

《别怕，Excel VBA其实很简单》

图书基本信息

书名：《别怕，Excel VBA其实很简单》

13位ISBN编号：9787115289094

10位ISBN编号：7115289093

出版时间：2012-10-1

出版社：人民邮电出版社

作者：Excel之家 (Excel Home)

页数：314

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《别怕，Excel VBA其实很简单》

内容概要

《别怕，Excel VBA其实很简单》是Excel Home网站考虑到大多数读者没有编程基础的实际情况，用浅显易懂的语言和生动形象的比喻，并配合大量插画，介绍Excel中看似复杂的概念和代码；从简单的宏录制、VBA编程环境和基础语法的介绍，到常用对象的操作与控制、Excel事件的调用与控制、用户界面设计、代码调试与优化都进行了形象的介绍。

《别怕，Excel VBA其实很简单》适合想提高工作效率的办公人员，特别是经常需要处理、分析大量数据的相关人员，以及财经专业的高校师生阅读。

《别怕，Excel VBA其实很简单》

作者简介

罗国发，网名ggsmart，人称叶枫，Excel Home官方微博管理员，Excel Home免费在线培训中心最受欢迎的讲师之一。中学教师，拥有丰富的教育培训经验，擅长Excel数据处理与分析，在使用Excel VBA编程方面拥有丰富的学习心得和实践经验。其发表于Excel Home的技术帖和课件成为大量Excel VBA初学者的启蒙教材。

周庆麟，网名Kevin，中国最资深的微软全球最有价值专家（MVP），Excel Home创始人、站长，多年来从事于企业财务和信息化管理工作，对Microsoft Office在销售、财务、物流和公司管理方面的应用有着深入的研究，曾参与策划和编写了几十本畅销Office技术图书。同时也是一名优秀的技术顾问和培训讲师，有丰富的授课经验。

书籍目录

第1章 走进Excel VBA的世界

- 1.1 不会Excel的人，真伤不起
 - 1.1.1 做不完的表
 - 1.1.2 神速的"超人"
 - 1.1.3 你是怎样做工资条的
- 1.2 走自己的"录"，让别人重复去吧
 - 1.2.1 什么是宏
 - 1.2.2 用宏录下Excel操作
 - 1.2.3 让录下的操作再现一遍
- 1.3 还可以怎样执行宏
 - 1.3.1 给宏设置快捷键
 - 1.3.2 将宏指定给按钮
 - 1.3.3 将宏指定给常用工具栏按钮
- 1.4 是谁"挡住"了宏
 - 1.4.1 宏为什么不能工作了
 - 1.4.2 怎样修改宏安全级
- 1.5 VBA，Excel里的编程语言
 - 1.5.1 录制宏不能解决的问题
 - 1.5.2 让工资条一"输"到底
 - 1.5.3 VBA编程，让你的表格更加灵活
 - 1.5.4 什么是VBA
 - 1.5.5 宏和VBA有什么关系

第2章 开始VBA编程的第一步

- 2.1 揭开神秘面纱背后的真面目
 - 2.1.1 程序保存在哪里
 - 2.1.2 应该怎样编写程序
- 2.2 程序里都有什么
 - 2.2.1 代码
 - 2.2.2 过程
 - 2.2.3 模块
 - 2.2.4 对象
 - 2.2.5 对象的属性
 - 2.2.6 对象的方法
 - 2.2.7 关键字
- 2.3 VBA的编程环境--VBE
 - 2.3.1 打开VBE编辑器
 - 2.3.2 主窗口
 - 2.3.3 菜单栏
 - 2.3.4 工具栏
 - 2.3.5 工程资源管理器
 - 2.3.6 属性窗口
 - 2.3.7 代码窗口
 - 2.3.8 立即窗口
- 2.4 试写一个简单的VBA程序
 - 2.4.1 添加或删除模块
 - 2.4.2 动手编写程序
- 2.5 解除疑惑，一"键"倾心

