

# A Biblioteca Padrão do C++

# Biblioteca Padrão do C++

Ao compilar no modo padrão, o compilador tem acesso à biblioteca completa especificada pelo padrão C++ . Os componentes da biblioteca incluem o que é informalmente conhecido como **Standard Template Library** (STL), bem como os seguintes componentes: classes de string, classes numéricas ...

O termo **STL** não tem uma definição formal, mas geralmente é entendido como incluindo contêineres, iteradores e algoritmos. O seguinte subconjunto de cabeçalhos de biblioteca padrão pode ser considerado como incluindo o **STL** .

Arquivo de cabeçalho	Descrição
<algorithm>	Algoritmos padrão que operam em contêineres
<bitset>	Sequências de bits de tamanho fixo
<complex>	O tipo numérico que representa números complexos
<deque>	Sequências que suportam adição e remoção em cada extremidade
<exception>	Classes de exceção predefinidas
<fstream>	Stream I / O em arquivos
<functional>	Objetos de função
<iomanip>	manipuladores iostream
<ios>	classes de base iostream
<iosfwd>	Declarações de encaminhamento de classes iostream
<iostream>	Funcionalidade de fluxo I / O básico
<istream>	Fluxos de entrada / saída
<iterator>	Classe para atravessar uma sequência
<limits>	Propriedades dos tipos numéricos
<list>	Sequências ordenadas
<locale>	Suporte para internacionalização

<map>	Recipientes associativos com pares de chave / valor
<memory>	Alocadores de memória especial
<new>	Alocação e desalocação de memória básica
<numeric>	Operações numéricas generalizadas
<ostream>	Fluxos de E / S de saída
<queue>	Sequências que apoiam a adição na cabeça e a remoção na cauda
<set>	Contêiner associativo com chaves exclusivas
<sstream>	Stream I / O usando uma string na memória como fonte ou coletor
<stack>	Sequências que suportam adição e remoção na cabeça
<stdexcept>	Classes de exceção padrão adicionais
<streambuf>	Classes de buffer para iostreams
<string>	Sequências de personagens
<typeinfo>	Identificação do tipo de tempo de execução
<utility>	Operadores de comparação
<valarray>	Matrizes de valores úteis para programação numérica
<vector>	Sequências que suportam acesso aleatório

## Biblioteca Padrão do C++

Existe também a Biblioteca Padrão do C que é utilizada pelo C++ . Onde os cabeçalhos em C são terminados em .h e no C++ retira-se o .h e adiciona a letra c no início, todos os headers separados por categoria pode ser encontrado no endereço: <https://en.cppreference.com/w/cpp/header> . Exemplos:

### Bibliotecas de utilitários

<cstdlib> → Utilitários de uso geral: controle de programa , alocação de memória dinâmica , números aleatórios , classificação e pesquisa

<csignal> → Funções e macro constantes para gerenciamento de sinal

<csetjmp> → Macro (e função) que salva (e salta) para um contexto de execução

<cstdarg> → Tratamento de listas de argumentos de comprimento variável

<typeinfo> → Utilitários de informação de tipo de tempo de execução

<typeindex>(C++ 11) → std::type\_index

<type\_traits>(C++ 11) → Informações de tipo de tempo de compilação

<bitset> → modelo de classe std::bitset

<functional> → Objetos de função, invocações de função, operações Bind e wrappers de referência

<utility> → Vários componentes utilitários

<ctime> → Utilitários de data / hora estilo C

<chrono>(C++ 11) → Utilitários de tempo C++

<cstddef> → Macros e typedefs padrão

<initializer\_list>(C++ 11) → modelo de classe std::initializer\_list

<tuple>(C++ 11) → std::modelo de classe de tupla

<any>(C++ 17) → std::qualquer classe

<optional>(C++ 17) → std::template de classe opcional

<variant>(C++ 17) → modelo de classe std::variant

<compare>(C++ 20) → Suporte ao operador de comparação de três vias

<version>(C++ 20) → Fornece informações de biblioteca dependentes de implementação

<source\_location>(C++ 20) → Suprimentos significa obter a localização do código-fonte

## Biblioteca Padrão do C++

### Bibliotecas de strings

<cctype> → Funções para determinar a categoria de caracteres estreitos

<cwctype> → Funções para determinar a categoria de caracteres largos

<cstring> → Várias funções de manipulação de cadeia de caracteres estreitos

<cwchar> → Várias funções de manipulação de string ampla e multibyte

<cuchar>(C++ 11) → Funções de conversão de caracteres Unicode estilo C

<string> → modelo de classe std::basic\_string

<string\_view>(C++ 17) → modelo de classe std::basic\_string\_view

<charconv>(C++ 17) → std::to\_chars e std::from\_chars

<format>(C++ 20) → Biblioteca de formatação incluindo std::format

### Bibliotecas de contêineres

<array>(C++ 11) → std::array container

<vector> → std::vector container

<deque> → std::deque container

<list> → std::list container

<forward\_list>(C++ 11) → std::forward\_list container

<set> → recipientes associativos std::set e std::multiset

<map> → recipientes associativos std::map e std::multimap

<unordered\_set>(C++ 11) → std::unordered\_set e std::unordered\_multiset contêineres associativos não ordenados

<unordered\_map>(C++ 11) → std::unordered\_map e std::unordered\_multimap contêineres associativos não ordenados

<stack> → std::stack container adapter

<queue> → adaptadores de contêiner std::queue e std::priority\_queue

<span>(C++ 20) → std::span view

## Biblioteca Padrão do C++

### Bibliotecas numéricas

<cmath> → Funções matemáticas comuns

<complex> → Tipo de número complexo

<valarray> → Classe para representar e manipular matrizes de valores

<random>(C++ 11) → Geradores e distribuições de números aleatórios

<numeric> → Operações numéricas em valores em intervalos

<ratio>(C++ 11) → Aritmética racional em tempo de compilação

<cfenv>(C++ 11) → Funções de acesso ao ambiente de ponto flutuante

<bit>(C++ 20) → Funções de manipulação de bits

<numbers>(C++ 20) → Constantes matemáticas

**Veja todas em:**

<https://en.cppreference.com/w/cpp/header>