque : Observer la date de publication qui correspond à l'index 206. Ca nécessite un nettoyage et une correction : il faut éliminer certains symboles

7 Cleaning columns using the `.apply` function

Ecrire une fonction qui permet de nettoyer la colonne des dates

Entrée [7]:

```
unwanted_characters = ['[', ',', '-']
# Completer la fonction suivante
def clean_dates(item):
    dop= str(item.loc['Date of Publication'])

    if dop == 'nan' or dop[0] == '[':
        pass

    for character in unwanted_characters:
        if character in dop:
            pass
            pass

    return dop

df['Date of Publication'] = df.apply(clean_dates, axis = 1)
```

Entrée [8]:

```
df.head()
```

Out[8]:

Identifier	Place of Publication	Date of Publication	Publisher	Title	Author	Flickr URL
206	London	1879	S. Tinsley & Co.	Walter Forbes. [A novel.] By A. A	A. A.	http://www.flickr.com/photos/britishlibrary/ta...
216	London; Virtue & Yorston	1868	Virtue & Co.	All for Greed. [A novel. The dedication signed...	A., A. A.	http://www.flickr.com/photos/britishlibrary/ta...
218	London	1869	Bradbury, Evans & Co.	Love the Avenger. By the author of “All for Gr...	A., A. A.	http://www.flickr.com/photos/britishlibrary/ta...
472	London	1851	James Darling	Welsh Sketches, chiefly ecclesiastical, to the...	A., E. S.	http://www.flickr.com/photos/britishlibrary/ta...
480	London	1857	Wertheim & Macintosh	[The World in which I live, and my place in it...	A., E. S.	http://www.flickr.com/photos/britishlibrary/ta...

Entrée [9]:

```
#Voici une autre méthode
#exécuter pour voir le résultat
unwanted_characters = ['[', ',', '-']

def clean_dates(dop):
    dop = str(dop)
    if dop.startswith('[') or dop == 'nan':
        return 'NaN'
    for character in unwanted_characters:
        if character in dop:
            character_index = dop.find(character)
            dop = dop[:character_index]
    return dop

df['Date of Publication'] = df['Date of Publication'].apply(clean_dates)
df.head(10)
```

Out[9]:

Identifier	Place of Publication	Date of Publication	Publisher	Title	Author	Flickr URL
206	London	1879	S. Tinsley & Co.	Walter Forbes. [A novel.] By A. A	A. A.	http://www.flickr.com/photos/britishlibrary/ta...
216	London; Virtue & Yorston	1868	Virtue & Co.	All for Greed. [A novel. The dedication signed...	A., A. A.	http://www.flickr.com/photos/britishlibrary/ta...
218	London	1869	Bradbury, Evans & Co.	Love the Avenger. By the author of “All for Gr...	A., A. A.	http://www.flickr.com/photos/britishlibrary/ta...
472	London	1851	James Darling	Welsh Sketches, chiefly ecclesiastical, to the...	A., E. S.	http://www.flickr.com/photos/britishlibrary/ta...
480	London	1857	Wertheim & Macintosh	[The World in which I live, and my place in it...	A., E. S.	http://www.flickr.com/photos/britishlibrary/ta...
481	London	1875	William Macintosh	[The World in which I live, and my place in it...	A., E. S.	http://www.flickr.com/photos/britishlibrary/ta...
519	London	1872	The Author	Lagonells. By the author of Darmayne (F. E. A....	A., F. E.	http://www.flickr.com/photos/britishlibrary/ta...
667	pp. 40. G. Bryan & Co: Oxford, 1898	NaN	NaN	The Coming of Spring, and other poems. By J. A...	A., J.|A., J.	http://www.flickr.com/photos/britishlibrary/ta...
874	London]	1676	NaN	A Warning to the inhabitants of England, and L...	Rema˙.	http://www.flickr.com/photos/britishlibrary/ta...
1143	London	1679	NaN	A Satyr against Vertue. (A poem: supposed to b...	A., T.	http://www.flickr.com/photos/britishlibrary/ta...

