

Informatique fondamentale

Flux et outils

R.Gosswiller

1 Les fichiers en Python

2 Manipuler des fichiers

3 Lecture/Ecriture

Les fichiers en Python

Pourquoi des fichiers ?

- Manipulation de données
- Stockage persistant
- Séparer données et traitement

Nommage des fichiers

Le path

Le path d'un fichier est son nom complet

Il est composé du chemin + le nom local du fichier

Exemple

/home/bob/folder/file.txt

C :/ProgramFiles/Ressources/file.txt

Nommage des fichiers

Path relatif/absolu

Path relatif : accéder à un fichier depuis un lieu donné

Path absolu : accéder à un fichier depuis la racine du système

Organisé selon une arborescence

Exemple

C : /home/bob/folder/file.txt

../folder/file.txt depuis /home/bob/test/

Nommage des fichiers

Règles de nommage

Eviter les espaces et caractères spéciaux

Le format n'est qu'une information utile pour l'OS (ou certains logiciels)

Il n'a aucune utilité réelle

Manipuler des fichiers

Répertoire courant

Principe

Par défaut, un programme Python cherche un fichier depuis le répertoire courant

Défini par le lieu d'exécution du programme

La fonction chdir

Principe

Changement du répertoire de référence

Exemple

```
1  from os import chdir
2  chdir("/home/bob/test")
```

Le curseur

Principe

Un fichier est manipulé à partir d'un endroit nommé handle (curseur)
Généralement, le handle est au début du fichier à l'ouverture du fichier.
Le handle n'est pas remis à 0 dans la plupart des cas !

Lecture/Ecriture

La fonction open

Principe

Création d'une variable de type fichier (File)

Accès aux fonctions de lecture/écriture

Exemple

```
1 f = open("file.txt","r")  
2  
3 f = open("/path/file.txt", "w")
```

La fonction open

Modes d'ouverture

Chaque fichier peut être ouvert dans un objectif précis

- `r` → lecture
- `w` → écriture
- `a` → écriture en fin de fichier. Création si fichier inexistant.
- `b` → fichier binaire
- `t` → fichier texte
- `x` → création fichier pour écriture.

La fonction close

Principe

Fermeture du flux d'accès au fichier

Appel à chaque fin de traitement sur un fichier

Exemple

```
1     f = open("file.txt","r")  
2     ...  
3     f.close()
```

Lecture

Principe

Récupération du contenu du fichier

Fonction read

Exemple

```
1     f = open("file.txt","r")
2     r = f.read()
3
4     print(r)
```


Lecture

Principe

Il est possible de sélectionner un nombre de caractères précis

Exemple

```
1     f = open("file.txt","r")
2     r = f.read(10)
3
4     #Affiche 10 chars
5     print(r)
```

Lecture

Exemple

```
1      #Lecture ligne par ligne
2      r = f.readLine()
```

Exemple

```
1      #Lecture ligne par ligne
2      allLines = f.readlines()
3      for line in allLines :
4          print line
```

Ecriture

Principe

Ecriture dans le fichier

Fonction write

Exemple

```
1 f = open("file.txt","r")  
2 f.write("HelloWorld!\n")
```

Ecriture

Principe

Fonctions similaires à la lecture

Exemple

```
1 f = open("file.txt","r")  
2 allLines = ["This", "is", "Python!"]  
3 f.writelines(allLines)
```

Seek

Principe

Seek permet de placer le handle au n-ieme bit du fichier.

Exemple

```
1      f = open("file.txt","r")
2      f.seek(10)
3      #on prends les chars 10      20
4      texte = f.read(10)
```

Synthèse

- Ouvrir les fichiers en lecture ou écriture
- Gestion du handle
- Fermer les fichiers après utilisation