

14/10/2024 - CM5

API

LE WOOCRAP



1

Allez sur wooclap.com

2

Entrez le code d'événement dans le
bandeau supérieur



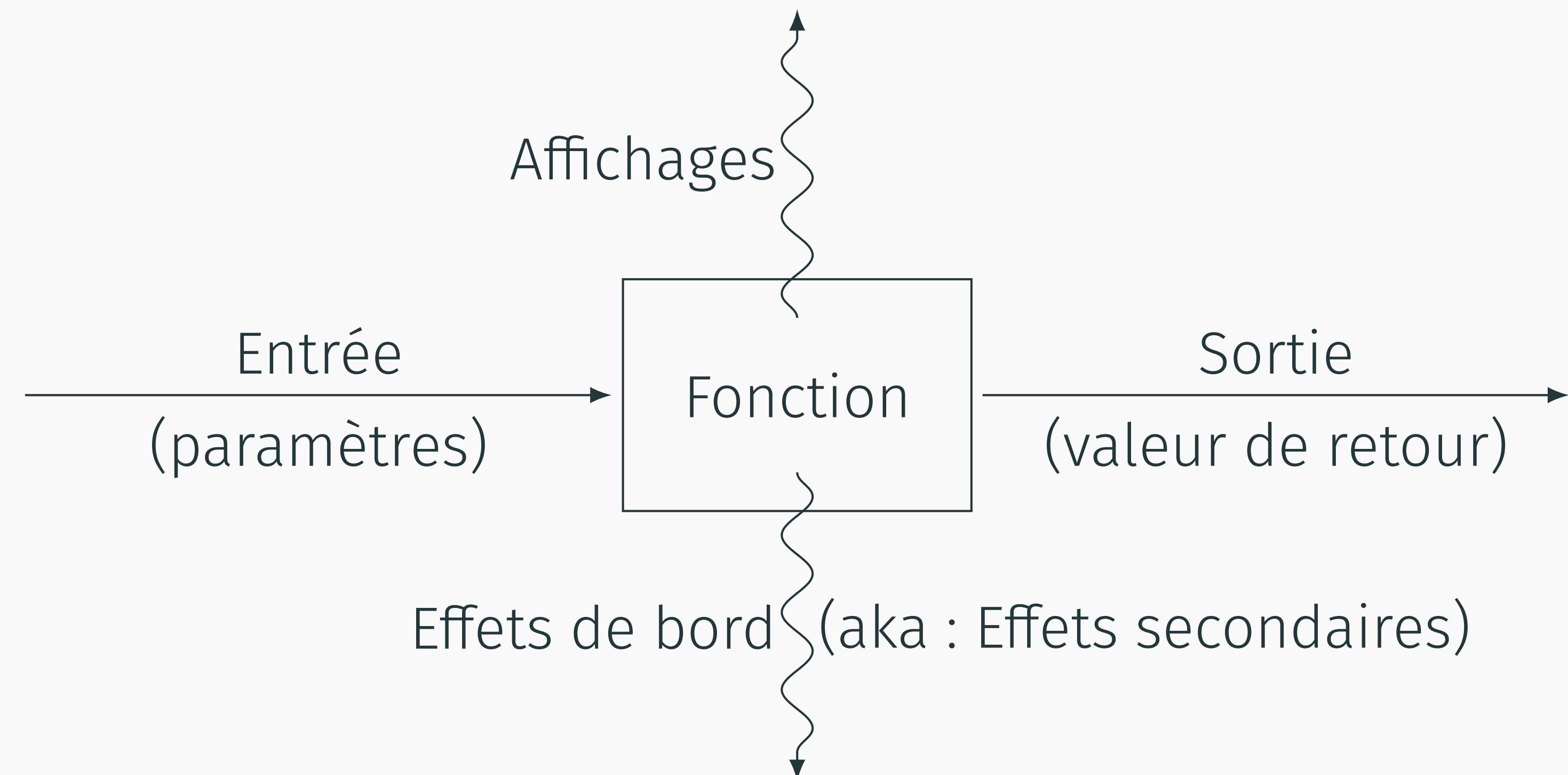
Activer les réponses par SMS

Code d'événement
IQDDPS

DANS L'ÉPISODE PRÉCÉDENT....

DANS L'ÉPISODE PRÉCÉDENT....

Anatomie d'une fonction



DANS L'ÉPISODE PRÉCÉDENT....

Pile d'appels

Quand un nouvel appel commence

- l'exécution de la fonction en cours s'interrompt
- un nouvel espace de noms local est créé
- l'exécution de la fonction appelée commence

