

別再問我Python打包成exe了！（終極版）

接地氣學堂 今天

以下文章來源於凹凸數據，作者朱小五555



凹凸數據

一個不務正業的數據人！爬蟲、數據分析、可視化、方法論，一條龍服務！業務範圍：Python、SQL、Excel、Tableau……



凹凸数据（ID：alltodata）



大家好，我是小五🐱

《老闆又出難題，氣得我寫了個自動化軟件》

上次這篇文章中，評論區有好幾條留言都是關心如何將python腳本打包成10多M的？

那今天小五就給大家全面總結一下：Python如何打包成exe，以及如何打得足夠小。



目前比較常見的打包exe方法都是通過 `Pyinstaller` 來實現的，本文也將使用這種常規方法。如果對這塊已經很熟悉的小伙伴，可以直接下滑到本文下半部分。

為什麼要打包？

眾所周知，Python腳本不能在沒有安裝Python的機器上運行。

那我們如果寫了一個數據分析/自動化辦公的小腳本，想分享給同事小姐姐使用，可她電腦又沒有裝Python。

這個時候如果將腳本打包成exe文件，微信發送給她，即使她的電腦上沒有安裝Python解釋器，這個exe程序也能在上面運行。豈不美哉？



（當然，想通過幫安裝Python跟小姐姐建立感情的話，就當我沒說）

安裝Pyinstaller

首先我們要先安裝Pyinstaller，直接在cmd使用pip命令

```
pip install pyinstaller
```

如果網速太慢可以切換國內源來加速，本文其他部分直接使用國內源，不再重複註釋。

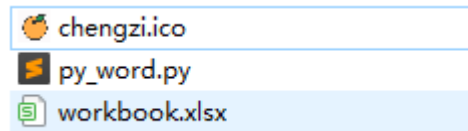
```
pip install -i https://pypi.tuna.tsinghua.edu.cn/simple  
tuna.tsinghua.edu.cn/simple  
singhua.edu.cn/packages/79/87/8bb36bd4ebae147612c73d1  
nd64.whl (8.7 MB)  
| 3.4 MB 1.7 MB/s eta 0:00:04_
```

```
pip install -i https://pypi.douban.com/simple/ pyinstaller #豆瓣源  
pip install -i https://pypi.tuna.tsinghua.edu.cn/simple pyinstaller #清华源
```

Pyinstaller打包步驟

這裡我們拿之前《[Python自動化辦公|同事要我幫忙補寫178份Word日報！](#)》這篇的python代碼，作為案例來演示。將其中腳本

`py_word.py`，待處理的表格文件 `workbook.xlsx`，以及準備好的軟件圖標圖片 `chengzi.ico` 放在了我電腦的 `F:\py_word` 目錄下（如果大家感興趣的話，可以在文末下載獲取）

