

TP fonctions et structures conditionnelles

Exercice 1

1. Ecrire la fonction `str_est_pair` qui prend en argument `n` un nombre entier et retourne la chaîne de caractères `est pair` si `n` est pair et `n'est pas pair` sinon.
2. La tester avec les nombres 2, 7, -11 et -12.

Exercice 2

1. Ecrire une fonction `str_est_entier` qui prend en argument `n`. Si `n` est un entier cette fonction retourne `str(n) + 'est un entier'` sinon elle retourne `str(n) + "n'est pas entier"`.
2. Tester votre fonction en l'appelant avec le paramètre 4 puis 3.5 et enfin 'test'.

Exercice 3

1. Ecrire une fonction `str_un_octet` qui prend en paramètre un entier `n` et qui retourne la chaîne "s'écrit sur un octet" si `n` est compris entre 0 et 255 et "ne s'écrit pas sur un octet" sinon.
2. Tester votre fonction sur les nombres 3, -3, 257, 0 et 255.
3. Modifier votre fonction afin qu'elle affiche en plus le nombre `n`. Ainsi l'appel `str_un_octet(253)` doit retourner la chaîne `253 s'écrit sur un octet`.

Exercice 4

1. Ecrire une fonction `str_est_majeur` qui prend en paramètres une chaîne de caractères `nom` et un entier `annee` et qui retourne le nom de la personne suivi de `est majeur(e)` si la personne est majeure et le nom de la personne suivi de `est mineur(e)` sinon.
2. Vérifier que l'appel `str_est_majeur('Jean', 2000)` retourne `Jean est majeur(e)` et que l'appel `str_est_majeur('Anna', 2020)` retourne `Anna est mineur(e)`.

Exercice 5

1. Ecrire une fonction `lancer_piece_truquee` ne prend pas d'argument et qui retourne `Pile` 70 % du temps. Pour cela vous utiliserez la fonction `random()` du module `random` qui retourne un nombre pseudo-aléatoire compris entre 0 et 1. Ainsi si la fonction `random()` retourne un nombre inférieur à 0,70 la fonction doit retourner `Pile` sinon, elle doit retourner `Face`. Voici un exemple d'utilisation de la fonction `random()`

```
import random as rd
print(rd.random()) # Affiche un nombre pseudo-aléatoire compris
↪ entre 0 et 1.
```

2. En dessous, de votre fonction recopier le programme qui suit. Il affiche le nombre de piles obtenu lors de la simulation de 100 lancers de cette pièce truquée.

```
compteur_piles = 0
for _ in range(100):
    if lancer_piece_truque() == 'Pile':
        compteur_piles = compteur_piles + 1
print('Nb de pile',compteur_piles,',nb de face',100 - compteur_piles)
```

3. Vérifier que l'on obtient bien environ 70 fois pile sur 100.

Exercice 6

1. Ecrire une fonction discriminant qui prend en paramètres trois flottants a, b et c qui représentent les coefficients d'un trinôme du second degré $ax^2 + bx + c$ et qui retourne l'entier représentant le nombre de racines de ce trinôme. On rappelle que le discriminant se calcule avec la formule: $\Delta = b^2 - 4ac$.
 - Si $\Delta > 0$ Le trinôme a deux racines.
 - Si $\Delta = 0$ le trinôme a une racine.
 - Sinon quand $\Delta < 0$ le trinôme n'a pas de racine réelle.
2. Vérifier que l'appel `discriminant(1,2,1)` retourne 1, que l'appel `discriminant(1,1,1)` retourne 0 et que l'appel `discriminant(1,0,-1)` retourne 2.