



Excel 名称应用技术大全

完美 Excel 出品
微信公众号: [excelperfect](#)



内容提要

定义名称是 Excel 中一项非常有用的功能，不仅可以让数据更易理解、更方便使用公式和数据，而且可以突破很多 Excel 的局限。通过在数据有效性、条件格式、公式、图表、VBA 中灵活应用名称，可以创作出让人眼界大开的作品来。

本电子书《[Excel 名称应用技术大全](#)》详细讲解了 Excel 中定义名称的应用，帮助读者快速掌握并发挥 Excel 提供的有用功能。



目录

主要内容.....	1
1. 名称概述.....	1
2. 如何定义名称	3
使用“名称框”定义名称.....	3
使用“新建名称”对话框.....	4
直接使用工作表中的文本快速定义名称.....	6
命名常量.....	7
命名公式.....	8
名称的命名约定.....	9
3. 名称的秘密.....	11
名称引用单元格的方式.....	11
名称与数组.....	13
名称与公式.....	14
工作表级名称.....	15
使用其他工作簿的名称.....	15
特殊名称.....	15
快速输入名称.....	16
应用名称到已有的公式.....	16



4. 管理名称.....	17
查看名称.....	18
新建名称.....	18
编辑名称.....	19
删除名称.....	20
5. 名称的使用.....	21
将公式定义为名称.....	22
使用名称创建级联下拉列表.....	24
在条件格式中使用名称.....	25
在图表中使用名称.....	26
6. 动态的名称.....	27
7. 在图表中应用名称.....	29
创建动态更新的图表.....	29
使用滚动条控制图表.....	30
使用复选框控制图表系列.....	32
小结.....	34
8. VBA 中对名称的基本操作.....	35
创建名称.....	35
重命名现有的名称.....	37
改变所选区域的名称的引用区域.....	37
提取/引用命名区域.....	38
隐藏名称.....	38
删除名称.....	39
9. VBA 操作名称的 8 个代码示例.....	41



代码示例 1：检查当前工作簿中是否存在某名称	41
代码示例 2：使工作簿中的所有名称都可见	42
代码示例 3：列出当前工作簿中所有名称的相关信息	42
示例代码 4：列出所有隐藏的名称	43
示例代码 5：显示当前单元格所命名的名称	43
示例代码 6：删除当前工作簿中含有“excel”字符的名称	44
示例代码 7：判断某单元格或单元格区域是否与命名区域部分重叠	44
示例代码 8：为名称框定义快捷键	45
10. 名称在 VBA 代码中的应用示例—公式	47
11. 名称在 VBA 代码中的应用示例——数据有效性	51
12. 名称在 VBA 代码中的应用示例——	55
筛选数据	55
目的	55
示例工作簿与程序代码	55
小结	63
示例工作簿下载	63
13. 名称在 VBA 代码中的应用示例——自动绘制图表	65
14. 名称在 VBA 代码中的应用示例——绘制动画图表	67
关于完美 Excel	71
完美 Excel 微信公众号使用指南	72





主要内容

定义名称是 Excel 中一项非常有用的功能，不仅可以让数据更易理解、更方便使用公式和数据，而且可以突破很多 Excel 的局限。通过在数据有效性、条件格式、公式、图表、VBA 中灵活应用名称，可以创作出让人眼界大开的作品来。

本书各章节主要内容如下：

1. 概述

讲解了名称的定义，概要叙述了使用名称的一些优势。

2. 如何定义名称

介绍了定义名称的方法，以及命名约定。

3. 名称的秘密

详细讲解名称引用单元格的方式、名称与数组、名称与公式，以及工作表级名称（局部名称），如何使用其他工作簿中的名称，Excel 内置名称，如何快速输入名称，如何应用名称到已有公式。

4. 管理名称

详细介绍 Excel 的“名称管理器”，包括查看名称、新建名称、编辑名称和删除名称。



5. 名称的使用

名称的一些使用方法，例如定义公式的名称的使用、创建级联下拉列表、在条件格式中使用名称、在图表中使用名称。

6. 动态的名称

使用命名公式，定义动态的名称。

7. 在图表中应用名称

通过使用名称作为图表系列，创建动态更新的图表、创建交互式的图表。

8. VBA 中对名称的基本操作

使用 VBA 代码创建名称、重命名现有的名称、改变所选区域的名称的引用区域、提取/引用命名区域、隐藏名称、删除名称。

9. VBA 操作名称的 8 个代码示例

列举了 8 个示例代码，有些可以作为通用代码使用，以便更好地理解名称。

10. 名称在 VBA 代码中的应用示例—公式

在 VBA 代码使用的公式中使用名称。

11. 名称在 VBA 代码中的应用示例—数据有效性

通过在 VBA 代码中使用名称作为数据有效性验证数据。

12. 名称在 VBA 代码中的应用示例—数据筛选

通过示例展示如何使用名称来高效筛选数据。



13. 名称在 VBA 代码中的应用示例—自动绘制图表

使用动态命名区域来创建图表数据，并结合表单控件和 VBA 代码来调整图表的绘制比例和图表元素。

14. 名称在 VBA 代码中的应用示例—绘制动画图表

通过使用名称定义的数据，结合少量的 VBA 代码，实现动画图表。



温馨提示：

在完美 Excel 微信公众号中发送消息：名称应用，即可获得本电子书文档下载链接和密码。



如果您对本书有什么建议或者还有什么好的示例,欢迎前往完美 Excel 微信公众号沟通交流。

欢迎分享本电子书，让更多的人方便地得到所需要的知识。



本章内容 2018 年 6 月 11 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
[Excel 技术|名称 1: 概述](#)

1. 名称概述

名称是 Excel 的一项强大的功能。

所谓名称，就是给单元格、单元格区域、公式或者常量（当然，也可以是图表或形状）起一个通俗易懂的名字，以后这个名字就代表相应的单元格、单元格区域、公式或者常量，就可以像 Excel 内部元素一样在工作表操作中使用这个名字。

例如，在下图 1.1 所示的工作表中，在单元格 B3 中输入公式：

=B1-B2

求出收入减去支出后的节余。

	B3		<i>f_x</i>	=B1-B2
	A	B	C	
1	收入：	200		
2	支出：	100		
3	节余：	100		
4				

图 1.1

如果不仔细看工作表，只看这个公式，我们当然不清楚它的意思。然而，如果我们将单元格 B1 命名为“收入”，单元格 B2 命名为“支出”，那么计算节余的公式也可以写成：

=收入-支出

如下图 1.2 所示：



B3		\sum	=收入-支出	
	A	B	C	
1	收入：	200		
2	支出：	100		
3	节余：	100		

图 1.2

结果相同，但意思更清楚。

从上面这个简单的示例可以看出，使用名称增强了公式的可读性，使公式更直观且更易于理解。不仅仅是这样，以后我们将会进一步看到，养成给单元格或单元格区域命名的习惯，将会给你随后的工作表操作带来方便。

实际上，创建一个名称就是创建了一个命名公式，它存放于内存中。在使用创建的名称时，会调用并计算该名称表示的公式并返回供呈现或者进一步使用的结果。因此，就像公式一样，在名称中可以使用工作表函数，也可以利用相对引用动态扩展，甚至还可以使用 XLM 宏函数实现 VBA 的功能。

此外，可以将数值、文本或者数组定义为名称，供我们在工作簿中使用。如果工作簿中有多处使用同一名称，那么只需修改该名称代表的值或者引用，就可以一次性完成对所有使用该名称的数据修改。对使用名称的区域进行添加或删除操作时，名称引用区域会自动调整。因此，工作表更容易维护。

还有特别重要的一点是，将名称与 VBA 相结合，可以编写出简洁且功能强大的代码，更有效地实现我们的自动化操作。

最后，一个有意义的名字当然比单元格地址或者长长的公式更容易记住。难道不是这样的吗？

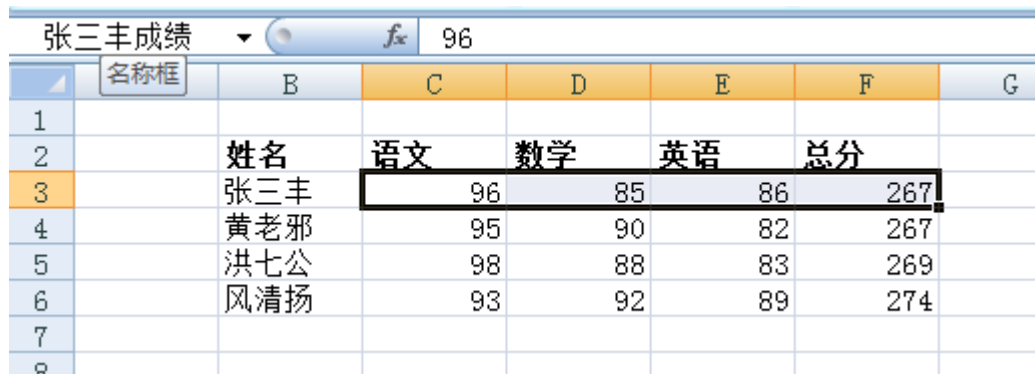


2. 如何定义名称

在 Excel 中，可以通过多种方式来定义名称。下面，我们将分别介绍定义名称的方法。

使用“名称框”定义名称

名称框在编辑栏的左侧，通常我们在工作表单元格中操作时，其中显示单元格的地址。其实，我们可以使用名称框来快速定义名称，如图 2.1 所示，首先选择要定义名称的单元格区域，然后直接在名称框中输入名称。



The screenshot shows the Excel Name Box on the left, which contains the text '张三丰成绩' (Zhang Sanfeng's Scores). The main table displays scores for four subjects: Chinese (语文), Math (数学), English (英语), and Total Score (总分). The data is as follows:

	姓名	语文	数学	英语	总分
3	张三丰	96	85	86	267
4	黄老邪	95	90	82	267
5	洪七公	98	88	83	269
6	风清扬	93	92	89	274

图 2.1

在定义了名称后，我们可以通过名称框直接选择相应的单元格区域。如图 2.2 所示，单击名称框右侧的下拉箭头，会出现已定义的名称下拉列表，选取名称后，会选择该名称相应的单元格区域。



图 2.3 所示的“新建名称”对话框中，我们可以通过“范围”列表来指定名称的作用范围，如图 2.4 所示。通常，Excel 默认为“工作簿”，即定义的名称在工作簿的所有工作表中都可使用，称之为工作簿级名称。因此，如果我们将“数学成绩”定义为工作簿级名称，那么虽然该名称对应的数据在工作表 Sheet1 中，但我们可以其他工作表例如 Sheet5 中使用这个名称。

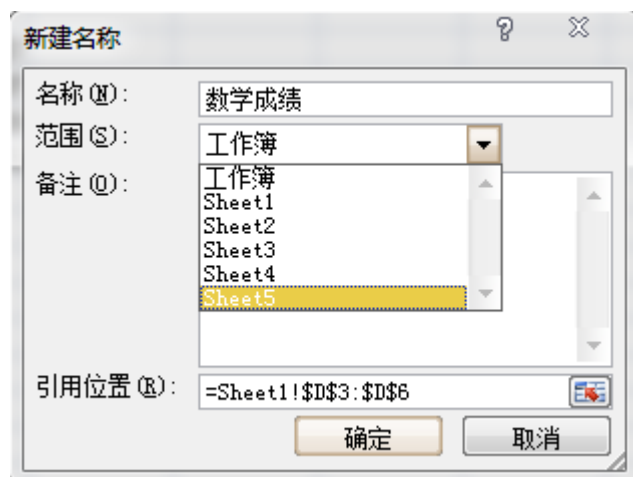


图 2.4

在定义名称时，我们可以在“范围”列表中选择应用该名称的工作表，如图 2.5 所示，这称之为工作表级名称。如图 5，我们定义“英语成绩”为工作表 Sheet1 中使用的名称，那么它只能在工作表 Sheet1 中使用。使用工作表级名称的好处是，我们可以在不同的工作表中定义相同的名称，只是它们作用的工作表不同而已。

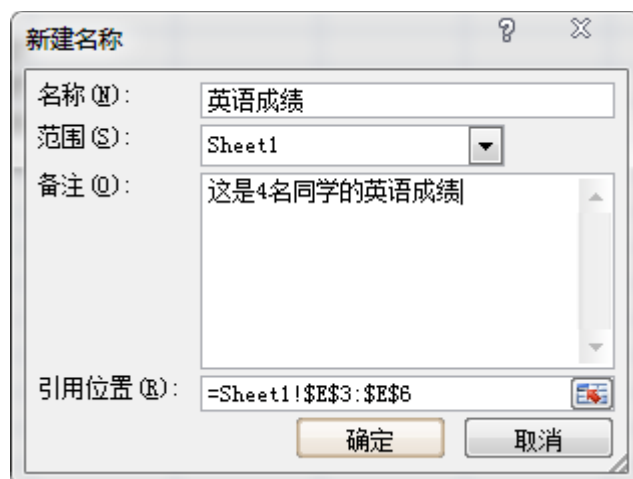


图 2.5

单击“名称框”下拉箭头，在列表中只会显示工作簿级名称和本工作表名称，定义作用范围为其他工作表的名称不会显示。



直接使用工作表中的文本快速定义名称

通常，我们先选择单元格区域，然后打开“新建名称”对话框时，Excel 会提供行列标题作为建议名称，如图 2.6 所示。

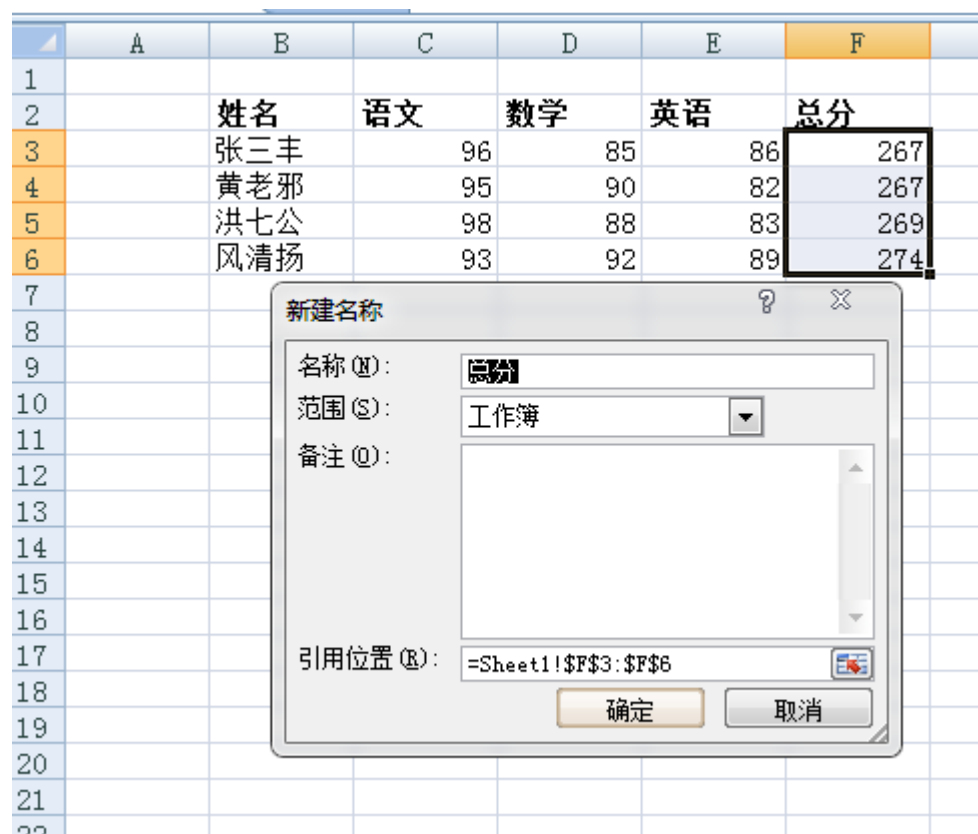


图 2.6

其实，Excel 可以智能辨识相应的文本作为单元格区域名称。如图 2.7 所示，选择包含要将其用作名称的文本单元格的单元格区域 C2:F6，单击“公式——根据所选内容创建”，在弹出的“以下列选定区域的值创建名称”中选取“首行”，单击确定即为每列数据定义了名称。