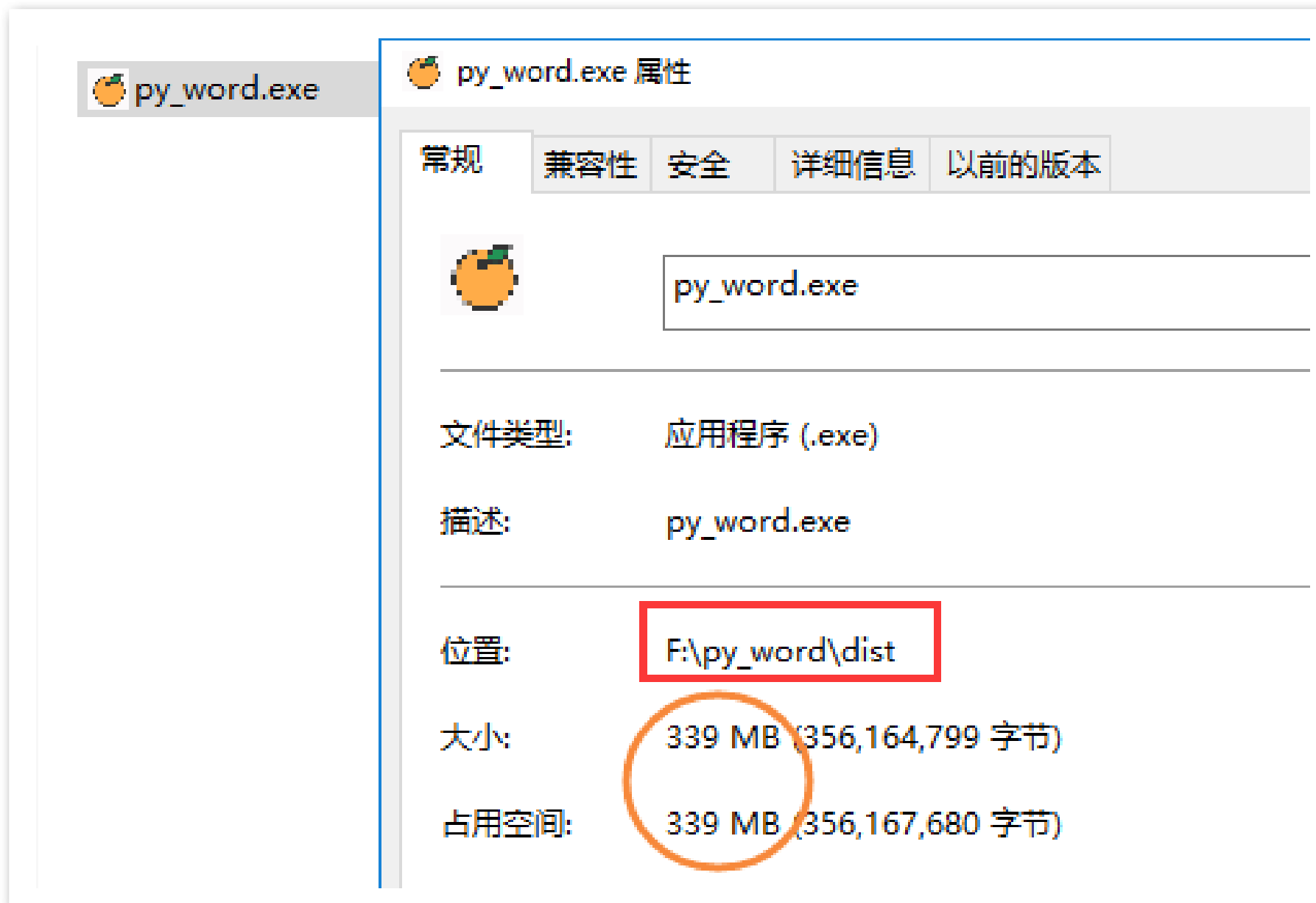


1、cmd切換到我們剛剛放文件的目錄

```
管理员: Anaconda Powershell Prompt (anaconda3)  
(base) PS C:\Users\Administrator> cd F:\py_word  
(base) PS F:\py_word>
```

2、執行命令 `Pyinstaller -F -w -i chengzi.ico py_word.py`，執行過程特別漫長，就沒有錄製動圖。

執行完畢會發現當前目錄多了幾個文件夾，打開其中名為dist的文件夾。



已經生成了一個名為 `py_word` 的exe應用程序，並且圖標也是我們設置的橙子圖案，看來已經成功一半了。

要不運行一下，試試能否正常生成word日報？

名称	修改日期	类型	大小
 __pycache__	2021/1/12/周二 ...	文件夹	
 build	2021/1/12/周二 ...	文件夹	
 dist	2021/1/12/周二 ...	文件夹	
 chengzi.ico	2021/1/9/周六 2...	看图王 ICO 图片...	1 KB
 py_word.exe	2021/1/12/周二 ...	应用程序	347,818 KB
 py_word.py	2021/1/10/周日 ...	PY 文件	9 KB
 py_word.spec	2021/1/12/周二 ...	SPEC 文件	1 KB
 workbook.xlsx	2020/9/20/周日 ...	XLSX 工作表	13 KB

成功運行，可惜就是文件有點大（339M）



Pyinstaller參數詳解

說回剛剛執行的命令

```
Pyinstaller -F -w -i chengzi.ico py_word.py
```

給大家解釋一下其中Pyinstaller的參數，其中 `-F` 參數代表製作獨立的可執行程序。

`-w` 是指程序啟動的時候不會打開命令行。如果不加`-w`的參數，就會有黑洞洞的控制台窗口出來。比如在剛才的腳本里我加一行 `print('Hello World!')`，那麼就不要放`-w`參數了，不然運行會報錯，畢竟 `Hello World!` 需要在命令行里打印出來。此外，`-w` 參數在GUI界面時非常有用。