第3章 Excel VBA基础语法

3.1 语法，编程的基础

3.1.1 这个笑话很凉快

3.1.2 VBA也有语法

3.1.3 学习VBA语法难吗

3.2 VBA里的数据类型

3.2.1 打酱油的故事

3.2.2 走进Excel的商店

3.2.3 VBA中有哪些数据类型

3.3 存储数据的容器：常量和变量

3.3.1 常量和变量

3.3.2 使用变量

3.3.3 使用常量

3.3.4 使用数组

3.4 集合、对象、属性和方法

3.4.1 对象，就像冰箱里的鸡蛋

3.4.2 对象的属性

3.4.3 对象的方法

3.5 连接的桥梁，VBA中的运算符

3.5.1 算术运算符

3.5.2 比较运算符

3.5.3 连接运算符

3.5.4 逻辑运算符

3.5.5 应该先进行什么运算

3.6 内置函数

3.6.1 VBA中的函数

3.6.2 VBA中有哪些函数

3.7 控制程序执行，VBA的基本语句结构

3.7.1 If...Then语句

3.7.2 Select Case语句

3.7.3 For...Next语句

3.7.4 Do While语句

3.7.5 Do Until语句

3.7.6 For Each...Next语句

3.7.7 其他的常用语句

3.8 Sub 过程，基本的程序单元

3.8.1 关于VBA过程

3.8.2 编写Sub过程需要了解的内容

3.8.3 从另一个过程执行过程

3.8.4 过程的作用域

3.9 自定义函数，Function过程

3.9.1 试写一个函数

3.9.2 使用自定义函数

3.9.3 怎么统计指定颜色的单元格个数

3.9.4 声明函数过程，规范的语句

3.10 合理地组织程序，让代码更优美

3.10.1 代码排版，必不可少的习惯

3.10.2 怎样排版代码

3.10.3 注释，让代码的意图清晰明了

第4章 常用对象

- 4.1 与Excel交流，需要熟悉的常用对象
 - 4.1.1 VBA编程与炒菜
 - 4.1.2 VBA是怎么控制Excel的
 - 4.1.3 应该记住哪些对象
- 4.2 一切由我开始，最顶层的Application对象
 - 4.2.1 ScreenUpdating属性
 - 4.2.2 DisplayAlerts属性
 - 4.2.3 EnableEvents属性
 - 4.2.4 WorksheetFunction属性
 - 4.2.5 给Excel梳妆打扮
 - 4.2.6 她和她的孩子们
- 4.3 管理工作簿，了解Workbook对象
 - 4.3.1 Workbook与Workbooks
 - 4.3.2 认识Workbook，需要了解的信息
 - 4.3.3 实际操作，都能做什么
 - 4.3.4 ThisWorkbook与ActiveWorkbook
- 4.4 操作工作表，认识Worksheet对象
 - 4.4.1 认识Worksheet对象
 - 4.4.2 操作工作表
 - 4.4.3 Sheets与Worksheets
- 4.5 核心，至关重要的Range对象
 - 4.5.1 多种方法引用Range对象
 - 4.5.2 还可以怎样得到单元格
 - 4.5.3 操作单元格，还需要了解什么
 - 4.5.4 亲密接触，操作单元格
- 4.6 不止这些，其他常见的对象
 - 4.6.1 名称，Names集合
 - 4.6.2 单元格批注，Comment对象
 - 4.6.3 给单元格化妆
- 4.7 典型的技巧与示例
 - 4.7.1 创建一个工作簿
 - 4.7.2 判断工作簿是否打开
 - 4.7.3 判断工作簿是否存在
 - 4.7.4 向未打开的工作簿中录入数据
 - 4.7.5 隐藏活动工作表外的所有工作表
 - 4.7.6 批量新建工作表
 - 4.7.7 批量对数据分类
 - 4.7.8 将工作表保存为新工作簿
 - 4.7.9 快速合并多表数据
 - 4.7.10 汇总同文件夹下多工作簿数据
 - 4.7.11 为工作表建立目录