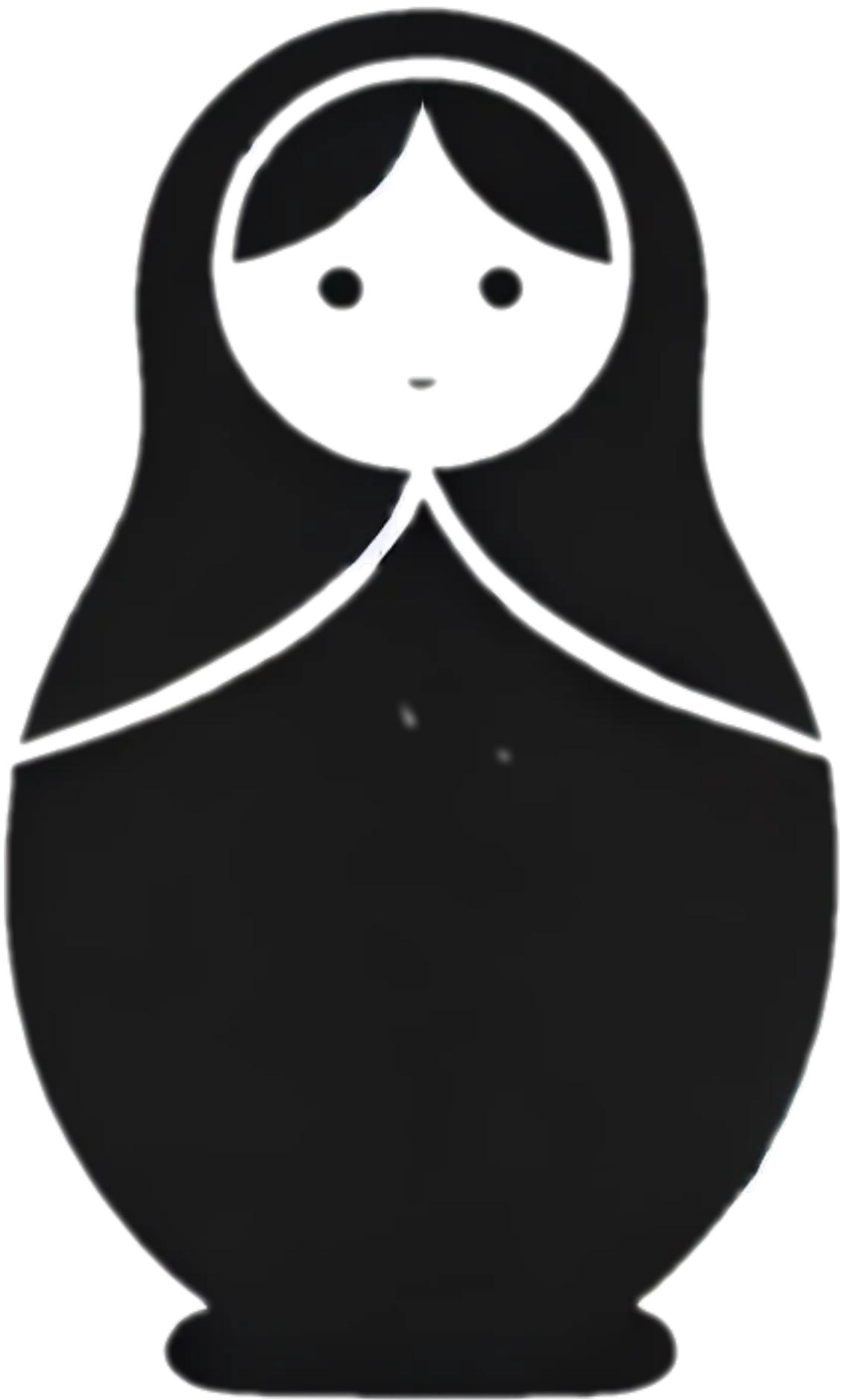
Quand l'appel en cours se termine

- son espace de nom est supprimé de la pile
- l'exécution de l'appel précédent reprend

APPELS IMBRIQUÉS - UNE ALLÉGORIE



APPELS IMBRIQUÉS - EXEMPLE



```
def f(phrase):
    phrase = phrase + "petite "
    return g(h(phrase))

def g(phrase):
    phrase = phrase + "brise "
    return h(phrase)

def h(phrase):
    if phrase[0] == "L":
        phrase = phrase + "la "
        return k(phrase)
    else :
        return "La " + phrase

def k(phrase):
    return phrase + "glace."

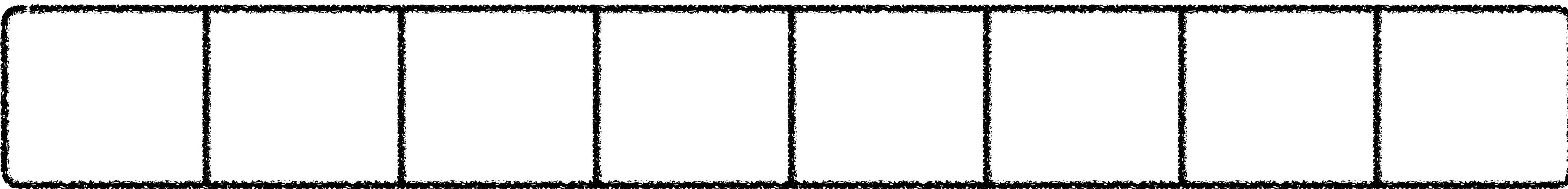
phrase_test = ""
phrase_test = f(phrase_test)
```

Les listes !!!

