《现代C++编程实现》电子版：https://leanpub.com/c01

C++语言介绍

hwdong
董洪伟

YouTube频道： hwdong

博客： hwdong-net.github.io

https://github.com/hwdong-net/cplusplus17

目录

- 为什么要学习C++语言?
- 为什么要学习现代C++?
- 教材说明
- 程序和编程语言
- C++程序的编译与运行

为什么要学习C++?

hwdong
董洪伟

YouTube频道： hwdong

博客： hwdong-net.github.io

C++语言

- C++语言是对C语言的扩展和增强: **面向对象**、**通用算法**（泛型编程）
- 1979年，贝尔实验室Bjarne Stroustrup(C++之父，现在Texas A&M)发明。
- 1998年被ISO委员会批准，2003发布了修正版。2011开始增加了很多新特征：C++11、C++14、C++17、C++20、C++23。

C/C++：**底层控制、性能王者**

C++ 是从 C 语言发展而来的，继承了 C 语言强悍的高性能和底层控制能力

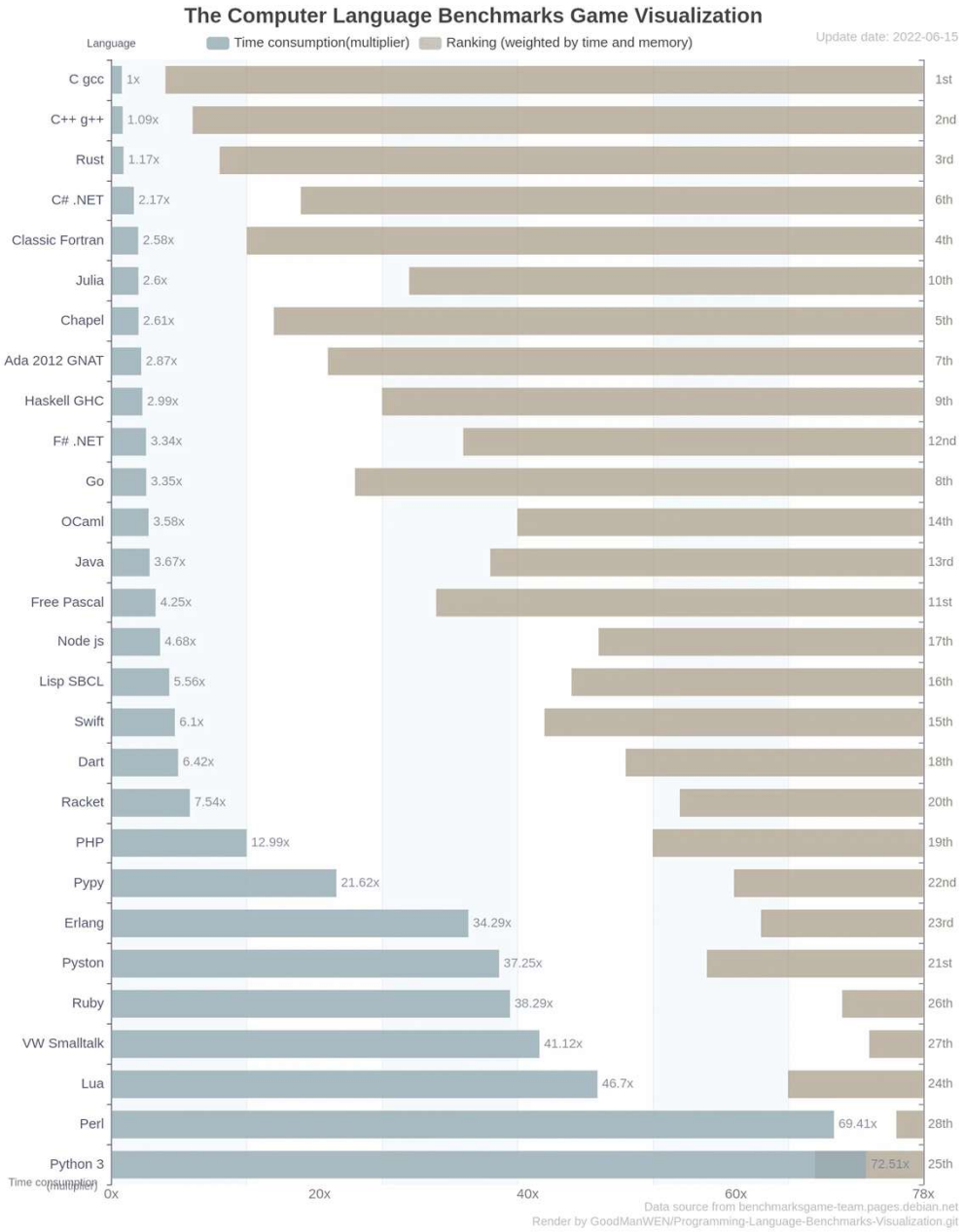
- 高效执行：C++ 代码可以直接编译为高效的机器码，运行速度快，适用于对性能要求极高的场景。
- 手动内存管理：C++ 允许开发者直接管理内存（new/delete、指针操作），可以优化资源使用，提高程序的效率。
- 底层硬件控制：C++ 可以直接访问 CPU 寄存器、内存、IO 端口，非常适合系统编程、驱动开发和嵌入式系统。

