

办公自动化

本章介绍 PPT、Excel、Word、声频、视频等各种文件的自动化处理。



5.1 PyCharm 的使用

PyCharm 是高效的、免费的 Python 语言开发工具,从本章开始,使用 PyCharm 编写 Python 代码。

1. 下载与安装

进入 PyCharm 官网,如图 5-1 所示。



图 5-1 PyCharm 官网

单击 PyCharm 官网右上角的 Download 按钮,进入下载界面,如图 5-2 所示。

单击右侧的 Download 按钮,下载免费的社区版(Community),双击下载好的 PyCharm-community-2021.3.2. exe 程序进行安装。

依次单击【运行】→【是】→Next→Next,直到出现安装选项窗口,如图 5-3 所示。 全部勾选,单击 Next→Install 按钮进行安装,直到进入完成安装界面,如图 5-4 所示。



图 5-2 PyCharm 下载界面

PyCharm Community Edition Se Installat Configur	up — — — — · · · · · · · · · · · · · · ·	×
Create Desktop Shortcut PyCharm Community Edition Update Context Menu Add "Open Folder as Project"	Update PATH Variable (restart needed)	
Create Associations		
	< Back Next > Cance	ł

图 5-3 PyCharm 安装选项



图 5-4 PyCharm 完成安装

选择 Reboot now,单击 Finish 按钮完成安装,然后重启计算机。开机后双击桌面上的 PyCharm Community Edition 2021.3.2 的快捷方式运行程序,进入协议界面,如图 5-5 所示。

).0at (2), 近可	PyCharm User Agreement	×			
Ç harm	JETBRAINS COMMUNITY EDITION TERMS				
	IMPORTANT! READ CAREFULLY:				
	THESE TERMS APPLY TO THE JETBRAINS INTEGRATED DEVELOPMENT ENVIRONMENT TOOLS CALLED 'INTELLU IDEA COMMUNITY EDITION' AND 'PYCHARM COMMUNITY EDITION' (SUCH TOOLS, "COMMUNITY EDITION" PRODUCTS) WHICH CONSIST OF 1) OPEN SOURCE SOFTWARE SUBJECT TO THE APACHE 2.0 LICENSE (AVAILABLE HERE: https://www.apache.org/licenses/LICENSE- 2.0), AND 2) JETBRAINS PROPRIETARY SOFTWARE PLUGINS PROVIDED IN FREE- OF-CHARGE VERSIONS WHICH ARE SUBJECT TO TERMS DETAILED HERE: https: //www.jetbrains.com/legal/community-bundled-plugins.				
	"JetBrains" or "we" means JetBrains s.r.o., with its principal place of business at Na Hrebenech II 1718/10, Prague, 14000, Czech Republic, registered in the Commercial Register maintained by the Municipal Court of Prague Section C. File				
	I confirm that I have read and accept the terms of this User Agreement				
	Exit	nue			

图 5-5 PyCharm 协议窗口

勾选 I confirm that I have read and accept the terms of this User Agreement,再单击 Continue 按钮,进入 PyCharm 欢迎界面,如图 5-6 所示。

Welcome to PyCharm				¢
PyCharm 2021.2			×	ChinesePir
Projects			SDK	± 257.4K ± 4.5
Customize	SDK ChinesePin	yin-C 163000 r 4,59		
Plugins Learn PyCharm	Chinese (Si ≟ 4.2M ☆ 4	mplifi		
	Easy-Transl	ation Install 4.13	Englist	
	Zh-Translat	te-En		e original intention

图 5-6 PyCharm 欢迎界面

单击 Plugins 按钮安装插件,在搜索栏中输入 chinese,单击 ☐ 右边的 Install 按钮,安装 官方汉化插件;在搜索栏中输入 translation,单击 Install 按钮,安装翻译插件,也可以在打 开 PyCharm 后选择 File→Settings→Plugins 进行安装,安装完成后重启 PyCharm,以便使 插件生效。

2. 打开项目

用 PyCharm 打开项目有 3 种方法。

1) 新建项目

运行 PyCharm,进入如图 5-7 所示的欢迎界面, PyCharm 已经被汉化。

📓 欢迎访问 PyCharm				- 🗆 X
Pr PyCharm 2021.3.2		Q 搜索项目	新建项目	打开 从 VCS 获取
项目		run51.py D:\PyQt5\ZTYdatikashibie\ZTYda	atikashibie21	
自定义		2 2class D:\biji2021python\day15\2class	5	
插件 学习 PyCharm	0	1 1main D:\biji2021python\day15\1main	n	
		Orumenpython D:\0rumenpython		
5		ariuan5		

图 5-7 PyCharm 欢迎界面

单击【新建项目】按钮,进入新建项目界面,如图 5-8 所示。

新建1	页目		-		×
位置(L):	C:\Users\L	NOVO\PycharmProjects\pythonProject			1
▼ Pytł	non 解释器: P	ython 3.6			
○使	用此工具新建筑	⊼境 🔩 Virtualenv 👻			
位	1 1:	C:\Users\LENOVO\PycharmProjects\pythonProject\venv			Mir
基	本解释器:	Python 3.6 C:\Users\LENOVO\AppData\Local\Programs\Python\Python36-32\python.exe		*	
	〕继承全局站,	点软件包			
E]可用于所有	项目			
◎ 先前	前配置的解释器	20 27			
解	释器: 🕐	Python 3.6 C:\Users\LENOVO\AppData\Local\Programs\Python\Python36-32\python.exe		•	
创建 创建	! main.py 欢迎 ←个 Python 脚	迎脚本 本来为 PyCharm 中的明码提供入口术。			
			创建	Į	则消

