



学习 Excel VBA 数据结构 • 字典详解

完美 Excel 出品
微信公众号: [excelperfect](#)



内容提要

Dictionary 对象是 VBA 中一个很强大的特性，然而似乎没有完整的帮助文档对其进行详解。

本电子书《学习 Excel VBA 数据结构：字典详解》整理了 <http://www.snb-vba.eu> 和 <http://www.vbaexpress.com>，以及《Paul Lomax: VB & VBA in a Nutshell: The Language》中的相关资料并进行了优化和补充，作为在 VBA 编程过程中使用 Dictionary 对象的参考。



目录

什么是字典 (Dictionary)	1
如何创建字典 (Dictionary) 对象?	2
早期绑定.....	2
后期绑定.....	3
示例 1：文本字符串键	4
示例 2：数字键	5
示例 3：日期键	5
示例 4：对象键	5
示例 5：工作表中的 ActiveX 控件	6
示例 6：一个 Dictionary 对象中的几种键	6
示例 7：元素内容为空	8
示例 8：空文本字符串	8
示例 9：文本字符串	9
示例 10：不可打印字符	9
示例 11：数字	9
示例 12：日期	9
示例 13：一维数组	10
示例 14：多维数组	10



示例 15 : 对象.....	10
示例 16 : 用户窗体控件.....	11
示例 17 : ActiveX 控件.....	11
示例 18 : 工作簿中的所有工作表.....	11
字典 (Dictionary) 对象的属性和方法.....	13
Add 方法.....	13
CompareMode 属性.....	14
Count 属性.....	14
Exists 方法.....	14
Item 属性.....	15
Items 方法.....	16
Key 属性.....	16
Keys 方法.....	17
Remove 方法.....	17
RemoveAll 方法.....	18
字典 (Dictionary) 对象的常用操作.....	19
自动创建唯一键.....	19
获取唯一元素列表.....	19
检查键/项目是否在字典中存在.....	20
从 Dictionary 中获取元素.....	21
获取指定键的元素内容.....	24
获取包含指定数据的键的元素内容.....	25
操作字典中的键.....	26
Items 方法在字典中的运用.....	29



复制 Dictionary 对象中的元素.....	31
修改 Dictionary 对象中元素的内容.....	32
从 Dictionary 对象中的删除元素.....	37
示例.....	38
示例 1：基于键删除重复的行	38
示例 2：判断一列数据是否存在重复	38
示例 3：筛选列 B 中包含最高值的记录.....	39
示例 4：筛选列 B 中包含最小值的记录.....	40
示例 5：合并记录	40
示例 6：合并工作表	41
示例 7：合并 CSV 文件	42
示例 8：创建唯一元素列表	42
示例 9：在工作表的两列中获取唯一值并填充组合框	43
示例 10：检查是否所有的 ActiveX 控件都有数据.....	46
示例 11：检查工作表是否存在.....	46
示例 12：转换文本字符串为二维数组.....	47
关于完美 Excel	49
完美 Excel 微信公众号使用指南	50



温馨提示：

在完美 Excel 微信公众号中发送消息：字典对象，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。



什么是字典 (Dictionary)

在 VBA 中，字典是一个集合对象，可以存储数字、文本、日期、数组、单元格区域、变量和对象等。这些数据临时存储在内存中，因此可以非常容易且快速地访问。

字典中每个项目都有自己唯一的键，可以使用该键直接访问字典元素，包括读取、写入或修改。这是 Dictionary 的一个重要特征，因此无需使用循环或使用 Excel 函数 Match 来查找项目，这是字典相对于数组和矩阵变量的优势。

Dictionary 有一些属性和方法比使用集合有优势，例如 .keys、.items、.removeall 等。因此，在某些情况下使用 Dictionary 会更好。例如，由于 Dictionary 仅能包含唯一的键，因此可以使用属性 keys 来创建唯一值列表。



如何创建字典（Dictionary）对象？

Dictionary 对象不是标准 VBA 库的一部分，它是 Microsoft Scripting Runtime 库的元素，在文件..\Windows\system32\scrrun.DLL 或类似的目录下。因此，必须先连接到该库，才能够使用其方法和属性。实现这种连接的技术称为“绑定”，可以早期绑定也可以后期绑定。

早期绑定

早期绑定是在编写和使用代码之前建立对该库的连接。在 VBE 中，单击“工具——引用”，在“引用”对话框中设置对 Microsoft Scripting Runtime 库的引用。

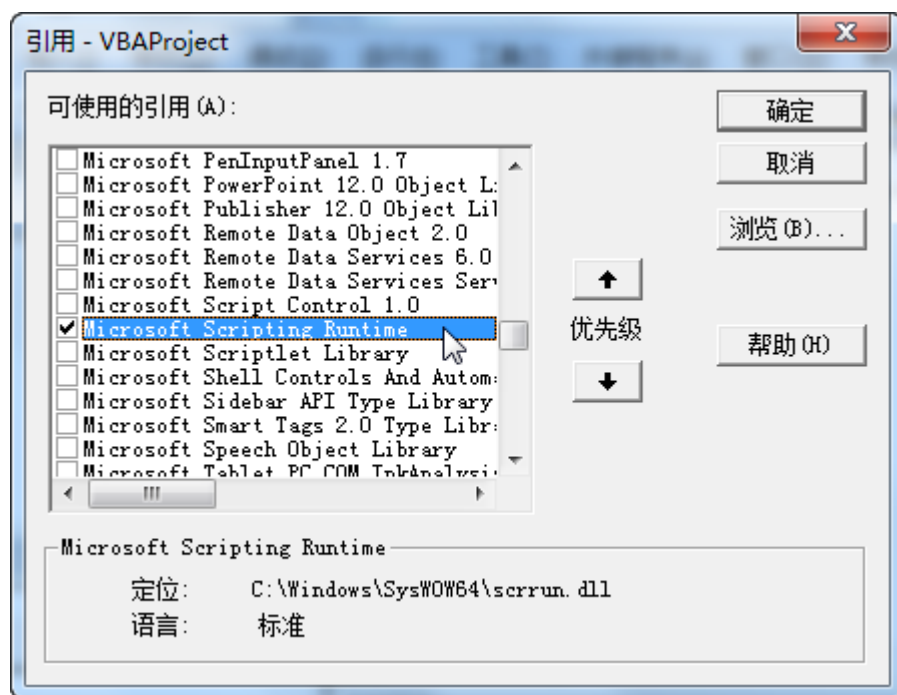


图 1

也可以使用宏来实现：

```
Sub bindingScripting()  
    ThisWorkbook.VBProject.References.AddFromFile  
    "C:\windows\system32\scrrun.dll"
```