- **运行速度对比（C++ vs Java）**：

任务	C++耗时（秒）	Java耗时（秒）
regex-redux	1.83	10.50
n-body	9.42	22.00
binary-trees	3.68	8.39

- **游戏开发性能对比**（FPS）[7]：

框架	C++ (Unreal)	C# (Unity)	Python (Pygame)
平均FPS	60	45	15


The Computer Language Benchmarks Game Visualization
Update date: 2022-06-15
Language
Time consumption(multiplier)
Ranking (weighted by time and memory)
C gcc 1x 1st
C++ g++ 1.09x 2nd
Rust 1.17x 3rd
C# .NET 2.17x 6th
Classic Fortran 2.58x 4th
Julia 2.6x 10th
Chapel 2.61x 5th
Ada 2012 GNAT 2.87x 7th
Haskell GHC 2.99x 9th
F# .NET 3.34x 12nd
Go 3.35x 8th
OCaml 3.58x 14th
Java 3.67x 13rd
Free Pascal 4.25x 11st
Node js 4.68x 17th
Lisp SBCL 5.56x 16th
Swift 6.1x 15th
Dart 6.42x 18th
Racket 7.54x 20th
PHP 12.99x 19th
Pypy 21.62x 22nd
Erlang 34.29x 23rd
Pyston 37.25x 21st
Ruby 38.29x 26th
VW Smalltalk 41.12x 27th
Lua 46.7x 24th
Perl 69.41x 28th
Python 3 72.51x 25th
Time consumption (multiplier)
0x
20x
40x
60x
78x
Data source from benchmarksgame-team.pages.debian.net
Render by GoodManWEN/Programming-Language-Benchmarks-Visualization.git

全能王：从系统软件到应用软件，无所不能

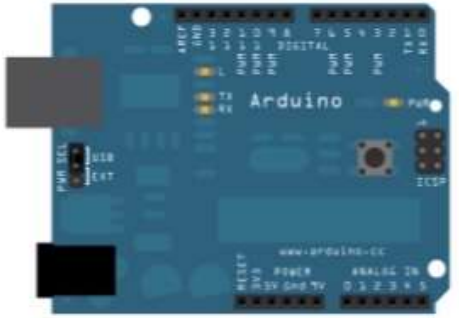
由于其高性能和底层控制能力，C++ 被广泛用于开发各种关键软件：

- 操作系统（如 Windows 部分内核、Linux 组件）。
- 数据库系统（如 MySQL、MongoDB）。
- 网络和浏览器（如 Chrome、Firefox 内核）。
- 图形与多媒体（如 OpenGL、Blender）。
- 游戏和图形引擎（如 Unreal Engine、Unity）。
- 高性能计算（如 TensorFlow、CUDA 计算库）。

- 硬件驱动和操作系统

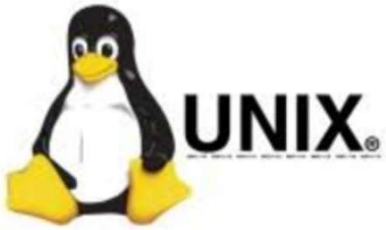
ARM嵌入式

单片机



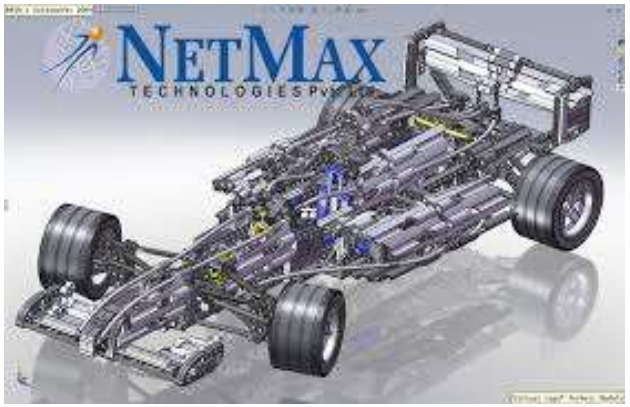
Arduino

- 硬件驱动和操作系统



- 应用软件：游戏、 CAD 专业软件

动漫制作



CAD

游戏

- 应用软件：图形计算、高性能计算、人工智能

Tensorlow

Pytorch
Caffe2

- 应用软件：聊天软件

QQ

不可替代的应用场景：嵌入式、实时、高性能

- 在嵌入式系统（如智能手表、工业机器人）、实时系统（如自动驾驶）、图形计算（如OpenCV、Unreal引擎）等领域，C/C++是唯一选择！

培养严谨的编程思维：底层逻辑、高层思想

- 它要求开发者手动管理内存（如new/delete）、理解指针操作、掌握多范式编程（面向对象、泛型、过程式）。
- 这种"知其所以然"的学习过程，能帮助程序员建立对计算机系统架构的深刻认知。
- 许多编程语言（如Java、 Python）的底层实现都采用C/C++，学好C++等于掌握了理解其他语言的钥匙。

一句话：提升编程内功

为什么学习现代 C++?