图 5-8 新建项目界面

【位置】是用来设置新建项目(程序文件夹)路径的,选中【先前配置的解释器】指定已安装的 Python 程序,单击【创建】按钮,创建新的项目。

2) 打开

运行 PyCharm,进入如图 5-7 所示的欢迎界面,单击【打开】按钮,选择项目文件夹,打开项目。

3) 右击打开

最常用的方法是右击项目文件夹,选择 Open Folder as PyCharm Community Edition

Project,打开项目。

3. PyCharm 主界面各区域功能介绍

打开项目后,进入 PyCharm 程序主界面,如图 5-9 所示。



图 5-9 PyCharm 程序主界面

PyCharm 程序主界面由菜单栏、导航条、工具条、工具栏、代码编辑窗口、状态栏等组成。

工具栏各按钮的功能如下。

(1) Synthesize Python Packages: 包管理工具,用于查看、搜索、安装各种库。

(2) 灅 Python 控制台:打开交互编程窗口。

(3) 2 终端:进入 cmd 命令行窗口。

右击项目管理窗口内的文件,弹出的菜单有【新建】、【复制】、【粘贴】等各种功能,常用的 是【新建】功能,用于在项目目录中新建 Python 程序或目录等,如图 5-10(a)所示,还可以直 接打开程序所在的文件夹、cmd 路径等,如图 5-10(b)所示。

					8719里	,	
	文件(6) 编	掼(F) 汕肉(\) 弓航(N) (代码)(〇) 重构(8)	运行(1) 工具(1)	※ 剪切① 值 复制(C) 复制路径/引用 首 粘贴(P)	Ctrl+X Ctrl+C Ctrl+V	
	rumenpyth	non		1911(0) T 24(1)	查找用法(U) 检查代码(I)	Alt+F7	
ш	■项目・		0 2 ¥ \$	x —	重构(B)	>	
1 1 1	🖿 🖿 Orum	新建	>	創 文件	书签	>	
Г	> 0	X 剪切(I)	Ctrl+X	■ <□ ● 新建临时文件	删除(D) 重写文件类型	Delete	
	> 2	恒复制(C)	Ctrl+C	■ 日求 ■ Duthen th/th	① 在右侧拆分部分打开	Shift+Enter	
	> 3	复制路径/引用		III Python \$214	打开于	>	Explorer
	> 14	□ 粘贴(P)	Ctrl+V	Python 文件	本地历史记录(H) 〇 从磁盘重新加载	,	又件路径(2) 圖 终端
	> 5 > 6	查找用法(<u>U</u>) 在文件中查找…	Alt+F7 Ctrl+Shift+F	a 竹ML 文件 指资源包	→ [◆] 比较对象 与编辑器比较文件(<u>M</u>)	Ctrl+D	
	É.	7. 77 /H ch ##10./ A \	Ctrl . Chift . D	The EditorConfig	A Althe Clas		1
		(a) 养	所建		(b)	打开于	

图 5-10 PyCharm 项目管理

状态栏在底部,6:11 表示光标在第 6 行第 11 列; UTF-8 是编码类型; Python 3.6 表示所用 Python 的版本为 3.6。

4. 常用功能

1) 注释与缩进

在英文状态下,选中一段程序,按组合键 Ctrl+/可增加或去掉注释。

选中多行代码,按下 Tab 键,选中的代码向右缩进,按下组合键 Shift + Tab 键,代码向 左缩进。

2) 查看源码

按 Ctrl 键,再单击 Python 函数,会打开函数源码。

3) 跳转

单击右下角行列显示,输入行数后按 Enter 键,光标可以直接跳转到指定行。

4) 搜索

先选中需要搜索的内容,再按组合键 Ctrl+F,可以搜索选中的内容。

5) 替换

在命令栏中选择【编辑】→【查找】→【替换】,可以批量替换代码。

6) 自动引用库

当程序中没有引用库而直接使用库时,可以把光标放在库名上,此时库名下就会出现引 用库提示,单击需要的引用项后 PyCharm 会自动引用库。

7) 自动安装库

当程序没有安装库而直接引用库时,可以把光标放在库名上,当出现提示菜单时,单击选择需要安装的库,PyCharm 会自动安装库。

8) 代码格式化

如果出现 Tab 和空格混用的情况,则可在命令栏中选择【代码】→【重新格式化代码】, PyCharm 会自动统一格式。

9) 错误提示

PyCharm 主界面的右上角有一个问题代码报警数字,单击后在下面会出现报警类型及 所在的行数,双击报警信息后,光标会自动跳转到问题代码所在的行。

10) 自定义文件头

在实际的代码编写中,经常需要定义属于自己的文件头,例如程序的创建时间、作者、编码等。在 PyCharm 中,可以自定义文件头,以后新建的程序都会自动加载文件头内容,设置方法如下。

