

Explorer la documentation

Le langage C++ et la bibliothèque standard sont extrêmement riches, il est difficile (et inutile) d'apprendre toutes les subtilités et détails dès le début. Cela viendra avec l'expérience. Par contre, il est important pour votre apprentissage d'acquérir une certaine autonomie, en particulier pour trouver les informations dont vous aurez besoin. Ce cours vous donnera les éléments de compréhension indispensables, mais c'est dans votre intérêt de ne pas hésiter à faire vos propres recherches et lire des sources complémentaires.

L'une des sources les plus importantes est bien sûr la “documentation”.

Le site cppreference.com

La référence officielle du C++ est le document publié par l'organisme international de normalisation (ISO). Ce document est assez austère à lire (plus de 1300 pages), il est avant tout destiné à ceux qui implémentent les compilateurs. Il est payant, mais vous pouvez consulter le document de travail du comité C++, qui contient également les propositions pour le futur standard ([Working Draft, Standard for Programming Language C++](#)).

En pratique, vous n'allez donc pas utiliser ce document. Il existe différents sites internet qui présentent les spécifications du C++, de façon plus abordable. Dans ce cours, nous allons utiliser le site [cppreference.com]. Ce n'est pas la documentation officielle à proprement parler, mais c'est l'une des plus à jour et complète.

La page principale est divisée en deux parties. En haut, la documentation du C++, et celle du C en dessous. Nous allons bien sûr utiliser celle concernant le C++.

* screenshot page principale

La documentation du C++ se décompose en plusieurs parties. En pratique, on peut distinguer trois parties : le langage proprement dit (<http://en.cppreference.com/w/cpp/language>), les fichiers d'en-tête (<http://en.cppreference.com/w/cpp/header>) et la bibliothèque standard (tous les autres parties en fait).

Faire une recherche dans la documentation

Vous allez utiliser la documentation dans deux situations. Soit vous recherchez les détails techniques d'une fonctionnalités du C++ à partir d'un mot-clé, soit vous recherchez quels outils du C++ permettent de réaliser une idée que vous avez en tête.

Partir d'un mot-clé est beaucoup plus simple. Il suffit de rechercher dans un moteur de recherche ce mot clé (par exemple en tapant "C++11 mot-clé") pour trouver des références sur le sujet (évitiez de rechercher dans un premier temps "C++ mot-clé", il existe beaucoup de références obsolètes sur internet. Ne faites cela que dans un second temps, lorsque la recherche sur le C++11 a échouée).

Lorsque vous lirez ce cours ou lorsque vous poserez des questions sur les forums internet, on vous donnera probablement des mots-clés, comme par exemple `string` pour les chaînes de caractères ou `vector` pour les tableaux.

Sur [cppreference.com](http://en.cppreference.com), vous avez un champ de recherche en haut à droite. Il n'est pas nécessaire de préciser "C++" ici, puisque la recherche se focalisera sur le site. Vous pouvez donc taper directement votre mot-clé. Si vous tapez `vector` par exemple, vous allez obtenir la page principale de `std::vector` : <http://en.cppreference.com/w/cpp/container/vector>. Si vous essayez avec `string`, plusieurs pages peuvent correspondre, vous obtiendrez donc une page listant les différents résultats possibles <http://en.cppreference.com/mwiki/index.php?title=Special%3ASearch&search=string> (pour les chaînes de caractères, c'est la première qui nous intéresse : `std::string` http://en.cppreference.com/w/cpp/string/basic_string).

Trouver la fonctionnalité du C++ correspond à une idée

La documentation du C++ (et la bibliothèque standard) est organisée par thématique. Il suffit généralement d'avoir une idée du module qui sera utilisé pour retrouver facilement le mot-clé correspondant. Mais pour cela, il faut être un peu familier avec l'organisation de la documentation, pour savoir où chercher. Il n'y a pas 36 méthodes pour cela : il faut utiliser la documentation le plus souvent possible, même pour rechercher une fonctionnalité que vous connaissez déjà.

Les titres de modules est suffisamment explicite, nous n'allons pas les détailler maintenant. Ils seront vu tout au long de ce cours, en fonction des besoins. Pour résumer les modules :

- “Utilities library” : différentes fonctionnalités utilitaires, pour gérer les types, la mémoire, les erreurs, les dates et temps, etc. ;
- “Strings library” : les chaînes de caractères ;
- “Containers library” : les collections de données (tableau, liste, etc.) ;
- “Algorithms library” : différents algorithmes génériques ;
- “Iterators library” : fonctionnalités pour manipuler les collections et algorithmes de façon générique ;
- “Numerics library” : tout ce qui concerne la manipulation des nombres, les nombres complexes et les nombres aléatoires ;
- “Input/output library” : entrée et sorties, aussi bien le clavier, que les fichiers ;
- “Localizations library” : prise en charge de l'internationalisation ;
- “Regular expressions library” : expressions régulières ;
- “Atomic operations library” : les atomiques ;
- “Thread support library” : la programmation multi-threads.

