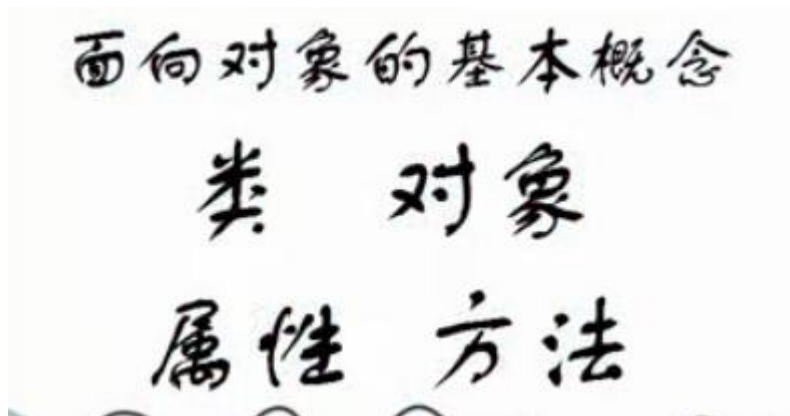


基础篇 第十三回

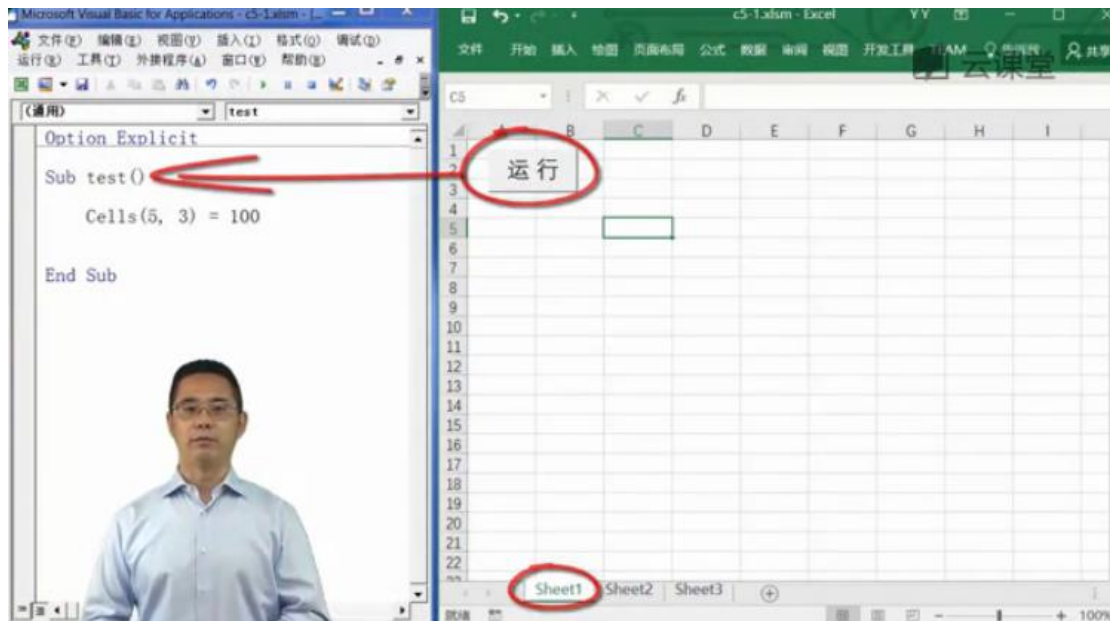
面向对象初显威力，Worksheets 玩转表单

<http://study.163.com/course/courseLearn.htm?courseId=1003088001#/learn/video?lessonId=1003510012&courseId=1003088001>