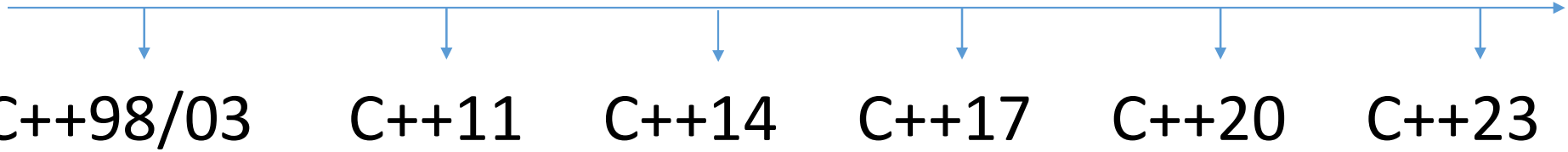
C++11 及更高版本的重要性

```
// 传统 C++
std::vector<int>::iterator it = myVector.begin();

// 现代 C++ (使用 auto)
auto it = myVector.begin();
```

C++ 的演进

- C++ 并非一成不变，而是在不断发展！



- 现代 C++ (C++11 及更高版本) 带来了革命性的改变。
- 掌握现代 C++ 不仅能提升开发效率，还能构建更健壮、更安全、性能更优的应用程序。

更简洁、更易读的代码

- **auto 类型推断**: 简化变量声明，减少冗余。

```
// 传统 C++
std::vector<int>::iterator it = myVector.begin();

// 现代 C++ (auto)
auto it = myVector.begin();
```

更简洁、更易读的代码

- **范围 for 循环**: 简化容器遍历，告别迭代器。

```
// 传统 C++
for (std::vector<int>::iterator it = myVector.begin(); it !=
                                        myVector.end(); ++it) {
    std::cout << *it << std::endl;
}

// 现代 C++ (range-based for loop)
for (int element : myVector) {
    std::cout << element << std::endl;
}
```

更高性能的应用程序

- **移动语义**: 避免不必要的拷贝，大幅提升性能。

  资源转移，而非复制！

- **emplace_back/emplace**: 直接构造对象，避免临时对象。
- **constexpr**: 编译时计算，减少运行时开销。

更安全的内存管理

- **智能指针**: 自动管理内存，告别内存泄漏和悬挂指针。

  unique_ptr: 独占所有权

  shared_ptr: 共享所有权

  weak_ptr: 观察者指针

- **nullptr**: 类型安全的空指针，避免与整数 0 混淆。

拥抱现代化和标准化

- **标准库扩展**: 更多、更强大的标准库组件 (例如 thread, chrono, random, regex)
- 与时俱进: C++ 持续发展，不断引入新特性。
- 行业需求: 越来越多的公司要求使用现代 C++。

提升开发效率

- **代码简化**: 减少代码量，提高开发速度。
- **更强大的工具**: 标准库提供更强大的工具，减少重复造轮子。
- **更好的可维护性**: 简洁的代码更容易维护，降低维护成本。

为什么学习现代 C++?

- 更简洁、更高效、更安全、更易于维护的代码。
- 提高开发效率，降低维护成本。
- 保持与技术发展同步，提高职业竞争力。
- 立即行动，拥抱现代 C++!

