



Université de Monastir  
Institut Supérieur d'Informatique et de Mathématiques de Monastir  
Département d'Informatique



# TRAITEMENT ET ANALYSE DE DONNÉES AVEC PANDAS

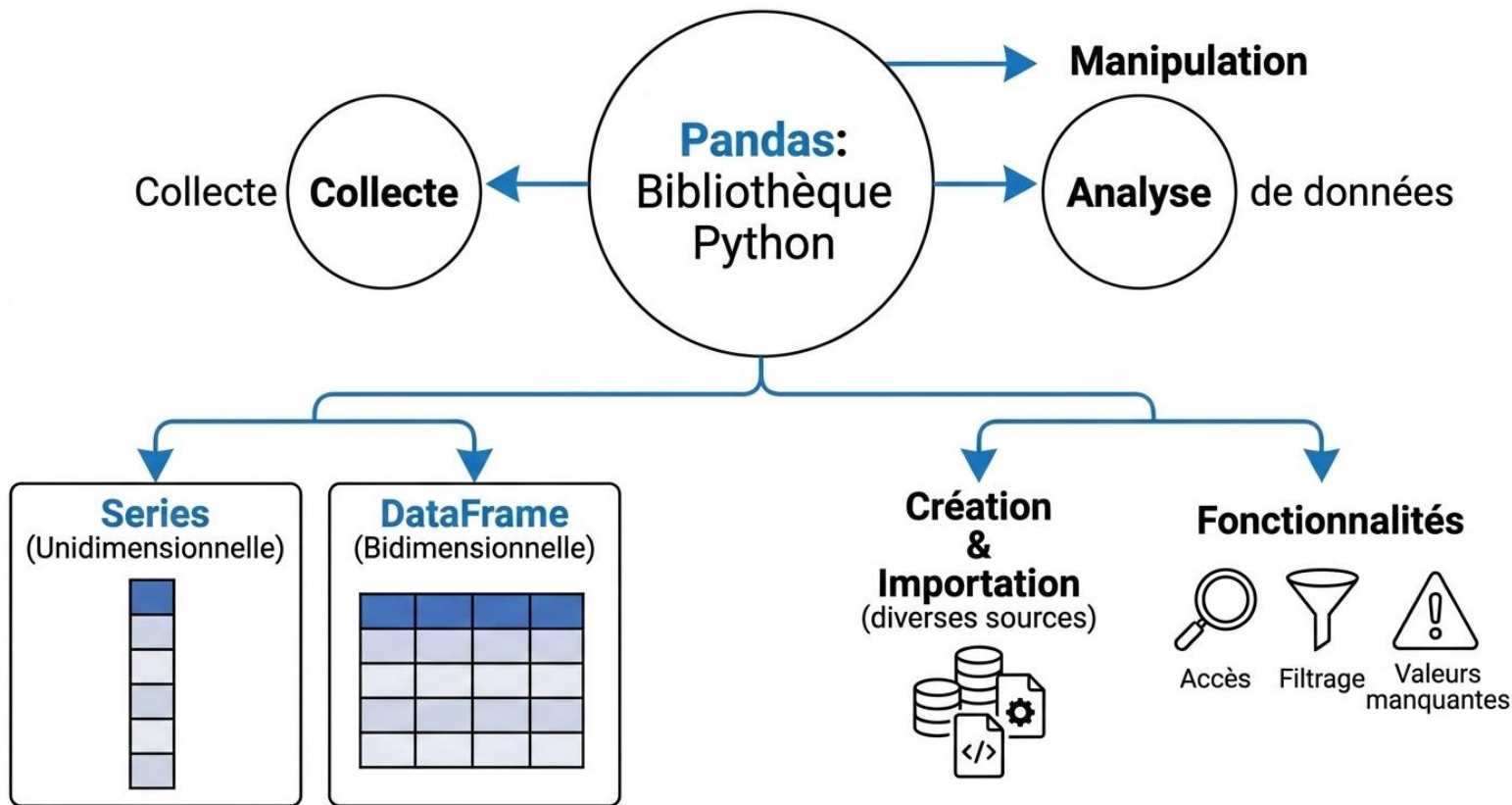


Première Année Préparatoire Intégré  
2025-2026

Séance 02



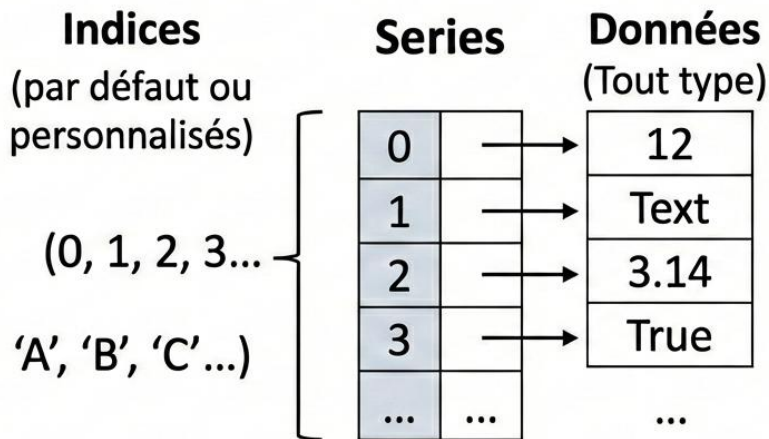
# Rappel (Séance 1)





# Rappel (Séance 1)

## Series: Un Vecteur à 1 Dimension



Series Vide  
(possible)


### Affichage & Accès



Afficher Totalité



Valeur Simple



Valeurs Multiples  
Successives



Valeurs Multiples  
Non Successives



- ⦿ La structure la plus utilisée pour analyser des données.
- ⦿ C'est une matrice avec des lignes et des colonnes, comme une feuille de calcul Excel.
- ⦿ Un **DataFrame** peut stocker en mémoire des données dans des formats différents.
- ⦿ Un **DataFrame** contient une ou plusieurs **Series**.
- ⦿ Un **DataFrame** est mutable.

## 4

# Qu'est ce qu'un objet DataFrame ?

Colonnes  
axe=1

→

Nom Colonne

↓

Autres Colonnes à afficher