第5章 Excel事件

- 5.1 让Excel自动响应你的行为
 - 5.1.1 让Excel自动问好
 - 5.1.2 事件，VBA里的自动开关
 - 5.1.3 事件过程
 - 5.1.4 编写事件过程
- 5.2 Worksheet事件

- 5.2.1 关于Worksheet事件
- 5.2.2 常用的Worksheet事件
- 5.2.3 Worksheet事件列表
- 5.3 Workbook事件
 - 5.3.1 关于Workbook事件
 - 5.3.2 常用的Workbook事件
 - 5.3.3 Workbook事件列表
- 5.4 别样的自动化
 - 5.4.1 MouseMove事件
 - 5.4.2 不是事件的事件
- 5.5 典型的技巧与示例
 - 5.5.1 一举多得，快速录入数据
 - 5.5.2 我该监考哪一场
 - 5.5.3 让文件每隔一分钟自动保存一次
- 第6章 用户界面设计
 - 6.1 在Excel中自由地设计界面
 - 6.1.1 关于用户界面
 - 6.1.2 控件，必不可少的调色盘
 - 6.2 使用控件，将工作表当作画布
 - 6.2.1 在工作表中使用窗体控件
 - 6.2.2 在工作表中使用ActiveX控件
 - 6.2.3 窗体控件和ActiveX控件的区别
 - 6.3 与用户交互，简单的输入输出对话框
 - 6.3.1 InputBox 函数
 - 6.3.2 Application对象的InputBox方法
 - 6.3.3 MsgBox函数
 - 6.3.4 Application对象的FindFile方法
 - 6.3.5 Application对象的GetOpenFilename方法
 - 6.3.6 Application对象的GetSaveAsFilename方法
 - 6.3.7 Application对象的FileDialog属性
 - 6.4 构建用户窗体，自己设计交互界面
 - 6.4.1 关于用户窗体
 - 6.4.2 添加一个用户窗体
 - 6.4.3 设置窗体的属性
 - 6.4.4 在窗体上添加控件
 - 6.4.5 显示窗体
 - 6.4.6 关闭窗口
 - 6.4.7 使用控件
 - 6.4.8 用键盘控制控件
 - 6.5 改造Excel现有的界面
 - 6.5.1 更改标题栏的程序名称
 - 6.5.2 显示或隐藏菜单栏
 - 6.5.3 显示或隐藏工具栏
 - 6.5.4 设置窗口
 - 6.5.5 其他设置
 - 6.6 典型的技巧或示例
 - 6.6.1 设计一张调查问卷
 - 6.6.2 职工信息管理界面
 - 6.6.3 一个简易的登录窗体