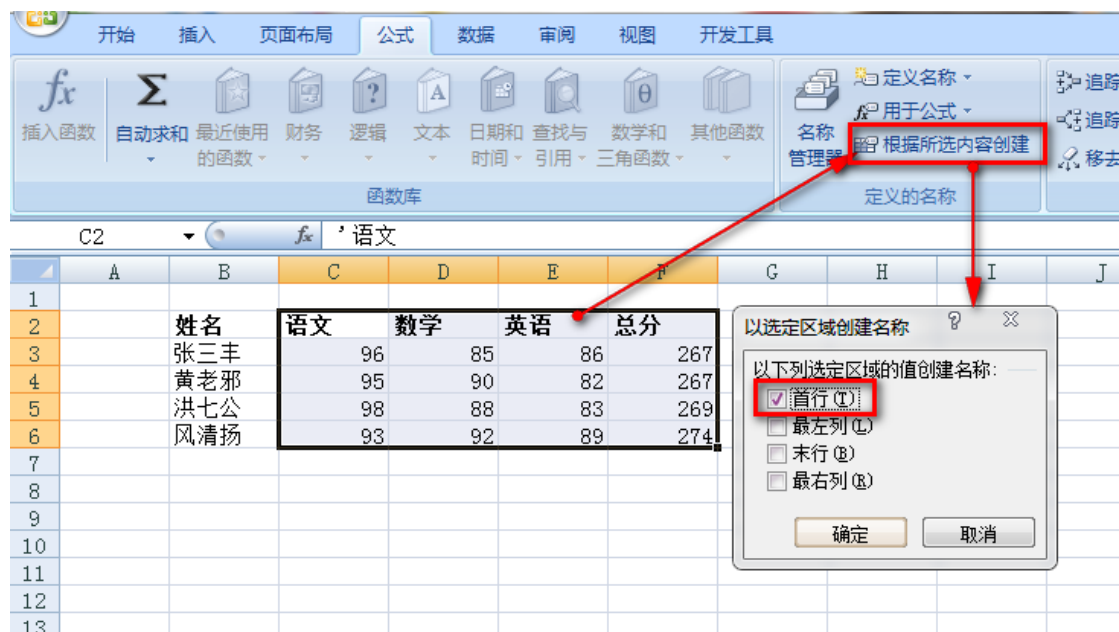


图 2.7

如果选择包括行列的区域，如图 2.8 中的单元格区域 B2:F6，那么在“根据所选内容创建”名称时，将为每行和每列分别指定名称，并且将单元格区域的左上角单元格的内容作为整个数据区域的名称，即单元格区域 C3:F6 的名称为“姓名”。

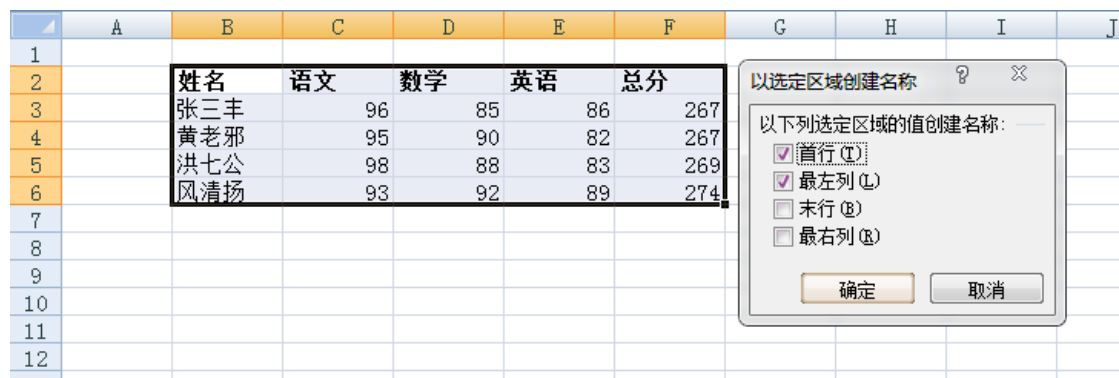


图 2.8

命名常量

不仅可以命名工作表中的单元格区域，还可以为常量值定义名称，这是一个非常有用的功能，特别是用于公式或 VBA 代码中，我们后面的名称运用中还会讲到。

命名常量非常简单，在“新建名称”对话框的名称文本框中输入常量名，在引用



位置输入=加上常量值，如图 2.9 所示。这样，就可以在公式或 VBA 代码中使用这个名称常量了。

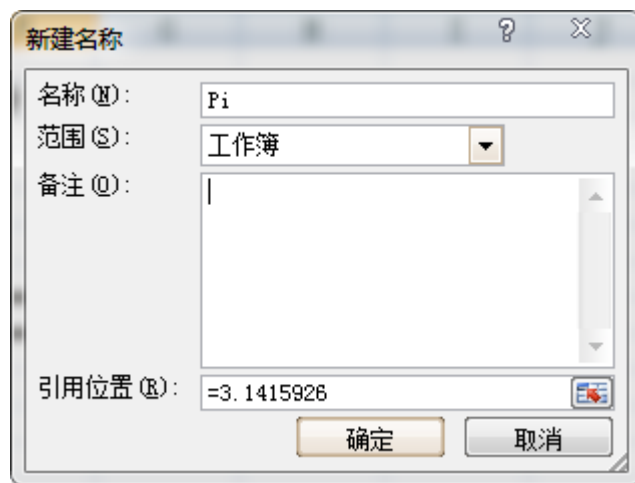


图 2.9

还有一种命名常量的方式，即在单元格中放置常量值，在引用位置中引用该单元格即可。

命名公式

可以为常用的公式指定一个名称，这样在单元格中不需要输入公式，只需要使用该名称，让工作表可读性更强。

例如，使用“新建名称”对话框为计算圆面积的公式定义名称 Circle，如图 2.10 所示，其中 Radius 为含有半径值的单元格名称。此时，在工作表中直接输入：

=Circle

即可求得指定半径的圆的面积。





图 2.10

后面的章节中，我们会讲到使用此技巧来定义动态的名称。注意，常量命名和公式命名都不会出现在名称框中。

名称的命名约定

在命名名称时，需要遵守一些规则：

- 命名的名称最长不能超过 255 个字符。
- 命名的名称必须以字母（汉字）或下划线开头，在名称的其他部分可使用任意字母、数字或符号，但不能使用空格。
- 名称不区分大小写。
- 在名称命名时，不能使用单元格地址或者任何运算符。
- 不要使用 Excel 的内置名称。
- 命名名称时，在保持简单的同时尽量让名称表现相应的含义。





3. 名称的秘密

在上一章中，我们详细介绍了定义名称的不同方法。下面，我们将探讨一下名称中的一些细节，以便更好地理解和使用名称。

名称引用单元格的方式

在定义名称时，Excel 默认使用单元格绝对引用，如图 3.1 所示。这表明，无论怎样，名称“姓名”总是指向单元格区域 B3:B6。

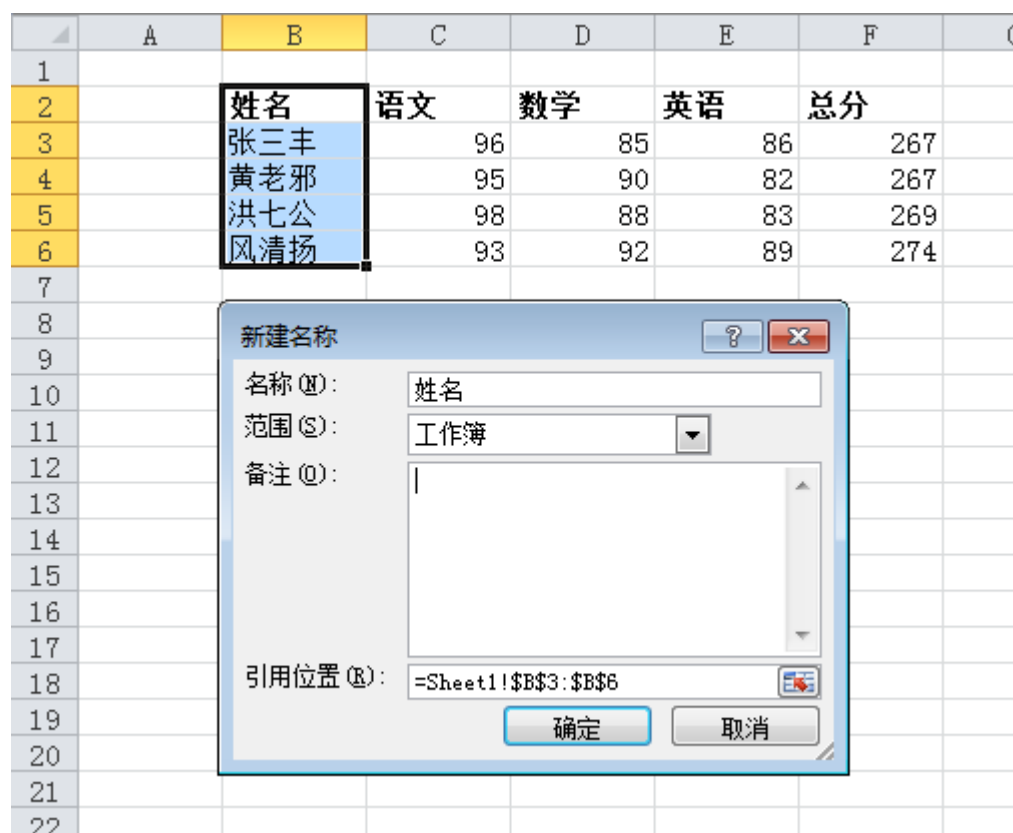


图 3.1



然而，也可以在定义名称时使用单元格相对引用，这将带来很多有趣的应用。例如，如图 3.2 所示，当前单元格为工作表 Sheet2 中的单元格 C2，定义的名称 LeftUp 的引用位置为：