选择【文件】→【设置】→【编辑器】→【文件和代码模板】→ Python Script,如图 5-11 所示。

输入文件头内容,单击【确定】按钮,以后新建 Python 文件时,模板内容会自动加载,如图 5-12 所示。

11) 实时模板

选择【文件】→【设置】→【编辑器】→【实时模板】→ 🛉 增加快捷命令,如图 5-13 所示。

【缩写】即快捷键,例如 cs;【描述】是对快捷键内容的描述,例如"测试";【模板文本】即快 捷键对应的文本;最关键的步骤是选择"定义",然后选择 Python;【展开方式】默认为 Tab 键,

		×
Q-	编辑器 > 文件和代	IB模板 重
▶ 外观和行为	方套: Default	▼ Q
快速键		
	模板 包含	代码
▶ 编编器	+ - 10 5	
代码编辑	🚢 HTML File	
字体	HTML4 File	
▶ 颜色和字体	A XHTML File	//疗/问: \${DATE} \${TIME}
▶ 代码风格	Ball XML Properties	He 加約: 873366830aa.com
检查	Python Unit Test	at III 14
文件和代码模板	Setup Script	111.2E :
文件编码	0	
沃动植物		ппп
文社展開		
h KER	20	
一般な		図 検護式服装株式(V/R) 図 皮田立ち物(M (1)
Emmet	1.22	
TestMate Willia		and the second
Textmate interes		Apache Velocity template language is used
(20) (20) (20) (20)		
101/148-94		
「「「「「」」	175	
INT-STARE		
HIM/LA	23	
1017	24	

图 5-11 PyCharm 添加文件头



图 5-12 PyCharm 自动加载文件头

a 1211		×
Q.	编辑器 > 实时模板	EZ ← →
> 外現与行为 健立決射 > 編輯器 > 常規 代兵協研	RtXJR行方式 Tab ▼ O IFTML/XML V 図 Python C < (時点) Remod (形点)	
字体 > 配色方案 > 化码杆式 检查 文件编码 文件编码 文件编码 文件集型 文件类型 > 腕权 > 約4	<pre>Compd (Dict comprehension with 'if') C compd (Dict comprehension with 'if') C compd (Dict comprehension with 'if') C compd (List comprehension with 'if') C compd (List comprehension) C comps (Set comprehension) C comps (S</pre>	c
Emmet TextMate 測明和 TODO 参照) 自然语言 语言注入 = 術院建築様式 = 話件 次 = > 版本投制 =	編写(約): << 描述(0): 測点 機械文本(1): 測试内容 > 所約位置 > HTML > JSON > ダ Python > ダ Python > Shell 脚本 > XML ② マ ○ Other	線理空風(E) 通項 原行方式(X) 繁弘((1ab) ・ 接照料3C類新修式(R/R)
(?)		職定 取済 成用(A)

图 5-13 PyCharm 添加快捷键

不用更改;最后单击【确定】按钮,编程时输入快捷键 cs 再按 Enter 键或 Tab 键,这样快捷 健 cs 对应的文本内容会自动补齐。

笔者已经把 Python 编程常用的 36 个实时模板汇总在一起,供读者参考,详见 18.2.2 节。 12) 运行程序

运行 Python 程序常用的方法有以下 3 种:

(1) 组合键 Ctrl+Shift+F10。

(2) 右击【代码编辑】窗口,选择菜单中的 > 运行。

(3) 在【项目管理】窗口右击程序的文件名,选择 运行。

5.2 txt 文件的读写

txt 文本文件最常用的操作方法有全文读取、列表读取、分行写入。

1. 全文读取

全文读取是一次读出全部文本内容,示例代码如下:

```
with open('txt/1.txt',mode = 'r') as t:
    print(t.read())
```

运行结果如下:

张三 666666

打开文件 open('txt/1.txt', mode='r')命令的参数'txt/1.txt'为 1.txt 文件的路径, mode='r'表示只读模式。

将打开的文件对象命名为 t, read()用于全文读取文本。

2. 列表读取

列表读取是把文本内容的每行作为列表的一个元素,可以索引取出,适用于记录程序配 置信息,例如用户名、密码等,示例代码如下:

```
with open('txt/1.txt',mode = 'r') as t:
    print(t.readlines()[0])
```

运行结果如下:

张三

readlines()用于列表读取文本。

3. 写入

文本写入,常用的方式是逐行写入,也可以遍历列表循环写入,示例代码如下:

```
with open('txt/2.txt',mode = 'w') as t:
t.write('李四\n')
t.write('999999')
```

运行代码,打开保存的 2. txt 文件,内容如下:

李四 999999

open('txt/2.txt',mode='w')的参数 mode='w'是覆盖写模式,表示文件内容全部删除 重写。如果设置成 mode='a',则表示追加写模式,每次都从文本尾部追加写入,'\n'表示 换行。

