

Support de cours

Cours:

## Introduction à la programmation orientée objet (en C++)

Vidéo:

### INIT-CPP-7\_3-Pointeurs

Concepts (extraits des sous-titres générés automatiquement) :

**Vrais pointeurs. Page d'un carnet d'adresses. Prochaine séquence vidéo. Nouvelle variable. Variable x. Différents cas. Syntaxe c. Nouvelle page. Principale différence. Adresse. Site internet. Séquence vidéo. Premier cas. Adresse d'un site. Abus de langage.**



[vers la recherche de séquences vidéo](#)

(dans Introduction à la programmation orientée objet (en C++).)



[vers la vidéo](#)

Center for Digital Education. Plus de matériel de soutien pédagogique ici :

<https://www.epfl.ch/education/educational-initiatives/cede/educational-technologies-gallery/boocs-en/>

# Pointeurs : concept et analogie

## Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s





Nous avons vu en détail les références. Passons maintenant aux vrais pointeurs. La principale différence c'est qu'un pointeur,

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

.....

.....

.....

.....

.....

0m 1s

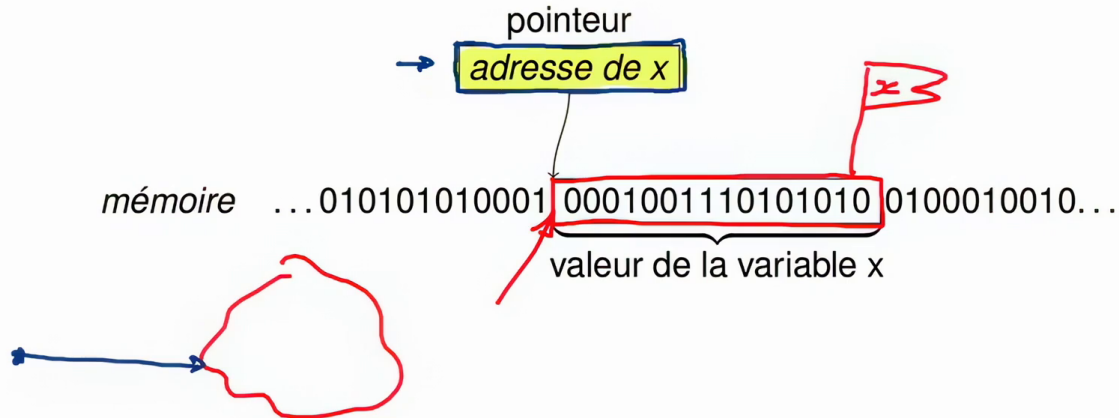


## Les (vrais) pointeurs

Une variable est physiquement identifiée de façon unique par son **adresse**, c'est-à-dire l'adresse de l'emplacement mémoire qui contient sa valeur.

Un **pointeur** est une variable qui contient l'**adresse** d'un autre objet informatique.

☞ une « *variable de variable* » en somme



c'est une variable qui existe vraiment en mémoire. C'est pas juste une étiquette sur une variable qui existait au préalable mais ça devient une nouvelle variable. Pour bien comprendre ce dont il s'agit, il faut d'abord savoir que les variables, en général, sont identifiées par une adresse en mémoire. Supposons par exemple que j'ai une variable  $x$  qui existe en mémoire qui contient quelque chose, peu importe, et que je regarde donc la mémoire comme ça au fur et à mesure que j'avance. A un certain endroit, la valeur de  $x$  va commencer par être écrite en mémoire sous forme binaire, et puis à un autre moment ce sera autre chose, d'autres variables. Eh bien donc cette variable  $x$  est identifiée par son adresse, l'endroit où elle démarre en mémoire. On peut donc prendre l'adresse en mémoire d'une variable. Et un pointeur, c'est rien d'autre justement, qu'une variable qui contient l'adresse d'un autre objet qui existe en mémoire. Un pointeur c'est donc bien une variable contenant l'adresse d'une autre variable. C'est en somme une variable de variables, exactement comme le signet sur un site internet, que j'ai pris dans l'exemple d'introduction, est une adresse d'un site qui existe par ailleurs. Et donc en ce sens,

