

Entrées/sorties Fichiers

000

Chaine caracteres

Les chaines de caracteres sont des listes

```
mot="ma maison"

print(mot[0])
print(mot[2:6])
print(mot[::-2])

for c in mot:
    print(c)
```

m	a		m	a	i	s	o	n
0	1	2	3	4	5	6	7	8

001

Chaine caracteres

On ne modifie pas une chaine de caractere
(immutable)

```
mot="ma maison"

mot[2]="r"
```

'str' object does not support item assignment

002

Concatenation

+ concatène une chaine avec une autre

```
mot_1="bonjour"
mot_2="a toi"
mot_3="!!"

mot_4=mot_1+mot_2+mot_3
mot_5=mot_1+" "+mot_2+" "+mot_3
mot_6=mot_1+" "+mot_2[2:]+mot_3

print(mot_4)
print(mot_5)
print(mot_6)
```

→ bonjoura toi!!

→ bonjour a toi !!

→ bonjour toi!!

003

Majuscule, minuscule

```
mot="ma maison"

mot2=mot.replace("maison","demeure")
mot3=mot.upper()
mot4=mot[0].upper()+mot[1:].lower()

print(mot2)
print(mot3)
print(mot4)
```

004

Séparation

Split sépare en plusieurs entités une chaîne à l'aide d'un délimiteur

```
chaine="125+784-52.2+845=?"
resultat=chaine.split("+")

print(resultat[0])
print(resultat[1])
print(resultat[2])
```

125+784-52.2+845=?
resultat
[0] 125
[1] 784-52.2
[2] 845=?

005

Utilisation typique: espace

```
phrase="Maitre Corbeau, sur un arbre perche"
mots=phrase.split(" ")

for k,mot in enumerate(mots):
    print("mot",k," :",mot)
```

```
mot 0 : Maitre
mot 1 : Corbeau,
mot 2 : sur
mot 3 : un
mot 4 : arbre
mot 5 : perche
```

006

Application: nom/prenom

```
etudiant="jean dumont"
mots=etudiant.split(" ")

prenom=mots[0]
nom=mots[1]

prenom=prenom[0].upper()+prenom[1:].lower()
nom=nom[0].upper()+nom[1:].lower()

print(nom," ",prenom)
```

Dumont , Jean

007

Liste de mots

```
liste_mots=["maison","chocolat","elephant"]  
  
print(liste_mots[1][2])  
  
for mot in liste_mots:  
    print(mot[-1])
```

008

Application

Construire et afficher une liste de nom/prénoms telle que la liste:

```
etudiant=["jean dumont"  
          ,"FABRICE rene"  
          ,"Antoine GONTRAND"  
          ,"romain SEVERIN"]
```

Soit transformée en:

Dumont	Jean
Gontrand	Antoine
Rene	Fabrice
Severin	Romain

- liste ordonnée dans l'ordre alphabétique du nom
- le nom et le prénom sont séparés
- la première lettre du nom et prénom sont en majuscule (le reste en minuscule)

009

Application

Construire et afficher une liste de nom/prénoms telle que la liste:

```
etudiant=["jean dumont"  
          ,"FABRICE rene"  
          ,"Antoine GONTRAND"  
          ,"romain SEVERIN"]
```

Soit transformée en:

Dumont	Jean
Gontrand	Antoine
Rene	Fabrice
Severin	Romain

- liste ordonnée dans l'ordre alphabétique du nom
- le nom et le prénom sont séparés
- la première lettre du nom et prénom sont en majuscule (le reste en minuscule)

```
nouvelle_liste=[]  
for e in etudiant:  
    prenom,nom=e.split(" ")  
    nom=nom[0].upper()+nom[1:].lower()  
    prenom=prenom[0].upper()+prenom[1:].lower()  
    nouvelle_liste.append([nom,prenom])  
  
nouvelle_liste=sorted(nouvelle_liste)  
print(nouvelle_liste)
```

010

Conversion nombre/mot

```
a=7  
b=12.2
```

```
mot_a=str(a)  
mot_b=str(b)
```

```
mot_c=mot_a+mot_b
```

```
c=float(mot_c)
```

```
d=c+a  
print(d)
```