最後的 `-i chengzi.ico` 就是指設置自己的圖標圖案，因為默認打包圖片是下圖這樣的。這個參數也可以寫成 `--icon=chengzi.ico`



最後稍微總結一下：

```
Pyinstaller -F py_word.py 打包exe
```

```
Pyinstaller -F -w py_word.py 不帶控制台的打包
```

```
Pyinstaller -F -w -i chengzi.ico py_word.py 打包指定exe图标打包
```

以上三個是比較常用的參數，其他參數詳見下表

-F, -onefile	产生一个文件用于部署 (参见XXXXX).
-D, -onedir	产生一个目录用于部署 (默认)
-K, -tk	在部署时包含 TCL/TK
-a, -ascii	不包含编码. 在支持Unicode的python版本上默认包含所有的编码.
-d, -debug	产生debug版本的可执行文件
-w, -windowed, -noconsole	使用Windows子系统执行. 当程序启动的时候不会打开命令行(只对Windows有效)
-c, -nowindowed, -console	使用控制台子系统执行(默认)(只对Windows有效)
-s, -strip	可执行文件和共享库将run through strip. 注意Cygwin的strip往往使普通的win32 DLL无法使用.
-X, -upx	如果有UPX安装(执行Configure.py时检测), 会压缩执行文件(Windows系统中的DLL也会)(参见note)
-o DIR, -out=DIR	指定spec文件的生成目录, 如果没有指定, 而且当前目录是PyInstaller的根目录, 会自动创建一个用于输出(spec和生成的可执行文件)的目录. 如果没有指定, 而当前目录不是PyInstaller的根目录, 则会输出到当前的目录下.
-p DIR, -path=DIR	设置导入路径(和使用PYTHONPATH效果相似). 可以用路径分割符(Windows使用分号, Linux使用冒号)分割, 指定多个目录. 也可以使用多个-p参数来设置多个导入路径
-icon=<FILE.ICO>	将file.ico添加为可执行文件的资源(只对Windows系统有效)
-icon=<FILE.EXE,N>	将file.exe的第n个图标添加为可执行文件的资源(只对Windows系统有效)
-v FILE, - version=FILE	将verfile作为可执行文件的版本资源(只对Windows系统有效)
-n NAME, - name=NAME	可选的项目(产生的spec的)名字. 如果省略, 第一个脚本的主文件名将作为spec的名字

Pyinstaller参数大全

自己做的軟件都喜歡放上自己的圖標，不過哪來那麼多ico圖片呢？

一個是可以找專門的ico圖片網站，不過都很小眾，圖片庫也很小。

另一個是可以自己生成，這裡就給大家分享一個網站，可以把其他格式圖片轉成ico格式：<https://app.xunjiepdf.com/img2icon/>



ico圖片格式轉換



好了，小伙伴們

來到了最激動人心的時刻，剛剛生成的exe實在太大了，300多M的軟件程序想用微信傳一下都費勁。

我也試過很多方法，比如：修改spec文件自定義打包、pipenv 虛擬環境、使用開源的upx壓縮等等，但是往往要么過程比較麻煩，要么成功率不高（壓縮成不成功全看臉）。

而我要分享的，是自己一直在用的，最簡單且成功率極高的方法——conda創建虛擬環境。



Python打包為什麼大？

在壓縮打包之前，先簡單說一下為什麼Python打包過大？

Python打包exe，不但體積大而且運行奇慢。解釋型語言大都是這個樣子，只不過Python尤其突出。要解決大而慢，只能用編譯型語言，如C，C++，甚至VB都好很多，體積最小的是彙編。^[1]

此外，還有知乎大佬說是因為“Anaconda里內置了很多庫，打包的時候打包了很多不必要的模塊進去，要用純淨的Python來打包。”

所以我們可以模擬一個新環境，其中只安裝我們本次打包所必要的工具包即可。

那最適合的就是——虛擬環境了！

虛擬環境

Python創建虛擬環境的方法有很多，而我是個Anaconda忠實用戶，如果你跟我一樣，那就簡單了。（大家也可以使用Virtualenv、Pipenv來設置虛擬環境，善用搜索，方法大同小異）

先記幾個命令，很簡單

```
conda create -n 虚拟环境名字 python==3.6 #创建虚拟环境
```

```
conda activate 虚拟环境名字 #激活虚拟环境
```

```
conda deactivate #退出虚拟环境
```