End Sub

连接将被存储在 Excel 文件中。如果其他系统包含 Office，那么分发该工作簿时，文件会自动连接到该库。

在使用之前，要创建基于该库的新的对象（实例）。

后期绑定

后期绑定是仅在代码需要时连接。可以使用 `CreateObject` 方法来创建 Dictionary 对象：

```
CreateObject("scripting.dictionary")
```

代码将作出对 Microsoft Scripting Runtime 库的连接，并基于该库创建新的对象（实例）。

两种绑定技术是等同的，下面介绍的内容主要使用后期绑定。

仅能逐一使用键和元素内容来填充 Dictionary 对象。正如本文开头所讲的，元素内容可以是下列任何数据：数字、文本字符串、日期、数组、单元格区域、变量、集合、字典、空字符串、对象、`nothing`。键可以由数字、文本字符串、日期或对象，或者包含数字、文本字符串、日期或对象的变量组成，Dictionary 对象可以包含几种不同类型的键。

可以使用下列方法为 Dictionary 对象添加元素。

方法 1：使用.Add 方法

```
With CreateObject("scripting.dictionary")  
    .Add "one","item1"  
End With
```

在 Dictionary 对象中添加一个元素，其键为“one”，内容为“item1”。

方法 2：使用.Item()=方法

```
With CreateObject("scripting.dictionary")  
    .Item("one"),"item1"
```



End With

在 Dictionary 对象中添加一个元素，其键为 “one”，内容为 “item1”。

方法 3：使用=.Item()方法

```
With CreateObject("scripting.dictionary")
```

```
    x=.Item("one")
```

End With

将键为 “one” 的元素赋值给变量 x。如果不存在键为 “one” 的元素，那么将在 Dictionary 中添加该元素，但没有内容。

方法 4：使用对象变量

```
Set myDict=CreateObject("scripting.dictionary")
```

```
myDict("one")=Date
```

或者：

```
myDict.Item("one")=Date
```

在 Dictionary 中添加一个元素，其键为 “one”，内容为日期。

或者：

```
myDict.Add("aa1","today")
```

在 Dictionary 中添加一个元素，其键为 “aa1”，内容为 “today”。

在使用上述方法给 Dictionary 对象添加元素时，如果要添加元素的键已经存在，那么使用 Add 方法会报错，而使用 Item()=方法或者使用对象变量时会覆盖或替换掉已存在的元素。

下面是不同类型的键示例。

示例 1：文本字符串键

```
With CreateObject("scripting.dictionary")
```

```
    .Items("aa1") = "example 1 "
```



```
.Add "aa2", "example 2"  
  
x = .Item("aa3")  
  
myDict("aa4") = "example 3"  
  
End With
```

示例 2：数字键

```
With CreateObject("scripting.dictionary")  
  
    .Items(2) = "example 4"  
  
    .Items(1234589) = "example 5"  
  
    .Add 23, "example 6"  
  
    x = .Item(45)  
  
    myDict(56788) = "example 7"  
  
End With
```

示例 3：日期键

```
With CreateObject("scripting.dictionary")  
  
    .Items(Date) = "example 8"  
  
    .Items(CDate("12-03-2013")) = "example 9"  
  
    .Add DateAdd("m", 2, 1), "example 10"  
  
    x = .Item(DateSerial(2013, 5, 5))  
  
    myDict(56788) = "example 11"  
  
End With
```

示例 4：对象键

```
With CreateObject("scripting.dictionary")  
  
    .Items(TextBox1) = "example 12"
```



```

.Items(TextBox2) = "example 13"
.Add ListBox1, "example 14"
x = .Item(Sheet1)
myDict(ComboBox1) = "example 15"
End With

```

示例 5：工作表中的 ActiveX 控件

```

With CreateObject("scripting.dictionary")
    For Each it In Sheets("sheet1").OLEObjects
        .Item(it) = it.Name
    Next
End With

```

示例 6：一个 Dictionary 对象中的几种键

```

With CreateObject("scripting.dictionary")
    For Each it In Array("bb1", 12, 45673, Date + 10, sheet4)
        .Item(it) = "Item 1" & .Count
    Next
End With

```

可以使用 `CompareMode` 属性来设置键是否区分大小写。如果没有指定该属性或者将其值指定为 0，即二进制比较（`BinaryCompare`）则区分大小写，指定为 1，即文本比较（`TextCompare`）则不区分大小写。

下面的代码：

```

Sub test1()
    Dim i, y, it
    With CreateObject("scripting.dictionary")
        For Each it In Array("aa1", "AA1", "Aa1", "aA1", "bb1", "BB1", "Bb1", "bB1")

```



```

        y = .Item(it)
    Next

    MsgBox .Count
    MsgBoxJoin(.Keys, vbLf)

End With
End Sub

```

运行后的结果如下图 2 所示：

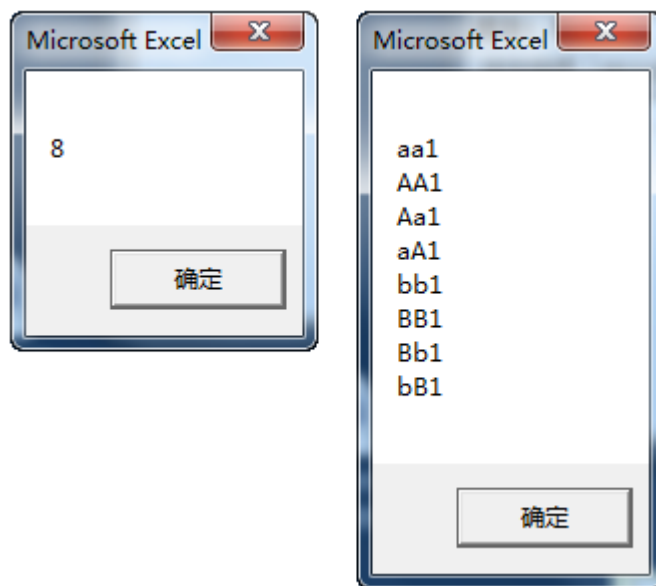


图 2

下面的代码：

```

Sub test2()
    Dim i, y, it
    With CreateObject("scripting.dictionary")
        .CompareMode = 1
        For Each it In Array("aa1", "AA1", "Aa1", "aA1", "bb1", "BB1", "Bb1", "bB1")
            y = .Item(it)
        Next
    End With

    MsgBox .Count

