

Support de cours

Cours:

Initiation à la programmation (en C++)

Vidéo:

Pointeurs - allocation dynamique (partie 3)

Concepts (extraits des sous-titres générés automatiquement) :

Zone mémoire. Adresse de cette zone. Carnet d'adresses. Allocation dynamique. Nouvelle page. Réservation d'un terrain. Emplacement mémoire. Zone mémoire int. Contexte du carnet d'adresses. Page ptr. Besoin de cet emplacement mémoire. Petite analogie du carnet d'adresses. Objet d'un new. Zones mémoires. Instruction de cette nature.



[vers la recherche de séquences vidéo](#)
(dans Initiation à la programmation (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 : allocation dynamique

(Partie 3)

Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

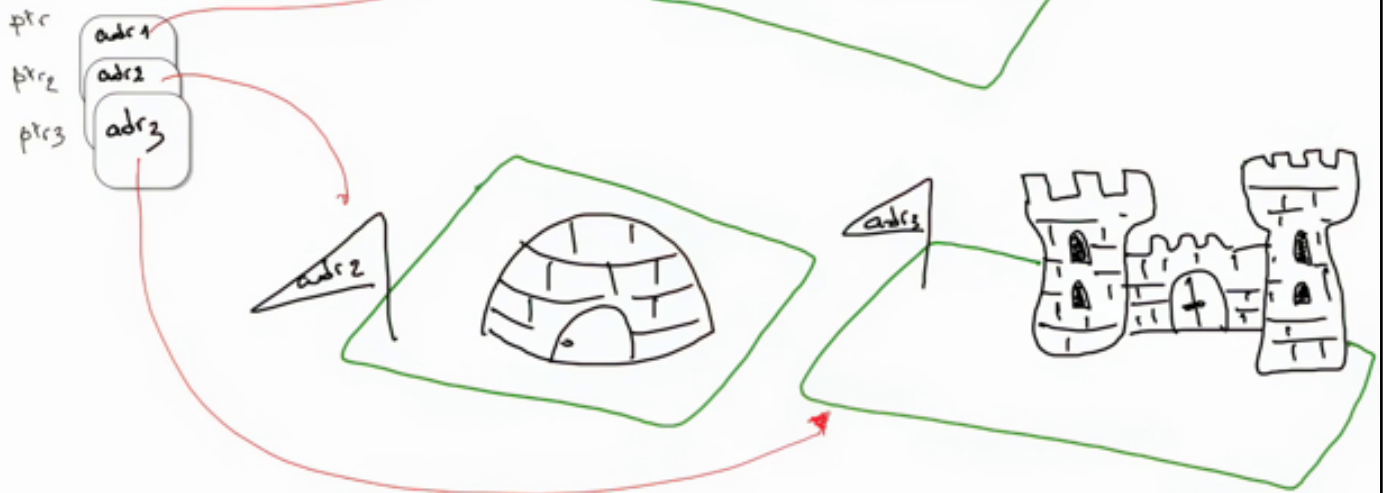
notes

résumé

0m 0s



→ int * ptr (new int(3)) ;
delete ptr



Je peux, dans un programme, allouer des zones mémoires dynamiquement pour tout type de données, et c'est au programmeur d'être attentif à restituer l'emplacement mémoire lorsqu'il ne lui est plus nécessaire. Par exemple, si je n'ai plus besoin de cet emplacement mémoire, alors ce qu'il faut faire, c'est que je dois faire un delete de ptr,