從開始菜單運行“Anaconda Prompt”，出現的界面輸入創建虛擬環境的指令。成功創建了一個名字為 `aotu`，且基於python版本3.6的虛擬環境。

```
管理员: Anaconda Powershell Prompt (anaconda3)
(base) PS C:\Users\Administrator> conda create -n aotu python=3.6
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

在創建過程中需要回復（y/n），Yes，再激活虛擬環境

```
(base) PS C:\Users\Administrator> conda activate aotu
(aotu) PS C:\Users\Administrator> _
```

conda安裝的虛擬環境，會把虛擬環境的目錄生成在anaconda安裝目錄下的env目錄下。

本地磁盤 (C:) > Users > Administrator > anaconda3 > envs >

名称

修改日期

类型

 aotu	2021/1/12/周二 ...	文件夾
 env5	2020/10/8/周四 ...	文件夾
 env66	2020/10/8/周四 ...	文件夾
 gui	2021/1/9/周六 1...	文件夾

當然我們也可以在剛剛的窗口，使用命令 `conda info --envs`，來查看conda環境下所有的虛擬環境

```
(aotu) PS C:\Users\Administrator> conda info --envs
# conda environments:
#
base                  C:\Users\Administrator\anaconda3
aotu                  * C:\Users\Administrator\anaconda3\envs\aotu
env5                  C:\Users\Administrator\anaconda3\envs\env5
env66                 C:\Users\Administrator\anaconda3\envs\env66
gui                   C:\Users\Administrator\anaconda3\envs\gui
```

安裝所需的庫

上面我們已經創建並激活了名為 `aotu` 虛擬環境，輸入 `conda list` 可以查看當前虛擬環境裡已經安裝的庫。

```
(aotu) PS C:\Users\Administrator> conda list
# packages in environment at C:\Users\Administrator\anaconda3\envs\aotu:
#
```

# Name	Version	Build	Channel
certifi	2020.12.5	py36haa95532_0	
pip	20.3.3	py36haa95532_0	
python	3.6.12	h5500b2f_2	
setuptools	51.1.2	py36haa95532_3	
sqlite	3.33.0	h2a8f88b_0	
vc	14.2	h21ff451_1	
vs2015_runtime	14.27.29016	h5e58377_2	
wheel	0.36.2	pyhd3eb1b0_0	
wincertstore	0.2	py36h7fe50ca_0	
zlib	1.2.11	h62dcd97_4	

我們打開所要打包的Python腳本，對比上圖，發現 `pandas`，`docx` 這兩個庫還需要額外安裝。當然，也不能少了打包必不可少的 `pyinstaller` 庫。

```
import pandas as pd
from docx import Document
from docx.shared import Pt
from docx.shared import Inches
from docx.oxml.ns import qn
from docx.enum.text import WD_PARAGRAPH_ALIGNMENT
from docx.enum.section import WD_ORIENTATION

'''----- 读取excel表数据整理后写入word （纯文本） -----'''
def wu_to_word(filepath):
    df = pd.read_excel(filepath, sheet_name="无")
    date_list = list(df['日期'])
    for d in date_list:
        filename = wordname+str(d)+"_docx" # 输出的word文件名
```

```

title = "("+str(d)[:4]+":"+str(d)[4:6]+":"+str(d)[6:8]+")" # 副标题日期XXXX.XX.XX
word = str(d)[:4]+"年"+str(d)[4:6]+"月"+str(d)[6:8]+"日" # 开头、落款日期XXXX年XX月XX日
wu_doc(title, word, filename)
print(f"文件: {filename},{title},{word} 已保存")

```

待打包脚本

安装库的过程不再赘述

```

pip install -i https://pypi.tuna.tsinghua.edu.cn/simple pandas

pip install -i https://pypi.tuna.tsinghua.edu.cn/simple python-docx

pip install -i https://pypi.tuna.tsinghua.edu.cn/simple pyinstaller

```

安装后再看

```

(aotu) PS C:\Users\Administrator> conda list
# packages in environment at C:\Users\Administrator\anaconda3\envs\autu:
#
# Name                      Version           Build    Channel
altgraph                    0.17              pypi_0    pypi
certifi                     2020.12.5         py36haa95532_0
future                      0.18.2            pypi_0    pypi
lxml                        4.6.2             pypi_0    pypi
numpy                       1.19.5            pypi_0    pypi
pandas                      1.1.5             pypi_0    pypi
pefile                      2019.4.18         pypi_0    pypi

```