=Sheet2!B1

使用的是相对引用，即该名称将指向当前单元格向左一列向上一行的单元格。

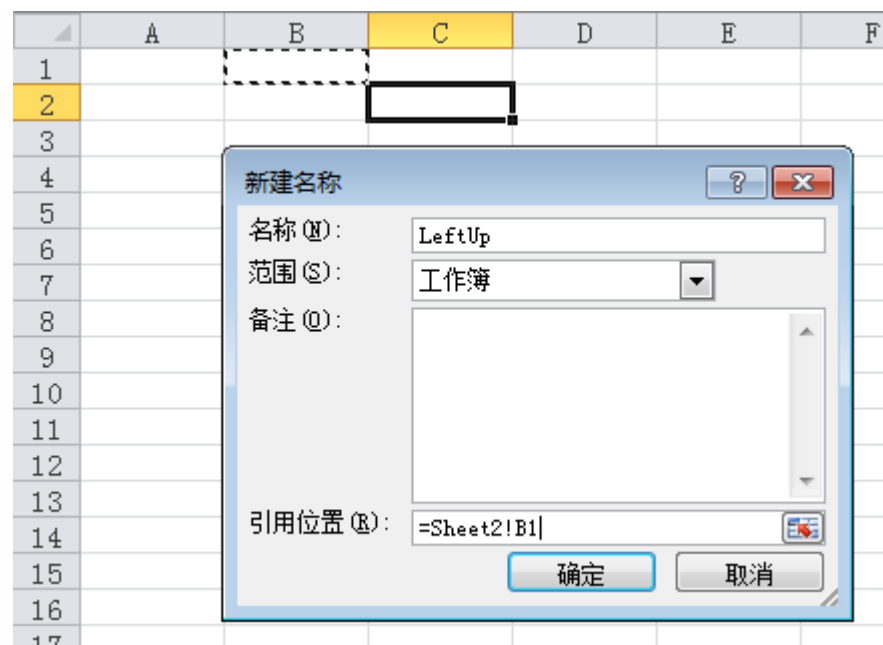


图 3.2

因此，当在工作表的单元格 B2 中输入公式：

=LeftUp

将引用单元格 A1，如图 3.3 所示。

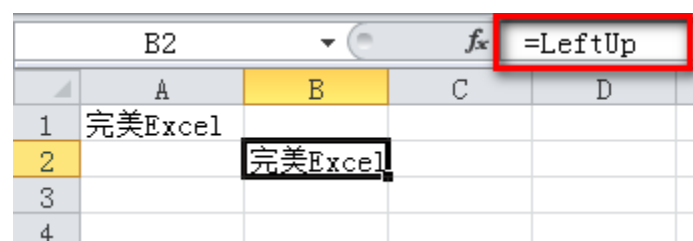


图 3.3

如果我们创建名称：**TotalCells**

引用位置：**=SUM(!A1:!A10)**

那么，在工作表第 10 行以下的任意单元格中输入公式：

=TotalCells



将返回该单元格上方 10 个单元格数值之和。

也可以使用混合引用来创建名称。如果创建名称：**ValueInFirstCol**

引用位置：**=!\$A1**

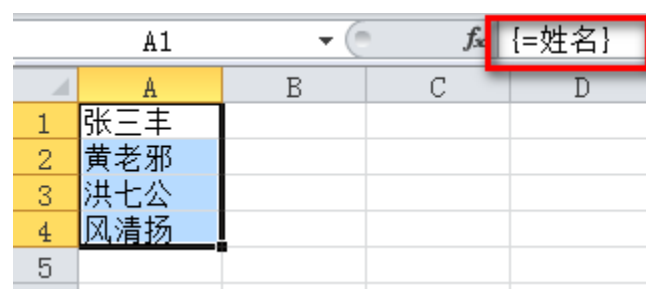
将总是返回工作表当前单元格所在行的列 A 中的值。

名称与数组

在上文图 3.1 中，我们定义了名称“姓名”，那么这个名称实际上是一个数组。选取单元格区域 A1:A4，输入

=姓名

按 Ctrl+Shift+Enter 组合键，结果如图 3.4 所示，得到名称引用的 4 个单元格的数据。



	A1			
	A	B	C	D
1	张三丰			
2	黄老邪			
3	洪七公			
4	风清扬			
5				

图 3.4

在使用“数据有效性”功能时，我们经常会运用到名称，使用该名称引用的单元格区域作为下拉列表的数据源，如图 3.5 所示。



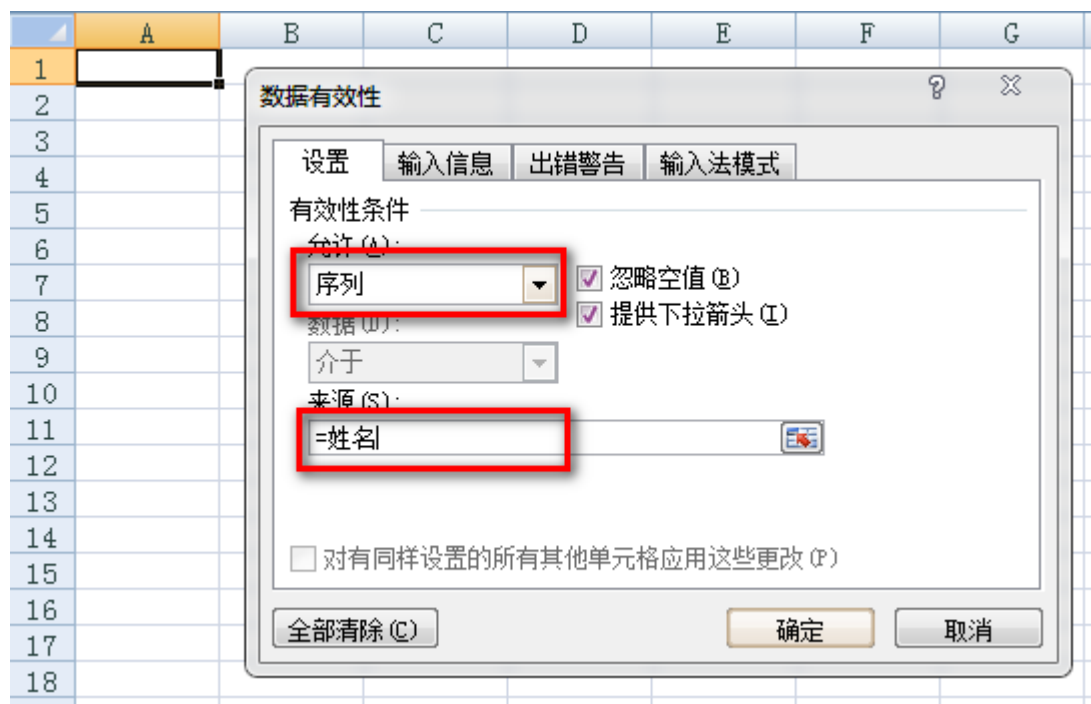


图 3.5

结果如图 3.6 所示。此外，使用名称，我们可以将其他工作表中的单元格区域作为数据源。

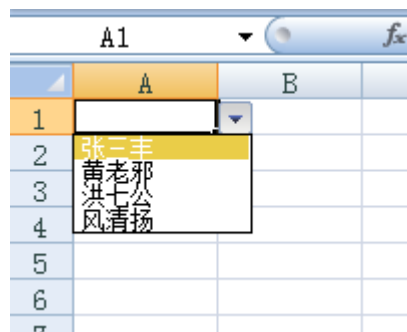


图 3.6

名称与公式

如果定义的名称引用的是一个公式，那么该公式将被评估为是一个数组公式。



工作表级名称

上一篇文章介绍过，在定义名称时可以定义名称的作用范围，默认为工作簿级名称。可以定义为工作表级名称，只能在该工作表中使用。

如果工作表名中含有空格或者其他特殊字符，那么在定义工作表级名称时必须在工作表名两侧添加单引号，例如：

`'完美 Excel'!MyWorksheetName`

`'完美 Excel (1)'!MyName`

如果工作表级名称与工作簿级名称同名，那么在定义工作表级名称的工作表中使用，将会使用该工作表级名称，而在其他工作表中使用这个名称时，则会使用工作簿级名称。如果要在其他工作表中使用某工作表级名称，则要在名称前加上该工作表名。

使用其他工作簿的名称

如果要使用“完美 Excel.xlsx”工作簿中的名称“excelperfect”，输入如下：

`= '完美 Excel.xlsx'!excelperfect`

如果名称“excelperfect”为工作表级名称，工作表为 Sheet1，则输入如下：

`= '[完美 Excel.xlsx]Sheet1'!excelperfect`

特殊名称

Excel 定义了一些内置的名称（有时称之为隐藏的名称），你可以在 VBA 代码中灵活使用这些名称，但是最好不要人为修改这些名称，或者自己命名与之同名的名称。

下面列出了一些 Excel 定义的名称：

Print_Area、Print_Titles、Consolidate_Area、Database、Criteria、



Extract、FilterDatabase、Sheet_Title

例如，如果在工作表中设置了一个打印区域，那么 Excel 会为此区域创建名称 Print_Area。一些 Excel 加载项也会创建隐藏的名称。

快速输入名称

在单元格中输入名称时，随着输入，Excel 会自动感应出以输入的文本开头的名称和函数，如图 3.7 所示，此时选择要输入的名称，按 Tab 键即可完成输入。

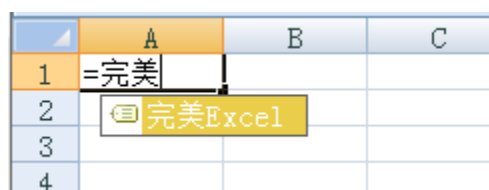


图 3.7

应用名称到已有的公式

如果工作表中已经定义有公式：

=A1+B1

后来，定义名称 FirstCell 引用单元格 A1，名称 SecondCell 引用单元格 B1，那么可以选取使用公式的单元格，然后单击定义名称中的应用名称，选择相应的名称，单击确定，Excel 将公式变为：

=FirstCell+SecondCell



4. 管理名称

使用 Excel 的名称管理器，可以方便地管理名称。单击“公式”选项卡的“定义的名称”组中的“名称管理器”，调出“名称管理器”对话框，如图 4.1 所示。

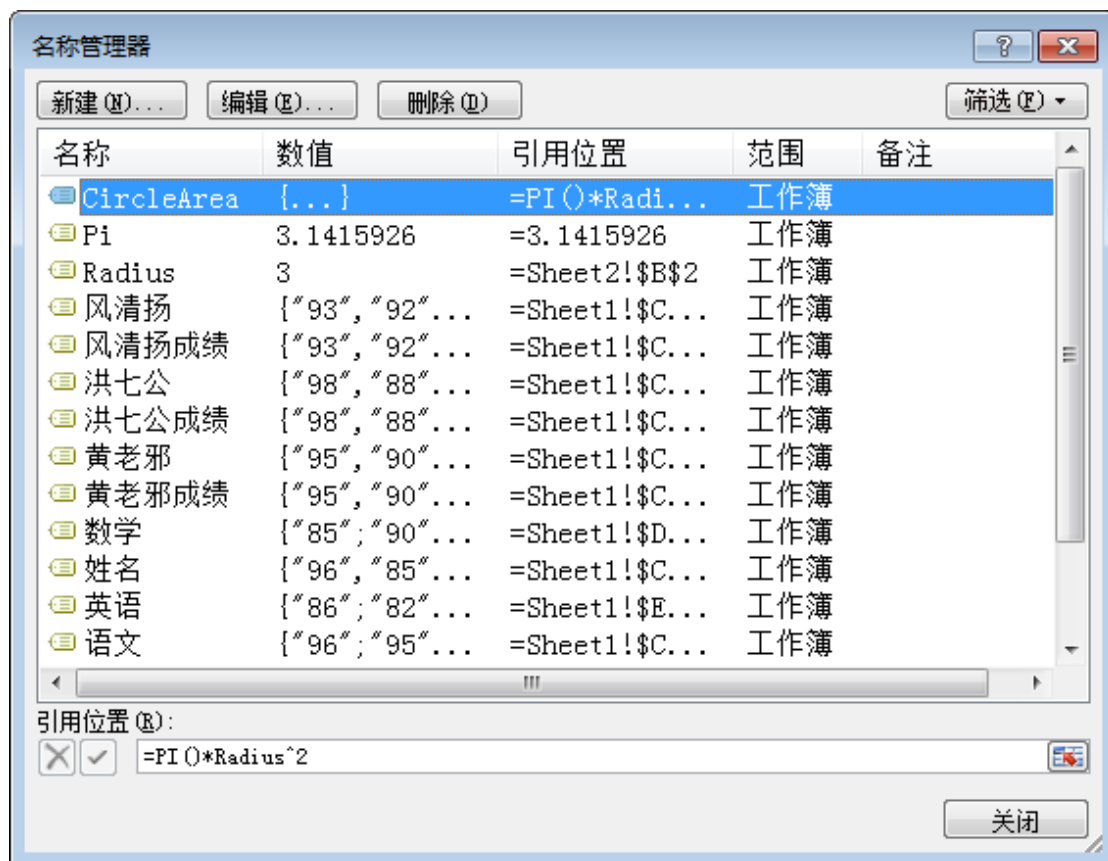


图 4.1

在“名称管理器”对话框中，我们可以查看、新建、编辑、删除名称。



查看名称

在“名称管理器”对话框中，可以方便地查看工作簿中的所有名称。单击某名称后，在下方的“引用位置”中，可以看到名称引用的单元格区域或者公式。可以通过拖动对话框右下角来调整其大小，从而完整地显示引用位置。也可以拖动列表中的列宽，显示由于空间不够而隐藏的内容。还可以单击列表标题来给名称排序，通过筛选器只显示特定的名称，如图 4.2 所示。

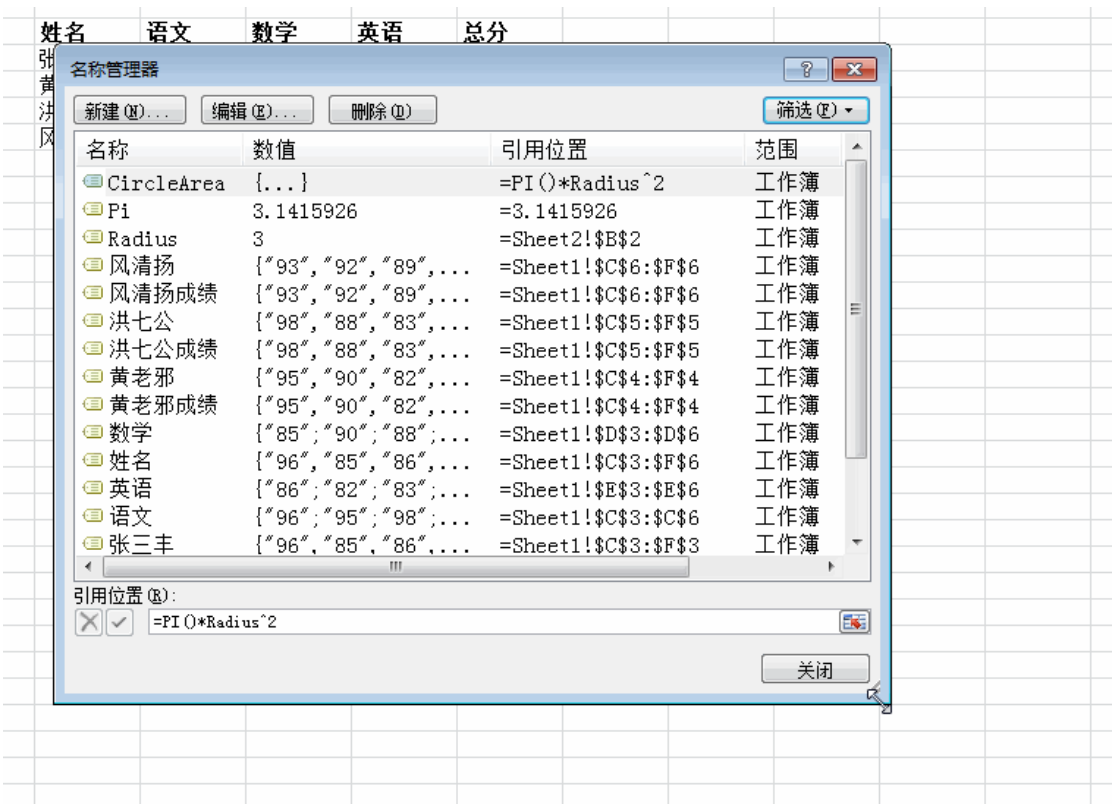


图 4.2 (演示视频截图，详细视频见完美 Excel 微信公众号)

新建名称

单击“名称管理器”对话框顶部的“新建”按钮，打开“新建名称”对话框，如图 4.3 所示。



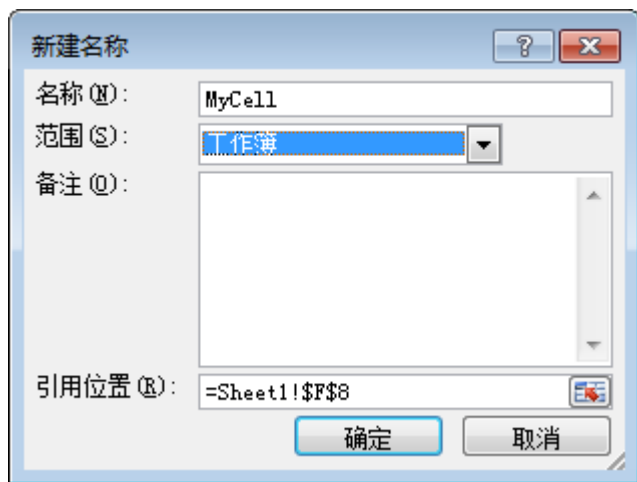


图 4.3

与直接单击“公式”选项卡中的“定义名称”一样，可以在“新建名称”对话框中命名名称，设置该名称的作用范围和引用位置，添加名称的备注。在“引用位置”中会自动显示当前选择的单元格区域地址，可以手动输入或者单击其右侧的单元格选取按钮来设置引用位置。

