

C++语言基础

迂者 - 贺利坚

<http://blog.csdn.net/sxhelijian/>

<http://edu.csdn.net>



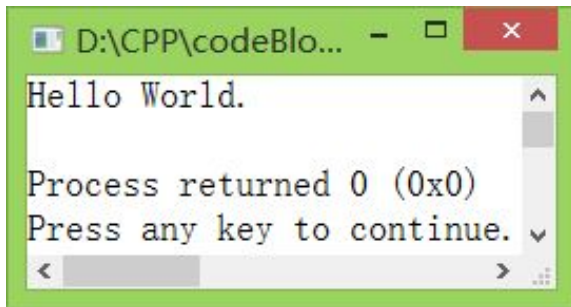


本节主题：

C++的数据类型和运算符

第一个C++程序

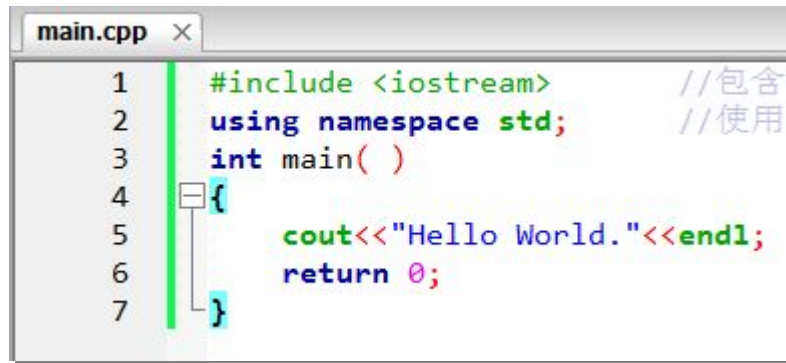
```
#include <iostream>    //包含头文件iostream
using namespace std;   //使用命名空间std
int main( )
{
    cout<<"Hello World."<<endl; //输出结果
    return 0;
}
```



A screenshot of a Windows console window. The title bar shows the path 'D:\CPP\codeBlo...'. The console output displays 'Hello World.' on the first line, followed by 'Process returned 0 (0x0)' and 'Press any key to continue.' on the next two lines. The cursor is positioned at the end of the third line.

头文件的小差别

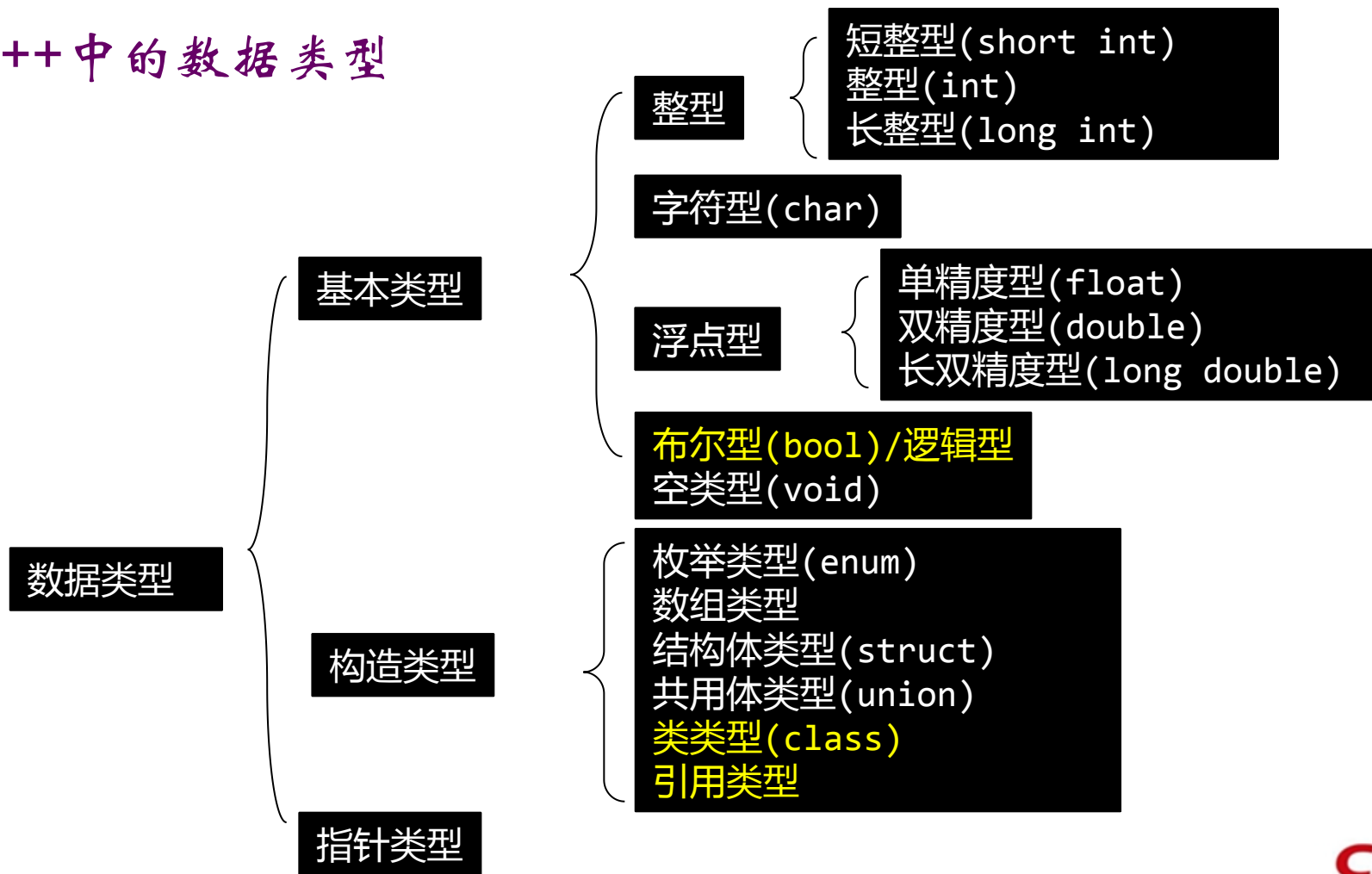
C	C++
stdio.h	iostream
math.h	cmath
string.h	cstring
stdlib.h	cstdlib
.....



