

## 基础篇 第二回

## 用变量实现灵活机动

## 记规则以免非法命名

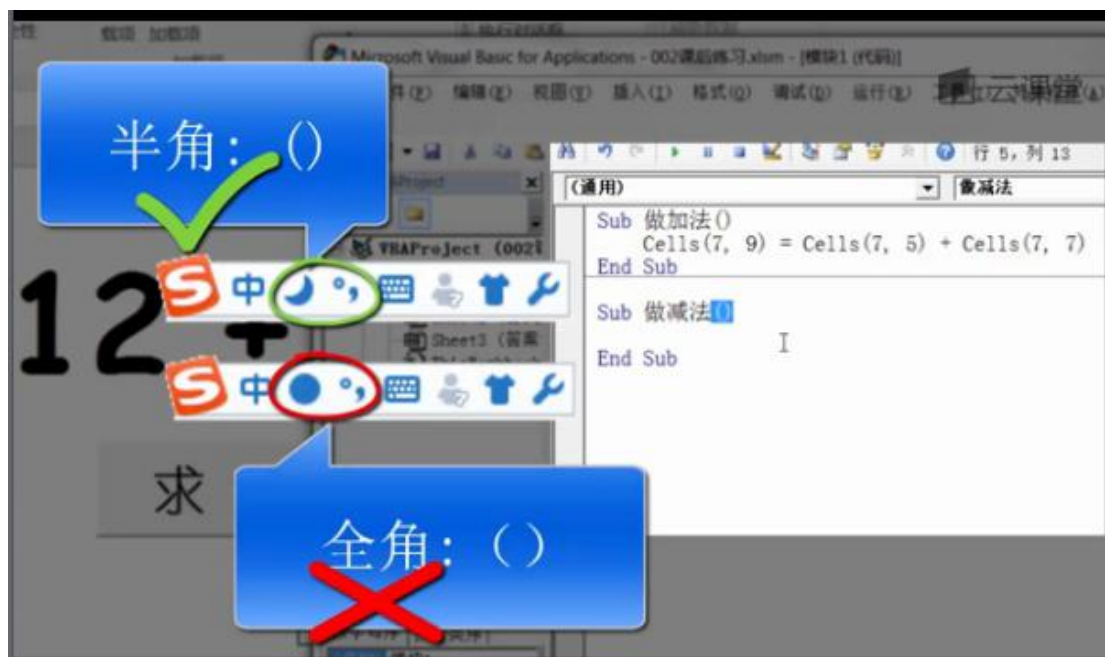
<http://study.163.com/course/courseLearn.htm?courseId=1003088001#/learn/video?lessonId=1003510006&courseId=1003>

**088001**

上节课我们介绍了如何用 `Cells` 属性操作单元格，并且用它写下了一段小程序，能够执行一道简单的算数试题，不知道大家回去做一个练习没有，也就是增加一个求差的操作，首先我们看一下参考答案，

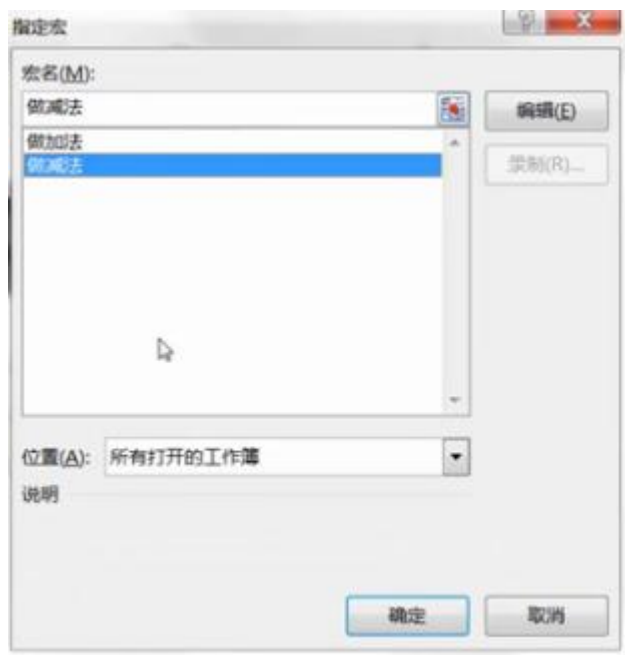


这就是我做的一个参考答案，正好利用我做的这个参考答案，我给大家介绍一种新的也是更通用的宏的创建方法，首先我们还是从“开发工具”里面插入一个新的按钮。不过这一次在弹出的“指定宏”的对话框中，我们不去点击“新建”一个宏，而是点击“取消”，这样我们的这个“求差”的按钮，这样我们的按钮就没有关联到任何的宏上。点击也没有任何的反应。那么到哪里去新建这个宏呢？我们点击 **Visual Basic**。在弹出的 VBA 编辑器里面双击“模块 1”，然后在“做加法”这个宏下面人工的添加，输入：`sub` 做减法，VBE 自动生成了后面的 `End Sub` 和后面的括号，而且还添加了一个横线，在我们的两个宏之间，把他们分割开，让他们看起来更清楚，这里特别要提醒大家，我们在编程序所有的圆括号都是英文的半角的圆括号，千万不要使用中文的全角的括号。