Les listes

Les listes

~ Tableaux

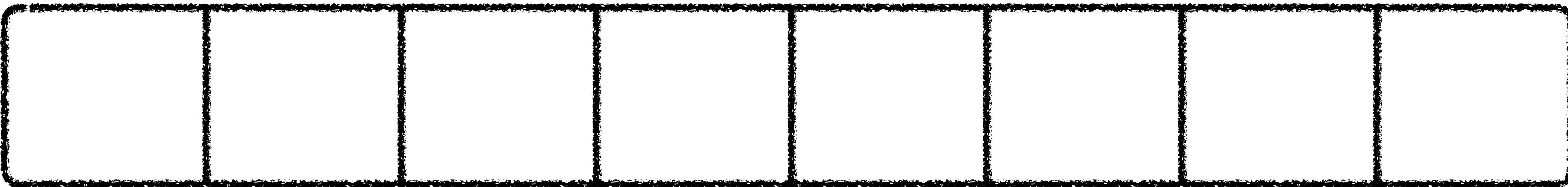


Les listes

~ Tableaux

lst =

liste ←



Les listes

~ Tableaux

lst =



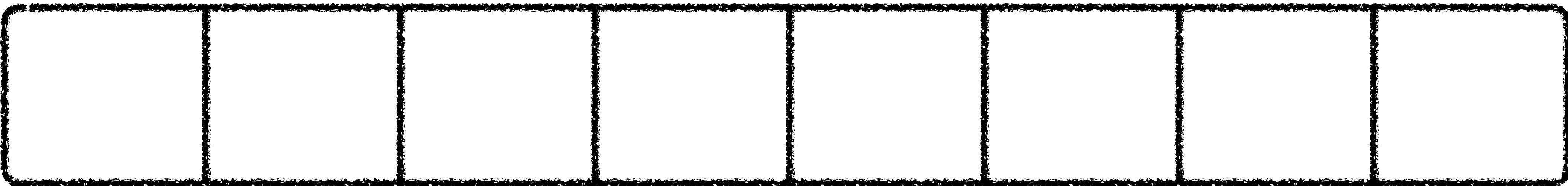
liste ←

Les listes

~ Tableaux

```
lst = ['coucou', 'les', 'API']
```

liste ←

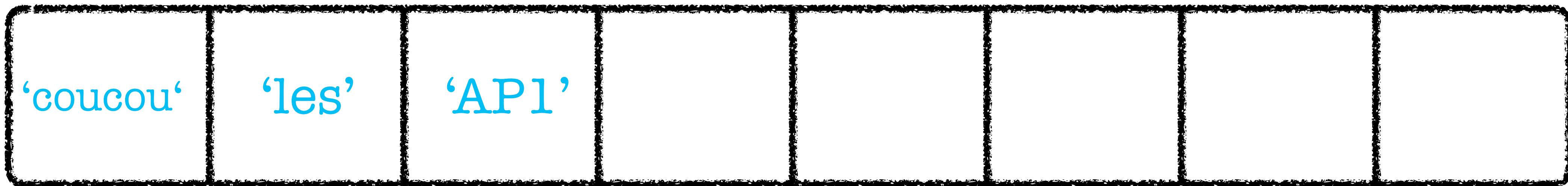


Les listes

~ Tableaux

```
lst = ['coucou', 'les', 'API']
```

liste ← ['coucou', 'les', 'API']

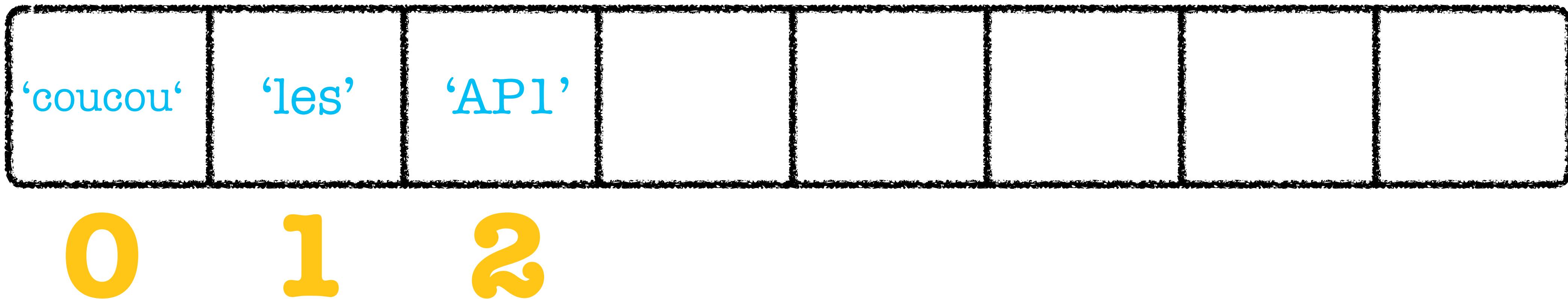


Les listes

~ Tableaux

```
lst = ['coucou', 'les', 'API']
```

liste ← ['coucou', 'les', 'API']



Les listes

~ Tableaux

lst = ['coucou', 'les', 'הוּא']

liste ← ['coucou', 'les', 'הוּא']

