

# Listes

## Listes, le type list

Une *liste Python* est une collection numérotée d'éléments.

### Création

Pour créer une liste on a deux approches :

1. Directement avec ses éléments :

```
amis = ["Thiago", "Aby", "Zoé"]
```

Le type de `amis` est `list`, ses éléments sont des `str` (*string* : chaînes de caractères).

```
notes = [20, 12, 16, 12, 18]
```

Le type de `notes` est encore `list`, ses éléments sont des `int` (*integer* : entiers relatifs).

2. À l'aide d'une boucle :

```
valeurs = []  
for x in range(3):  
    valeurs.append(2 * x + 1)
```

Soit, ligne par ligne :

1. on créer une liste vide appelée `valeurs`
2. pour  $x$  de 0 inclu à 3 exclu,
3. ajouter  $2x + 5$  à la liste.

Après l'exécution du script, `valeurs` contient : `[1, 3, 5]`.

Détaillons encore, tour par tour après la ligne `valeurs.append(2 * x + 1)` :

Étape	x	2 * x + 1	valeurs
<b>Avant la boucle</b>	<i>pas défini</i>	<i>pas défini</i>	<code>[]</code>
premier tour	0	1	<code>[1]</code>
second tour	1	3	<code>[1, 3]</code>
troisième tour	2	5	<code>[1, 3, 5]</code>
<b>Après la boucle</b>	<i>pas défini</i>	<i>pas défini</i>	<code>[1, 3, 5]</code>

### Accéder à un élément.

Les éléments d'une liste sont numérotés, on peut y accéder grâce à leurs indices.

```

>>> amis = ["Thiago", "Aby", "Zoé"]
>>> amis[0]
"Thiago"
>>> amis[1]
"Aby"
>>> amis[2]
"Zoé"
>>> amis[3]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: list index out of range

```

La dernière instruction a provoqué une erreur : il n'y a pas d'élément d'indice 3 dans `amis`.

Comme on peut le voir en faisant `amis[0]`, le premier élément est numéroté par l'indice 0.

Ainsi, Thiago est bien le *premier élément de la liste* mais son indice est 0. Aby est le second élément de liste et son indice est 1. etc.

## Ajouter des éléments à une liste.

Cela se fait avec la *méthode* `append` comme on l'a vu plus haut :

```

>>> fruits = []
>>> fruits
[]
>>> fruits.append("Pomme")
>>> fruits.append("Poire")
>>> fruits
["Pomme", "Poire"]

```

Les instructions précédentes produisent le même résultat que : `fruits = ["Pomme", "Poire"]`

Ainsi qu'on l'a déjà dit, cette démarche est surtout employée dans des *boucles*.

## Longueurs, autres outils

1. La longueur d'une liste est son nombre d'éléments, elle s'obtient avec `len`.

```

>>> amis ["Pierre", "Paul"] # je me suis disputé avec les précédents
>>> len(amis)
2

```

Cette fonction `len` accepte aussi une chaîne de caractères :

```

>>> len("Bonjour")
7
>>> len("Python")
6

```

2. Présence d'un élément.

Python permet de répondre directement à la question : "Est-ce que la liste `l` contient l'élément `x` ?"

```

>>> amis = ["Pierre", "Paul"]
>>> "Thiago" in amis # c'est pas mon pote !
False
>>> "Paul" in amis # lui je l'aime bien !
True

```

## Boucles et listes

On peut écrire des boucles bornées (`for`) qui parcourent une liste.

Pour chacun de mes amis, écrire : "Tu est mon ami **prénom**".

Comme en maternelle.

```
amis = ["Gustave", "Léon"] # je change d'amis toutes les 5 minutes
for prenom in amis:
    print("Tu est mon ami", prenom)
```

Ce script va produire l'affichage :

```
Tu es mon ami Gustave
Tu es mon ami Léon
```

## Exercices

### Exercice 0

On considère mes *vrais* amis (les autres n'étaient vraiment pas cools) :

```
amis = ["John", "Sharon", "Jack", "Marlon"]
```

1. Quel est l'indice de "John" ? de "Marlon" ?
2. Comment accéder à l'ami d'indice 2 ? Qui est-ce ?
3. Comment compter le nombre d'amis avec la fonction `len` ?

### Exercice 1

1. Créer directement la liste des notes du tableau ci-dessous

Note
12
13
10
2
14

2. À l'aide d'une boucle, calculer la somme des notes.
3. Vérifier avec la fonction `sum`

```
>>> sum([1, 2, 3])
6
```

4. En utilisant le résultat précédent et la fonction `len`, calculer la moyenne du devoir.

### Exercice 2

On considère mes nouveaux amis : Gandalf, Saroumane, Frodon et Bilbon.

*Vive les amis imaginaires.*

1. Créer la liste de mes amis.
2. Produire l'affichage suivant en utilisant une boucle :

Gandalf comporte 7 lettres  
Saroumane comporte 9 lettres  
Frodon comporte 6 lettres  
Bilbon comporte 6 lettres

### Exercice 3

On considère la fonction mathématique définie sur  $\mathbb{R}$  par  $f(x) = 3x - 2$ .

1. Définir la fonction.
2. Créer la liste des images par  $f$  des entiers entre 1 et 10 inclus.

### Conclusion

Les listes Python sont très pratiques pour collecter des éléments, de n'importe quelle nature. On les rencontre très souvent.

Il existe de nombreuses autres méthodes pour les manipuler. Les plus curieux peuvent consulter le cours de première NSI ou la documentation officielle [ici](#) et [là](#).