

Les chaînes de caractères en détail

Fondamentalement, l'informatique englobe tout ce qui concerne le traitement de l'information. Vous avez vu dans les chapitres précédents les bases du calcul numérique. Un autre type de données qui souvent manipulé en informatique sont les chaînes de caractères. Les chapitres suivants détaillent leurs manipulations en C++.

Rappel sur les chaînes de caractères

Les chaînes de caractères en C++ sont manipulées via la classe `std::string` de la bibliothèque standard. L'initialisation ou la modification d'une chaînes se fait en utilisant l'initialisation avec des crochets et l'opérateur d'affectation `=` :

main.cpp

```
#include <iostream>
#include <string>

int main() {
    std::string const s1 { "hello, world!" }; //
    initialisation avec une littérale
    std::string s2 {}; //
    initialisation par défaut
    s2 = "bonjour tout le monde !"; // affectation

    std::cout << "hello, world!" << std::endl; // afficher
    une littérale
    std::cout << s2 << std::endl; // afficher
    une variable
}
```

Un point important concernant les littérales chaînes de caractères : elles ne sont pas de type `string`, mais du type `const char *` hérité du C. L'utilisation de ce type n'est pas recommandé, sauf pour initialiser une

chaîne de type `string`. La classe `string` apporte des garanties plus forte que `const char *` (fuite mémoire) et offre beaucoup plus de fonctionnalités (que vous avez voir dans la suite de ce chapitre et dans les chapitres suivants).

Les littérales string

Dans la prochaine norme du C++14, il sera possible de créer directement un littérale chaîne de caractères de type `string`, en ajoutant le suffixe "s" après la chaîne. Cela permettra d'éviter les conversions de `'const char *` en `string` et permettra d'utiliser directement `auto` :

```
string s1 { "hello, world!"s };  
auto s2 = "hello, world!"s;
```

Hors sujet : les fonctions membres et le fonctions non membres

[a voir où mettre cette partie](#)

Fonction : applique un traitement spécifique sur des données.

Membre : de la forme "objet.fonction()"

Non membre : de la forme "fonction()"

Entre parenthèses : paramètres de la fonction

Par exemple :

```
string s;  
s.size(); // membre, retourne taille de la chaine  
  
float f { 123.456 };  
std::exp(f); // non membre, calcul exponentielle
```

fonction membre permet de rendre des services liés à la classes, càd de modifier l'objet et d'obtenir des informations

Manipuler directement une chaîne

size, empty, clear, concaténation

Conversion

Chapitre précédent	Sommaire principal	Chapitre suivant
------------------------------------	------------------------------------	----------------------------------

[Cours, C++](#)