5.3 CSV 文件的读写

CSV 文件是逗号连接的字符串存储的文本,CSV 文件也是手机通讯录支持的格式。

1. 写入单行文本

Python 写入单行文本时,把文本放入列表,列表用逗号连接成字符串写入即可,示例代码如下:

```
l = ['张三','李四','王五']
with open('csv/1.csv',mode = 'w') as t:
    t.write(','.join(1) + '\n')
```

CSV 文件的内容如下:

张三 李四 王五

2. 读取单行文本

读取单行文本,示例代码如下:

```
with open('csv/1.csv',mode = 'r') as t:
    l = t.read().strip('\n').split(',')
    print(1)
```

运行结果如下:

['张三', '李四', '王五']

与读取 txt 文本的方法相似,open()函数的参数 mode='r'表示只读。程序读出逗号连接的字符串后先用 strip('\n')去掉换行符,再用 split(',')分割成列表。

3. 多行多列数据的写入

多行多列数据的写入,示例代码如下:

```
1 = [['姓名', '性别'], ['张三', '男'], ['李四', '女']]
with open('csv/2.csv', mode = 'w') as t:
    for i in l:
        t.write(', '.join(i) + '\n')
```

运行代码,打开保存的 2. csv 文件,内容如下:

姓名	性别
张三	男
李四	女

当将多行数据写入文件时,一行数据用一个列表表示,遍历循环所有列表写入。

4. 多行多列数据的读取

多行多列数据的读取,示例代码如下:

```
l = []
with open('csv/2.csv',mode = 'r') as t:
    for i in t:
        l.append(i.strip('\n').split(','))
print(1)
```

运行结果如下:

```
[['姓名','性别'],['张三','男'],['李四','女']]
```

程序读取多行多列数据时,用 for 循环遍历每行数据,用 strip('\n')函数去掉每行的换 行符,用 split(',')函数把一行的内容根据','分割成一个列表元素,然后加入列表 l=[]中。

5.4 图形界面的自动操作

如果需要用计算机模拟鼠标、键盘自动操作一些重复性工作,就要用到 Pyautogui 库了。安装命令如下:

pip install pyautogui

1. 模拟鼠标的移动和单击

图形界面常用的操作是移动光标和单击动作,例如打开网页、窗口最大化、窗口最小化等,代码如下:

#//第 5 章/5.1.py	
import pyautogui, time	
time.sleep(5)	
<pre>pyautogui.moveTo(10,10,duration = 0.25)</pre>	#移动到 (10,10)
time.sleep(5)	
<pre>pyautogui.moveTo(300, 300, duration = 0.25)</pre>	#移动到(300,300)
time.sleep(5)	
<pre>pyautogui.moveRel(200,200,duration = 0.25)</pre>	#移动到 (200,200)
time.sleep(5)	
pyautogui.click()	#单击

pyautogui. moveTo(10,10,duration=0.25)函数的参数"10,10"用于将光标移动到绝 对坐标(10,10)位置,duration=0.25表示用时 0.25s; pyautogui. moveRel(200,200, duration=0.25)函数用于将光标移动到相对坐标(200,200)位置;绝对坐标是以屏幕左上 角为坐标原点,向右为x轴正方向,向下为y轴正方向的坐标;相对坐标是把光标当前位置 看作坐标原点的坐标。

2. 模拟键盘操作

键盘操作常用的是输入内容和按下组合键,例如输入用户名、密码、全选的组合键 Ctrl+A、 复制的组合键 Ctrl+C、粘贴的组合键 Ctrl+V 等,代码如下:

```
#//第5章/5.2.py
import pyautoqui, time
time.sleep(3)
pyautoqui.moveTo(100,100,duration = 0.25)
pyautogui.click()
                                              #输入字符串或列表
pyautoqui.typewrite('hellow\n', 0.25)
pyautoqui.PAUSE = 1
                                              #每个动作停1s
                                              #按下组合键 Ctrl + A
pyautoqui.hotkey('Ctrl', 'a')
pyautoqui.hotkey('Ctrl', 'c')
                                              #按下组合键 Ctrl+C
                                              ♯按下组合键 Ctrl+V
pyautoqui.hotkey('Ctrl', 'v')
pyautoqui.hotkey('Ctrl', 'v')
pyautoqui.hotkey('Ctrl', 'v')
                                              # 杳看所有键名
# print(pyautogui.KEYBOARD_KEYS)
```

运行程序前,首先打开一个文本编辑器并放置到屏幕左上角,将输入法调成英文输入法,然后运行程序,文本编辑器的内容如下:

hellow hellow

pyautogui.typewrite('hellow\n',0.25)函数的参数'hellow'是模拟键盘输入的内容,'\n'是 换行符,参数"0.25"表示输入每个字符停顿 0.25s; pyautogui.hotkey('Ctrl','c')函数用于 模拟键盘按下组合键'Ctrl+C';不同的键盘每个键的键名可能不同,用 print(pyautogui. KEYBOARD KEYS)命令可以查看键盘对应的键名。

3. 自动截屏

pyautogui 库有截屏功能,写个自动截屏程序监控计算机,代码如下:

```
#//第5章/5.3.py
import pyautogui,time
for i in range(3):
    im = pyautogui.screenshot() #截屏
    im.save('pyautogui/{}.jpg'.format(i)) #保存
    time.sleep(3)
```

