

Manipulation de Listes Python : Indexation et Découpage

Guide pratique pour accéder aux données et créer des sous-ensembles

```
liste = ['Liz', 1.73, 'Emma', 1.68, 'mom', 1.71, 'dad', 1.89]
```

Une introduction visuelle à la syntaxe d'accès aux données.

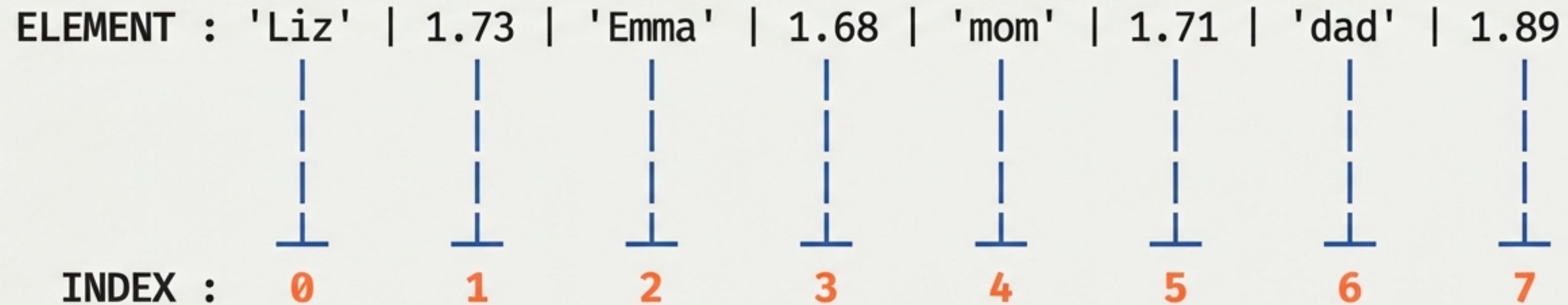
Notre objet d'étude : La liste 'famille'

'Liz'	1.73	'Emma'	1.68	'mom'	1.71	'dad'	1.89
-------	------	--------	------	-------	------	-------	------

Structure : Une liste mixte contenant des chaînes de caractères (noms) et des nombres flottants (tailles).

Objectif : Pour manipuler ces données, nous devons comprendre leur position exacte dans la liste.

Le concept de ‘Zero Indexing’



Règle Fondamentale :

Pour Python, le comptage commence toujours à zéro.

Index = Position - 1

Accéder à un élément spécifique (Float)

INDEX	:	0	1	2	3	4	5	...
ELEMENT	:	'Liz'	1.73	'Emma'	1.68	'mom'	1.71	...