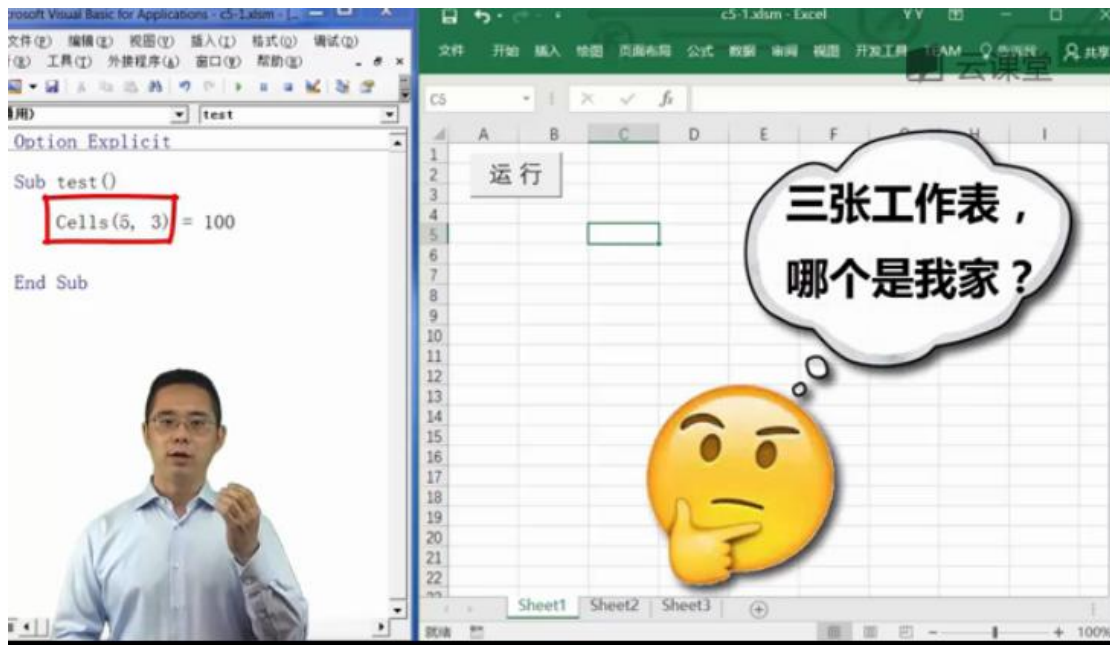
上一届课里我们讲了一些云山雾绕的方法，什么类，对象，方法啊属性啊，等等



概念，相信大家现在还没想明白，没关系，我们现在想不明白也不要紧，因为计算机是一个非常实用的技术，所以它里面看上去很多难懂难以理解的概念，归根到底都是和实际的问题挂到一起的，所以后面在用这些技术解决实际问题的時候，回过头来再来看这些概念，会感觉这些概念很好理解，好的现在回到正题，看一下现在再 excel 中，怎样用面向对象技术来解决实际的问题，我们从工作表开始，大家看这是非常简单的小程序。



在 sheet1 这张表里有个按钮，按下按钮之后就会执行左边的一段小程序，程序的内容很简单，也就是把第五行第三列中的单元格的内容写成 100，这个程序我们都能理解，那么我要问一个问题，我们的这个工作簿里面有三个按钮，请问这个 Cells(5,3)指的是那张表里面的呢？



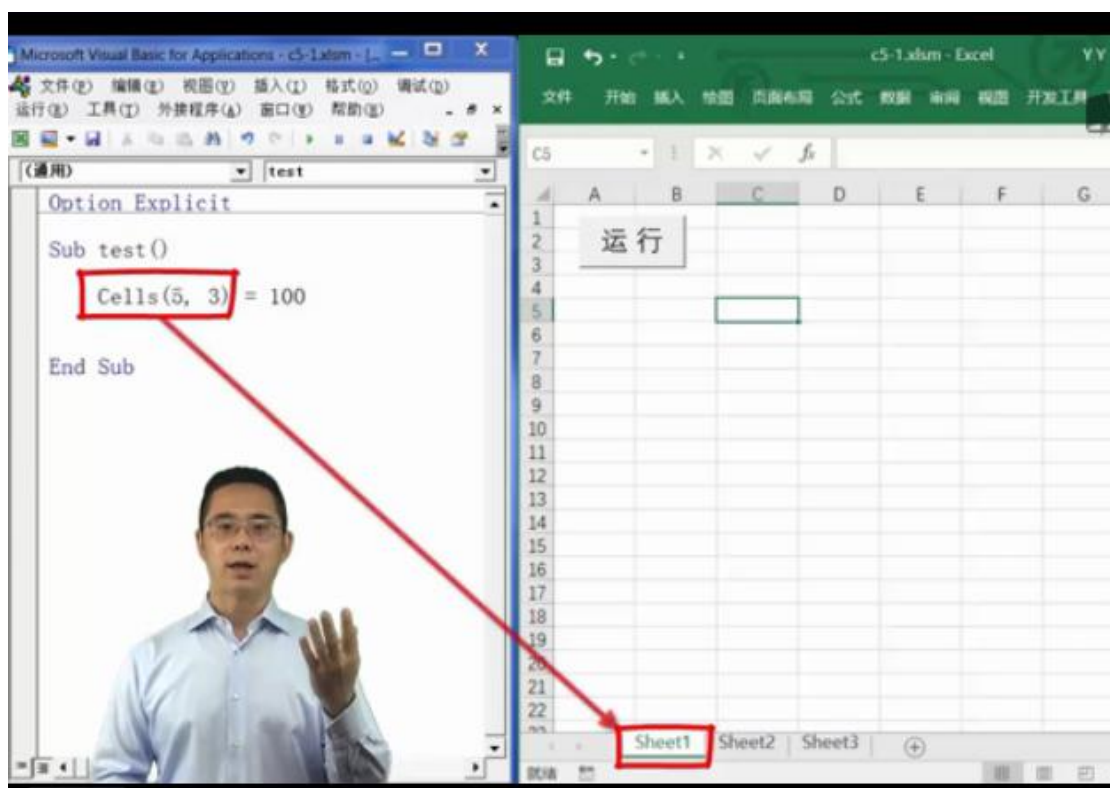
我想大家都明白，一定是 sheet1 这个工作表，可是为什么呢？我们并写下 sheet1 的第五行第三列啊。那么这就是 VBA 它的默认的

原则，VBA 认为如果你没有指定是那张工作表，它就认为是显示在屏幕上的这张工作表。

未指明工作表时，Cells(x,y)
代表当前显示在最上面（台
前）的工作表中，第x行第y
列单元格



我们知道 sheet1 显示出来了，而 sheet2 和 sheet3 并没有显示出来，只露出一个标签来，所以他就是 sheet1 的 cells(5,3)



