

# Listes

# Définition

- Une collection d'éléments séparés par des virgules, l'ensemble étant enfermé dans des crochets

```
>>> jour = ['lundi', 'mardi', 'mercredi', 1800, 20.357, 'jeudi', 'vendredi']  
>>> print(jour)  
['lundi', 'mardi', 'mercredi', 1800, 20.357, 'jeudi', 'vendredi']
```

- Les éléments individuels qui constituent une liste peuvent être de types variés
- On accède aux éléments d'une liste par son *index* dont la numération comment à partir du 0 (et non de 1)

```
>>> jour = ['lundi', 'mardi', 'mercredi', 1800, 20.357, 'jeudi', 'vendredi']  
>>> print(jour[2])  
mercredi  
>>> print(jour[4])  
20.357
```

# Définition d'une liste – accès à ses éléments

```
>>> nombres = [5, 38, 10, 25]
>>> mots = ["jambon", "fromage", "confiture", "chocolat"]
>>> stuff = [5000, "Brigitte", 3.1416, ["Albert", "René", 1947]]
>>> print(nombres[2])
10
>>> print(nombres[1:3])
[38, 10]
>>> print(nombres[2:3])
[10]
>>> print(nombres[2:])
[10, 25]
>>> print(nombres[:2])
[5, 38]
>>> print(nombres[-1])
25
>>> print(nombres[-2])
10
```

# Opérations sur les listes

- Changer des éléments d'une liste (contrairement aux chaînes de car. qui constituent des données non-modifiables)

```
>>> print(jour)
['lundi', 'mardi', 'mercredi', 1800, 20.357, 'jeudi', 'vendredi']
>>> jour[3] = jour[3] + 47
>>> print(jour)
['lundi', 'mardi', 'mercredi', 1847, 20.357, 'jeudi', 'vendredi']
```

- Remplacer des éléments d'une liste par d'autres

```
>>> jour[3] = 'Juillet'
>>> print(jour)
['lundi', 'mardi', 'mercredi', 'Juillet', 20.357, 'jeudi', 'vendredi']
```

- Déterminer la longueur d'une liste : fonction *len()*

```
>>> print(len(jour))
7
```

- Supprimer d'une liste un élément : fonction *del()*

```
>>> del(jour[4])
>>> print(jour)
['lundi', 'mardi', 'mercredi', 'juillet', 'jeudi', 'vendredi']
```

# Opérations sur les listes (suite)

- Ajouter un élément à une liste :
  - utilisation d'une méthode *append* à l'objet *jour* avec l'argument 'samedi'

```
>>> jour.append('samedi')
>>> print(jour)
['lundi', 'mardi', 'mercredi', 'juillet', 'jeudi', 'vendredi', 'samedi']
>>>
```

# Appartenance d'un élément à une séquence : l'instruction *in* utilisée seule

```
n = 5
premiers = [1, 2, 3, 5, 7, 11, 13, 17]
if n in premiers:
    print(n, "fait partie de notre liste de nombres premiers")
```

# Les listes sont des objets

```
>>> nombres = [17, 38, 10, 25, 72]
>>> nombres.sort()                # trier la liste
>>> nombres
[10, 17, 25, 38, 72]

>>> nombres.append(12)            # ajouter un élément à la fin
>>> nombres
[10, 17, 25, 38, 72, 12]

>>> nombres.reverse()            # inverser l'ordre des éléments
>>> nombres
[12, 72, 38, 25, 17, 10]

>>> nombres.index(17)            # retrouver l'index d'un élément
4

>>> nombres.remove(38)           # enlever (effacer) un élément
>>> nombres
[12, 72, 25, 17, 10]

>>> del nombres[2]
>>> nombres
[12, 72, 17, 10]
>>> del nombres[1:3]
>>> nombres
[12, 10]
```

## Insertion d'un ou plusieurs éléments n'importe où dans une liste

```
>>> mots = ['jambon', 'fromage', 'confiture', 'chocolat']
>>> mots[2:2] = ["miel"]
>>> mots
['jambon', 'fromage', 'miel', 'confiture', 'chocolat']

>>> mots[5:5] = ['saucisson', 'ketchup']
>>> mots
['jambon', 'fromage', 'miel', 'confiture', 'chocolat', 'saucisson', 'ketchup']
```