```



```
MsgBoxJoin(.Keys, vbLf)
```

```
End With
```

```
End Sub
```

运行后的结果如下图 3 所示：

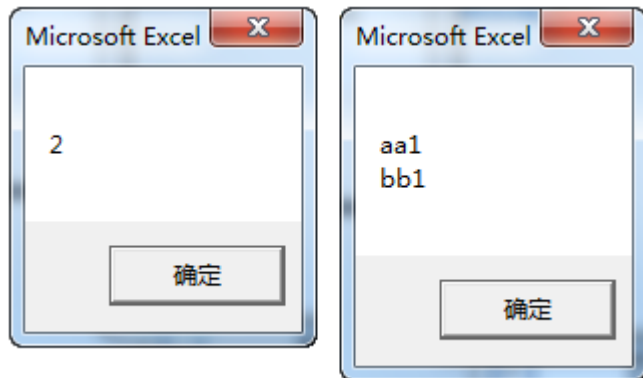


图 3

下面是给 Dictionary 对象添加不同类型的元素内容的示例代码。

示例 7：元素内容为空

```
With CreateObject("scripting.dictionary")
```

```
    x = .Item("aa1")
```

```
End With
```

示例 8：空文本字符串

```
With CreateObject("scripting.dictionary")
```

```
    .Item("aa2") = vbNullString
```

```
    .Item("aa3") = ""
```

```
    .Add "aa4", ""
```

```
    myDict("aa5") = vbNullString
```

```
End With
```



示例 9 : 文本字符串

```
With CreateObject("scripting.dictionary")  
    .Item("aa6") = "abcde"  
    .Add "aa7", "abcde"  
    myDict("aa8") = "abcde"  
End With
```

示例 10 : 不可打印字符

```
With CreateObject("scripting.dictionary")  
    .Item("aa9") = vbTab  
    Add "aa10", vbLf  
    myDict("aa11") = vbCrLf  
End With
```

示例 11 : 数字

```
With CreateObject("scripting.dictionary")  
    .Item("aa12") = 12345 ' Integer  
    .Add "aa13" = 1234589 ' Long  
    myDict("aa14") = RGB(23, 45, 678) ' Long  
End With
```

示例 12 : 日期

```
With CreateObject("scripting.dictionary")  
    .Item("aa15") = Date
```



```
.Add "aa16", CDate("23-04-2012")  
myDict("aa17") = DateSerial(2013, 10, 12)  
End With
```

示例 13：一维数组

```
With CreateObject("scripting.dictionary")  
    .Item("aa18") = Array("aa1", "aa2", "aa3")  
    Add "aa19", Split("bb1_cc1_dd1", "_")  
    myDict("aa20") = Array("aa1", "aa2", "aa3")  
End With
```

示例 14：多维数组

```
With CreateObject("scripting.dictionary")  
    ReDimmy(6, 10)  
    .Item("aa21") = my  
    .Add "aa22", Range("A1:K10")  
    myDict("aa23") = Range("A1:K10").Formula  
End With
```

示例 15：对象

```
With CreateObject("scripting.dictionary")  
    Set .Items("aa24") = Range("A1:K10") ' Range  
    Set myDict("aa25") = Range("A1:K10")  
    .Add "aa26", Range("A1:K10")  
End With
```

如果使用.Items=对象变量，则需要使用 Set 语句来将对象赋给 Dictionary，否则



该对象的缺省属性将是 Dictionary 对象元素内容。如果使用 Add 方法，那么将直接在放置对象到 Dictionary 中。

示例 16：用户窗体控件

```
With CreateObject("scripting.dictionary")  
    For Each it In Controls  
        Set .Item(it.Name) = it  
        .Add it.name & "_", it  
    Next  
End With
```

示例 17：ActiveX 控件

```
With CreateObject("scripting.dictionary")  
    For Each it In Sheets("sheet1").OLEObjects  
        Set .Item(it.Name) = it  
        .Add it.name & "_", it  
    Next  
End With
```

示例 18：工作簿中的所有工作表

```
With CreateObject("scripting.dictionary")  
    For Eachsh In Sheets  
        Set .Item(sh.Name) = sh  
        .Add sh.Name & "_", sh  
        SetmyDict(sh.Name & "_#") = sh  
    Next  
End With
```



End With



字典 (Dictionary) 对象的属性和方法

在上文中已经使用过 Dictionary 对象的一些属性和方法。

Dictionary 对象有下列 4 个属性：

CompareMode 属性、Count 属性、Item 属性、Key 属性。

Dictionary 对象有下列 6 个方法：

Add 方法、Exists 方法、Items 方法、Remove 方法、RemoveAll 方法。

下面详细介绍这些属性和方法。

Add 方法

语法为：

```
Dictionary 对象.Addkey,item
```

在字典对象中添加元素。其中，参数 key 为字典对象中的键，参数 item 为字典元素内容，均为 String 类型。

示例代码：

```
Set objDict=New Dictionary  
iVal=1  
objDict.AddCStr(iVal),"item1"
```



CompareMode 属性

语法为：

Dictionary 对象.CompareMode [=CompareMethodConst]

设置或者返回字典对象中键的比较模式。其中，CompareMethodConst 可以是 BinaryCompare(0, 二进制比较)、TextCompare(1, 文本比较)和 DatabaseCompare(2, 数据库)。

Count 属性

Dictionary 对象.Count

获取字典对象中元素的数量，因为每个元素有自己唯一的键，所以 Count 属性也指明了唯一键的数量。

示例代码：

```
Dim varArray As Variant
varArray = objDict.Items
For i = 0 To objDict.Count - 1
    Set oObj = varArray(i)
Next
```

Exists 方法

语法为：

Dictionary 对象.Exists (key)

确定在字典对象中是否存在指定的键。若存在，则返回 TRUE，否则返回 FALSE。

如果试图返回不存在的键或者将新键赋给不存在的键，那么这个不存在的键就会添加到字典中并且其元素内容为空。因此，先使用 Exists 来检查以确保存在指定的键。



示例代码：

```
If objDict.Exists(strOldKey) Then  
    objDict.Key(strOldKey) = strNewKey  
End If
```

Item 属性

语法为：

```
Dictionary 对象.Item(key) = item
```

设置指定键的元素内容。

```
value =Dictionary 对象.Item(key)
```

获取指定键的元素内容。

如果试图设置的元素指定的键不存在，那么会自动在字典中添加该键和元素内容。

示例代码：

```
Dim strKey As String  
Dim strName As String  
strName = "NewName"  
strKey = "Name"  
If objDict.Exists(strKey) Then  
    objDict.Item(strKey) = strName  
Else  
    MsgBox"键"&strKey&"不存在"  
End If
```



Items 方法

语法为：

Dictionary 对象.Items

返回包含字典对象内所有元素内容的一维数组。该数组是以 0 为基数的变体数组，其数据子类型与字典对象中的元素相匹配。

通常，通过键来访问字典元素。但是，也可以使用 Items 方法将字典元素内容存储在变体数组中，然后通过访问该数组来访问相应的字典元素。Ubound 属性表明数组中元素的数目。由于一维数组的下限值是 0，因此使用 Ubound(.Items)+1 表示字典中元素的数目。