除此之外, pyautogui 还有颜色判断、查找图像等功能。

获取屏幕像素坐标的技巧:按下 PrtScn 键截取屏幕,打开画图板后单击 ☐ 粘贴命令, 再单击 铅笔工具,移动鼠标指针,在状态栏左下角会显示铅笔工具所在位置的坐标。

5.5 Excel 文件的操作

Openpyxl 是处理 Excel 文件的第三方库,安装命令如下:

pip install openpyxl

1. 对文件的操作

Openpyxl对 Excel 文件有新建、打开、关闭、保存4种操作,代码如下:

井//第 5 章/5.4.py	
import openpyxl	
wb = openpyxl.Workbook()	♯新建 Excel 文件
<pre># wb = openpyxl.load_workbook('excel/example.xlsx')</pre>	#打开 Excel 文件
<pre># wb = openpyxl.load_workbook('excel/example.xlsx',data_only</pre>	= True)
wb.save('excel/example.xlsx')	#保存 Excel 文件
wb.close()	#关闭 Excel 文件

Workbook()命令用于创建一个新文件; load_workbook()命令用于打开已有的 Excel 文件;参数 data_only=True 的作用是把表中公式转换为运算结果,例如将'=1+3'转换 为运算结果 4,否则只能读出字符串'=1+3'; save()用来保存文件,最后用 close()关闭 文件。

2. 对工作表的操作

Openpyxl对 Excel 工作表有新建、删除、重命名、获得名称列表等操作,代码如下:

```
#//第5章/5.5.py
import openpyxl
wb = openpyxl.load workbook('excel/1.xlsx')
                                                      #打开文件
# sh1 = wb['Sheet1']
                                                      #打开工作表
                                                      #在索引号 0 处新建表 1
# wb.create sheet(index = 0, title = '1')
# wb.create sheet(index = 1, title = '2')
                                                      #在索引号1处新建表2
# wb.remove sheet(wb.get sheet by name('2'))
                                                      #删除名为2的表
# print(sh1.title)
#sh1.title = '重命名'
                                                      #重命名表
# print(sh1.title)
print(wb.sheetnames)
                                                      #获取表名列表
for sheet in wb:
                                                      #遍历打印表名
   print(sheet.title)
print('1' in wb. sheetnames)
                                                      #判断有没有表'1'
wb.save('excel/1.xlsx')
wb.close()
```

运行结果如下:

```
['11', '2', '1', '重命名', 'Sheet1']
11
2
1
重命名
Sheet1
True
```

wb['Sheet1']用表名索引文件对象,这是打开表最简单的方法; create_sheet(index=0, title='1')表示在0位置新建名称为"1"的表,没有索引时建立在最后面。依次去掉注释,查看程序运行的效果。

3. 对行、列的操作

Openpyxl 对行、列常用的操作有统计、插入、删除等,代码如下:

```
#//第5章/5.6.pv
import openpyxl
wb = openpyxl.load workbook('excel/1.xlsx')
sh1 = wb['重命名']
rows = sh1.max row
                            #获取总行数
cols = sh1.max column
                          # 获取总列数
print(rows,cols)
                          #第1行插入1空行
sh1.insert rows(1)
sh1.insert_cols(2)
                           #第2列插入1空列
sh1.delete cols(2, 2)
                           #删除第2列数据
sh1.delete_rows(1, 2)
                           #删除1~2两行数据
wb.save('excel/2.xlsx')
wb.close()
```

Openpyxl 对工作表从 0 开始索引, Openpyxl 对 Excel 行、列操作时, 起始行和列都是 1, 而且删除时包含右边界, 例如 sh1. delete_rows(1, 2)删除了第 1 列和第 2 列。

4. 对单元格的操作

Openpyxl 对单元格常用的操作有读取、写入、字体设置、字号设置、颜色设置、判断等, 代码如下:

```
#//第5章/5.7.py
import openpyxl
wb = openpyxl.load workbook('excel/1.xlsx')
sh1 = wb['重命名']
s = sh1.cell(row = 2, column = 1).value
                                                         #获取单元格的值
print(s)
sh1.cell(row = 5, column = 1).value = '王五'
                                                         #写入单元格的值
from openpyxl.styles import Font
                                                          #导入字体模块
fontred = Font(u'微软雅黑', size = 9, bold = True, italic = False, \
strike = False, color = 'FF0000')
                                                        #字体字号颜色
sh1.cell(row=5, column=3, value='记者').font = fontred #写人文本"记者"
print(sh1.cell(row = 5, column = 1).value)
print(sh1.cell(row = 5, column = 1).value is None)
                                                         #判断单元是否为空
sh1.cell(row = 5, column = 1).value = ''
                                                         #将单元格设置为 None
wb.save('excel/1.xlsx')
wb.close()
```