编辑名称

在“名称管理器”对话框中，选择要编辑的名称，在该对话框下方的引用位置中对名称进行修改，然后单击其左侧带有勾的按钮，如图 4.4 所示。

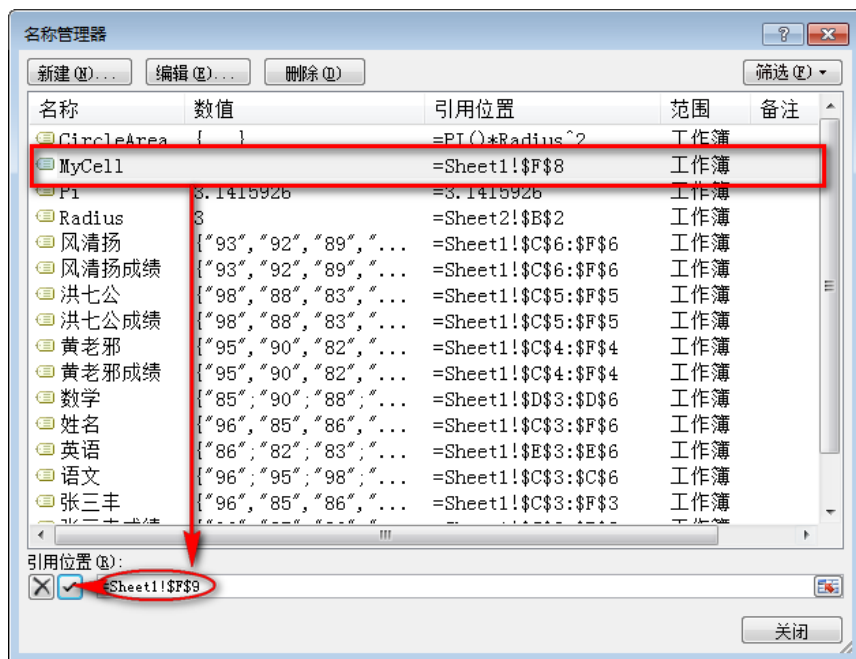


图 4.4



如果修改了名称而没有单击其左侧带有勾的按钮，在关闭对话框或者选择其他名称时，Excel 会弹出如图 4.5 所示的信息提示。

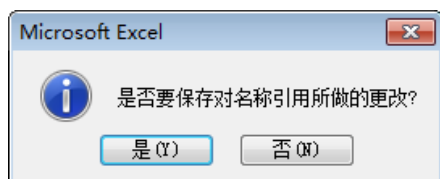


图 4.5

还可以单击“名称管理器”对话框顶部的“编辑”按钮，打开如图 4.6 所示的“编辑名称”对话框。在其中，可以修改名称、备注和引用位置，但不能修改作用范围。如果要修改作用范围，最好删除该名称后重新创建。



图 4.6

删除名称

在“名称管理器”中，选择要删除的名称，单击顶部的“删除”按钮。此时，Excel 会弹出如图 4.7 所示的警告消息。如果单击“确定”，那么 Excel 会永久删除该名称。



图 4.7

在删除名称后，Excel 中应用了该名称的公式不会使用相应的单元格引用进行替换，而是会显示错误值#NAME?。



5. 名称的使用

定义了名称之后，我们就可以使用这些名称了。如图 5.1 所示的工作表，我们使用行列标题定义标题所在行和列的名称，例如，名称“苹果”引用单元格区域 B2:B13，名称“一月”引用单元格区域 B2:E2。

	A	B	C	D	E	F
1	水果销量	苹果	香蕉	榴莲	菠萝	
2	一月	500	1000	830	200	
3	二月	650	800	500	210	
4	三月	700	950	680	260	
5	四月	550	900	390	300	
6	五月	390	650	320	290	
7	六月	890	880	300	260	
8	七月	760	380	360	210	
9	八月	330	980	390	320	
10	九月	210	780	460	360	
11	十月	200	660	480	390	
12	十一月	290	760	560	410	
13	十二月	300	800	600	300	
14						
15						

图 5.1

下面的公式：

=五月 苹果

返回五月份苹果的销量 390，即单元格区域 B6:E6 与单元格区域 B2:B13 的交叉单元格 B6 中的值。如图 5.2 所示。



	A	B	C	D	E	
1	水果销量	苹果	香蕉	榴莲	菠萝	
2	一月	500	1000	830	200	
3	二月	650	800	500	210	
4	三月	700	950	680	260	
5	四月	550	900	390	300	
6	五月	390	650	320	290	
7	六月	890	880	300	260	
8	七月	760	380	360	210	
9	八月	330	980	390	320	
10	九月	210	780	460	360	
11	十月	200	660	480	390	
12	十一月	290	760	560	410	
13	十二月	300	800	600	300	
14						
15						
16		390	=五月 苹果			
17						

图 5.2

下面的公式：

=SUM(水果销量)

计算表中的 4 种水果全年的总销量，公式中的“水果销量”为表中数据区域 B2:E13 的名称。

下面的公式：

=INDEX(水果销量,6,2)

返回数据区域 B2:E13 中第 6 行第 2 列的单元格中的值，即 880。

将公式定义为名称

以上图 5.1 所示的工作表为例，定义名称：**ColTotal**

引用位置为公式：**=SUM(B\$2:B13)**

如图 5.3 所示，公式使用了相对引用原理，单元格 B14 作为当前单元格，即锚单元格，也就是说，求第 14 行以上第 2 行以下的单元格数值之和。





图 5.3

效果如图 5.4 所示。在单元格 B14 中输入公式：

=ColTotal

即可求区域 B2:B13 中数值之和。同样，在单元格 C14、D14、E14 中输入相同的公式都可求所在列上第 2 行至第 13 行中单元格数值之和。

	A	B	C	D	E	F
1	水果销量	苹果	香蕉	榴莲	菠萝	
2	一月	500	1000	830	200	
3	二月	650	800	500	210	
4	三月	700	950	680	260	
5	四月	550	900	390	300	
6	五月	390	650	320	290	
7	六月	890	880	300	260	
8	七月	760	380	360	210	
9	八月	330	980	390	320	
10	九月	210	780	460	360	
11	十月	200	660	480	390	
12	十一月	290	760	560	410	
13	十二月	300	800	600	300	
14		=ColTotal				
15		ColTotal				
16						

图 5.4

本技巧的优势在于，在复杂的公式中使用时，如果需要修改公式中任逻辑，那么无需更新使用公式的多个单元格，而只需修改名称定义中的公式。



使用名称创建级联下拉列表

可以使用名称来创建级联下拉列表，也就是说，使用一个下拉列表来控制第二个下拉列表中的项，即第二个下拉列表取决于第一个下拉列表中选择的值。如图 5.5 所示，在单元格 A9 的下拉列表中选择某项后，单元格 B9 中会出现相应的下拉列表。

	A	B	C	D	E	F
1	图书	计算机	数学	历史		
2	计算机	数据科学入门	高等数学	大秦帝国		
3	数学	Python基础教程	代数	康熙大帝		
4	历史	完美Excel学习宝典	解析几何	大明王朝		
5		Excel VBA自学宝典	立体几何	春秋战国		
6			微积分			
7						
8	图书类别	图书名				
9						
10						
11						
12						
13						
14						
15						

图 5.5 (演示视频截图，详细视频见[完美 Excel 微信公众号](#))

在这里，我们定义了 4 个名称，其中名称“图书”引用单元格区域 A2:A4，其他的 3 个名称“计算机”、“数学”、“历史”都是“图书”名称引用的单元格的值，其中名称“计算机”引用单元格区域 B2:B5，名称“数学”引用单元格区域 C2:C6，名称“历史”引用单元格区域 D2:D5。

在单元格 A9 中设置数据有效性，如图 5.6 所示。

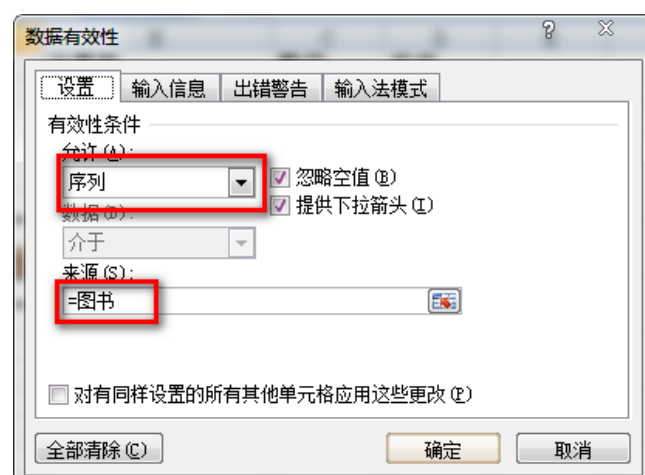


图 5.6



单元格 B9 中设置数据有效性，如图 5.7 所示。

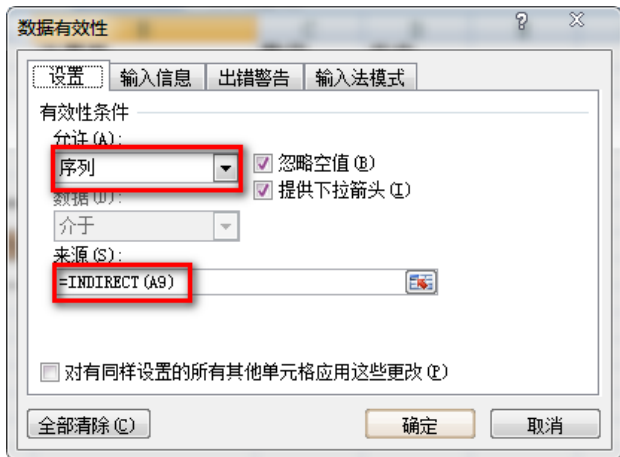


图 5.7

在条件格式中使用名称

在设置条件格式时，如果使用公式来设置条件，那么不能使用对其他工作表或工作簿的引用。此时，我们可以将对其他工作表或工作簿的引用命名为名称，然后在设置条件格式的公式中使用。如图 5.8 所示为工作表 Sheet3 中的数据，设置条件格式公式为：

=B2>Val

其中，“Val”为工作表 Sheet4 中单元格 A1 的名称。当所选择的工作表 Sheet3 中的单元格区域 B2:E13 的值大于 Sheet4 中单元格 A1 的值时，设置相应的格式。

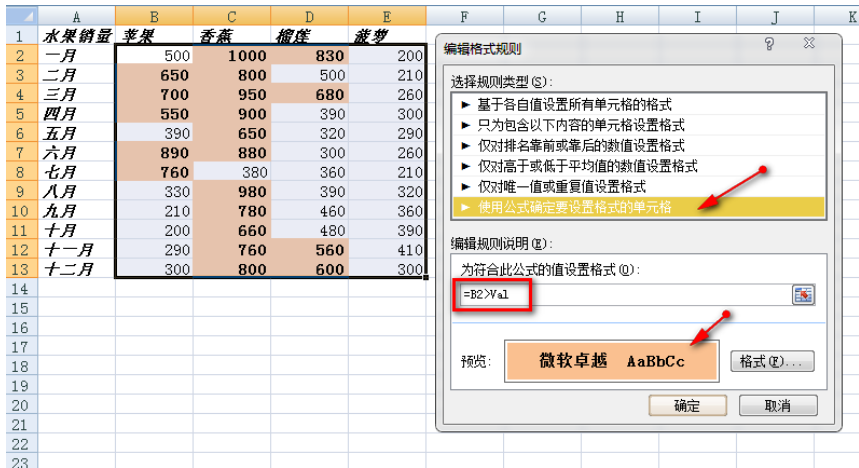


图 5.8



在图表中使用名称

在创建图表时，我们可以在 SERIES 公式中使用名称来替换单元格区域引用，此时，Excel 会自动修改 SERIES 公式中的引用来包含工作簿名称，如图 5.9 所示，在图表公式中使用了名称“月份”和“苹果”。

`=SERIES(Sheet5!B1,ExcelName5.xlsx!月份,ExcelName5.xlsx!苹果,1)`

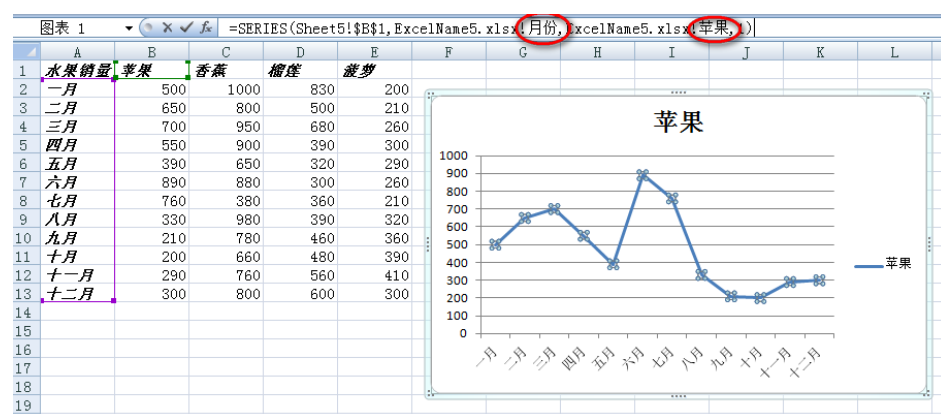


图 5.9

在图表中使用名称，可以使得名称引用的区域改变时，图表会自动更新为显示新的数据。并且，SERIES 公式中不能直接使用工作表函数，然而可以通过创建命名公式并在 SERIES 中使用来解决。



6. 动态的名称

使用命名公式，我们可以定义动态的名称。一个常用的示例是，使用 OFFSET 函数创建可以自动调整的单元格区域。如图 6.1 所示，定义名称：**苹果销售量**

引用位置：**=OFFSET(Sheet7!\$B\$2,0,0,COUNTA(Sheet7!\$A:\$A)-1,1)**



图 6.1

此时，在工作表中添加数据时，名称引用范围将自动调整。

基于名称“苹果销售量”，我们可以定义其他列。例如，定义名称“香蕉销售量”，引用位置：

=OFFSET(苹果销售量,0,1)

定义名称“榴莲销售量”，引用位置：

=OFFSET(苹果销售量,0,2)



定义名称“菠萝销售量”，引用位置：

=OFFSET(苹果销售量,0,3)

即基于“苹果销售量”区域向右依次偏移 1 列、2 列、3 列。这些名称都是动态名称，将随着列 A 中数据的增减而自动调整引用的单元格区域。

可以使用命名公式让整个表的数据区域为动态的区域，定义名称：**水果销售量**
引用位置：

=OFFSET(Sheet7!\$B\$2,0,0,COUNTA(Sheet7!\$B:\$B)-1,COUNTA(Sheet7!\$1:\$1)-1)

也可以使用 INDEX 函数来实现同样的效果，此时引用位置：

=Sheet7!\$B\$2:INDEX(Sheet7!\$B:\$E,COUNTA(Sheet7!\$B:\$B),4)

通过创建动态的名称，结合数据有效性功能，可以实现动态的下拉列表，动态的级联列表。在图表中运用，可以创建动态的图表。关于更多创建动态名称的技巧，请参见完美 Excel 微信公众号文章《**Excel 实战技巧 2：创建动态命名区域的 3 个公式**》。



本章内容 2018 年 6 月 17 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
[Excel 技术|名称 7：在图表中应用名称](#)