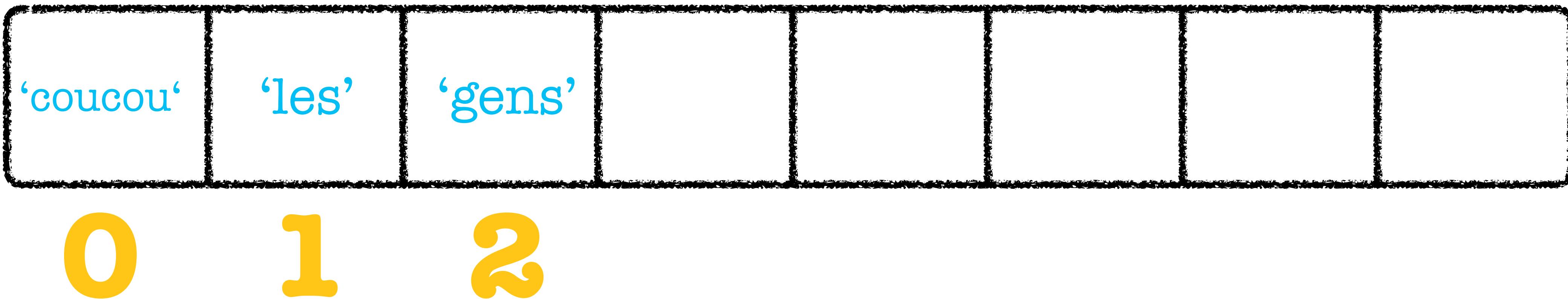


Les listes

~ Tableaux

`lst[2] = 'gens'`

`liste ← ['coucou', 'les', 'AP1']`

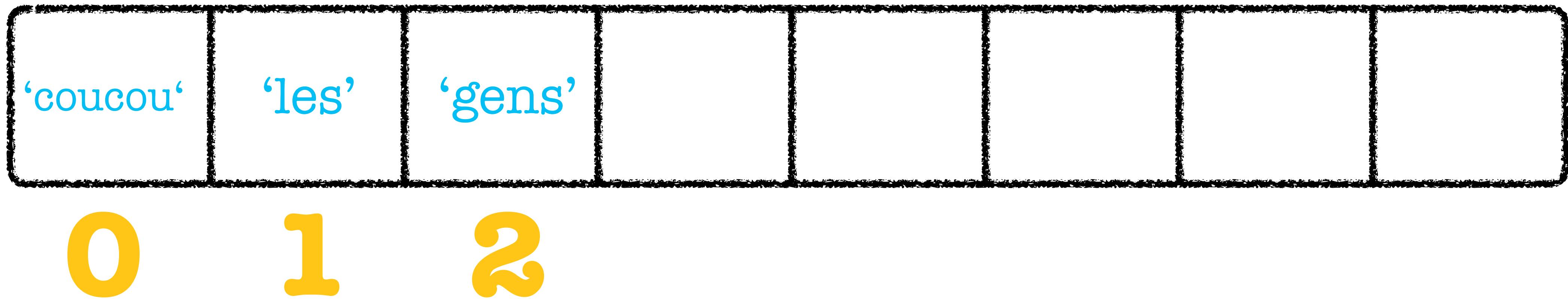


Les listes

~ Tableaux

`lst[2] = 'gens'`

`liste ← ['coucou', 'les', 'AP1']`



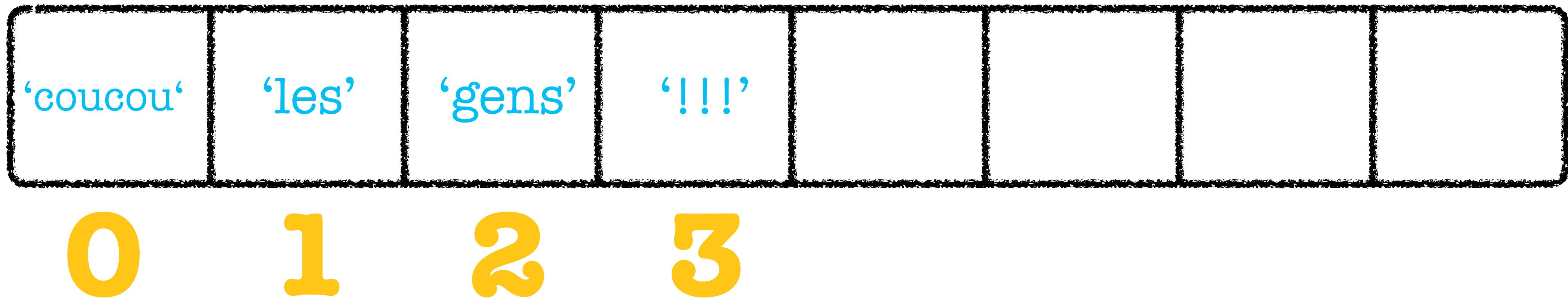
`lst.append('!!!')`

Les listes

~ Tableaux

`lst[2] = 'gens'`

`liste ← ['coucou', 'les', 'AP1']`



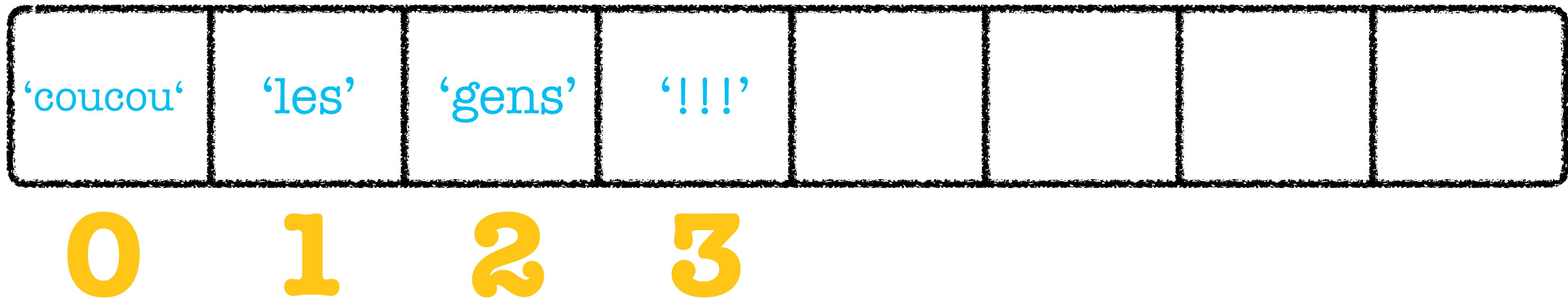
`lst.append('!!!')`

Les listes

~ Tableaux

`lst[2] = 'gens'`

`liste ← ['coucou', 'les', 'AP1']`



`lst.append('!!!')`

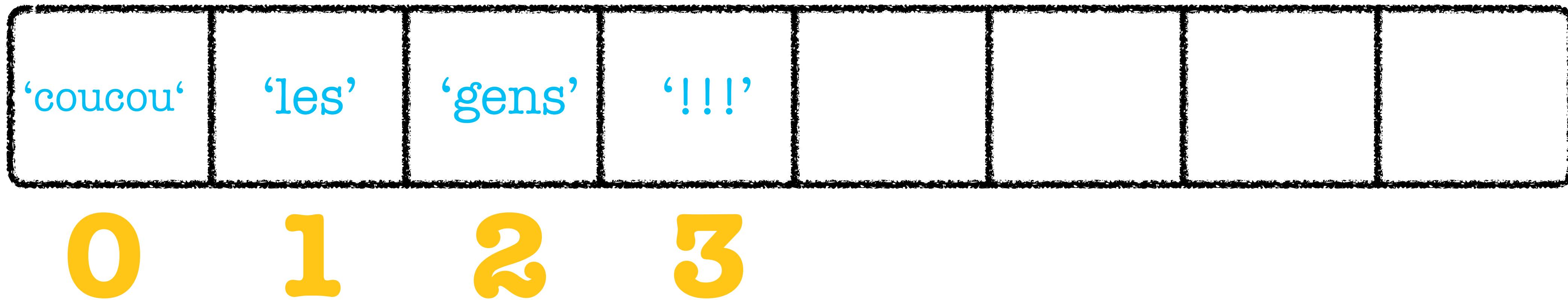
Ajouter '!!!' à (la fin de) liste

Les listes

~ Tableaux

`lst[2] = 'gens'`

`liste ← ['coucou', 'les', 'AP1']`



`lst.append('!!!')`

`lst.pop()`

Ajouter '!!!' à (la fin de) liste