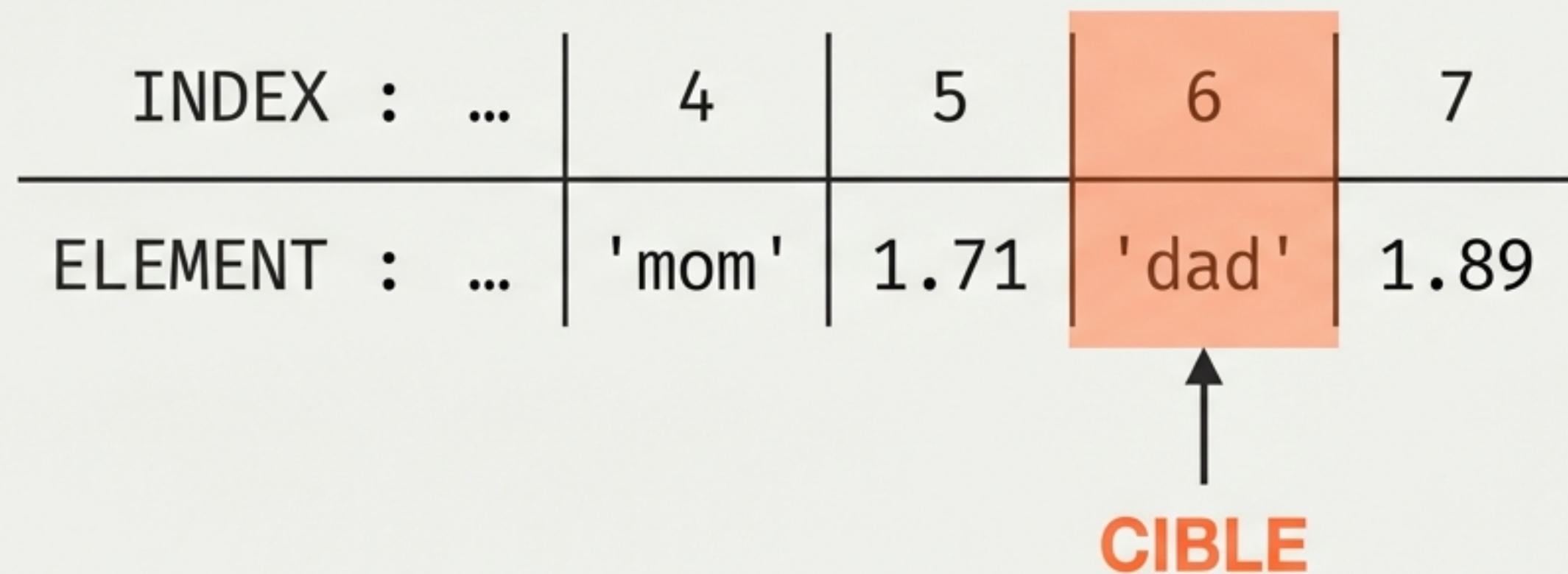
↑
CIBLE

Code :
`print(liste[3])`

Résultat :
1.68

Scénario : Nous voulons la taille d'Emma (4ème élément).
Logique : 4ème élément -> Index 3.

Accéder à un élément spécifique (String)



Code :
print(liste[6])

Résultat :
'dad'

Scénario : Nous voulons extraire la chaîne de caractères 'dad' (7ème élément).
Logique : 7ème élément -> Index 6.

L'Indexation Négative

ELEMENT :	'Liz' 1.73 'Emma' 1.68 'mom' 1.71 'dad' 1.89
INDEX + :	0 1 2 3 4 5 6 7
INDEX - :	-8 -7 -6 -5 -4 -3 -2 -1

Python permet de compter à l'envers depuis la fin de la liste.
Le dernier élément a toujours l'index -1.

Exemple d'Indexation Négative

ELEMENT :	...	'mom'	1.71	'dad'	1.89
INDEX - :	...	-4	-3	-2	-1

↑
CIBLE

Code :

```
print(liste[-1])
```

Résultat :

```
1.89
```

Scénario : Obtenir la taille de 'dad' (dernier élément).

Note : liste[7] et liste[-1] retournent le même résultat.

Le Découpage de Liste (Slicing)

Création de sous-listes

NOUVELLE_LISTE = ANCIENNE_LISTE [DÉBUT : FIN]