notes

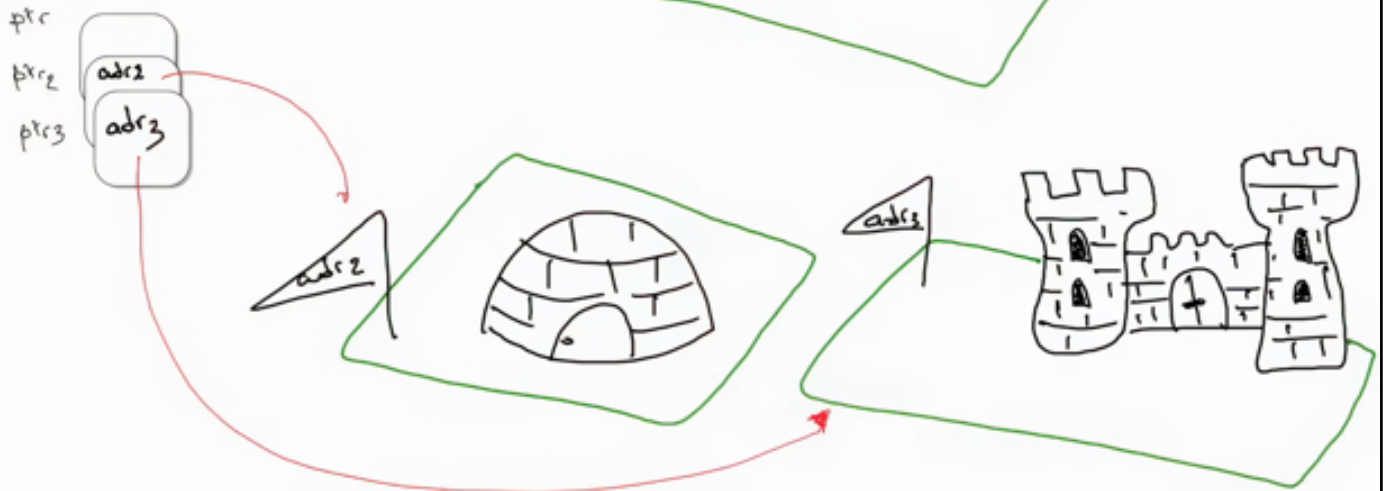
résumé

0m 49s



```
int * ptr (new int(3)) ;
```

```
delete ptr;
ptr = nullptr;
```



restituer, libérer l'emplacement mémoire, ce qui veut dire ici que mon terrain ne contient plus aucune donnée utilisable. Et le terrain est désormais disponible pour quelqu'un d'autre. Si je me contente de cela, eh bien ça voudra dire que dans mon carnet d'adresses, figure toujours l'adresse de ce terrain qui est à vendre, qui est disponible pour quelqu'un d'autre. Donc, la précaution que je dois impérativement prendre est d'indiquer que ptr ne pointe vers plus rien, en lui affectant nullptr. Ce qui revient ici à effacer la page de mon carnet d'adresses, et donc, supprimer ce lien,

notes

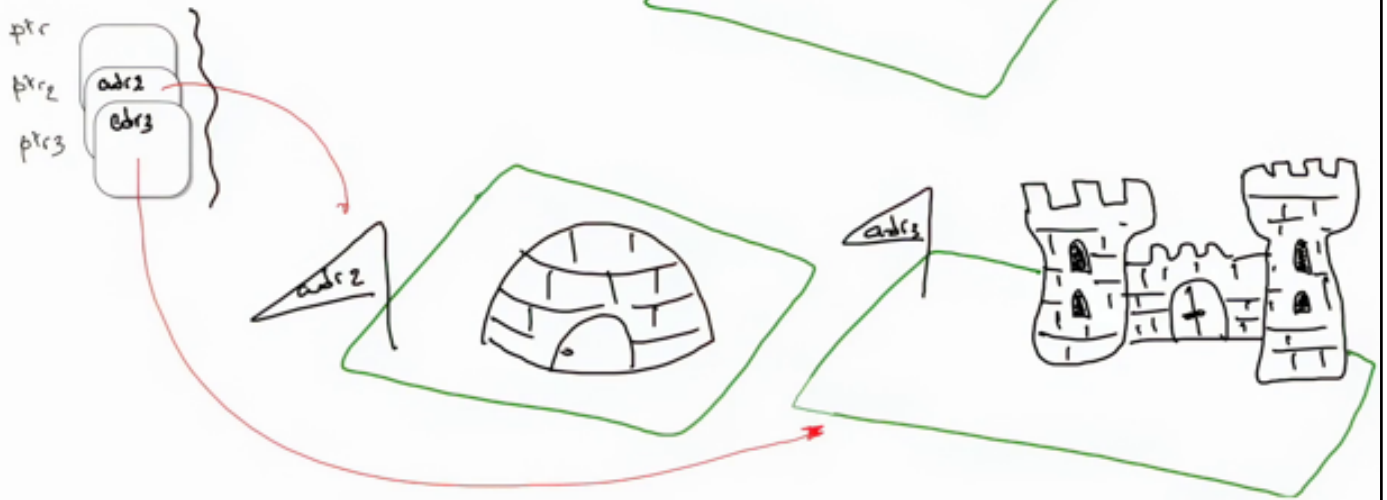
résumé

1m 13s



→ int * ptr (new int(3)) ;

delete ptr;
ptr = nullptr;



qui n'est plus pertinent, qui n'est plus utile.

