

## Trancher des listes

- Tu connais déjà `maliste[a:b]` qui renvoie la sous-liste des éléments du rang  $a$  au rang  $b - 1$ .
- `maliste[a:]` renvoie la liste des éléments du rang  $a$  jusqu'à la fin.
- `maliste[:b]` renvoie la liste des éléments du début jusqu'au rang  $b - 1$ .
- `maliste[-1]` renvoie le dernier élément, `maliste[-2]` renvoie l'avant-dernier élément,...

**Exercice.**

7	2	4	5	3	10	9	8	3
rang : 0	1	2	3	4	5	6	7	8

Avec `maliste = [7, 2, 4, 5, 3, 10, 9, 8, 3]`, que renvoient les instructions suivantes ?

- `maliste[3:5]`
- `maliste[4:]`
- `maliste[:6]`
- `maliste[-1]`

## Trouver le rang d'un élément

`liste.index(element)` renvoie la première position à laquelle l'élément a été trouvé. Exemple : avec `liste = [12, 30, 5, 9, 5, 21]`, `liste.index(5)` renvoie 2.

Si on souhaite juste savoir si un élément appartient à une liste, alors l'instruction :

`element in liste`

renvoie `True` ou `False`. Exemple : avec `liste = [12, 30, 5, 9, 5, 21]`, «`9 in liste`» est vrai, alors que «`8 in liste`» est faux.

# Liste par compréhension

$$E = \{0, 2, 4, 6, 8, 10\}$$

$$E = \{x \in \mathbb{N} \mid x \leq 10 \text{ et } x \text{ est pair}\}$$

## Liste par compréhension Python

- Partons d'une liste, par exemple `maliste = [1, 2, 3, 4, 5, 6, 7, 6, 5, 4, 3, 2, 1]`.
- La commande `liste_doubles = [ 2*x for x in maliste ]` renvoie une liste qui contient les doubles des éléments de la liste `maliste`. C'est donc la liste `[2, 4, 6, 8, ...]`.
- La commande `liste_carres = [ x**2 for x in maliste ]` renvoie la liste des carrés des éléments de la liste initiale. C'est donc la liste `[1, 4, 9, 16, ...]`.
- La commande `liste_partielle = [x for x in maliste if x > 2]` extrait la liste composée des seuls éléments strictement supérieurs à 2. C'est donc la liste `[3, 4, 5, 6, 7, 6, 5, 4, 3]`.

## Liste de listes

Une liste peut contenir d'autres listes, par exemple :

```
maliste = [ ["Harry", "Hermione", "Ron"], [101,103] ]
```

contient deux listes.

Listes qui contiennent des listes d'entiers :des *tableaux*.

Par exemple :

```
tableau = [ [2,14,5], [3,5,7], [15,19,4], [8,6,5] ]
```

2	14	5
3	5	7
15	19	4
8	6	5

Alors `tableau[i]` renvoie la sous-liste de rang  $i$ , alors que `tableau[i][j]` renvoie l'entier situé au rang  $j$  dans la sous-liste de rang  $i$ . Par exemple :

- `tableau[0]` renvoie la liste `[2,14,5]`,
- `tableau[1]` renvoie la liste `[3,5,7]`,
- `tableau[0][0]` renvoie l'entier 2,
- `tableau[0][1]` renvoie l'entier 14,
- `tableau[2][1]` renvoie l'entier 19.