7. 在图表中应用名称

在创建图表时结合名称的使用，可以创建动态更新的图表或者交互式的图表。

创建动态更新的图表

如图 7.1 所示，使用 Excel 的图表功能很容易创建一个柱状图。

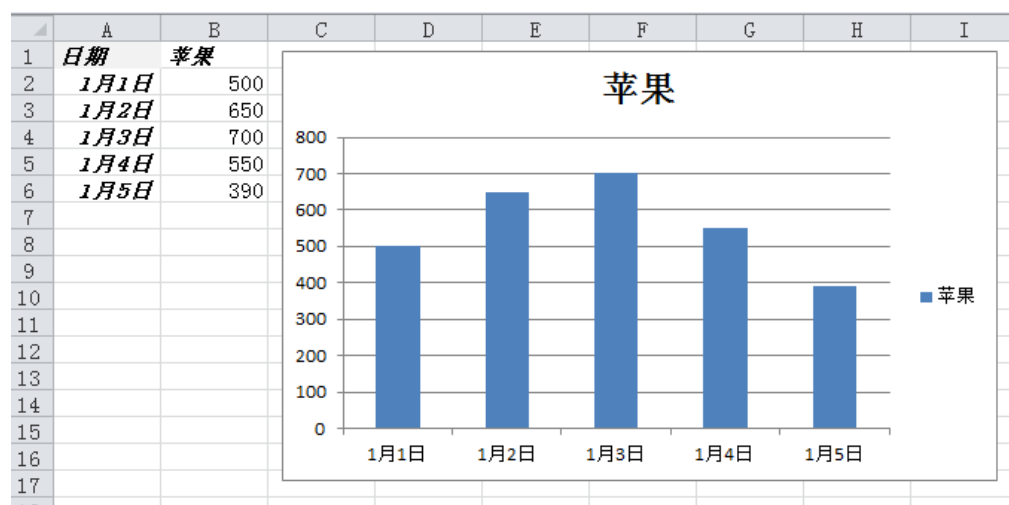


图 7.1

如果想要在添加数据时，图表能够自动更新，那么该如何实现呢？此时，在上一章中介绍的技术就可以用上了。

首先，定义 2 个名称。

名称：日期

引用位置：`=OFFSET(Sheet8!A2,0,0,COUNTA(Sheet8!$A:$A)-1,1)`



名称：苹果日销量

引用位置：=OFFSET(Sheet8!\$B\$2,0,0,COUNTA(\$B:\$B)-1,1)

然后，将系列公式：

=SERIES(Sheet8!\$B\$1,Sheet8!\$A\$2:\$A\$6,Sheet8!\$B\$2:\$B\$6,1)

修改为：

=SERIES(Sheet8!\$B\$1,Sheet8!日期,Sheet8!苹果日销量,1)

按下回车键后，Excel 会自动添加工作簿名称。

此时，在工作表中添加数据，图表会自动更新，如图 7.2 所示。

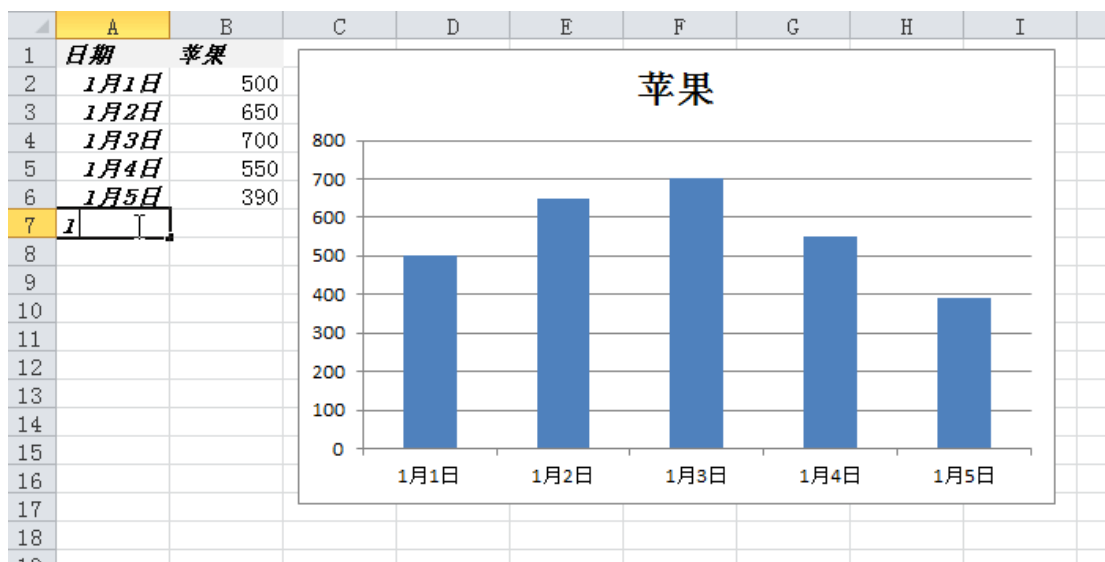


图 7.2 (演示视频截图，详细视频见完美 Excel 微信公众号)

使用滚动条控制图表

如图 7.3 所示，当单击滚动条使数值变化时，图表会自动调整绘制的数据。





图 7.3 (演示视频截图，详细视频见[完美 Excel 微信公众号](#))

首先，给单元格区域命名。

名称：**最大月数**

引用位置：**=Sheet9!\$E\$1**

名称：**月份数**

引用位置：**=OFFSET(Sheet9!\$A\$2,0,0,最大月数,1)**

名称：**点击量**

引用位置：**=OFFSET(Sheet9!\$B\$2,0,0,最大月数,1)**

然后，插入“开发工具”选项卡“表单控件”中的“滚动条”，设置格式如图 7.4 所示。

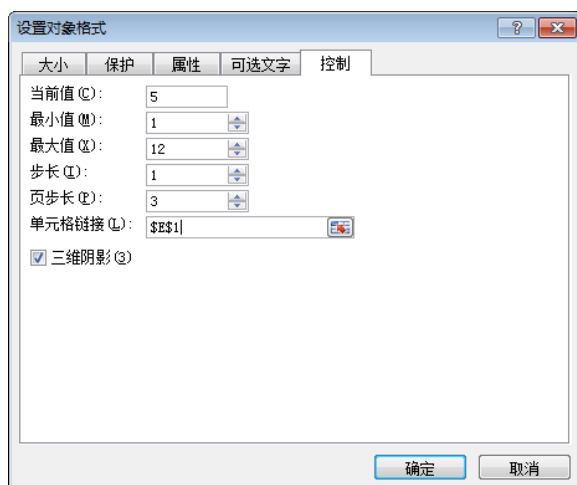


图 7.4



最后，插入图表，并将图表系列公式修改为：

`=SERIES(Sheet9!B1,ExcelName7.xlsx!月份数,ExcelName7.xlsx!点击量,1)`

使用复选框控制图表系列

如图 7.5 所示，选取某复选框，则绘制相应的系列，取消选取某复选框，其相应的系列从图表中消失。

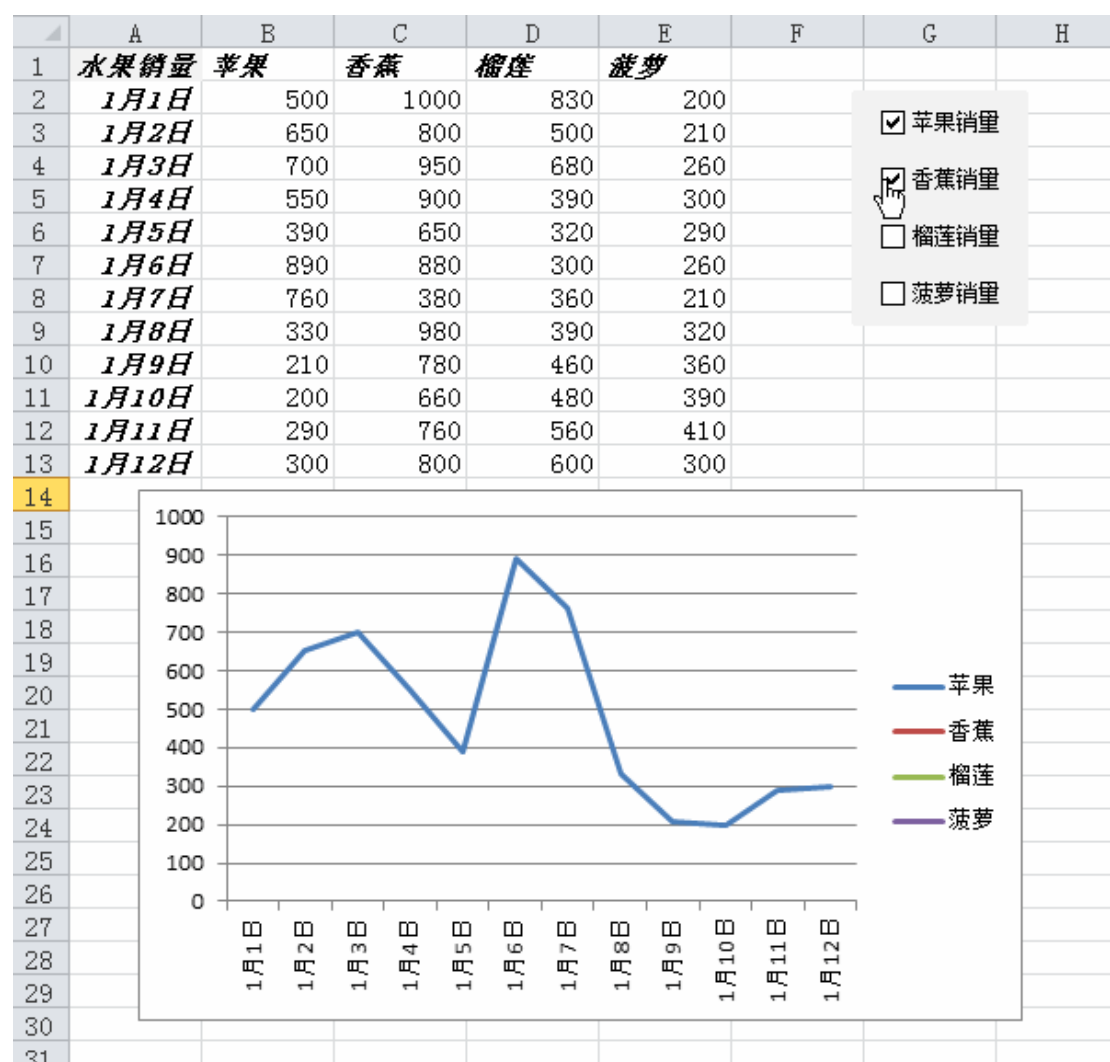


图 7.5 (演示视频截图，详细视频见完美 Excel 微信公众号)

首先，定义名称。

名称：销售日期

引用位置：=Sheet10!\$A\$2:\$A\$13



名称：苹果销量

引用位置：=Sheet10!\$B\$2:\$B\$13

名称：香蕉销量

引用位置：=Sheet10!\$C\$2:\$C\$13

名称：榴莲销量

引用位置：=Sheet10!\$D\$2:\$D\$13

名称：菠萝销量

引用位置：=Sheet10!\$E\$2:\$E\$13

名称：是否显示苹果销量

引用位置：=Sheet10!\$G\$3

名称：是否显示香蕉销量

引用位置：=Sheet10!\$G\$4

名称：是否显示榴莲销量

引用位置：=Sheet10!\$G\$5

名称：是否显示菠萝销量

引用位置：=Sheet10!\$G\$6

名称：苹果系列

引用位置：=IF(是否显示苹果销量,苹果销量,空销量)

名称：香蕉系列

引用位置：=IF(是否显示香蕉销量,香蕉销量,空销量)



名称：榴莲系列

引用位置：=IF(是否显示榴莲销量,榴莲销量,空销量)

名称：菠萝系列

引用位置：=IF(是否显示菠萝销量,菠萝销量,空销量)

接着，插入复选框控件。单击“开发工具”选项卡“插入”——“表单控件”——“复选框”，插入 4 个复选框控件并重命名，依次链接到单元格 G3、G4、G5、G6。

最后，插入图表并修改图表系列公式。例如，将苹果销量的系列公式修改为：

=SERIES(Sheet10!\$B\$1, Sheet10!销售日期,Sheet10!苹果系列,1)

其他 3 个系列进行相似的修改。

小结

巧妙地使用名称功能，能够创建出很多富有生命力的图表！



本章内容 2018 年 6 月 18 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
[Excel 技术|名称 8: VBA 中对名称的基本操作](#)

8.VBA 中对名称的基本操作

在 VBA 中, Name 对象代表定义的名称,所有的 Name 对象组成了 Names 集合。下面,详细介绍一些对名称进行基本操作的 VBA 代码。

创建名称

可以使用 Names 集合的 Add 方法来创建名称。

下面的代码在当前工作簿中创建名称 “MyName”:

```
ActiveWorkbook.Names.Add _  
    Name:="MyName", _  
    RefersTo:="=Sheet1$B$2:$D$6"
```

或者:

```
ActiveWorkbook.Names.Add _  
    Name:="MyName", _  
    RefersToR1C1:="=Sheet1R2C2:R6C4"
```

代码将当前工作簿的工作表 Sheet1 中的单元格区域 B2:D6 命名为 MyName, 该名称为全局名称,即工作簿级名称。在命名的名称中,应遵守名称的命名约定,例如不能出现空格和单元格引用。在对命名区域使用 A1 样式的引用时,如果不是使用绝对引用,则所命名的区域将会不确定。

如果要创建局部名称,即工作表级名称,则要在所命名的名称前加上工作表名,例如代码:



```
ActiveWorkbook.Names.Add _  
    Name:="Sheet1!MyWKSName", _  
    RefersToR1C1:="=Sheet1R2C2:R6C4"
```

在工作表 Sheet1 中创建名为 MyWKSName 的名称。

也可以通过引用指定的工作表来创建局部名称，例如代码：

```
Worksheets("Sheet1").Names.Add _  
    Name:="MyWKSName1", _  
    RefersTo:="=Sheet2!$A$1:$B$6"
```

在工作表 Sheet2 中创建名为 MyWKSName1 的名称。

可以简单地使用 Range 对象的 Name 属性来创建名称。例如代码：

```
Worksheets("Sheet1").Range("C2:D8").Name = "MyWBName"
```

将工作表 Sheet1 中的单元格区域 C2:D8 命名为 MyWBName。

```
Worksheets("Sheet2").Range("C5:E9").Name =  
"Sheet2!MyLocalName"
```

在工作表 Sheet2 中创建局部名称 MyLocalName。

注意，使用 Name 属性只能命名单元格区域，不能够用来命名公式、数字等。

可以在当前工作表中命名其他工作表中的区域为工作表级名称，例如代码：

```
Worksheets("Sheet1").Range("D6:F8").Name =  
"Sheet2!MyLocalName1"
```

或者：

```
Worksheets("Sheet2").Names.Add _  
    Name:="MyLocalName1", _  
    RefersTo:="=Sheet1!$D$6:$F$8"
```

在工作表 Sheet2 中命名工作表级名称，该名称引用的区域为工作表 Sheet1 中的单元格区域 D6:F8。

下面的代码命名常量。代码：

```
Names.Add Name:="MyNum", RefersTo:=100
```

将数字 100 命名为 “MyNum”。

代码：



```
Names.Add Name:="MyStr", RefersTo:="完美 Excel"
```

将字符串“完美 Excel”命名为“MyStr”。

下面的代码命名数组。代码：

```
Dim LocArray(10)
Dim i As Integer
For i = 1 To 10
    LocArray(i) = i
Next i
Names.Add Name:="MyArray", RefersTo:=LocArray
```

代码先对数组 LocArray 赋值，然后将该数组命名为 MyArray。

下面的代码对公式命名：

```
Names.Add Name:="MyFormulas", _
RefersTo:="=OFFSET(Sheet1!$A$1,0,0,COUNTA(Sheet1!$A:$A),1)"
```

代码命名了一个公式，可用来创建动态区域。

通过命名数字、字符串、数组或公式来存储经常要使用的值，可以避免被意外修改并减少对象引用。

重命名现有的名称

下面的代码将工作表 Sheet2 中的名称 MyWBName 修改为 MyName：

```
Worksheets("Sheet2").Names("MyWBName").Name = "MyName"
```