那么我们提出一个新的要求，如我我们按下按钮之后，不处理 sheet1 这张表格，而是处理 sheet3 这张表格的 cells(5,3).写成 100。

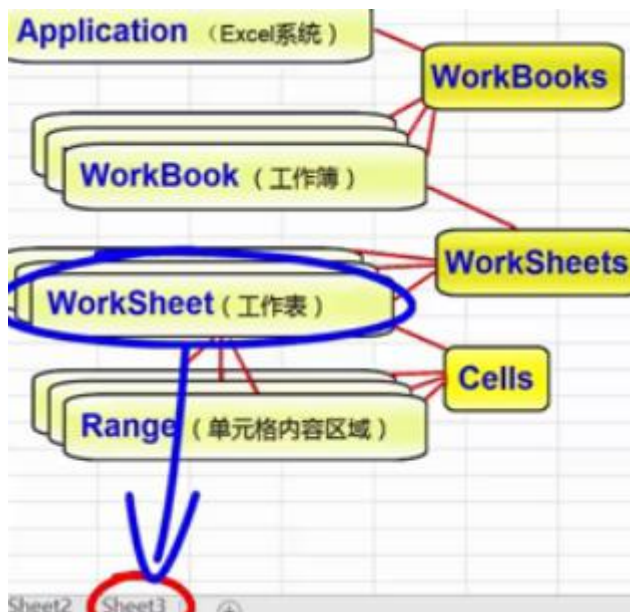
该怎么做呢？

我想大家应该这样理解，让“heet3 工作表的 cells(5,3)=100”

```
Option Explicit

Sub test()
    Sheet3工作表的Cells(5, 3) = 100
End Sub
```

这句话非常正确，那么怎么让计算机看懂呢？换句话说怎么在表示 sheet3 工作表这个东西呢？我想大家应该能想到，就是我们上节课讲的对象。我们说万事万物皆是对象，那么 sheet3 工作表这个事物，也是一个对象，我们可以用一个对象化来代表他。当然是 worksheet（工作表）类的对象了。



我们就声明 w1 这个变量是工作表类的对象。

```
Option Explicit

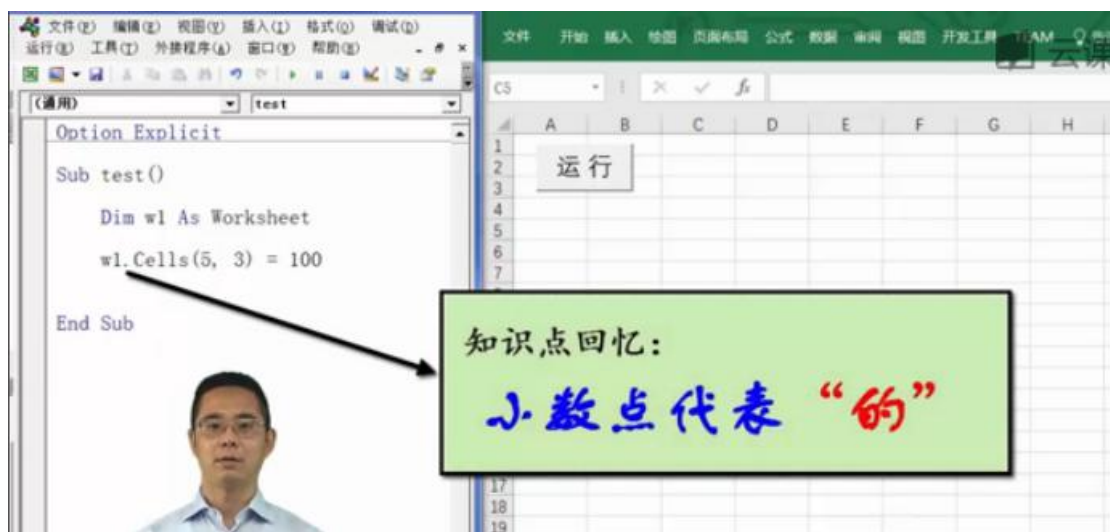
Sub test()

    Dim w1 As Worksheet

    Sheet3工作表的Cells(5, 3) = 100

End Sub
```

然后用 w1 的第五行第三列等于 100.



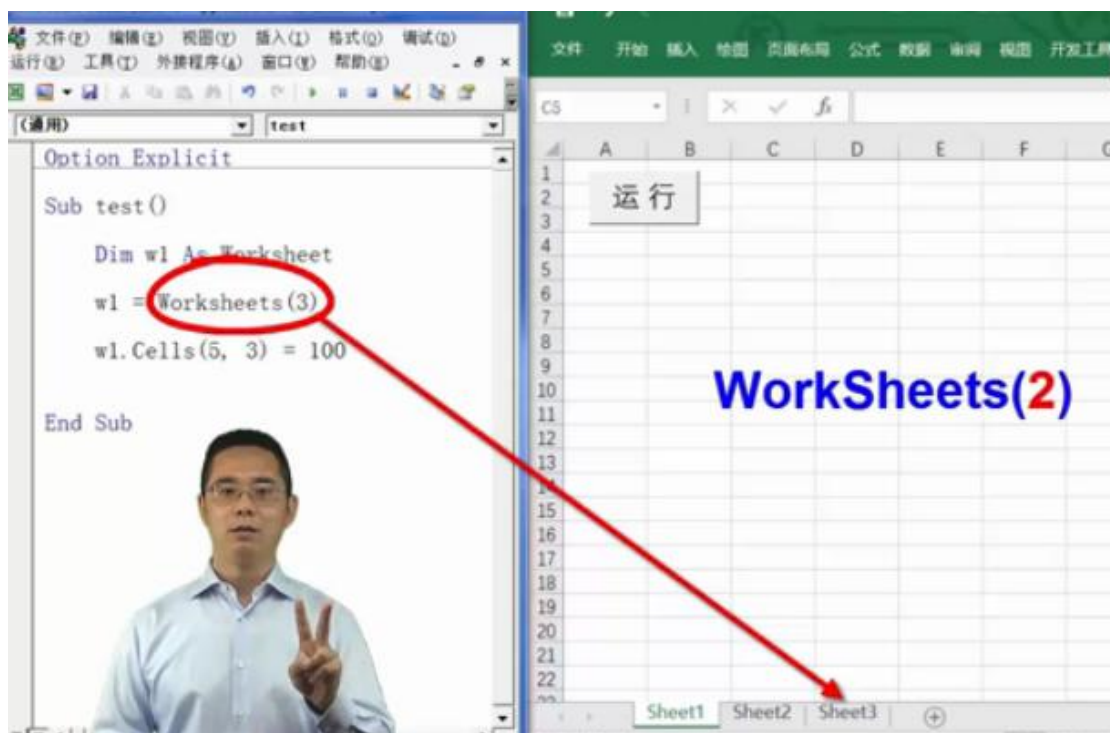
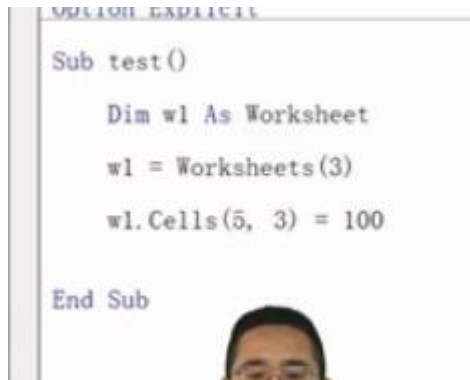
大家注意一下这个 as，它指明了这个变量属于什么类型。



