

Python (cours 2)

1) Introduction

Structures : Une structure de données est un type de variable dont le rôle est de stocker un ensemble d'éléments.

Utilisation : Manipuler et traiter des ensembles regroupés sous la même variable.

Critères de construction d'une structure :

- Taille
- Indexation
- Doublons
- Accès concurrent
- Méthodes de parcours
- Méthodes de tri
- Organisation en mémoire

Taille : Chaque structure peut avoir une taille fixe ou dynamique en mémoire

Taille fixe : Déterminée à l'initialisation

Taille dynamique : Adaptée aux ajouts. Taille non connue à l'avance, modifiée à l'exécution

Indexation : Pointeur sur chaque élément. Ordre (ou non-ordre) de classement en mémoire

Indexation numérique : Chaque élément est associé à une clé numérique unique

Clé/valeurs : Chaque clé peut être d'un type quelconque

Accès concurrent : Dans le cas d'un accès parallèle, comment sont gérées les priorités d'accès ?

Ordres de priorité —> Cadencement —> Sémaphores et mécanismes d'attente —> Déterminisme

2) Type majeurs

Listes :

```
lst = ["a", "b", "c", "d", "e", "f"]  
print lst[0] —> print "a"  
print lst[1:3] —> print "bcd"
```

Ajout d'un élément dans une liste :

```
lst = ["a", "b", "c"]
```

En tête : —> `lst = ["new"] + lst`

En queue : —> `lst.append("new")` // ["a", "b", "c", "new"]
 —> `lst.extend("new")` // ["a", "b", "c", ["new"]]

Indexé : —> `lst.insert("new", 2)` // ["a", "b", "new", "c"]

Suppression dans une liste :

```
lst.pop(2)       // ["a", "b"]  
lst.remove("b")   // ["a", "c"]
```

Range : Range permet de définir un intervalle de valeurs. Il renvoie une liste. Boucles et parcours.

```
range(1, 10, 2)       // 1, 3, 5, 7, 9
```

3) Dictionnaires

Un dictionnaire est un ensemble de paires clé/valeur. Chaque association est unique. Tableaux associatifs.

Construction :

```
dico = {"toto":15, "titi":20, "tata":30 }  
print dico["toto"]    //print 1  
type(dico)            // print dict  
"toto" in dico         //return true  
"theo" in dico         //return false  
list(dico.keys())     //[ "toto", "titi", "tata"]
```

Construire un dictionnaires :

La fonction dict permet de convertir un ensemble de dictionnaire. Conversion de listes, de tuples.

```
dico = dict([("bob",15),("alice",23)])  
dico = dict({"bob":15,"alice":23})
```

Ajout :

```
dico["bill"] = 15
```

Supprimer :

```
del dico["bill"]  
del dico[3]  
del dico  
dico.clear()
```

4) Autres structures

Piles :

```
lst.append("d")    //Ajout d à la fin d'une liste  
lst.pop()          //Supprimer la dernière valeur d'une liste  
lst.popleft()      //Supprimer la première valeur d'une liste
```

Tri :

```
a[1, 5, 3, 2, 4]  
sorted(a)         // [1, 2, 3, 4, 5]  
a.sort()          //Auto assignation
```