A screenshot of a code editor window titled 'main.cpp'. The code is as follows:

```
1  #include <iostream>    //包含
2  using namespace std;   //使用
3  int main( )
4  {
5      cout<<"Hello World."<<endl;
6      return 0;
7  }
```

C++ 中的数据类型



布尔型(bool)/逻辑型数据及其运算

- ❏ C和C++中都用数值1代表“真”，用0代表“假”。
- ❏ C++增加了逻辑型数据类型：bool型
- ❏ 逻辑型常量只有两个，即false(假)和true(真)。

```
bool found, flag=false;
```

```
found=true;
```

- ❏ 逻辑型使程序直观易懂。
- ❏ 在编译系统处理逻辑型数据时，将false处理为0，将true处理为1。
- ❏ 在实际中，非0被视为“真”，0被视为“假”
- ❏ 逻辑型数据可以与数值型数据进行算术运算
- ❏ 逻辑运算的规则照旧

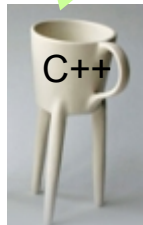


3<2的值？

假, 0



假, 0, false



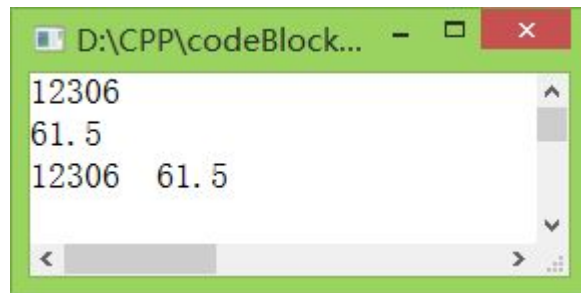
A	B	A&&B
T	T	T
T	F	F
F	T	F
F	F	F

A	B	A B
T	T	T
T	F	T
F	T	T
F	F	F

A	!A
T	F
F	T

结构体类型在定义变量时，其前struct可以省略

```
#include <iostream>
using namespace std;
struct student
{
    int no;
    float math;
};
int main( )
{
    int n;
    cin>>n;
    student wang; //C语言中，必须struct student wang
    wang.no=n;
    cin>>>wang.math;
    cout<<wang.no<<" "<<wang.math<<endl;
    return 0;
}
```



强制类型转换

支持两种格式：

（数据类型）（表达式）——C语言风格

数据类型（表达式）——C++新增

例

```
double x=2.5, y=4.7;
```

```
int a=25, b;
```

```
b=a+(int)(x+y);
```

```
y=double(5%3);
```



新增动态内存分配的运算符

分配内存用new运算符

格式

指针变量 = new 数据类型;

指针变量 = new 数据类型[数组大小];