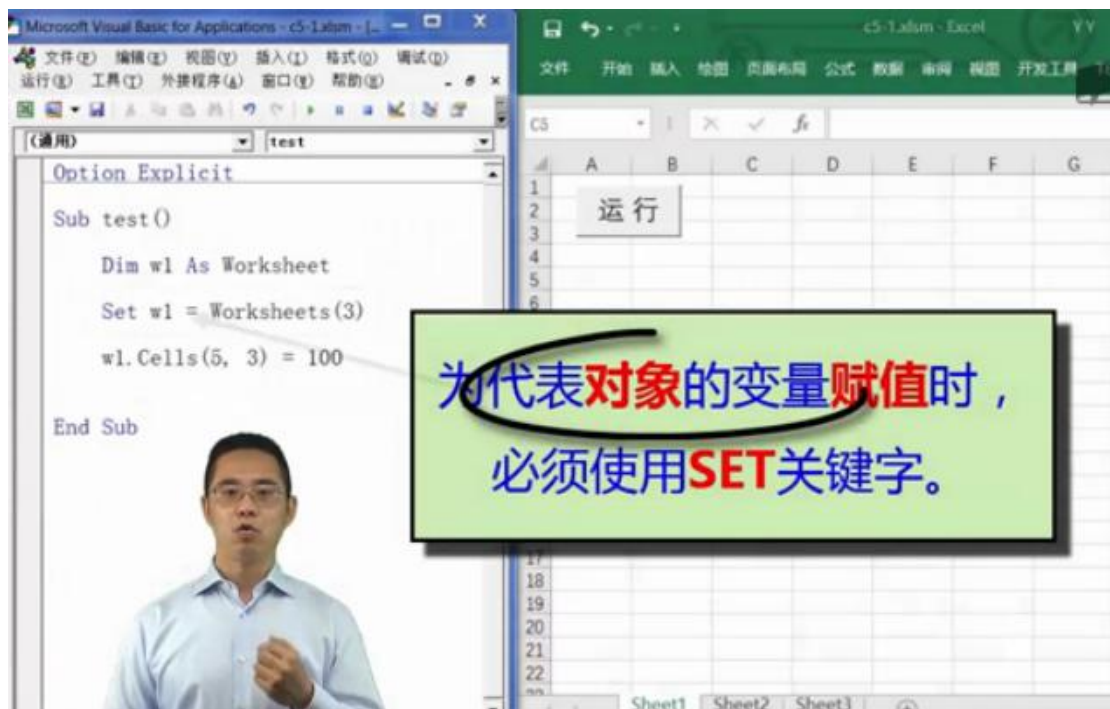
也就是说这个 cells (5,3) 是 w1 表中所代表的单元格。这回我们在看一下效果，按下按钮提示出错。



因为计算机说 **w1** 是一个工作表类的对象，我懂，但是他是那一个工作表的对象啊。这里有三张工作表，他是谁啊？你没有告诉我，。所以出错了。所以我们在使用之前需要制定 **w1** 具体指代那张工作表。怎么写呢？



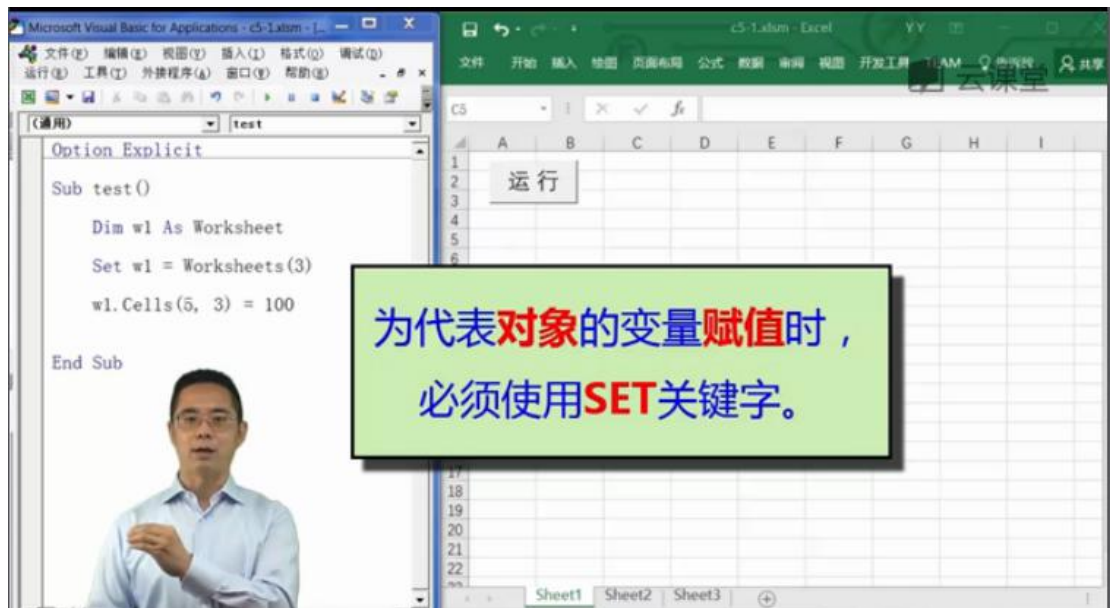
Worksheets(2)代表所有工作表中的第二个。运行让然出错，因为对于一个对象变量赋值；需要加一个 Set 关键词。