示例代码：

```
Dim varArray As Variant
varArray = objDict.Items
For i = 0 To objDict.Count - 1
    Debug.Print varArray(i)
Next
```

Key 属性

语法为：

Dictionary 对象.Key(key) = newKey

返回字典对象中的键或者使用新键替换已存在的键。

示例代码：

```
Private Function ChangeKeyValue(strOldKey As String, strNewKey As string) As Boolean
    If objDict.Exists(strOldKey) Then
        objDict.Key(strOldKey) = strNewKey
    End If
End Function
```



```
        ChangeKeyValue = True
    Else
        ChangeKeyValue = False
    End If
End Function
```

Keys 方法

语法为：

Dictionary 对象.Keys

返回包含字典对象中所有键的一维数组，以 0 为基数，因此 Ubound(.Keys)+1 也表示字典中元素的数目。

示例代码：

```
Dim varArray As Variant
varArray = objDict.Keys
For i = 0 To objDict.Count - 1
    Debug.Print varArray(i)
Next
```

Remove 方法

语法为：

Dictionary 对象.Remove key

从字典对象中删除指定键及相关元素内容。如果删除的键不存在，则会产生运行时错误。



RemoveAll 方法

语法为：

```
Dictionary 对象.RemoveAll
```

清空字典，即从字典中删除所有键及相关元素内容。



字典 (Dictionary) 对象的常用操作

自动创建唯一键

有时，必须自动为每个元素生成唯一的键，可以使用 Dictionary 对象的 Count 属性：

```
Sub CreateUniqueKey()  
    Dim it  
    With CreateObject("scripting.dictionary")  
        For Each it In Array("aa1", "aa2", "aa1", "aa3", "aa4", "aa2")  
            .Item(.Count) = it & "_contents"  
        Next  
    End With  
End Sub
```

获取唯一元素列表

在 Dictionary 对象中添加元素时，会创建唯一键列表。因此，可以利用这个特点获取唯一元素列表。例如下面的代码：

```
Sub GetUniqueValue()  
    Dim it, x  
    With CreateObject("scripting.dictionary")  
        For Each it In Array(22, 33, 44, 22, 3, 22, 55, 44)  
            x = .Item(it)  
        Next  
        '因为 44 和 22 重复,所以有 5 个唯一键  
        MsgBox "唯一值有" & .Count & "个,分别为:" & vbCrLf & _
```



```
Join(.Keys, vbCrLf)

End With

End Sub
```

运行后的结果如下图 4 所示：

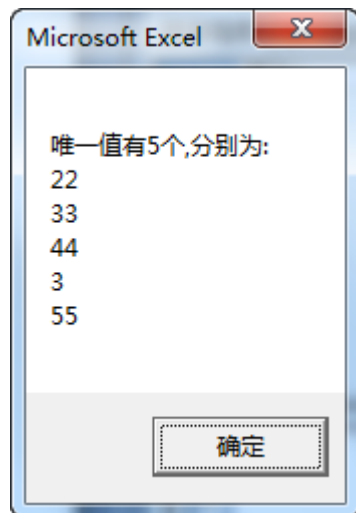


图 4

检查键/项目是否在字典中存在

在使用 Add 方法添加元素时，如果键已存在，就会出错。因此，在添加元素前，使用 Exists 方法检查键是否已存在：

```
With CreateObject("scripting.dictionary")

    For Each it In Array("aa1", "aa2", "aa1", "aa3", "aa4", "aa2")

        If Not .Exists(it) Then .Add it, it & "_content"

    Next

End With
```

在使用方法.Item=在字典中添加元素时，若键已经存在，VBA 不会给出提示消息，而是替换已存在的元素内容。因此，如果不想已存在的元素被替换，那么应该先检查键是否存在：

```
With CreateObject("scripting.dictionary")

    For Each it In Array("aa1", "aa2", "aa1", "aa3", "aa4", "aa2")
```



```
        If Not .Exists(it) Then .Item(it) = it & "_content"  
    Next  
End With
```

如果想要修改字典中某元素的内容,可以先检查该元素是否存在。如果键不存在,那么代码在读取内容时会自动创建:

```
x=.Item("one")
```

如果想要避免在字典中添加空项,那么先要检查键是否存在:

```
With CreateObject("scripting.dictionary")  
    For Each it In Array("aa1", "aa2", "aa1", "aa3", "aa4", "aa2")  
        .Item(it) = it & "_content"  
    Next  
    If .Exists("aa2") Then .Item("aa2") = "new value"  
End With
```

从 Dictionary 中获取元素

方法 1: 使用键

通过键直接读取或者写入元素内容,这是 Dictionary 对象的特点。

例如,下面的代码使用.Item("aa2")获取键为“aa2”的元素内容:

```
With CreateObject("scripting.dictionary")  
    For Each it In Array("aa1", "aa2", "aa1", "aa3", "aa4", "aa2")  
        .Item(it) = it & "_content"  
    Next  
    MsgBox .Item("aa2")  
End With
```

注意到 Item 属性仅使用键作为参数,没有索引号。

方法 2: 使用 Items 方法



也可以使用 Items 方法, 基于添加到 Dictionary 对象中的元素顺序来获取元素项。Items 方法返回含有所有元素内容的变体数组, 使用数组的索引值来读取元素内容。

```
Sub GetDicValue()  
    Dim it, sp  
    With CreateObject("scripting.dictionary")  
        For Each it In Array("aa1", "aa2", "aa1", "aa3", "aa4", "aa2")  
            .Item(it) = it & "_content"  
        Next  
  
        '第 1 个元素  
        MsgBox Application.Index(.Items, 1, 1)  
        MsgBox Filter(.Items, "")(0)  
        MsgBox Filter(.Items, "")(LBound(.Keys))  
  
        '第 2 个元素  
        MsgBox Application.Index(.Items, 1, 2)  
        MsgBox Filter(.Items, "")(1)  
  
        '最后一个元素  
        MsgBox Application.Index(.Items, 1, .Count)  
        MsgBox Filter(.Items, "")(UBound(.Keys))  
        MsgBox Filter(.Items, "")(.Count - 1)  
  
        sp = .Items ' 变体数组  
        MsgBox sp(0) ' 第 1 个元素  
        MsgBox sp(1) ' 第 2 个元素  
        MsgBox sp(UBound(sp)) '最后一个元素  
    End With  
End Sub
```