第7章 代码调试与优化

7.1 VBA中可能会发生的错误

7.1.1 编译错误

7.1.2 运行时错误

7.1.3 逻辑错误

7.2 VBA程序的3种状态

7.2.1 设计模式

7.2.2 运行模式

7.2.3 中断模式

7.3 Excel已经准备好的调试工具

7.3.1 让程序进入中断模式

7.3.2 为程序设置断点

7.3.3 使用Stop语句

7.3.4 使用立即窗口

7.3.5 使用本地窗口

7.3.6 使用监视窗口

7.4 错误处理的艺术

7.4.1 Go Error GoTo标签

7.4.2 On Error Resume Next

7.4.3 On Error GoTo 0

7.5 让代码跑得更快一些

7.5.1 合理地使用变量

7.5.2 避免反复引用相同的对象

7.5.3 尽量使用函数完成计算

7.5.4 去掉多余的激活和选择

7.5.5 合理使用数组

7.5.6 关闭屏幕更新

《别怕，Excel VBA其实很简单》

章节摘录

版权页： 插图：

专业书评：VBA，让效率飞起来当加班成为常态，改变在所难免十年前的我，加班是家常便饭的事，每天成堆的工作任务总是压得我喘不过气来，我和同事们就像一枚枚棋子蜷缩在办公室里，在电脑前紧张地忙碌着，不知目睹了多少个华灯初上到灯火阑珊的夜晚。2000年8月的一天，我的目标是完成生产成本核算系统的最后一个报表模块的开发。有了它，就可以方便地查询和计算每一种产成品在任意一个工序上的成本明细项目，还可以在不同月份之间进行结转、对比。我喜欢在晚间写程序，因为晚上安静，能让我思路飞扬，让代码随着键盘声快速的推进，等待大功告成的那一刻出现。你不会以为我是程序员吧？不，我是一名财务部的成本主管，就职于一家制造型企业。公司虽有IT部门，但是没有程序员。说白了，我只是千百万个成天和Excel表格耳鬓厮磨的普通一员。我们的产品有几十种，涉及到的材料有几千种，每个产品又有N道工序，每道工序由数量不等的作业人员进行生产。我的工作，就是计算和分析所有产成品和半成品的生产成本，包括材料、人工和杂费。计算依据主要包括生产部门提交的各产品工序的工时记录表，仓库提交的材料进销数据，HR部门提供的工资单明细。这样的计算任务并不轻松，且不论计算目标复杂，原始数据繁多，有些甚至不是电子档，而且只有我一个人，要命的是时间总是非常的紧张，因为每个月交报表的时间是固定的。也许你想问，这么复杂的计算用Excel？你们公司难道没有ERP吗？有的，而且我们公司的ERP声名不小，价格不菲。但在我看来，公司的ERP虽有一定的作用，不过却存在很多局限性，局部线条偏粗，又有些笨拙，很难给出我需要的结果。所以，我必须借助Excel。回想中学时政治课所学的知识，资本家为了榨取更多的剩余价值，有两种方法：一是延长工作时间，二是提高生产效率。我现在清楚地意识到，为了及时准确地完成计算任务，方法是同样的两个：要么加班，要么提高工作效率。我当然不愿意做无意义的加班。同时，这类加班的产出也是有限的，并不能解决任务重时间紧的根本性问题。所以，我必须提高效率。一次自动核算成本的开发经历，让我受益无穷自从接了公司的生产成本核算系统这个活，我的Excel水平突飞猛进。从最初的茫然无措到现在的从容应对，我通过不断优化计算方法，完善成本核算模板，大大减轻了工作量。有人说，学好Excel可以以一当十。年轻的我凭着一腔热情努力完成工作，即使公司的产品规模不断扩大，计算任务随之加重，我依然可以按时交报表。由于我舍得下力气去研究，直到后来协助工作的同事被上司调派去负责别的内容。在我看来，多做就是多学习，付出一定有回报。我的成本核算模板以产品作为区分，主要应用了Excel的函数公式和数据透视表，能够实现成本计算的半自动化，即通过输入原始数据，自动生成结果。原始数据，一部分来源于上个月的成本数据，一部分从ERP中导出。模板完善后，我的工作重心不再是计算，而是处理繁琐无聊的原始数据，千篇一律的执行导出、保存、打开、复制、粘贴、切换和关闭等动作，奈何我200APM的手速，因为涉及到几百个文件的数据处理，至少得一两天时间，处理过程中还很容易出错。让人抓狂的是公司的产品数量一直在增加，并处于变化中，这让我再次想起了“水管工的故事”。于是，我决定继续挖掘Excel的潜力，其实也是我自身的潜力。两个月以后，我用ExcelVBA代码替代了80%的成本原始数据处理工作。只要按一次键，数据就能乖乖的按规定的路线在几百个Excel文件之间流转，就像欢快的浪花在河水中荡漾。说真的，没有什么事情，比看着自己写的代码正常运行，而复杂无比的工作在瞬间灰飞烟灭的感觉更有成就感了。了解到ExcelVBA与众不同的威力后，我的激情再一次被点燃，我决定要自己写一个成本计算分析系统，我希望以后每个月的成本计算分析都是全自动的。经过持续不断地学习和研究，我想，今天晚上，终于可以达成这一目标了。透视VBA的知与行转眼间迈入2012年了。我们都生活在信息社会中，生活在一个前所未有的充斥着海量数据的年代。无论是企业还是个人，每天都要接触无数以数据为载体的信息。数据，甚至已经成为了企业或个人的替代品。不相信？一家你未曾亲身到访甚至未曾接触过其产品的企业，对你来说意味着什么？它无非会成为财务报表或统计报表上的一堆林林总总的数字，诸如生产规模、员工人数、利润水平……一个你未曾谋面未曾听说过的人，对你来说意味着什么？就好像进入婚恋网站搜索对象，这些陌生人只不过是个人指标数据的集合体，诸如身高、体重、职业、收入……想要在这样一个时代生存，处理数据的能力是必须的，因为实在有太多数据要处理了。广大Excel的用户，尤其是Excel的重度用户肯定对此深有体会。作为ExcelHome的站长和一名培训讲师，我接触过许多各式各样的数据处理要求，也体验或亲身参与过许多基于Excel的解决方案。这些宝贵经验让我对Excel提供的各项功能有更深入的理解。在Excel中制作计算模型，主力军非函数与公式莫属。300多个不同功能的函数在公式中灵活组合，可以创造无数种算法，再加上数组和名称的配合使用，几乎可以完成绝大多数计算任务。要论数据分析和报表生成