↓

	Titre	Nom_Prenom	CP	Ville	...	Age	Nbre_Enfants
0	M.	Ali Nebli	3000	Sfax		51	3
1	Mme	Leila Tounsi	9000	Beja		48	2
2	M.	Walid Sfaxi	NaN	Kairouan		33	1
3	Mlle	Samia Soussi	7000	Bizerte		25	0
4	Mme	Amel Gafsi	3200	Tataouine		45	3
5	NaN	Mehdi Jandoubi	2200	Tozeur		54	4

Numéro  
d'Indice

→

Indices  
axe=0

↑

Valeurs  
Manquantes

↑

Données  
(Valeurs)

↑

## Comment créer un objet DataFrame ?



```
Id_DataFrame = alias.DataFrame (columns = [Colonne1, ..., ColonneN],  
                                index = [Idx1, ..., IdxM],  
                                data = [ [Val11, ..., Val1N], ..., [ValM1, ..., ValMN] ] )
```



```
df1 = ps.DataFrame(columns = ['Prénom', 'Age', 'Ville'],  
                   index = [1, 2, 3, 4],  
                   data = [ ['Amine', 14, 'Sousse'],  
                           ['Sarrah', 15, 'Gafsa'],  
                           ['Malek', 15, 'Bizerte'],  
                           ['Ahmed', 13, 'Gabes'] ] )
```

```
print(df1)
```

