

Support de cours

Cours:

Initiation à la programmation (en C++)

Vidéo:

String - introduction (partie 2)

Concepts (extraits des sous-titres générés automatiquement) :

Aide d'une affectation des variables. Chaîne de caractères. Type string. Valeur de chaîne. Chaîne vide. Nom de famille. Deuxième variable. Affectation suivante. Combinaisons suivantes. Type char. Variables de type string. Nouvelle chaîne de caractères. Dernière affectation. Chaîne. Conversion automatique de cette valeur.



[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/>
page 1/12

string : introduction

(Partie 2)

Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s



Une variable de type `string` déclarée mais non initialisée est automatiquement initialisée à la chaîne vide (`""`).

Et comme pour n'importe quel autre type, une variable de type `string` (qui n'a pas été déclarée comme constante) peut être modifiée par une affectation.

Exemples :

```
→ string chaine ;           // -> chaine vaut ""
→ string chaine2("test");   // -> chaine2 vaut "test"
→ chaine = "test3" ;       // -> chaine vaut "test3"
→ chaine = chaine2 ;       // -> chaine vaut "test"
→ chaine = 'a' ;           // -> chaine vaut "a"
```



Remarque : dans le cas de l'affectation d'un caractère, la valeur affectée à la chaîne est la chaîne réduite au caractère affecté.

Comme n'importe quel autre type, on peut modifier à l'aide d'une affectation des variables déclarées de type `string`. Alors par exemple, ici je commence par déclarer une variable qui s'appelle "chaine", de type `string`, et j'ai donné aucune valeur pour initialiser, je vous rappelle que dans ce cas, "chaine" va être initialisé à la chaîne vide. Ensuite, ici, je déclare une deuxième variable toujours de type `string`, qui s'appelle "chaine2" "chaine2" est initialisée à la chaîne "test". Ensuite, ici, je fais une affectation qui va changer la valeur de chaine en mettant "test3" dans "chaine". Dans cette affectation suivante, je copie la valeur de "chaine2" dans "chaine". Alors "chaine2" contient "test" ce "test3" ici va être remplacé par la chaîne de caractères "test" et dans cette dernière affectation, j'affecte le caractère 'a' à la variable "chaine" c'est-à-dire celle-ci. Alors il s'agit bien d'un caractère

notes

résumé

0m 1s



La **concaténation de chaînes** est représentée par l'opérateur **+**.

chaîne1 + chaîne2 correspond à une **nouvelle chaîne** associée à la valeur littérale constituée de la concaténation des valeurs littérales de **chaîne1** et de **chaîne2**.

Exemple : constitution du nom complet à partir du nom de famille et du prénom :

```
→ string nom;  
→ string prenom;  
→ string famille;  
→ nom = famille + " " + prenom;
```



parce que j'ai utilisé des simple quotes ou des apostrophes pour délimiter les caractères. Mais le C++ va faire une conversion automatique de cette valeur qui est de type char en une valeur qui est de type string, c'est-à-dire en une chaîne de caractères que je peux représenter de cette façon-ci. Donc maintenant, après cette dernière affectation, "chaîne" contient la chaîne "a". On peut concaténer deux chaînes, c'est-à-dire mettre à bout ces deux chaînes, à l'aide de l'opérateur + Par exemple, si "chaîne1" et "chaîne2" sont deux variables de type string, je peux écrire ceci, qui me donnera une nouvelle chaîne de caractères qui commencera par les caractères de "chaîne1" et qui finira par les caractères de "chaîne2". Par exemple, dans ce code je déclare trois variables de type string qui sont pour l'instant initialisées à la chaîne vide. Supposons qu'ici j'aurais donné une valeur à la variable "prenom" par exemple "Jean" et un nom de famille dans la variable "famille". Quand j'exécute cette affectation, je vais d'abord concaténer

notes

résumé

1m 13s



Les combinaisons suivantes sont possibles pour la concaténation de deux chaînes :

	string + string	
string + "..."		"..." + string
string + char		char + string

Exemple revisité (avec `char`) :

```
string nom;  
string prenom;  
string famille;  
...  
nom = famille + ' ' + prenom;
```

la valeur de la variable "famille" avec cette chaîne de caractères qui contient simplement un espace. Ça va donc me donner "Dupont" suivi d'un espace que je vais représenter comme ceci. Et je vais ensuite ajouter le contenu de la valeur "prenom" c'est-à-dire "Jean". Tout ceci est une nouvelle chaîne de caractères que je vais mettre dans la variable "nom" au moment de l'affectation. Les combinaisons suivantes sont possibles

notes

résumé

2m 37s



Les combinaisons suivantes sont possibles pour la concaténation de deux chaînes :