大家可能会问问什么会给对象一个特殊的待遇呢？总之对象就是特殊。为什么要使用 set,这其实涉及 VBA 自己可能发生的一种混淆。

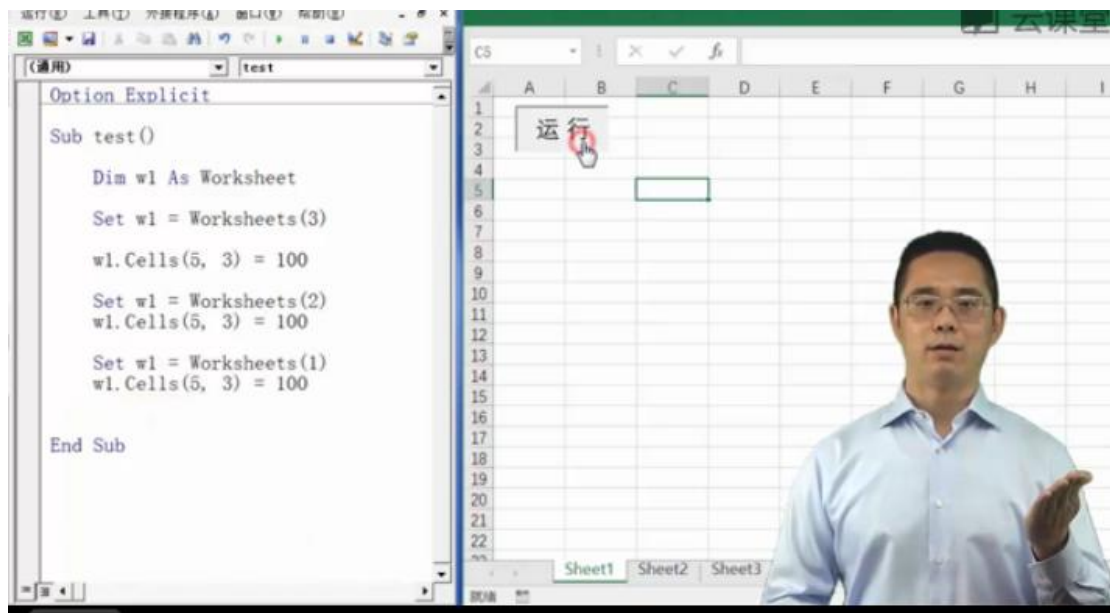


现在只需要记住，如果只指明一个对象它代表哪一个物体的时候一定要加上一个 set.