代码中, 使用了 Excel 的 Index 函数查找元素内容。也使用了 VBA 的 Filter 函数来



获取相应的元素内容，此时所有的值都将被转换成文本字符串。

方法 3：使用 Keys 方法

也可以使用 Keys 方法，基于添加到 Dictionary 对象中的元素顺序来获取元素项。Keys 方法返回含有所有元素的键的变体数组，使用数组的索引值来读取元素的键，然后使用该键来获取相应元素的内容。

```
Sub GetDicValue3()  
    Dim it, sn  
  
    With CreateObject("scripting.dictionary")  
        For Each it In Array("aa1", "aa2", "aa1", "aa3", "aa4", "aa2")  
            .Item(it) = it & "_contents"  
        Next  
  
        '第 1 个键  
        MsgBox Application.Index(.Keys, 1, 1)  
        MsgBox Filter(.Keys, "")(0)  
        MsgBox Filter(.Keys, "")(LBound(.Keys))  
  
        '第 2 个键  
        MsgBox Application.Index(.Keys, 1, 2)  
        MsgBox Filter(.Keys, "")(1)  
  
        '最后一个键  
        MsgBox Application.Index(.Keys, 1, .Count)  
        MsgBox Filter(.Keys, "")(UBound(.Keys))  
        MsgBox Filter(.Keys, "")(.Count - 1)  
  
        '第 1 个键对应的元素内容  
        MsgBox .Item(Application.Index(.Keys, 1, 1))  
        MsgBox .Item(Filter(.Keys, "")(0))
```



```

'第 2 个键对应的元素内容
MsgBox .Item(Application.Index(.Keys, 1, 2))
MsgBox .Item(Filter(.Keys, "")(1))

sn = .Keys ' 变体数组
MsgBox sn(0) ' 第 1 个键
MsgBox sn(1) ' 第 2 个键
MsgBox sn(UBound(sn)) '最后一个键

MsgBox .Item(sn(0)) ' 第 1 个键对应的元素内容
MsgBox .Item(sn(1)) ' 第 2 个键对应的元素内容
MsgBox .Item(sn(UBound(sn))) ' 最后一个键对应的元素内容

End With
End Sub

```

与前面使用的方法相同。代码中，使用了 Excel 的 Index 函数查找元素内容。也使用了 VBA 的 Filter 函数来获取相应的元素内容，此时所有的值都将被转换成文本字符串。

获取指定键的元素内容

要求获取指定键为 aa4 的元素内容。

方法 1：使用 Keys 方法

```

Sub GetSpecDicValue1()
    Dim it
    With CreateObject("scripting.dictionary")
        For Each it In Array("aa1", "aa2", "aa1", "aa3", "aa4", "aa2")
            .Item(it) = it & "_content"
        Next
    End With
End Sub

```



```

        For Each it In .Keys
            If it = "aa4" Then MsgBox .Item(it)
        Next
    End With
End Sub

```

方法 2：使用对象变量

```

Sub GetSpecDicValue2()
    Dim myDic, it
    Set myDic = CreateObject("scripting.dictionary")
    For Each it In Array("aa1", "aa2", "aa1", "aa3", "aa4", "aa2")
        myDic.Item(it) = it & "_content"
    Next

    For Each it In myDic
        If it = "aa4" Then MsgBox myDic.Item(it)
    Next
End Sub

```

获取包含指定数据的键的元素内容

下面的代码查找包含“cc”的键的元素内容。

```

Sub GetSpecDicValue3()
    Dim it
    With CreateObject("scripting.dictionary")
        For Each it In Array("aa14", "bb345", "cc392", "rrr987")
            .Item(it) = it & "_content"
        Next
    End With
End Sub

```



```

        For Each it In Filter(.Keys, "cc")
            MsgBox .Item(it)
        Next

    End With
End Sub

```

操作字典中的键

示例 1：将所有键放置到工作表中

```

Sub UseKeys1()
    Dim it
    With CreateObject("scripting.dictionary")
        For Each it In Array("aa1", "aa2", "aa1", "aa3", "aa4", "aa2")
            .Item(it) = it & "_content"
        Next

        Cells(1, 1).Resize(, .Count) = .Keys ' 按行
        Cells(1, 1).Resize(.Count) = Application.Transpose(.Keys) ' 按列
    End With
End Sub

```

示例 2：将所有键放置到变量中

```

Sub UseKeys2()
    Dim it, sn
    With CreateObject("scripting.dictionary")
        For Each it In Array("aa1", "aa2", "aa1", "aa3", "aa4", "aa2")
            .Item(it) = it & "_content"

```




```

        Next

        sn = .Keys
    End With
End Sub

```

上述代码在变体数组 `sn` 中存放了 `Dictionary` 对象的所有键。

示例 3：将所有键放置到文本字符串中

```

Sub UseKeys3()
    Dim it
    With CreateObject("scripting.dictionary")
        For Each it In Array("aa1", "aa2", "aa1", "aa3", "aa4", "aa2")
            .Item(it) = it & "_contents"
        Next

        MsgBoxJoin(.Keys, vbCrLf)
    End With
End Sub

```

示例 4：使用键进行计算

如果 `Dictionary` 对象中的键都是数字，那么可以应用许多 Excel 函数对 `Keys` 方法返回的数组进行计算。

下面的代码获取数字最大的键：

```
Application.Max(.Keys)
```

下面的代码获取数字最小的键：

```
MsgBoxApplication.Min(.Keys)
```

下面的代码获取第 3 大的键：

```
MsgBoxApplication.Large(.Keys, 3)
```



示例 5：根据条件筛选键

```
Sub UseKeys4()  
    Dim it  
    With CreateObject("scripting.dictionary")  
        For Each it In Array("aa1", "aa2", DateSerial(2012, 12, 3), DateSerial(2012,  
12, 4), _  
DateSerial(2012, 12, 8), 22, DateSerial(2013, 12, 3), 44)  
            .Item(it) = it & "_content"  
        Next  
  
        MsgBox Join(Filter(.Keys, "2012"), vbCrLf)  
    End With  
End Sub
```

代码获取键中含有“2012”的所有键。Filter 函数将所有键转换为文本字符串，因此筛选条件必须是文本字符串。如果想要使用筛选得到的键来获取相应的元素内容，那么必须将键转换成其原来的类型。

```
Sub UseKeys5()  
    Dim it  
    With CreateObject("scripting.dictionary")  
        For Each it In Array("aa1", "aa2", DateSerial(2012, 12, 3), DateSerial(2012,  
12, 4), _  
DateSerial(2012, 12, 8), 22, DateSerial(2013, 12, 3), 44)  
            .Item(it) = it & "content"  
        Next  
  
        For Each it In Filter(.Keys, "2012")  
            MsgBox .Item(CDate(it))  
        Next  
    End With  
End Sub
```

