

MCAP

1

Info 41 Panda en python

Pandas: une bibliothèque interact gen
sante permettant la manipulation et
l'analyse de données de manière
simple et intuitive en pb python.
la force de Panda

- structure de données (data frame)
rapide et efficace pour la manipulation
des données.
- open source
- outils liés et écrites dans des
fichiers différents (csv, txt, ...)

import pandas as pd (importation)

- les Series: ^{un tableau} contiennent un tableau étiqueté
uni dimensionnel pouvant contenir tout
type de données (entiers, chaînes de caractères
...)

Ser = pd.Series (data, index=index)

- création d'un objet series grâce à un

dictionnaire

import pandas as pd

notes = { "math": 19, ... }

print (Ser)

print (Ser.index) = affiche l'index de la
série

print (Ser.away) = "date"

- création d'un objet Series grâce à

un ndarray

Si data est un ndarray, l'index passé
en paramètre doit être de la même
longueur que data

import numpy as np

import pandas as pd

Ser = pd.Series (np.array ([6700, 6780]),

index = ("France", "Tunisie"),

Ser = France: 6700, Tunisie: 6780 (Pandas)

Création d'un objet grâce à un

Valeur scalaire

L'index doit être fourni la valeur de

captée par correspondance à la longueur

de l'index.

Ser = pd.Series (5, index = ["a", "b", ...])

Ser

a 5

b 5

- l'attribut name d'un objet Series

nommer grâce à l'attribut (name)

import pandas as pd

import numpy as np

data = np.array ([6700, 6780])

Ser = pd.Series (data, index = ["France",

"Allemagne"], name = "Série nbre

d'habitants")

Ser

Résultat:

France 6700

Allemagne 6780

- pour renommer on utilise

rename ()

Ser2 = Ser.rename ("Nb habitants")

tp Ser et Ser2 sont 2 objets différents

print (Ser2)

- on peut accéder à un objet grâce à des étiquettes et des indices de position comme exemple:-

```

import pandas as pd
notes = { "Maths": 19, "FR": 12, "SN": 18, "Biologie": 17 }
ser = pd.Series(notes)

```

```

print(ser["Maths"])
print(ser[1])

```

le résultat:

```

19
12
print(ser[2:])
print(1+1+1+1)
print(ser["FR": "Biologie"])

```

le résultat:

```

18
1 2 3 4
FR SN 1 2

```

- utilisation de loc() et iloc()

```

import pandas as pd
flours = pd.Series(["rose", "tournesol", "muguet"])
index = [1, 2, 3, 4, 5])

```

- loc() fait référence l'étiquette
- iloc() fait référence à l'indice de position

exemple:

```

print(flours.loc[1])
print(flours.iloc[1])

```

Résultat:

```

rose
tournesol

```

(3)

chap 3 TKinter python:

avantages: sa portabilité sur les OS les plus utilisés par le grand public.

Hello world:

```

from tkinter import *
fenetre = Tk()
label = Label(fenetre, text="Hello world")
label.pack()

```



- Les Boutons: proposer une action à l'utilisateur.

```

button = Button(fenetre, text="Fermer",
command=fenetre.quit)
button.pack()

```



Objets labels: pour écrire un texte, comme input.

```

label = Label(fenetre, text="Abdoul", bg="yellow")
label.pack()

```



• Entrée / Input

• Les listes: regroupent une valeur sélectionnée par l'utilisateur

```

liste = listbox(fenetre)
liste.insert(1, "python")
liste.insert(2, "phy")
liste.insert(3, "css")
liste.pack()

```



(2)

3) Dataframe: structure de données la plus puissante de pandas

(Dataframe) est une structure de données qui organise les données en lignes et en colonnes, (bidimensionnelle), comme une feuille de calcul ou table SQL de types différents -

- un dict de ndarrays 1D, liste, ...
- un numpy.ndarray bidimensionnelle
- un ndarray structuré
- une série

La méthode de base pour créer une Dataframe est:

```
df = pd.DataFrame(data, index=index, columns=columns)
```

création d'une Dataframe à partir d'un dictionnaire de séries

```
import pandas as pd
ndg = {"Natho": pd.Series([18, 20], index=[
    "sylvie", "Gilles"]), "Sciens": pd.Series([15, 7], index=["sylvie", "Gilles"])
df = pd.DataFrame(ndg)
df
```

résultat

	Natho	Sciens
Gilles	20	7
Sylvie	18	15

Les étiquettes des lignes et colonnes sont accessibles respectivement en accédant aux attributs index et columns de la façon suivante:

```
import pandas as pd
ndg = pd.DataFrame(ndg)
```

```
print(df.index)
print(df.columns)
Index(['Gilles', 'sylvie'])
Index(['Natho', 'Sciens'])
```

- création d'un Dataframe à partir d'un dictionnaire de ndarray ou de liste

Les ndarrays doivent avoir la même longueur. Le résultat sera rang(n) où n: long du tableau

exemple:

```
import pandas as pd
data = {"prenom": ["sylvie", "Gilles"],
        "Age": [18, 20]}
df = pd.DataFrame(data)
print(df)
```

Le résultat est =

	Age
0	18
1	20

- sélection de données =

Accès aux données d'un objet Séries
il existe 3 méthodes d'accès à pandas:

- l'opérateur d'indexation `[]`
- méthodes `loc()` et `iloc()`
- l'opérateur d'indexation

Canvas:

pour dessiner on écrit ce que 'on veut'.

ex Canvas

canvas = canvas = (fonctio, width=150,
height=100, background='yellow')

ligne 1 = canvas.create_line(70, 0, 70, 100)

ligne 2 = canvas.create_line(70, 70, 70)

txt = canvas.create_text(70, text='civili')

canvas.pack()



Scale:

permet de récupérer une valeur numérique
via Scale

value = val DoubleVar()

scale = Scale (fonctio, variable=variable)

scale.pack()

39

Frame: séparer des éléments

fonctio [bg] = white

ex frame 1

frame 1 = Frame (fonctio, borderwidth=2, relief=
GROOVE)

frame 1. pack (side=Left, padx=20, pady=20)

ex Ajout de labels

label (Frame 1, text='Frame 1').pack(padx=10,

pady=10)



⑪ chap 1 P00

P00: permet de mieux modéliser la réalité
en conservant des modèles d'objets, dans.

- classe: permet de construire
des objets interactifs entre eux
et avec le monde extérieur.

- objet: créé indépendamment les
uns des autres, grâce à l'encapsulation,
qui permet d'embarquer leurs ppts.