

TP de Langages de script n° 2 : Scripts, fichiers et dictionnaires.

Exercice 1 : Premier module.

1. Créez un module `tp2` dans un fichier `tp2.py`, en suivant la structure donnée en cours. N'oubliez pas d'inclure un chaîne de documentation décrivant le module.
2. Définissez un variable `uteur` contenant votre nom, ainsi qu'une fonction `copyright()` affichant à l'écran que vous êtes l'auteur de ce programme et que vous poursuivez à justice quiconque le copiera sans autorisation. N'oubliez pas d'inclure un chaîne de documentation décrivant la fonction.
3. Faites en sorte que, lorsque le fichier est exécuté en tapant son nom dans le terminal, il affiche le nom de son auteur et le message de copyright.
4. Importez le fichier dans l'interpréteur Python. Qu'est-ce qui se passe-t-il ? Modifiez votre script pour que les commandes de la question précédente ne s'exécutent que si le programme est appelé depuis le terminal, et pas quand il est importé dans l'interpréteur.
5. Depuis un interpréteur, consultez la documentation sur votre module grâce à la commande `help` et son environnement local grâce à la commande `dir()`. Importez la variable `uteur` du module dans l'environnement global grâce à la commande `from ... import ...`.

Exercice 2 : Traduction automatique

On souhaite écrire un programme de traduction automatique simplifié. Le dictionnaire de traduction est stocké dans un fichier dont chaque ligne est de la forme `mot:traduction` (on suppose que `traduction` est un mot unique sans espace).

1. En utilisant la méthode `split` du type `string`, trouvez la commande permettant de transformer un chaîne de caractères contenant : un tableau contenant la première et la seconde partie de la chaîne séparée par ces deux points.
2. Écrivez une fonction `read_lexic` prenant comme argument le nom d'un fichier et renvoyant un dictionnaire PYTHON dont chaque couple (clé, valeur) correspond à un mot du lexique et à sa traduction.
3. Écrivez une fonction `translate_word` prenant un mot et un dictionnaire et renvoyant sa traduction selon le lexique.
4. Écrivez une fonction `add_word` prenant deux mots et un dictionnaire et ajoutant le mot et sa traduction au dictionnaire.
5. Écrivez une fonction `write_lexic` réciproque de la fonction `read_lexic`. Améliorez cette fonction en faisant en sorte que les entrées soient classées par ordre alphabétique.
6. Écrivez une fonction principale qui demande à l'utilisateur un mot et lui fournit la traduction si elle existe dans le lexique. Si elle n'existe pas, il demande à l'utilisateur de donner la traduction et la sauvegarde dans le dictionnaire.
7. Écrivez un fichier contenant un lexique de base et utilisez-le pour tester votre script.

Exercice 3 : Convertisseur

L but d c t x reic st d cré r un script `convertisseur` qui conv rtit l s t mpératur s t l s longu urs ntr l systèm anglo-saxon t l systèm français. On pourra l'utilis r dans un `termin l` d la façon suivant :

```
./convertisseur [-tl] [-s <systeme initi l>] [nombres]
```

`-t` = temper ture

`-l` = longueur

`-s` = introduit le systeme initi l :

A = nglo-s xon

F = fr nc is

1. Écrire l s fonctions `celsius`, `f hrenheit`, `metre t y rd` qui p rm tt nt d fair l s différ nt s conv rsions. L s t st r.
2. Écrire un fonction `help` qui écrit dans l `termin l` un m ssag aidant à la bonn utilisation du script. La t st r.
3. En vous aidant du modul `sys` t d la méthod `argv`, écrire la fonction principal du script qui récupèr l s différ nts argum nts, puis s lon l s cas, app ll un fonction d conv rsion sur l'argum nt adéquat ou app ll la fonction d'aid .

Exercice 4 : Géographie

L but d c t x reic st d cré r un script `c pit les` qui p rm t d t st r s s connaissances sur l s capital s du mond . On pourra l'utilis r dans un `termin l` d la façon suivant :

```
./c pit les -v ville
```

-> le script dem nde l'utilisateur le nom du p ys dont l c pit le est ville, puis verifie l reponse

```
./c pit les -n
```

-> le script donne un nom de c pit le et le joueur doit trouver le noms de p ys ssocié. Le joueur peut decider de continuer ou d'rreter, l fin il obtient une st tistique sur le nombre de bonne reponse.

1. Écrire un fonction qui conv rtit un fichi r cont nant d s pays t l ur capital n un dictionnair . T st r c tt fonction sur l fichi r `c pit les.txt` qu vous trouv r z sur Did l.
2. Écrire un fonction pour chacun d s options `-v`, `-n` (on s'appui ra pour la d rnière sur l modul `r ndom`).
3. Écrire la parti principal du cod qui s lon l s options app ll la bonn fonction.