<u>string</u> + <u>string</u>	
<u>string</u> + <u>"..."</u>	<u>"..."</u> + <u>string</u>
string + char	char + string

Exemple revisité (avec `char`) :

```
string nom;  
string prenom;  
string famille;  
...  
nom = famille + ' ' + prenom;
```

pour la concaténation de deux chaînes. Je peux concaténer deux variables de type `string`. Mais je peux aussi concaténer une variable de type `string` avec une valeur littérale de type `string`. Ou je peux commencer par la valeur littérale suivie de la variable de type `string`.

notes

résumé

3m 13s



Les combinaisons suivantes sont possibles pour la concaténation de deux chaînes :

string + string
string + "..." "..." + string
string + char char + string

Exemple revisité (avec `char`) :

```
string nom;  
string prenom;  
string famille;  
...  
nom = famille + ' ' + prenom;
```

Je peux également commencer par une variable de type `string` et lui ajouter un caractère. Ou je peux commencer par le caractère

notes

résumé

3m 34s



Les combinaisons suivantes sont possibles pour la concaténation de deux chaînes :

string + string
string + "..." "..." + string
string + char char + string

Exemple revisité (avec `char`) :

```
string nom;  
string prenom;  
string famille; " " string  
...  
nom = famille + ' ' + prenom;
```

et le concaténer avec une variable de type `string`. Si je reprends l'exemple du transparent précédent, où j'avais utilisé une valeur littérale de type `string` pour représenter l'espace

notes

résumé

3m 45s



Les combinaisons suivantes sont possibles pour la concaténation de deux chaînes :

string + string
string + "..." "..." + string
string + char char + string

Exemple revisité (avec `char`) :

```
string nom;  
string prenom;  
string famille; " " string  
...  
nom = famille + (' ') + prenom;
```

que je voulais insérer entre le nom de famille et le prénom, et bien en fait je peux aussi utiliser un caractère. C'est ce que j'ai fait ici

notes

résumé

4m 1s



Les combinaisons suivantes sont possibles pour la concaténation de deux chaînes :

string + string
string + "..." "..." + string
string + char char + string

Exemple revisité (avec `char`) :

```
string nom;  
string prenom;  
string famille; " " string  
...  
nom = famille + ' ' + prenom;
```

en utilisant des simples quotes.

notes

résumé

4m 11s



Remarque : les concaténations de la forme string+char (resp. char+string) constituent un moyen très pratique pour ajouter des caractères à la fin (resp. au début) d'une chaîne.

Exemple : Ajout d'un 's' final au pluriel :

```
string reponse("solution");  
...  
if (n > 1) {  
    reponse = reponse + 's';  
}  
...
```