```

pip 20.3.3 py36haa95532_0
pyinstaller 4.1 pypi_0 pypi
pyinstaller-hooks-contrib 2020.11 pypi_0 pypi
python 3.6.12 h5500b2f_2
python-dateutil 2.8.1 pypi_0 pypi
python-docx 0.8.10 pypi_0 pypi
pytz 2020.5 pypi_0 pypi
pywin32-ctypes 0.2.0 pypi_0 pypi
setuptools 51.1.2 py36haa95532_3
six 1.15.0 pypi_0 pypi
sqlite 3.33.0 h2a8f88b_0
vc 14.2 h21ff451_1
vs2015_runtime 14.27.29016 h5e58377_2
wheel 0.36.2 pyhd3eb1b0_0
wincertstore 0.2 py36h7fe50ca_0
zlib 1.2.11 h62dcd97_4

```

已成功安裝，還有一些因為這幾個庫所附帶安裝的，就不管他們了。

Pyinstaller打包步驟

這裡就不再重複了，只換個蘋果圖標試試

```
Pyinstaller -F -w -i apple.ico py_word.py
```

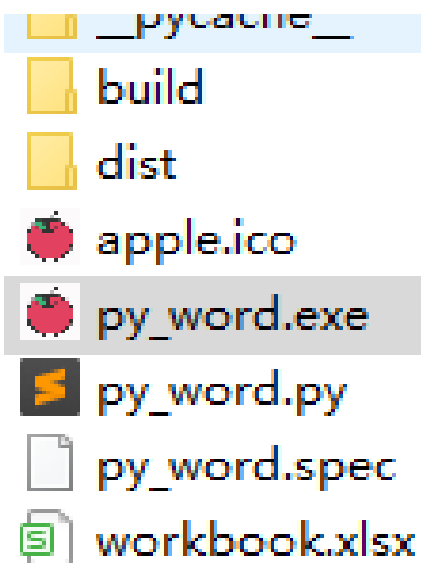
```

(aotu) PS C:\Users\Administrator> cd F:\py_word
(aotu) PS F:\py word> Pyinstaller -F -w -i pingguo.ico py word.py

```

```
105 INFO: PyInstaller: 4.1
105 INFO: Python: 3.6.12 (conda)
106 INFO: Platform: Windows-10-10.0.14393-SP0
107 INFO: wrote F:\py_word\py_word.spec
109 INFO: UPX is not available.
112 INFO: Extending PYTHONPATH with paths
['F:\\py_word', 'F:\\py_word']
126 INFO: checking Analysis
```

生成



- __pycache__
- build
- dist
- apple.ico
- py_word.exe
- py_word.py
- py_word.spec
- workbook.xlsx

大小: 29.8 MB (31,319,558 字节)

占用空间: 29.8 MB (31,322,112 字节)

创建时间: 2021年1月12日, 今天, 3 分钟之前

修改时间: 2021年1月12日, 今天, 3 分钟之前

成功壓縮到29.8M，如果不導入pandas這位大神，應該就可以10多M了

運行一下，毫無問題

 __pycache__	2021/1/12/周二 ...	文件夹	
 build	2021/1/12/周二 ...	文件夹	
 dist	2021/1/12/周二 ...	文件夹	
 apple.ico	2021/1/12/周二 ...	看图王 ICO 图片...	13 KB
 py_word.exe	2021/1/12/周二 ...	应用程序	30,586 KB
 py_word.py	2021/1/12/周二 ...	PY 文件	9 KB
 py_word.spec	2021/1/12/周二 ...	SPEC 文件	1 KB
 workbook.xlsx	2020/9/20/周日 ...	XLSX 工作表	13 KB

最後再簡單總結一下**虛擬環境+打包的全過程**（只三步）：

```
#创建虚拟环境
conda create -n aotu python=3.6

#激活虚拟环境
conda activate aotu

#Pyinstaller打包
Pyinstaller -F -w -i apple.ico py_word.py
```

總結一些小坑

- 1、說起來還是有點玄學，上文中一模一樣的過程我在兩個電腦都執行過一遍。在其中一個上顯示缺少 `xlrd` 這個庫，安裝後成功打包，也是一樣的大小。小伙伴們可以在文末獲取文件，也試試。
- 2、在安裝庫是要注意一些庫名，比如docx這個庫不要 `pip install docx`，而是需要

