

TP de Langages de script n° 2 : premiers scripts PYTHON

Exercice 1 : Premier module.

1. Créez un module `tp2` dans un fichier `tp2.py`, en suivant la structure donnée en cours. Ne pas oublier d'inclure une chaîne de documentation décrivant le module.
2. Définissez une variable `auteur` contenant votre nom, ainsi qu'une fonction `copyright()` affichant à l'écran que vous êtes l'auteur de ce programme et que vous poursuivrez en justice quiconque le copiera sans autorisation. N'oubliez pas d'inclure une chaîne de documentation décrivant la fonction.
3. Faites en sorte que, lorsque le fichier est exécuté en tapant son nom dans le *shell*, il affiche le nom de son auteur et le message de copyright.
4. Importez le fichier dans l'interpréteur Python. Que 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 *shell*, et pas quand il est importé dans l'interpréteur.
5. Consultez l'aide en ligne de votre module grâce à la commande `help` et son environnement local grâce à la commande `dir()`. Importez ensuite la variable `auteur` du module directement dans l'environnement global grâce à la commande `from ...import ...`.

Exercice 2 : On observe les types et on imprime en grand.

1. Créez une variable `x` de valeur `'toto'`. La fonction `type` vous donne le type de cette variable. Comparez le résultat de l'application de la fonction `dir` sur `x` et sur son type. Toutes les variables que vous déclarez (entier, chaînes de caractère, ...) sont en fait des objets d'une classe d'où le fait que `dir(x)` et `dir(str)` retournent la même chose. Pour utiliser une méthode il suffit d'utiliser la syntaxe `x.methode()`. Par exemple si `x` vaut `'toto'` alors `x.isupper()` va tester si `x` est bien écrit en majuscules. On peut également utiliser la syntaxe `str.isalpha('toto')` si on veut utiliser la méthode `isalpha` définie dans la classe `str` sans déclarer de variables.
2. Comment mettre une chaîne de caractères en minuscules ?
3. Téléchargez le fichier `grandes_lettres.py` et importez le module correspondant dans l'interpréteur python. Regardez l'aide en ligne de ce module.
4. À l'aide du module `grandes_lettres`, écrivez une fonction `grand_message` qui prend en argument une chaîne de caractères et réécrit le texte en grandes lettres. Pour cela il faudra imprimer les étoiles ligne à ligne.

Exemple : `'toto'` sera réécrit

```
*****  **  *****  **
 *    *  *    *    *  *
 *    *  *    *    *  *
 *    *  *    *    *  *
 *    **   *    **
```

5. À l'aide de l'instruction `input()` et du code que vous avez mis au point précédemment, créez un script `grand_message.py` demandant à l'utilisateur de rentrer une chaîne de caractère et l'imprimant en grand.

Exercice 3 : Mot sans cube.

Pour cet exercice vous écrirez votre code dans un module.

On considère les mots sur un alphabet à deux lettres $\{a, b\}$. On dit qu'un mot (c'est-à-dire une suite de lettres) possède un cube s'il possède un facteur de la forme uuu où u est un mot non vide. Par exemple *baaa* et *bababa* possèdent un cube alors que *bababb* est sans cube. On peut montrer qu'il existe un mot infini sans cube (mais ce n'est pas l'objet de cet exercice). Nous allons être plus modestes et écrire une fonction qui teste si un mot est sans cube¹.

1. Que fait la méthode `endswith` du module `str`? Trouvez la méthode symétrique.
2. Dans un interpréteur Python, regardez ce que donne `3*'ab'`.
3. Écrivez une fonction `est_sans_prefixe_cube` qui prend en argument une chaîne de caractères et détermine si elle possède un cube comme préfixe.
4. Écrivez une fonction `est_sans_cube` qui prend en argument une chaîne de caractères et détermine si elle est sans cube. Pour cela vous pourrez tester les suffixes de l'argument grâce à la fonction précédente.

Exercice 4 : Codes secrets.

1. À partir de votre terminal (et sans avoir lancé l'interpréteur Python), consultez grâce à `pydoc` l'aide en ligne des fonctions `chr` et `ord`.
2. Créez la chaîne de caractères `message` contenant la valeur `'ceci est mon message a chiffrer'`. À l'aide d'une boucle `for` sur cette chaîne, chiffrez-la par un décalage de 3 (*chiffre de César*). Par exemple la lettre `a` sera chiffrée par la lettre `d` (et la lettre `x` par la lettre `a`).
3. Créez maintenant une chaîne de caractères `clef` dont la valeur est `'secret'`. Chiffrez la chaîne `message` par décalage grâce à la clef `clef` (*chiffre de Vigenère*). Ce chiffrement se fait en décalant la i^e lettre du message grâce à la i^e lettre de la clef (on reprend au début de la clef quand on a fini de lire celle-ci). Mathématiquement on peut écrire : lettre chiffrée = (lettre + clef) modulo 26.
Pour simplifier, on ne modifiera pas les caractères blancs (espace, tabulation et retour chariot), la ponctuation...

1. Dans un TP ultérieur nous construirons la liste des mots sans cube d'une longueur donnée.