# Parcourir et utiliser des listes

**Module 7 • 60 minutes • BTS SIO1** • Nom : _____ Date : _____

*Votre mission : suivre les stocks de plusieurs matériels dans une seule variable. Vous apprendrez à lire, modifier et parcourir une liste, puis à rechercher une valeur avec une fonction. Mémo et aides : page 3. Les stocks sont des entiers positifs ou nuls.*

## Comprendre les éléments et leurs indices 12 min

Une liste regroupe des valeurs dans un ordre précis. En Python, les indices commencent à 0. Avec stocks = [4, 0, 6], les trois indices sont 0, 1, 2. len(stocks) donne le nombre d'éléments : 3.

```
stocks = [4, 0, 6]
stocks[1] = 2
stocks.append(5)
```

L'affectation stocks[1] = 2 remplace un élément. stocks.append(5) ajoute 5 à la fin. Complétez sans exécuter le code.

| Après l'instruction | Liste stocks | len(stocks) |
|---|---|---|
| stocks = [4, 0, 6] | [4, 0, 6] | 3 |
| stocks[1] = 2 | | |
| stocks.append(5) | | |

Valeur finale de stocks[0] : _____ ; de stocks[3] : _____.
Peut-on alors lire stocks[4] ? oui / non. Pourquoi ?

.....

## Compléter un parcours 10 min

On veut compter les matériels dont le stock est nul. Pour Chaque stock dans stocks lit directement les valeurs de la liste, une à une.

```
Fonction compter_vides(stocks : liste) : entier
    nombre ← 0
    Pour Chaque stock dans stocks Faire
        Si __________ Alors
            nombre ← __________
        Fin Si
    Fin Pour
    Retourner __________
Fin Fonction
```

Pour stocks = [4, 0, 6, 0], complétez la trace du compteur.

| Valeur stock lue | 4 | 0 | 6 | 0 |
|---|---|---|---|---|
| nombre après ce passage | | | | |

Résultat retourné : _____. Pourquoi nombre est-il initialisé avant la boucle ?

.....

# Traduire rechercher et vérifier

## Traduire la fonction  10 min

Traduisez compter_vides. Pour Chaque stock dans stocks devient for stock in stocks:. Appelez la fonction avec [4, 0, 6, 0] puis affichez son résultat. Écrivez au dos si nécessaire.

.....

.....

.....

.....

.....

.....

.....

## Déboguer un indice  10 min

On veut afficher les trois stocks, un par ligne. Le programme affiche des valeurs puis provoque une erreur.

```
stocks = [4, 0, 6]
for i in range(1, len(stocks) + 1):
    print(stocks[i])
```

1. Quels indices le programme essaie-t-il d'utiliser ? _____
2. Quel premier élément est ignoré ? Quel indice provoque l'erreur ? _____
3. Corrigez la boucle pour lire tous les éléments.

.....

.....

## Créer une fonction de recherche  15 min

Écrivez contient(valeurs, cible), une fonction qui reçoit une liste et un entier à chercher. Elle retourne Vrai si cet entier est présent, Faux sinon. Parcourez les valeurs avec une boucle. En Python, utilisez True et False ; les deux paramètres sont séparés par une virgule. Écrivez pseudo-code puis Python au dos ou dans votre éditeur.

On peut retourner Vrai dès qu'on trouve la cible. Pour conclure Faux, il faut avoir terminé le parcours. Prévoyez puis vérifiez :

| Appel | Prévision | Résultat vérifié |
|---|---|---|
| contient([4, 0, 6], 6) | | |
| contient([4, 0, 6], 2) | | |
| contient([], 6) | | |

## Bilan individuel  3 min

Le premier indice d'une liste est _____. Pour 4 éléments, le dernier indice est _____.
Pour ajouter à la fin : _____. Une liste vide contient _____ élément.
Je sais suivre un parcours : oui / avec aide / pas encore.