C++17从入门到精通

清华大学出版社 2019年8月

YouTube频道： hwdong

博客： hwdong-net.github.io

C++17从入门到精通

- 教材： C++17从入门到精通. 董洪伟，清华大学出版社,2019.8

C++编程语言具有“可操纵底层硬件”、“程序效率高”和“面向对象”的优势。

被广泛应用于系统软件和应用软件的开发，不但是企业界开发重量级软件或平台的首选语言，也是国内外高校广泛采用的计算机编程教学语言，更是衡量一个程序员功力的标尺。

C++17从入门到精通

- 教材： C++17从入门到精通. 董洪伟，清华大学出版社,2019.8

  尽管企业界早已经使用最新的C++11/14/17标准，国内高校仍然延用的是传统的过时的C++98标准，已经和业界普遍使用的现代C++语法标准有很大的脱节。

  网上课程甚至还有VC6、C++Builder、VS2008等环境

C++17从入门到精通

- 教材： C++17从入门到精通. 董洪伟，清华大学出版社,2019.8

  市场上还未见到国内作者编写的现代C++语言教材。
  国外教材如C++编程语言(第4版)、C++Primer5等大部头、琐碎语法、缺少实战例子、C++14标准。

  国外作者的思维模式和语言文化差异使得这些书难以被国内读者特别是初学者阅读理解

C++17从入门到精通

- 教材：C++17从入门到精通. 董洪伟，清华大学出版社,2019.8

  针对没有任何编程基础的学生，直接讲解最新的C++17标准，避免传统的“从C到C++”的教学模式和国内高校采用的过时的C++98标准语法。

  突出重点，讲解主要的常用语法而不是面面俱到，尽量用浅显易懂的例子说明语法概念，避免空洞和冗长的描述。

  丰富的实战案例：游戏、数据结构、信息管理、机器学习等。避免“只会考试而不会编程”的普遍问题

C++17从入门到精通

- 教材： C++17从入门到精通. 董洪伟，清华大学出版社,2019.8

本书资源： https://github.com/hwdong-net/cplusplus17
作者博客： https://hwdong-net.github.io
Youtube频道： https://www.youtube.com/c/hwdong

程序和编程语言

hwdong
董洪伟

YouTube频道： hwdong

博客： hwdong-net.github.io

程序

- 程序 = 一系列对数据加工的指令

做饭程序：

从容器（米桶）取出米，放入洗米盆
用自来水对洗米盆的米进行冲洗
如果(电饭锅没洗净)
　清空洗净电饭锅
打开电饭锅盖，将米和水放入电饭锅
插上电源，按下开关
饭好后，拔下电源，任务结束

计算机程序：

计算机指令(语句)1
计算机指令(语句)2
…
计算机指令(语句)n

指令

- 算术：加，减，乘或除数。这些通常被称为**算术操作**。
- 比较：比较两个数字，看哪个更大，或者它们是否相等。这些通常被称为**逻辑操作**。
- 分支：跳转到程序中的另一条指令，并从那里继续。这些通常被称为**控制语句**。

计算机的组成

- 输入：允许计算机从用户接收信息的任何设备。这包括键盘，鼠标，扫描仪和麦克风。
- 处理：处理信息的计算机组件。中央处理单元（CPU）、图形处理单元(简称GPU)。
- 存储：存储信息的组件。内存（ROM、RAM）、外存（磁带、磁盘、优盘）
- 输出：用于向用户显示信息的任何设备。这包括显示器，扬声器和打印机。

CPU

内存条

中央处理单元（CPU）

是计算机的大脑，负责计算、处理数据、控制其他设备等

- 算术/逻辑单元（ALU）执行算术和比较操作。
- 控制单元确定下一个要执行的指令。
- 寄存器形成一个高速存储区以保存临时结果。

远程服务器

自动售票机

存储

处理

输入

输出

自动售票机

- 输入：投币口和选择按钮是自动售票机的输入设备。
- 处理：当您进行选择时，自动售票机会执行以下几个步骤：验证是否有满足条件的票、验证身份信息、检查和验证是否收到足够的资金、修改数据库、计算差额。执行所有这些步骤的机器部分可以被认为是处理器。
- 输出：自动售票机显示结果、打印票。
- 记忆：自动售票机需要在某个地方保存诸如票的库存、价格等信息。

算法、程序和编程

- 算法是完成某个任务或解决某个问题的一系列步骤（指令）。

　如一道菜(满汉全席)的制作过程说明、祖冲之计算圆周率的方法等。

- 程序和编程：

　程序就是算法在计算机中的表示和实现。

　编程就是如何用计算机的指令来表示算法，即将算法转换成计算机可以执行的程序。

二进制

- 晶体管只有“开”和“关”两种状态，1个晶体管元器件只能表示1位二进制数（0或1），称为1**比特**（**Bit**）或1位
- 8个元器件可以表示8位二进制数字，即可表示有2^8个不同的数值。8位二进制数，称为1**字节**（**Byte**）。
- 16个元器件可以表示16位二进制数，即2Byte，…。

  8×1024个元器件就可以表示1024Byte，即1KB

  8×1024×1024个元器件就可以表示1024KB，即1MB

  8×1024×1024×1024个元器件就可以表示1024MB，即1GB

编程语言

- 表示指令和数据的规则。如同人类语言：英语、汉语都有一套语法规则。

- 机器语言、汇编语言、高级语言

机器语言(machine language)

- 机器语言是用0和1表示指令和数据，因为计算机只能识别0和1。这种二进制代码表示的计算机能直接识别和执行的一种机器指令集合，称为**机器语言**。
- 下面是将17和20相加的机器指令（采用Intel 8086机器语言，Intel Pentium机器语言的子集）：

