

超简单：用Python让Excel飞起来（零基础学python，用python实现办公自动化，让excel化繁为简，减少重复工作）

作者：王秀文

超简单：用Python让Excel飞起来

王秀文 郭明鑫 王宇韬 编著

ISBN: 978-7-111-65976-1

本书纸版由机械工业出版社于2020年出版，电子版由华章分社（北京华章图文信息有限公司，北京奥维博世图书发行有限公司）全球范围内制作与发行。

版权所有，侵权必究

客服热线：+ 86-10-68995265

客服信箱：service@bbvip.com

官方网址：www.hzmedia.com.cn

新浪微博 @华章数媒

微信公众号 华章电子书（微信号：hzebook）

目录

前言

如何获取学习资源

第1章 Python快速上手

1.1 为什么要学习用Python控制Excel

1.2 Python编程环境的搭建

1.2.1 安装Python官方的编程环境IDLE

1.2.2 安装与配置Anaconda和PyCharm

1.3 Python的模块

1.3.1 初识模块

1.3.2 模块的安装

1.4 让Excel飞一下

第2章 Python的基础语法知识

2.1 变量

2.2 数据类型：数字与字符串

2.2.1 数字

2.2.2 字符串

2.2.3 数据类型的查询

2.2.4 数据类型的转换

2.3 数据类型：列表、字典、元组与集合

2.3.1 列表

2.3.2 字典

2.3.3 元组和集合

2.4 运算符

2.4.1 算术运算符和字符串运算符

2.4.2 比较运算符

2.4.3 赋值运算符

2.4.4 逻辑运算符

2.5 编码基本规范

2.5.1 缩进

2.5.2 注释

2.6 控制语句

2.6.1 if语句

2.6.2 for语句

2.6.3 while语句

2.6.4 控制语句的嵌套

2.7 函数

2.7.1 内置函数

2.7.2 自定义函数

第3章 Python模块

3.1 模块的导入

3.1.1 import语句导入法

3.1.2 from语句导入法

3.2 处理文件和文件夹的模块——os

3.2.1 获取当前运行的Python代码文件路径

3.2.2 列出指定路径下的文件夹包含的文件和子文件夹名称

3.2.3 分离文件主名和扩展名

3.2.4 重命名文件和文件夹

3.3 批量处理Excel文件的模块——xlwings

3.3.1 创建工作簿

3.3.2 保存工作簿

3.3.3 打开工作簿

3.3.4 操控工作表和单元格

3.4 数组计算的数学模块——NumPy

3.4.1 数组的基础知识

3.4.2 数组的创建

3.5 数据导入和整理模块——pandas

3.5.1 二维数据表格DataFrame的创建与索引的修改

3.5.2 文件的读取和写入

3.5.3 数据的选取和处理

3.5.4 数据表的拼接

3.6 数据可视化模块——Matplotlib

3.6.1 绘制折线图

3.6.2 绘制柱形图

3.7 模块的交互

3.7.1 xlwings模块与pandas模块的交互

3.7.2 xlwings模块与Matplotlib模块的交互

第4章 使用Python批量处理工作簿和工作表

案例01 批量新建并保存工作簿

举一反三 批量新建并关闭工作簿

案例02 批量打开一个文件夹下的所有工作簿

举一反三 列出文件夹下所有文件和子文件夹的名称

案例03 批量重命名一个工作簿中的所有工作表

举一反三 批量重命名一个工作簿中的部分工作表

案例04 批量重命名多个工作簿

举一反三 批量重命名多个工作簿中的同名工作表

案例05 在多个工作簿中批量新增工作表

举一反三 在多个工作簿中批量删除工作表

案例06 批量打印工作簿

举一反三 批量打印多个工作簿中的指定工作表

案例07 将一个工作簿的所有工作表批量复制到其他工作簿

举一反三 将某个工作表的数据批量复制到其他工作簿的指定工作表中

案例08 按条件将一个工作表拆分为多个工作簿

举一反三 按条件将一个工作表拆分为多个工作表

举一反三 将多个工作表拆分为多个工作簿

案例09 批量合并多个工作簿中的同名工作表

举一反三 将工作簿中名称有规律的工作表合并到一个工作表

第5章 使用Python批量处理

