

# 头文件知识

每个 Object-C/C/C++ 程序模块通常分为两个文件。一个文件用于保存程序的声明（declaration），称为头文件。另一个文件用于保存程序的实现（implementation），称为定义（definition）文件。

## 1、头文件主要包含如下作用：

①通过头文件来调用库功能。在很多场合，源代码不便（或不准）向用户公布，只要向用户提供头文件和二进制的库即可。用户只需要按照头文件中的接口声明来调用库功能，而不必关心接口怎么实现的。编译器会从库中提取相应的代码。

②头文件能加强类型安全检查。如果某个接口被实现或被使用时，其方式与头文件中的声明不一致，编译器就会指出错误，这一简单的规则能大大减轻程序员调试、改错的负担。

③头文件在多个代码文件全局变量(函数)的重用、防止定义的冲突，对各个被调用函数给出一个描述，其本身不需要包含程序的逻辑实现代码，它只起描述性作用，用户程序只需要按照头文件中的接口声明来调用相关函数或变量，链接器会从库中寻找相应的实际定义代码。

④编译时，编译器通过头文件找到对应的函数库，进而把已引用函数的实际内容导出来代替原有函数。进而在硬件层面实现功能。

## 2、什么样的内容适合放在头文件里？

头文件一般包含三部分内容：（1）头文件开头处的版权和版本声明。（2）预处理块与宏定义。（3）函数和类结构声明等。

对于具有外部存储类型的标识符，可以在其他任何一个源程序文件中经声明后引用，因此用户完全可以将一些具有外部存储类型的标识符的声明放在一个头文件中。具体地说，头文件中可以包括：用户构造的数据类型（如枚举类型），外部变量，外部函数、常量和内联函数等具有一定通用性或常用的量。而一般性的变量和函数定义不宜放在头文件中。例如：`#include<stdio.h>`中的头文件 `stdio.h` 作用是让链接器通过头文件里的函数声明找到函数实际代码所在的位置即所在的库文件，这样才能使用该函数的实际代码，函数的实际代码的实现过程是先让链接器通过头文件里函数的申明找到函数实际代码所在的位置即所在的库文件，再通过`#include` 语句把链接器所找到的函数实际代码用链接器把函数的实际代码链接到当前文件即所要执行的程序中。当然有些函数的使用不需要提供头文件，但是在 ISO/ANSI C 已经对有些函数的使用必须提供哪些头文件制定了标准。

## 3、常用头文件

传统 C++

1	<code>#include&lt;assert.h&gt;</code> //设定插入点
2	<code>#include&lt;ctype.h&gt;</code> //字符处理
3	<code>#include&lt;errno.h&gt;</code> //定义错误码
4	<code>#include&lt;float.h&gt;</code> //浮点数处理
5	<code>#include&lt;fstream.h&gt;</code> //文件输入/输出
6	<code>#include&lt;iomanip.h&gt;</code> //参数化输入/输出
7	<code>#include&lt;iostream.h&gt;</code> //数据流输入/输出
8	<code>#include&lt;limits.h&gt;</code> //定义各种数据类型最值常量
9	<code>#include&lt;locale.h&gt;</code> //定义本地化函数

10	#include<math.h>//定义数学函数
11	#include<stdio.h>//定义输入/输出函数
12	#include<stdlib.h>//定义杂项函数及内存分配函数
13	#include<string.h>//字符串处理
14	#include<strstrea.h>//基于数组的输入/输出
15	#include<time.h>//定义关于时间的函数
16	#include<wchar.h>//宽字符处理及输入/输出
17	#include<wctype.h>//宽字符分类

标准 C++

1	#include<algorithm>//STL 通用算法
2	#include<bitset>//STL 位集容器
3	#include<cctype>//C 字符处理
4	#include<cerrno>//C 的错误报告机制
5	#include<locale>
6	#include<cmath>//兼容 C 语言数学库
7	#include<complex>//复数类
8	#include<cstdio>//C 语言输入输出工具
9	#include<cstdlib>//C 语言通用工具
10	#include<cstring>//C 字符串
11	#include<ctime>
12	#include<deque>//STL 双端队列容器
13	#include<exception>//异常处理类
14	#include<fstream>//文件输入输出流
15	#include<functional>//STL 定义运算函数（代替运算符）
16	#include<limits>
17	#include<list>//STL 线性列表容器
18	#include<map>//STL 映射容器
19	#include<iomanip>
20	#include<ios>//基本输入/输出支持
21	#include<iosfwd>//输入/输出系统使用的前置声明
22	#include<iostream>//基本输入输出流
23	#include<queue>//STL 队列容器
24	#include<set>//STL 集合容器
25	#include<sstream>//基于字符串的流
26	#include<stack>//STL 堆栈容器
27	#include<stdexcept>//标准异常类
28	#include<streambuf>//底层输入/输出支持
29	#include<string>//字符串类
30	#include<utility>//STL 通用模板类
31	#include<vector>//STL 动态数组容器
32	#include<wchar>
33	#include<wctype>

在 C++中，标准库的命名空间为 std，因而包含了上述头文件时，一般会使用下列语句：

1	using namespace std;
---	----------------------

C99 版本

1	#include<complex.h>//复数处理
2	#include<fenv.h>//浮点环境
3	#include<inttypes.h>//整数格式转换
4	#include<stdbool.h>//布尔环境
5	#include<stdint.h>//整型环境
6	#include<tgmath.h>//通用类型数学宏