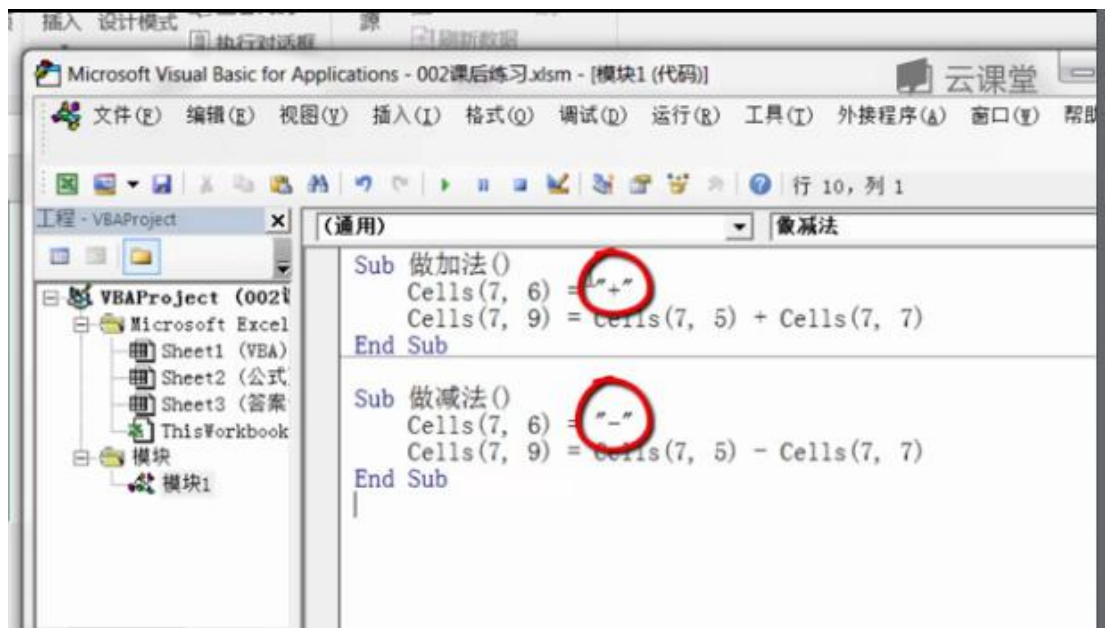


否则一定会出现错误的。那么做减法的宏如何写呢？

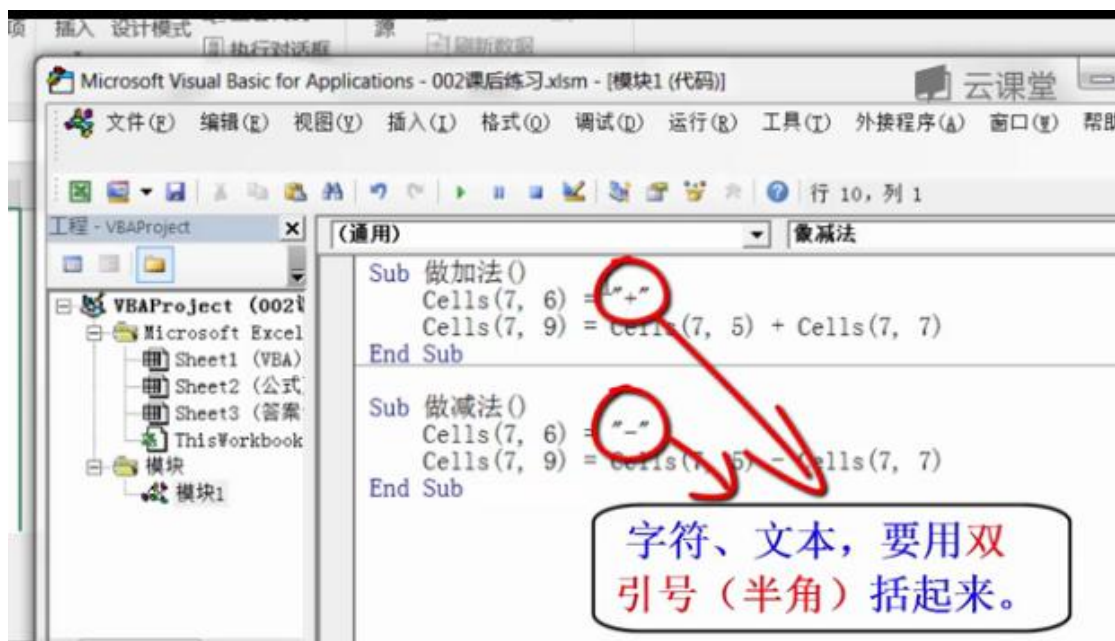
和做加法的过程几乎是一样的，只不过是把加号改成了减号。复制代码然后保存运行，但是按钮没有反应，为什么呢？因为我们还没有把这个按钮关联到我们的宏上，怎么关联？在新按钮上点击右键，“指定宏”。在出现的对话框里，选择“做减法”这个宏，