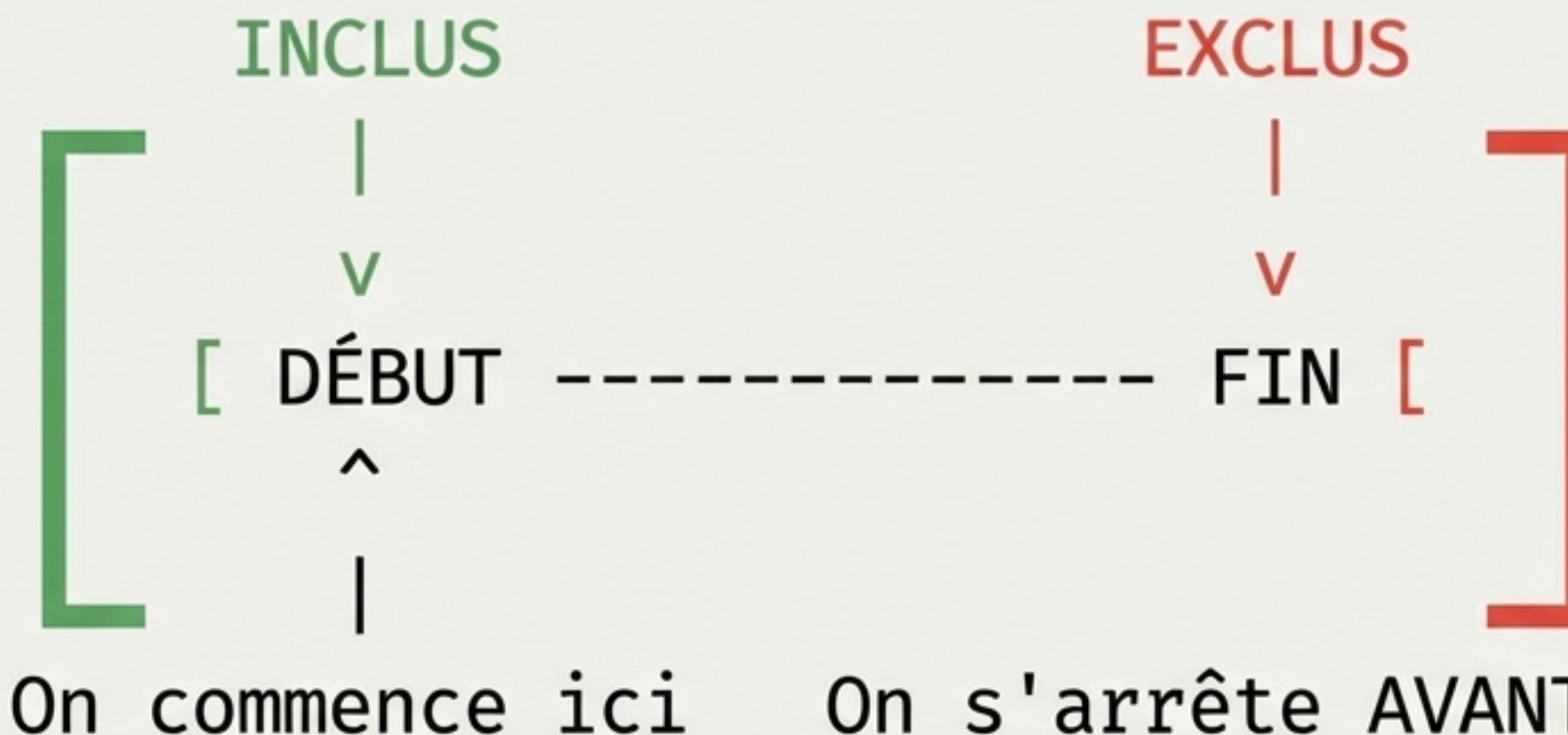


Opérateur de découpage

Le slicing permet de sélectionner une plage d'éléments.

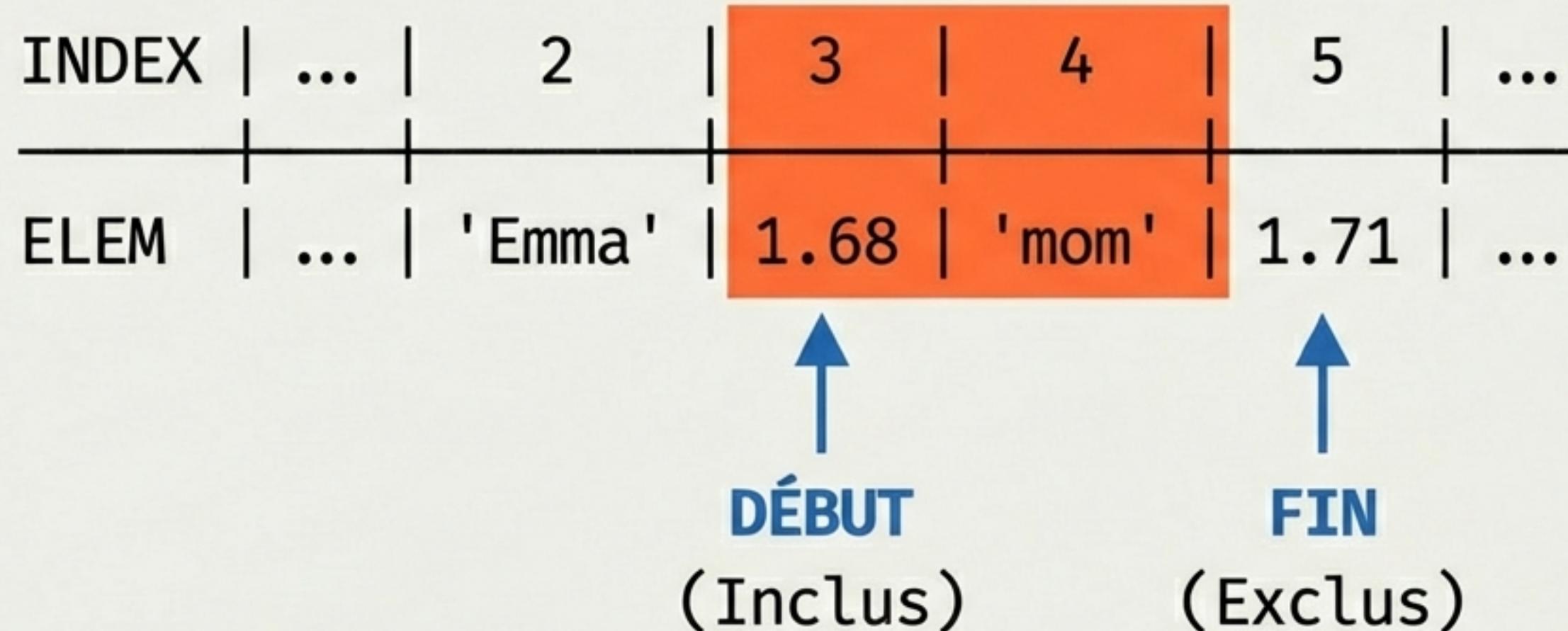
Cette opération ne modifie pas la liste originale, elle en crée une nouvelle.