Conversion en texte

Concaténation de texte

Conversion texte en nombre

Addition de nombre

011

Application:

Extraire les notes de ce texte
Afficher la moyenne correspondante

```
notes="chimie:14.5 physique:9.5 math:8.7"
```

012

Application:

Extraire les notes de ce texte
Afficher la moyenne correspondante

```
notes="chimie:14.5 physique:9.5 math:8.7"
```

```
avg=0.0
matiere=notes.split(" ")
for m in matiere:
    nom,note=m.split(":")
    avg+=float(note)

avg/=len(matiere)
print(avg)
```

013

Expression Régulière

```
import re
nom="Vincent Dupont, Georges Hulk, Jean Francis, Vincent Ran"
m=re.findall('Vincent (\w+)',nom)
print(m)
```

['Dupont' , 'Ran']

Voir doc complète:
<http://docs.python.org/3/library/re.html>

014

Fichiers

015

Lecture d'un fichier



Lecture ligne à ligne

```
fichier=open("fichier.txt")  
  
for ligne in fichier:  
    print(ligne)  
  
fichier.close()
```

016

Traitement d'un texte

```
fichier=open("fichier.txt")  
  
for ligne in fichier:  
    nouvelle_phrase=ligne.replace("Corbeau","Pelican")  
    print(nouvelle_phrase)  
  
fichier.close()
```

Maître Pelican, sur un arbre perché,
Tenait en son bec un fromage.
Maître Renard, par l'odeur alléché,
Lui tint à peu près ce langage :
"Hé ! bonjour, Monsieur du Pelican."
...

017

Ecriture fichier

write/ecrit
(supprime/crée un fichier vierge)

```
fichier=open("mon_fichier.txt","w")  
  
fichier.write("Bonjour fichier \n")  
fichier.write("3+3="+str(3+3))  
  
fichier.close()
```

```
Bonjour fichier  
3+3=6
```

018

Application: Analyse données

Les fichiers disponibles fournissent les températures moyennes de Lyon en février et avril 2011, 2012, et 2013.

Afficher et comparer les courbes de températures

Algorithme:

```
Pour chacun des fichiers  
| Pour chaque ligne à l'exception de la première  
| | Séparer les données suivant le symbol "/"  
| | Lire la 4ème donnée en tant que nombre  
| | Stocker cette valeur dans un vecteur de température  
| Afficher le vecteur de température
```

http://www.meteorologic.net/metar-climato_LFLY.html

019

Application: Analyse données

```
import matplotlib.pyplot as plt

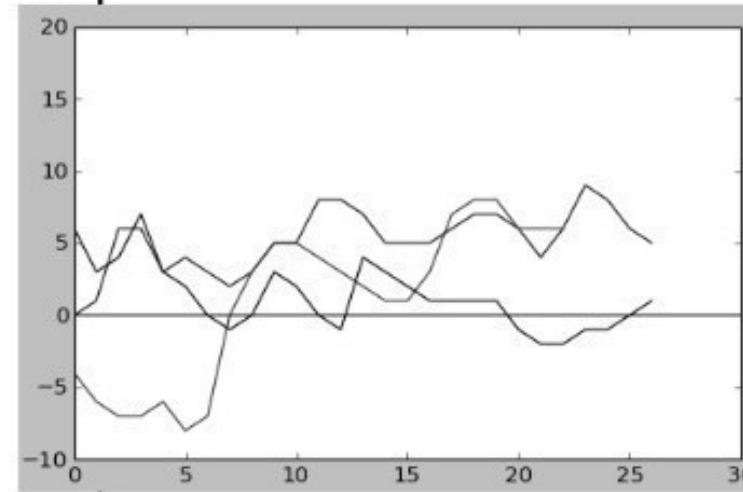
files=[open("../data/fevrier_2011_lyon.txt"),
        open("../data/fevrier_2012_lyon.txt"),
        open("../data/fevrier_2013_lyon.txt")]

temp=[]

for k_year,f in enumerate(files):
    temp.append([])
    for k_line,line in enumerate(f):
        if k_line>0:
            data=line.split("/")
            temp[k_year].append(float(data[3]))

plt.plot(temp[0],"b")
plt.plot(temp[1],"r")
plt.plot(temp[2],"k")
plt.axis([0,30,-10,20])

plt.show()
```



