

Tableaux, parcours de tableaux

1ere NSI

Année 2020-2021

Nous avons vu que en pseudo-code ou en python les variables peuvent être de plusieurs types:

Nous avons vu que en pseudo-code ou en python les variables peuvent être de plusieurs types:

Entier, Nombres décimaux, du texte, ...

Nous avons vu que en pseudo-code ou en python les variables peuvent être de plusieurs types:

Entier, Nombres décimaux, du texte, ...

Nous allons maintenant voir un nouveau type: le tableau.

Il s'agit d'un ensemble d'éléments, généralement de même type (entiers, ...).

Qu'apporte l'utilisation de tableaux ?

1 Structure d'un tableau

Avons-nous déjà vu des tableaux ?

Lorsqu'on fait une boucle for en python, et que l'on écrit **range(5)**. Ce que l'on fait, c'est créer un tableau qui ne sera utilisé que dans notre boucle for.

On peut représenter un tableau de la manière suivante:



Ce tableau est celui qui est créé par l'instruction `range(5)`, d'après vous, que contient-il ?

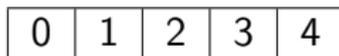
Avons-nous déjà vu des tableaux ?

Lorsqu'on fait une boucle for en python, et que l'on écrit **range(5)**. Ce que l'on fait, c'est créer un tableau qui ne sera utilisé que dans notre boucle for.

On peut représenter un tableau de la manière suivante:



Ce tableau est celui qui est créé par l'instruction `range(5)`, d'après vous, que contient-il ?



Lorsqu'on utilise des tableaux, il faut connaître leur fonctionnement:

- Les cases d'un tableau sont numérotés, allant de 0 a la taille du tableau - 1
- Les cases d'un tableau sont continues, si une case 2 existe ainsi qu'une case 7, on a forcément toutes les cases entre 2 et 7 qui existent.

Propriétés des tableaux

Lorsqu'on utilise des tableaux, il faut connaître leur fonctionnement:

- Les cases d'un tableau sont numérotés, allant de 0 a la taille du tableau - 1
- Les cases d'un tableau sont continues, si une case 2 existe ainsi qu'une case 7, on a forcément toutes les cases entre 2 et 7 qui existent.

Un exemple de tableau:

| Numéro de case | 0 | 1 | 2 | 3 | 4 |
|----------------|-------|--------|-----------|-------|--------|
| Donnée | "Bob" | "Éric" | "Ludovic" | "Dio" | "Jojo" |

Utiliser les tableaux en python: Création

Pour pouvoir utiliser un tableau, il faut d'abord le créer. Pour cela il y a 2 possibilités en python:

```
1 mon_tableau_1 = []  
2 mon_tableau_2 = list()
```

Ces 2 instructions permettent de créer un tableau et de le stocker dans les variables `mon_tableau_1` et `2`.

A noter que les tableaux créés de cette manière sont vides: c'est à dire qu'ils contiennent 0 cases.

Utiliser les tableaux en python: Ajout d'éléments

Nous avons maintenant un tableau, afin de nous en servir, nous allons ajouter des éléments a celui-ci.

Pour ce premier tableau, nous allons lui ajouter une série de nombres. Ceci est réalisé en utilisant la fonction `append`.

```
1 mon_tableau_1.append(5)
2 mon_tableau_1.append(7)
```

Nous avons maintenant 2 elements dans notre tableau, la représentation de celui-ci ressemble donc a cela:

| Numéro de la case | 0 | 1 |
|-------------------|---|---|
| Donnée | | |

Si on utilise la fonction `print()` pour faire `print(mon_tableau_1)`, la représentation dans python sera la suivante:

```
1 [5, 7]
```

Utiliser les tableaux en python: Ajout d'éléments

Nous avons maintenant un tableau, afin de nous en servir, nous allons ajouter des éléments a celui-ci.

Pour ce premier tableau, nous allons lui ajouter une série de nombres. Ceci est réalisé en utilisant la fonction `append`.

```
1 mon_tableau_1.append(5)
2 mon_tableau_1.append(7)
```

Nous avons maintenant 2 elements dans notre tableau, la représentation de celui-ci ressemble donc a cela:

| | | |
|-------------------|---|---|
| Numéro de la case | 0 | 1 |
| Donnée | 5 | 7 |

Si on utilise la fonction `print()` pour faire `print(mon_tableau_1)`, la représentation dans python sera la suivante:

```
1 [5, 7]
```

Quel code utiliser pour:

- Créer un tableau.
- Insérer les valeurs 3, 5, 9 dans le tableau que l'on vient de créer.