Bien sûr, nombreux concepts cités ici vous sont inconnus. Pas de panique, tout sera détaillé dans la suite de ce cours.

Le langage C++

La première partie intéressante est la documentation concernant le langage proprement dit. C'est probablement la partie la plus importante de votre apprentissage (sans de bonnes bases sur le langage, vous ne saurez pas utiliser correctement la bibliothèque standard), mais c'est paradoxalement la plus petite partie de la documentation (comparé à la bibliothèque standard). Et probablement celle que vous utiliserez le moins, tout au moins dans un premier temps.

La raison est simple : comme le langage est indispensable à connaître, tout ce dont vous aurez besoin de connaître sera expliqué dans le cours. Cela ne veut pas dire que vous maîtriserez la totalité du langage dans ce premier cours (il faudra un peu plus de temps), mais que si une fonctionnalité n'est pas abordée dans ce cours, elle ne vous sera probablement pas nécessaire (sauf si vous travaillez sur un code qui n'est provient pas de ce cours ou si vous êtes curieux - mais dans ce cas, ne vous perdez pas trop dans vos explorations, essayez de suivre globalement le parcours proposé dans ce cours).

Au contraire, la bibliothèque standard contient de nombreuses fonctionnalités simples à comprendre (quand on connaît les bases) et qui nécessitent surtout d'apprendre par la pratique. Tous les détails de la bibliothèques standard en seront donc pas détaillés, il vous sera proposé à la place des travaux dirigés pour explorer et utiliser ces fonctionnalités. D'où l'importance de la documentation dans ce cas.

Pour autant, si vous avez la curiosité d'explorer cette page de documentation, ne vous privez pas. Et n'hésitez pas à poser des questions sur les forums. Et pour vous aider à faire vos propres recherches, vous trouverez dans ce cours les mots-clés en anglais correspondant aux notions qui sont présentées.

La liste des fichiers d'en-tête

La page des fichiers d'en-tête <http://en.cppreference.com/w/cpp/header>

contient tous les fichiers de la bibliothèque standard que vous pouvez utiliser avec `#include`. Lorsque vous rencontrez une fonctionnalité, vous pouvez regarder dans le fichier d'en-tête correspondant, pour savoir les fonctionnalités qui sont liées.

Les fichiers d'en-tête commençant par "c" sont en fait des importations de la bibliothèque standard du C, disponible pour des raisons de compatibilité.

Un fichier d'en-tête de la bibliothèque standard du C sera appelée en C en utilisant une syntaxe différente du C++. Le nom ne contient pas le préfixe "c" et contient l'extension ".h" :

```
#include <math.h>
```

Vous trouverez peut-être ce type d'inclusion dans de vieux code C++ sur internet. Ne reprenez pas directement le code à l'identique, mais remplacez par le fichier d'en-tête C++ correspondant (par exemple `<cmath>` dans cet exemple). Votre code pourra fonctionner si vous oubliez de la faire, mais cela posera des problèmes de conflits avec le code C++ sur le long terme.

Savoir lire une page de documentation

Voyons plus en détail le contenu d'une page de la documentation. Pour cela, regardons la page de `std::cout` : <http://en.cppreference.com/w/cpp/io/cout>.

- screenshot

Le premier point important est que les pages sont organisées sous forme hiérarchique (un peu comme les chapitres et sous-chapitres d'un livre). Vous pouvez voir en haut de la page la liste suivante :

- C++
- Input/output library
- `std::basic_ostream`

Cela signifie que `std::cout` est définie dans `std::basic_ostream`, qui fait partie du module “entrée et sortie” de la partie “C++”. Si on souhaite trouver les fonctionnalités similaires, vous pouvez regarder dans `std::basic_ostream` et/ou dans “Input/output library”.

En dessous, vous avez une autre ligne importante, dont on a déjà parlé :

```
Defined in header <iostream>
```

Cela signifie que cette fonctionnalité est définie dans le fichier d'en-tête `<iostream>`. Il suffit donc d'écrire la ligne suivante au début d'un code pour utiliser cette fonctionnalité :

```
#include <iostream>
```

Pour terminer, vous aurez systématiquement une description de la fonctionnalité (en anglais) et des codes d'exemple. Certaines fonctionnalités sont très complexes et très riches, n'hésitez pas à étudier le code fourni si vous avez des problèmes.

Lire l'anglais

Il est possible que vous ne soyez pas habitué à lire l'anglais. Pas de panique, il est possible de s'en sortir autrement... mais ce n'est pas une bonne idée. L'anglais est omniprésent en informatique et plus encore en programmation. Il est classique que la documentation et les articles de référence soient en anglais et il est préférable d'écrire ses codes en anglais aussi.

Si vous lisez ce cours, c'est que normalement vous souhaitez apprendre correctement le C++, ce qui va prendre des années. Et que vous apprenez principalement en autodidacte. Donc vous avez le temps et de bonnes raisons de faire les choses correctement. Même si vous prenez une journée pour lire une simple page (voire moins), ce n'est pas grave. Le principal est de s'y mettre progressivement. Et tout de suite. Cela viendra avec la pratique.

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

Cours, C++