  1011 0000 0001 0001

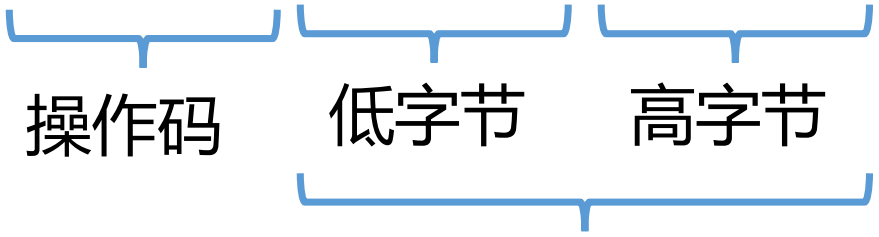
  0000 0100 0001 0100

  1010 0010 0100 1000 0000 0000

汇编语言

- 机器语言的字符化表示，每个汇编指令对应一个机器语言指令

1011 0000 0001 0001
0000 0100 0001 0100
1010 0010 0100 1000 0000 0000

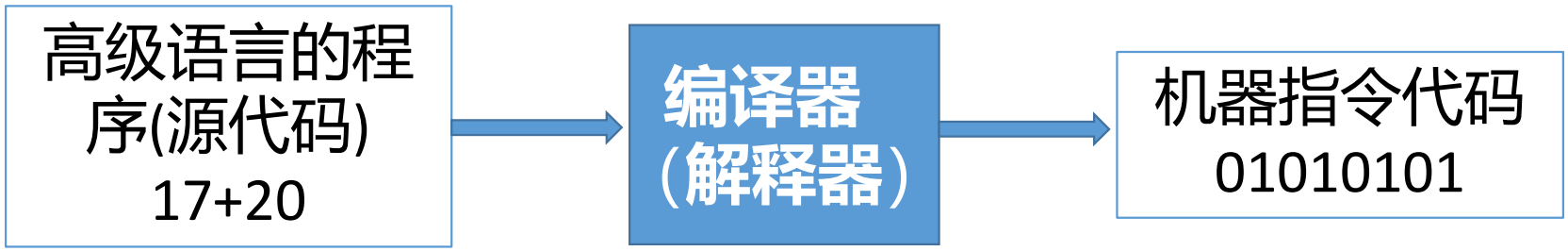


MOV AL, 17D
ADD AL, 20D
MOV [0048h], AL

高级语言

- 用类人类语言表示指令和数据。
- **编译器或解释器**：将高级语言编程的程序转化为机器指令程序

sum = 17 + 20



程序开发步骤

- 理解问题: 是一个什么样的问题？输入数据是什么？要产生什么结果?
- 提出算法：解决这个问题的指令（步骤）序列
- 编写程序：将算法转换成某种编程语言的程序
- 测试：各种可能性的不同的输入，是否产生预期的结果

“计算一组数值的平均值”

- 理解问题：问题含义、数据表示、输入输出
- 提出算法：

  用2个数值分别表示总和和数值的个数；

  将输入的这些数累加到总和上；

  最后除以数值的个数，得到平均值。

“计算一组数值的平均值”

- 伪代码表示算法：

总和 sum=0.
计数器为 count=0.
重复：
    读一个值，
    如果读取值失败，结束这个“重复”过程
    否则:
        将读取的值 value 加到 sum。
        计数器增加 1.即

通过“总和”和“计算器”相除得到平均值
显示/打印平均值

“计算一组数值的平均值”