案例01 精确调整多个工作簿的行高和列宽

举一反三 精确调整一个工作簿中所有工作表的行高和列宽

案例02 批量更改多个工作簿的数据格式

举一反三 批量更改多个工作簿的外观格式

案例03 批量替换多个工作簿的行数据

举一反三 批量替换多个工作簿中的单元格数据

举一反三 批量修改多个工作簿中指定工作表的列数据

案例04 批量提取一个工作簿中所有工作表的特定数据

举一反三 批量提取一个工作簿中所有工作表的列数据

举一反三 在多个工作簿的指定工作表中批量追加行数据

案例05 对多个工作簿中指定工作表的数据进行分列

举一反三 批量合并多个工作簿中指定工作表的列数据

举一反三 将多个工作簿中指定工作表的列数据拆分为多行

案例06 批量提取一个工作簿中所有工作表的唯一值

举一反三 批量提取一个工作簿中所有工作表的唯一值并汇总

第6章 使用Python批量进行

案例01 批量升序排序一个工作簿中的所有工作表

举一反三 批量降序排序一个工作簿中的所有工作表

举一反三 批量排序多个工作簿中的数据

案例02 筛选一个工作簿中的所有工作表数据

举一反三 在一个工作簿中筛选单一类别数据

案例03 对多个工作簿中的工作表分别进行分类汇总

举一反三 批量分类汇总多个工作簿中的指定工作表

举一反三 将多个工作簿数据分类汇总到一个工作簿

案例04 对一个工作簿中的所有工作表分别求和

举一反三 对一个工作簿中的所有工作表分别求和并将求和结果写入固定单元格

案例05 批量统计工作簿的最大值和最小值

举一反三 批量统计一个工作簿中所有工作表的最大值和最小值

案例06 批量制作数据透视表

举一反三 为一个工作簿的所有工作表制作数据透视表

案例07 使用相关系数判断数据的相关性

举一反三 求单个变量和其他变量间的相关性

案例08 使用方差分析对比数据的差异

举一反三 绘制箱形图识别异常值

案例09 使用描述统计和直方图制定目标

举一反三 使用自定义区间绘制直方图

案例10 使用回归分析预测未来值

举一反三 使用回归方程计算预测值

第7章 使用Python制作简单的图表并设置图表元素

案例01 在Python中制作简单的图表

举一反三 在Python中制作柱形图

举一反三 在Python中制作条形图

举一反三 在Python中制作饼图

案例02 在Python中导入Excel数据制作简单的图表

举一反三 导入数据制作散点图

举一反三 导入数据制作面积图

案例03 在Python中制作组合图表

举一反三 制作双折线图

案例04 添加并设置图表标题和坐标轴标题

举一反三 添加图例

案例05 添加并设置数据标签

举一反三 设置y轴的取值范围

案例06 为组合图表添加并设置次坐标轴

举一反三 添加并设置网格线

第8章 使用Python制作常用图表

案例01 制作柱形图展示数据的对比关系

举一反三 批量制作条形图

案例02 制作折线图展示数据变化趋势

举一反三 制作折线图并为最高点添加数据标签

举一反三 制作平滑折线图

[案例03 制作散点图判断两组数据的相关性](#)

[举一反三 为散点图添加线性趋势线](#)

[举一反三 制作气泡图](#)

[案例04 制作饼图展示部分和总体的比例关系](#)

[举一反三 制作圆环图](#)

[案例05 制作雷达图对比多项指标](#)

[举一反三 制作某一品牌性能评价指标雷达图](#)

[案例06 制作温度计图展示工作进度](#)

[举一反三 制作上半年销售业绩的温度计图](#)

[第9章 在Excel中调用Python代码](#)

[9.1 在工作表中调用Python自定义函数](#)

[9.1.1 在Excel中加载xlwings插件](#)

[9.1.2 导入并调用Python自定义函数](#)

[9.2 在VBA中调用Python自定义函数](#)

[9.2.1 通过命令创建文件并调用Python自定义函数](#)

[9.2.2 手动创建文件并调用Python自定义函数](#)

[9.2.3 VBA代码和Python代码的混合使用](#)

[9.3 将Python代码转换为可执行程序](#)

[9.3.1 PyInstaller模块的语法和参数含义](#)

[9.3.2 将Python代码打包成可执行程序](#)

[9.3.3 打包文件的实际应用](#)