8 Observer la colonne `Author`

Completer la fonction `clean_author_names` qui permet de corriger les valeurs de cet colonne

Entrée [10]:

```python
# compléter la fonction suivante ( à la place de pass)
def clean_author_names(author):

    author = str(author)

    if author == 'nan':
        return 'NaN'

    author = author.split(',')

    if len(author) == 1:
        name = filter(lambda x: x.isalpha(), author[0])
        return reduce(lambda x, y: x + y, name)

    last_name, first_name = author[0], author[1]

    first_name = first_name[:first_name.find('-')] if '-' in first_name else first_name

    if first_name.endswith(('.', '.|')):
        pass

        if len(parts) > 1:
            pass
            pass
            pass
        else:
            pass

    last_name = last_name.capitalize()

    return f'{first_name} {last_name}'


df['Author'] = df['Author'].apply(clean_author_names)
```

Résultat attendu

Identifier	Place of Publication	Date of Publication	Publisher	Title	Author	Flickr URL
206	London	1879	S. Tinsley & Co.	Walter Forbes. [A novel.] By A. A	AA	http://www.flickr.com/photos/britishlibrary/ta...
216	London; Virtue & Yorston	1868	Virtue & Co.	All for Greed. [A novel. The dedication signed...	A. A A.	http://www.flickr.com/photos/britishlibrary/ta...
218	London	1869	Bradbury, Evans & Co.	Love the Avenger. By the author of "All for Gr...	A. A A.	http://www.flickr.com/photos/britishlibrary/ta...
472	London	1851	James Darling	Welsh Sketches, chiefly ecclesiastical, to the...	E. S A.	http://www.flickr.com/photos/britishlibrary/ta...
480	London	1857	Wertheim & Macintosh	[The World in which I live, and my place in it...	E. S A.	http://www.flickr.com/photos/britishlibrary/ta...

Entrée [11]:

```python
df.head()
```

Out[11]:

Identifier	Place of Publication	Date of Publication	Publisher	Title	Author	Flickr URL
206	London	1879	S. Tinsley & Co.	Walter Forbes. [A novel.] By A. A	AA	http://www.flickr.com/photos/britishlibrary/ta...
216	London; Virtue & Yorston	1868	Virtue & Co.	All for Greed. [A novel. The dedication signed...	A. A A.	http://www.flickr.com/photos/britishlibrary/ta...
218	London	1869	Bradbury, Evans & Co.	Love the Avenger. By the author of "All for Gr...	A. A A.	http://www.flickr.com/photos/britishlibrary/ta...
472	London	1851	James Darling	Welsh Sketches, chiefly ecclesiastical, to the...	E. S A.	http://www.flickr.com/photos/britishlibrary/ta...
480	London	1857	Wertheim & Macintosh	[The World in which I live, and my place in it...	E. S A.	http://www.flickr.com/photos/britishlibrary/ta...

Observer la colonne `title`

9 Completer la fonction `clean_title` qui permet de corriger les valeurs de cet colonne

Résultat attendu

Identifier	Place of Publication	Date of Publication	Publisher	Title	Author	Flickr URL
206	London	1879	S. Tinsley & Co.	Walter Forbes	AA	http://www.flickr.com/photos/britishlibrary/ta...
216	London; Virtue & Yorston	1868	Virtue & Co.	All For Greed	A. A A.	http://www.flickr.com/photos/britishlibrary/ta...
218	London	1869	Bradbury, Evans & Co.	Love The Avenger	A. A A.	http://www.flickr.com/photos/britishlibrary/ta...
472	London	1851	James Darling	Welsh Sketches, Chiefly Ecclesiastical, To The...	E. S A.	http://www.flickr.com/photos/britishlibrary/ta...
480	London	1857	Wertheim & Macintosh	The World In Which I Live, And My Place In It	E. S A.	http://www.flickr.com/photos/britishlibrary/ta...

Entrée [12]:

```
## complétrer la fonction suivante ( à la place de pass)

def clean_title(title):

    if title == 'nan':
        return 'NaN'

    if title[0] == '[':
        pass

    if 'by' in title:
        pass
    elif 'By' in title:
        pass

    if '[' in title:
        pass

    title = title[:-2]

    title = list(map(str.capitalize, title.split()))
    return ' '.join(title)

df['Title'] = df['Title'].apply(clean_title)
df.head()
```

Out[12]:

Identifier	Place of Publication	Date of Publication	Publisher	Title	Author	Flickr URL
206	London	1879	S. Tinsley & Co.	Walter Forbes	AA	http://www.flickr.com/photos/britishlibrary/ta...
216	London; Virtue & Yorston	1868	Virtue & Co.	All For Greed	A. A A.	http://www.flickr.com/photos/britishlibrary/ta...
218	London	1869	Bradbury, Evans & Co.	Love The Avenger	A. A A.	http://www.flickr.com/photos/britishlibrary/ta...
472	London	1851	James Darling	Welsh Sketches, Chiefly Ecclesiastical, To The...	E. S A.	http://www.flickr.com/photos/britishlibrary/ta...
480	London	1857	Wertheim & Macintosh	The World In Which I Live, And My Place In It	E. S A.	http://www.flickr.com/photos/britishlibrary/ta...