- 编写程序： 将算法某种编程语言如C/C++/Python语言表示出来
- 测试： 输入不同的测试数据，看看结果是否正确？输入非法数据会怎么样？

C++程序的编译和运行

hwdong
董洪伟

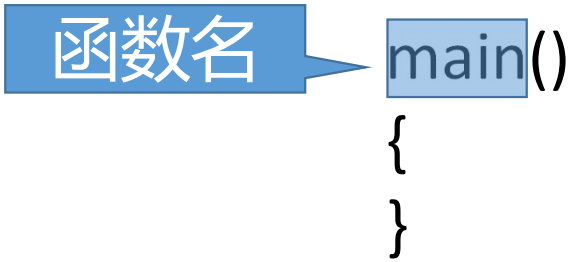
最简单的C++程序

- C++是对C语言的扩展，C语言程序也是C++程序。
- C/C++程序只执行叫做main()的主函数。

```
main()
{
}
```

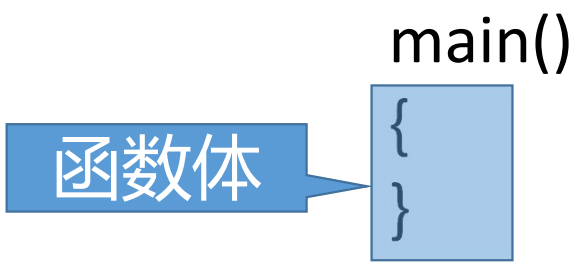
最简单的C++程序

- C++是对C语言的扩展，C语言程序也是C++程序。
- C/C++程序只执行叫做main()的主函数。



最简单的C++程序

- C++是对C语言的扩展，C语言程序也是C++程序。
- C/C++程序只执行叫做main()的主函数。



最简单的C++程序

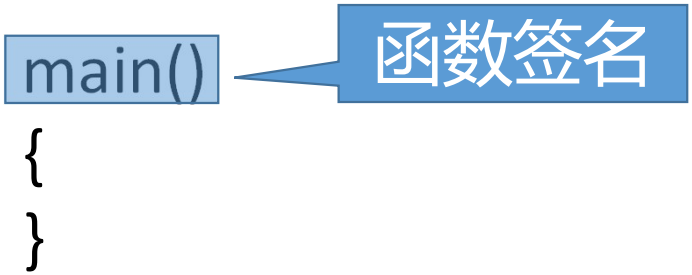
- C++是对C语言的扩展，C语言程序也是C++程序。
- C/C++程序只执行叫做main()的主函数。

```
main()
{
}
```

参数列表

最简单的C++程序

- C++是对C语言的扩展，C语言程序也是C++程序。
- C/C++程序只执行叫做main()的主函数。
- 函数名和参数列表构成了函数签名。



return关键字

- 结束函数执行，返回(return)到调用者。

```
int main()
{
    return 0;
}
```

return结束函数执行，返回到调用者

return关键字

- 结束函数执行，返回(return)到调用者。
- main()函数的调用者是操作系统。

```
int main()
{
    return 0;
}
```

return结束函数执行，返回到调用者

return关键字

- 结束函数执行，返回(return)到调用者。
- main()函数的调用者是操作系统。
- return可以返回一个结果给调用者。

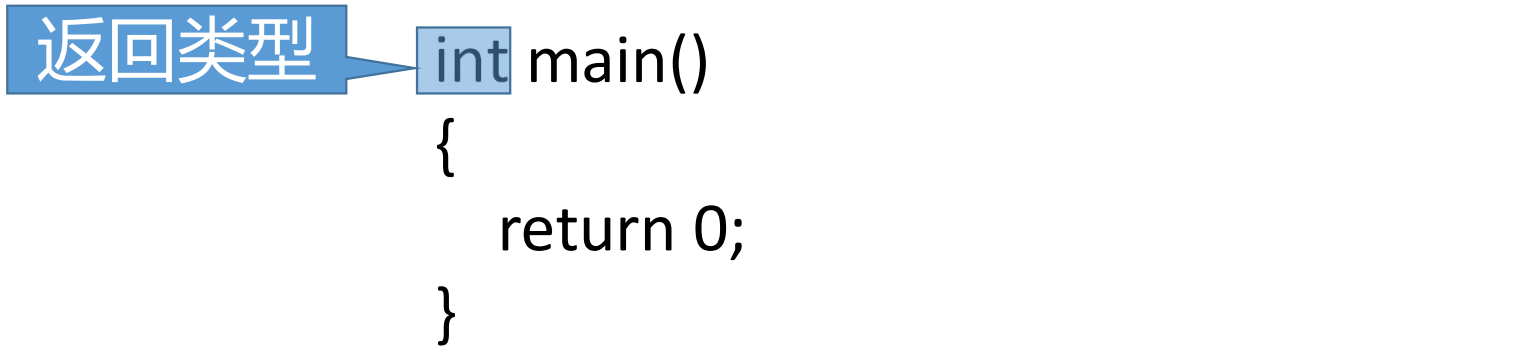
```
int main()
{
    return 0;
}
```

返回0给操作系统

返回0表示程序没有错误，返回某个非0 数表示某种错误码

return关键字

- 结束函数执行，返回(return)到调用者。
- main()函数的调用者是操作系统。
- return可以返回一个结果给调用者。
- 函数名前的int表示返回结果的数据类型是int（整数类型）



输出

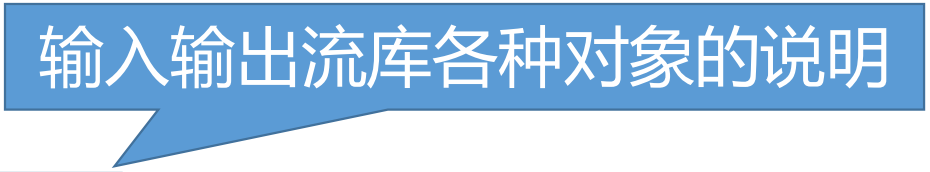
- 程序：输出一个字符串“hello world”

```
#include <iostream>
int main()
{
    std::cout<<“hello world!”;
}
```

输出

- C++**标准输入输出流库**包含了已经编写好的输入输出工具

  预处理指令#include用于包含头文件： iostream




```
#include <iostream>
int main()
{
    std::cout<<“hello world!”;
}
```

输出

- C++标准输入输出流库包含了已经编写好的输入输出工具
- 输出流对象cout表示控制台窗口，输出运算符<<

```
#include <iostream>
int main()
{
    std::cout << “hello world!”;
}
```