La Règle d'Or du Slicing



1. Le **Début** (Start) : L'index avant les deux-points est inclus.
2. La **Fin** (End) : L'index après les deux-points est exclu.

Application : Extraire une sous-liste (1/2)

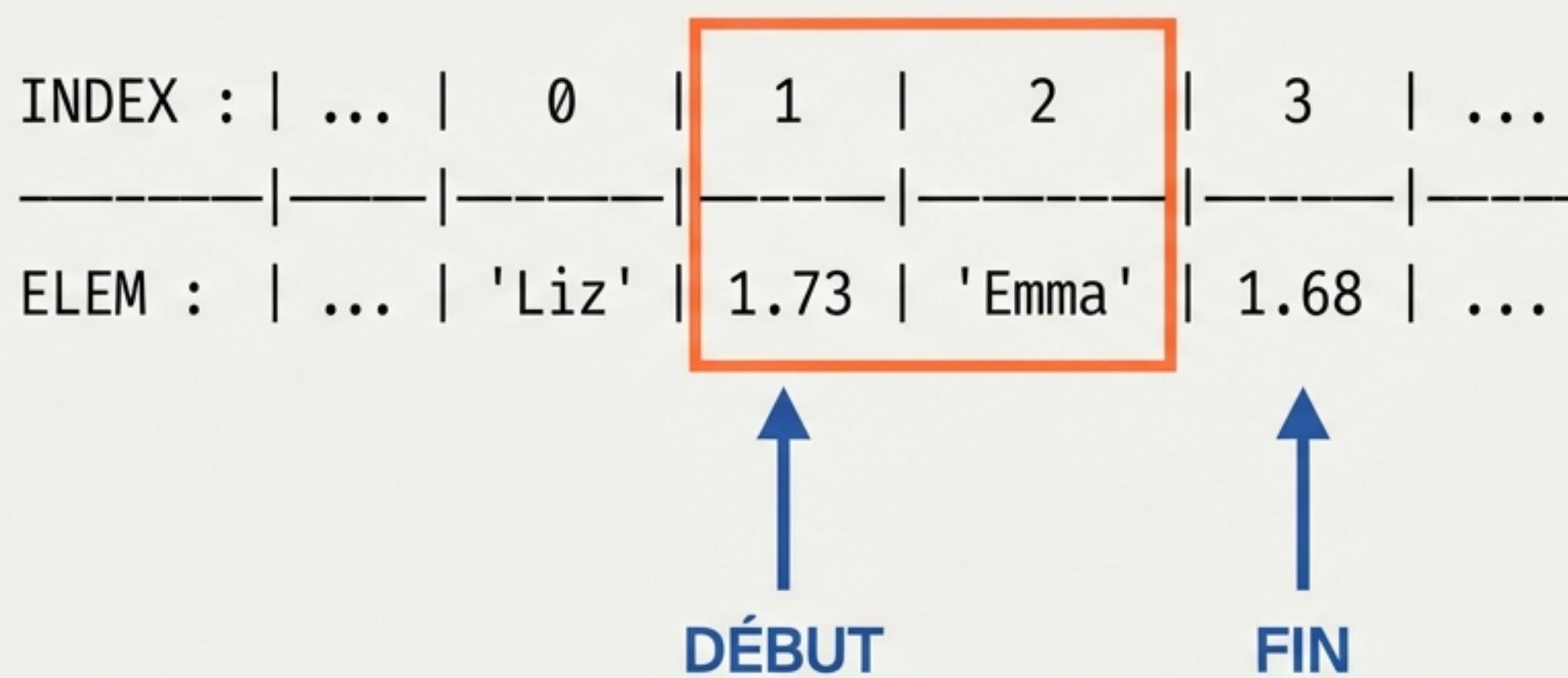


Code :
`print(liste[3:5])`

Résultat :
[1.68, 'mom']

Analyse : De l'index 3 jusqu'à (mais sans inclure) l'index 5.