前言

Excel作为当今最流行的办公软件之一，在数据编辑、处理和分析方面的表现都很出色。但是许多办公人员会发现，即便有了Excel的帮助，重复性、机械性的事务仍然要花费大量时间，而且如果要处理的数据体量较大，连Excel都变得有些力不从心了。那么有没有办法弥补Excel的这些“短板”呢？本书给出的答案是：用Python为Excel插上飞翔的翅膀。

可能有人会说，Python不是专门供程序员编程使用的吗？对于没有编程基础的普通办公人员来说会不会太难学了？其实这样的担心是多余的。Python的语法简洁易懂，因而很容易上手。更重要的是，学习Python能带给我们巨大的回报：用Python编程操控Excel，不仅能又快又好地完成机械性、重复性的枯燥工作，而且能借助各种功能强大的第三方模块，将大数据分析、机器学习等先进的数据科学工具以“平易近人”的方式应用到日常办公当中，提高工作的“含金量”。

本书就是一本讲解如何用Python和Excel“强强联手”打造办公利器的案例型教程。全书共9章。第1~3章主要讲解Python编程环境的搭建、Python的基础语法知识、模块的安装和导入、常用模块的基本用法等内容，为后面的案例应用打下坚实的基础。第4~8章通过大量典型案例讲解如何用Python编程操控Excel，实现数据整理、数据分析、数据可视化等工作的自动化和批量化处理。第9章主要讲解如何在Excel中调用Python代码，进一步拓宽办公自动化的应用范围。

本书采用生动的情景对话方式引入案例，代码附有详细、易懂的注解，能有效帮助读者快速理解代码的适用范围及编写思路，并通过“举一反三”栏目对案例的应用场景进行扩展和延伸，引导读者开拓思路，从机械地套用代码进阶到随机应变地修改代码，独立解决更多实际问题。

本书适合有一定Excel基础又想进一步提高工作效率的办公人员，如从事文秘、行政、人事、营销、财务等职业的人士阅读，也可供Python编程爱好者参考。

由于编者水平有限，本书难免有不足之处，恳请广大读者批评指正。读者除了可扫描二维码关注公众号获取资讯以外，也可加入QQ群711374122与我们交流。

编者

2020年5月

如何获取学习资源

一 扫描关注微信公众号

在手机微信的“发现”页面中点击“扫一扫”功能，进入“二维码/条码”界面，将手机摄像头对准下图中的二维码，扫描识别后进入“详细资料”页面，点击“关注公众号”按钮，关注我们的微信公众号。

□

二 获取学习资源下载地址和提取密码

点击公众号主页面左下角的小键盘图标，进入输入状态，在输入框中输入6位数字“200610”，点击“发送”按钮，即可获取本书学习资源的下载地址和提取密码，如下图所示。

□

三 打开学习资源下载页面

在计算机的网页浏览器地址栏中输入前面获取的下载地址（输入时注意区分大小写），如下图所示，按【Enter】键即可打开学习资源下载页面。

□

四 输入密码并下载文件

在学习资源下载页面的“请输入提取密码”文本框中输入前面获取的提取密码（输入时注意区分大小写），再单击“提取文件”按钮。在新页面中单击打开资源文件夹，在要下载的文件名后单击“下载”按钮，即可将其下载到计算机中。如果页面中提示选择“高速下载”或“普通下载”，请选择“普通下载”。下载的文件如果是压缩包，可使用7-Zip、WinRAR等软件解压。

提示：读者在下载和使用学习资源的过程中如果遇到自己解决不了的问题，请加入QQ群711374122，下载群文件中的详细说明，或向群管理员寻求帮助。

第1章 Python快速上手

要想借助Python让Excel更高效地工作，首先要搭建Python的编程环境，然后要安装与Excel相关的Python第三方模块。本章就将详细讲解这些知识，并在最后带领大家编写一个小程序，让大家实际感受一下Python是如何让Excel飞起来的。

1.1 为什么要学习用Python控制Excel

众所周知，Excel拥有直观的工作界面、出色的数据处理和计算功能以及丰富的图表工具，这些优势使得Excel在办公和商务领域有着广泛的应用。而且在Excel中通过VBA编程能让Excel更好地实现重复性工作的自动化和批量化处理，那么我们为什么还要舍近求远地学习用Python控制Excel呢？原因主要有以下几个方面：

- Python简单易学。相比Python而言，VBA的语法更加复杂和冗长，理解起来也更加困难，而Python的语法知识简单，代码也很简洁，所以对初学者来说Python更容易学习。

