

# Design and Implementation of Office Automation Software Based on Python

Xiaojie Chen

Zhanjiang Institute of Science and Technology Zhanjiang 523000, Guangdong, China

**Abstract:** The development of the technological era, artificial intelligence is more and more popular among the public. Not only students need to learn the software, but also by the office workers love and embrace. python need to be designed before the use of their own desired module. It is more important to familiarize with the design of the software, each step of the inadmissible errors, the software's prudence and the variety of modes is the cornerstone of the development.

**Keywords:** Python, Implementation, Office automation.

## 1. Introduction

Since the workload is increasing, offices are understaffed and inefficient. Need to quote efficient and accurate software Python, non-professional people can also use. Simple and clear way of fast processing, organizing and saving of documents. On the computer, the design and implementation for the public easy to use, recognized.

## 2. Software Design and Implementation

### 2.1. Realization of batch file name creation function

In the actual work process, people often need to work according to the needs of the new folder and rename the operation. It seems to be a simple operation, however, once the need to create a batch of folder naming time, it will feel boring and heavy workload. os module and xlrd module of Python interactive use of Python, can help people, the creation of a batch of folder creation and naming, greatly

improving the efficiency of the work, reduce the workload. Now the functional design of batch folder name creation is carried out, the program flow is shown below.

#### 2.1.1. Creating an Excel workbook

Python's xlrd module can read an Excel workbook through the built-in functions open\_workbook and sheet\_by\_name to get the cell contents needed for the workbook.

When using the xlrd module in Python to create an Excel workbook, use the 'xls' format. This is because the xlrd module in Python is only compatible with the 'xls' format, not the 'xlsx' format. Once the Excel workbook is created using 'xlsx' format, the program will report an error.

The contents of the formulated Excel workbook format, as shown in Table 1. The need to batch build the file name naming, fill in the corresponding cell column, you can through Python's xlrd module built-in function open\_workbook method and function sheet\_by\_name method, to get the folder named one by one. Then you can create folders one by one by using the mkdir method built in Python's os module.

Table 1. Batch Creating Folder Naming

serial number	Batch folder naming	Remarks

#### 2.1.2. Importing Module Libraries

To implement the batch folder name creation function, you need to use the Python os module and xlrd module. PyCharm development software tool is a development environment designed by programmers for programmers, which provides you with all the tools you need for efficient Python development. Here we use the PyCharm development software tools to implement the batch folder name creation feature.

Here we import the Python os and xlrd modules as shown in Figure 1.

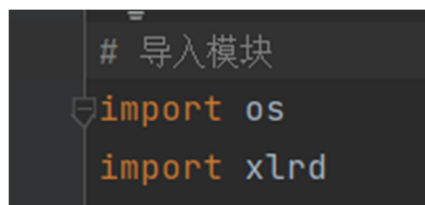


Figure 1. os module and xlrd module importing

#### 2.1.3. Read Excel workbook

Place the created Excel workbook "Batch Create Folder Naming" in the same folder with the programming code, the default search path of PyCharm development software tool is to automatically search the same folder for the Excel workbook that meets the requirements.

PyCharm development software tools through the default search, search for the production of Excel workbook "batch

create folder named", and then through the Python xlrd module built-in function open\_workbook method, get the specified Excel workbook. Then use the xlrd module built-in function sheet\_by\_name method to read the Excel worksheet Sheet1, as shown in Figure 2.

```
# 读取Excel工作簿
wb = xlrd.open_workbook('批量创建文件夹命名.xls')
# 获取Excel工作表
ws = wb.sheet_by_name('Sheet1')
```

Figure 2. Function open\_workbook method and function sheet\_by\_name method

### 2.1.4. Get Folder Naming

This process, into reading the Excel workbook "batch create folder named" within the Sheet table cell content link. First of all, you need to get the cell contents of the largest line, the purpose is to cycle through the corresponding folder named rows of columns, one by one to get the folder named, and then added to the Name\_List list, as shown in Figure 3

```
# 获取最大行
maxrows = ws.nrows
# 建立空列表
Name_list = []
# 逐一提取
n = 1
while n <= ws.nrows:
    # 逐一获取【批量创建文件夹命名】列
    c = ws.cell_value(n, 1)
    # 逐一添加到空列表内
    Name_list.append(c)
    # 每循环一次，系数增加一次
    n += 1
```

Figure 3. Get batch folder naming

### 2.1.5. Creating Folders

Python's os module creates folders in the current directory using the built-in mkdir method. Utilizing the mkdir function, coupled with Python's for loop statement, the final result is to create a batch of file names.

In 3.1.4, the folder names in the Excel worksheet have been added to the Name\_List list as strings, and Python's for loop statement can be used to traverse the Name\_List list one by one to create bulk folder names.

