

Documentation Python

Table des matières

Références	1
Gestion de fichiers	1
Gestion des chaînes de caractères	2
Scripts avec paramètres sur la ligne de commande	3

Références

A.P.I. : <https://docs.python.org/3.4/library/index.html>

tutoriel : <http://fr.openclassrooms.com/informatique/cours/apprenez-a-programmer-en-python>

livre : http://inforef.be/swi/download/apprendre_python.pdf

Gestion de fichiers

1. Changement du répertoire de travail

```
from os import chdir
chdir("/users/olivier/projetpython")
```

La première ligne permet d'importer la commande `chdir()` du module `os` contenant une série de fonctions permettant de dialoguer avec le système d'exploitation (`os` = operating system).
La deuxième ligne permet de spécifier le répertoire courant de l'interpréteur python, i.e. le répertoire où il va récupérer les fichiers.

2. Ouverture d'un fichier

```
f=open('test.txt','r') # Pour ouvrir un fichier en mode lecture
f=open('test.txt','w') # Pour ouvrir un fichier en mode écriture
f=open('test.txt','a') # Pour ouvrir un fichier en mode ajout
```

Le fichier `test.txt` doit être dans le répertoire courant

En lecture le fichier doit exister.

En écriture le fichier est soit créé s'il n'existe pas, soit effacé s'il existe

En ajout le fichier est ouvert et les données seront ajoutées à la fin.

3. Fermeture d'un fichier

```
f.close()
```

4. Ecriture dans un fichier

```
f.write("Bonjour\n") # le \n à la fin du texte permet
d'aller à la ligne.
```

5. Lecture dans un fichier

- Lit tout le fichier et le stocke dans une variable :

```
b=f.read()
```

- Lit la ligne courante du fichier et la stocke dans une variable :

```
b=f.readline()
```

- Boucle de type "pour chaque" qui lit et affiche toutes les lignes d'un fichier

```
for ligne in fichier:
    print(ligne)
```

La variable "ligne" est vue comme un tableau de chaînes de caractères.

Attention à deux choses : le ":" après le nom du fichier et la tabulation devant les instructions de la boucle.

Exemple complet

- On veut afficher le contenu du fichier test.txt suivant :

**Bonjour
Il fait beau
à bientôt**

```
from os import chdir
chdir("/users/olivier/projetpython")
f=open('test.txt','r')
for ligne in f :
    print(ligne)
f.close()
```

6. Existence d'un fichier

```
import os
os.path.exists(nomfichier)
renvoie vrai si le fichier existe faux sinon.
```

Gestion des chaînes de caractères

Une chaîne de caractères est vue comme un tableau de caractères. L'indice du tableau commence à 0.

- Exemples de mise en oeuvre

```
s= "Bonjour"
print s[2]    # affiche "n"
print s[2:5]  # affiche "njou"
print len(s)  # affiche 7 la longueur de la chaîne
```

- Une chaîne peut être « éclatée » en mots.

```
tMots=s.split() # les mots de la chaîne s sont stockés dans le tableau tMots
phrase = 'It is raining cats and dogs'
tMots=phrase.split()
print (tMots[3]) # affiche le 4 ème mot de la phrase : "cats"
print (tMots[3][1]) # affiche la deuxième lettre du quatrième mot : "a"
```

- La chaîne peut être « éclatée » selon un séparateur désigné :

```
chemin = '/home/btssio/www/index.html'
listeRep = chemin.split('/')
for elem in listeRep :
    print (elem)
```

Résultat :

```
home
btssio
www
index.html
```

- On peut remplacer des caractères dans une chaîne :

```
cheminBis = chemin.replace('/', '_')
print cheminBis # affichera "_home_btssio_www_index.html"
```

- Recherche d'une sous-chaîne dans une chaîne :

```
sousChemin = chemin.find('btssio') # affichera 5
rang du premier caractère de la sous-chaîne "btssio" dans la chaîne
"/home/btssio/www/index.html "
```

Scripts avec paramètres sur la ligne de commande

On doit parfois pouvoir exploiter des paramètres passés en ligne de commande :

Exemple : `python3 genererScriptCreateV3.py toulouse.txt creerUtilisateurs.sql`

- Le 1^{er} paramètre

```
import sys
premier=sys.argv[1]
```

- Le 2^{ème} paramètre

```
deuxieme=sys.argv[2]
```

- Le nom de la commande

```
commande=sys.argv[0]
```

- Exécution d'un script avec des arguments :

- depuis un terminal

```
$python3 nom_script argument1 argument2
```

- depuis NetBeans :

- modifier les propriétés du projet comme suit :

The image shows two parts of the NetBeans IDE interface. The top part is the 'Project Properties - projet1_1sio' dialog box. In the 'Categories' list on the left, 'Run' is selected. On the right, the 'Main Module' is set to 'testPython/testArguments.py' and the 'Application Arguments' field contains 'argumentNum1argumentNum1'. To the right of this dialog, two callout boxes with arrows point to the 'Main Module' and 'Application Arguments' fields, containing the text 'Choisir le script à exécuter' and 'Saisir les valeurs des arguments' respectively. The bottom part of the image shows the 'testArguments.py' source code in the editor. The code is as follows:

```
1 import sys
2
3 script = sys.argv[0]
4 p1 = sys.argv[1]
5 p2 = sys.argv[2]
6
7 print(script, p1, p2)
```

Below the code editor, the 'Output - projet1_1sio' window shows the command executed: `'testPython/testArguments.py argument1 argument2'`.