《别怕，Excel VBA其实很简单》

的便捷，不得不提到数据透视表，这是Excel最厉害的本领，厉害在于其功能强大的同时，使用起来极其简单。但如果只会这两样，仍然会有很多时候感到束手束脚，究其根本在于：1.函数和公式只能在其所在的位置返回结果，而无法操作数据表格的任意位置，更不能操作表格的任意属性（比如设置单元格的填充色，或者删除单元格）。2.函数和公式、数据透视表都需要规范的数据源，但往往我们工作量最大之处就在于获取和整理原始数据。比较麻烦的情况之一就是原始数据很可能是位于某个文件夹下的几十份表格。3.使用函数和公式、数据透视表制作的解决方案，难以具备良好的交互性能。因为它们只能存在于单元格中，与普通数据是处于同一个平面的。4.对于业务流程较为复杂、数据项经常变化的计算很难处理。5.无法迅速省力地完成大量的重复操作。所以，永远不要忘记Excel还有一个杀手级的功能--VBA。VBA是什么，怎么用，在本书中会给出详细的答案。这里，我只想说说，只有这个功能才真正让Excel成为了无所不能的数据处理利器，才让我们有机会可以彻底地高效办公。很多人认为VBA很神秘，认为会写代码是自己不可能实现的事情。虽然我不能保证人人都能学会VBA，但我可以保证如果你能学会函数和公式，你也能学会VBA，因为它们的本质是相同的。函数和公式无非是写在单元格中的一种简短代码罢了。所以，如果你曾经觉得自己连Excel函数和公式也搞不定，现在却能熟练地一口气写下好几个函数嵌套的公式，那么你学VBA不会有问题。在我眼里，VBA就好像“独孤九剑”剑法。这武功最大的特点是遇强则强，遇弱则弱。如果你每天面对的数据非常有限，计算要求也很简单，那么用VBA就是高射炮打蚊子了。但如果你是Excel重度用户，经常需要处理大量数据，而Excel现有功能无法高效完成计算任务时，就可以考虑让VBA上场，一举定乾坤。今时不同往日，互联网的发展使得技术和经验的分享非常方便。如果说十年前你想用VBA实现任何一个小功能都需要先掌握全部语法，然后一行一行代码自己写的话，那么现在ExcelHome上有太多太多现成的实现不同目标的VBA代码，许多代码甚至已经到了拿来即用的程度。所以，如果你的时间非常有限，也没有兴趣成为一个Excel开发者，你只需要快速的学习掌握ExcelVBA的基本语法，然后到互联网上去淘代码来用到自己的工作中。如果你投入的时间多一点点，你会发现你很快就能看懂别人的代码，然后做出简单的修改后为自己所用。这个过程，是不是和你当年学Excel函数和公式的经历很类似？等你做到这一步，你会发现原来公司里那个很厉害的会写代码的Excel高手的秘密武器原来是这样的啊，以后可以少请他吃饭以换取他为你写个小功能了……因为工作的关系，我接触过很多信息化工具，也了解过一些编程语言，我发现所有工具的本质是相通的。每种工具都有其优缺点，有其专属的场合。这种专属并非指不可替代，而是说最佳选择。因此，我不赞成VBA至上的观点，因为尽管VBA无所不能，但如果我们事事都写代码，那还要Excel本身的功能干嘛？我也反对VBA无用的观点，你暂时用不上怎么能说明此工具无用？甚至说，你根本就不会用这工具，怎么知道你用不上？用VBA，是为了更高效。不用，也是因为同样的目的。但是，会了VBA，你将拥有高效的更多种选择。不会，你就没有。这一点，高效人士都懂的。——ExcelHome创始人、站长：周庆麟