When iterating through the bulk folder names in the Name\_List list, there are empty strings, so in order to avoid creating filenames without names, which can cause errors. Here Python's If conditional statement is needed to isolate the empty strings in the list and create only non-empty strings for folder naming. As shown in Figure 4.

```
# 遍历列表，创建文件夹
for Folder_Name in Name_list:
    # 去除列表中的空字符串
    if Folder_Name != '':
        # 创建文件夹
        os.mkdir(f'{Folder_Name}')
```

Figure 4. Batch Creation of Folders

## 2.2. The realization of batch merge Excel workbook function

In reality, people sometimes according to the functional requirements of the workplace, the need to summarize the statistical information of all parties. At this time, it is necessary to develop an Excel form template, the object or department required to meet the requirements to fill out the Excel form template, and then unified collection and summarization.

In the face of summarizing the return of the batch of Excel workbook, the usual way of working is to rely on manual methods, manually open each copy of the Excel workbook, and then inside the cell content, copy and paste to the total Excel summary table inside.

This manual manual method, in the collection of a small number of Excel table information, can be manually operated, often within half an hour can be completed. However, when the collection of Excel table information, hundreds of thousands of times, just rely on manual methods, will be time-consuming and laborious, as a last resort, but also affect the rest time.

In order to reduce the pressure of dealing with batch Excel table information, to realize the batch merge Excel workbook function, the program flow is shown below.

### 2.2.1. Importing Module Library

To realize the function of batch merge Excel workbook, we need to use Python's os module, Python's xlrd module and Python's xlwt module.

Here we import Python's os module, xlrd module and xlwt module, as shown in Figure 5.

```
# 导入模块
import os
import xlrd
import xlwt
```

Figure 5. Import of Python's os module, xlrd module and xlwt module

### 2.2.2. Traversing the Excel workbook

The collected and summarized Excel workbooks are put into the same folder as the programming code, and the default search path of the PyCharm development software tool is to automatically search the same folder for the Excel workbooks that meet the requirements.

Through Python's os module built-in function listdir method and the For loop statement, to obtain the current folder, all the Excel workbook, and establish the Excel workbook list List\_Excel.

Using Python's xlrd module built-in function open\_workbook method and function sheet\_by\_name method, from the Excel workbook list List\_Excel, read the Excel workbook one by one, and get the cell contents of the Sheet in the Excel workbook, as shown in Figure 6.

```

# 获取当前目录下, 各Excel工作簿并建立列表
List_Excel = [i for i in os.listdir()]
# 逐一获取Excel工作簿列表, 读取
for Excel in List_Excel:
    # 读取每份Excel工作簿
    wb = xlrd.open_workbook(f'{Excel}')
    # 获取每份Excel工作簿内单元格内容
    nwb = wb.sheet_by_name('Sheet1')

```

Figure 6. Batch Excel Workbook Open, Read

### 2.2.3. Extract Excel worksheet cell content

The purpose of the batch merge Excel workbook, in order to Excel worksheet Sheet, the corresponding cell content, in accordance with the requirements of the prescribed template, to be summarized and integrated, and then filled into the specified cells.

The use of Python's dictionary method, the corresponding columns of the Excel worksheet cell content, in the form of a Python dictionary key value to save. In this way, even if you need to add more columns, you do not need to significantly modify.

Python dictionary value value, generally with the role of keywords to distinguish, for example, ID, name, student number, work number and other words, in order to avoid duplication of the value value, to cover the original Python dictionary key value, as shown in Figure 7.

```

r = 0
# 建立空字典
dict = {}
# 遍历
while r < nwb.nrows - 1:
    r += 1
    # 获取Excel表格每一行值
    l = nwb.row_values(r)
    # 提取需要的行信息, 创建字典的键和值
    if (l[0], l[2], l[3]) in dict.keys():
        dict[l[0], l[2], l[3]] += l[4]
    else:
        dict[l[0], l[2], l[3]] = l[4]

```

Figure 7. Extract Excel worksheet cell content to create a dictionary

### 2.2.4. Cell Content Writing and Saving

In 3.2.3 above, the key and value values of the Python dictionary have been obtained. The key and value values of the Python dictionary can be traversed one by one from the items value of the Python dictionary by using python's For loop statement.

Python's xlwt module to create Excel workbooks and write to specified cell contents. Through Python's xlwt module built-in function Workbook method and function add sheet method to create Excel workbooks and worksheets, as shown in Figure 8.

```

# 创建Excel工作簿
wt = xlwt.Workbook(encoding='uft-8')
# 建立Excel工作表
nws = wt.add_sheet('Sheet')

```

Figure 8. Creating Excel workbooks and worksheets

After creating Excel workbooks and worksheets, one is able to realize the operation of writing. People can according to different job requirements, different job nature requirements, different Excel sheet template requirements, through Python's xlwt module built-in function write method, fill in the information into the corresponding Excel table to complete the task.

Finally, after making sure that the filled Excel sheet is correct, use the built-in function save of Python's xlwt module to save it to a local directory.

## 2.3. Batch PDF copy to move to the specified folder function implementation

PDF is a portable document file format, this file format and operating system platforms are compatible. This feature makes it in the Internet for electronic document distribution, information technology, digital dissemination, data document preservation and other ideal document format.

More and more e-books, product descriptions, company notices, network information, e-mail in the beginning of the use of PDF format files, organizing, summarizing, classifying, archiving and other operations have become increasingly frequent.