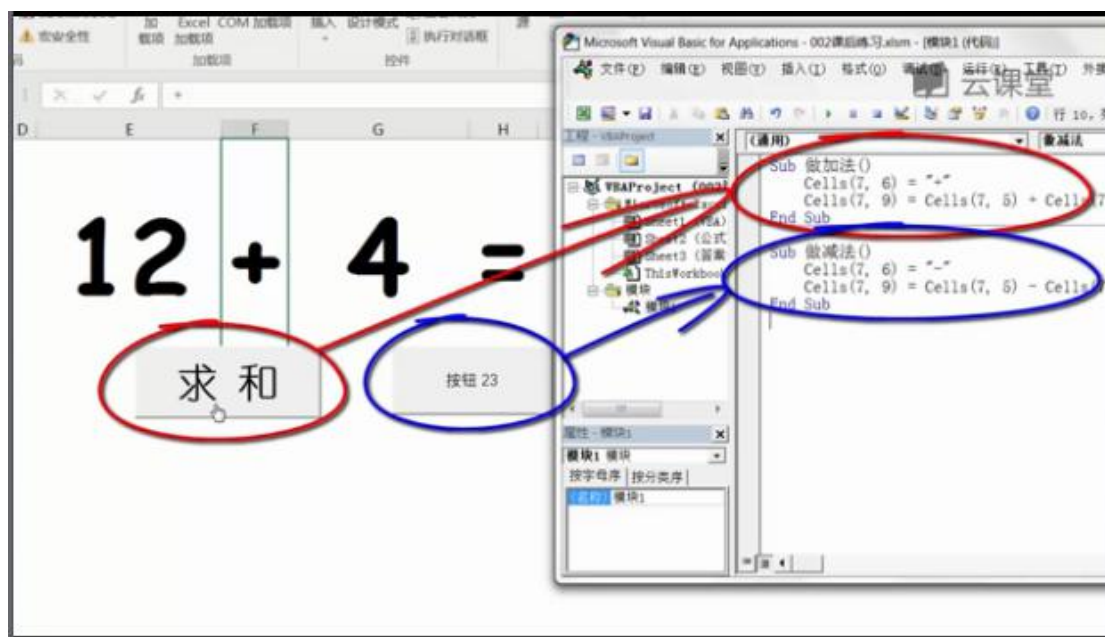
现在再摁下这个按钮，起作用了，不过单元格的加号还没有改成减号。那么显然不科学。在做加法的时候，我们让操作符所在的单元格等于一个加号。而做减法的时候，该单元格为减号，大家注意到加号和减号两边都有双引号，



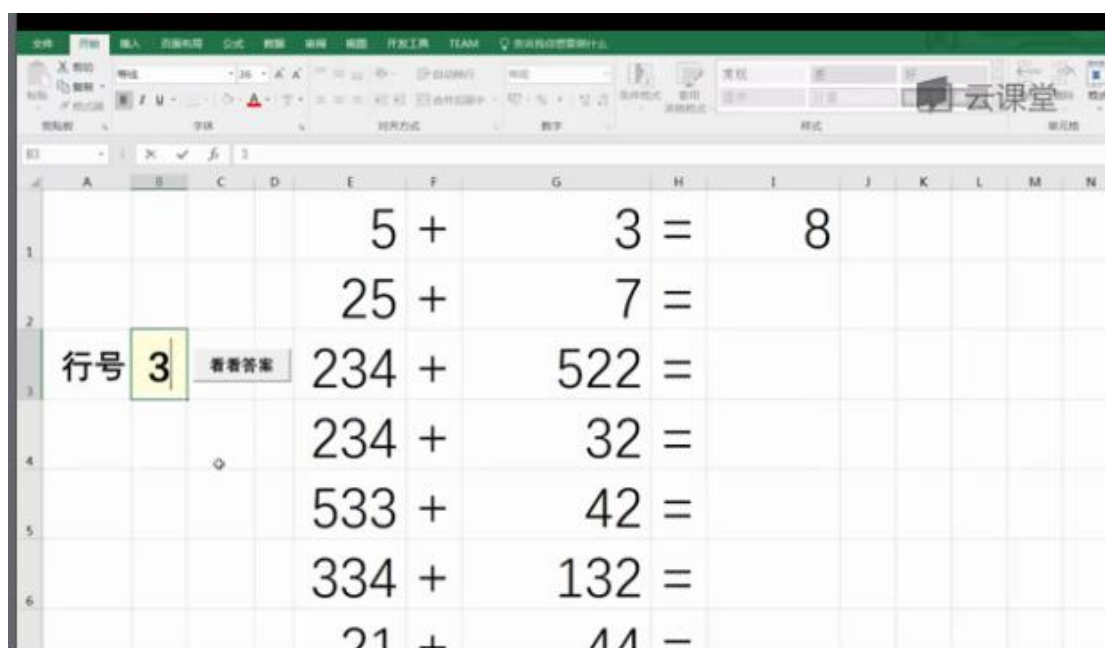
这是因为加号和减号既不是变量也不是数字。而是一个字符。



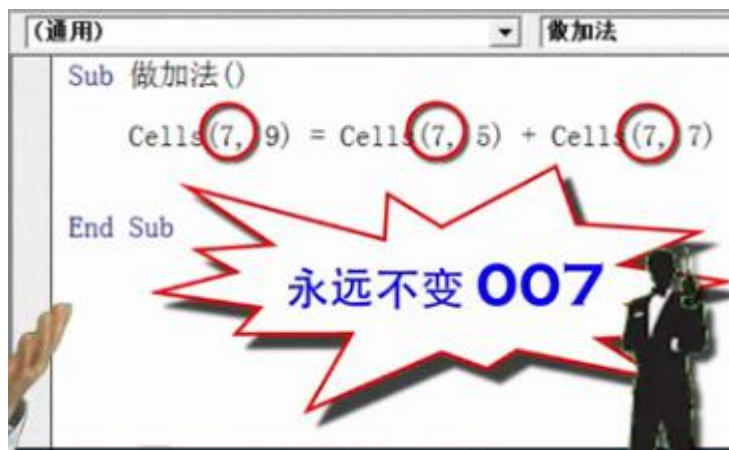
在 VBA 中所有的字符和字符串都用双引号括起来，关于字符串的内容，我们后面再讲。这样下来我们的程序中有了两个宏，也有了两个按钮，不同的按钮执行不同的宏。



我们的任务就完成了，这节课我们就在这节课的基础上，进一步的扩展，让它更加符合实际。什么任务呢？我们知道老师上课的时候，绝不会只出一道题的，那就享受不到为难学生的乐趣了，是吧？我们会准备很多道题，根据同学的接受程度，选择显示其中某些答案。比如像这样我们看一下第三题的答案：