Application : Extraire une sous-liste (2/2)



Code :

```
print(liste[1:3])
```

Résultat :

```
[1.73, 'Emma']
```

Objectif : Récupérer les 2ème et 3ème éléments (Index 1 et 2).

Raccourci : Le Début Implicit

[:4]

INDEX :	0	1	2	3	4	...
ELEM :	'Liz'	1.73	'Emma'	1.68	'mom'	...
					STOP	

ZONE SÉLECTIONNÉE

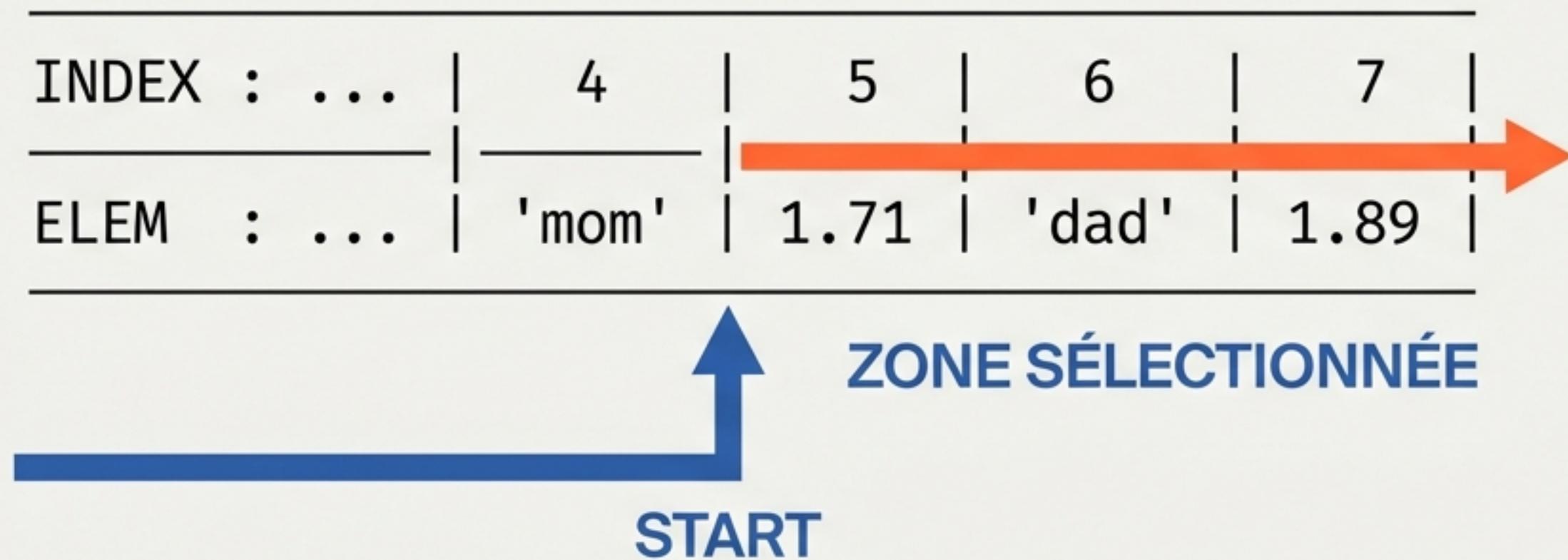
Code :
`print(liste[:4])`

Résultat :
['Liz', 1.73, 'Emma', 1.68]

Règle : Si on omet l'index de début, Python commence automatiquement à l'index 0.

Raccourci : La Fin Implicit

[5:]



Code :
print(liste[5:])

Résultat :
[1.71, 'dad', 1.89]

Règle : Si on omet l'index de fin, Python va jusqu'au tout dernier élément inclus.

Vue d'ensemble : La ‘Carte’ de la Liste

INDEX	0	1	2	3	4	5	6	7
VALEUR	'Liz'	1.73	'Emma'	1.68	'mom'	1.71	'dad'	1.89
NEG	-8	-7	-6	-5	-4	-3	-2	-1

- Index = Position - 1
- Index = Position - 1
- **Négatif** : -1 est le dernier

- **[start:end]** : Start inclus, End exclu
- **[::end]** : Du début
- **[start:]** : Jusqu'à la fin