

# Structures de données

## Les listes - Rappels

- Une liste est une structure de données qui contient une série de valeurs.
- Types de valeurs hétérogènes autorisés.
- Une liste est déclarée par une série de valeurs encadrée par des crochets et séparées par des virgules.
- Accès aux éléments par leur indice (position) avec l'opérateur `[]`. Les indices d'une liste à  $n$  éléments vont de 0 à  $n-1$ .
- On peut accéder aux éléments avec des indices négatifs, l'indice est alors relatif à la fin de la séquence : `l[-1]` est équivalent à `l[len(l)-1]`.

# Structures de données

## Opérations sur les listes

- Opérateur + : concaténation.
- Opérateur \* : duplication.
- $x \text{ in } l$  : True si un élément de  $l$  est égal à  $x$ , False sinon.
- $l[i:j]$  : Sous-liste de  $l$  à partir des éléments compris entre les indices  $i$  et  $j$  ( $j$  non inclus).
- $\text{len}(l)$  : Nombre d'éléments de la liste  $l$ .
- $\text{min}(l)$  /  $\text{max}(l)$  : Valeur minimale / maximale de  $l$ .
- $l.\text{index}(x[,i [, j]])$  : indice de la première occurrence d'un élément égal à  $x$  dans  $l$  (éventuellement à partir de l'indice  $i$  et jusqu'à l'indice  $j$ ).
- $l.\text{count}(x)$  : Nombre d'occurrences de  $x$  dans  $l$ .

# Structures de données

## Opérations sur les listes

- `append ()` : ajout d'un élément à la fin de la liste.
- `l [i:j] = l2` : les éléments d'indices compris dans l'intervalle `[i, j]` sont remplacé par les éléments de la liste `l2`.
- `clear ()` : supprime tous les éléments de la liste.
- `l.remove (x)` : supprime de `l` le premier élément dont la valeur est égale à `x`.
- `l.insert(i, x)` : insère l'élément `x` à l'indice `i`.
- <https://docs.python.org/3/library/stdtypes.html#list>

# Structures de données

## Compréhension de listes

- Syntaxe spéciale pour effectuer des opérations sur les éléments d'une liste.
- `[expr (i) for i in l ]` : création d'une nouvelle liste à partir des éléments d'une liste existante l
- `[expr(i) for i in l if predicat(i)]` : au passage on filtre certains éléments de la liste de départ.

```
l1 = [1, 2, 3, 4]
l2 = [i*3 for i in l1]
l3 = [i*2 for i in l2 if (i%2==0)]
print (l2) # [3, 6, 9, 12]
print (l3) # [12, 24]
```

# Structures de données

## Dictionnaires

- Un dictionnaire (ou tableau associatif) permet de stocker des associations clef-valeur.
- notation : entre accolades, les entrées sont séparées par des virgules, et les clefs séparées des valeurs par des :
- On accède à une valeur avec sa clef entre crochets

```
etu = { 'nom': 'Sy',  
        'Prénom': 'Omar',  
        'Num' : '234567'  
        'notes' : {  
            'Methodo': 12  
            'Intro_moteurs_jeux': 14  
            'Prog_fonctionnelle': 11  
        }  
}  
print (etu['nom']) # Sy  
print (etu['notes']['Methodo']) # 12
```