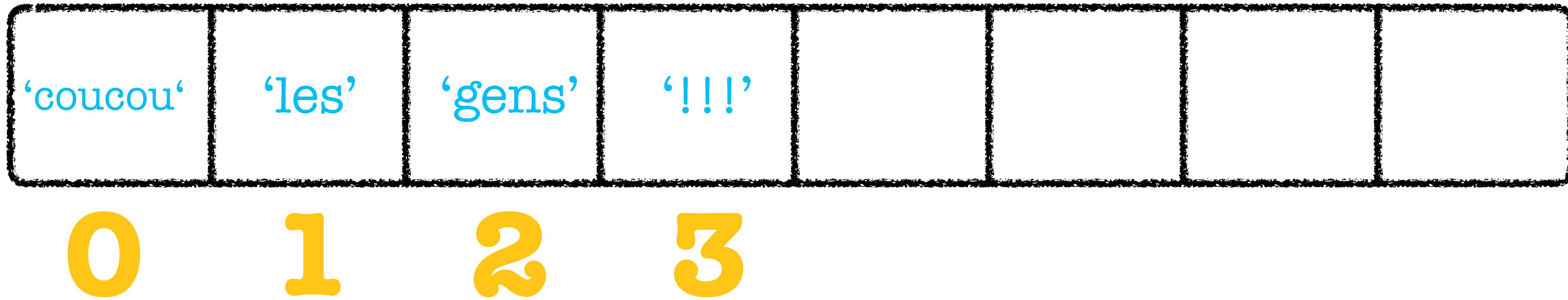
Extraire le (dernier) élément de liste

Les listes

~ Tableaux

`lst[2] = 'gens'`

`liste ← ['coucou', 'les', 'AP1']`



`lst.extend(lst2)` Ajouter les éléments de `lst2` à (la fin de) `liste`
`lst.index(elm)` Renvoyer le (premier) indice où se trouve `elm`

Les listes

~ Tableaux

`lst[2] = 'gens'`

`liste ← ['coucou', 'les', 'AP1']`



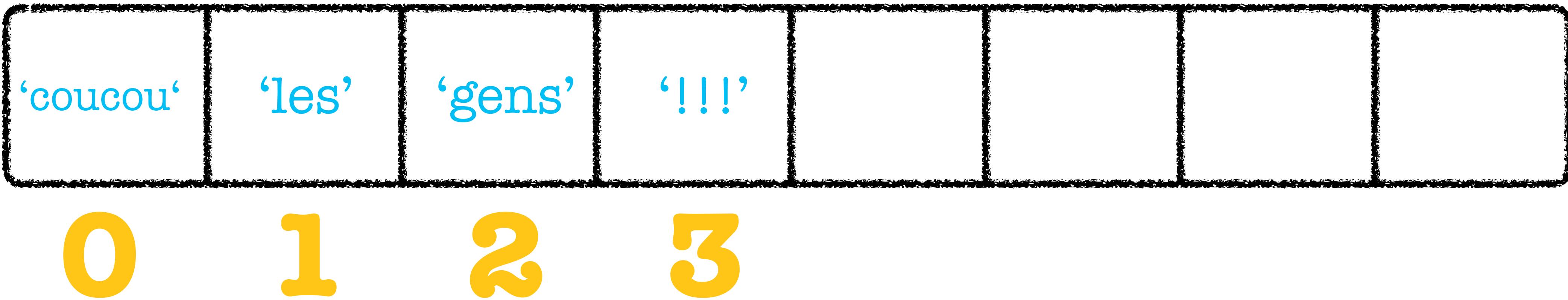
Plus : <https://docs.python.org/fr/3/library/stdtypes.html#sequence-types-list-tuple-range>

Les listes

Les utiliser, les parcourir...

Les listes

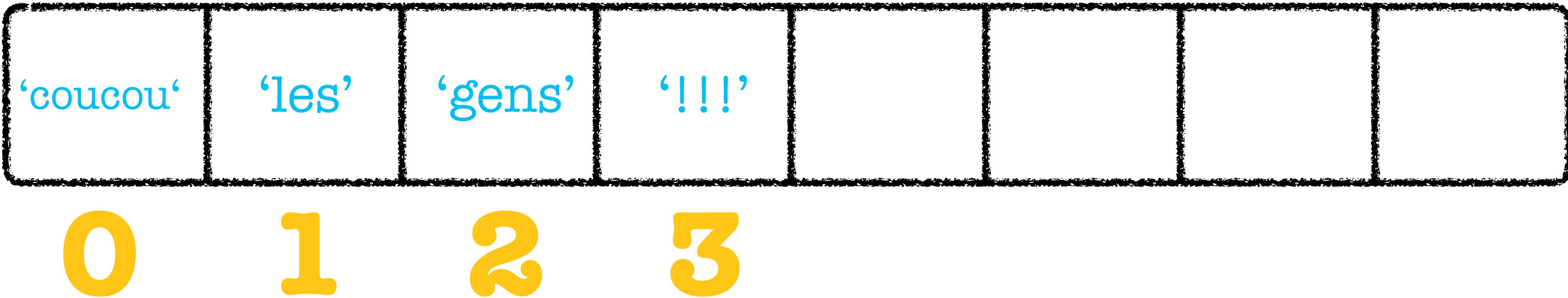
Les utiliser, les parcourir....



En général : de gauche à droite, case par case

Les listes

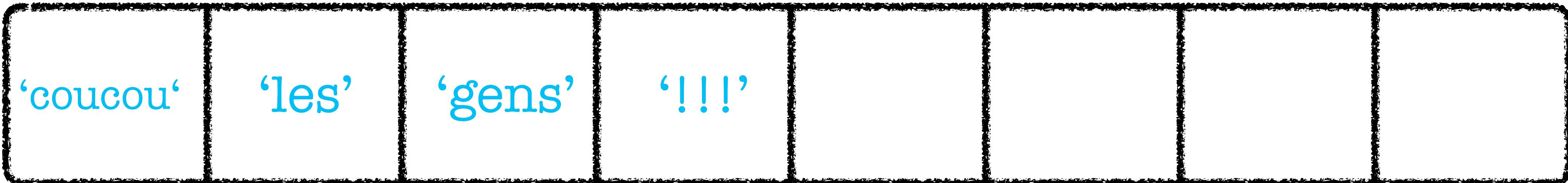
Les utiliser, les parcourir....