PART II

1 Cleaning entire dataset

On considère le fichier suivant 'university_towns.txt'

Entrée [11]:

```
!head Datasets/university_towns.txt
```

```
Alabama[edit]
Auburn (Auburn University)[1]
Florence (University of North Alabama)
Jacksonville (Jacksonville State University)[2]
Livingston (University of West Alabama)[2]
Montevallo (University of Montevallo)[2]
Troy (Troy University)[2]
Tuscaloosa (University of Alabama, Stillman College, Shelton State)[3][4]
Tuskegee (Tuskegee University)[5]
Alaska[edit]
```

Entrée [13]:

```
with open('Datasets/university_towns.txt', 'r') as text:
    textfile = text.read()
    print(textfile)
```

```
Riverside (University of California, Riverside, California Baptist University, La Sierra University)
Sacramento (California State University, Sacramento)
University District, San Bernardino (California State University, San Bernardino, American Sports University)
San Diego (University of California, San Diego, San Diego State University)
San Luis Obispo (California Polytechnic State University)[2]
Santa Barbara (Fielding Graduate University, Santa Barbara City College, University of California, Santa Barbara, Westmont C
ollege)[2]
Santa Cruz (University of California, Santa Cruz)[2]
Turlock (California State University, Stanislaus)
Westwood, Los Angeles (University of California, Los Angeles)[2]
Whittier (Whittier CollegeRio Hondo College)
Colorado[edit]
Alamosa (Adams State College)[2]
Boulder (University of Colorado at Boulder)[12]
Durango (Fort Lewis College)[2]
Fort Collins (Colorado State University)[13]
Golden (Colorado School of Mines)
Grand Junction (Colorado Mesa University)
Greeley (University of Northern Colorado)
Gunnison (Western State College)[2]
```

2 Créer un pandas data frame à partir d'un fichier txt

On veut créer un DataFrame nommé `towns_df` à partir du fichier 'university_towns.txt'

Nous voyons que nous avons des noms d'états périodiques suivis des villes universitaires de cet état: StateA TownA1 TownA2 StateB TownB1 TownB2 .... Si nous regardons la façon dont les noms d'états sont écrits dans le fichier, nous verrons qu'ils ont tous la sous-chaîne «[edit]» en eux.

Nous pouvons tirer parti de ce modèle en créant une liste de tuples (état, ville) et en enveloppant cette liste dans un DataFrame:

Entrée [14]:

```
university_towns = []
with open('Datasets/university_towns.txt') as file:
    for line in file:
        if '[edit]' in line:
             # se rappeler de ce `state` jusqu'à ce qu'on trouve le suivant
            pass
        else:
             # Sinon, nous avons une ville; garder `state`
            pass

university_towns[:5]
```

Out[14]:

```
[('Alabama[edit]\n', 'Auburn (Auburn University)[1]\n'),
 ('Alabama[edit]\n', 'Florence (University of North Alabama)\n'),
 ('Alabama[edit]\n', 'Jacksonville (Jacksonville State University)[2]\n'),
 ('Alabama[edit]\n', 'Livingston (University of West Alabama)[2]\n'),
 ('Alabama[edit]\n', 'Montevallo (University of Montevallo)[2]\n')]
```

Entrée [15]:

```
towns_df = pd.DataFrame(university_towns,
                        columns=['State', 'RegionName'])
towns_df.head()
```

Out[15]:

	State	RegionName
0	Alabama[edit]\n	Auburn (Auburn University)[1]\n
1	Alabama[edit]\n	Florence (University of North Alabama)\n
2	Alabama[edit]\n	Jacksonville (Jacksonville State University)[2]\n
3	Alabama[edit]\n	Livingston (University of West Alabama)[2]\n
4	Alabama[edit]\n	Montevallo (University of Montevallo)[2]\n