于是输入 3，按下按钮，显示第三题答案。我们看下第五题的答案，于是输入 5，然后显示第五题的答案。那么 we 想一想，这样的任务上我们用上次的代码能不能实现呢？好像有点麻烦！原因吗很简单，



我们把行号写成了固定的数字 7，也就是把行号写死了，所以不管什么时候，按下按钮只计算第七行，那么我们想计算其他行的题怎么办呢？恐怕只能在每一行后面增加一个按钮，每个按钮，每个按钮关联到一个宏上，每一个宏几乎相同，只是行号不同。



如果我们要这样做的话，我们不要学 VBA 了，只要人工去解决好了。那么问题出在哪里？我想大家都明白，我们应该把行号这样一个变化的值写成了永远不变化的固定的数字。那么怎样解决这个问题呢？这就涉及计算机变成里面一个最基本的概念就是**变量**。什么叫做变量呢？顾名思义就是它的值可以根据需要随时变化，说道变量我们会下想到初中的代数。



变量!

$$(a+b)^2 = a^2 + 2ab + b^2$$

在这样的一个式子里， $a$  和  $b$  就是变量。因为他们只是代号，他们不是具体的数字，我们在计算的时候可以根据需要随时带入一个真实的值  $a=1, b=2$ 。他们会输出不同的响应的结果。计算机程序也是一样，比如在下面这个里面：

云课堂

Cells ( 7 , 9 ) = Cells ( 7 , 5 ) + Cells ( 7 , 7 )

	A	B	C	D	E	F	G	H	I	J
1					5 +		3 =			
2					25 +		7 =			
3	行号		看看答案		234 +		522 =			
4					234 +		32 =			
5					533 +		42 =			
6					334 +		132 =			
7					21 +		44 =			
8					234 +		32 =			
9					234 +		32 =			

我们的行不希望它是一个具体的数字 7。而是希望他是能够随着第三单元格的数值不断变化的这样的一个量。于是我们干脆就把它写成一个变量，把它用代号来表示。

超级 Cells (变, 9) = Cells (变, 5) + Cells (变, 7)

云课堂

行号 8 看看答案

	A	B	C	D	E	F	G	H	I	J
1					5 +	3 =				
2					25 +	7 =				
3					234 +	522 =				
4					234 +	32 =				
5					533 +	42 =				
6					334 +	132 =				
7					21 +	44 =				
8					234 +	32 =				
9					234 +	32 =				

什么代号呢？那就用 x 吧，

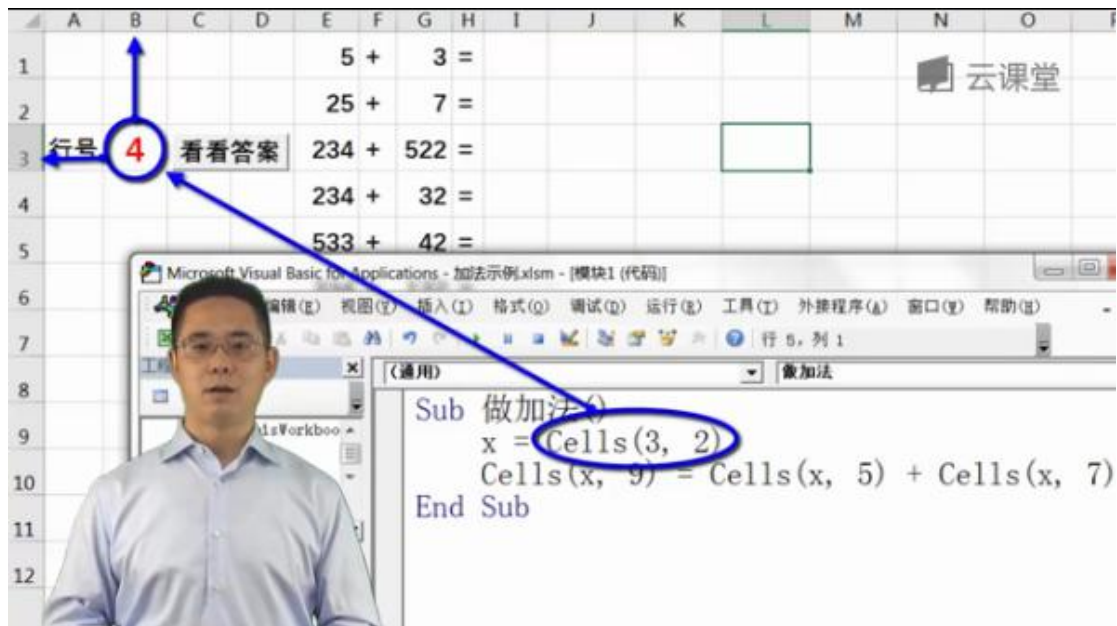
Microsoft Visual Basic for Applications - 加法示例.xlsm - [模块1 (代码)]

编辑(E) 视图(V) 插入(I) 格式(O) 调试(D) 运行(R) 工具(T) 外接程序(A) 窗口(W) 帮助(H)