En général : de gauche à droite, **case par case**

Les listes

Les utiliser, les parcourir....



0 1 2 3

En général : de gauche à droite, **case par case**

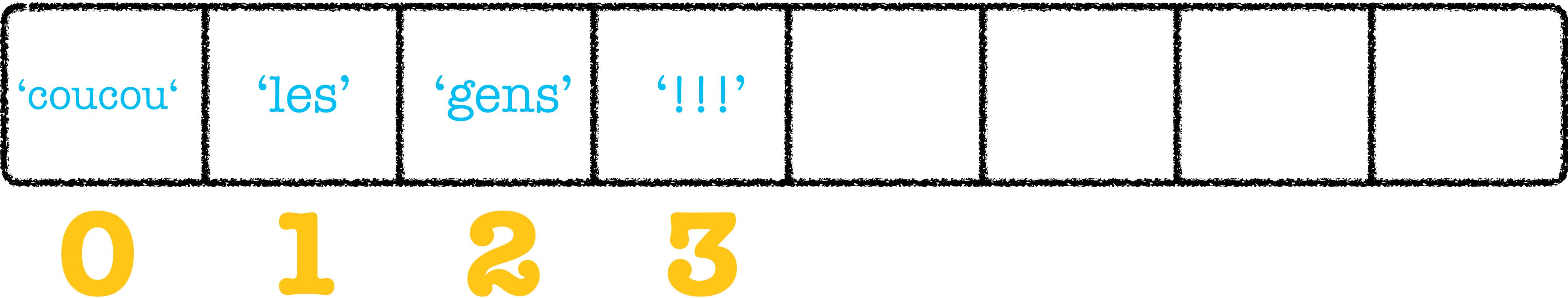
Élément par élément

ou

Indice par indice

Les listes

Les utiliser, les parcourir....

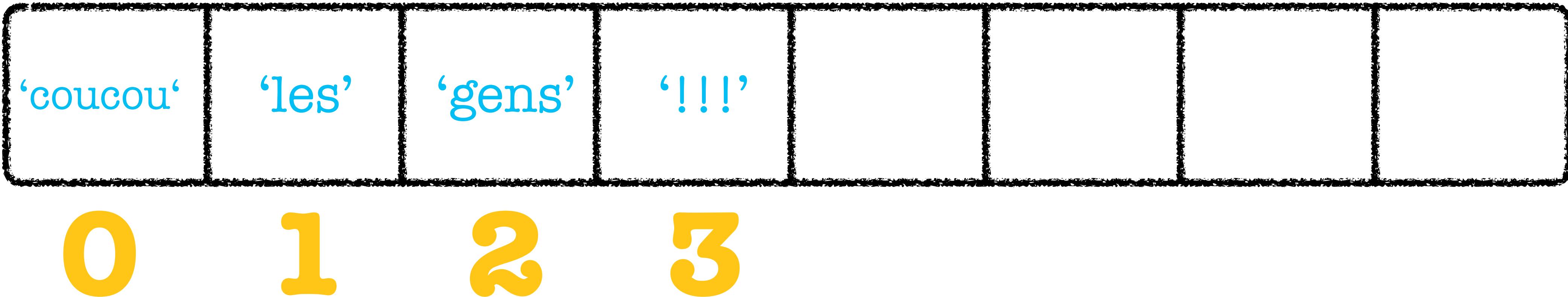


Élément par élément :

```
for elem in liste :  
    | instructions
```

Les listes

Les utiliser, les parcourir...



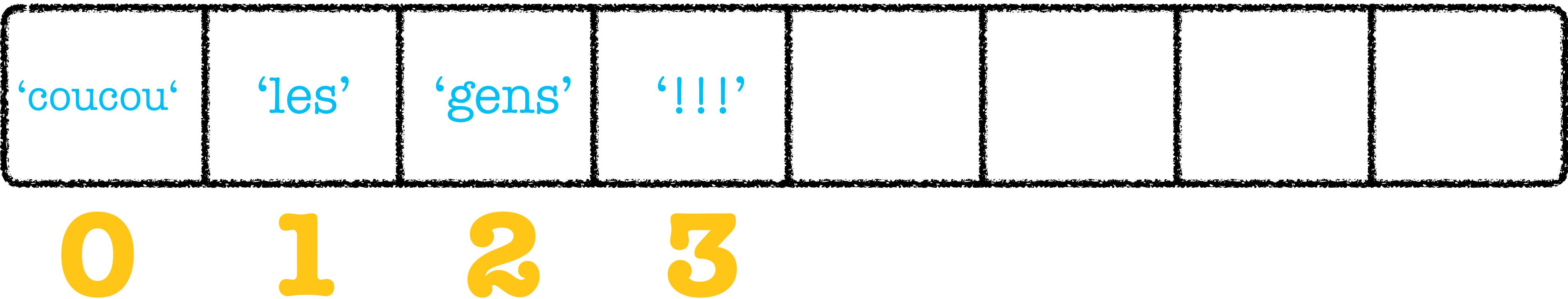
Élément par élément :

```
for elem in liste :  
    | instructions
```

Pour chaque élément de liste :
 | *instructions*

Les listes

Les utiliser, les parcourir...



