Info224 - TD1

Rappels : expressions affectables

Exercice 1. Donnez des exemples d'*expressions* qui ne sont pas affectables.
Parmi les morceaux de Python suivant, lesquels sont des *expressions*, des *expressions affectables* ou des *instructions* ?

`t[n]`	`t[-1]`	`t[n+1]`	`t[n]+1`
`t(n)`	`t(n+1)`	`t(f[n+1])`	`t[f(n)+1]`
`x=y`	`x==y`	`x=3`	`3=x`
`import math`	`math.sqrt`	`return 0`	`print(t)`

Boucles

Exercice 2. Écrivez, en utilisant une boucle `for`, une fonction qui calcule la somme de tous les nombres contenus dans un tableau.
Ré-écrivez cette fonction en utilisant cette fois ci une boucle `while`.
Quelle version préferrez vous?

Exercice 3. Écrivez, en utilisant une boucle `for`, une fonction qui calcule la somme de tous les nombres contenus dans les cases impaires d'un tableau.
Ré-écrivez cette fonction en utilisant cette fois ci une boucle `while`.
Quelle version préferrez vous?

Exercice 4. Écrivez, en utilisant une boucle `for`, une fonction qui calcule la somme de tous les nombres pairs dans un tableau.
Ré-écrivez cette fonction en utilisant cette fois ci une boucle `while`.
Quelle version préferrez vous?

Exercice 5. On dispose d'un tableau `precipitations` qui donne le montant annuel des précipitations pour chaque année depuis 1900. Par exemple, le montant des précipitations de l'année 1925 est contenu dans la case 25 du tableau.

Écrivez une fonction qui calcule le maximum des précicipations annuelles pour toutes les années non bissextiles. (Rappel : les années bissextiles sont les années divisible par 4 ou 400, mais pas par 100.)
Même question, mais pour les années bissextiles.

Exercice 6. Écrivez une fonction à deux arguments `c` et `s` qui teste si une lettre (`c`) apparait dans une chaine (`s`):
- avec une boucle `for` qui parcourt tout le tableau,
- avec une boucle `for` qui s'arrête dès que la lettre cherchée est trouvée,
- avec une boucle `while` qui s'arrête dès que la lettre cherchée est trouvée.

Quelle version préferrez vous?

Exercice 7. Écrivez une fonction qui teste si un tableau est trié dans l'ordre croissant ou décroissant: si le tableau est trié (croissant ou décroissant), le résultat est `True`, sinon, le résultat est `False`.
Écrivez plusieurs versions:
- avec une boucle `for` qui parcourt tout le tableau,
- avec une boucle `for` qui s'arrête le plus tôt possible,
- avec une boucle `while` qui s'arrête le plus tôt possible,
- avec des fonctions auxiliaires.

Quelle version préferrez vous?

Exercice 8. Une balle en caoutchouc rebondit et perd, à chaque rebond, de la hauteur. Par exemple, un balle avec un coefficient de rebond de 85% perdra 15% de hauteur à chaque rebond.
Écrivez une fonction `nb_rebonds` à un paramètre `coeff` qui calcule le nombre de rebond nécessaires pour que une balle dont le coefficient est `coeff` rebondisse en dessous de 50cm lorsqu'elle est lachée de 2m de hauteur.
Ajoutez deux arguments à la fonction pour qu'on puisse facilement changer la hauteur initiale (2m) et la hauteur finale (50cm).
Pouvez-vous utiliser une boucle `for` pour cette fonction ?

Exercice 9. Écrivez une fonction `duree` qui prend 3 arguments `montant_initial`, `taux` et `montant final`, et qui calcule le nombre d'années nécessaires pour que le montant initial rapporte au moins le montant final pour le taux d'intérêt donné.

Pouvez-vous utiliser une boucle `for` pour cette fonction ?

Exercice 10. On peut simuler un lancer de dés avec la fonction `randint(1,6)` qui tire un nombre entier entre 1 et 6 au hasard. (Il faut charger la fonction avec une ligne `from random import randint` au début de votre fichier...)
Écrivez une fonction `nb_lancers` avec un argument `score` qui compte le nombre de lancers de dés qu'il faut ajouter pour obtenir au moins le score.
Pouvez-vous utiliser une boucle `for` pour cette fonction ?

Exercice 11. Modifiez la fonction précédente pour que le dernier lancer « tombe juste ». Autrement dit, si le score actuel est 23 et qu'il faut obtenir 25, un 4 comptera comme un lancer mais ne participera pas au score (on ne peut pas dépasser) ; par contre, un 1 comptera normalement car il ne fait pas dépasser.

Exercice 12. Écrivez une fonction qui parcourt un tableau de chaines et compte le nombre de cases dont la dernière lettre est un point «`.`». Votre fonction devra s'arêter soit lorsqu'elle tombe sur une case contenant la chaine vide (`""`) soit quand elle tombe sur la chaine `"THE END"`.

Exercice 13. *Rappel :* la fonction `input` attend que l'utilisateur tape quelque chose au clavier et renvoie la chaine correspondante.
Écrivez une fonction `couleur` qui affiche le menu suivant:

```
rouge : R
vert : V
bleu : B
```

et qui attend que l'utilisateur rentre une lettre. Tant que l'utilisateur ne rentre ni `R`, ni `V`, ni `B`, votre fonction devra redemander :

```
Mauvais choix
  rouge : R
  vert : V
  bleu : B
```

Votre fonction renverra le nom complet de la couleur finalement choisie.

Recherche dichotomique

Exercice 14. La fonction suivante permet de calculer l'indice d'un élément dans un tableau (ou `-1` si l'élément n'apparait pas dans la tableau).

```
def indice(T, e):
    """recherche l'indice de la valeur e dans le tableau T
Si la valeur e n'apparait pas dans T, renvoie -1"""
    r = -1
    i = 0
```

```
    while 0 <= i < len(T) and r == -1:
        if T[i] == e:
            r = i
        i = i+1
    return r
```

Lorsque la valeur `e` apparait en position n, la fonction inspecte exactement n cases du tableau. Lorsque la valeur `e` n'apparait pas, elle inspecte toutes les cases du tableau.

Lorsque le tableau `T` est trié par ordre croissant, il est possible d'inspecter moins de cases: on regarde la case au milieu de `T` et

- si cette case est égale à `e`, on a trouvé la valeur et son indice,
- si cette case est plus grande que `e`, on continue la recherche dans la première moitié de `T`,
- inversement, si cette case est plus petite que `e`, on continue la recherche dans la seconde moitié de `T`.

Bien entendu, on continue de la même manière en cherchant dans une moitié de moitié, puis une moitié de moitié de moitié, etc. (Lorsque qu'il n'y a plus de cases dans la partie concernée du tableau, c'est que la valeur n'était pas présente: on renvoie alors `-1`.)

Indice: pour chercher la valeur `e` entre les indices `i` et `j` du tableau, il faut regarder la case `(i+j)//2`, et continuer sur la bonne moitié:

- soit entre `i` et `(i+j)//2 - 1`,
- soit entre `(i+j)//2` et `j`.