示例 6：修改元素的键



```

Sub ChangeKey()
    With CreateObject("scripting.dictionary")
        .Item("aa") = "this is the first item"
        MsgBox .Item("aa")
        .Key("aa") = "bb"
        MsgBox .Item("bb")
    End With
End Sub

```

Items 方法在字典中的运用

示例 1：将所有元素内容放置到工作表中

```

Sub UseItems1()
    Dim it
    With CreateObject("scripting.dictionary")
        For Each it In Array("aa1", "aa2", "aa1", "aa3", "aa4", "aa2")
            .Item(it) = it & "_content"
        Next

        Cells(1, 1).Resize(, .Count) = .Items ' 按行
        Cells(1, 1).Resize(.Count) = Application.Transpose(.Items) ' 按列
    End With
End Sub

```

注：元素内容应该是数字、文本字符串或者日期。

示例 2：将所有元素内容存放到变量中

```

Sub UseItems2()
    Dim it, sn
    With CreateObject("scripting.dictionary")

```



```

        For Each it In Array("aa1", "aa2", "aa1", "aa3", "aa4", "aa2")
            .Item(it) = it & "_content"
        Next

        sn = .Items

    End With
End Sub

```

示例 3：将所有元素内容放置到文本字符串中

```

Sub UseItems3()
    Dim it
    With CreateObject("scripting.dictionary")
        For Each it In Array("aa1", "aa2", "aa1", 22, 3, 22, DateSerial(2013, 12, 3), 44)
            .Item(it) = it & "_content"
        Next

        MsgBox Join(.Items, vbCrLf)
    End With
End Sub

```

示例 4：使用元素内容进行计算

如果 Dictionary 对象中元素内容包含数字，那么可以应用许多 Excel 函数对 Items 方法返回的数组进行计算。

下面的代码获取数字最大的元素内容：

```
Application.Max(.Items)
```

下面的代码获取数字最小的元素内容：

```
Application.Min(.Items)
```

下面的代码获取数字第 3 大的元素内容：

```
Application.Large(.Items, 3)
```

示例 5：根据条件筛选元素



```

Sub Useltmes4()
    Dim it
    With CreateObject("scripting.dictionary")
        For Each it In Array("aa1", "aa2", DateSerial(2012, 12, 3), DateSerial(2012, 12, 4), _
DateSerial(2012, 12, 8), 22, DateSerial(2013, 12, 3), 44)
            .Item(it) = it & "_content"
        Next

        MsgBox Join(Filter(.Items, "2012"), vbCrLf)
    End With
End Sub

```

代码获取元素内容中含有“2012”的所有元素。**Filter** 函数将所有元素内容转换为文本字符串，因此筛选条件必须是文本字符串。

复制 Dictionary 对象中的元素

可以将已存在的元素内容赋值给另一个元素或者新键。

```

Sub CopyItem()
    With CreateObject("scripting.dictionary")
        .Item("aa") = "the first item"
        .Item("bb") = .Item("aa")
        MsgBox .Item("bb")
    End With
End Sub

```



修改 Dictionary 对象中元素的内容

示例 1：替换 Dictionary 对象中已有元素的内容

使用 Item 属性将使用同一键的新元素内容替换已有的元素内容。

```
Sub Changeltem1()  
    Dim it  
    With CreateObject("scripting.dictionary")  
        For Each it In Array("aa1", "aa2", "aa3", "aa4", "aa5", "aa6")  
            .Item(it) = it & "_content"  
        Next  
        For Each it In Array("aa1", "aa2", "aa5", "aa6")  
            .Item(it) = it & "_new content"  
        Next  
  
        MsgBox Join(.Items, "|")  
    End With  
End Sub
```

示例 2：在已存在的元素内容中添加数据

仅应用于包含单精度数字、字符串或日期的元素内容。

```
Sub Changeltem2()  
    Dim it  
    With CreateObject("scripting.dictionary")  
        For Each it In Array("aa1", "aa2", "aa1", "aa3", "aa4", "aa2")  
            .Item(it) = "aa"  
        Next  
        For Each it In Array("aa1", "aa2", "aa1", "aa3", "aa4", "aa2")  
            .Item(it) = .Item(it) & "_bb"  
        Next  
    End With  
End Sub
```



```
MsgBox Join(.Items, "|")  
End With  
End Sub
```

示例 3：对元素内容执行计算

仅应用于包含单精度数字、字符串、日期或布尔值的元素。

（1）数字

```
Sub CalcItem1()  
Dim it  
With CreateObject("scripting.dictionary")  
For Each it In Array("aa1", "aa2", "aa1", "aa3", "aa4", "aa2")  
.Item(it) = 10  
Next  
For Each it In Array("aa1", "aa2", "aa1", "aa3", "aa4", "aa2")  
.Item(it) = .Item(it) + 40  
Next  
  
MsgBox Join(.Items, "|")  
End With  
End Sub
```

（2）日期

```
Sub CalcItem2()  
Dim it  
With CreateObject("scripting.dictionary")  
For Each it In Array("aa1", "aa2", "aa3", "aa4")  
.Item(it) = Date + .Count  
Next  
For Each it In Array("aa1", "aa2", "aa3", "aa4")  
.Item(it) = DateAdd("m", 1, .Item(it))  
Next  
End With  
End Sub
```



```

        Next

        MsgBox Join(.Items, vbCrLf)

    End With
End Sub

```

（3）布尔值

```

Sub CalcItem3()
    Dim it
    With CreateObject("scripting.dictionary")
        For Each it In Array("aa1", "aa2", "aa3", "aa4", "aa2")
            .Item(it) = False
        Next
        For Each it In Array("aa1", "aa2", "aa3", "aa4", "aa2")
            .Item(it) = Format(.Item(it), "Yes/No")
        Next

        MsgBox Join(.Items, "|")
    End With
End Sub

```

示例 4：修改内容包含数组的元素

（1）一维数组

读取包含数组内容的元素的代码：

```

With CreateObject("scripting.dictionary")
    .Item("aa") = Array("zz1", "zz2", "zz3", "zz4")
    MsgBox .Item("aa")(3)
End With

```

如果以下的代码来修改包含数组的元素内容，虽然不会报错，但元素也不会被修改。




```
With CreateObject("scripting.dictionary")
```

```
.Item("aa") = Array("zz1", "zz2", "zz3", "zz4")  
MsgBox .Item("aa")(3)  
  
.Item("aa")(3) = "第 4 个元素是:" & .Item("aa")(3)  
MsgBox .Item("aa")(3)
```

```
End With
```

可以使用一种变通的方式，即先将数组读取到变量中，然后修改该数组变量的一个或多个元素，最后使用该数组变量替换字典元素的内容。

```
With CreateObject("scripting.dictionary")
```

```
.Item("aa") = Array("zz1", "zz2", "zz3", "zz4")  
  
sn = .Item("aa")  
sn(3) = "第 4 个元素是:" & sn(3)  
.Item("aa") = sn  
  
MsgBox .Item("aa")(3)
```

```
End With
```

