

08 Variables in C++

1. 变量类型

C++中, 不同变量类型之间唯一的区别就是 **size(大小)**, 即这个变量占用了多少内存

数据类型的实际大小取决于编译器, 可能会由于编译器不同而大小不一样

1.1 int

int 类型是为了在一定范围内存储整数, 因为它大小是4个字节

$1\text{byte} = 8\text{bits}, 4\text{bytes} = 32\text{bits}$

变量是 **有符号数(signed)**, 即有一位需要来表示符号 (\pm), 因此只有31位留给实际数字表示。

1 bit可为“0”或“1”, $2^{31} = 2,147,483,648$ (21亿多), 由于还需要表示0, 所以能表示的最大数字是 $2^{31} - 1$

\therefore **int** 取值范围 $(-2147483648, 2147483647)$

只想要正数怎么办?

→ 无符号数 (unsigned), 可表示 2^{32}

C++

```
unsigned int
```

1.2 常见的类型

char	short	int	long	long long	(long int)
1 byte	2 bytes	4 bytes	4 bytes	8 bytes	4 bytes

任何一个类型前都可以加 **unsigned**

1.3 char

对某些数据类型做假设

C++

```
char a = 'A'; // A
char a = 65;  // A
```

因为将一个 **char** 传入 **std::cout**, 会把它当作一个字符而不是数字

C++

```
short a = 65; // 65
short a = 'A'; // 65
```

1.4 存小数

float & double

`float` 占用4个字节，通过在数字后面加"F/f"与 `double` 进行区分
`double` 占用8个字节

```
float variable = 8.5;
```

(double)(8.5)
Search Online

```
double var = 8.2f;
```

(float)(8.2F)

1.5 bool

0 for false
anything except 0 for true

占用1个字节

由于内存寻址无法直接访问bit，只能寻址到bytes，所以想少占内存可以巧妙地把8个bools存在一个byte里

2. sizeof操作符

```
std::cout << sizeof(bool) << std::endl; // 1
```

C++

3. pointers & references

指针：类型* `bool*`

引用：类型& `bool&`