- 用VBA对当前Excel工作簿中的内容进行操作会比较方便，但对多个工作簿或不同格式文件的控制就要比Python复杂，如批量修改某一文件夹下的Excel工作簿名，用Python会方便很多。因此，使用Python能让自动化办公的实现范围更广、过程更轻松。

- 当数据量很大时，Python的处理速度更快。

- Python拥有一个丰富的模块库，用户通过编写简单的代码就能直接调用这些模块实现复杂的功能，快速解决实际工作中的问题，而无须自己从头开始编写复杂的代码。简单来说就是能“拿来就用”，这也是Python最大的一个魅力。

1.2 Python编程环境的搭建

俗话说得好：“工欲善其事，必先利其器。”要将Python与Excel结合使用，在安装好Excel的基础上还需要在计算机中搭建一个Python的编程环境，这样才能编写和运行Python代码。本节将介绍两种搭建Python编程环境的方法。

1.2.1 安装Python官方的编程环境IDLE

IDLE是Python的官方安装包中自带的一个集成开发与学习环境，它可以创建、运行和调试Python程序。对初学者来说，IDLE无须进行烦琐的配置，使用起来非常简单和方便。Python的官方安装包按照适用的操作系统分为多种类型，因此，在安装前要先清楚自己的计算机上运行的操作系统是哪种类型，再下载对应的安装包。

以Windows操作系统为例，单击桌面左下角的“开始”按钮，在打开的“开始”菜单中单击“Windows系统”，在展开的列表中单击“控制面板”选项，如下图所示。

□

在打开的“控制面板”窗口中单击“系统”图标，可以看到当前操作系统为Windows 10，“系统类型”为64位操作系统，如下图所示。

□

了解操作系统的信息后，就可以去Python的官网下载Python安装包了。打开浏览器，在地址栏中输入网址“<https://www.python.org>”，按【Enter】键，进入Python官网，单击“Downloads”按钮，在展开的列表中可看到多个系统类型，此处选择“Windows”，如下图所示。

□

进入下载安装包的页面，可看到Python的两个安装版本及各个版本下的多个安装包。此处以Python 3.7.4版本为例，介绍下载Python安装包的方法。因为前面查看到的操作系统类型是64位的Windows，所以在Python 3.7.4版本下单击“Download Windows x86-64 executable installer”链接。如果操作系统类型为32位，则单击“Download Windows x86 executable installer”链接。单击链接后，在页面下方弹出的下载提示框中单击“保存”按钮，如下图所示，即可开始下载Python的安装包。

□

等Python安装包下载完成后，进入安装包的下载位置，可看到如下左图所示的以“.exe”为扩展名的安装包文件。

双击下载好的Python安装包（也可以在步骤04中直接单击下载提示框中的“运行”按钮），在打开的程序安装窗口中勾选“Add Python 3.7 to PATH”复选框。如果要将程序安装在C盘的默认路径下，直接单击“Install Now（现在安装）”按钮。如果想要改变安装路径，可单击“Customize installation（自定义安装）”按钮，如下右图所示。

□

跳转到下一个安装界面，不用更改任何设置，直接单击右下角的“Next”按钮。继续跳转到另一个界面，在界面中可单击“Browse（浏览）”按钮，在打开的对话框中设置自定义安装路径，也可以直接在文本框中输入自定义的安装路径。然后单击“Install（安装）”按钮，如下图所示，即可看到Python的安装进度。

□

等待一段时间，如果看到“Setup was success-ful”的提示文字，说明Python已安装成功，单击“Close”按钮，如下图所示，完成安装。

□

完成安装后就可以启动IDLE。单击桌面左下角的“开始”按钮，在打开的“开始”菜单中单击“Python 3.7”文件夹，在展开的列表中单击“IDLE（Python 3.7 64-bit）”选项，如下图所示。

□

随后会打开一个名为“Python 3.7.4 Shell”的窗口，如下图所示。在该窗口中有一个“>>>”符号，称为提示符，在提示符的后面就可以输入代码。需要注意的是，在输入代码时一定要将输入法切换至英文模式。

□

欢迎访问：电子书学习和下载网站 (<https://www.shgis.cn>)
文档名称：《超简单：用Python让Excel飞起来》王秀文.epub
请登录 <https://shgis.cn/post/1862.html> 下载完整文档。
手机端请扫码查看：