notes

résumé

0m 8s

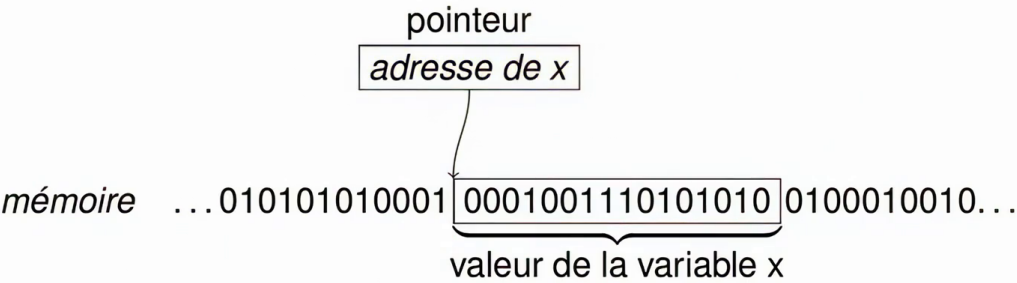


# Les (vrais) pointeurs

Une variable est physiquement identifiée de façon unique par son **adresse**, c'est-à-dire l'adresse de l'emplacement mémoire qui contient sa valeur.

Un **pointeur** est une variable qui contient l'**adresse** d'un autre objet informatique.

☞ une « *variable de variable* » en somme



Note : une référence n'est pas un «vrai pointeur» car ce n'est pas une variable en tant que telle, c'est juste une «autre étiquette» :

différence entre «une autre étiquette» (référence) et «une variable contenant une adresse» (pointeur, un niveau de plus).

références et pointeurs sont quand même fondamentalement différents.

notes

résumé

1m 25s





Une référence c'est pas une variable qui contient une adresse, c'est juste une autre étiquette. Il y a une grosse différence entre une autre étiquette sur le même tiroir et un autre tiroir qui contient l'adresse de quelque chose. Dans un pointeur il y a donc un niveau de plus, une indirection.

#### notes

---

---

---

---

---

---

---

---

---

---

---

---

#### résumé

1m 26s

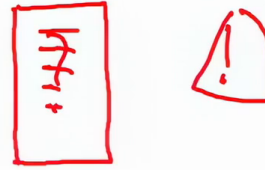


## Les pointeurs : une analogie

Un pointeur c'est comme la *page d'un carnet d'adresse*  
(sur laquelle on ne peut écrire qu'une seule adresse à la fois) :

→ déclarer un pointeur

p.ex. `int* ptr;`



→ affecter un pointeur `ptr`

p.ex. `ptr = &x;`



allouer un pointeur `ptr`

p.ex. `ptr = new int(123);`

Un pointeur c'est donc vraiment comme une page d'un carnet d'adresses. Une page sur laquelle on pourrait écrire qu'une seule adresse à la fois. Et je voudrais maintenant utiliser cette analogie pour illustrer tout un tas d'applications concrètes d'utilisation des pointeurs dans les programmes. Ces différents cas vont être ici illustrés par de la syntaxe C++ que vous n'avez pas encore vue et qui sera présentée dans la prochaine séquence vidéo qui présente les aspects plus pratiques, plus techniques. Nous les avons néanmoins laissés ici pour si vous voulez revoir cette séquence vidéo ou réviser les transparents une fois que vous aurez vu cette syntaxe. Ça pourrait peut-être être utile. Commençons donc par le premier cas : la déclaration d'un pointeur. Je vous ai dit qu'un pointeur c'était une variable et donc on peut déclarer un pointeur. Qu'est-ce que ça veut dire déclarer un pointeur ? Ça veut juste dire ajouter une nouvelle page dans notre carnet d'adresses. Mais attention ajouter une nouvelle page, ça peut pas dire qu'il y a une adresse écrite dessus. Ajouter une nouvelle page, c'est juste ajouter une page sans avoir spécifié ce qui est écrit dessus. Il y a à priori quelque chose d'écrit dessus mais qu'on ne sait pas interpréter. Deuxième chose : c'est affecter un pointeur. Qu'est ce que ça veut dire affecter un pointeur ? Ça veut dire donner une adresse,

notes

résumé

1m 47s





écrire une adresse sur la page du carnet d'adresses. On parle aussi parfois d'allouer un pointeur. Qu'est-ce que ça veut dire allouer un pointeur ? Allouer un pointeur ça veut dire aller acheter un terrain quelque part. C'est allouer de la mémoire et y construire une maison. C'est à dire y mettre une valeur et reprendre l'adresse de cette maison pour la recopier sur la page du carnet d'adresses. Voilà ce que c'est qu'allouer un pointeur. Mais notez bien ici, erreur fréquemment commise, que le pointeur n'est pas ni le terrain, ni la maison. Ce sont deux choses complètement différentes. Le pointeur c'est juste la page du carnet d'adresses qui contient l'adresse de la maison. Ça n'est pas la maison elle même.

