

Après avoir vu différents types de données, nous allons maintenant, à l'aide du langage de programmation Python, apprendre à effectuer des traitements sur ces données.

Pour traiter des données, nous allons utiliser la bibliothèque Python Pandas. Une bibliothèque Python permet de rajouter des fonctionnalités par rapport au langage de base. La bibliothèque Pandas est donc très utilisée pour tout ce qui touche au traitement des données.

Pour nos premiers pas avec Pandas, nous allons utiliser des données très simples au format CSV : ces données sont contenues dans le fichier `ident_virgule.csv`.

1^{re} partie : un premier code

- Télécharger le fichier `ident_virgule.csv` et placer dans un répertoire que vous nommerez `pandas`.
- Ouvrez le logiciel Spyder puis dans la partie "Editeur de texte" de Spyder, saisissez le code Python suivant :

```
1 import pandas
2 iden=pandas.read_csv("ident_virgule.csv",encoding
    = "ISO-8859-1")
```

Une fois le code saisi, enregistrer le fichier contenant ce code dans le même répertoire que le fichier `"ident_virgule.csv"`.

2^e partie : explications du code

Le code ci-dessus est simple dans les explications :

- À la première ligne : importation de la bibliothèque `pandas` pour son utilisation.
- À la deuxième ligne : création d'une variable `"iden"` qui va contenir les données présentes dans le fichier `"ident_virgule.csv"`

3^e partie : une première commande

- Exécuter le programme en cliquant sur le triangle vert puis placez ensuite le curseur de la souris dans la console de Spyder juste à côté d'un `"In [X]"` (avec X égal à 1, 2, 3..., selon les cas, dans l'exemple ci-dessous nous avons `"In [3]"`). Tapez alors `"iden"`

Vous devriez voir apparaître les données contenues dans la variable `"iden"` rangées sous la forme d'un tableau, un peu comme ce que nous obtenions en ouvrant le fichier `"ident_virgule.csv"` avec un tableur.

```
In [2]: iden
Out[2]:
```

	nom	prenom	date_naissance
0	Ainhua	Alan	23/05/2005
1	Otega	Loïc	15/12/2007
2	Barsacq	Lorena	12/06/2008

- Quelle remarque peut-on faire ?

.....

.....

.....

Les nombres présents dans cette colonne sont appelés des index. Chaque ligne du tableau a un index (première ligne : index 0, deuxième ligne index 1...)

Les index commencent à 0 et pas à 1

Les colonnes possèdent également des index, dans notre exemple ces index correspondent au "nom" (index de la première colonne), au "prenom" (index de la deuxième colonne) et à "date_naissance" (index de la troisième colonne)

En résumé : les lignes possèdent des index (0,1,2..), les colonnes possèdent aussi des index ("nom", "prenom",...)

Il est possible de récupérer certaines données du tableau, par exemple, certaines lignes, certaines colonnes ou bien encore des valeurs uniques. Pour cela, il suffit d'utiliser l'instruction "loc" avec les index des lignes et les index des colonnes. Le principe de fonctionnement de "loc" est relativement simple puisque l'on aura une instruction de la forme "loc[index_ligne,index_colonne]"

3^e partie : d'autres commandes

- Testez le programme suivant :

```
1 import pandas
2 iden=pandas.read_csv("ident_virgule.csv")
3 info=iden.loc[2,'nom']
```

Que contient la variable "info" en fin de programme?

- Modifiez le programme précédent pour que la variable info contienne "Ainhoa".

Il est possible de récupérer toutes les colonnes d'une ligne particulière, cette fois en remplaçant la partie "index_colonne" de "loc" par ":"

Vérifiez que la variable "info" contient bien toutes les données de la dernière ligne.

Il est aussi possible de récupérer seulement certaines lignes et certaines colonnes en utilisant la notation suivante : loc[[index_ligne_1,index_ligne_2,...],[index_colonne_1,index_colonne_2,...]]

Testez le programme suivant :

```
1 import pandas
2 iden=pandas.read_csv("ident_virgule.csv")
3 info=iden.loc[[0,1],['nom','date_naissance']]
```

Vérifiez que la variable "info" contient bien un tableau avec uniquement les colonnes "nom" et "date_naissance" de la première ligne (index 0) et de la deuxième ligne (index 1).