将双引号包围的字符串输出到cout代表的控制台窗口

输出

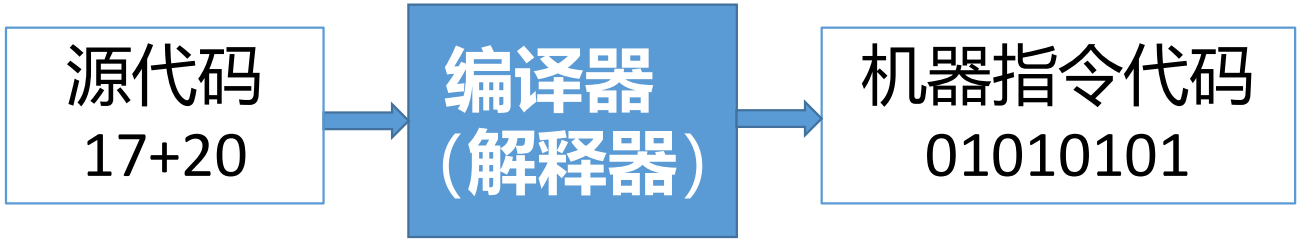
- C++标准输入输出流库包含了已经编写好的输入输出工具
- 输出流对象cout表示控制台窗口，输出运算符<<
- C++每个名字都属于某个名字空间，防止名字冲突！

```
#include <iostream>
int main()
{
    std::cout << “hello world!”;
}
```

名字限定：**名字空间名::**

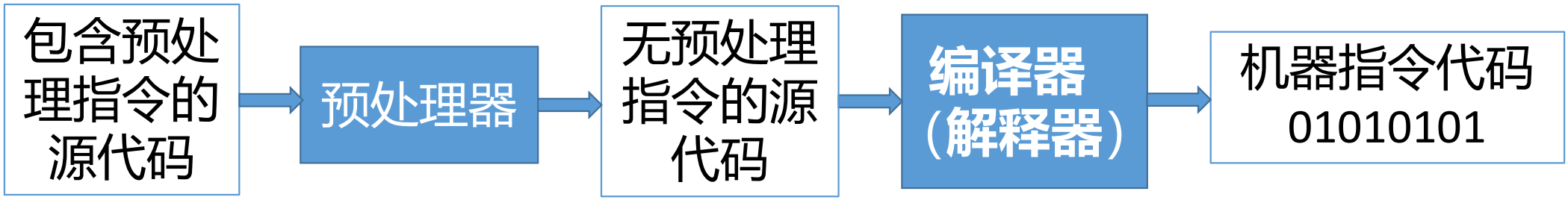
编译程序

- 编译器： 将源程序转变为机器指令代码。



编译程序

- 编译器： 将源程序转变为机器指令代码。
- 预处理器：对源代码中“预处理指令”进行处理，得到新的源代码。




```
#include <iostream>
```

编译器 (C++compiler)

- g++： Linux/Unix、 Mac、 minGW （windows移植版本）
- Clang c++:
- Microsoft Visual C++（ Visual Studio C++ Compiler ） ： windows

编译器 g++

- 编译源文件, 生成 二进制的目标文件(.obj)

  g++ -c file.cpp → file.obj

  g++ -std=c++17 -c file.cpp -o prog → file.obj; prog.out

- 链接目标文件, 生成可执行程序(.exe)

  g++ -o prog file1.obj file2.obj …

集成开发环境IDE

- integrated development environment (**IDE**)是包含了代码编辑、调试、编译运行等多种工具的软件。
- **Visual Studio**：widnows平台
- **CodeBlocks**：跨平台(Linux/Unix、 Mac、 Windows)
- **CLion**：跨平台(Linux/Unix、 Mac、 Windows)