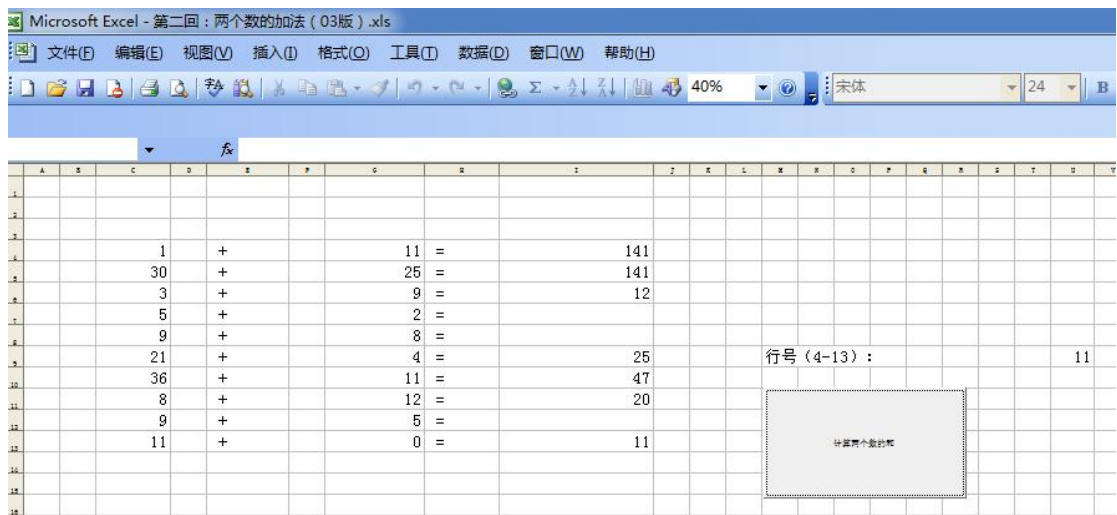
工程 通用 做加法

```
Sub 做加法()
    Cells(x, 9) = Cells(x, 5) + Cells(x, 7)
End Sub
```

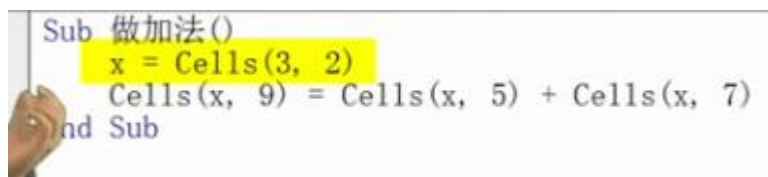
那么 x 到底是多少？我们需要在第三单元格的数值发生变化，于是我们在运算之前再加一句。