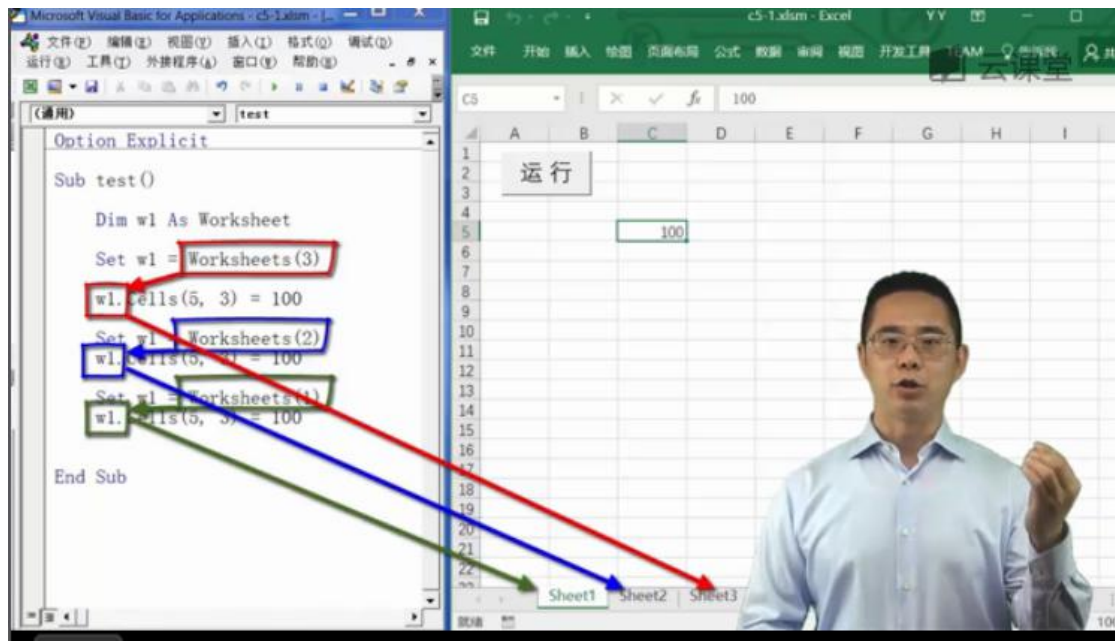


这回在摁下按钮，可以看到 sheet1 没有变化，而 sheet3 的 Cells(5,3) 赋值了 100.这一招隔山打牛还是很漂亮的吧。

现在回到代码里，我们在做一点修改，



运行一下，我们看一下，三张工作表 c5 单元格都赋值了 100.这是因为在程序的执行过程中，我们不断的修改 w1 的值，指向的表格也发生了变化。



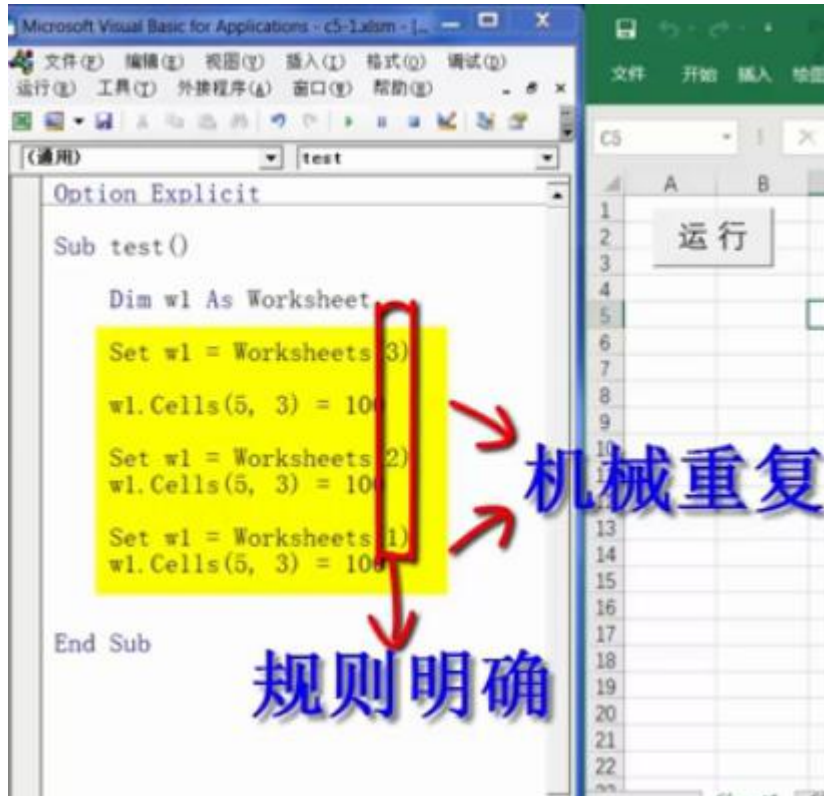
现在再看这个代码；

```
Set w1 = Worksheets(3)
w1.Cells(5, 3) = 100

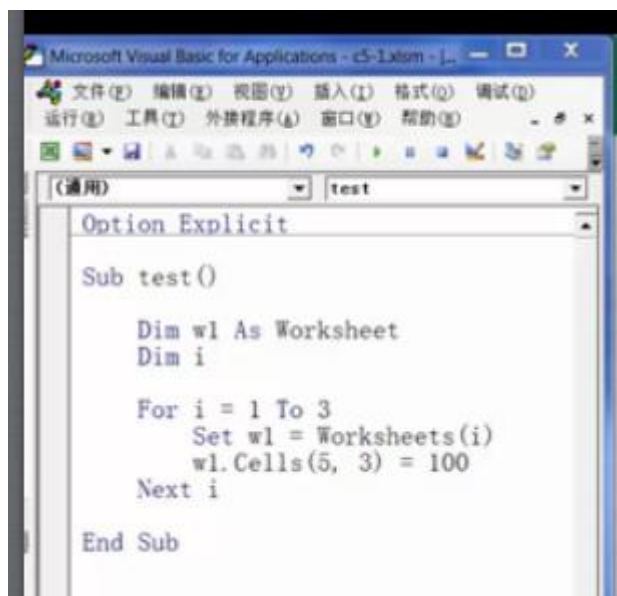
Set w1 = Worksheets(2)
w1.Cells(5, 3) = 100

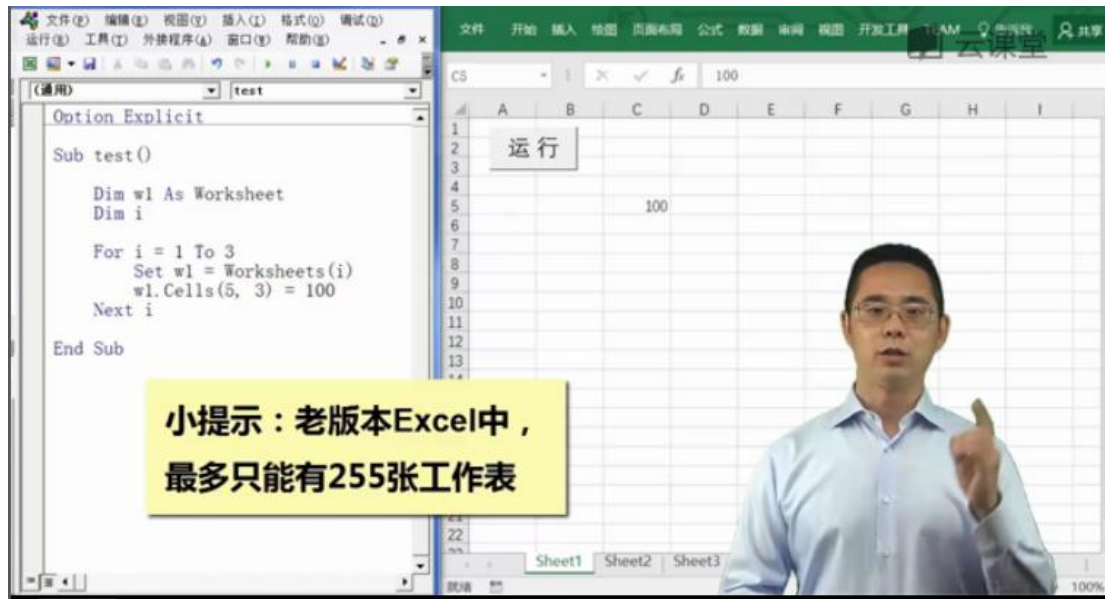
Set w1 = Worksheets(1)
w1.Cells(5, 3) = 100
```