## Suppression / remplacement d'éléments

```
>>> mots[2:5] = [] # [] désigne une liste vide
>>> mots
['jambon', 'fromage', 'saucisson', 'ketchup']

>>> mots[1:3] = ['salade']
>>> mots
['jambon', 'salade', 'ketchup']

>>> mots[1:] = ['mayonnaise', 'poulet', 'tomate']
>>> mots
['jambon', 'mayonnaise', 'poulet', 'tomate']
```



# La fonction *range()*

- renvoie une séquence d'entiers qu'on peut utiliser directement ou convertir en une liste avec la fonction **list()**

```
>>> list(range(10))  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
>>> list(range(5,13))  
[5, 6, 7, 8, 9, 10, 11, 12]  
>>> list(range(3,16,3))  
[3, 6, 9, 12, 15]
```

=> *FROM, TO et STEP*

```
>>> list(range(10, -10, -3))  
[10, 7, 4, 1, -2, -5, -8]
```

# Parcours d'une liste

```
>>> prov = ['La', 'raison', 'du', 'plus', 'fort', 'est', 'toujours', 'la', 'meilleure']
>>> for mot in prov:
...     print(mot, end = ' ')
...
La raison du plus fort est toujours la meilleure
```

```
>>> for n in range(10, 18, 3):
...     print(n, n*2, n*3)
...
10 100 1000
13 169 2197
16 256 4096
```

```
fable = ['Maître', 'Corbeau', 'sur', 'un', 'arbre', 'perché']
for index in range(len(fable)):
    print(index, fable[index])
```

L'exécution de ce script donne le résultat :

```
0 Maître
1 Corbeau
2 sur
3 un
4 arbre
5 perché
```

# Opérations sur les listes

## + (concatenation)

```
>>> fruits = ['orange', 'citron']
>>> legumes = ['poireau', 'oignon', 'tomate']
>>> fruits + legumes
['orange', 'citron', 'poireau', 'oignon', 'tomate']
>>> fruits * 3
['orange', 'citron', 'orange', 'citron', 'orange', 'citron']
```

## \* (multiplication)

```
>>> sept_zeros = [0]*7
>>> sept_zeros
[0, 0, 0, 0, 0, 0, 0]
```

## Test d'appartenance

```
>>> v = 'tomate'
>>> if v in legumes:
...     print('OK')
...
OK
```

# Fonctions

- la fonction *sorted* permet de parcourir les éléments d'une liste de façon ordonnée.

```
li = [4, 5, 3, -6, 7, 9]
```

```
for n in sorted(li): # on parcourt la liste li de façon triée  
    print(n)
```

```
print(li) # la liste li n'est pas triée
```

```
-----
```

```
li.sort() # la liste est triée
```

```
for n in li:  
    print(n)
```

# Fonctions

- La fonction *enumerate* permet d'éviter l'emploi de la fonction *range* lorsqu'on souhaite parcourir une liste alors que l'indice et l'élément sont nécessaires.

```
li = [4, 5, 3, -6, 7, 9]
```

```
for i in range(0, len(li)):
    print(i, li[i])
```

-----

```
for i, v in enumerate(li):
    print(i, v)
```

# Dernière remarque

- Le niveau d'indentation n'a pas d'importance : l'interpréteur détecte la fin de l'instruction la ou la paire syntaxique est refermée.

```
couleurs = ['noir', 'brun', 'rouge',  
            'orange', 'jaune', 'vert',  
            'bleu', 'violet', 'gris', 'blanc']
```

# Travail à faire

Soient les listes suivantes :

```
t1 = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
t2 = ['Janvier', 'Février', 'Mars', 'Avril', 'Mai', 'Juin',
      'Juillet', 'Août', 'Septembre', 'Octobre', 'Novembre', 'Décembre']
```

Écrivez un petit programme qui crée une nouvelle liste **t3**. Celle-ci devra contenir tous les éléments des deux listes en les alternant, de telle manière que chaque nom de mois soit suivi du nombre de jours correspondant :

```
['Janvier', 31, 'Février', 28, 'Mars', 31, etc...].
```

---

Rechercher le plus grand élément dans une liste de nombres donnée.

Input : **[13, 79, 1, 30, 456, 16]**

Output : **Le plus grand élément de la liste a la valeur de 456.**

A partir d'une liste de nombres donnée, créer deux nouvelles listes : avec les nombres divisibles par 4 et les autres.

Input : **[13, 79, 1, 30, 456, 16]**

Output :

**liste des nombres divisibles par 4 : [456, 16]**

**liste des nombres non divisible par 4 : [13, 79, 1, 30]**