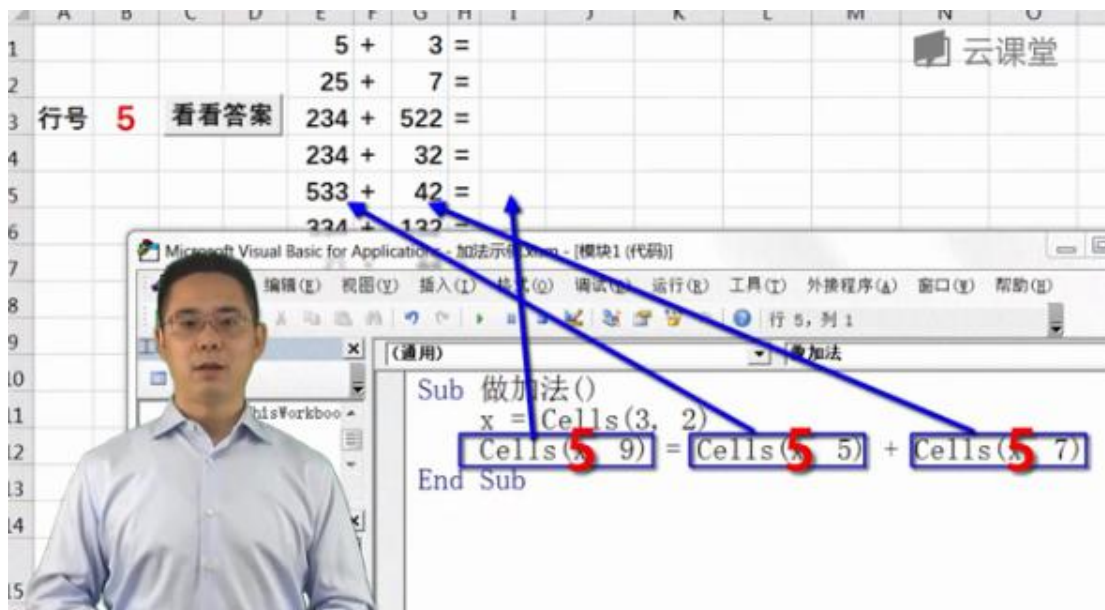
好，我们在看一下我们程序的运行效果，达到了运行的效果了（自己的测试结果）。



那么这程序的具体的执行流程是怎么样的呢？当我们按下按钮以后，VBA 替我们自动给我们找到了做加法这个宏，然后执行第一行语句：



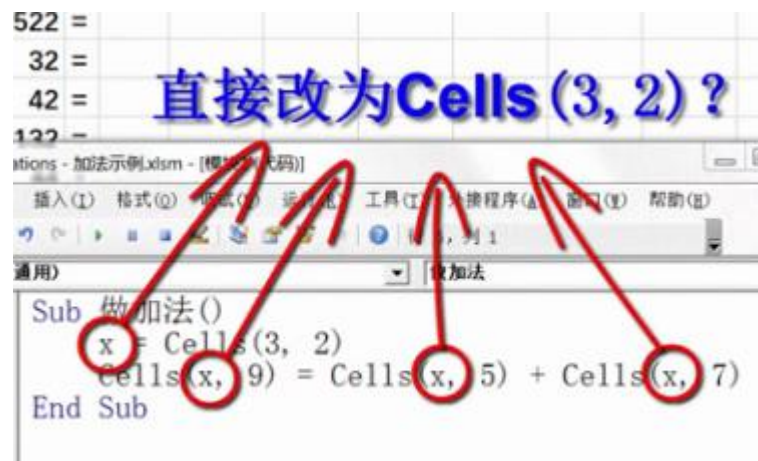




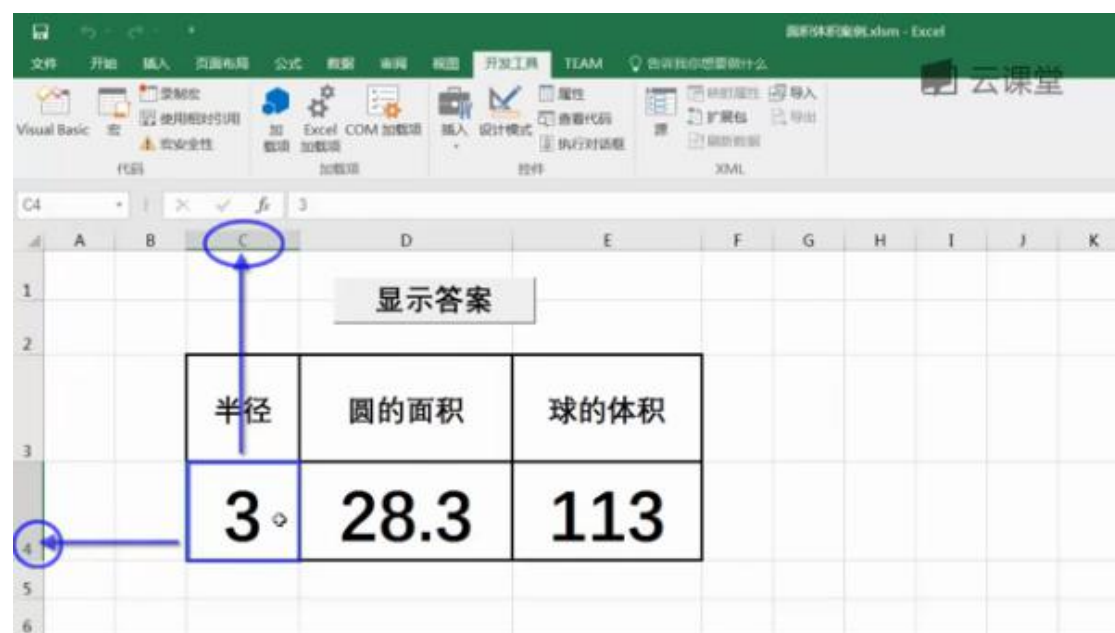
如果我们将第三单元格的值改成了8，x的值就变成了8。  
 就这么简单用变量使我们的程序变得灵活起来。



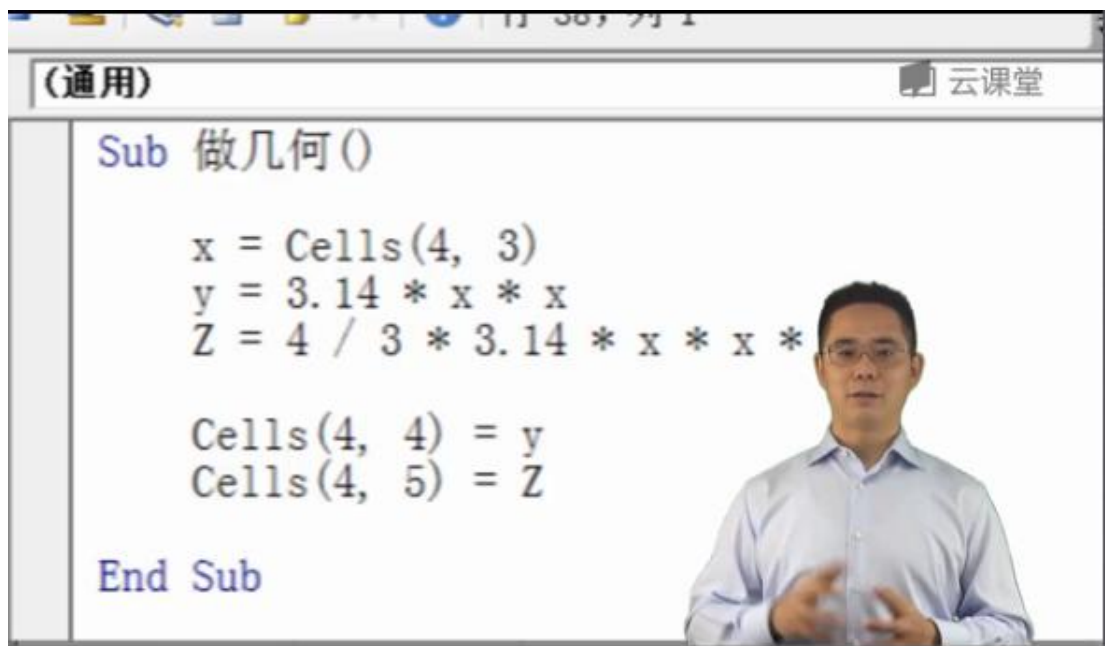
用同学会问我不用变量可以吗？我为什么不把 Cells(3,2)写道我们的程序里。比如：



