

2.3

Importer des données depuis un fichier csv avec Python

NSI TLE - JB DUTHOIT

2.3.1 Méthode 1

```
import csv
pays = []
with open("pays.csv", "r") as fichier:
    table = csv.reader(fichier, delimiter = ';')
    for ligne in table:
        pays.append(ligne)
```

On a ainsi une liste de listes

- Quel est le nombre de pays présents dans cette liste ?
- Afficher le premier élément de la liste pays.
- Et afficher le 110 ème

```
>>> pays
[['countryCode', 'countryName', 'area', 'population', 'continent', 'currencyCode', 'curren',
'capital'], ['AD', 'Andorra', '468', '84000', 'EU', 'EUR', 'Euro', 'Andorra la Vella'], [
ted Arab Emirates', '82880', '4975593', 'AS', 'AED', 'Dirham', 'Abu Dhabi'], ['AF', 'Afgha
'647500', '29121286', 'AS', 'AFN', 'Afghani', 'Kabul'], ['AG', 'Antigua and Barbuda', '44:
```

Cette solution présente des inconvénients.

- La 1ère ligne, celle des attributs, a été chargée comme une ligne de données.
- Toutes les données ont été converties en chaînes de caractères même les entiers.
- Plus gênant, le lien entre les valeurs du tableau pays[1] et le nom des enregistrements, contenus dans pays[0], n'est pas direct.
- Nous allons donc utiliser une seconde méthode !

2.3.2 Méthode 2

```
1 import csv
2 pays = []
3 with open("pays.csv", "r") as fichier:
4     table = csv.DictReader(fichier, delimiter = ';')
5     for ligne in table:
6         pays.append(ligne)
```

```
>>> pays
[OrderedDict([('countryCode', 'AD'), ('countryName', 'Andorra'), ('area', '468'), ('population', '84000'), ('continent', 'EU'), ('currencyCode', 'EUR'), ('currencyName', 'Euro'), ('capital', 'Andorra la Vella')]), OrderedDict([('countryCode', 'AE'), ('countryName', 'United Arab Emirates'), ('area', '82880'), ('population', '4975593'), ('continent', 'AS'), ('currencyCode', 'AED'), ('currencyName', 'Dirham'), ('capital', 'Abu Dhabi')]), OrderedDict([('countryCode', 'AF'), ('countryName', 'Afghanistan'), ('area', '647500'), ('population', '29121286'), ('continent', 'AS'), ('currencyCode', 'AFN'), ('currencyName', 'Afghani'), ('capital', 'Kabul')]), OrderedDict([('countryCode', 'AG'), ('countryName', 'Antigua and Barbuda'), ('area', '443'), ('population', '86754'), ('continent', 'NA'), ('currencyCode', 'XCD'), ('currencyName', 'Dollar'), ('capital', 'St. John's')]), OrderedDict([('countryCode', 'AI'), ('countryName', 'Anguilla'), ('area', '102'), ('population', '13394'), ('continent', 'NA'), ('currencyCode', 'XCD'), ('currencyName', 'Dollar'), ('capital', 'The Valley')])]
```

Examinons les avantages de cette solution :

- La 1ère ligne, celle des attributs, n'a pas été chargée comme une ligne de données.
- Le lien entre les valeurs du tableau pays[1] et le nom des enregistrements, contenus dans pays[0], est direct.
- Quel est la longueur de la liste ? Est-ce en accord avec le nombre de pays ? Expliquer .

Mais :

- toutes les données restent converties en chaînes de caractères même les entiers.
- Le format d'affichage peut déconcerter

☛ Nous allons donc utiliser une autre amélioration !

2.3.3 Méthode 3

```
import csv
pays = []
with open("pays.csv", "r") as fichier:
    table = csv.DictReader(fichier, delimiter = ';')
    for ligne in table:
        pays.append(dict(ligne))
```

```
>>> pays
[{'countryCode': 'AD', 'countryName': 'Andorra', 'area': '468', 'population': '84000', 'continent': 'EU', 'currencyCode': 'EUR', 'currencyName': 'Euro', 'capital': 'Andorra la Vella'}, {'countryCode': 'AE', 'countryName': 'United Arab Emirates', 'area': '82880', 'population': '4975593', 'continent': 'AS', 'currencyCode': 'AED', 'currencyName': 'Dirham', 'capital': 'Abu Dhabi'}, {'countryCode': 'AF', 'countryName': 'Afghanistan', 'area': '647500', 'population': '29121286', 'continent': 'AS', 'currencyCode': 'AFN', 'currencyName': 'Afghani', 'capital': 'Kabul'}, {'countryCode': 'AG', 'countryName': 'Antigua and Barbuda', 'area': '443', 'population': '86754', 'continent': 'NA', 'currencyCode': 'XCD', 'currencyName': 'Dollar', 'capital': 'St. John's'}, {'countryCode': 'AI', 'countryName': 'Anguilla', 'area': '102', 'population': '13394', 'continent': 'NA', 'currencyCode': 'XCD', 'currencyName': 'Dollar', 'capital': 'The Valley'}]
```

Une liste de dico, c'est beaucoup plus lisible !



Exercice 2.5

| Ajouter quelques lignes au code précédent afin de convertir les str en entiers ou en flottants