

1 Définition

Définition 1 : dictionnaire

Un dictionnaire est une structure de données. La différence entre un dictionnaire et un tableau est la suivante :

dans un tableau `tab`, une **valeur** se situe à la case correspondant à un **indice**. Un indice est un entier, compris entre 0 et la longueur du tableau moins un. On appelle alors `tab[i]`.

dans un dictionnaire, une **valeur** se situe à la case correspondant à une **clé**. Une clé peut être de n'importe quel type. On appelle aussi `tab[cle]`.

Schématiquement, un tableau statique et un dictionnaire peuvent se représenter ainsi :

i	0	1	2	3	4	5	6	7	8	9
tab[i]	1	2	3	5	8	13	21	34	55	89

clés	"bob"	"zoe"	"ben"	"ada"	"moe"	"lea"	"kit"	"bea"	"sam"	"pam"
valeurs	14	12	7	19	6	13	10	6	17	9

1.1 Syntaxe des dictionnaires

initialiser un dictionnaire vide	<code>dico = {}</code>
demander l'élément lié à la clé <code>key</code>	<code>dico[key]</code>
rajouter une nouvelle case	<code>dico[nv_cle] = nv_valeur</code>
changer une case	<code>dico[cle] = nv_valeur</code>
supprimer une case	<code>del dico[cle]</code>

Remarque : la syntaxe pour créer une nouvelle case et modifier une case préexistante est la même.

2 Parcours

Une différence importante entre dictionnaires et tableau se situe dans l'interaction avec les boucles « pour ».

```
dico = {'vrai': 'true', 'balai': 'broom', 'ordinateur': 'computer', 'rouge': 'red'}
for a in dico:
    print(a)
```

Le script successivement affiche successivement 'vrai', 'balai', 'ordinateur' et 'rouge' : dans une boucle `for`, on procède à un parcours **par les clés** du dictionnaire. Malgré les schémas présentés ici, un dictionnaire ne dispose pas d'un ordre inhérent : il faut plutôt le considérer comme un ensemble non ordonné. Ainsi, trier un dictionnaire n'a pas de sens.