这种方法只能重命名定义的单元格区域名称，不能重命名代表公式、常量的名称。

改变所选区域的名称的引用区域

假设工作表 Sheet11 中有一个名称 MyCustomName 的单元格区域 A1:B4，选



择该区域并运行下面的代码：

```
Worksheets("Sheet11").Names.Add Selection.Name.Name,  
Sheet11.Range("B6:C9")
```

此时，名称 MyCustomName 代表的单元格区域被修改为 B6:C9。

提取/引用命名区域

下面的代码：

```
Names("MyCustomName").RefersTo
```

获取名称 MyCustomName 代表的单元格地址。

也可以使用 Evaluate 方法来提取命名区域，代码：

```
Evaluate("MyCustomName").Interior.ColorIndex = 3
```

设置名称 MyCustomName 代表的单元格区域的背景为红色。可以使用简写的代码：

```
[MyCustomName].Interior.ColorIndex = 3
```

还可以使用代码：

```
Range("MyCustomName").Interior.ColorIndex = 3
```

隐藏名称

下面的代码：

```
Names.Add Name:="HideName", _  
          RefersTo:="=Sheet1!$A$1:$F$10", _  
          Visible:=False
```

隐藏了所创建的名称“HideName”。注意，如果后面创建的名称与所隐藏的名称相同，则将覆盖掉被隐藏的名称。

代码：

```
Worksheets(1).Range("A1:B1").Name.Visible = False
```

隐藏第 1 个工作表中代表单元格区域 A1:B1 的名称。



隐藏的名称不会显示在“名称框”或者“名称管理器”中，因此对用户是不可见的，但可以像可见的名称一样使用。

删除名称

下面的代码：

```
Names("MyWBName").Delete
```

删除名称 MyWBName。

注意，在重命名已有名称或者删除名称时，要注意所要操作的名称是工作簿级名称（全局名称）还是工作表级名称（局部名称）。如果是工作表级名称（局部名称），则必须在代码中加上该名称所在工作表的引用。





本章内容 2018 年 6 月 19 日首发于
[完美 Excel]微信公众号 *excelperfect*
原标题为
Excel 技术|名称 9: VBA 操作名称的
8 个代码示例

9.VBA 操作名称的 8 个代码示例

下面列举一些名称操作的 VBA 代码示例，以更好地在 VBA 中的熟练应用名称。

代码示例 1：检查当前工作簿中是否存在某名称

下面的代码给出了两个测试名称是否存在于工作簿中的函数，过程 test 用于测试代码。

```
Sub test()  
    Dim bln As Boolean  
    bln = NameIfExists("myName")  
    If bln = True Then  
        MsgBox "这个名称在当前工作簿中已存在."  
    Else  
        MsgBox "这个名称不存在."  
    End If  
End Sub  
  
Function NameIfExists(strFindName As String) As Boolean  
    Dim rng As Range  
    Dim myName As Range  
  
    On Error Resume Next  
  
    myName = ActiveWorkbook.Names(strFindName).Name
```



```

    If Err.Number = 0 Then NameIfExists = True
End Function

Function NameExists(strName As String) As Boolean
    On Error Resume Next
    NameExists = Len(ThisWorkbook.Names(strName).Name) <> 0
End Function

```

代码示例 2：使工作簿中的所有名称都可见

```

Sub UnHideName()
    Dim nm As Name
    For Each nm In Names
        nm.Visible = True
    Next nm
End Sub

```

代码示例 3：列出当前工作簿中所有名称的相关信息

```

Sub AllNamesInfo()
    Dim i As Integer
    For i = 1 To ActiveWorkbook.Names.Count
        On Error Resume Next
        Cells(i, 1) = "" & ActiveWorkbook.Names(i).Name
        Cells(i, 2) = "" & ActiveWorkbook.Names(i).RefersToRange.Address
        Cells(i, 3) = "" & ActiveWorkbook.Names(i).ShortcutKey
        Cells(i, 4) = "" & ActiveWorkbook.Names(i).Visible
    Next i
End Sub

```



示例代码 4：列出所有隐藏的名称

在“名称框”或者“名称管理器”对话框中不会显示隐藏的名称，下面的代码在新工作表中列出所有隐藏的名称。

```
Sub ListHideNames()  
    Dim nm As Name, r As Long  
    Worksheets.Add  
    r = 1  
    For Each nm In ActiveWorkbook.Names  
        If Not nm.Visible Then  
            Cells(r, 1) = nm.Name  
            Cells(r, 2) = "'" & nm.RefersTo  
            r = r + 1  
        End If  
    Next nm  
End Sub
```

示例代码 5：显示当前单元格所命名的名称

```
Sub ShowActiveCellName()  
    On Error Resume Next  
    MsgBox ActiveCell.Name.Name  
    Select Case Err.Number  
        Case 0  
            Case 1004  
                MsgBox "单元格" & ActiveCell.Address(4) & "没有被命名."  
            Case Else  
                MsgBox Err.Number & " - " & Err.Description  
            End Select  
End Sub
```



示例代码 6：删除当前工作簿中含有“excel”字符的名称

```
Sub DeleleNameWithString()  
    Dim nm As Range  
    For Each nm In ActiveWorkbook.Names  
        If nm.Name Like "*excel*" Then  
            nm.Delete  
        End If  
    Next nm  
End Sub
```

示例代码 7：判断某单元格或单元格区域是否与命名区域部

分重叠

如果参数 rng 所代表的单元格或单元格区域与命名区域存在重叠，则返回命名区域的名称，否则返回空。

```
Function NameOfParentRange(rng As Range) As String  
    Dim nm As Name  
    For Each nm In ThisWorkbook.Names  
        If rng.Parent.Name = nm.RefersToRange.Parent.Name  
Then  
            If Not Application.Intersect(rng,  
nm.RefersToRange) Is Nothing Then  
                NameOfParentRange = nm.Name  
                Exit Function  
            End If  
        End If  
    Next nm  
    NameOfParentRange = ""  
End Function
```



示例代码 8：为名称框定义快捷键

可以使用 VBA 代码为名称框设置快捷键。在标准模块中输入下面的代码：

```
Public Declare Function SetFocus Lib "user32" (ByVal hwnd As Long) As Long

Public Declare Function FindWindow Lib "user32" Alias "FindWindowA" _
    (ByVal lpClassName As String, ByVal lpWindowName As String) As Long

Public Declare Function FindWindowEx Lib "user32" Alias "FindWindowExA" _
    (ByVal hwnd1 As Long, ByVal hwnd2 As Long, _
    ByVal lpsz1 As String, ByVal lpsz2 As String) As Long

Sub SetFocusNameBox()
    Dim res As Long
    res = SetFocus( _
        FindWindowEx( _
            FindWindowEx( _
                FindWindow("XLMAIN", Application.Caption) _
                , 0, "EXCEL;", vbNullString) _
                , 0, "combobox", vbNullString))
End Sub
```

在 Excel 中，选择“开发工具”选项卡中的“宏”，调出“宏”对话框，选择 SetFocusNameBox 过程，单击“选项”，指定快捷键例如 Ctrl+Shift+N。那么，在该工作簿中，按下 Ctrl+Shift+N 组合键，可直接定位到名称框。



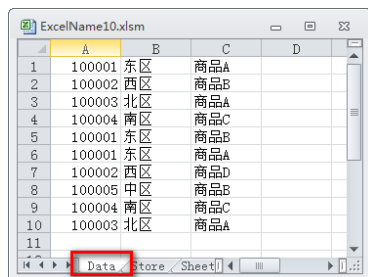


本章内容 2018 年 6 月 20 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
[Excel 技术|名称 10: 名称在 VBA 代码中的
应用示例——公式](#)

10. 名称在 VBA 代码中的应用示例——公式

本章的示例改编自 Bill Jelen 的经典 VBA 图书《[VBA and Macros Microsoft Excel](#)》系列中的内容。原示例为每日将一个包含各零售店销售数据的文件导入 Excel 工作簿，文件中包含商店的编号但没有商店的名称，使用 VBA 代码将导入的数据与工作簿中现有的商店编号和商店名称的表对比，如果存在编号，则在数据中添加该编号对应的商店名称，如果不存在编号，则在商店编号和名称表中添加编号并要求输入商店名称。

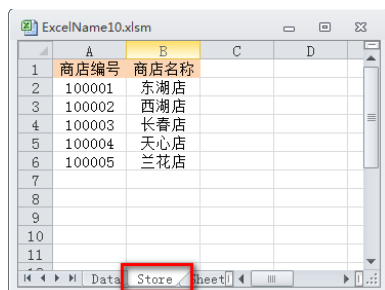
本文介绍的示例稍作简化，略去了导入外部数据的操作，假设数据已经存在于工作表中（如图 10.1 所示），或者用户经常手动添加数据，然后运用代码查找与编号对应的商店名（如图 10.2 所示）。



	A	B	C	D
1	100001	东区	商品A	
2	100002	西区	商品B	
3	100003	北区	商品A	
4	100004	南区	商品C	
5	100001	东区	商品B	
6	100001	东区	商品A	
7	100002	西区	商品D	
8	100005	中区	商品B	
9	100004	南区	商品C	
10	100003	北区	商品A	
11				

图 10.1

如图 10.2 所示，为商店编号和对应的商店名称数据，我们将单元格区域 A1:B6 命名为“StoreList”。



	A	B	C	D
1	商店编号	商店名称		
2	100001	东湖店		
3	100002	西湖店		
4	100003	长春店		
5	100004	天心店		
6	100005	三花店		
7				
8				
9				
10				
11				

图 10.2



代码如下：

```
Sub InsertData()  
    Dim wksData As Worksheet  
    Dim wksInfo As Worksheet  
    Dim wb As Workbook  
    Dim lngLastData As Long  
    Dim lngLastTemp As Long  
    Dim lngLastStore As Long  
    Dim i As Long  
    Dim strStore As String  
  
    Set wb = ThisWorkbook  
  
    '在工作表 Data 中存储着数据  
    Set wksData = ThisWorkbook.Worksheets("Data")  
  
    '在工作表 Store 中存储着编号列称表  
    Set wksInfo = ThisWorkbook.Worksheets("Store")  
  
    '将工作表 Data 变为当前工作表并在列 N 中放置列 A 的唯一值  
    wksData.Activate  
    lngLastData = Cells(Rows.Count, 1).End(xlUp).Row  
    Range("A1").Resize(lngLastData, 1).AdvancedFilter _  
        Action:=xlFilterCopy, _  
        CopyToRange:=Range("N1"), _  
        Unique:=True  
  
    '对于唯一值, 查看是否在当前的商店列表中  
    lngLastTemp = Range("N" & Rows.Count).End(xlUp).Row  
    Range("O1").Value = "不存在?"  
    Range("O2:O" & lngLastTemp).FormulaR1C1 = _  
        "=ISNA(VLOOKUP(RC[-1],StoreList,1,False))"
```



'如果没有找到编号对应的商店,那么添加新商店

```
lngLastStore = wksInfo.Range("A" &  
Rows.Count).End(xlUp).Row + 1  
For i = 2 To lngLastTemp  
    If Cells(i, 15).Value = True Then  
        strStore = Cells(i, 14).Value  
        wksInfo.Cells(lngLastStore, 1).Value = strStore  
        wksInfo.Cells(lngLastStore, 2).Value = _  
            InputBox(Prompt:="请为编号为" & strStore & _  
                "的商店输入名称", Title:="找到新商店")  
        lngLastStore = lngLastStore + 1  
    End If  
Next i
```

'删除临时存放商店编号的区域

```
Range("N1:O" & lngLastTemp).Clear
```

'重新定义名称 StoreList 以适应新添加的商店

```
lngLastStore = wksInfo.Range("A" &  
Rows.Count).End(xlUp).Row  
wksInfo.Range("A1:B" & lngLastStore).Name = "StoreList"
```

'使用 VLOOKUP 函数查找并在工作表 Data 的列 B 添加商店名

```
wksData.Range("B1").EntireColumn.Insert  
Range("B1:B" & lngLastData).FormulaR1C1 = _  
    "=VLOOKUP(RC1,StoreList,2,False)"
```

'将公式转换成值

```
Range("B1:B" & lngLastData).Value = Range("B1:B" &  
lngLastData).Value
```

'释放内存空间

```
Set wksData = Nothing
```



```

Set wksInfo = Nothing

Set wb = Nothing

End Sub

```

运行代码后，将在工作表 Data 中的商店编号列右侧插入一列并输入对应的商店名称，如图 10.3 所示。

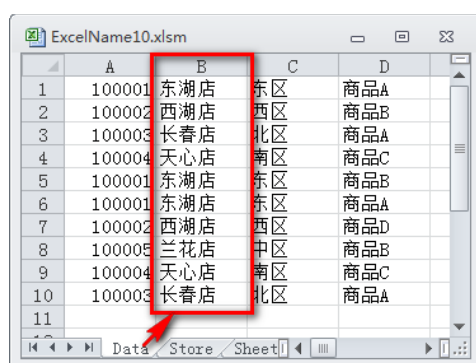


图 10.3

如果用户在工作表 Data 中输入了新数据，但新的编号不在已有的商店编号列表中，那么 Excel 会弹出输入框，要求用户为新编号输入商店名，如图 10.4 所示。

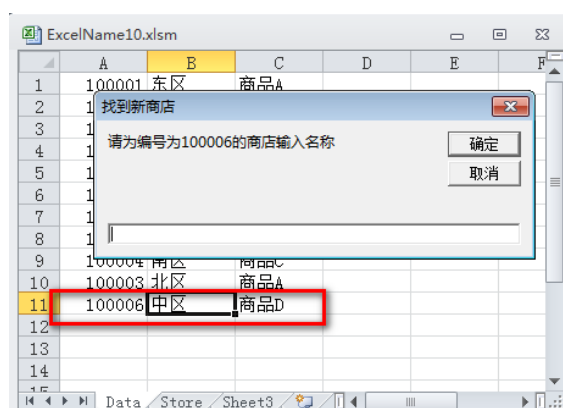


图 10.4

输入完后，单击“确定”，会自动更新商店列表且在工作表 Data 中添加商店列。



本章内容 2018 年 6 月 21 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
[Excel 技术|名称 11: 名称在 VBA 代码中的
应用示例——数据有效性](#)

11. 名称在 VBA 代码中的应用示例——数据有效性

本章示例来自于 Excel VBA 帮助，学习使用 VBA 代码如何将名称应用于数据有效性。

示例中，工作表 Sheet2 的单元格区域 A2:A100 中提供了所需要的数据有效性数据（如图 11.1 所示），用于验证在工作表 Sheet1 的单元格区域 D2:D10 中输入的数据。