(2) 二维数组

要从包含二维数组的 Dictionary 对象元素中读取元素内容，可以使用下面的代码：

```
With CreateObject("scripting.dictionary")
```

```
ReDim sn(3, 4)  
For j = 0 To UBound(sn)  
    For jj = 0 To UBound(sn, 2)  
        sn(j, jj) = j + 5 * jj  
    Next  
Next  
  
.Item("aa") = sn  
MsgBox .Item("aa")(2, 3)
```



End With

与一维数组一样，如果使用下面的代码直接替换字典元素数组中的内容，没有效果。

```
With CreateObject("scripting.dictionary")
    ReDim sn(3, 4)

    For j = 0 To UBound(sn)
        For jj = 0 To UBound(sn, 2)
            sn(j, jj) = j + 5 * jj
        Next
    Next

    .Item("aa") = sn

    MsgBox .Item("aa")(2, 3)

    .Item("aa")(2, 3) = 10 * .Item("aa")(2, 3)
    MsgBox .Item("aa")(2, 3)
End With
```

此时，可以选将元素内容数组读取到数组变量中，然后修改该数组变量的一个或多个元素，最后使用该数组变量替换 Dictionary 中元素的内容。

```
With CreateObject("scripting.dictionary")
    ReDim sn(3, 4)

    For j = 0 To UBound(sn)
        For jj = 0 To UBound(sn, 2)
            sn(j, jj) = j + 5 * jj
        Next
    Next

    .Item("aa") = sn

    MsgBox .Item("aa")(2, 3)

    sp = .Item("aa")
```



```
sp(2, 3) = 10 * sp(2, 3)
.Item("aa") = sp

MsgBox .Item("aa")(2, 3)

End With
```

从 Dictionary 对象中的删除元素

删除某个元素：

```
Sub DeleteItem()
    Dim it
    With CreateObject("scripting.dictionary")
        For Each it In Array("aa1", "aa2", "aa3", "aa4", "aa2")
            .Item(it) = it & "_content"
        Next

        .Remove "aa3"
    End With
End Sub
```

删除所有元素：

```
Sub DeleteAllItems()
    Dim it
    With CreateObject("scripting.dictionary")
        For Each it In Array("2", "33", "aa3", "aa4", "aa2")
            .Item(it) = it & "_content"
        Next

        .RemoveAll
    End With
End Sub
```



示例

示例 1：基于键删除重复的行

工作表列 A 中是记录的键，列 B 中是记录数据。下面的代码删除重复的键的记录：

```
Sub DeleteDuplicates()  
    Dim sn, j  
    sn = Sheets("Sheet1").Cells(1).CurrentRegion.Resize(, 5)  
  
    With CreateObject("scripting.dictionary")  
        For j = 1 To UBound(sn)  
            .Item(sn(j, 1)) = Application.Index(sn, j, 0)  
        Next  
  
        Sheets("Sheet1").Cells(1, 4).Resize(.Count, UBound(sn, 2)) = Application.Index(.Items, 0, 0)  
    End With  
End Sub
```

示例 2：判断一列数据是否存在重复

下面的代码判断列 A 中是否存在重复数据。

```
Sub test()  
    Dim d As Object, Rng As Range, cel As Range  
    Set d = CreateObject("Scripting.Dictionary")
```



```

Set Rng = Range(Cells(1, 1), Cells(1, 1).End(xlDown))
For Each cel In Rng
    On Error Resume Next
    d.Addcel.Text, cel.Text
Next
If d.Count<>Rng.Cells.Count Then MsgBox "该列存在重复数据!"
End Sub

```

示例 3：筛选列 B 中包含最高值的记录

列 A 中是记录的键，包含相同键的记录。列 B 包含记录的数据，可以排序。下面的代码筛选出各键在列 B 中数值最大的记录：

```

Sub FilterRecordsMaxValueInColB()
    Dim sn, j
    Sheets("Sheet3").Cells(1).CurrentRegion.Sort
    Sheets("Sheet3").Cells(1, 2)
    sn = Sheets("Sheet3").Cells(1).CurrentRegion

    With CreateObject("scripting.dictionary")
        For j = 1 To UBound(sn)
            .Item(sn(j, 1)) = Application.Index(sn, j, 0)
        Next

        Sheets("Sheet3").Cells(1, 4).Resize(.Count, UBound(sn, 2)) = Application.Index(.Items, 0, 0)
    End With
End Sub

```



示例 4 : 筛选列 B 中包含最小值的记录

列 A 中是记录的键，包含相同键的记录。列 B 包含记录的数据，可以排序。下面的代码筛选出各键在列 B 中数值最小的记录：

```
Sub FilterRecordsMinValueInColB()  
    Dim sn, j  
    Sheets("Sheet3").Cells(1).CurrentRegion.Sort  
    Sheets("Sheet3").Cells(1, 2), 2  
    sn = Sheets("Sheet3").Cells(1).CurrentRegion  
  
    With CreateObject("scripting.dictionary")  
        For j = 1 To UBound(sn)  
            .Item(sn(j, 1)) = Application.Index(sn, j, 0)  
        Next  
  
        Sheets("Sheet3").Cells(1, 4).Resize(.Count, UBound(sn, 2)) = Application.Index(.Items, 0, 0)  
    End With  
End Sub
```

示例 5 : 合并记录

工作表列 A 中的记录的键，列 B 是记录的数据内容；列 E 包含列 A 中的键，列 F 包含要附加的内容。下面的代码合并以上 4 列中的数据为键和相应的记录内容：

```
Sub RecordsAppend()  
    Dim sn, sp, st, j  
    sn = Sheets("Sheet4").Cells(1).CurrentRegion.Resize(, 3)  
    sp = Sheets("sheet4").Cells(1, 5).CurrentRegion  
  
    With CreateObject("scripting.dictionary")  
        For j = 1 To UBound(sn)
```



```

        .Item(sn(j, 1)) = Application.Index(sn, j, 0)
    Next

    For j = 1 To UBound(sp)
        st = .Item(sp(j, 1))
        st(3) = sp(j, 2)
        .Item(sn(j, 1)) = Application.Index(st, 0, 0)
    Next

    Sheets("Sheet4").Cells(1, 10).Resize(.Count,
    UBound(sn, 2)) = Application.Index(.Items, 0, 0)

    End With
End Sub