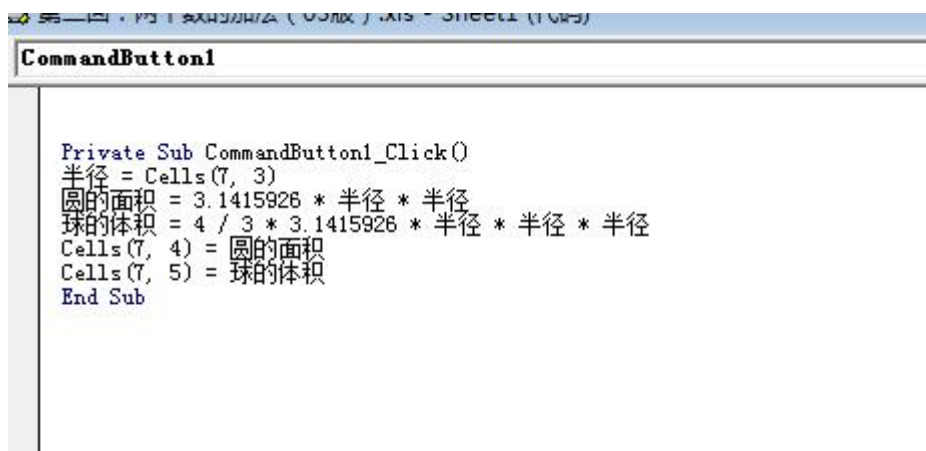
非常聪明，完全可以实现一样的效果，并且不使用变量，只不过这个程序第一：这个程序写起来非常的长，而且容易出错，第二，VBA 每次去读取一个单元格都要消耗很多的时间。要远远的超过在内存中读取一个变量的时间。所以这种操作非常多，而且反复执行的话，我们的程序的效率会变得非常的差，所以使用变量的好处就非常明显了，第一它可以使我们的程序变得灵活，第二他可以让我们的代码变得简洁，不会那么啰嗦出错，比如我们再举一个例子。假如我们给的任务是给出单元格中的半径的数值，给出圆的面积和球的体积。



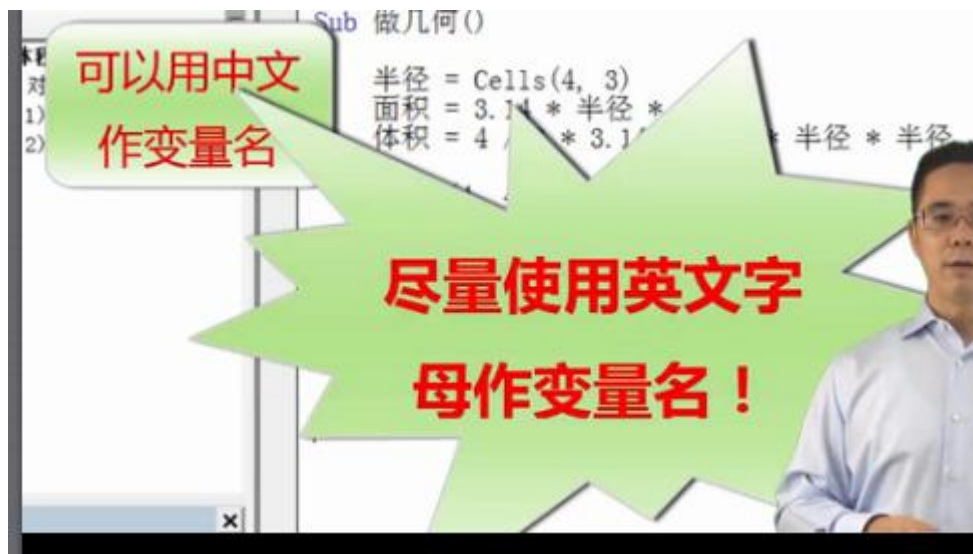
分别显示在 D4 和 E4 的两个单元格里，那么该怎么写代码呢？我们看一下：



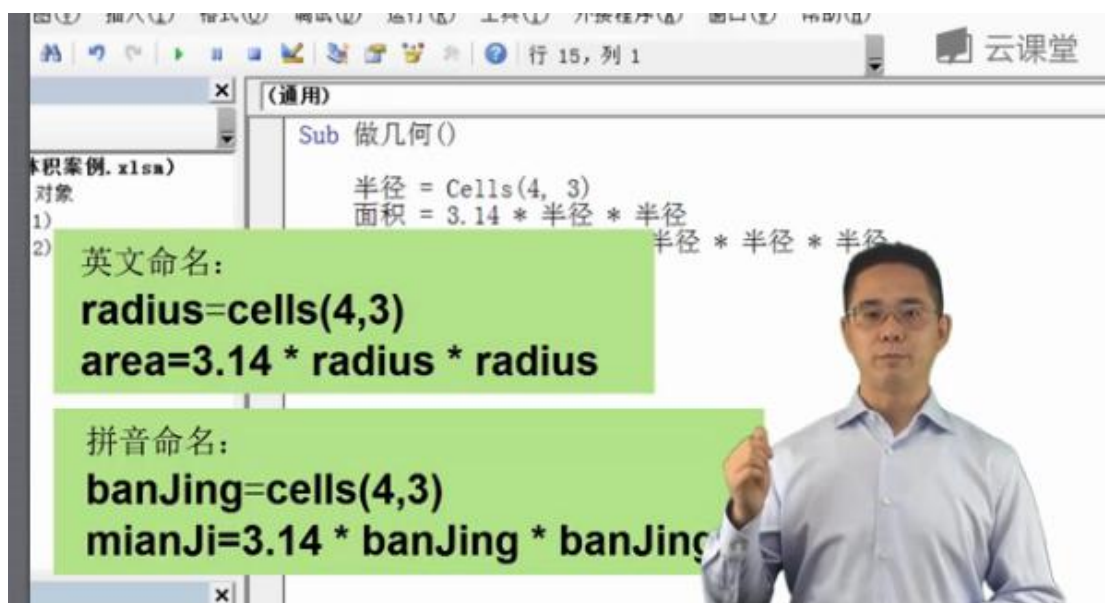
大家看一下这个代码是不是很清晰呢？到这里我想大家已经理解了变量的好处。不过还有一个问题，大家绝不觉得这个代码还是有一点混乱呢？ $x, y, z$  分别代表什么呢？如果不读代码我们恐怕很难理解它。而且  $x, y, z$  是不是让大家想起了高考复习数学的那段经历？要想编写好的代码，必须给你的变量起好的名字，比如现在的程序，我的每个变量不再叫  $x, y, z$ ，而是用有意义的名字来代替它。是不是就好的多了呢？