# Mémo et aides de la fiche 7

## Lire modifier et ajouter

| Action | Pseudo-code retenu | Python |
|---|---|---|
| Créer une liste | stocks ← [4, 0, 6] | stocks = [4, 0, 6] |
| Lire un élément à l'emplacement 0 | stocks[0] | stocks[0] |
| Modifier un élément | stocks[1] ← 2 | stocks[1] = 2 |
| Ajouter à la fin | Ajouter 5 à stocks | stocks.append(5) |
| Compter les éléments | Longueur(stocks) | len(stocks) |
| Créer une liste vide | numeros ← [] | numeros = [] |

## Deux façons de parcourir

```
for stock in stocks:      # valeurs
    print(stock)
for i in range(len(stocks)):  # indices
    print(stocks[i])
```

Pour 3 éléments, range(len(stocks)) donne 0, 1, 2. Une liste vide donne zéro passage dans ces deux boucles. Une valeur et son indice sont deux informations différentes. Pour cette fiche, on recherche d'abord avec une boucle explicite.

## Aides graduées

| Activité | Aide 1 | Aide 2 |
|---|---|---|
| Comprendre | Suivez remplacement puis ajout. | La longueur change seulement lors de append. |
| Compléter | Le compteur change seulement pour un zéro. | Test : stock = 0 ; ajout : nombre + 1. |
| Traduire | Une boucle peut lire les valeurs sans indice. | Utilisez for stock in stocks: puis if stock == 0:. |
| Déboguer | Comparez les indices demandés à 0, 1, 2. | Utilisez range(len(stocks)). |
| Créer | Trouver suffit pour dire Vrai. | Retourner Faux doit être après la boucle. |

## Aide 3 Recherche en pseudo-code

```
Fonction contient(valeurs : liste, cible : entier) : booléen
    Pour Chaque valeur dans valeurs Faire
        Si valeur = cible Alors
            Retourner Vrai
        Fin Si
    Fin Pour
    Retourner Faux
Fin Fonction
```

Défi facultatif page 4 : choisir les matériels à réapprovisionner. Masquez l'aide complète pendant le premier essai.

# Vers le CCF préparer un réapprovisionnement

Défi facultatif • Environ 12 minutes • Pseudo-code puis Python

Le catalogue contient quatre matériels numérotés de 1 à 4. Leurs stocks, dans cet ordre, sont [0, 5, 2, 3]. Un matériel est à commander si son stock est strictement inférieur à 3. La fonction a_commander(stocks) doit retourner la liste des numéros concernés, dans l'ordre du catalogue.

Attention : le matériel numéro 1 est à l'indice 0 ; le numéro à ajouter est donc i + 1. On retourne les numéros des matériels, pas leurs stocks.

## Compléter et tester

```
Fonction a_commander(stocks : liste) : liste
    numeros ← ______
    Pour i de 0 à Longueur(stocks) - 1 Faire
        Si __________ Alors
            Ajouter __________ à numeros
        Fin Si
    Fin Pour
    Retourner __________
Fin Fonction
```

1. Complétez et expliquez le rôle de numeros.
2. Traduisez puis testez les trois listes ci-dessous.
3. Pourquoi un stock égal à 3 n'est-il pas retenu ?

| stocks | Liste attendue | Liste obtenue |
|---|---|---|
| [0, 5, 2, 3] | | |
| [3, 4] | | |
| [] | | |

## Aides graduées du défi

Aide 1 : la liste résultat est vide au départ ; elle grandit seulement quand le test est vrai.

Aide 2 : utilisez stocks[i] < 3 et ajoutez i + 1. En Python : numeros.append(i + 1). Pour parcourir les indices, utilisez range(len(stocks)).

## Aide 3 Pseudo-code complet

```
Fonction a_commander(stocks : liste) : liste
    numeros ← []
    Pour i de 0 à Longueur(stocks) - 1 Faire
        Si stocks[i] < 3 Alors
            Ajouter i + 1 à numeros
        Fin Si
    Fin Pour
    Retourner numeros
Fin Fonction
```

Convention : si la liste est vide, la boucle Pour de 0 à -1 n'effectue aucun passage. En Python, range(0) ne fournit aucun indice.