3 Compléter la fonction `clean_up` pour nettoyer les colonnes de `towns_df`

Entrée [16]:

```
## complétrer la fonction suivante ( à la place de pass)
def clean_up(item):
    if '(' in item:
        pass
        return item[:item.find('(') - 1]

    if '[' in item:
        return item[:item.find('[')]


towns_df =  towns_df.applymap(clean_up)
towns_df
```

Out[16]:

	State	RegionName
0	Alabama	Auburn
1	Alabama	Florence
2	Alabama	Jacksonville
3	Alabama	Livingston
4	Alabama	Montevallo
...	...	...
512	Wisconsin	River Falls
513	Wisconsin	Stevens Point
514	Wisconsin	Waukesha
515	Wisconsin	Whitewater
516	Wyoming	Laramie

517 rows × 2 columns

PART III (OPTIONNELLE)

1 Renaming columns and skipping rows

1.1 Convertir le fichier `olympics.csv` en dataframe pandas `olympics_df`

Entrée [17]:

```
# exporter 'olympics.csv' sous forme de data frame nommé olympics_df


# Afficher les 5 premières lignes

```

Out[17]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	NaN	? Summer	01 !	02 !	03 !	Total	? Winter	01 !	02 !	03 !	Total	? Games	01 !	02 !	03 !	Combined total
1	Afghanistan (AFG)	13	0	0	2	2	0	0	0	0	0	13	0	0	2	2
2	Algeria (ALG)	12	5	2	8	15	3	0	0	0	0	15	5	2	8	15
3	Argentina (ARG)	23	18	24	28	70	18	0	0	0	0	41	18	24	28	70
4	Armenia (ARM)	5	1	2	9	12	6	0	0	0	0	11	1	2	9	12

1.2 Supprimer la première ligne

Entrée [21]:

```
# exporter 'olympics.csv' sous forme de data frame nommé olympics_df en supprimant la première ligne

olympics_df = pd.read_csv('./Datasets/olympics.csv', skiprows = 1, header = 0)

# Afficher les 5 premières lignes
olympics_df.head()
```

Out[21]:

	Unnamed: 0	? Summer	01 !	02 !	03 !	Total	? Winter	01 !.1	02 !.1	03 !.1	Total.1	? Games	01 !.2	02 !.2	03 !.2	Combined total
0	Afghanistan (AFG)	13	0	0	2	2	0	0	0	0	0	13	0	0	2	2
1	Algeria (ALG)	12	5	2	8	15	3	0	0	0	0	15	5	2	8	15
2	Argentina (ARG)	23	18	24	28	70	18	0	0	0	0	41	18	24	28	70
3	Armenia (ARM)	5	1	2	9	12	6	0	0	0	0	11	1	2	9	12
4	Australasia (ANZ) [ANZ]	2	3	4	5	12	0	0	0	0	0	2	3	4	5	12

1.3 Changer le nom des colonnes en utilisant le dictionnaire suivant :

```
new_names =  {'Unnamed: 0': 'Country',
              '? Summer': 'Summer Olympics',
              '01 !': 'Gold',
              '02 !': 'Silver',
              '03 !': 'Bronze',
              '? Winter': 'Winter Olympics',
              '01 !.1': 'Gold.1',
              '02 !.1': 'Silver.1',
              '03 !.1': 'Bronze.1',
              '? Games': '# Games',
              '01 !.2': 'Gold.2',
              '02 !.2': 'Silver.2',
              '03 !.2': 'Bronze.2'}
```

Entrée [10]:

```
# Changer le nom des colonnes en utilisant le dictionnaire new_names









# Afficher les 5 premières lignes
```

Entrée [23]:

```
olympics_df.head()
```

Out[23]:

	Country	Summer Olympics	Gold	Silver	Bronze	Total	Winter Olympics	Gold.1	Silver.1	Bronze.1	Total.1	# Games	Gold.2	Silver.2	Bronze.2	Combined total
0	Afghanistan (AFG)	13	0	0	2	2	0	0	0	0	0	13	0	0	2	2
1	Algeria (ALG)	12	5	2	8	15	3	0	0	0	0	15	5	2	8	15
2	Argentina (ARG)	23	18	24	28	70	18	0	0	0	0	41	18	24	28	70
3	Armenia (ARM)	5	1	2	9	12	6	0	0	0	0	11	1	2	9	12
4	Australasia (ANZ) [ANZ]	2	3	4	5	12	0	0	0	0	0	2	3	4	5	12

Entrée [ ]:

```

```