

# PYTHON № 3 / LISTE

---

## 1 liste : niveau 1

L = L.clear() : vide la liste

L = L.append(8) : insère 8 à la fin

L = L.insert(2,7) : insère en 2<sup>e</sup> position 7 (attention numéroté à partir de 0)

L = L.pop() : enlève le dernier élément

L = L.pop(2) : enlève le 2<sup>e</sup> élément ; voir aussi del L[0:4] et L.remove(8)

## 2 liste : niveau 2

génération et occurrence

---

```
1 L = [i for i in range(2, 10)]
2 print(4 in L)
3 print(L.index(4))
```

---

fonction native

---

```
1 L = [4, 15, 7]
2 print(len(L))
3 print(min(L))
4 print(max(L))
```

---

fonction native

---

```
1 L = [4, 15, 7]
2 print(len(L))
3 print(min(L))
4 print(max(L))
```

---

parcours d'une liste

---

```
1 L = [4, 15, 7]
2 for v in L:
3     print(v+10)
```

---

### affichage d'une liste

---

```
1 L = [4, 15, 7]
2 print(L)      # affiche la liste           (avec crochets)
3 print(*L)     # affiche le contenu de la liste (sans crochets)
```

---

### concaténation rapide d'une liste

---

```
1 L = ["4", "15", "7"]
2 print("".join(L))
3 # méthode join associée au string vide
4 # la liste doit être constituée de string
```

---

### génération d'une liste via générateur - cache la complexité du générateur

---

```
1 # générateur
2 def generate(alphabet, max_len):
3     if max_len <= 0: return
4     for c in alphabet:
5         yield c
6     for c in alphabet:
7         for next in generate(alphabet, max_len-1):
8             yield c + next
9 # exploitation
10 a = generate('ab', 3)
11 for i in a:
12     print(i, end=' ')
13
14 # résultat
15 # a b aa ab aaa aab aba abb ba bb baa bab bba bbb
```

---

## 3 liste : niveau 3

### différence de copie de listes

#### copy d'une liste

---

```
1 import copy
2 M = L
3 N = L.copy()
4 P = copy.deepcopy(L)
```

---

## 4 Exercice

**Ex 1 :** manipulation liste

**Ex 2 :** lien avec la fiche 4 - cryptage et décryptage d'un texte stocké par mot 1 liste (on pourra aller plus loin en le stockant dans 1 fichier texte via json)