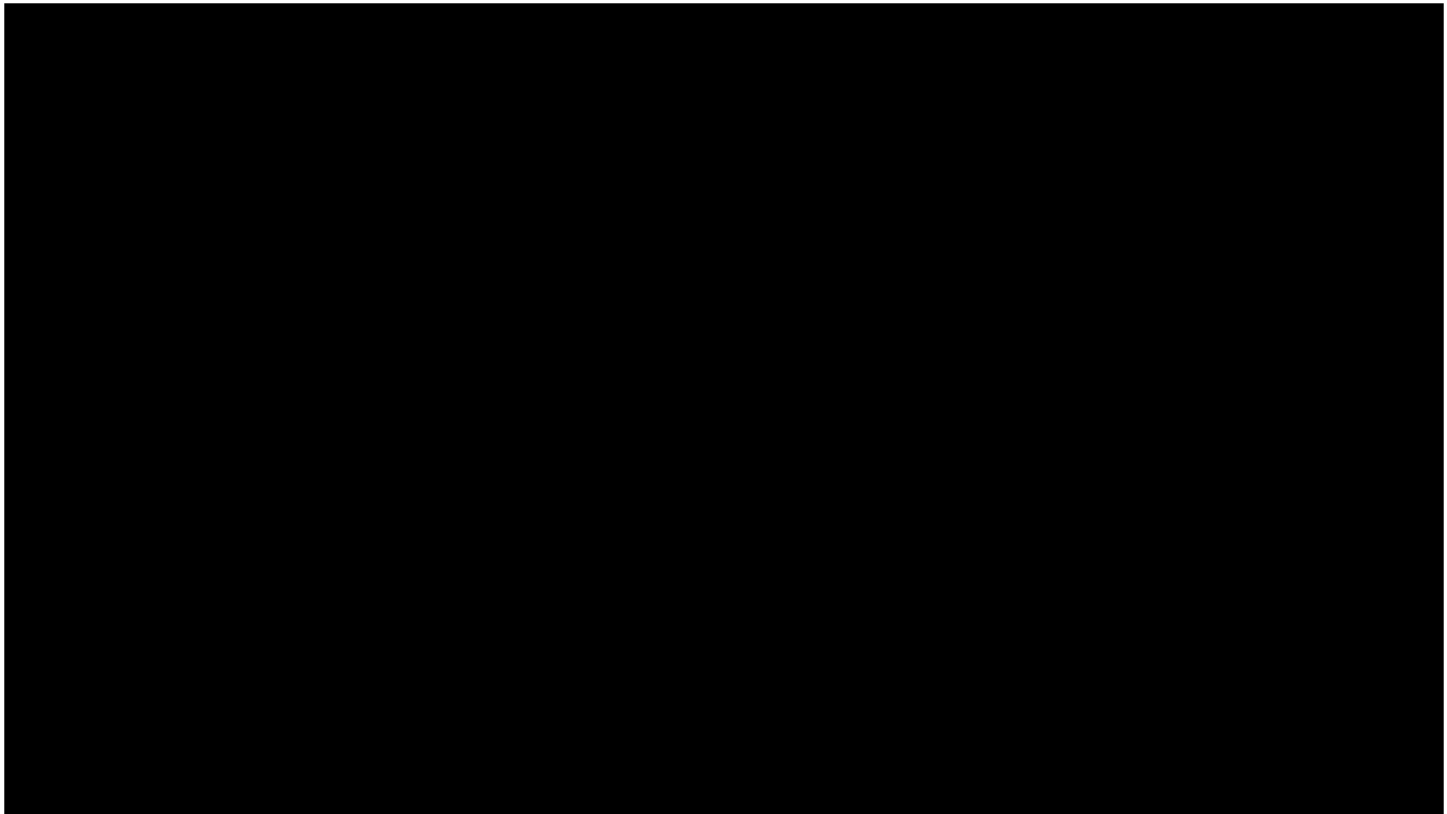
Cette chose-ci est une valeur littérale de type char. Et j'ai parfaitement le droit de concaténer un char avec une variable de type string. C'est ce qu'on voit ici. Les concaténations de la forme string+char sont très pratiques pour pouvoir ajouter des caractères à la fin d'une chaîne de caractères. Par exemple, je déclare ici

notes

résumé

4m 15s





une variable "reponse" qui contient la chaîne de caractères "solutions" et quand une variable "n" est supérieure à 1, j'ai envie de mettre "solutions" au pluriel. Pour cela je vais utiliser cette affectation : "reponse" contient pour l'instant "solutions". Et je vais lui concaténer le caractère 's'. Je vais donc obtenir la chaîne "solutions" avec un "s" à la fin. Et je vais mettre le résultat dans ma variable "reponse". C'est-à-dire que dans "reponse", je vais obtenir "solutions" au pluriel. Et on peut utiliser exactement le même truc pour ajouter des caractères au début d'une chaîne. au début d'une chaîne.

notes

résumé

4m 37s