

Exercices en Python

1 Convertisseur de température

On veut écrire un programme python qui convertisse une température des degrés Celsius vers les degrés Fahrenheit et inversement.

↪ Formules de conversion :

$$T_c = \frac{5}{9} * (T_f - 32)$$

$$T_f = \frac{9}{5} * T_c + 32$$

où T_c est la température en degrés Celsius et T_f celle en degré Fahrenheit.

Première version Le programme demandera à l'utilisateur la température, puis le sens de la conversion, puis affichera le résultat, avec une précision de deux chiffres après la virgule.

```
Entrez une température : 12
Convertir vers des degrés (C)elsius ou (F)ahrenheit ? F
Résultat : 53.60
```

Deuxième version Faites-en une application en ligne de commande, le premier argument étant la température et le second le sens de conversion.

```
user@host> python temp_convertter.py 12 F
Résultat : 53.60
user@host>
```

2 Vérificateur d'URLs

Écrire un programme qui teste une URL donnée en paramètre :

```
C:\dir> python urlcheck.py malformed.url
Unknown URL format
C:\dir> python urlcheck.py http://non-existing.ch
Impossible to load URL
C:\dir> python urlcheck.py http://www.he-arc.ch
The URL is valid
C:\dir>
```

↪ En utilisant `urllib2.urlopen()` et les exceptions, c'est très simple !

3 Un clone de `strings`

La commande Unix `strings` sert à trouver et afficher les séquences de caractères imprimables contenues dans un fichier binaire.

Écrire un programme en ligne de commande qui prend un nom de fichier en argument et affiche toutes les chaînes de caractères imprimables d'une longueur supérieure ou égale à 4.

↪ `string.printable` pourrait vous être utile

4 Compter les mots

Écrire un programme en ligne de commande qui prend un nom de fichier texte en argument et affiche, en ordre alphabétique, chaque mot présent dans le fichier accompagné de son nombre d'occurrences.

```
allusion : 1
buisson : 5
définition : 2
...
```

5 Rot13

Écrire un programme interactif qui demande une chaîne à l'utilisateur et la convertit en codage rot13 (décalage circulaire de 13 lettres dans l'alphabet). La conversion respectera les majuscules et les minuscules et ne changera rien aux caractères hors de `[A..Z]` et `[a..z]`.

Le programme tournera en boucle tant que l'utilisateur n'a pas entré une chaîne vide.

Exemple :

```
C : \> python rot13.py
-> Hello
Uryyb
-> Salut, Dédé !
Fnyhg, Qéqé !
->
C : \>
```

6 Exercices facultatifs

Pour les plus rapides, les plus motivés... ou ceux qui connaissaient déjà python !

6.1 Un petit bug...

Vous avez un accès en lecture (et en exécution) au programme suivant, mais pas en écriture :

```
from random import randint

to_guess = randint(0,100)

guessed = input("Entrez un nombre (entre 0 et 100): ")

if guessed == to_guess:
    print "Gagné!"
else:
    print "Perdu!"
```

1. Chaque partie coûte 10.– Une partie gagnée rapporte 900.–, une partie perdue rien du tout. Pouvez-vous trouver une manière de vous enrichir à coup sûr ?
2. Le propriétaire du jeu, avant d'être ruiné, vous a engagé pour rendre son code plus sûr. Que proposez-vous ?

7 Un petit jeu vidéo

Le cours python en ligne "Livewires" (<http://www.livewires.org.uk/python/>) propose un module permettant de développer facilement de petits jeux vidéos¹.

1. Installez PyGame (<http://www.pygame.org/download.shtml>)
2. Récupérez et décompressez l'archive
3. Étudiez et testez le code du fichier `exemple.py`
4. Sur la base de cet exemple et de la doc fournie (fichier `W-livewires.pdf`), développez un petit jeu vidéo de votre choix.

Quelques idées pour ceux qui manquent d'inspiration :

- un casse-briques,
- un clone de Space Invaders ,
- ou de Asteroids,
- ou d'un autre jeu cité dans les "most popular games" de http://en.wikipedia.org/wiki/Golden_age_of_arcade_games
- ...

¹Il s'agit au fait d'une couche de simplification au-dessus du module PyGame.