《别怕，Excel VBA其实很简单》

编辑推荐

《别怕，Excel VBA其实很简单》是Excel Home又一力作，由Excel Home网站创始人、站长，中国最资深的微软全球最有价值专家（MVP）Kevin，以及Excel Home官方微博管理员、Excel Home免费在线培训中心最受欢迎的讲师之一叶枫所著，传授一句顶一万句的“懒人心法”，教你一键搞定所有报表。

《别怕，Excel VBA其实很简单》内容浅显易懂，即便初学者没有编程基础也能看得懂；采用双色印刷，配有大量插画，采用轻松易懂的方式介绍VBA。VBA不是Excel的全部，却是成为Excel高手的必经之路，阅读《Excel VBA其实很简单》，体验恍然大悟的愉悦，发现节约时间的快乐。《别怕，Excel VBA其实很简单》的视频教程和相关代码可从Excel Home网站（<http://www.excelhome.net>）免费获取。

《别怕，Excel VBA其实很简单》

精彩短评

《别怕，Excel VBA其实很简单》

精彩书评

- 1、是的，真的是很简单 %74%2E%63%6E/RvKzqkV 兴趣也很重要，任何的学习都有一个过程，由浅入深，其实学习什么都一样，好的教程至关重要，好的教程就是一个好的老师，真的很简单，所有的操作都有例子，真的是手把手教你学习，学习了这个，已经很有基础了，再学习其它的可以很兴松。
- 2、刚总体翻了一遍，对vba有了一个总体认识。讲解了vba的几个最主要概念，语句。读完以后已经可以读懂现成的程序，并在原有基础上做了适应性的改造，应用到实际工作中去。应当说确实是一本带我们入门的好书，不枯燥，不乏味。准备在第一遍阅读和初步实践基础上再读一遍。希望每一遍都能有新的收获。vba最主要的还是要实践。。。。。。。
- 3、相当不错的一本书，Excel VBA入门用好，例子挺好，能调动下积极性。很能提起大家的兴趣，也算是任务驱动的吧，比填鸭式的教科书式的要好了不少。如果后者看不下去的话，可以试下这本，书中的例子能操练好会受益很多。书里面的会了，再去excelhome论坛去继续提升就行了。其实最主要在使用中，也有任务去驱动，是学得最快的。用KPW和安卓版kindle看的，图片跟代码看着有点小了。前后的目录就各占了6%了。
- 4、15年8月中旬于家中阅读。之前读过另一本教授VBA的书，读了一半，因回家的问题换了这一本，也算有了对比阅读吧。本书有两大特色。一大特色是排版配图清新，语言流畅，按道理说适合非程序员人群入门。另一大特色是作者将一些经验、注意点与辨析也一并教授给了读者，使读者可以在实践前绕过这些坑。本书后半部分就难免如其他书的桎梏了。依次介绍各个对象的主要属性和方法，配以实例。一时半会记不完，等有实践机会的时候再自然而然的吸收了。这本书是一本很不错的VBA入门书，感谢它带领我进入VBA的殿堂！

