



TP N°1.1 - Python

Premiers programmes

Thibault Napoléon
thibault.napoleon@isen-ouest.yncrea.fr

Préambule

Vous prendrez soin, pour l'ensemble des exercices, de rendre votre code le plus lisible possible en le commentant, en utilisant les *docstrings* et en utilisant des fonctions lorsque c'est nécessaire.

1 Le nombre caché

Écrivez un programme permettant de proposer à l'utilisateur un petit jeu où il doit retrouver un nombre tiré aléatoirement entre deux bornes. Afin de le guider, à chacune de ses propositions vous lui donnerez comme indication : « *plus grand* » ou « *plus petit* ». Afin d'améliorer le programme, vous demanderez le nom du joueur et vous limiterez le nombre d'essais. Ci-dessous un exemple de déroulement du jeu avec en rouge les saisies de l'utilisateur :

```
Entrez votre nom > Tibo

Essai 1/6
Entrez un nombre entre 0 et 20 > aa
Entier incorrect.

Essai 1/6
Entrez un nombre entre 0 et 20 > 10
Le nombre est plus grand

Essai 2/6
Entrez un nombre entre 0 et 20 > 15
Le nombre est plus grand

Essai 3/6
Entrez un nombre entre 0 et 20 > 17
Le nombre est plus grand

Essai 4/6
Entrez un nombre entre 0 et 20 > 18
Le nombre est plus grand

Essai 5/6
Entrez un nombre entre 0 et 20 > 19
Bravo vous avez gagné Tibo
```

Exemple d'une exécution du programme (en rouge les saisies de l'utilisateur)

2 Min et max

Écrivez un programme qui demande à l'utilisateur de saisir une liste d'entiers dont le nombre est fixé. Vous prendrez soin de vérifier que la saisie est bien un entier. Vous écrierez ensuite une

première fonction permettant de chercher et retourner le minimum et le maximum de la liste de l'utilisateur grâce à une boucle. Vous écrierez ensuite une seconde fonction faisant la même chose mais en utilisant les fonctions *min()* et *max()* qui s'applique sur une liste python. Ci-dessous un exemple de déroulement du programme avec en rouge les saisies de l'utilisateur :

```
Entrez un entier > -87
Entrez un entier > f
Entier incorrect.
Entrez un entier > 12
Entrez un entier > 55
Entrez un entier > 5
Le min et le max de [-87, 12, 55, 5] sont -87 et 55.
Le min et le max de [-87, 12, 55, 5] sont -87 et 55.
```

Exemple d'une exécution du programme (en rouge les saisies de l'utilisateur)

3 Code IBAN (*idée de Jean-Marie Guyader*)

Dans un certain nombre de pays, les comptes bancaires sont caractérisés par un identifiant unique appelé IBAN (*International Bank Account Number*). Un IBAN peut contenir uniquement des chiffres et des lettres de l'alphabet latin sans caractères diacritiques. De plus, il doit présenter la structure suivante :

IE29 AIBK 9311 5212 3456 78

Les deux premiers caractères d'un IBAN correspondent à un code pays (dans l'exemple ci-dessus IE est le code pays de l'Irlande). Les deux caractères suivants (signalés en italique ci-dessus : 29) sont une clé de contrôle. Le reste des caractères de l'IBAN (AIBK 9311 5212 3456 78, dans l'exemple ci-dessus) constitue le BBAN (*Basic Bank Account Number*). Ce dernier peut contenir jusqu'à 30 caractères. Les deux caractères de la clé de contrôle ne sont pas choisis au hasard. Ils peuvent être calculés en utilisant l'algorithme suivant (*Wikipedia*) :

1. Replace the two check digits by 00 (e.g., IE00 for the Ireland).
2. Move the four initial characters to the end of the string.
3. Replace the letters in the string with digits, expanding the string as necessary, such that A or a = 10, B or b = 11, and Z or z = 35. Each alphabetic character is therefore replaced by 2 digits.
4. Convert the string to an integer (i.e. ignore leading zeroes).
5. Calculate mod-97 of the new number, which results in the remainder.
6. Subtract the remainder from 98, and use the result for the two check digits. If the result is a single digit number, pad it with a leading 0 to make a two-digit number.