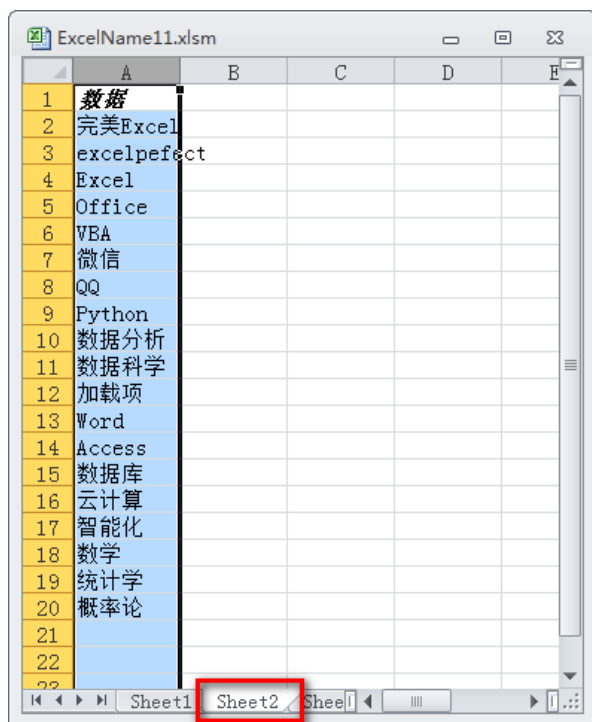


图 11.1

代码如下：



```

Sub Add_Data_Validation_From_Other_Worksheet()
    '声明当前工作簿和工作表及单元格区域
    '提供验证数据的单元格区域
    '输入数据的单元格区域

    Dim wb As Workbook
    Dim wksTarget As Worksheet
    Dim wksSource As Worksheet
    Dim rngTarget As Range
    Dim rngSource As Range

    '初始化 Excel 对象
    '删除上次运行宏造成的改变
    Set wb = ThisWorkbook
    With wb
        Set wksSource = .Worksheets("Sheet2")
        Set wksTarget = .Worksheets("Sheet1")
        On Error Resume Next
        .Names("Source").Delete
        On Error GoTo 0
    End With

    '命名指定的单元格区域
    '该区域提供验证的数据
    With wksSource
        .Range(.Range("A2"), .Range("A100").End(xlUp)).Name = "Source"
    End With

    '使用数据有效性的目标区域
    Set rngTarget = wksTarget.Range("D2:D10")

    '清除上次运行宏后的结果

```



```

' 设置目标区域使用数据有效性验证
With rngTarget
    ' 清除目标区域的内容
    .ClearContents

    ' 在目标区域添加数据有效性
    With .Validation
        .Delete
        .Add Type:=xlValidateList, _
            AlertStyle:=xlValidAlertStop, _
            Formula1:="=Source"

        ' 使用合适的标题和消息设置错误对话框
        .ErrorTitle = "值错误"
        .ErrorMessage = "你仅能从列表中选择值"
    End With
End With
End Sub

```

运行代码后，在工作表 Sheet1 的单元格区域 D2:D10 中添加了数据有效性下拉列表，如图 11.2 所示。

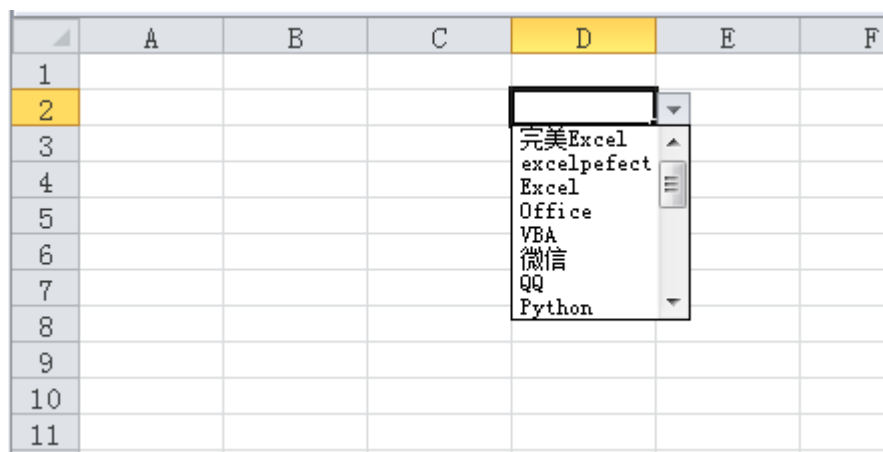


图 11.2

如果用户在工作表 Sheet1 的单元格区域 D2:D10 中输入的数据不在下拉列表中，即工作表 Sheet2 的单元格区域 A2:A100 中的数据，那么 Excel 会弹出



提示信息，如图 11.3 所示。

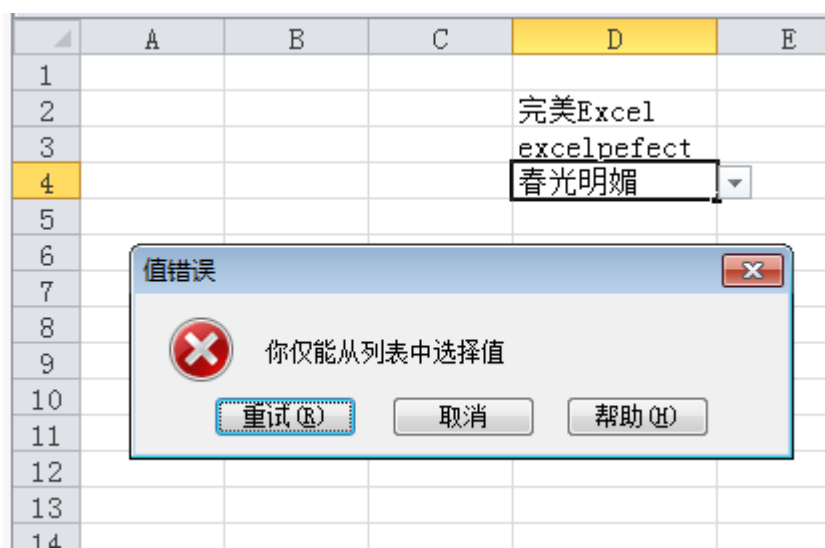


图 11.3



本章内容 2018 年 6 月 22 日首发于
[完美 Excel]微信公众号 *excelperfect*
原标题为
**Excel 技术 | 名称 12: 名称在 VBA 代码中的
应用示例——筛选数据**

12. 名称在 VBA 代码中的应用示例—— 筛选数据

在完美 Excel 微信公众号文章《[经典代码解析 5: 查找并显示部分匹配的数据](#)》一文中，充分展示了如何使用名称来高效筛选数据。下面，我将该文介绍的示例技巧再次粘贴于此，作为讲解名称的内容之一，并于文后添加了示例工作簿下载信息。

目的

在大量的数据中，快速查找与查找值部分匹配的所有数据。

示例工作簿与程序代码

如下图 12.1 和图 12.2 所示，工作簿包含两个工作表，一个工作表 Data 用作数据库，包含大量的数据；另一个工作表 Search 用来从数据库工作表 Data 中查找想要的数据并显示。



项目	描述	零售价	批发价
B900000004	B900000004 756 SUBST LCD	448.58	0.00
0308670106	0308670106 A200 DIAL UNIT ENCODER AND BOARD *98410157	34.95	24.50
0319232902	0319232902 F4 ON/OFF/VOL POT WITH VR BOARD	24.06	20.08
0324290105	0324290105 DSP BOARD #02 M802	243.98	165.89
0324940201	0324940201 EX2500 #02 ENC 746PRO 0880000950*	55.91	56.88
0325911202	0325911202 EX2500 #12 SENSOR 703 0880001360*	61.10	33.18
0800009560	0800009560 SDCFJ-128-388 7800 CF CARD SDCFJ-64 *0800007460	257.03	218.48
0900005413	0900005413 2178-3C BS449A 756PRO3 BOARD FOR 98415074	63.24	46.77
0900005825	0900005825 B-5603C (#2458-1E) 0910053673*	46.41	25.06
0900005904	0900005904 B5690C-5696 746PRO 0910054561* MIC BOARD ASSMBLY	84.17	84.39
0900006384	0900006384 #2681-1D B6033D F50 B6034B B6035C 2 EACH F60	31.40	15.51
0910000630	0910000630 CABLE RIBBON B789	19.08	9.12
0910011691	0910011691 CABLE RIBBON B1031A 275 735	7.04	5.49
0910012020	0910012020 B1045 CAB RIBBON \$ is correct, special prod. ru	99.38	58.94
0910012061	0910012061 CABLE RIBBON B1046A	6.84	5.95
0910012080	0910012080 CABLE RIBBON B1044 H16 U16	7.20	4.94
0910012740	0910012740 PCB SHIELD B1178 A2	12.76	8.10
0910045921	0910045921 FLEX R4598 706 706MK2G	9.27	9.08

图 12.1

在数据库中查找部分匹配的数据

查找的数据:

找到的记录数: 0

描述	零售价	批发价
----	-----	-----

图 12.2

方法 1：使用 Range 对象的 AdvancedFilter 方法

在工作表中定义名称

在 Search 工作表中定义的名称如下：

C3 单元格名称：search_string

C8 单元格名称：result_target

在 Data 工作表中将单元格区域 z1:z2 定义的名称为：filter_criteria

在“定义名称”对话框中直接定义的名称（隐藏的名称）：



```
Database=Data!$B$1:INDEX(Data!$1:$1048576,COUNTA(Data!$A:$A),COUNTA(Data!$1:$1))
```

```
result=Search!$C$8:INDEX(Search!$C$8:$XFD$1048576,COUNTA(Search!$C$8:$C$1048576),COUNTA(Search!$C$8:$XFD$1048576))
```

在 Search 工作表单元格 C5 中，使用下面的公式统计找到的记录数：

```
=SUMPRODUCT( -- (C9:C1048576>"") )
```

设置工作表 Search

设置 Search 工作表单元格 C3 的格式，在“单元格格式”的“保护”选项卡中取消“锁定”复选，以便于在“保护工作表”时仍能编辑该单元格。

保护工作表 Search。

编写代码

在 Search 工作表对象模块中的代码为：

```
Private Sub Worksheet_Activate()  
    [search_string] = "在这里输入你要查找的内容."  
    [search_string].Select  
End Sub  
  
Private Sub Worksheet_Change(ByVal Target As Range)  
    Const CRITERIA_HEADER = "Description"  
    Const TEMPORARY_NAME_CREATED_BY_ADVANCED_FILTER = "Extract"  
  
    ' 如果改变的不是要求输入查找条件的单元格,则退出  
    If Intersect(Target, [search_string]) Is Nothing Then Exit Sub  
    [search_string].Select  
  
    '撤消工作表保护  
    Unprotect
```



```

With Application
    ' 清除上一次搜索结果
    .Calculation = xlCalculationManual
    [result].ClearContents

    ' 进行搜索
    [filter_criteria] = .Transpose(Array(CRITERIA_HEADER, "*" & [search_string]
& "*"))
    [database].AdvancedFilter xlFilterCopy, [filter_criteria], [result_target]

    ' 清理
    [filter_criteria] = vbNullString
    Names(TEMPORARY_NAME_CREATED_BY_ADVANCED_FILTER).Delete
    .Calculation = xlCalculationAutomatic
End With

'保护工作表
Protect

End Sub

```

说明：

- 代码中的方括号直接获取名称定义的单元格或单元格区域。
- 先撤销对工作表的保护，然后才能对工作表操作。
- 代码中设置了一个临时的条件区域，用于高级筛选，使用完后删除。使用 Transpose 函数将水平数组转换为垂直数组，即在同一列的两行中放置数据。
- AdvancedFilter 方法的使用详见完美 Excel 微信号文章《[Excel VBA 解读（53）：高级筛选—AdvancedFilter 方法](#)》。
- 设置计算模式，让工作表中统计记录数的 SUMPRODUCT 公式自动计算以实时获取最新得到的记录数。



方法 2：使用 ADO

在工作表中定义名称

除以下变化外，工作表中名称与方法 1 相同。

C9 单元格名称：**result_target**

在“定义名称”对话框中直接定义的名称（隐藏的名称）：

```
Database=Data!$A$2:INDEX(Data!$2:$1048576,COUNTA(Data!$A:$A),COUNTA(Data!$1:$1))
```

```
result=Search!$C$9:INDEX(Search!$C$9:$XFD$1048576,COUNTA(Search!$C$9:$C$1048576),COUNTA(Search!$C$8:$XFD$1048576))
```

在 Search 工作表单元格 C3 中，使用下面的公式统计找到的记录数：

```
=SUMPRODUCT( (C9:C65536>"") * 1 )
```

设置工作表 Search

除以下变化外，工作表设置与方法 1 相同。

与方法 1 相比，本方法增加了一个复选框，其目的是为了避免内存泄漏而导致计算机崩溃。其工作表界面如下图 12.3 所示。



图 12.3



编写代码

在 Search 工作表对象模块中的代码为:

```
Private Sub Worksheet_Activate()  
    [search_string] = "在这里输入你要查找的内容."  
    [search_string].Select  
End Sub  
  
Private Sub Worksheet_Change(ByVal Target As Range)  
    Dim sExcelFileName As String  
    Dim sDatabaseRangeAddress As String  
  
    Const EXTERNAL_DATA_FILE_SUFFIX = "_external_copy.xlsx"  
    Const CHECKBOX_TRUE = 1  
  
    ' 如果改变的不是要求输入查找条件的单元格,则退出  
    If Intersect(Target, [search_string]) Is Nothing Then Exit Sub  
    [search_string].Select  
  
    ' 清除以前搜索的结果  
    Unprotect  
    Application.Calculation = xlCalculationManual  
    [result].ClearContents  
  
    .....  
  
    ' 当使用 ADO 处理 Excel 数据时,有一个文档 bug  
    ' 导致内存泄漏,除非数据不在与 ADO 工作簿相同的工作簿中.  
    ' http://support.microsoft.com/kb/319998  
    '  
    ' 这段小代码演示如何复制正调用的工作簿到新工作簿阻止泄漏.  
    '
```



```

' 如果选择"外部数据方法"
sExcelFileName = Parent.FullName
If CheckBoxes(1).Value = CHECKBOX_TRUE Then
    sExcelFileName = sExcelFileName & EXTERNAL_DATA_FILE_SUFFIX
    Parent.SaveCopyAs sExcelFileName
End If
.....

' 使用包含搜索字符串的单元格区域填充结果
sDatabaseRangeAddress = [database].Parent.Name & "$" &
Replace([database].Address, "$", vbNullString)
SqlFind UCase([search_string]), [result_target], sExcelFileName,
sDatabaseRangeAddress

' 清除
If CheckBoxes(1).Value = CHECKBOX_TRUE Then
    Kill Parent.FullName & EXTERNAL_DATA_FILE_SUFFIX
End If
Application.Calculation = xlCalculationAutomatic
Protect
End Sub

Private Sub SqlFind(sSearch As String, rDest As Range, sExcelFileName As String,
sDatabaseRangeAddress As String)
    Dim sSQLWhereClause As String
    Dim oConn As New ADODB.Connection
    Dim oRS As New ADODB.Recordset

' 打开 ADO 连接到当前 Excel 工作簿
oConn.Open "Provider=Microsoft.ACE.OLEDB.12.0;" & "Data Source=" &
sExcelFileName & ";" & _
    "Extended Properties="""Excel 12.0;HDR=No;""""

```



' 准备 SQL 语句的 WHERE 子句

```
sSQLWhereClause = " Where UCase([F2]) Like '%" & sSearch & "%"
```