章节试读

- 1、《别怕，Excel VBA其实很简单》的笔记-第55页

```
Const p As Single=3.14
```

- 2、《别怕，Excel VBA其实很简单》的笔记-VBA中有哪些数据类型

- 3、《别怕，Excel VBA其实很简单》的笔记-第62页

声明动态数组的方式

```
Sub dtsz()
```

```
Dim arr() As String
```

```
Dim n As Long
```

```
'统计A列到底有多少个非空单元格
```

```
n = Application.WorksheetFunction.CountA(Range("a:a"))
```

```
ReDim arr(1 To n) As String
```

```
End Sub
```

- 4、《别怕，Excel VBA其实很简单》的笔记-第66页

```
Sub ArrToRng1()
```

```
Dim arr As Variant          '定义变量
```

```
arr = Array(1, 2, 3, 4, 5, 6, 7, 8, 9) '利用Array函数创建数组
```

```
'将数组批量写入单元格
```

```
Range("A1:A9").Value = Application.WorksheetFunction.Transpose(arr)
```

```
End Sub
```

一维数组写入垂直单元格用函数Transpose转换

```
Sub ArrToRng2()
```

```
Dim arr(1 To 2, 1 To 3) As String '定义一个2行3列的二维数组
```

```
arr(1, 1) = 1          '给数组的各个元素赋值
```

```
arr(1, 2) = "张勇"
```

```
arr(1, 3) = "男"
```

```
arr(2, 1) = 2
```

```
arr(2, 2) = "林梅"
```

```
arr(2, 3) = "女"
```

```
Range("A1:C2").Value = arr '将数组批量写入单元格区域
```

```
End Sub
```

- 5、《别怕，Excel VBA其实很简单》的笔记-第62页

《别怕，Excel VBA其实很简单》

```
sub arraytest ()
dim arr as variant
'将1到10十个自然数赋值给数组arr
arr = array(1,2,3,4,5,6,7,8,9,10)
msgbox "arr数组的第2个元素为： “ & arr(1)

end sub
```

自己尝试的内容灵感来自上一篇

```
sub sztest_2
dim arr (1 to 10) as integer, i as integer
for i = 1 to 10
arr(i) = i
next
end sub
```

这个里面省去了一一个个赋值的麻烦。

然后编辑的代码是这样的

```
Sub arraytestofme()

Dim arr As Variant
'将1到10个自然数赋值给数组 arr
redim arr(1 To 10) As Integer, redim i As Integer
For i = 1 To 10
arr(i) = i
MsgBox "arr数组的第2个元素为:" & arr(i)

End Sub
```

各种报错啊。。。

然后请教了别人后得出了这样的答案

```
Sub arraytestofme()
'将1到10个自然数赋值给数组 arr
Dim arr(1 To 10) As Integer, i As Integer
For i = 1 To 10
arr(i) = i
MsgBox "arr数组的第2个元素为:" & arr(i)
Next
End Sub
```