Batch PDF copy and move to the specified folder function, you can to some extent alleviate the problem of archiving PDF format files, the program process is shown below.

### 2.3.1. Module import

Batch PDF copy to move to the specified folder function of the realization of the use of Python's module library has os module, shutil module and pdfplumber module three, as shown in Figure 9.

```

# 导入模块
import os
import shutil
import pdfplumber

```

Figure 9. Module Import

### 2.3.2. Traversing the current folder PDF

Python's os module's built-in function walk method is an easy-to-use file, a directory traverser that can quickly and efficiently search for things in terms of files and directories. Get the path to a folder with the walk method, a built-in function of Python's os module.

Python's os module built-in function scandir method, return lightweight DirEntry objects, these objects are very useful to the file information, the use of the file's information, the ability to filter PDF files, as shown in Figure 10.

```

# 获取当前文件夹位置
r = os.getcwd()
# 使用os.walk将 下面所有的文件夹都找到
for 文件夹路径, 子文件夹列表, 文件列表 in os.walk(f'{r}'):
    # print(子文件夹列表)
    # print(文件列表)
    # 遍历所有文件夹中的所有文件
    for 文件 in os.scandir(文件夹路径):
        # 将所有 pdf 类型的文件
        if 文件.name.endswith(".pdf"):

```

Figure 10. Search all PDF types

### 2.3.3. Iterate through each PDF file

Python's os module has a built-in function, walk, which can get the path to a folder, a list of subfolders, and a list of files. Combined with the above 3.3.2 of Python's os module built-in function scandir method, you can get each PDF file of the original PDF file corresponding path.

Python's pdfplumber module built-in function open method, in the original file corresponding to the path, it will be able to open the background way to read. Then use the built-in function pages method and extract\_text method, line by line to read the text content, as shown in Figure 11.

```
with pdfplumber.open(对应路径) as pdf:
    for 页码 in pdf.pages:
        文本 = 页码.extract_text() # 提取文本
```

Figure 11. Read PDF text content

### 2.3.4. Search PDF text keywords

Read PDF text content, according to the actual needs of the ability to ID card, name, student number, work number and other words as a classification index items.

PDF text index, you can use Python's re module, that is, regular expressions. Regular expression is a small, highly specialized programming language that can efficiently and concisely match strings. It is recommended to use Python's re module's built-in function findall method, which iterates through the matches and can get all the matches in a string, returning a list. This is shown in Figure 12.

```
with pdfplumber.open(对应路径) as pdf:
    for 页码 in pdf.pages:
        文本 = 页码.extract_text() # 提取文本
```

Figure 12. Search PDF text words

### 2.3.5. Copy and move PDF to the specified folder

Through the above 3.3.4, search to meet the requirements of the PDF text words. In order to file migration to the specified folder, you can use Python's shutil module built-in function copy method to achieve the effect of copy migration function.

## 3. Summary of the Chapter

Based on the basis of PyCharm development tools, the use of Python programming language different module library. We have realized the batch creation of folder name function, batch merge Excel workbook function and batch PDF copy move to the specified folder function and other software

design solutions.

Based on PyCharm development tools is the advantage of being able to adapt to local conditions, according to the actual needs, just in the original code has been written, a slight modification, you can make the software design to meet the new work requirements.

## 3.1. Conclusion

The advent of the era of artificial intelligence, so that the Python programming language began to receive people's attention, and even the state also advocates that primary and secondary schools should carry out Python instruction education. Coupled with the growing maturity of PyCharm development tools, Python programming is becoming increasingly simpler and more accessible to people.

We hope that through this research, we can make people pay attention to the achievements of Python language in office automation, and bring people a new way of working.

## Acknowledgment

Fund Projects: Ministry of Education University-Industry Cooperation Collaborative Education Project (202101301019、220602362291702); Zhanjiang Institute of Science and Technology Brand Enhancement Plan (PPJHKCSZ-2022278、PPJH2021009、PPJHYLKC-2022257).

About the author:Chen Xiaojie, Master, Teacher of Zhanjiang University of Science and Technology. Research interests: educational management, business administration.

## References

- [1] Zhang Yanling. Design and implementation of enterprise office automation system [D]. North China University of Science and Technology,2017.
- [2] Jian Zhang,Dan Zhao,Yiji Liu,Manli Bao. Design and Implementation of Chicken Run Game Based on Python[J]. Computer Programming Skills and Maintenance,2022(07):150-152.DOI:10.16184/j.cnki.comprg.2022.07.054.
- [3] Luo Ya. The impact of office automation on the management of paper files and the integration strategy[J]. Lantai inside and outside,2022(21):35-37.
- [4] Huang Haibin. Design and realization of office automation system in universities[J]. Digital Technology and Application,2022,40(03):168-170.DOI:10.19695/j.cnki.cn12-1369.2022.03.53.
- [5] Sun HL. Discussion on the promotion and application of office automation system in colleges and universities[J]. Industry and Technology Forum,2022,21(01):204-205.