020

Application: Formatage

Soit le fichier suivant:

```
Nom;Prenom;Note Math;Note Physique;Note Chimie
herz;jean;12;4.8;9.9
hill;bejamin;9.5;12;12
amenda;francis;15;14;11
gorgie;andy;12;9.8;8
```

Ecrire le traitement qui viendra écrire le fichier suivant:

```
AMENDA francis 13.33
GORGIE andy 9.93
HERZ jean 8.9
HILL bejamin 11.17
```

021

Application: Formatage

Soit le fichier suivant:

```
Nom;Prenom;Note Math;Note Physique;Note Chimie
herz;jean;12;4.8;9.9
hill;bejamin;9.5;12;12
amenda;francis;15;14;11
gorgie;andy;12;9.8;8
```

```
f=open("note_etudiant.txt")
liste=[]
fout=open("note_formatée.txt","w")
for k_line,line in enumerate(f):
    if k_line>0:
        data=line.split(";")

        nom=data[0]
        prenom=data[1]

        avg=0
        for note in data[2:]:
            avg+=float(note)
        avg/=len(data[2:])
        avg=round(avg,2)

        liste.append([nom.upper(),prenom.lower(),avg])

liste=sorted(liste)
for etudiant in liste:

    fout.write(etudiant[0]+" ")
    fout.write(etudiant[1]+" ")
    fout.write(str(etudiant[2])+"\n")
```

022

Programmes externes

023

Programmes externes

Le module "os" permet de communiquer avec le système d'exploitation

os.system(...) appelle un programme externe

```
import os  
  
os.system("firefox http://prepas.org/")
```

024

Répertoires

```
import os  
  
files=os.listdir(".")  
  
print(files)  
for f in files:  
    print("fichier ",f)
```

Tous les fichiers
du répertoire
dans une liste

025

Traitement de fichiers

Convertir tous les fichiers .png ou .tiff d'un répertoire en .jpg

Utilisez l'outil "convert" en ligne de commande
\$ convert a.png b.jpg

<str>.endswith(".png") → chaîne finit par ".png" ?
os.path.splitext(filename) → sépare nom et extension

026

Traitement de fichiers

Convertir tous les fichiers .png ou .tiff d'un répertoire en .jpg

```
import os  
  
rep="pic/"  
files=os.listdir(rep)  
  
for f in files:  
    if f.endswith(".png") or f.endswith(".tiff"): fini par  
        nom,ext=os.path.splitext(f) extension  
        output=rep+nom+".jpg"  
        input=rep+f  
        os.system("convert "+input+" "+output)
```

027

Parcours répertoires

```
import os

for racine, repertoire, fichiers in os.walk("data/code/"):
    print("dans le repertoire", racine)
    print(" il y a", len(repertoire), "repertoire :", repertoire)
    print(" et", len(fichiers), "fichiers:", fichiers)
```

```
os.path.isdir(directory)
os.path.isfile(filename)
```

fichier/répertoire

028

Application: répertoire

Afficher les chemins de tous les fichiers contenant la chaîne "int main()"

029

Application: répertoire

Afficher les chemins de tous les fichiers contenant la chaîne "int main()"

```
import os

for racine, repertoire, fichiers in os.walk("data/code/"):
    for fichier in fichiers:

        chemin=racine+"/"+fichier
        f=open(chemin, "r")

        for line in f:
            if line.find("int main()")==0:
                print(chemin)

        f.close()
```

030

Récupérer ligne commande

```
import os

file="data/code/fichier1.c"

retour_wc=os.popen("wc "+file).read()
nbr=retour_wc.split()
print(nbr[0], "lignes", nbr[1], "mots")

nbr_commentaire=os.popen("grep \"/\"/\" "+file+" | wc -l").read()

pourcentage=float(nbr_commentaire)/int(nbr[0])
print(100*pourcentage, "% de commentaire")
```

Affiche nombre de lignes / mots
Affiche pourcentage de commentaires

031