## 4

## Comment créer un objet DataFrame ?



```
Id_DataFrame = alias.DataFrame( { Id_Colonne1 : [Val1, ..., ValN], ...,  
                                  Id_ColonneM : [Val1, ..., ValN] } )
```



```
df2 = ps.DataFrame({'Prénom':['Amine','Sarra','Malek','Ahmed'],  
                    'Age':[14,15,15,13], 'Ville':['Sousse','Gafsa','Bizerte','Gabes']})  
print(df2)
```



	Prénom	Age	Ville
0	Amine	14	Sousse
1	Sarra	15	Gafsa
2	Malek	15	Bizerte
3	Ahmed	13	Gabes

## Comment créer un objet DataFrame ?



```
Id_DataFrame = alias.DataFrame({Id_Colonne1:[Val1, ..., ValN], ...,  
                                Id_ColonneM:[Val1, ..., ValN]}, index=[Idx1, Idx2, ..., IdxN])
```



```
df3 = ps.DataFrame({'Prénom':['Amine','Sarrra','Malek','Ahmed'],  
                    'Age':[14,15,15,13], 'Ville':['Sousse','Gafsa','Bizerte','Gabes']},  
                    index=['Candidat1','Candidat2','Candidat3','Candidat4'])  
print(df3)
```



	Prénom	Age	Ville
Candidat1	Amine	14	Sousse
Candidat2	Sarrra	15	Gafsa
Candidat3	Malek	15	Bizerte
Candidat4	Ahmed	13	Gabes



## Remarque

- Il est possible de créer un DataFrame vide de la façon suivante :



```
import pandas as alias  
Id_DataFrame = alias.DataFrame ()
```



```
df0 = ps.DataFrame()  
print(df0)
```



```
Empty DataFrame  
Columns: []  
Index: []
```

## 4

## Comment importer les données d'un fichiers Excel (.xls, .xlsx) dans un DataFrame ?



```
Id_DataFrame = alias.read_excel ('chemin\nom_fichier.xlsx',  
                                'Nom_Feuille', index_col=N°_Colonne, header=0)
```



- **chemin\nom\_fichier.xlsx** : Le chemin physique du fichier sur le disque dur (L'utilisation du symbole antislash "\" dans le chemin physique peut générer des erreurs. Pour cela, vous pouvez le remplacer par le symbole slash "/").
- **Nom\_Feuille** : Indique le nom de la feuille à importer (en cas d'omission, on importera la feuille1).
- **index\_col = N°\_colonne** : Elle permet d'indiquer le numéro de la colonne par lequel on débutera l'importation des données (en cas d'omission, on commencera par la colonne 0).
- **header = 0** : Ce paramètre indique que les libellés des colonnes se trouvent sur la ligne 0. Si les données ne contiennent pas des libellés, on mettra **header=None** (En cas d'omission, c'est header=0).



```
fifa = ps.read_excel ('c:/Data/fifa2018.xlsx','Equipes')  
print(fifa)
```

## 4

## Comment importer les données d'un fichier texte (.csv, .txt) dans un DataFrame ?



Un fichier **CSV** (Comma-Separated Values) est un fichier texte simple utilisé pour stocker des données tabulaires (lignes et colonnes). Chaque ligne du fichier correspond à



une ligne de tableau, et chaque cellule est séparée par un délimiteur, généralement une virgule, un point-virgule ou une tabulation. C'est un format universel pour échanger des données entre applications.

```
Employee,Position,Department,Age,City of Residence,Date of Hire,Annual Income,  
Alice Johnson,Software Engineer,IT,28,New York,05-15-20,$90000.00,  
Bob Davis,Marketing Specialist,Marketing,35,Los Angeles,09-20-18,$75000.00,  
Charlie Smith,Financial Analyst,Finance,30,Chicago,03-10-19,$85000.00,  
Diana Rodriguez,HR Manager,Human Resources,40,Miami,07-05-17,$95000.00,  
Eva Williams,Project Manager,Operations,32,San Francisco,01-12-16,$110000.00,
```