Openpyxl 一般用循环读取或写入内容,代码如下:

```
#//第5章/5.8.py
import openpyxl
wb = openpyxl.load_workbook('excel/2.xlsx')
shnm = wb['重命名']
sh1 = wb['1']
print(shnm.max_row)
for hang in range(shnm.max_row): #读出第1列的内容
print(shnm.cell(row = hang + 1, column = 1).value)
print(shnm.max_column)
```

```
for lie in range(shnm.max_column): #读出第1行的内容
    print(shnm.cell(row = 1, column = lie + 1).value, end = ' ')
for hang in range(shnm.max_row): #将表内数据写入表'1'
    for lie in range(shnm.max_column):
        sh1.cell(row = hang + 1, column = lie + 1).value\
            = shnm.cell(row = hang + 1, column = lie + 1).value
wb.save('excel/2.xlsx')
wb.close()
```

嵌套循环类似钟表的分针与秒针,分针走一格秒针走一圈,外层每取一个元素,内层就 会循环全部元素。

5.6 Word 文件的操作

Python-docx 是处理 Word 文档的第三方库,安装命令如下:

pip install python - docx

Python-docx 模块把一个 Word 文档分成了 3 个层级,第 1 层级为 Document(文档),一 个文件只有一个 Document,一个 Document 有 3 个部件(包含文档信息 sections);第 2 层 级为 Paragraph(段落),一个文件可以有多个段落;第 3 层级为 Run(块),一个段落可以有 多个 Run,一个 Run 就是相同的格式信息的连续字符串,如一张图像、一张表、一个字都可 以是一个 Run。

1. 文件的操作

Python-docx 对 Word 文档有新建、打开、保存 3 种操作,代码如下:

```
#//第5章/5.9.py
import docx
doc = docx.Document() #新建文档
# doc = docx.Document('word/2.docx') #打开文档
doc.save('word/1.docx') #保存文档
```

如果 docx. Document()函数不带参数,则表示新建文件,如果带文件名参数,则表示打 开已有文档,当保存为已存在文档时会覆盖已有文档。

2. 文本的操作

文本操作有添加段落和添加段内 Run 两种方法,代码如下:

```
#//第5章/5.10.pv
import docx
doc = docx.Document()
p1 = doc.add paragraph('检查')
                                      #添加第1段
p2 = doc.add_paragraph('作者:张三')
                                      #添加第2段
                                      #添加第3段
p3 = doc.add_paragraph()
run31 = p3.add run('这是第3段第1句话,')
                                     # 第 3 段, 第 1 个 Run
                                     #第3段,第2个Run
run32 = p3.add_run('这是第3段第2句话,')
run32 = p3.add run('这是第3段第3句话。')
                                      #第3段,第3个Run
doc.save('word/1.docx')
```

add_paragraph()用于添加段落,既可以把文本内容直接传入,也可以通过 Run 添加 文本。

3. 图像的操作

Python-docx 添加图像时可以设置图像的大小和位置,代码如下:

#//第 5 章/5.11.py	
import docx	
from docx.enum.text import WD_PARAGRAPH_ALIGNMENT	#引入对齐函数
from docx.shared import Cm	#引入单位(厘米)
doc = docx.Document()	
<pre>doc.add_picture("word/1.jpg", width = Cm(6))</pre>	#插入图像的宽为 6cm
last_paragraph = doc.paragraphs[-1]	#获得图像段落
last_paragraph.alignment = WD_PARAGRAPH_ALIGNMENT.CENTER	#图像居中
doc.save('word/1.docx')	

doc. add_picture("word/1.jpg", width=Cm(6))的参数"word/1.jpg"是插入图片的 地址,参数 width=Cm(6)是插入图片的宽度。

WD_PARAGRAPH_ALIGNMENT. CENTER 表示设置图像居中,也可以把 CENTER 换成 LEFT 或 RIGHT,以便让图像靠左或靠右。

4. 表格的操作

Python-docx 可以设置表格行数、列数、内容、位置等,代码如下:

#//第 5 章/5.12.py	
import docx	
from docx.enum.table import WD_TABLE_ALIGNMENT	#引人对齐函数
from docx.shared import Cm	#引入单位(厘米)
doc = docx.Document()	# 增加 2 行 2 列表格
<pre>table = doc.add_table(rows = 2, cols = 4, style = 'Table Grid')</pre>	
table.alignment = WD_TABLE_ALIGNMENT.CENTER	#表格居中
<pre># table.allow_autofit = False</pre>	#允许手动调节
for row in table.rows:	#设置表格列宽
<pre>row.cells[0].width = Cm(2)</pre>	#第1列宽2cm
<pre>row.cells[1].width = Cm(1)</pre>	#第2列宽1cm
table.cell(0,0).text = "姓名"	#设置单元格的值
table.cell(0,1).text = "性别"	
table.cell(1,0).text = "张三"	
table.cell(1,1).text = "男"	
<pre>doc.save('word/1.docx')</pre>	

doc. add_table(rows=2, cols=4, style = 'Table Grid')的参数 rows=2, cols=4 表示 插入表格为 2 行 4 列,参数 style = 'Table Grid'为表格的样式,关于更多的样式信息,详见 后面的样式设置。

5. 纸张设置 常用的纸张设置的代码如下:

♯//第 5 章/5.13.py from docx import Document from docx.shared import Cm doc = Document()section = doc.sections[0] # 获取 section 对象 section.page_width = Cm(21) #设置 A4 纸的宽度 section.page height = Cm(29.7)#设置 A4 纸的高度 #纸张上边距 2cm section.top margin = Cm(2)#纸张下边距 2cm section.bottom margin = Cm(2)#纸张左边距 2cm section.left margin = Cm(2) # 纸张右边距 2cm section.right margin = Cm(2)doc.save('word/1.docx')

doc. sections[0]表示文档的纸张设置信息。

6. 样式设置

文字、图像、表格等都有样式,样式的获取详见本章资源包 word 文件夹内,这里主要介 绍文本样式设置,代码如下:

```
#//第5章/5.14.py
import docx
                                                      #磅的单位及颜色
from docx. shared import Pt, RGBColor
doc = docx.Document()
#1 全局样式
doc.styles['Normal'].font.color.rgb = RGBColor(0, 0, 0)
                                                     #颜色
doc.styles['Normal'].font.size = Pt(12)
                                                     #字号
doc.styles['Normal'].font.name = u'宋体'
                                                      #字体
doc.styles['Normal']. element.rPr.rFonts.set(qn('w:eastAsia'), u'宋体')
#2标题样式,0,1,2...表示标题字号由大到小
titlenr = doc.add heading('物理', 1)
                                                     #标题
                                                     #居中
titlenr.alignment = WD_PARAGRAPH_ALIGNMENT.CENTER
#3段内样式
p1 = doc.add_paragraph()
run = p1.add_run('为红色文字,9px,斜体,楷体,非粗体,缩进 18px')
run.font.size = Pt(9)
run.font.color.rgb = RGBColor(255, 0, 0)
run.font.name = u'楷体'
run. element.rPr.rFonts.set(qn('w:eastAsia'), u'楷体')
run.italic = True
                                                      #倾斜
run.bold = False
                                                      #加粗
                                                      #设置下画线
run.underline = True
pl.paragraph format.first line indent = Pt(18)
                                                     #首行缩进 18 像素
pl.space after = Pt(5)
                                                     #上一段间隔 Pt(5)
pl.space_before = Pt(10)
                                                      #下一段间隔 Pt(10)
doc.add_page_break()
                                                      #插入分页符
doc.save('word/1.docx')
```

文本样式一般先设置全局样式,再设置标题样式,最后用 Run 设置段内样式。不同的 Run 名字不同,如 run1、run2等,在最后统一修改不同 Run 的格式。如果所有 Run 的名字 相同,则无法修改每个 Run 的格式。

7. 文本、图像、表格、样式的统计和索引

Python-docx 可以对文本、图像、表格、样式形成列表,进行统计和索引访问,代码如下:

```
#//第 5 章/5.15.py
import docx
```

```
doc = docx. Document('word/2. docx')
print(len(doc.paragraphs))
                                            #段落个数
print(len(doc.tables))
                                            #表格个数
print(len(doc.sections))
                                            #节个数
print(len(doc.styles))
                                            #样式个数
print(type(doc.tables))
                                            #图像个数
print(len(doc.inline shapes))
                                            #文字个数
print(len(doc.paragraphs[0].text))
                                           #获取第1段文字内容
print(doc.paragraphs[0].text)
print(doc.tables[0].rows[0].cells[0].text)
                                           #0表0行0列内容
doc.tables[0].rows[0].cells[0].text = 'www'
                                           #修改内容
doc.save('word/2.docx')
```

使用上述方法可以索引、修改已有文档。

5.7 PPT 文件的操作

Python-pptx 是处理 PPT 演示文档的第三方库,安装命令如下:

pip install python - pptx

Python-pptx 模块把一个 PPT 文档分成了 4 个层级,第 1 层级为 Presentations(文档), 一个文件只有一个 Presentations; 第 2 层级为 Sliders(页),一个文件可以有多页; 第 3 层 级为 Shapes(容器),Shapes 用于容纳文本框、表格和图像,slide.shapes.add_textbox()函数 用于添加文本框,add_paragraph()函数用于在文本框内添加段落; 第 4 层级为 Run(块),每 个段落可以用 add_run()函数添加多个 Run,每个 Run 可以设置不同格式,Run 的操作与 Word 中的 Run 的操作完全相同; slide.shapes.add_table()函数用于添加表格,slide. shapes.add_picture()函数用于添加图像。

1. 文档操作

Python-pptx 对 PPT 文档有新建、打开、保存 3 种操作,代码如下:

#//第5章/5.16.py	
from pptx import Presentation	
prs = Presentation()	#新建文档
<pre># prs = Presentation('ppt/1.pptx')</pre>	#打开已有文档
prs.save('ppt/1.pptx')	#保存文档

当用 save()函数保存文件时,如果文件已存在,则会覆盖原文件。

2. 幻灯片操作

Python-pptx 对幻灯片常用的操作有插入、删除等,代码如下:

```
#//第5章/5.17.py
from pptx import Presentation
prs = Presentation() #新建 PPT 文档
slide1 = prs.slides.add_slide(prs.slide_layouts[6]) #插入第1页空幻灯片
slide2 = prs.slides.add_slide(prs.slide_layouts[6]) #插入第2页空幻灯片
slide3 = prs.slides.add_slide(prs.slide_layouts[6]) #插入第3页空幻灯片
print(len(prs.slides))
del prs.slides._sldIdLst[-1] #删除最后一页幻灯片
print(len(prs.slides))
prs.save('ppt/1.pptx')
```

