

Les structures de données

LES LISTES ET LES CHAÎNES DE CARACTÈRES



Les listes

En python, la liste est une structure de données qui agit comme un conteneur de données. Nous pouvons comparer la liste comme un livre où les données sont chacune des pages, dans notre cas, des chaînes de caractères, int, float, etc.

L'utilisation des listes est critique en programmation, car elles permettent de regrouper plusieurs valeurs (données) dans une seule variable et d'y effectuer des opérations principalement à l'aide des boucles.

Il est possible de faire plusieurs actions sur les listes telles que:

- Ajouter une donnée
- Retirer une donnée
- Modifier une donnée
- Consulter une donnée

Création d'une liste

Pour créer une liste en Python, il ne suffit que de créer une nouvelle variable et de lui assigner la valeur « [] », ce qui représente une liste vide

Pour ajouter des valeurs de départ dans la liste, il est possible d'insérer des valeurs entre les crochets, tel que par exemple: [1,2,3] ou ["patate", "jambon", "miel"]

```
#initialisation d'une nouvelle liste vide
```

```
listeVide : list = []
```

```
#initialisation d'une nouvelle liste qui contient les entiers 1, 2, 3 et 4
```

```
listeEntiers : list[int] = [1, 2, 3, 4]
```

Les indices

Suivant l'analogie du livre, une liste est composée de plusieurs éléments, chaque élément a une position dans la liste qui est dénoté par son indice (un peu comme le numéro d'une page). Les indices commencent toujours à zéro et se terminent à « $n - 1$ » où n est le nombre d'items dans le tableau.

Prenons par exemple notre liste « listeEntiers » que nous avons créée précédemment.

```
listeEntiers : list[int] = [1, 2, 3, 4]
```

- Combien d'éléments y a-t-il dans cette liste?
- Quel sera son index maximal?

Indice	0	1	2	3
Valeur	1	2	3	4

Consulter une valeur du tableau

Pour consulter une valeur du tableau, nous pouvons utiliser les indices entre crochets!

```
# Initialisation de la liste
listeEntiers : list[int] = [1, 2, 3, 4]

# Consultation d'une valeur dans la liste (troisième valeur, indice 2)
nombreEntier : int = listeEntiers[2]

# Afficher la valeur dans la console
print(nombreEntier)
```

Indice	0	1	2	3
Valeur	1	2	3	4

Ajout d'une valeur

Pour ajouter une valeur il est possible d'utiliser la fonction `.append(valeur)`.
Cela ajoutera la valeur à la fin de la liste!

```
# Initialisation de la liste
listeEntiers : list[int] = [1, 2, 3, 4]

# Ajout de la valeur 5 à la fin de la liste
listeEntiers.append(32)
```

Indice	0	1	2	3	4
Valeur	1	2	3	4	32

Retirer une valeur (avec remove)

Pour retirer une valeur d'une liste, nous pouvons utiliser la fonction `.remove(valeur)`. Cela retirera la première occurrence de la valeur de la liste

```
# Initialisation de la liste
listeEntiers : list[int] = [1, 2, 3, 4]

# Retire la première occurrence de la valeur 3 de la liste
listeEntiers.remove(3)
```

Indice	0	1	2
Valeur	1	2	4

Insérer une valeur (avec insert)

Pour insérer une valeur dans une liste, nous pouvons utiliser la fonction `.insert(valeur, indice)`. Cela retirera la première occurrence de la valeur de la liste

```
# Initialisation de la liste
listeEntiers : list[int] = [1, 2, 3, 4]

# Insère 15 à l'avant-dernière case de la liste
listeEntiers.insert(15, 3)
```

Indice	0	1	2	3	4
Valeur	1	2	3	15	4

Retirer une valeur (avec pop)

Pour retirer une valeur d'une liste, nous pouvons utiliser la fonction `.pop(indice)`.

```
# Initialisation de la liste
listeEntiers : list[int] = [1, 2, 3, 4]

# Retire la valeur à l'indice 1 de la liste
listeEntiers.pop(1)
```