6、《别怕，Excel VBA其实很简单》的笔记-第65页

array

split (不清楚split和array比的好处在哪。)

range (可以直接赋值定义的区域)

ubound (最大索引号) Lbound (最小索引号)

join 函数 可以把一维数组里的元素使用指定的分隔符练成一个新的字符串

7、《别怕，Excel VBA其实很简单》的笔记-第58页

单个过程

在一个过程中使用Dim或Static语句声明的变量，作用域为本过程，即只有声明变量的语句所在的过程可以使用它。这样的变量成为本地变量

单个模块

在模块的第一个过程之前使用Dim或Private语句声明的变量，作用域为声明变量的语句所在模块里的所有过程，即该模块里所有的过程都可以使用它。这样的变量成为模块级变量

所有模块

在一个模块的第一个过程之前使用Public语句声明的变量，作用域为所有模块，即所有模块的过程都可以使用它。这样的变量称为公共变量

8、《别怕，Excel VBA其实很简单》的笔记-第52页

```
Dim str as string
```

```
str="一起来学VBA。"
```

给对象变量赋值

```
Dim rng as range
```

```
Set rng=Worksheets("sheet1").Range("A1")
```

```
rng.Value="欢迎来到excelhome论坛！"
```

日期赋值

```
d=#2011-7-1#
```

9、《别怕，Excel VBA其实很简单》的笔记-第123页

10、《别怕，Excel VBA其实很简单》的笔记-第66页

```
Sub JoinTest()
```

```
Dim arr As Variant, txt As String '定义两个变量
```

```
arr = Array(0, 1, 2, 3, 4, 5, 6, 7, 8, 9) '利用Array函数创建一个数组arr
```

```
txt = Join(arr, "@") '将arr数组的元素连成字符串，用@作分隔符
```

```
MsgBox txt '用对话框显示字符串
```

```
End Sub
```

《别怕，Excel VBA其实很简单》

11、《别怕，Excel VBA其实很简单》的笔记-第43页

12、《别怕，Excel VBA其实很简单》的笔记-第48页

强制声明所有变量

Option Explicit

13、《别怕，Excel VBA其实很简单》的笔记-第43页

Byte
Boolean
Integer
Long
Single
Double
Currency
Decimal
Date
String
Object
Variant
用户自定义

14、《别怕，Excel VBA其实很简单》的笔记-第57页

```
Dim 七6班(1 to 50) As String
Dim 七6班(50) As String
七6班(1)="张青"
```

多维数组

```
Dim 酱油(2,19) As String
```

/使用array创建数组

```
Sub ArrayTest()
    Dim arr As Variant      '定义变量
    '将1到10十个自然数赋给数组arr
    arr = Array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
    MsgBox "arr数组的第2个元素为：" & arr(1)
End Sub
```

/使用split创建数组

```
Sub SplitTest()
    Dim arr As Variant      '定义变量
    '利用split生成数组
    arr = Split("邓成,林梅,张青,孔丽,冯吉维,孔佳", ",")
    MsgBox "arr数组的第2个元素为：" & arr(1)
End Sub
```

《别怕，Excel VBA其实很简单》

/使用range创建数组

```
Sub RngArr()  
    Dim arr As Variant      '定义变量  
    arr = Range("A1:C3").Value  '将A1 : C3单元格的内容存储到数组arr里  
    Range("E1:G3").Value = arr  '将数组arr的数据写入E1:G3单元格区域  
End Sub
```

/使用UBound和LBound统计数组个数

```
Sub arrcount()  
    Dim arr(10 To 50)      '定义数组  
    MsgBox "数组的最大索引号是：" & UBound(arr) & Chr(13) _  
        & "数组的最小索引号是：" & LBound(arr) & Chr(13) _  
        & "数组的元素个数是：" & UBound(arr) - LBound(arr) + 1  
End Sub
```

```
Sub dwsz()  
    Dim arr(1 To 10, 1 To 100)  
    MsgBox "第一维的最大索引号是：" & UBound(arr, 1) & Chr(13) & _  
        "第二维的最大索引号是：" & UBound(arr, 2)  
End Sub
```

《别怕，Excel VBA其实很简单》

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com