```

示例 6：合并工作表

下面的代码将工作簿中的所有工作表按顺序合并到名为 **total** 的工作表中：

```

Sub IntegrationSheets()
    Dim sh, it
    With CreateObject("scripting.dictionary")
        For Each sh In Sheets
            .Item(sh.Name) = sh.UsedRange
        Next
        Sheets.Add.Name = "total"

        For Each it In .Items
            Sheets("total").Cells(Rows.Count,
            1).End(xlUp).Offset(1).Resize(UBound(it), UBound(it, 2)) =
            it
        Next
    End With
End Sub

```



示例 7：合并 CSV 文件

```
Sub IntegratieCSV()  
    Dim sn, j, it  
    sn=Split(CreateObject("wscript.shell").exec("cmd /c  
Dir \"G:\OF\*.csv\" /b").StdOut.ReadAll, vbCrLf)  
  
    With CreateObject("scripting.dictionary")  
        For j = 0 To UBound(sn)  
            .Item(sn(j)) = GetObject("G:\OF\"  
&sn(j)).Sheets(1).UsedRange.Value  
            GetObject("G:\OF\" &sn(j)).Close False  
        Next  
  
        Sheets.Add.Name = "total"  
  
        For Each it In .Items  
            Sheets("total").Cells(Rows.Count,1).End(xlUp).O  
ffset(1).Resize(UBound(it),UBound(it, 2)) = it  
        Next  
    End With  
End Sub
```

示例 8：创建唯一元素列表

添加元素到字典中以创建唯一键列表，无需元素必须有内容。可以使用方法 `=.Item(key)`，如果键不存在，将添加；如果键存在，则忽略，因此不会导致代码错误。



可以使用唯一元素列表来作为数据有效性列表项，或者填充组合框或列表框。例如下面的代码：

```
With CreateObject("scripting.dictionary")
    For Each it In Array("aa1", "aa2", "aa3", "aa2", "aa2",
"aa4", "aa5")
        y = .Item(it)
    Next

    Sheets("sheet1").Cells(1, 10).Validation.Add 3, , ,
Join(.Keys, ",")

    Sheets("sheet1").OLEObjects("Combobox1").Object.List
= .Keys
    Sheets("sheet1").ListBox1.List = .Keys
End With
```

在用户窗体中：

```
Private Sub Userform_initialize()
    With CreateObject("scripting.dictionary")
        For Each it In Array("aa1", "aa2", "aa3", "aa2",
"aa2", "aa4", "aa5")
            y = .Item(it)
        Next

        ComboBox1.List = .Keys
        ListBox1.List = .Keys
        Me("ComboBox2").List = .Keys
        Me("ListBox2").List = .Keys
        Controls("ComboBox3").List = .Keys
        Controls("Listbox3").List = .Keys

    End With
End Sub
```

示例 9：在工作表的两列中获取唯一值并填充组合框

下面的代码获取工作表列 C 和列 D 中的值，去掉重复值后，按字母顺序排序并



填充组合框。

```
Private Sub UserForm_Initialize()  
    Call Populate_cboCompType  
End Sub  
  
Private Sub Populate_cboCompType()  
    Dim i As Long, lrow As Long  
    Dim MakeListAs Range  
    Dim cel As Range  
    Dim d As Variant, It As Variant, a As Variant  
    Dim arr()  
    Dim wsAs Worksheet  
    Set ws = ThisWorkbook.Worksheets("Data")  
    On Error Resume Next  
    lrow = ws.Cells(Rows.Count, "A").End(xlUp).Row  
    If lrow = 2 Then  
        Me.cboCompType.Value = ws.Cells(2, "C").Value  
        Me.txtTypeDescription.Value = ws.Cells(2,  
"D").Value  
    Else  
        '创建一系列可用的组件类型列表  
        Set d = CreateObject("Scripting.Dictionary")  
        Set MakeList = ws.Range("C2", "C" &lrow)  
        '使用 Dictionary 对象创建唯一项列表  
        For Each It In MakeList  
            d.Add It.Value, It.Value '添加键和项  
        Next  
        '创建一组唯一项  
        a = d.items  
        '排序数组  
        Call BubbleSort(a)  
        '使用相应的值创建新数组
```



```

        i = 0
        ReDimarr(d.Count, 1)
        For Each It In a
            arr(i, 0) = It
            arr(i, 1) = Sheets("Data").Columns(3).Find(What:=It, _
                LookIn:=xlFormulas, _
                LookAt:=xlWhole, MatchCase:=False).Offset(,
1).Value
            i = i + 1
        Next
        '添加项到组合框
        Me.cboCompType.list() = arr
    End If
End Sub

Sub BubbleSort(MyArray As Variant)
    Dim First As Integer, last As Integer, i As Integer, j
    As Integer
    Dim temp As String, list As String
    First = LBound(MyArray)
    last = UBound(MyArray)
    For i = First To last - 1
        For j = i + 1 To last
            If MyArray(i) > MyArray(j) Then
                temp = MyArray(j)
                MyArray(j) = MyArray(i)
                MyArray(i) = temp
            End If
        Next j
    Next i
End Sub

```



示例 10：检查是否所有的 ActiveX 控件都有数据

```
Private dict As Dictionary

Sub M_vul()
    Set dict = CreateObject("scripting.dictionary")

    For Each it In Array("TextBox1", "TextBox2", "textbox3",
"combobox4", "combobox5")
        Set dict(it) = Sheets("sheet1").OLEObjects(it)
    Next
End Sub

Private Sub Textbox1_change()
    M_controle
End Sub

Private Sub M_controle()
    y = True

    For Each it Indict.Items
        y = y * (it.Object.Value<> "")
    Next

    Sheets("sheet1").CommandButton1.Visible = y
End Sub
```

示例 11：检查工作表是否存在

在标准模块中的代码：

```
Public dict_sheets As Dictionary

Sub M_sheet_exists()
    Set dict_sheets = CreateObject("scripting.dictionary")
```



```

    For Each it In Sheets
        x0 = dict_sheets(it.name)
    Next
End Sub

```

现在，可以在 VBproject (Workbook)的任何地方使用：

```

Sub M_continue()
    ' - - - If Not dict_sheets.Exists("Sheet10") Then
    Sheets.Add.Name = "Sheet10"

    ' - - -
End Sub

```

示例 12：转换文本字符串为二维数组

```

Sub ConvertStrToArray()
    Dim it
    With CreateObject("scripting.dictionary")
        For Each it In
Split("aa1_aa2_aa3_aa4_aa5|bb1_bb2_bb3_bb4_bb5|cc1_cc2_cc
3_cc4_cc5", "|")
            .Item(.Count) = Application.Index(Split(it,
"_"), 1, 0)
        Next

        Cells(10, 1).Resize(.Count, UBound(.Item(1))) =
Application.Index(.Items, 0, 0)
    End With
End Sub

```





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 上发表的关于 Dictionary 对象知识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。

