

Les tableaux et les dictionnaires.

Pour pouvoir manipuler les données en python, nous allons avoir besoin de deux structures de données: les tableaux et les dictionnaires.

I. Les tableaux.

Un tableau est un ensemble ordonné éléments.

En python, c'est un objet de type `list`.

a. Construction.

Pour créer un tableau, nous utiliserons la syntaxe suivante:

```
mon_tableau=["soleil","pluie","vent","neige"]
```

Les crochets permettent de délimiter le tableau et les virgules de séparer chacun des éléments.

Pour créer un tableau vide, on utilise la syntaxe suivante:

```
mon_tableau_vide=[]
```

b. Accessibilité et modification d'un élément du tableau.

Chacun des éléments du tableau est repéré grâce à sa place dans le tableau.

La numérotation commence à 0.

Pour accéder à un élément, on indique le nom du tableau ainsi que la place de élément souhaité entre crochets.

Pour modifier un élément du tableau, il suffit de lui affecter une nouvelle valeur.

```
>>> mon_tableau=["soleil","pluie","vent","neige"]
>>> mon_tableau[0]
'soleil'
>>> mon_tableau[2]
'vent'
>>> mon_tableau[3]="grêle"
>>> print(mon_tableau)
["soleil","pluie","vent","grêle"]
```

c. La longueur d'un tableau.

La fonction `len` permet d'obtenir la longueur d'un tableau.

```
>>> mon_tableau=["soleil","pluie","vent","neige"]
>>> len(mon_tableau)
4
```

d. Ajouter un élément au tableau.

La méthode `append` permet d'ajouter un élément au tableau.

```
>>> mon_tableau=["soleil","pluie","vent","neige"]
>>> mon_tableau.append("arc en ciel")
>>> print(mon_tableau)
["soleil","pluie","vent","neige","arc en ciel"]
```

e. Supprimer un élément d'un tableau.

Il existe deux manières de supprimer un élément :

- soit en avec la commande `del` avec l'index de l'élément :

```
>>> mon_tableau=["soleil","pluie","vent","neige"]
>>> del mon_tableau[1]
>>> print(mon_tableau)
["soleil","vent","neige"]
```

- soit avec la méthode `remove` (avec la valeur de l'élément).

```
>>> mon_tableau=["soleil","pluie","vent","neige"]
>>> mon_tableau.remove("vent")
>>> print(mon_tableau)
["soleil","pluie","neige"]
```

f. Appartenance.

L'appartenance à un tableau se teste avec l'opérateur `in`.

```
>>> mon_tableau=["soleil","pluie","vent","neige"]
>>> "vent" in mon_tableau
True
>>> "orage" in mon_tableau
False
```

g. Parcours.

Pour parcourir les éléments d'un tableau, on utilisera une boucle `for`.

```
>>> mon_tableau=["soleil","pluie","vent","neige"]
>>> for i in range(len(mon_tableau)):
    print(mon_tableau[i])
"soleil"
"pluie"
"vent"
"neige"
```

II. Les dictionnaires.

Un dictionnaire ressemble à un tableau. La principale différence est que les indices ne sont pas obligatoirement des nombres entiers et peuvent être de n'importe quel type.

Ces indices ne sont pas ordonnés et s'appellent des clés. A chaque clé correspond une valeur.

En Python, c'est un objet de type `dic`.

a. Construction.

Pour créer un dictionnaire, nous utiliserons la syntaxe suivante :

```
Harry={"Nom":"Potter","Prenom":"Harry","Age":17,"Animal":"Hedwige"}
```

Un dictionnaire est donc un ensemble de couples clé-valeur. Pour créer un dictionnaire, on écrit entre des accolades les couples séparés par des virgules, chaque couple étant composé d'une clé et de la valeur correspondante séparées par deux points.

Pour créer un dictionnaire vide, on utilise la syntaxe suivante :

```
mon_dictionnaire_vide = {}
```

b. Accessibilité et modification d'un élément du dictionnaire.

Pour accéder à un élément, on utilise la clé. De même que pour un tableau, on peut modifier la valeur d'un élément du dictionnaire.

```
>>> Harry={"Nom":"Potter","Prenom":"Harry","Age":17,"Animal":"Hedwige"}
>>> Harry["Nom"]
'Potter'
>>> Harry["Age"]=15
>>> print(Harry)
{"Nom":"Potter","Prenom":"Harry","Age":15,"Animal":"Hedwige"}
```

Attention, si la clé n'est pas dans le dictionnaire, on obtient alors un message d'erreur.

c. Longueur d'un dictionnaire.

La fonction `len` permet d'obtenir la longueur d'un dictionnaire.

```
>>> Harry={"Nom":"Potter","Prenom":"Harry","Age":17,"Animal":"Hedwige"}
>>> len(Harry)
4
```

d. Ajouter un élément au dictionnaire

Pour ajouter un élément dans un dictionnaire, il suffit d'affecter une valeur à une nouvelle clé.

```
>>> Harry={"Nom":"Potter","Prenom":"Harry","Age":17,"Animal":"Hedwige"}
>>> Harry["Taille"]=1.7
>>> print(Harry)
{'Nom':'Potter','Prenom':'Harry','Age':17,'Animal':'Hedwige','Taille':1.7}
```

e. Supprimer un élément d'un dictionnaire.

Pour supprimer un élément d'un dictionnaire, on utilise la méthode `del` en utilisant la clé.

```
>>> Harry={"Nom":"Potter","Prenom":"Harry","Age":17,"Animal":"Hedwige"}
>>> del Harry["Age"]
>>> print(Harry)
{'Nom':'Potter','Prenom':'Harry','Animal':'Hedwige'}
```

g. Parcours.

Il est possible de parcourir les clés d'un dictionnaire en utilisant la méthode `keys`.

```
>>> Harry={"Nom":"Potter","Prenom":"Harry","Age":17,"Animal":"Hedwige"}
>>> for cle in Harry.keys():
    print(cle)
'Nom'
'Prenom'
'Age'
'Animal'
```

Il est possible de parcourir les valeurs d'un dictionnaire en utilisant la méthode `values`.

```
>>> Harry={"Nom":"Potter","Prenom":"Harry","Age":17,"Animal":"Hedwige"}
>>> for elt in Harry.values():
    print(elt)
'Potter'
'Harry'
17
'Hedwige'
```

III. Mise en pratique.

Exercice 1:

Écrire un programme qui demande à l'utilisateur des nombres, puis qui les range dans un tableau.

Afficher ensuite l'ensemble des nombres contenus dans le tableau.

Exercice 2:

Écrire une fonction qui prend en paramètres deux tableaux `tab1` et `tab2` de même longueur et qui renvoie un tableau unique contenant les éléments de `tab1` et de `tab2` alternativement.

Exercice 3: Soient les dictionnaires suivants:

```
magasin_A={"Pomme":10,"Poire":15,"Fraise":3,"Banane":7}  
magasin_B={"Pomme":7,"Poire":4,"Fraise":8,"Banane":20}
```

Écrire une fonction qui prend en paramètres ces deux dictionnaires et qui retourne un dictionnaire ayant les mêmes clés que celles du magasin_A et du magasin_B et qui a pour valeur le nom du magasin pour lequel la valeur est la plus grande.

Exercice 4 : Écrire une fonction stat qui prend en paramètres un texte (type str) et qui renvoie un dictionnaire dont les clés sont les différentes lettres du texte et les valeurs le nombre d'occurrences de chaque lettre. On suppose le texte écrit en lettres capitales non accentuées.

Attention, le texte contient des caractères de ponctuation et des espaces qu'il ne faut pas comptabiliser.