Visual Sudio

Visual Studio

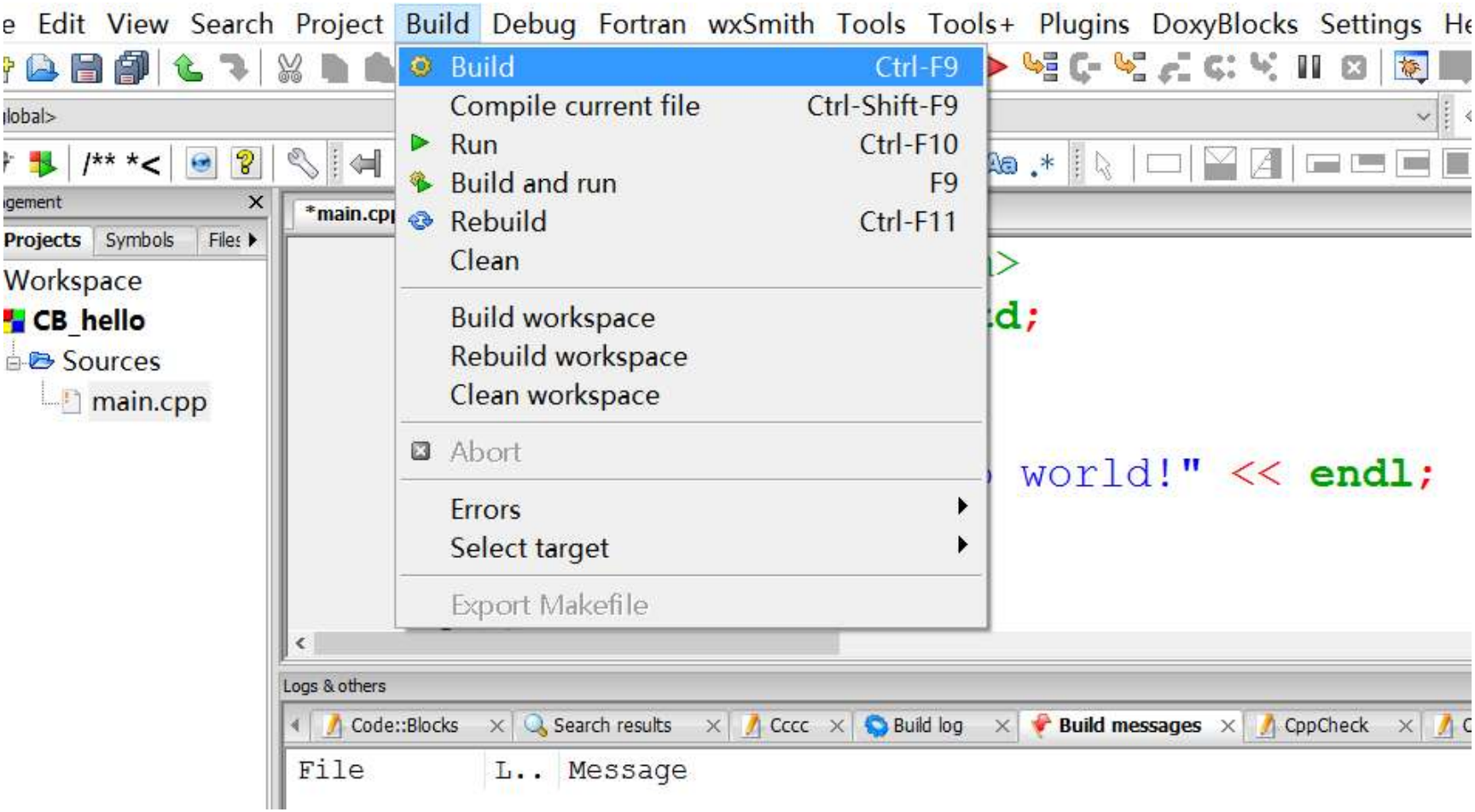
1)安装和使用:

https://hwdong-net.github.io ->tag ->” **C++17安装、配置**”

2)打开C++17开关：

右键弹出菜单->properities->configuration...->C/C++->Language->C++ language Standard->ISO C++17 standard

CodeBlocks


Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings
Build Ctrl-F9
Compile current file Ctrl-Shift-F9
Run Ctrl-F10
Build and run F9
Rebuild Ctrl-F11
Clean
Build workspace
Rebuild workspace
Clean workspace
Abort
Errors
Select target
Export Makefile
Projects Symbols
Workspace
CB_hello
Sources
main.cpp
*main.cpp
world!" << endl;
Logs & others
Code::Blocks Search results Cccc Build log Build messages CppCheck
File L.. Message

IDE集成开发环境

- CodeBlocks

Configuring your compiler: Choosing a language standard
https://www.learncpp.com/cpp-tutorial/configuring-your-compiler-choosing-a-language-standard/

Install Code::Blocks and GCC 9 on Windows - Build C, C++ and Fortran programs
https://solarianprogrammer.com/2019/11/16/install-codeblocks-gcc-windows-build-c-cpp-fortran-programs/

Online C++ Compiler

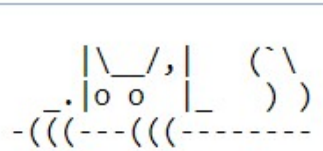
- 谷歌或bing搜索: “Online C++ Compiler”

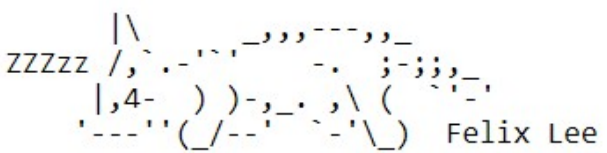
练习

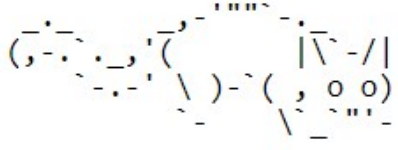
- 编写程序输出如下图案：

```
    1
  2   3
4   5   6
```

https://www.asciiart.eu/animals/cats







关注我

博客： hwdong-net.github.io

Youtube频道

hwdong

@hwdong · 5.01K subscribers · 558 videos

博客：https://hwdong-net.github.io >

youtube.com/c/4kRealSound and 4 more links