大家看一下有什么问题和特点？我想大家都会念出 8 个子“机械重复，规则明确”



唯一的变化就是 worksheets 里面的索引号的变化，从 1 变到 3，从工作表 1 变到了工作表 3.修改程序。





如果工作簿总有 100 张工作表，就从 1 变化到 100，如果有 1000 张就从 1 变化到 1000.

如果我也不知道工作簿里有多少个工作表？那该怎么办？

有一个办法可以让我们知道，

Worksheets的Count属性：
代表了所属工作簿中
一共有多少个工作表。

修改后就可以修改所有的工作表：

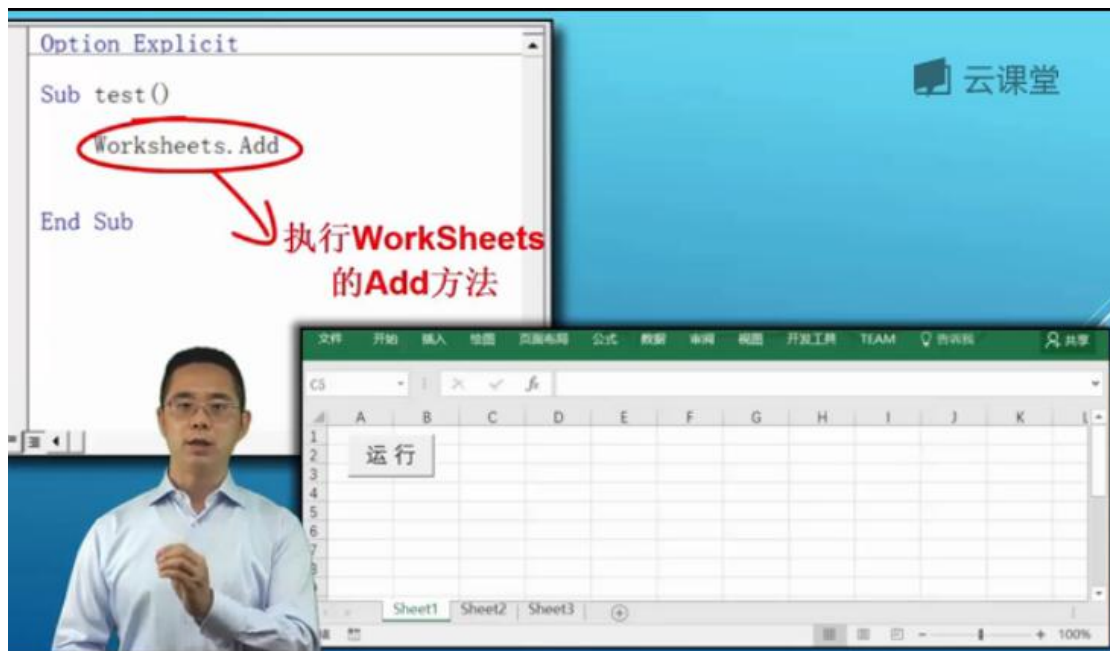
```
For i = 1 To Worksheets.Count
    Set w1 = Worksheets(i)
    w1.Cells(5, 3) = 100
Next i
```

那么有的同学可能会问？工作簿中能不能自动的添加一个工作表进去呢？

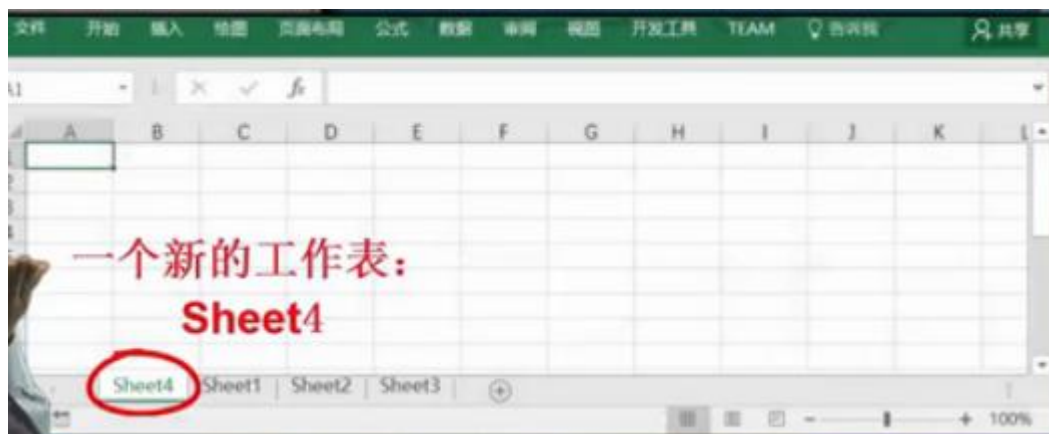


比如我们三张工作表都计算结束后，把计算结果都放到第四张工作表中，那么该怎么办呢？这个时候我们可以用到一个 **worksheets** 的一个方法叫做 **add** 方法。

