

Python: chaînes de caractères

1 Exercices d'échauffement

Exercice 1 - Manipulation des noms de fichiers/dossiers

Commencer par créer un répertoire de travail avec quelques fichiers (nommés `test1.txt`, `test2.txt`, `test3.txt`, etc mais au plus 100 et tous vides). Dans un programme Python, réaliser les opérations suivantes :

Q.1) Placer vous dans le répertoire de travail que vous avez créé

Q.2) Demander à l'utilisateur un nom de répertoire et créer ce nouveau répertoire (si il n'existe pas déjà)

Q.3) Créer un nouveau fichier (vide) nommé « `test.txt` » dans ce répertoire de travail (utiliser les commandes proche de celle ci-dessous)

```
fhandle = open('test.txt')
fhandle.close()
```

Q.4) Afficher ensuite la liste des fichiers de la forme « `test*.txt` » qui existent vraiment dans votre répertoire

Q.5) Pour chaque fichier existant « `test*.txt` », créer un nouveau fichier (vide) dont de la forme « `save_test*.txt` ». Si le fichier existait déjà, il sera préalablement supprimé

Q.6) Même question que précédemment en mettant les fichiers créés dans un dossier dont le nom sera de la forme « `save_AAMMDD_HHhMM` », c'est-à-dire que le nom du répertoire dépendra de l'heure d'exécution du programme (Vous consulterez des aides en ligne pour découvrir comment récupérer la date et l'heure).

Exercice 2 - Impression verticale

Réalisez un programme qui imprime un à un les caractères d'une chaîne saisie par l'utilisateur (demander la saisie utilisateur, faire le traitement). Par exemple, la saisie 'Timo' affichera

```
T
i
m
o
```

2 Manipulations de base des chaînes de caractères

Exercice 3 - Palindrome

Écrivez un programme qui demande la saisie d'une chaîne de caractère puis indique s'il s'agit d'un palindrome ou non par un message. *Aide* : L'accès au caractère `n` d'une chaîne `s` se fait par `s[n]`, sa longueur s'obtient par `len(s)` et, le numéro du premier caractère étant `0`, le dernier s'obtient par `s[len(s) - 1]`

Exercice 4 - Méthode `title`

Écrire un programme qui demande la saisie d'une phrase à l'utilisateur et affiche en retour une nouvelle chaîne dont la première lettre de chaque mot est en majuscule. Par exemple, «je mange du fromage» donnera «Je Mange Du Fromage»

(on suppose que chaque mot est séparé par un espace et qu'il n'y a pas de symboles de ponctuation). On utilisera pour cela la méthode `upper`.

Remarque : cette fonctionnalité existe déjà en Python sous forme de la méthode `title`.

Python: chaînes de caractères

Exercice 5 - Reverse Word

Écrire un programme qui demande la saisie d'une phrase à l'utilisateur et affiche en retour une nouvelle chaîne dont les mots sont dans un ordre d'apparition inverser ! Par exemple, «je mange du fromage» donnera «fromage du mange je».

3 Manipulation des chaînes comme commandes/noms de fichiers

Exercice 6 - Création d'une commande système

L'objectif des cet exercice est de concevoir une commande système de la forme ci-dessous :

```
gdalwarp -t_srs '+proj=utm +zone=XX +datum=WGS84' FILEIN FILEOUT
```

La zone noté XX correspond à un numéro de zone UTM de l'image à transformer (voir <http://www.dmap.co.uk/utmworld.htm>). Cette zone sera donné par l'utilisateur.

Dans cet exercice, on s'intéresse principalement à la construction de cette chaîne de caractères. Vous pourrez éventuellement tester son exécution ensuite.

Q.7) Commencer par créer un programme qui demande à l'utilisateur le nom du fichier à traiter, puis demande la zone UTM (doit être valide). Vous pourrez alors ajouter un message de confirmation de la saisie correcte ou non.

Q.8) Traitement du fichier d'entrée : vérifier tout d'abord que l'extension du fichier est bien l'extension « .tif », ensuite, vous transformerez le fichier « path/input.tif » en fichier « path/input_out.tif » (NB : path peut être vide ou non!)

Q.9) Construire ensuite la chaîne de caractères correspondant à la commande et l'afficher à l'écran

Q.10) Finalisation : utilisation de la commande `os.system`

NB : on verra dans un prochain cours comment traiter tous les fichiers d'un répertoire

Exercice 7 - Création d'une requête pour images MODIS

La première partie du travail consiste à comprendre comment récupérer des images. On dispose d'une "interface web" qui donne la possibilité de télécharger une image au format GeoTIFF en effectuant la requête suivante :

```
http://lance-modis.eosdis.nasa.gov/imagery/subsets/?  
project=aeronet&subset=Carpentras.2014076.terra.721.2km.tif
```

D'un point de vue informatique, il est possible d'automatiser la récupération des images à partir du moment où on sait synthétiser une telle requête HTTP. Il faut donc en comprendre la structure. Dans ce cas précis, cette requête permet de récupérer l'image de la zone de "Carpentras" pour la date du 17 mars 2014 (76ème jour de l'année 2014), pour le capteur Terra à une résolution de 2km et contenant les bandes 721. On comprend alors rapidement que si on veut récupérer l'image du jour précédent, il suffira de proposer une requête très similaire ...

On veut pouvoir synthétiser cette chaîne de caractères à partir de saisies utilisateurs : Celui-ci précisera sur *subset* qu'il désire télécharger¹ ainsi que deux dates : la date de début et la date de fin de la collecte d'image.

Q.1) Commencer par créer le programme qui demande aux utilisateurs les informations utiles (vous indiquerez à l'utilisateur le format de date qu'il est sensé utiliser, si il ne respecte pas ce format, vous mettrez fin au programme ou redemanderez une information). Pour vérifier vous afficherez les saisies effectuées.

¹ <https://lance-modis.eosdis.nasa.gov/imagery/subsets/?project=aeronet>

Python: chaînes de caractères

Q.2) Compléter ensuite votre programme en transformant les dates en nombre de jours (vous pourrez utiliser une opération de différence entre dates pour avoir cette valeur et le module datetime : datetime.datetime(2017, 1, 1) crée une date au 1^{er} janvier 2017 et datetime.datetime.now() donne la date d'aujourd'hui)

Q.3) Créer ensuite la chaîne de caractère correspondant à la requête pour la première date saisie par l'utilisateur

Q.4) Vous utiliserez cette chaîne de caractères avec la librairie urllib2 pour effectuer le chargement des images

```
filehandler =urllib2.urlopen( url )
fout = open(filename, "wb")
fout . write( filehandler .read() )
fout . close ()
```

Q.5) Finaliser en itérant le chargement sur toutes les dates.