```
pip install python-docx
```

還有一些庫可能因為版本不同導致不能使用，多碰碰壁就好了。

- 3、為了防止打包時候有些庫沒安裝好，可以先在虛擬環境中執行一下Python腳本。運行無誤的話再打包，比較保險。

```
(aotu) PS F:\py_word> python F:\py_word\py_word.py
文件: XX公司业务数据表 (日报20191120) .docx, (2019. 11. 20), 2019年11月20日 已保存
文件: XX公司业务数据表 (日报20191121) .docx, (2019. 11. 21), 2019年11月21日 已保存
文件: XX公司业务数据表 (日报20191122) .docx, (2019. 11. 22), 2019年11月22日 已保存
文件: XX公司业务数据表 (日报20191126) .docx, (2019. 11. 26), 2019年11月26日 已保存
文件: XX公司业务数据表 (日报20191127) .docx, (2019. 11. 27), 2019年11月27日 已保存
文件: XX公司业务数据表 (日报20191204) .docx, (2019. 12. 04), 2019年12月04日 已保存
文件: XX公司业务数据表 (日报20191209) .docx, (2019. 12. 09), 2019年12月09日 已保存
```

```

文件: XX公司业务数据表 (日报20191212) .docx, (2019. 12. 12), 2019年12月12日 已保存
文件: XX公司业务数据表 (日报20191217) .docx, (2019. 12. 17), 2019年12月17日 已保存
文件: XX公司业务数据表 (日报20191219) .docx, (2019. 12. 19), 2019年12月19日 已保存
文件: XX公司业务数据表 (日报20191220) .docx, (2019. 12. 20), 2019年12月20日 已保存
文件: XX公司业务数据表 (日报20191223) .docx, (2019. 12. 23), 2019年12月23日 已保存
文件: XX公司业务数据表 (日报20191224) .docx, (2019. 12. 24), 2019年12月24日 已保存
文件: XX公司业务数据表 (日报20191225) .docx, (2019. 12. 25), 2019年12月25日 已保存
文件: XX公司业务数据表 (日报20191226) .docx, (2019. 12. 26), 2019年12月26日 已保存
文件: XX公司业务数据表 (日报20200101) .docx, (2020. 01. 01), 2020年01月01日 已保存
文件: XX公司业务数据表 (日报20200103) .docx, (2020. 01. 03), 2020年01月03日 已保存
文件: XX公司业务数据表 (日报20200107) .docx, (2020. 01. 07), 2020年01月07日 已保存
文件: XX公司业务数据表 (日报20200108) .docx, (2020. 01. 08), 2020年01月08日 已保存
文件: XX公司业务数据表 (日报20191119) .docx 已保存
文件: XX公司业务数据表 (日报20191125) .docx 已保存

```

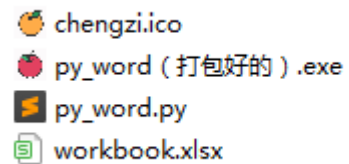
4、最後，如果想要刪除虛擬環境的話，可執行下方命令

```
conda remove -n aotu--all
```

5、打包為exe的話，版本盡量選擇python3.6+32位版本，因為win64位系統向下兼容32位程序，但是如果不考慮32位系統的話無所謂，直接python64位版本直接打包就可以，只是只能在win64位系統上跑。^[2]

下載鏈接

如果大家想測試Python打包，手頭又沒有合適的文件，可以在公眾號後台回復“**打包**”即可獲得以下文件：



參考資料

- [1] 弗拉基米爾: <https://www.zhihu.com/question/281858271/answer/613147412>
- [2] 《別再問我怎麼Python打包成exe了！》: https://mp.weixin.qq.com/s/zilDeFunWLG0mBS_x0vNnA



- 想看海量行業資料？
- 想學習陳老師的思維方式？
- 想和陳老師討論具體工作遇到的問題？
- 想在求職、轉行、轉崗道路上得到一對一親自指導？



↓↓↓加入十年數據人陳老師的知識星球↓↓↓



接地气的陈老师

星主：陈老师



知识星球

微信扫码预览星球详情

 接地气学堂

閱讀原文

喜歡此內容的人還喜歡

總結了這67個pandas函數，完美解決數據處理，拿來即用！

快學Python

將偷懶進行到底，實現雙擊直接打開.ipynb文件

凹凸數據



總結90條寫Python程序的建議！

法納斯特