Indice	0	1	2
Valeur	1	3	4

Modifier une valeur

Pour modifier une valeur du tableau, nous pouvons utiliser l'indice de la valeur que l'on veut modifier entre crochets et ensuite lui assigner une valeur!

```
# Initialisation de la liste
listeEntiers : list[int] = [1, 2, 3, 4]

# Modification d'une valeur
listeEntiers[2] = 7
```

Indice	0	1	2	3
Valeur	1	2	7	4

La taille de la liste

Pour recevoir la taille de la liste, il est possible d'utiliser la fonction `.len(liste)`. Cela nous retournera la taille actuelle de la liste

```
# Initialisation de la liste
listeNoms : list[str] = ["Pierre", "Jean", "Jacques", "Julie"]

# Recevoir la taille de la liste
tailleListe : int = len(listeNoms)

# Affichage dans la console de la taille de la liste
print(tailleListe)
```

Indice	0	1	2	3
Valeur	Pierre	Jean	Jacques	Julie

Parcourir une liste

Il est possible d'utiliser les listes dans les boucles, celles-ci sont primordiales pour effectuer des actions à répétition sur les valeurs d'une liste. Voici deux façons de parcourir la liste

```
# Initialisation de la liste
listeEntiers : list[int] = [15, 32, 12, 64]

i = 0
# à l'aide d'un while
while i < len(listeEntiers):
    print(listeEntiers[i])
    i++

# à l'aide d'un for
for nombre in listeEntiers:
    print(nombre)
```

Les chaînes de caractères

LES STRINGS

Les chaînes de caractères

Les chaînes de caractères sont utilisées pour contenir des données sous forme de texte. En Python, une chaîne de caractères est une séquence de caractères, c'est-à-dire une suite ordonnée de lettres, de chiffres, de symboles ou d'espaces.

Il est possible d'initialiser les chaînes de plusieurs façons différentes:

```
# Avec des guillemets simples
```

```
Chaine1 : str = 'Bonjour!'
```

```
# Avec des guillemets doubles
```

```
Chaine2 : str = "Bonjour!"
```

```
# ces méthodes sont équivalentes
```

```
# chaîne multi-lignes
```

```
chaineML : str = """Bonjour! Bienvenue au cours  
sur les chaîne de caractères en Python! Ceci est une  
chaîne qui s'étend sur plusieurs lignes pour aider  
avec la lisibilité!"""
```

Les chaînes de caractères

On peut apparenter leur manipulation à celle des listes, où chaque caractère dans la chaîne est à un indice dans celle-ci et donc y effectuer certaines opérations (consultation)!

Attention! Il n'est pas possible de faire une modification comme une liste avec une chaîne de caractères: `chaine[0] = 'a'`

Nous verrons comment faire des opérations avancées sur les chaînes de caractères plus tard dans la session!

```
# initialisation de la variable
chaine : str = 'Bonjour!'

premiereLettre : str = chaine[0] # 'B'
deuxiemeLettre : str = chaine[1] # 'o'
```

Indice	0	1	2	3	4	5	6	7
Valeur	B	o	n	j	o	u	r	!

Exercices

Quel sera la sortie dans la console du code suivant:

```
tableauEntiers : list[int] = [1, 2, 3, 4]  
print(tableauEntiers[2])
```

3

Exercices

Quel sera la sortie dans la console du code et les valeurs de la liste suivant:

```
tableauEntiers : list[int] = [1, 2, 3, 4]
tableauEntiers[2] = 57
print(tableauEntiers[1])
```

2

Indice	0	1	2	3
Valeur				

Exercices

Quel sera la sortie dans la console du code suivant:

```
tableauEntiers : list[int] = [1, 2, 3, 4]

for nombre in tableauEntiers :
    nombre = nombre + 10
    print(nombre)
```

```
11
12
13
14
```

Exercices

Quel sera la sortie dans la console du code suivant:

```
listeNoms : list[str] = ["Pierre", "Jean", "Jacques", "Julie"]

listeNoms.append("Luc")
listeNoms.pop(2)
listeNoms[2] = "Ti-Paul"

for nom in listeNoms :
    print(nom)
```

```
Pierre
Jean
Ti-Paul
Luc
```