Les listes en Python

Les listes sont un outil extrêmement puissant et pratique en Python, vous avez tout intérêt à les maîtriser un maximum afin de vous faciliter la vie.

1. Déclarer une liste vide

ma_liste = [ ] # Ce sont deux crochets qui se suivent, ouvrant et fermant.

2. Initialiser une liste, c'est-à-dire la déclarer non vide

ma_liste = ['lundi', 'mardi', 'mercredi', 'jeudi', 'vendredi']

ma_liste = [1, 3, 5, 7, 11]

ma_liste = list (range (10)) # contiendra la suite de 10 nombres de 0 à 9 compris.

ma_liste = list (range (5, 15)) # contiendra la suite commençant à 5 à et se terminant par 14

3. Référencement des éléments d'une liste

Chaque élément de la liste a un indice de position, le premier élément ayant l'indice 0.

ma_liste1 = [1, 3, 5, 7, 11]

print (ma_liste1 [1]) # affichera le nombre 3.

Au cas où l'indice serait en dehors de la liste, le message « ***list index out of range*** » apparaîtrait.

4. Parcourir une liste

```
for valeur in ma_liste1 :
    print (valeur)      # va imprimer tout le contenu de la liste, essayez.
```

5. Vérifier si un élément fait partie d'une liste

L'expression ***10 in ma_liste1*** donnera un résultat vrai ou faux, dans ce cas-ci, ce sera faux.

6. Connaître la longueur d'une liste

print (len (ma_liste1)) # affichera 5, le nombre d'élément dans la liste référencés de 0 à 4

7. Ajout d'un élément en fin de liste

ma_liste1.append (13) # ajoutera l'élément 13 à la fin de la liste qui contiendra 6 éléments.

8. Insertion d'un élément dans une liste

ma_liste.insert (position, élément) insérera l'élément à la position dans la liste. Exemple :

liste_jours = ['lundi', 'mercredi']

liste_jours.insert (1, 'jeudi']

print (liste_jours) # imprimera ['lundi', 'jeudi', 'mercredi']

9. Modification d'un élément d'une liste

liste_jours [1] = 'mardi'

print (liste_jours) # imprimera ['lundi', 'mardi', 'mercredi']

10. Combinaison de listes

Il y a moyen de compléter une liste originale avec le contenu d'une autre liste. Exemple :

semaine = ['lundi', 'mardi', 'mercredi', 'jeudi', 'vendredi']

week_end = ['samedi', 'dimanche']

semaine.extend (week_end)

print (semaine) # affichera [‘lundi’, ‘mardi’, ‘mercredi’, ‘jeudi’, ‘vendredi’, ‘samedi’, ‘dimanche’]

11. Suppression d’un élément de la liste

Soit sur base de l’élément proprement dit, soit sur base de son indice.

lettres = [‘a’, ‘b’, ‘c’, ‘d’]

lettres.remove (‘c’) # suppression sur base de l’élément proprement dit

print (lettres) # affichera [‘a’, ‘b’, ‘d’]

del lettres[1] # suppression sur base de la position

print (lettres) # affichera [‘a’, ‘d’]

lettres.pop (0) # aura le même effet de del. Dans ce cas, le premier élément de la liste sera supprimé puisque l’indice est ici 0.

12. Suppression de plusieurs éléments identiques d’une liste

nombres = [0, 1, 2, 0, 3, 0, 4, 0, 5]

while 0 in nombres :

    nombres.remove (0)

print (nombres) # affichera [1, 2, 3, 4, 5]

13. Vider une liste

nombres.clear ()

print (nombres) # affichera []

14. Compter le nombre d’occurrence d’un élément dans une liste

nombres = [0, 1, 2, 0, 3, 0, 4, 0, 5]

print (nombres.count (0)) # affichera 4

15. Trouver l’indice d’un élément de la liste

nombres = [0, 1, 2, 0, 3, 0, 4, 0, 5]

print (nombres.index (3)) # affichera la position 4 car le comptage commence à 0.

Si l’élément n’est pas dans la liste, une erreur se produira. Un programme robuste vérifiera d’abord si l’élément est bien dans la liste avant d’en demander son indice (utilisation d’un if : et d’un else:).

16. Trier une liste

nombres = [0, 1, 2, 0, 3, 0, 4, 0, 5]

nombres.sort()

print (nombres) # affichera [0, 0, 0, 0, 1, 2, 3, 4, 5]

17. Inverser une liste

nombres.reverse()

print (nombres) # affichera [5, 4, 3, 2, 1, 0, 0, 0, 0]

18. Copie d’une liste

liste_un = [‘adieu’, ‘veaux’, ‘vaches’, ‘cochons’, ‘couvées’]

liste_deux = liste_un.copy() # créera liste_deux contenant la copie de liste_un.

19. Qu’est-ce qu’un « tuple » ?

Un ***tuple*** est une liste qu’on ne pourra plus modifier par la suite. Voir détails dans le livre gratuit sur ce site fadagogo.com ou sur l’Internet.