

T.P. : introduction aux listes

Plan du T.P.

1	Les listes	1
1.1	Créer une liste en extension	1
1.2	Accéder aux éléments d'une liste	1
1.3	Créer une liste par ajouts successifs	2
2	Ensembles mathématiques et listes informatiques	2
3	Suites numériques	3
3.1	Suites et listes	3
3.2	Suites définies par récurrence et listes Python	3

1 Les listes

1.1 Créer une liste en extension



Définition :

Une **liste** est une **collection ordonnée d'éléments**.

Ces éléments peuvent être des nombres, des chaînes de caractères, ou encore d'autres objets.

Créons deux listes dans la console :

```
1 >>> chiffres = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
2 >>> jours = ["Lundi", "Mardi", "Mercredi", "Jeudi", "Vendredi", "Samedi", "Dimanche"]
```



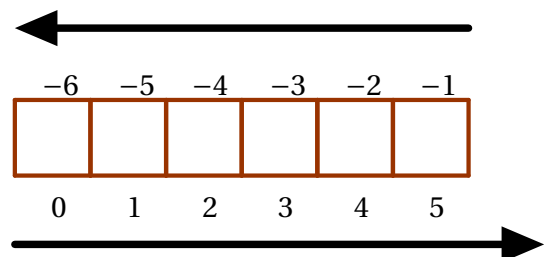
Définition :

Remarquons que nous avons créé ces listes à l'aide d'une saisie manuelle de chacun de ses éléments. Nous dirons dans ce cas que la **liste a été créée en extension**.

1.2 Accéder aux éléments d'une liste

- Saisissons dans la **console** les deux listes qui précèdent.
- Compléter le tableau suivant en vérifiant après chaque question à l'aide de la console.

Instruction à saisir dans la console	Résultat
chiffres[0]	
chiffres[2]	
	Lundi
	Jeudi
chiffres[-1]	
chiffres[-2]	
	Dimanche
	Samedi



1.3 Créer une liste par ajouts successifs



Définition :

La méthode `append` permet d'ajouter un nouvel élément en fin de liste.
Elle s'utilise selon la syntaxe : `liste.append(nouvel_element)`

Exemple :

```
1 >>> chiffres = [1, 2, 3, 4, 5, 6, 7, 8, 9]
2 >>> chiffres.append(10)
3 >>> chiffres
4 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Exercice 1 :

Saisir dans la console la liste : `chiffres = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]`.

- Ajoutez-lui maintenant l'élément 11 en fin de liste.
- Utiliser une boucle pour compléter la liste jusqu'à 30.

Exercice 2 :

- Il sera souvent utile de commencer par créer une liste vide :

```
1 liste = []
```

- Nous pourrons ensuite lui ajouter les éléments souhaités par la méthode `append`.

Utiliser ce mécanisme pour créer, par ajouts successifs, la suite des nombres pairs de 0 à 100.

2 Ensembles mathématiques et listes informatiques



Définition :

En mathématiques, un **ensemble** est une **collection d'éléments**.
Un ensemble se note entre accolades. Un ensemble n'est **pas ordonné**.

Exemples :

- l'ensemble des chiffres : $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$,
- l'ensemble des issues possibles lors d'un lancer de dé : $\Omega = \{1, 2, 3, 4, 5, 6\}$,
- l'ensemble des nombres entiers naturels : \mathbb{N} ,
- l'ensemble des entiers naturels inférieurs ou égaux à 20 : $E = \{n \in \mathbb{N} \text{ tels que } n \leq 20\}$.

Comparaison avec les listes :

- contrairement aux listes, **un ensemble n'est pas ordonné** :
 $\{1, 2, 3, 4, 5, 6\} = \{6, 1, 2, 3, 4, 5\}$ tandis que : `[1, 2, 3, 4, 5, 6] ≠ [6, 1, 2, 3, 4, 5]`
- une liste peut contenir plusieurs fois le même élément :
 $\{1, 2, 1\} = \{1, 2\}$ tandis que : `[1, 2, 1] ≠ [1, 2]`

Exercice 3 :

- a. $\{n \in \mathbb{N} \text{ tels que } n \leq 5\} = \dots\dots\dots$ b. $\{x \in \mathbb{Z} \text{ tels que } |x| \leq 3\} = \dots\dots\dots$
- c. $\{13, 13, 20, 7, 13, 6\} = \dots\dots\dots$ d. $[13, 20, 12]$ et $[13, 12, 20]$ sont égales?

3 Suites numériques

3.1 Suites et listes

Une suite u peut être perçue comme l'ensemble des termes u_n qui la composent : $\{u_n, n \in \mathbb{N}\}$.
Il sera souvent intéressant de stocker les premiers termes de la suite considérée dans une liste :

```
1 liste = [u0, u1, u2, u3, u4, u5, u6, u7, u8, u9, u10, ...]
```

Ce procédé permet un accès aisé aux termes de la suite.

Exercice 4 : accès aux termes de la suite

Soit $(u_n)_{n \in \mathbb{N}}$ une suite dont les premiers termes sont stocké dans liste :

```
1 liste = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59]
```

- a. Compléter le tableau suivant :

Instruction	Résultat
	u_0
	u_3
liste[1]	
liste[5]	
	59
liste[-2]	

- b. Quelle est la suite célèbre dont les premiers termes sont stockés dans liste?

3.2 Suites définies par récurrence et listes Python

Exercice 5 :

- a. Créer, par ajouts successifs, la listes des 20 premiers termes de la suite définie par :

$$\begin{cases} u_0 = 0 \\ u_{n+1} = u_n + 7 \quad \text{pour } n \geq 1 \end{cases}$$

Indication : pour une suite définie par récurrence, les indices négatifs simplifient beaucoup les choses.

- b. Quelle est la nature de la suite u ?
c. En termes de multiples ou de diviseurs, que dire la suite u ?

Exercice 6 :

Un particulier place l'année 0 sur un compte la somme de 10 000 €. Il perçoit chaque année des intérêts s'élevant à 3% de la somme présente sur le compte.

Objectif : déterminer l'année à partir de laquelle il disposera de 20 000 €.

- a. Quels outils autres outils que Python permettraient de traiter ce problème plus rapidement ?
Nous traiterons cependant ce problème avec Python afin de mettre en place certaines méthodes qui seront utiles par la suite.
- b. Définir une suite $(u_n)_{n \in \mathbb{N}}$ représentant les montants successifs de la somme présente sur le compte.
- c. Quelle est la nature de cette suite ?
- d. Écrire une fonction "placement(n)", renvoyant la liste des montants présents sur le compte, de l'année 0 jusqu'à l'année n.
***Indication :** on pourra penser aux indices négatifs .*
- e. Utiliser cette fonction pour déterminer à partir de quelle année cette personne disposera de 20 000€.