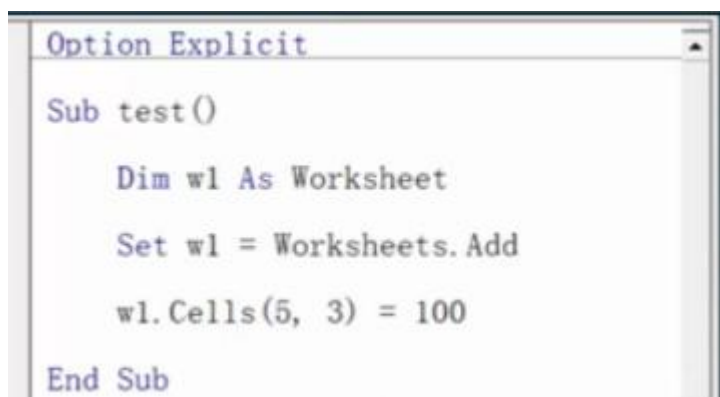
云课互
WorkSheets的Add方法：
在所属工作簿中新建
一张工作表



这句话执行的结果就是添加一个新的工作表。



更进一步的我们想操作这个工作表，可以这样写；



Option Explicit

```
Sub test()  
    Dim w1 As Worksheet  
    Set w1 = Worksheets.Add  
    w1.Cells(5, 3) = 100  
End Sub
```

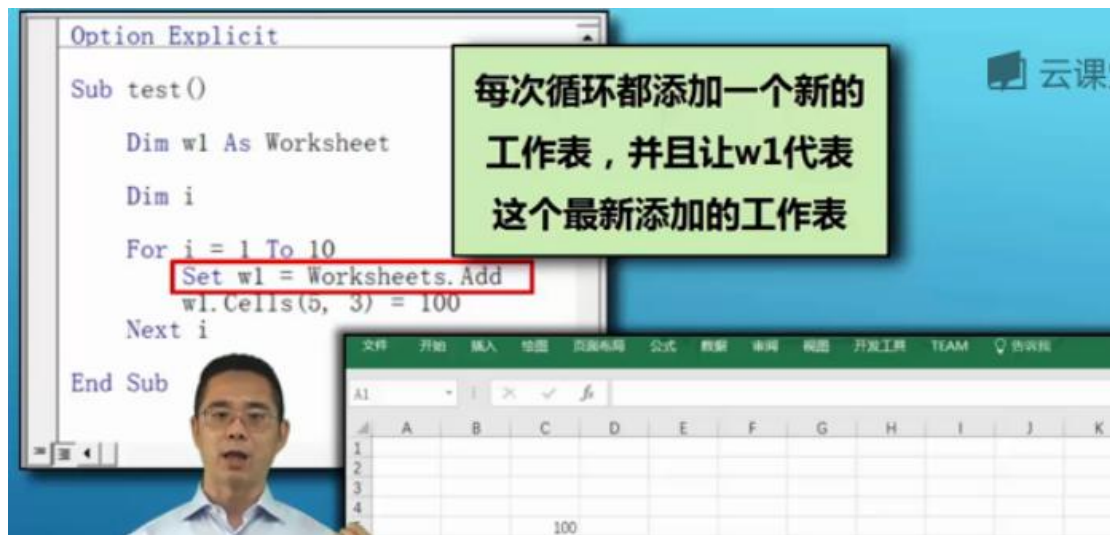
添加一个新的工作表 (Sheet5)

Option Explicit

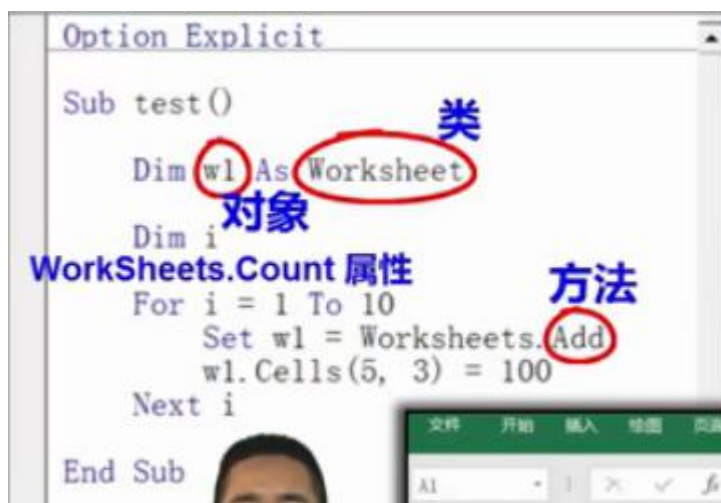
```
Sub test()  
    Dim w1 As Worksheet  
    Set w1 = Worksheets.Add  
    w1.Cells(5, 3) = 100  
End Sub
```

添加一个新的工作表 (Sheet5)

下面的程序，每新建一个 sheet，就进行一次赋值操作，



大家可以看一下，我们有了类、对象、属性、方法就能轻松的操作工作表这些都对象了。



工作表是我们处理 excel 是最经常处理的操作，所以下节课我们让然处理工作表的技巧。