Accéder a des éléments de notre tableau

En reprenant notre variable `mon_tableau_1` qui contient les données suivantes:

| | | |
|-------------------|---|---|
| Numéro de la case | 0 | 1 |
| Donnée | 5 | 7 |

Pour accéder aux données il faut utiliser la syntaxe suivante:

```
1 #mon_tableau_1[numéro_de_la_case] permet d'accéder a la case
2 #Par exemple:
3 mon_tableau_1[0]
4 #Renvoie la valeur 5
```

Que renvoie `mon_tableau_1[1]` d'après vous ?

Construire des tableaux par compréhension

Une autre méthode pour construire des tableaux est la méthode dite par compréhension.

Il s'agit, lors de la déclaration du tableau, d'utiliser les expressions for et/ou if.

Par exemple, quel est le résultat de l'exécution du code suivant d'après vous ?

```
1 a = [x for x in range(5)]
```

Questions sur la construction par compréhension

Quel est le résultat de l'exécution du code suivant d'après vous ?

```
1 a = [x*2 for x in range(5)]
```

Quel est le résultat de l'exécution du code suivant d'après vous ?

```
1 a = [x for x in range(5) if x % 2 == 0]
```

On peut aussi utiliser un tableau existant plutôt qu'un range:

```
1 tab = [1,5,2,4,7,6]
2 a = [x for x in tab if x % 2 == 0]
```

Pour finir, on peut les combiner:

```
1 a = [j for i in range(3) for j in range(2)]
```

La compréhension, les règles.

On peut utiliser, lors de la création d'un tableau par compréhension, des boucles for ainsi que des conditions en respectant que:

- Chaque for peut être suivi d'un autre for ou d'un if.
- On n'a jamais 2 fois un "if" à la suite.
- les variables peuvent être créés uniquement dans les for.

Utiliser les tableaux en python: Modification d'éléments

Une fois qu'on accede a la case d'un tableau, on peut la modifier en lui assignant une nouvelle valeur:

Avec ce tableau a la base:

| | | |
|-------------------|---|---|
| Numéro de la case | 0 | 1 |
| Donnée | 5 | 7 |

On effectue cette instruction:

```
1 mon_tableau_1[0] = 123
```

Et le résultat en mémoire est:

| | | |
|-------------------|-----|---|
| Numéro de la case | 0 | 1 |
| Donnée | 123 | 7 |

Parcourir les éléments d'un tableau

Maintenant on sait accéder a la valeur d'une case du tableau.

Si on veut parcourir toutes les valeurs du tableau, voici le pseudo-code

```
1 Role: écrire les valeurs dans le tableau
2 Entrée: tableau : Un tableau
3 Sortie: aucune
4 Pseudo-code:
5 pour i allant de 1 à |tableau| + 1
6     écrire tableau[i]
```

Notez que i prendra toutes les valeurs de 1 a la taille du tableau + 1.

En python la premiere case d'un tableau est 0; en pseudo-code c'est 1.

Parcourir les éléments d'un tableau (while)

Maintenant on sait accéder à la valeur d'une case du tableau.
Si on veut parcourir toutes les valeurs du tableau, on a 2 options principales:

```
1 #La fonction len() donne la taille du tableau
2 taille_tab = len(mon_tableau_1)
3 i = 0
4 while i < taille_tab:
5     print(mon_tableau_1[i])
6     i+=1
7 #Affiche toutes les valeurs de mon_tableau_1 dans l'ordre
```

L'utilisation d'une boucle **while** et d'une variable **i** qu'on incrémente

Parcourir les éléments d'un tableau (for)

On a vu au début du cours que **range(x)** crée un tableau de taille x avec des valeurs de 0 à x-1.

On peut utiliser une boucle **for** de la même manière avec un tableau que l'on a construit nous-même.

```
1 for valeur in mon_tableau_1:  
2     print(valeur)  
3 #Affiche toutes les valeurs du tableau
```

Ici on utilise donc une boucle **for**, et la variable **valeur** qui va prendre une par une toutes les valeurs du tableau.

Passons sur les PC.

Le TP est disponible sur e-lyco.