给变量起名字和给人起名字是很有讲究的。首先：（1）尽量使用有意义的名字，单纯的使用  $a, b, c$  我们很难理解变量的意思。我们写下半径这种英文单词就很容易理解了，这里可以看到我们 VBA 既支持我们用英文又支持我们用中文做变量名，不过杨老师绝对不建议大家用中文来命名变量，



因为用中文不仅拼写比较麻烦，而且在不同语言版本的 windows 里很有可能它就不被支持。特别的如果大家学习其他的编程语言，那么在绝大多数的编程语言里，都不支持使用中文做变量，所以大家还是保持用英文或者用拼音做变量命名这样的习惯。



再一点【1】需要说明的是并不是所有的字符都可以变量，一般做变量的都是英文字母、中文、数字或者下划线（不要使用特殊的字符），

【2】同时还有一个规定，变量名的第一位不能以数字开头。

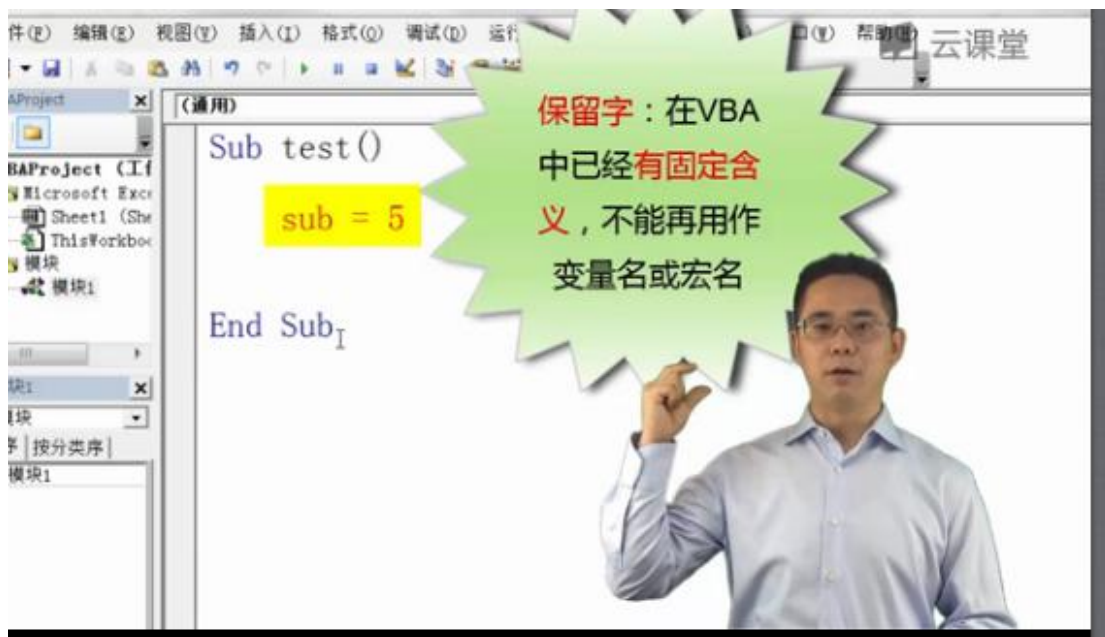
【3】变量名不能与系统的保留字重复。

这是什么意思呢？比如有个变量我们起名为 sub,如果写 sub=5,VBA 会怎么想呢？



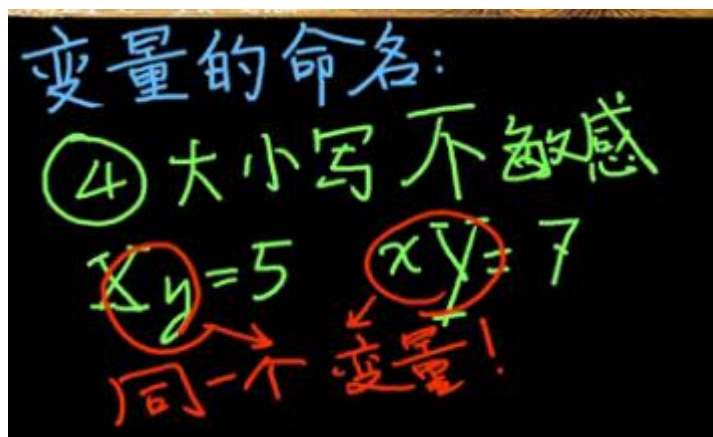


当 VBA 读到 Sub 的时候，他会以为这又是一个宏的开始，而没有想到是你自己定义的一个变量， 换句话说我们使用了一个在 VBA 里面已经固定化的单词，



作为我们的变量，让 VBA 产生了混淆。像 sub 这种就成为系统里面的保留字，还有很多比如说 sub/ end /if/for/while 等等，所以大家不要使用在 VBA 里面已经使用的保留字，最后需要注意的是 VBA 是对大小写不敏感的（也就是说大小含义相同），下面的两个变量是同一个变量。





其实关于变量我们还有很多细节需要讨论，我们下节课在深入的研究一下.