#### notes

#### résumé

3m 1s

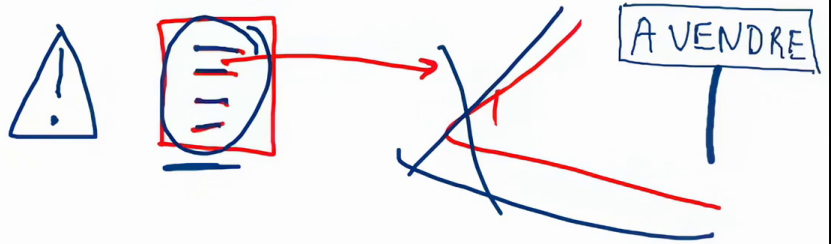




## Les pointeurs : une analogie

Un pointeur c'est comme la *page d'un carnet d'adresse*  
(sur laquelle on ne peut écrire qu'une seule adresse à la fois) :

→ « libérer un pointeur » `ptr`  
(en fait, c'est « libérer l'adresse  
pointée par le pointeur » `ptr`)  
p.ex. `delete ptr;`



En d'autres termes, le pointeur n'est pas égal à la valeur pointée. Ce sont deux choses complètement différentes. Dans le cas d'une page d'un carnet d'adresses qui possède l'adresse d'un terrain qu'on avait acheté, qu'on avait alloué ; libérer un pointeur, ça veut dire simplement que l'on autorise maintenant la mémoire à réutiliser cette zone, que cette zone n'est plus notre propriété et qu'on peut donc la réutiliser librement par ailleurs. Et donc cette zone n'est plus notre propriété. Alors attention, libérer un pointeur, et en fait c'est un abus de langage, c'est libérer le terrain pointé, ça ne veut pas dire que l'on a effacé la page du carnet d'adresse. L'adresse est toujours là mais c'est donc une erreur fréquemment commise

### notes

### résumé

3m 49s



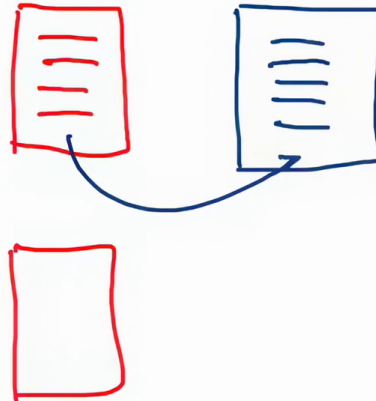
## Les pointeurs : une analogie

Un pointeur c'est comme la *page d'un carnet d'adresse*  
(sur laquelle on ne peut écrire qu'une seule adresse à la fois) :

→ copier un pointeur ptr

p.ex. `p1 = p2;`

→ annuler/effacer un pointeur `ptr` : `p1 = nullptr;`



que de réutiliser l'adresse d'un terrain qui ne nous appartient plus. C'est bien la raison pour laquelle plus tard, on vous conseillera d'effacer la page du carnet d'adresses une fois que vous avez libéré le pointeur. Copier un pointeur, c'est exactement ce que vous faites lorsque vous donnez une adresse à une de vos connaissances. Quand vous donnez une adresse à une de vos connaissances, elle recopie l'adresse que vous lui donnez dans une page d'un carnet d'adresses. Elle ne va pas copier la maison dont vous avez l'adresse. Vous ne passez pas la maison en entier. Vous ne passez que l'adresse, vous recopiez l'adresse. Voilà ce que ça veut dire copier un pointeur : c'est échanger des adresses. Enfin de même, effacer ou annuler un pointeur, c'est simplement effacer la page du carnet d'adresses.

notes

résumé

4m 37s



notes

## résumé

5m 25s

