

TP8

INTRODUCTION AUX LISTES**I Créer une liste explicitement**

Pour créer une liste, on écrit ses éléments **entre crochets**, séparés par des virgules.

[In]: `L1 = [1,2,3,4]`
`L1`

[Out]: `[1,2,3,4]`

1. Créer une liste L2 contenant les éléments 4, 7, 12, 11 et 8 dans cet ordre.

La commande **range(n,m)** utilisée dans les boucles for permet de créer des listes d'entiers consécutifs.

Attention, dans `range(n,m)` les éléments vont de n à $m - 1$.

[In]: `L3 = range(2,8) L3`

[Out]: `range(2, 8)`

Pour afficher les éléments d'un range, il faut faire une petite manipulation.

[In]: `list(L3)`

[Out]: `[2, 3, 4, 5, 6, 7]`

On peut également créer des listes d'entiers séparés d'un pas constant (suite arithmétique) avec **range(n,m,pas)**.

2. Créer la liste $L4 = [4,6,8,10,\dots,20]$ avec un range.

II Récupérer/modifier les éléments d'une liste.

Les éléments d'une liste sont numérotés avec une numérotation **commençant à zéro**. Ainsi, le premier élément est l'élément numéro 0, le second est l'élément numéro 1, etc. Pour obtenir l'élément numéro i d'une liste L , on utilise **$L[i]$** .

3. Afficher le premier élément de L3.

Il existe aussi une **numérotation négative**, en partant de la fin.

Le dernier élément est celui de numéro -1 , l'avant-dernier est celui de numéro -2 , etc.

4. Afficher le quatrième élément de L2 (le 11) de deux façons différentes.

On peut alors modifier tout élément d'une liste L en écrivant $L[i] = \text{valeur}$.

5. Modifier le 7 de L2 en un -1 . Vérifier en affichant L2.

Il est également possible d'accéder à une sous-liste d'une liste L :

- sous-liste des éléments numérotés de 0 à $m - 1$: $L[: m]$
- sous-liste des éléments numérotés de n à $m - 1$: $L[n : m]$
- sous-liste des éléments numérotés de n à la fin : $L[n :]$
- sous-liste des éléments numérotés de n à $m - 1$ avec un pas : $L[n : m : \text{pas}]$

6. Afficher la liste [3,4] à partir de L1.

III Ajouter/supprimer des éléments

Pour ajouter un élément x à la fin d'une liste L on utilise : **L.append(x)**.

Pour **concaténer** deux listes L1 et L2, on utilise **L1+L2**.

Pour répéter plusieurs fois une liste, on utilise **L*n**.

7. Ajouter un 13 à la fin de L2.

8. Concaténer L1 et L2 dans cet ordre.

9. En utilisant une concaténation, ajouter un 0 au début de L1.

10. Créer alors la liste [0,1,2,3,4,0,1,2,3,4] à partir de L1.

Pour supprimer un élément d'une liste L, on utilise `del L[i]`.

11. Supprimer le 13 dans L2.

IV Autres commandes

La longueur d'une liste L (nombre d'éléments) est obtenue avec `len(L)`.

On peut aussi tester l'appartenance d'un élément x à une liste L avec `x in L`. Ceci renvoie True si x est dans L et False sinon (booléen).

12. Afficher la longueur de L2.

13. Vérifier que 13 n'est plus dans L2.

V Créer une liste avec un for

V.1 En compréhension (suites explicites)

La commande [`element_k for k in range(n,m)`] permet de créer la liste des terme de la forme $element_k$, pour k allant de n à $m - 1$.

14. Créer la liste L4 = [4,6,8,10,...,20] avec cette méthode.

```
L4 = [..... for k in range(..., ...)]
```

V.2 Suites récurrentes

15. On considère la suite $(u_n)_{n \in \mathbb{N}}$ définie par

$$\begin{cases} u_0 = 2 \\ \forall n \in \mathbb{N}, u_{n+1} = 2u_n - n \end{cases}$$

Écrire un programme qui crée la liste $L = [u_0, u_1, u_2, \dots, u_{20}]$.

```
u = 2
L = ...
for k in range(...):
    u = .....
    L.append(...)
print(L)
```

VI Récapitulatif

Attention, la numérotation des éléments d'une liste commence à 0. Le premier élément est donc l'élément numéro 0.

	Syntaxe python
Créer une liste explicite	[1,2,3,4]
Créer une liste en compréhension	[element_k for k in range(n)]
Liste des entiers de 0 à $n - 1$	range(n)
Liste des entiers de n à $m - 1$	range(n,m)
Liste des entiers de n à $m - 1$, espacées d'un pas de p	range(n,m,p)
Élément numéro i de la liste L	L[i]
Premier élément de L	L[0]
Dernier élément de L	L[-1]
Modifier l'élément numéro i	L[i] = valeur
Supprimer l'élément numéro i	del L[i]
Sous-liste des éléments numérotés 0 à $m - 1$	L[: m]
Sous-liste des éléments numérotés n à $m - 1$	L[n : m]
Sous-liste des éléments numérotés n à la fin	L[n :]
Sous-liste des éléments numérotés n à $m - 1$ avec un pas de p	L[n : m : p]
Ajouter l'élément x à la fin de L	L.append(x)
Concaténation de deux listes	L1 + L2
Répétition de L n fois	L * n
Longueur de L	len(L)
Test d'appartenance (booléen)	x in L