## 4

## Comment importer les données d'un fichier texte (.csv, .txt) dans un DataFrame ?



```
Id_DataFrame = alias.read_csv('chemin\nom_fichier.extension',  
                               sep='séparateur', index_col=N°_colonne, header=0)
```

- **sep = 'séparateur'** : Elle permet d'indiquer le séparateur qui sépare les données. Les séparateurs les plus utilisés sont :
  - **'\t'** : Les données seront séparées par une tabulation (séparateur par défaut en cas d'omission).
  - **','** : Les données seront séparées par une virgule.
  - **';'** : Les données seront séparées par un point-virgule.



```
sante = ps.read_csv('c:/Data/mesures.csv', sep=',')  
print(sante)
```

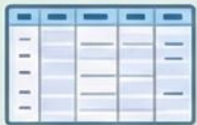




# Comment afficher toutes les données d'un DataFrame ?

Lorsque des ... apparaissent dans un DataFrame, Pandas tronque l'affichage. Voici comment afficher toutes les données :

## 1 Afficher toutes les lignes et colonnes



Supprime la limite sur le nombre de lignes et de colonnes affichées.

✓ L'ensemble du DataFrame sera affiché

```
#import pandas as ps
```

```
ps.set_option("display.max_rows", None)
```

```
ps.set_option("display.max_columns", None)
```

## 2 Afficher tout le contenu des cellules (chaînes longues)



Empêche la troncature du texte à l'intérieur des cellules.

✓ Le contenu textuel ne sera plus tronqué

```
*****
```

```
ps.set_option("display.max_colwidth", None)
```

```
-----  
-----
```

## 3 Revenir aux paramètres par défaut (si besoin)



Rétablit la configuration par défaut de Pandas.

```
ps.reset_option("display.max_rows")
```

```
ps.reset_option("display.max_columns")
```

```
ps.reset_option("display.max_colwidth")
```



**Afficher toutes les données est rarement recommandé** pour les grands volumes car cela peut vite bloquer l'environnement.

## 4

## Comment importer les données d'un fichier JSON (.json) dans un DataFrame ?



Un fichier **JSON** (.json) est un format de fichier texte léger, structuré en paires clé-valeur, utilisé pour le stockage et l'échange de données, notamment entre un serveur et une application web. Dérivé de JavaScript, il est indépendant du langage et facile à lire par l'homme et la machine.

```
{
  "prenom": "Pierre",
  "nom": "Giraud",
  "adresse": {
    "rue": "30 Impasse des Lilas",
    "ville": "Toulon",
    "cp": 83000,
    "pays": "France"
  },
  "mails": [
    "pierre.giraud@edhec.com",
    "pierre@pierre-giraud.com"
  ]
}
```



```
Id_DataFrame = alias.read_json ('chemin\nom_fichier.extension')
```



```
etudiant = ps.read_json('c:/Data/isimm.json')  
print(etudiant)
```



## Activité

1. Créer puis afficher le DataFrame nommé '**cereale**' suivant :

	Nom	Prix
Code1	Blé Dur	87.255
Code2	Blé Tendre	67.732
Code3	Orge	56.500
Code4	Triticale	NaN

```
import pandas as ps
cereale = ps.DataFrame(columns = ['Nom', 'Prix'],
                        index = ['Code1', 'Code2', 'Code3', 'Code4'],
                        data = [ ['Blé Dur', 87.255], ['Blé Tendre', 67.732],
                               ['Orge', 56.500], ['Triticale', None] ] )
print(cereale)
```



## Activité

2. Le fichier "**capture.csv**" contient la capture d'une trame réseau faite avec le logiciel **Wireshark** qui enregistre et analyse les paquets de données transitant sur un réseau local via une carte réseau.

Créer puis afficher un DataFrame nommé '**capt**' à partir de ce fichier csv sachant que le séparateur est un point-virgule.

```
import pandas as ps
capt = ps.read_csv('c:/Data/capture.csv', sep=';')
print(capt)
```