notes

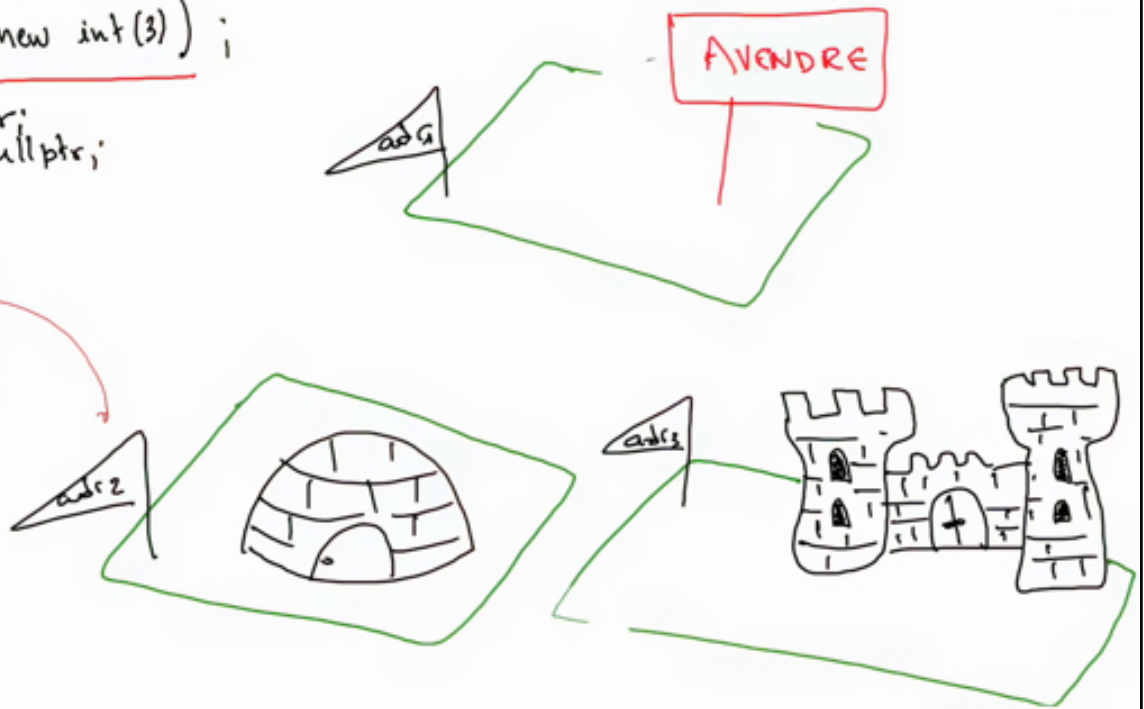
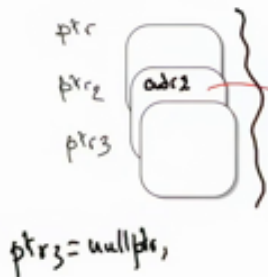
résumé

2m 1s



```
int * ptr (new int(3)) ;
```

```
delete ptr;
ptr = nullptr;
```



Si par contre on affecte la valeur `nullptr` sans avoir au préalable désalloué l'emplacement mémoire concerné, eh bien nous allons nous retrouver dans la situation suivante. Supposons que nous fassions sur l'équivalent de cette page du carnet d'adresses, ce qui reviendrait, ici, à écrire "`ptr3 = nullptr;`". Eh bien, nous savons maintenant que cela signifie que nous effaçons cette page, et le lien vers cet emplacement est perdu. Ce qui signifie que je ne peux plus y accéder via cette page du carnet d'adresses, mais l'emplacement est intact, le terrain n'est pas rendu disponible pour quelqu'un d'autre, et est laissé en l'état. Donc, si par exemple, aucune page de mon carnet d'adresses ne référence plus

notes

résumé

2m 4s



