

Cours : Concept d'une Liste Simple Chaînée

- Une **liste simplement chaînée** est une structure de données linéaire où chaque élément, appelé **cellule** ou **nœud**, contient deux informations essentielles :
 1. **La valeur de l'élément** (ou value).
 2. **Une référence (pointeur) vers l'élément suivant** (ou next).
- Dans une liste simplement chaînée, chaque cellule pointe uniquement vers l'élément qui vient après elle, c'est-à-dire vers le **prochain élément** de la liste. La dernière cellule de la liste pointe vers **null**, ce qui indique la fin de la liste.
- Pour ouvrir la documentation, vous pouvez taper dans le terminal :
 - **Windows** : start ./docs/allclasses-index.html
 - **macOS/Linux** : open ./docs/allclasses-index.html
 - **Navigateur (Windows)** : file:///C:/Users/Gamer/Desktop/devEfficace/SimpleList/docs/allclasses-index.htm

Schéma Conceptuel d'une Liste Simplement Chaînée

1. **Liste vide** (initialisation) :
 - La liste est vide lorsque **head** est égal à **null** (pas d'élément dans la liste).
 - **Tête** de la liste (**head**) pointe vers **null**.

Liste vide :
head -> null

2. **Liste avec une cellule** (ex. Cell 1) :
 - La liste contient une seule cellule, la première (et la dernière) cellule.
 - Cette cellule a un pointeur **next** qui pointe vers **null**, ce qui indique la fin de la liste.
 - La cellule elle-même est la tête de la liste.

```
+-----+
| Cell 1 |
| (head) |
+-----+
| next -->| null
+-----+
```

3. **Liste avec plusieurs cellules** (ex. Cell 1, Cell 2, et Cell 3) :
 - **Cell 1** : next pointe vers **Cell 2**.
 - **Cell 2** : next pointe vers **Cell 3**.
 - **Cell 3** : next pointe vers **null**, marquant la fin de la liste.

```
+-----+      next      +-----+      next      +-----+      next
| Cell 1 | -----> | Cell 2 | -----> | Cell 3 | -----> null
| (head) |           |         |           |         |
+-----+      +-----+      +-----+
```

Remarques sur la Liste Simplement Chaînée

- **Structure linéaire** : Chaque cellule a un seul lien, pointant vers l'élément suivant. Cela signifie qu'on peut seulement parcourir la liste dans une seule direction (du début à la fin).
- **Insertion en tête** : Ajouter un élément au début de la liste implique de créer une nouvelle cellule qui pointe vers l'ancienne tête, et de mettre à jour le pointeur **head** pour qu'il pointe vers cette nouvelle cellule.
- **Insertion à la fin** : Ajouter un élément à la fin nécessite de parcourir toute la liste pour trouver la dernière cellule, puis de modifier son pointeur **next** pour qu'il pointe vers la nouvelle cellule.
- **Suppression** : La suppression d'un élément dans une liste simplement chaînée implique de trouver l'élément précédent, afin de modifier son pointeur **next** pour qu'il saute l'élément à supprimer.
- **Accès direct** : Contrairement aux tableaux, il n'est pas possible d'accéder directement à un élément à une position donnée. Il faut parcourir la liste depuis le début jusqu'à atteindre l'élément souhaité.
- **Complexité des opérations** :
 - L'insertion en tête et la suppression en tête sont des opérations efficaces ($O(1)$).
 - L'insertion à la fin ou la suppression d'un élément quelconque nécessite un parcours de la liste, ce qui prend un temps linéaire ($O(n)$).

Conclusion

La liste simplement chaînée est une structure de données simple, mais puissante pour certaines applications, où l'insertion et la suppression d'éléments à l'avant de la liste sont fréquentes. Cependant, elle n'offre pas les avantages d'une liste doublement chaînée, comme un accès rapide aux éléments précédents, et ne permet pas un accès direct aux éléments au milieu de la liste.