

Formation Python : AJC Classroom

TP1 : Initiation vers le langage de programmation Python : Chapitre 3 (Listes)

(1) Créer une liste « test » et tester `type(test)`, `len(test)`, `test[i]`, `test[:j]`, `test[i:]`, `test[i:j]` où *i*, *j* sont des entiers relatifs.

(2) Modifier le *i*ème terme de la liste test.

(3) Concaténer deux listes, reproduire 20 fois la liste `L=[1,2,3]`.

(4) Appliquer chacune des opérations suivantes sur plusieurs exemples de votre choix :

`L = [16, 'mot', ['3', 'ab']]` : Crée une liste d'éléments mixtes

`L[i]` : Retourne le *i*ème élément de la liste.

`L[i][j]` : Retourne un élément d'une liste multidimensionnelle

`L[i:j]` : Retourne la tranche d'éléments jusqu'au *j*ème élément

`len(L)` : Retourne le nombre d'éléments de la liste

`L + l` : Concaténation de deux listes

`3 * L` : Concaténation multiple de la même liste

`x in L` : Opération logique retourne `true` si *x* est un élément de la liste, sinon `false`.

`L.append(x)` : ajoute l'élément *x* à la fin de la liste

`del L[j]` : Élimine l'élément *j* de la liste

`L[i:j]=[]` : Élimine la tranche *i:j* de la liste

`L.insert(2,x)` : insère *x* en 2ème position

`L.sort()` : Ordonne les éléments en ordre ascendant

`L.index(x)` : Retourne la position du premier élément de valeur *x*

`L.reverse()` : Réorganise les éléments en commençant par la fin

`L[i]='ceci'` : Assigne une valeur à l'élément *i*

`L[i:j]=[2,3,7]` : Assigne une valeur à la tranche *i:j*

`range(n)` : Crée la liste de *n* éléments `[0,1,2, ..., n-1]` pour l'acher `list(range())`.

`chaine.split()` : décompose une chaîne contenant des espaces en une liste de sous-chaînes.

`'X'.join(liste)` : rassemble les éléments de liste en les séparant par la chaîne *X*.

La fonction `list()` Transforme une chaîne de caractère en liste.

(5) Pour plus d'aide : `help(list)`

Formation Python : AJC Classroom

(6) Tester `range(n)`, `range(n,m)`, `range(n,m,s)` où `m`, `n` et `s` sont des entiers.

(7) Afficher une liste créée par `range()`.

(8) Constituez une liste semaine contenant les 7 jours de la semaine.

- À partir de cette liste, comment récupérez-vous seulement les 5 premiers jours de la semaine d'une part, et ceux du week-end d'autre part ? Utilisez pour cela l'indiciage.
- Cherchez un autre moyen pour arriver au même résultat (en utilisant un autre indiciage).
- Trouvez deux manières pour accéder au dernier jour de la semaine.
- Inversez les jours de la semaine en une commande.

(9) Créez 4 listes hiver, printemps, été et automne contenant les mois correspondants à ces saisons. Créer ensuite une liste saisons contenant les listes hiver, printemps, été et automne. Prévoyez ce que renvoient les instructions suivantes, puis vérifiez-le dans l'interpréteur :

- `saisons[2]`
- `saisons[1][0]`
- `saisons[1:2]`
- `saisons[:][1]`. Comment expliquez-vous ce dernier résultat ?

(10) Affichez la table de multiplication par 9 en une seule commande avec les instructions `range()` et `list()`.

(11) Répondez à la question suivante en une seule commande. Combien y a-t-il de nombres pairs dans l'intervalle `[2, 10000]` inclus ?