' 打开记录集作为执行 SQL 查询的结果

```
oRS.Open "Select [F2], [F3], [F4] From [" & sDatabaseRangeAddress & "]" &  
sSQLWhereClause, oConn, adOpenStatic
```

' 复制整个结果表到目标区域

```
rDest.CopyFromRecordset oRS
```

' 清除

```
oRS.Close
```

```
oConn.Close
```

```
Set oRS = Nothing
```

```
Set oConn = Nothing
```

```
End Sub
```

说明：

- 将 Excel 工作簿作为数据库，使用 ADO 结合数据库查询语句获取结果。
- 根据是否选取复选框，来确定是否创建工作簿副本，即是否在工作簿副本中查询数据。
- 基于 Excel 单元格区域地址进行查询时，使用的是 “[工作表名+\$+单元格区域地址]” 的格式。因此，语句 [database].Parent.Name & "\$" & Replace([database].Address, "\$", vbNullString)将单元格地址转换成正确的查询格式。
- 在 SQL 查询语句中，使用[F2]、[F3]、[F4]来表示第 2 列、第 3 列、第 4 列。
- 有关 ADO 的详细介绍参见完美 Excel 微信公众号文章《[经典代码解析 4：使用 ADO 整合 Excel 与 Access](#)》。



小结

- 在工作表中设置公式，通过公式来计算记录数。
- 定义名称并在代码中使用，充分利用名称简洁且自动更新的优势，简化代码。
- 方法 1 在需要时才设置临时筛选条件，使用完后删除，更灵活。
- 方法 2 将工作表作为数据库，使用 ADO 进行查询操作。
- 本示例体现了公式、名称与 VBA 的完美配合，以简短的代码获取所需的功能。我认为是程序优化的上佳之作。

示例工作簿下载

在完美 Excel 微信公众号底部发送消息：

筛选数据

下载示例工作簿。





本章内容 2018 年 6 月 23 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
[Excel 技术|名称 13: 名称在 VBA 代码中的
应用示例——自动绘制图表](#)

13. 名称在 VBA 代码中的应用示例——自动绘制图表

本章提供一个自动绘制图表的实例，由 [Jan Karel Pieterse](#) 开发，有兴趣的朋友可以根据文后提供的信息下载研究。

这个实例使用了动态命名区域来创建图表数据，并结合表单控件和 VBA 代码来调整图表的绘制比例和图表元素。你可以在工作表 Datasheet 中粘贴要绘制的数据，在工作表 Chartsheet 中会根据这些数据自动绘制图表。还可以调整工作表中的数值调节按钮来更改图表标题和轴标题，可以选取相应的复选框来调整绘图比例。

下图 13.1 简单地演示了本实例的效果。

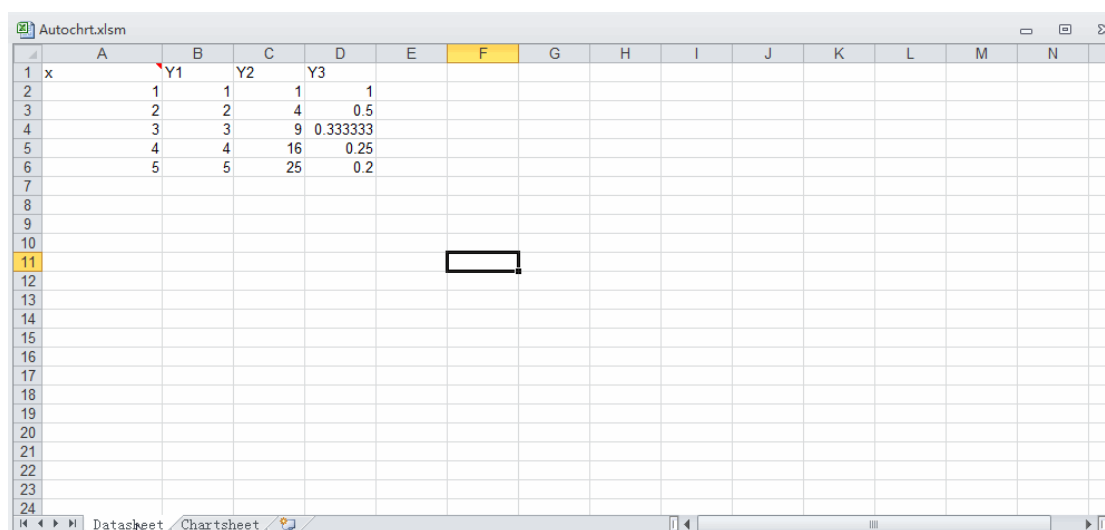


图 13.1 (演示视频截图，详细视频见[完美 Excel](#) 微信公众号)



在实例工作簿中给出了绘制该图表的简单说明，可以在[完美 Excel](#) 微信公众号底部发送消息：

自动绘制图表

下载本工作簿研究。



本章内容 2018 年 6 月 24 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
[Excel 技术|名称 14：名称在 VBA 代码中的
应用示例——绘制动画图表](#)

14.名称在 VBA 代码中的应用示例——绘制 动画图表

本文提供的动画图表示例，根据 [DanielFerry](#) 开发的作品改编，充分体现了名称的妙用，有兴趣的朋友可以根据文后提供的信息下载研究。

在图表中，一个小圆点围绕着正方形作逆时针运动，如图 14.1 所示。

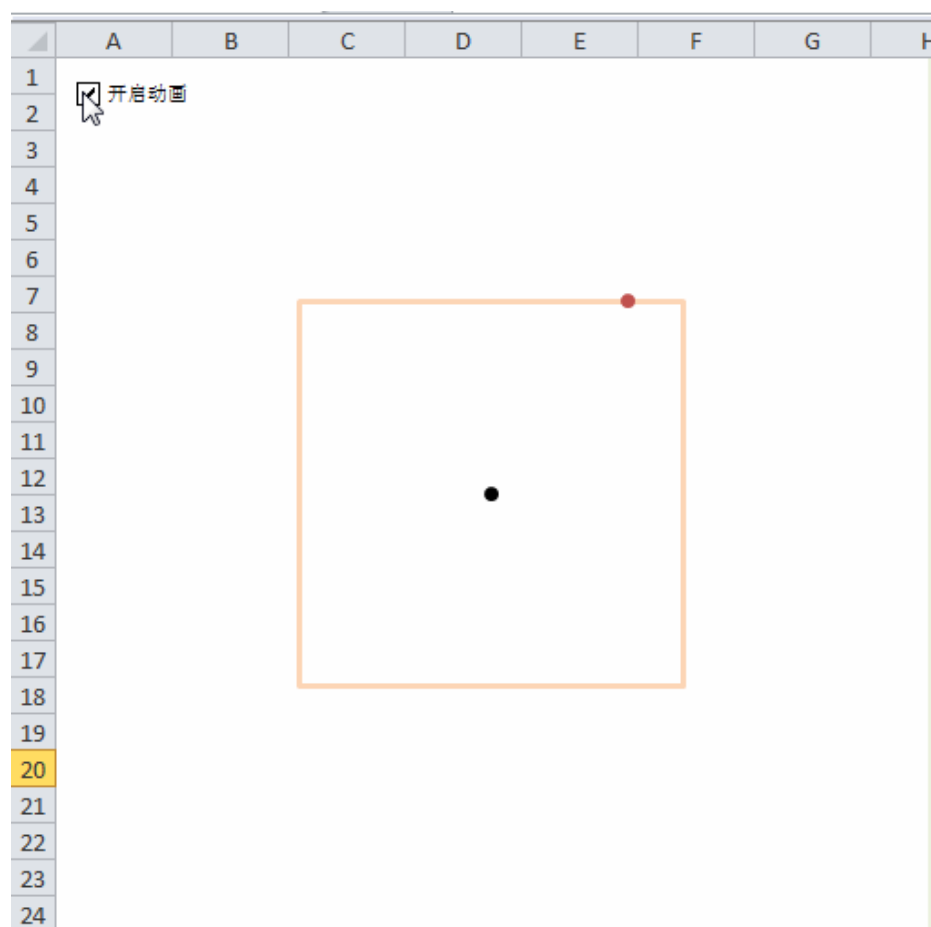


图 14.1 (演示视频截图，详细视频见[完美 Excel](#) 微信公众号)



下图 14.2 是在工作簿中定义的名称。

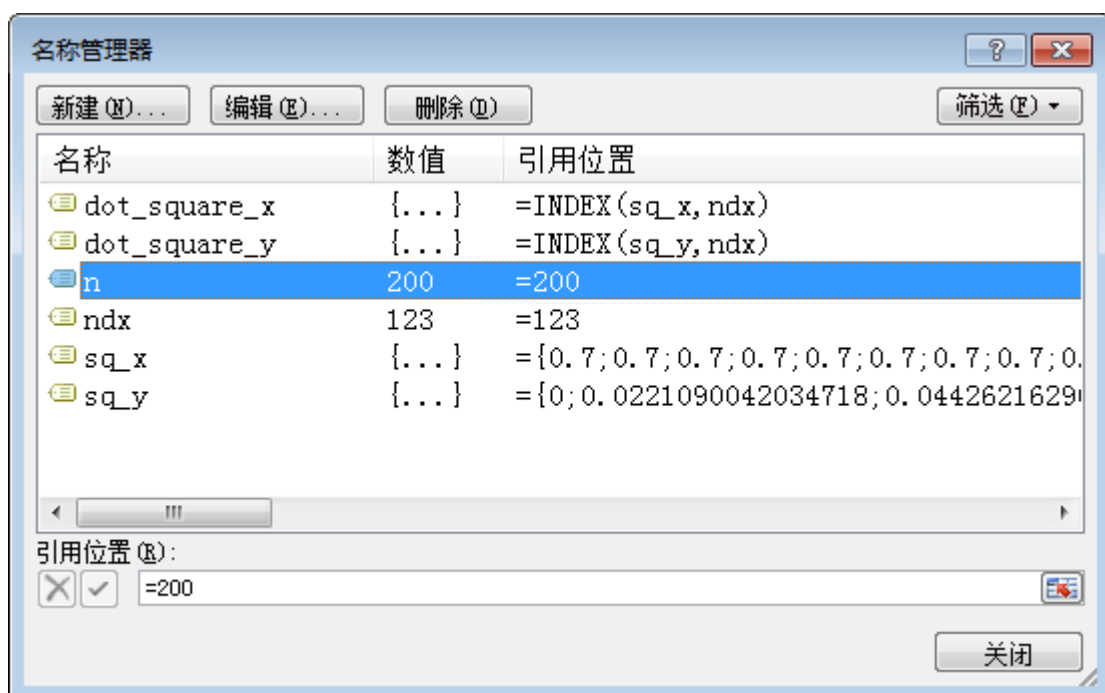


图 14.2

其中，sq_x 和 sq_y 用于绘制图表中固定的正方形；dot_square_x 和 dot_square_y 用于绘制运动着的点，该点的坐标通过名称 n 和 ndx 来驱动并在 sq_x 和 sq_y 中取值，而名称 n 的值为常量，ndx 由 VBA 代码来改变，从而使点的坐标不停变化，因而点在图表中不停运动。

代码如下：

```
Private i As Long
Private bRunning As Boolean

Public Sub Animate()
    Dim NM As Names
    Dim z As Long

    Const DELAY_FACTOR = 125

    If bRunning Then
```



```

        bRunning = False
        Exit Sub
    End If
    bRunning = True

    Application.Cursor = xlNorthwestArrow

    Set NM = Application.Names
    Do
        If bRunning = False Then Exit Do
        i = i + 1
        If i > 800 Then i = 1
        NM.Add "ndx", (i Mod [n]) + 1
        For z = 1 To DELAY_FACTOR: DoEvents: Next z
    Loop

    Application.Cursor = xlDefault
    bRunning = False

    Set NM = Nothing
End Sub

```

图表完全利用名称来提供数据，且结合公式来取值从而实现数据的不断变化，因而图表也不断变化，达到动画的效果。有兴趣的朋友可以在[完美 Excel](#) 微信公众号底部发送消息：

方形动画图表

下载本工作簿研究。





关于完美 Excel

完美 Excel 是一个坚持分享 Excel 与 VBA 技术知识的微信公众号，自 2017 年 5 月 15 日开始，每天推送一篇 Excel 与 VBA 技术和应用方面的文章。目前，共有 670 余篇实用文章可供广大 Excel 爱好者和使用者学习交流。这本电子书就是根据完美 Excel 上发表的名称应用技术的系列文章整理而成的。

每天清晨，完美 Excel 微信公众号：**excelperfect** 都会推送一篇关于 Excel 与 VBA 的相关文章。如果你有兴趣学习 Excel 和 VBA 的相关知识和实战技巧，可以关注完美 Excel 微信公众号，绝对不会让你失望！

可以通过下列方法关注[完美 Excel]微信公众号：

方法 1—在通讯录中搜索“完美 Excel”或者“**excelperfect**”后点击关注。

方法 2—扫一扫下面的二维码



完美 Excel 微信公众号使用指南

下图 1 是完美 Excel 微信公众号的界面。公众号名称显示在屏幕正上方，屏幕底部显示有“菜单栏”，目前设置的菜单为“技术精粹”、“VBA 精选”、“联系 me”。在底部左侧的小键盘图标为消息框入口，单击可进入消息框界面给完美 Excel 公众号发送消息。



图 1

下图 2、图 3、图 4 分别为底部 3 个菜单的子菜单。目前，菜单“技术精粹”中设置有“2018 年文章合集”、“480 篇文章合集”、“又一波 20 个函数”、“快速学会 30 个函数”、“玩转数据验证”等 5 个子菜单；菜单“VBA 精选”中设置有“Excel VBA 编程基础”、“最最基础入门篇”、“VBA 学习经验”等 3 个子菜单；菜单“联系 me”中设置有“投稿须知”、“网站集粹”、“个人声明”、“坚持的美好”、“2019 年目标”等 5 个子菜单。





图 2



图 3





图 4

单击这些子菜单会进入详细的文章页面或者文章整理的入口页面，方便读者浏览或查阅本公众号的文章。同时，这些子菜单会随着完美 Excel 微信公众号内容的增加而适时调整。

可以单击底部左侧的小键盘图标，进入发送消息界面，如图 5 所示。在文本框中输入想要发送的文字，单击底部的“发送”按钮，就可以将消息发送给完美 Excel 微信公众号。

大家应留意完美 Excel 微信公众号推送的文章中的一些信息，例如，我会在百度网盘中存放一些文档资料或者示例工作簿文件，并在文章中给出进入百度网盘下载的文本信息，你只需在发送消息框中输入我给出的文本，单击发送后，就会收到一条关于下载链接和密码的信息。单击链接并按提示输入密码后，即可获得



相关的文档资料或示例工作簿文件了。



图 5

例如，在图 5 所示的界面中输入“Excel 动画图 2”后，会自动收到图 6 所示的信息，根据信息即可获取这个 Excel 动画图表文件。



图 6



希望大家在完美 Excel 微信公众号中能够学习到所需要的知识，获取到所需要的 Excel 应用技巧，提高自己的水平。但愿在今后的日子里，完美 Excel 微信公众号能够真正帮助大家发挥 Excel 的威力，为大家解决问题，提高工作效率。

