**变量**能够方便我们访问、存储和管理数据。

抽象地举一个例子，我们向一排储物柜里放物品，在找东西的时候，我们需要记住东西放在了第几个储物柜里，不能每次都把所有的柜子翻个底朝天，那样效率太低了。那么同样，对于计算机来说，我们的数据指令等等都会存放到寄存器中，那么如果我要访问一个我存储在计算机中的数据，我就需要知道这个数据存在的地址——这个就很惨，因为寄存器的地址大多是十六进制的数，比如0x0000 0000 0100 1101，这对于人类来说，记忆起来太麻烦了，如果我们有999个数字......记999个地址编号？那真是惨绝人寰啊！那么如果我们能够用一个简单的、符合人类思维的字母、单词或者字符来代替这个地址数的话，岂不是美滋滋？这就是变量的意义。

怎么来使用这个“变量”呢——变量创建的语法就是：

数据类型 变量名称(=初始值)

int a=0；

这里的数据类型我们后续课程会详细讲解，这个int就是整型，也就是说a是个整数，那么这个语句就是将0赋值给整型变量a。

简单说一下这里的“等号”不是“等号”，什么意思，在C++语言中，“==”才是等号，“=”的意思是赋值号，什么意思呢，比如上述语句相当于我们把0赋值给了a。那有意思的就来了，如果不给a初始化，编译器会不会报错呢？

我们发现，虽然出现了警示框，但编译通过了——虽然没有报错，但输出的是垃圾数据。所以我们使用的变量，一定一定不要忘记初始化。

下面我们来说**常量**。常量，顾名思义，是程序中无论如何都不会改变的数据。比如说每分钟都是60秒、圆周率的值是3.1415926...，等等。这些常量的值一旦发生改变，它就失去了原本的含义，这个时候我们就要定义常量。

C++定义常量有两种方式：

（1）const 数据类型 变量名 = 常量值

const int min\_per\_hour=60;

修改就会报错

Int a=min\_per\_hour+1;

cout<<...

（2）宏定义：#define 变量名 常量值

#define min\_per\_hour 60（没有分号！先打上，让大家看到编译器报错）

不建议使用，define和const的实现机制不同，它是将所有的min\_per\_hour这个标示符用60代换了，而不是像const定义的min\_per\_hour的值是60，宏定义容易出现一些莫名奇妙的问题，因此大多数情况建议使用const常量。

学会了变量和常量，再也不用记那些奇奇怪怪惨无人道的寄存器地址了——这是多么伟大的计算机设计思想，你在编程的时候不需要知道数据和指令存放的地址是什么，就像你的储物柜都活了过来，之后你给它们挨个起了名，你拿着一篮子苹果，只需要叫一声——坨坨！苹果就放到了叫“坨坨”的这个储物柜里；再想吃的时候再喊一声“坨坨！”，苹果就取出来了。至于“坨坨”究竟放在哪儿，是第几个储物柜，我们不需要关心，我们只需要记住“坨坨”这个名字。

学习C++，往往就是这么朴实无华，且枯燥。