

# Première NSI - types construits

## Résumé

qkzk

### Type construits

Un type *construit* est une collection d'objets. Python en propose trois : `list`, `tuple` et `dict`.

### Structures linéaires (éléments numérotés) : `list` et `tuple`

Une structure linéaire est composée d'un nombre fini *d'éléments numérotés*, accessibles par leur *indice*.

### Mutabilité

Une structure de donnée est mutable si elle peut évoluer. Concernant les types construits cela signifie qu'on peut ajouter, modifier ou supprimer des éléments de la structure

#### `tuple` : collection indexée non mutable

Les `tuple` sont des collections numérotées non mutables. Une fois créés, on ne peut changer leur contenu.

```
>>> mon_tuple = (1, 2, 3)
>>> len(mon_tuple)
3
>>> mon_tuple[0]
1
>>> mon_tuple[-1] ## depuis la fin
3
>>> for elt in mon_tuple:
...     elt
1
2
3
>>> 3 in mon_tuple
True
>>> 4 in mon_tuple
False
>>> a, b, c = mon_tuple
>>> a
1
>>> c
3
```

Les `str` (chaînes de caractères) se comportent comme des `tuple`... de caractères.

#### `list` : collection indexée mutable

Les `list` se manipulent comme des `tuple` *mutables*. Plus pratiques mais plus lentes que les `tuple`.

```
>>> ma_liste = [1, 2, 3]
>>> len(ma_liste)
3
>>> ma_liste[0]
1
>>> ma_liste[-1] ## depuis la fin
3
>>> for elt in ma_liste:
...     elt
1
2
3
>>> 3 in ma_liste
True
>>> 4 in ma_liste
False
>>> a, b, c = ma_liste
>>> a
1
>>> c
3
```

## Mutabilité des list

```
>>> ma_liste.append(4)
>>> ma_liste
[1, 2, 3, 4]
>>> del ma_liste[0]
>>> ma_liste
[2, 3, 4]
>>> ma_liste.pop()
4
>>> ma_liste
[2, 3]
```

## Dictionnaires : cle → valeur

Un dict associe des *clés* à des *valeurs*. Pas d'indice mais mutables.

```
>>> mon_dict = {'a': 1, 'b': 2}
>>> mon_dict['a']
1
>>> len(mon_dict)
2
>>> 'c' in mon_dict
False
>>> mon_dict['c'] = 3
>>> mon_dict
{'a': 1, 'b': 2, 'c': 3}
>>> del mon_dict['a']
>>> mon_dict
{'b': 2, 'c': 3}
```

### Trois manières d'itérer sur les dict

```
>>> for cle in mon_dict.keys():
...     cle
'b'
'c'
>>> for valeur in mon_dict.values():
...     valeur
2
3
>>> for cle, valeur in mon_dict.items():
...     cle, valeur
'a', 2
'b', 3
```

## Types construits en Python : list, tuple et dict

Python utilise de nombreux types construits. Dans l'usage courant nous rencontrons un des trois types cités ci-dessus

Type	list	tuple	dict
créer directement	<code>l=[1, 2, 3]</code>	<code>t=(1, 2, 3)</code>	<code>d={'a':1, 'b': 2}</code>
accéder	<code>l[2] # 1</code>	<code>t[2] ## 1</code>	<code>d['a'] # 1</code>
contient ?	<code>2 in l # True</code>	<code>2 in t # True</code>	<code>'a' in d # True</code>
mutable ?	oui	non	oui
ajouter un élt.	<code>l.append(4)</code>	impossible	<code>d['c'] = 3</code>
supprimer un elt.	<code>del l[1]</code>	impossible	<code>del d['a']</code>

### Comment choisir ?

Le plus souvent : list

- regrouper des éléments de même nature : **tuple** ou **list** (si on doit les faire évoluer)
- associer un nom à des objets : **dict**
- savoir si un objet est dans une collection : **dict** (le plus rapide)
- itérer rapidement : **tuple**