Algorithme de calcul de la clé de contrôle d'un code IBAN

Écrivez un programme permettant de demander à l'utilisateur de saisir un code IBAN et retranscrivez ensuite l'algorithme présenté ci-dessus pour vérifier sa clé. Vous écrierez les fonctions nécessaires à la bonne lisibilité de votre programme. Ci-dessous un exemple de déroulement du programme avec en rouge les saisies de l'utilisateur :

```
Entrez un code IBAN > IE29 AIBK 9311 3456 78
La clé est 75
La clé est incorrect

Entrez un code IBAN > IE29 AIBK 9311 5212 3456 78
La clé est 29
La clé est correct.
```

Exemple de deux exécutions du programme (en rouge les saisies de l'utilisateur)

4 Palindrome

Un palindrome est une figure de style désignant un mot/phrased qui peut se lire, de façon identique, de gauche à droite ou de droite à gauche. Écrivez un programme qui permet de savoir si une chaîne, saisie par l'utilisateur, est un palindrome. Ci-dessous un exemple de déroulement du programme avec en rouge les saisies de l'utilisateur :

```
Entrez le mot > loupe
loupe n'est pas un palindrome.

Entrez le mot > Engage le jeu que je le gagne
Engage le jeu que je le gagne est un palindrome.
```

Exemple de deux exécutions du programme (en rouge les saisies de l'utilisateur)

5 Anagramme

Une anagramme est une figure de style désignant deux mots/phrases qui contiennent exactement les mêmes lettres dans un ordre différent. Écrivez un programme qui permet de savoir si deux chaînes, saisies par l'utilisateur, sont l'anagramme l'une de l'autre. Ci-dessous un exemple de déroulement du programme avec en rouge les saisies de l'utilisateur :

```
Entrez le premier mot > poule
Entrez le second mot > loupe
poule et loupe sont des anagrammes.

Entrez le premier mot > kayak
Entrez le second mot > loupe
kayak et loupe ne sont pas des anagrammes.

Entrez le premier mot > Tom Elvis Jedusor
Entrez le second mot > Je suis Voldemort
Tom Elvis Jedusor et Je suis Voldemort sont des anagrammes.
```

Exemple de trois exécutions du programme (en rouge les saisies de l'utilisateur)

6 Le pendu

Écrivez un programme permettant de proposer à l'utilisateur de jouer au jeu du pendu. Pour cela vous tirerez aléatoirement un mot parmi la liste fournie dans le fichier "*dictionary.txt*". Vous proposerez ensuite à l'utilisateur de donner une lettre qu'il pense être dans le mot masqué. Vous prendrez soin de vérifier que l'utilisateur a bien saisi une lettre et qu'il ne l'a pas déjà proposée. Vous indiquerez aussi à l'utilisateur la taille du mot masqué. Afin d'améliorer le programme, vous demanderez le nom du joueur et vous limiterez le nombre d'essais. Ci-dessous un exemple de déroulement du jeu avec en rouge les saisies de l'utilisateur :

```
Entrez votre nom > Tibo

Erreur 0/7
Mot : - - - - -
```

```
Entrez une lettre > E

Erreur 0/7
Mot : - - - - E - -
Entrez une lettre > A

Erreur 0/7
Mot : - - - - E A -
Entrez une lettre > S

La lettre ne se trouve pas dans le mot masqué.
Erreur 1/7
Mot : - - - - E A -
Entrez une lettre > T

La lettre ne se trouve pas dans le mot masqué.
Erreur 2/7
Mot : - - - - E A -
Entrez une lettre > O

Erreur 2/7
Mot : - O - - E A -
Entrez une lettre > N

La lettre ne se trouve pas dans le mot masqué.
Erreur 3/7
Mot : - O - - E A -
Entrez une lettre > M

Erreur 3/7
Mot : M O - - E A -
Entrez une lettre > R

Erreur 3/7
Mot : M O R - E A -
Entrez une lettre > T

Lettre déjà donnée.
Entrez une lettre > 8

Ce n'est pas une lettre.
Entrez une lettre > C

Erreur 3/7
Mot : M O R C E A -
Entrez une lettre > U

Mot : M O R C E A U
Bravo vous avez gagné Tibo
```

Exemple d'une exécution du programme (en rouge les saisies de l'utilisateur)