

# 一、基本数据类型

## 【大纲内容】

【难度 1】整数型: int、long long

【难度 1】实数型: float、double

【难度 1】字符型: char

【难度 1】布尔型: bool

## 什么是数据类型 (Data Type) ?

在程序里，我们需要告诉电脑：“这个信息是数字？是文字？还是真假？”

——就像生活中要分清楚苹果、水、硬币一样。

C++ 提供了几种最基本的数据类型，就像准备不同的储物盒子，来装不同的“数据东西”。

### (一) 整数型 (Integer Type)

#### 1、int (整数)

int 表示“**整型数据**”，用来存储没有小数点的数，比如：1、-10、2025。

就像你在数学课里学的“整数”，没有小数点，不能是 3.14，只能是 3、5、100 这种。

#### ☒ C++ 示例：

```
int apples = 5;  
  
cout << "我有" << apples << "个苹果。" << endl;
```

#### 2、long long (长整数)

long long 表示**更大的整数**，可以存储特别特别大的数，比如宇宙里的星星数！

就像一个超级大的书包，能装很多很多东西。

#### ☒ C++ 示例：

```
long long population = 1400000000; // 中国人口
```

```
cout << "中国大约有" << population << "人。" << endl;
```

## (二) 实数型 (Floating Point Type)

### 1、float (单精度小数)

float 表示带小数点的数，但是精度（准确度）一般，适合普通场景。

就像量体温的时候写成“36.5 度”，这种有点点的数字就是“实数”。

#### C++ 示例：

```
float temperature = 36.5;  
  
cout << "今天体温是：" << temperature << "度。" << endl;
```

### 2、double (双精度小数)

double 表示更精确的小数，可以保存更多小数位。

就像你在数学考试里写了“3.1415926”，特别长的小数，double 能存得更准。

#### C++ 示例：

```
double pi = 3.1415926535;  
  
cout << "圆周率是：" << pi << endl;
```

## (三) 字符型 (Character Type)

### 1、char (字符)

char 表示一个字符 (Character)，如字母、数字或符号，用英文单引号 ‘ ’ 括起来。

就像你写字母‘B’、标点‘!’、数字‘5’，但只能一个字哦！

#### C++ 示例：

```
char grade = 'A';  
  
cout << "你的成绩是：" << grade << endl;
```

注意：

- 'A' 是字符, "A" 是字符串;
- '7' 是字符 7, 不是数字 7。

#### (四) 布尔型 (Boolean Type)

##### bool (布尔类型)

bool 是“布尔型”，只能有两个值: **true (真)**、**false (假)**，用来判断对错。

就像你问“今天下雨了吗？”这个问题只有两种答案：是 (true) 或不是 (false)。

##### C++ 示例：

```
bool isRaining = false;

if (isRaining) {
    cout << "别忘记带伞！" << endl;
} else {
    cout << "天气不错，出门吧！" << endl;
}
```

#### 总结表格

类型 (中英文)	举例	用途描述	示例代码
int (整数型)	5, -3	没有小数的数字	int age = 10;
long long (长整数)	1234567890	很大的整数	long long pop = 1400000000;
float (小数型)	3.14	带小数，精度普通	float temp = 36.5;
double (高精度小数)	3.14159	更精确的小数	double pi = 3.14159;
char (字符)	'A', 'b'	单个字符，必须用单引号	char grade = 'A';

bool (布尔)

true/false

表示“是或不是”

bool isSunny = true;

**C++ 常用数据类型范围表**

数据类型	关键字	大小(字节)	范围	说明
字节型	char	1	-128 ~ 127	或 0 ~ 255 (无符号)
短整型	short	2	-32,768 ~ 32,767	
整型	int	4	-2,147,483,648 ~ 2,147,483,647	最常用
长整型	long	4 或 8	-2,147,483,648 ~ 2,147,483,647 (32 位) 或更大 (64 位)	依赖平台
长长整型	long long	8	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807	大数计算
单精度浮点数	float	4		精度约 6-9 位小数
双精度浮点数	double	8		最常用，精度 15-17 位
扩展双精度	long double	8 或 16		高精度计算
字符	char	1	-128 ~ 127 或 0 ~ 255	存储 ASCII 字符
布尔值	bool	1	true (1) 或 false (0)	逻辑判断