Indice par indice :

```
indice = 0
```

```
while indice < len(liste) :
```

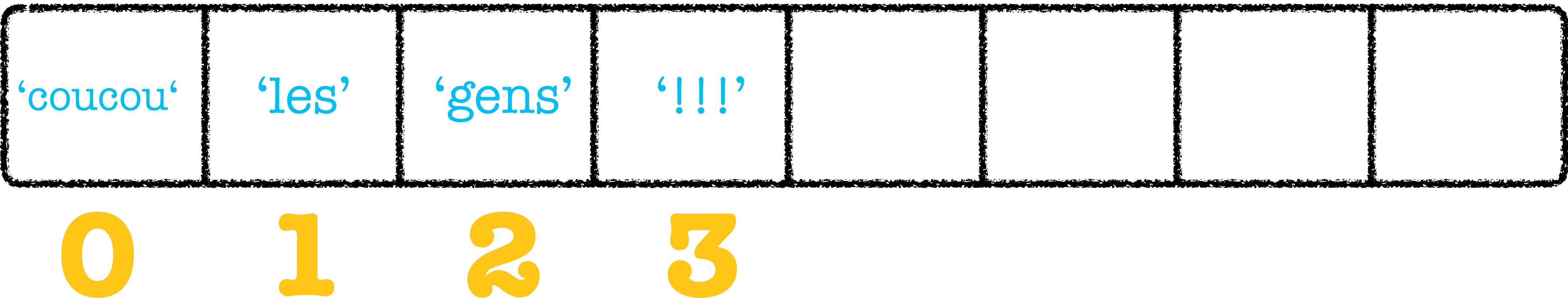
```
    | instructions
```

```
    | ....
```

```
    | indice += 1
```

Les listes

Les utiliser, les parcourir....



Indice par indice :

```
indice = 0
```

```
while indice < len(liste) : OU
```

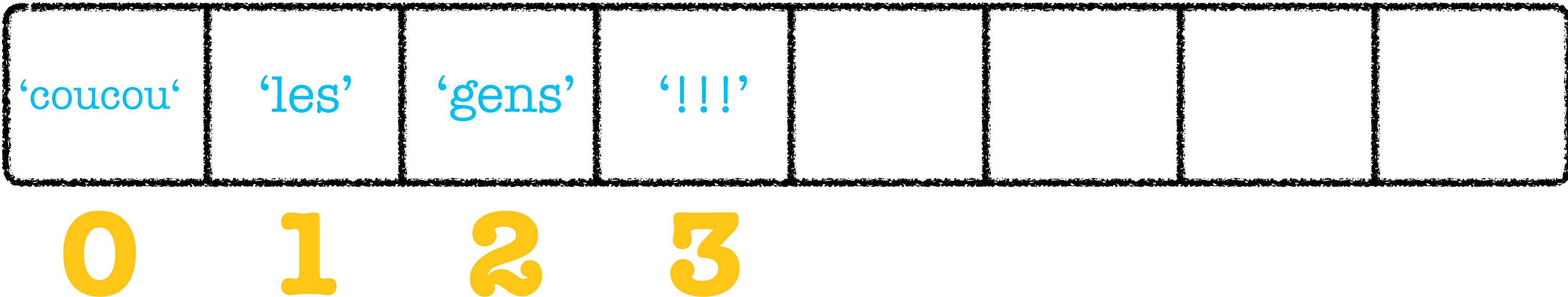
```
| instructions
```

```
| ....
```

```
| indice += 1
```

Les listes

Les utiliser, les parcourir....



Indice par indice :

indice = 0

while indice < len(liste) :

| instructions

|

| indice += 1

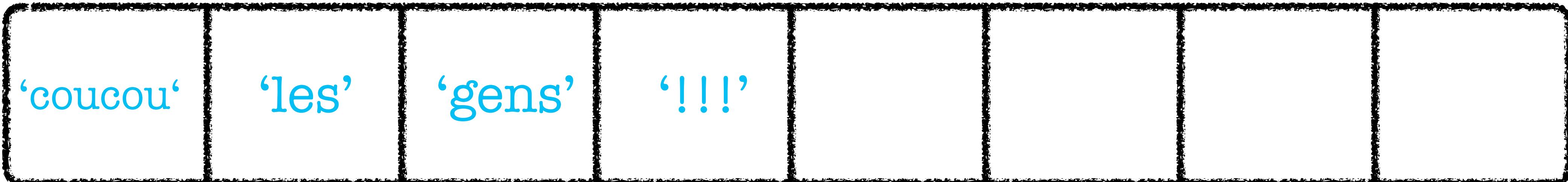
OU

for indice **in** range(len(liste)) :

| instructions

Les listes

Les utiliser, les parcourir....



0 1 2 3

Indice par indice :

indice = 0

while indice < len(liste) :

| instructions

|

| indice += 1

OU

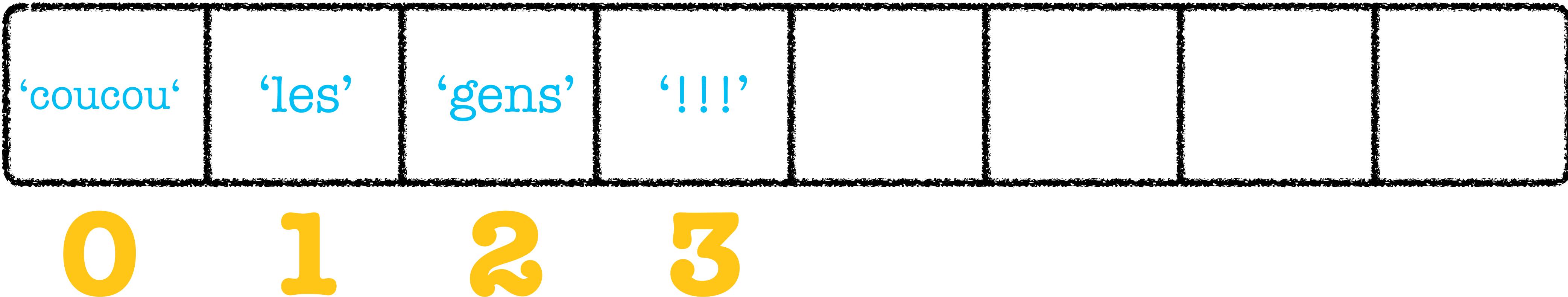
for indice **in** range(len(liste)) :

| instructions

Pour i allant de 0 à n :
instructions

Les listes

Les utiliser, les parcourir...

