Subject

Yan

October 2022

Fait avec amour pour la Fête de la science.

Contents

1 Introduction

Le but de cet atelier et de vous faire travailler sur des notions assez simples dans le langage Python. Cet atelier veut aussi vous familiariser avec le système de tests. En effet, à l'EPITA, il y a très régulièrement des travaux d'informatique pratique à rendre, qui seront ensuite testés et corrigés.

Ceci est le sujet du TP que vous vous allez devoir effectuer. En tant que rédacteur du sujet, je ne sais pas si j'ai su bien jauger le temps qu'il faudra pour tout faire. Donc ne vous attardez pas trop sur un exercice si vous voyez que vous perdez du temps ou bien si vous bloquez.
Le sujet est divisé en deux parties, indépendantes les unes des autres ! Vous pouvez donc commencer par celle que vous voulez (il n'y a pas d'ordre imposé).

2 Architecture du TP

Le TP est composé de 6 fichiers et s'organise comme ceci :

```
.
├── Caesar.py
├── Game.py
├── Hangman.py
├── subject.pdf
├── tests.py
└── words.txt
```

Le fichier "Caesar.py" contient tout ce qui est relié à l'exercice du code césar, aussi appelé chiffrement par décalage.

Les fichiers "Game.py" et "Hangman.py" concerne tous les deux le jeu du pendu. Le fichier "words.txt" est un dictionnaire de mots qui vous sera utile pour le jeu du pendu également.

Le fichier "tests.py" est présent uniquement pour tester votre code. Vous n'avez aucunement besoin de modifier ce fichier (sinon après ça casse tout...). Enfin, il y a le fichier "subject.pdf" qui contient le sujet que vous lisez actuellement.

3 Tester son code

Pour tester votre code, rien de plus simple ! Il vous suffit d'exécuter depuis votre terminal la commande :

```
python tests.py
```

Il vous suffira ensuite de répondre en appuyant sur "n" si vous voulez dire "non", et sur n'importe quelle autre touche de votre clavier pour répondre "oui". Attention, il se peut que cette commande ne marche pas. Si c'est le cas (normalement ça devrait aller mais on sait jamais...), remplacez "python" par "python3" lors de l'écriture de la commande.

4 Le code César

Le code César est une méthode de chiffrement qui consiste en une substitution mono-alphabétique : chaque lettre est remplacée par une seule autre, selon un certain décalage dans l'alphabet.

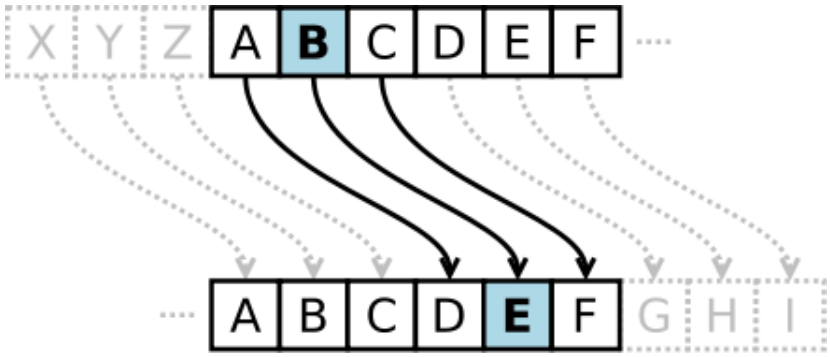

Figure 1: Exemple pour un décalage de 3 lettres. (Wikipedia)

Vous pouvez trouver plus d'informations sur Wikipedia (vous pouvez cliquer ici, ça ouvre direct la page. Oui, c'est super bien fait.

4.1 Caesar.py

Dans ce fichier, vous avez trois fonctions à écrire. Elles sont toutes les trois liées. Il est obligatoire d'écrire dans un premier temps la fonction shift, avant de pouvoir écrire les deux autres ensuite. Les fonctions sont documentées, mais si vous avez besoin d'aide n'hésitez pas à demander à votre/vos animateur(s) qui, même s'ils sont nuls, devraient pouvoir vous aider parce qu'ils ont la correction. Pensez bien à effacer la ligne avec le mot clé "pass" et écrivez votre code à la place.

```
# Cette premi re fonction permet de decaler une seule lettre

def shift(letter: str, s: int) -> str:
    pass # remove

# Ensuite, on la reutilise pour decaler une par une chaque lettre du mot/de
#   la phrase fournie en param tre.
# Les deux fonctions ci-dessous se ressemblent beaucoup, ne vous cassez pas
#   trop la t te une fois que vous avez reussie l'une des deux parce qu'elles se ressemblent.

def encode_word(word: str, s: int) -> str:
    pass # remove

def decode_word(word: str, s: int) -> str:
    pass # remove
```

Listing 1: Python example

5 Le jeu du pendu

Le jeu du pendu est un jeu qui consiste à deviner un mot en fonction des lettres qui le composent et en ayant un nombre limité d'essais possibles. Si vraiment vous connaissez pas, vous pouvez toujours cliquer ici pour voir sur vous pouvez toujours cliquer ici pour voir sur Wikipedia.

5.1 Game.py

Ce fichier contient des données utiles au programme que vous n'avez pas besoin de modifier (que vous ne DEVEZ PAS modifier, sinon ça casse tout). Il contient également trois fonctions que vous devez écrire cette fois. Ces fonctions sont les suivantes :

```
    # ... code above

    # fonction qui renvoie True ou False selon si la partie est
    gagnee ou non.
    def isWon(self) -> bool:
        pass # remove

    # fonction qui renvoie True ou False selon si la partie est
    perdue ou non.
    def isLost(self) -> bool:
        pass # remove

    # fonction qui renvoie True ou False selon si la partie est
    finie ou non.
    def isFinished(self) -> bool:
        pass # remove
```

Listing 2: Python example

Pensez bien à effacer la ligne avec le mot clé "pass" et écrivez votre code à la place.

5.2 Hangman.py

Ce fichier contient les trois dernières fonctions à effectuer :

```python
# Cette fonction renvoie un mot choisi aleatoirement dans la liste de mots ("dictionary").
def getSecret() -> str:
    pass # remove

# fonction pour remplacer une ou plusieurs lettres dans un mot selon un mod le.
def replaceLetter(currentWord: str, secretWord: str, letter: str) -> str:
    pass # remove

# fonction pour lancer le jeu
def play():
    # Cette ligne initialise une partie avec un mot secret choisi gr ce    la fonction que vous avez definie plus haut.
    game = Game(getSecret())
    pass # remove
```

Listing 3: Python example

Pensez bien à effacer la ligne avec le mot clé "pass" et écrivez votre code à la place.

6 Fin

Et voilà, vous avez fini le TP ! J'espère que c'était pas trop compliqué mais un peu quand même, je ne sais vraiment pas jauger. J'espère quand même que certains d'entre vous auront réussi à finir, et si c'est le cas, félicitations !