如

```
int *p1, *p2, *p3;
```

```
p1=new int;
```

```
p2=new int[5];
```

```
p3=new int(5);
```

new为程序分配一块内存空间，并返回指向该内存的首地址。

释放空间用delete运算符

格式

delete 指针变量;

如

```
delete p1;
```

```
delete p3;
```

```
delete []p2;
```

用new获取的内存空间，必须用delete进行释放；

C语言中由malloc()函数分配内存，
由free()函数释放内存

新增作用域运算符 ::

```
#include <iostream>

using namespace std;

float a=2.4;    // 全局变量a

int main()
{
    int a=8;    // 局部变量a
    cout<<a<<endl; // a为局部变量
    cout<<::a<<endl; // ::a表示全局变量a
}
```

```
#include <iostream>

//这儿不写使用的命名空间

float a=2.4;

int main()
{
    int a=8;
    std::cout<<a<<std::endl;
    std::cout<<::a<<std::endl;
}

std::, 表明使用命名空间std中定义的标识符
```

变量的引用(reference)

❏ C++对C的一个重要扩充：对一个数据可以使用“引用”

❏ 引用的作用：为一个变量起一个别名，例

```
int a;    //定义a是整型变量
```

```
int &b=a; //声明b是a的引用，b是a的别名
```

```
a=20;
```

```
cout<<b<<endl;
```

❏ 声明变量b为引用类型，不另开辟内存单元来存放b的值。

❏ b和a占内存中的同一个存储单元，它们具有同一地址。

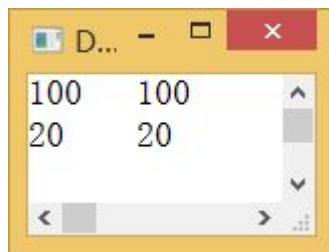
❏ a或b的作用相同，都代表同一变量。

❏ 引用可以理解为：使变量b具有变量a的地址。

❏ &是引用声明符，并不代表地址。不同于“把a的值赋给b的地址”。

引用的简单使用

```
#include <iostream>
#include <iomanip>
using namespace std;
int main( )
{
    int a=10;
    int &b=a; //声明引用类型变量要同时初始化
    a=a*a;
    cout<<a<<" "<<b<<endl;
    b=b/5;
    cout<<b<<" "<<a<<endl;
    return 0;
}
```



增加引用类型，主要用于扩充函数传递数据功能

```
#include <iostream>
using namespace std;
void swap(int &a,int &b);
int main( )
{
    int i,j;
    i=3,j=5;
    swap(i,j);
    cout<<"i="<<i<<" "<<"j="<<j<<endl;
    return 0;
}

void swap(int &a,int &b)
{
    int temp;
    temp=a;
    a=b;
    b=temp;
}
```

常量(constant variable)

☞ 常量：在定义变量时，加上关键字 `const`，则变量的值在程序运行期间不能改变

☞ 例如

```
const int a=3;
```

```
const int b=3+6, c=3*cos(1.5);
```

☞ 在定义常量时必须同时对它初始化，此后它的值不能再改变。

```
#include<iostream>
using namespace std;
const int price = 30;
int main ( )
{
    int num, total;
    num=10;
    total=num * price;
    cout<<"total="<<total<<endl;
    return 0;
}
```

符号常量(宏定义) VS. 常变量

```
#include<iostream>
```

```
using namespace std;
```

```
const int price = 30;
```

```
int main ( )
```

```
{
```

```
    int num, total;
```

```
    num=10;
```

```
    total=num * price;
```

```
    cout<<"total="<<total<<endl;
```

```
    return 0;
```

```
}
```

C++的程序员更
爱常变量

```
#include <iostream>
```

```
using namespace std;
```

```
#define PRICE 30 //不是语句，末尾不加分号
```

```
int main ( )
```

```
{
```

```
    int num, total;
```

```
    num=10;
```

```
    total=num * PRICE;
```

```
    cout<<"total="<<total<<endl;
```

```
    return 0;
```

```
}
```

THANKS

本课程由 迂者-贺利坚 提供

CSDN网站：www.csdn.net
企业服务：<http://ems.csdn.net/>
人才服务：<http://job.csdn.net/>
CTO俱乐部：<http://cto.csdn.net/>
高校俱乐部：<http://student.csdn.net/>
程序员杂志：<http://programmer.csdn.net/>

CODE平台：<https://code.csdn.net/>
项目外包：<http://www.csto.com/>
CSDN博客：<http://blog.csdn.net/>
CSDN论坛：<http://bbs.csdn.net/>
CSDN下载：<http://download.csdn.net/>