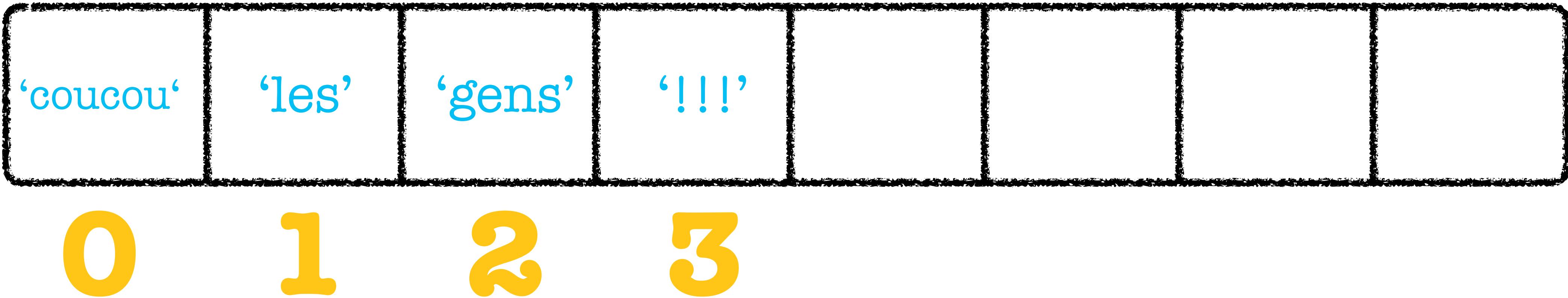


Les deux :

```
for i, elt in enumerate(liste) :  
    instructions
```

Les listes

Les utiliser, les parcourir...



Les deux :

```
for i, elt in enumerate(liste) :  
    print(i, elt)
```

Les listes

Les utiliser, les parcourir...



0 1 2 3

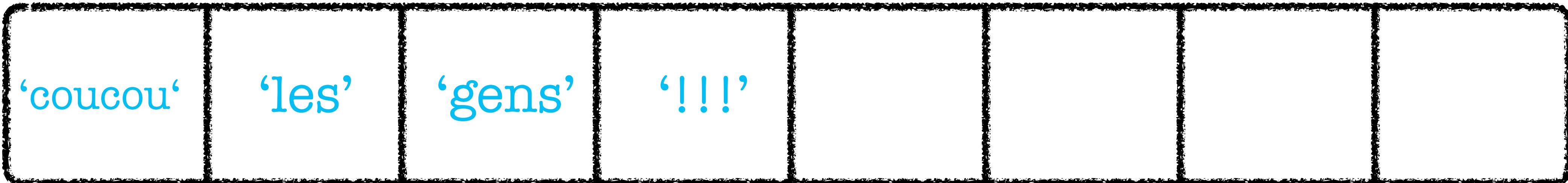
0 'coucou'

Les deux :

```
for i, elt in enumerate(liste) :  
    print(i, elt)
```

Les listes

Les utiliser, les parcourir...



0 1 2 3

Les deux :

0 'coucou'
1 'les'

```
for i, elt in enumerate(liste) :  
    print(i, elt)
```

Les listes

Les utiliser, les parcourir...



0 1 2 3

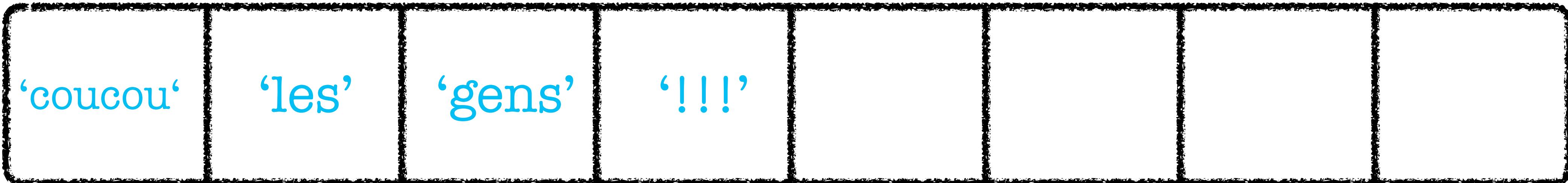
Les deux :

```
for i, elt in enumerate(liste) :  
    print(i, elt)
```

0 'coucou'
1 'les'
2 'gens'

Les listes

Les utiliser, les parcourir...



0 1 2 3

Les deux :

```
for i, elt in enumerate(liste) :  
    print(i, elt)
```

0 'coucou'
1 'les'
2 'gens'
3 '!!!'

En général : parcours de gauche à droite

Élément par élément

ou

Indice par indice

En général : parcours de gauche à droite

Élément par élément

ou

Indice par indice

SEULEMENT POUR LES LISTES ???

En général : parcours de gauche à droite

Élément par élément

ou

Indice par indice

SEULEMENT POUR LES LISTES ???

NON : « ITERABLES »

En général : parcours de gauche à droite

Élément par élément

ou

Indice par indice

- Listes
- Chaînes
- *Tuples*
- Ensembles
- Fichiers...

SEULEMENT POUR LES LISTES ???

NON : « ITERABLES »

Revenons aux listes

- Algorithmes classiques sur les listes :
 - Parcours pour :
 - Trouver un élément ou un indice
 - Etablir une propriété

Revenons aux listes

- Algorithmes classiques sur les listes :
 - Parcours pour :
 - Trouver un élément ou un indice
 - Etablir une propriété

Revenons aux listes

- Algorithmes classiques sur les listes :
 - Parcours pour :
 - Trouver un élément ou un indice
 - Etablir une propriété

Revenons aux listes

- Algorithmes classiques sur les listes :
 - Parcours pour :
 - Trouver un élément ou un indice
 - Etablir une propriété

Revenons aux listes

- Algorithmes classiques sur les listes :
 - Parcours pour :
 - Trouver un élément ou un indice
 - Etablir une propriété
 - Calculer une quantité, fabriquer une liste
 - Plus tard : *Modifier* la liste

Revenons aux listes

Revenons aux listes

Démo Thonny : cm5_14_10_2024.py

Et après ?

Liste de listes (de listes
de listes de)

Et après ?

Liste de listes (de listes
de listes de)

Et si on faisait un jeu de
« Morpion » ?

Et après ?

Liste de listes (de listes
de listes de)

Et si on faisait un jeu de
« Morpion » ?

Démo Thonny : brouillon_morpion.py