运行结果如下:

3 2

prs. slides. add_slide(prs. slide_layouts[6])用于插入空白幻灯片,参数 prs. slide_layouts[6]是插入幻灯片的样式,[6]表示空白幻灯片。

先插入了 3 张幻灯片, 删除最后 1 张后还有 2 张。删除之前, 一般用 print(len(prs. slides))命令查看共有几页, 再删除指定索引号的幻灯片。

3. 文本操作

用幻灯片的 shapes. add_textbox()方法添加文本框,用 textbox. text_frame. add_paragraph()方法添加段落,再设置段落内容,代码如下:

#//第 5 章/5.18.py	
from pptx import Presentation	
from pptx.dml.color import RGBColor	#颜色
from pptx.util import Inches, Pt	#英寸、磅等单位
<pre>prs = Presentation()</pre>	#新建 PPT 文档
slide1 = nrs slides add slide(nrs slide lavouts[6])	世插入旁白幻灯片
touthow = alide1 aboves add touthow(Inches(0.6) Inches	$\pi = \pi / (\pm / (\pm$
textbox - sinder. shapes. add_textbox(inches(0.0), inches	(0.5) Inches (9) , inches (0)
	H 你加义平性
tf = textbox.text_frame	# 获取文本框对象
<pre>para = tf.add_paragraph()</pre>	#文本框新增段落
para.text = "第1段"	#设置第1段内容
para.line_spacing = 4	#第1段1.5倍的行距
font = para.font	#设置第1段字体样式
font.name = '微软雅黑'	
font.bold = True	
<pre>font.size = Pt(20)</pre>	
<pre>font.color.rgb = RGBColor(255, 0, 0)</pre>	
<pre>new_para = textbox.text_frame.add_paragraph()</pre>	#添加第2段文本
new_para.text = "第2段"	#第2段文本内容
<pre>new_para = textbox.text_frame.add_paragraph()</pre>	#添加第3段文本
new_para.text = "第3段"	#第3段文本内容
prs.save('ppt/1.pptx')	

添加文本框 shapes. add_textbox()函数的4 个参数 x、y、w、h 即插入文本框的 x 坐标和 y 坐标,以及文本框的宽度 w 和高度 h。每个段落可以增加多个 Run,不同的 Run 可以设置 不同的格式,示例代码如下:

```
run1 = new_para.add_run()
run1.text = ("内容")
run1.font.name = u'楷体'
run2 = new_para.add_run()
run2.text = ("内容 2")
run2.font.name = u'黑体'
```

4. 图像操作

Python-pptx添加图像时可以设置图像的大小和位置,代码如下:

插入图像 slide2. shapes. add_picture()函数的 4 个参数 x、y、w、h 即插入图像的位置坐标 x 和 y,以及插入图像的宽度 w 和高度 h。

5. 表格操作

Python-pptx 可以设置表格的行数、列数、内容、位置等信息,代码如下:

```
#//第5章/5.20.py
from pptx import Presentation
from pptx.util import Inches
prs = Presentation()
                                                               ≠新建 PPT
slide3 = prs.slides.add slide(prs.slide layouts[6])
                                                               #插入空白幻灯片
rows, cols, left, top, width, height = 2, 3, Inches(3.5),
                     Inches(0.5), Inches(6), Inches(0.8)
                                                               #表格参数
table = slide3.shapes.add table(rows, cols, left, top,
                                       width, height).table
                                                              #添加表格
table.columns[0].width = Inches(2.0)
                                                               #第1列的宽度
table.columns[1].width = Inches(4.0)
                                                               #第2列的宽度
table.cell(0, 0).text = '语文'
                                                               #表格内容
table.cell(0, 1).text = '数学'
table.cell(0, 2).text = '英语'
table.cell(1, 0).text = '2'
table.cell(1, 1).text = '3'
table.cell(1, 2).text = '1'
prs.save('ppt/1.pptx')
```

Python-pptx 对表格的索引是从 0 开始的,第 1 行第 1 列的单元格索引为 table. cell(0, 0)。 当 Python-pptx 增加新的 Sliders、Shapes、textbox、paragraph、Run 时,尽管名字可以一样,但不能单独设置格式,如果想在程序尾部统一设置格式,则每个对象的名字不能相同。

5.8 JSON 文件的操作

JSON 是跨平台的文件格式,以键-值对形式存储,读写灵活,访问速度快,JSON 是 Python 的内置库,不需要安装。

1. 新建 JSON 文件

JSON 写入数据,代码如下:

```
#//第 5 章/5.21.py
import json
```

new_dict = {"张三":666," 李四":999} with open("jsonwj/test.json",'w') as f: json.dump(new_dict,f) #新建数据#新建 JSON 文件#写入数据

打开 test. json 文件,文件的内容如下:

{"\u51b0\u58a9\u58a9": 666, "\u96ea\u5bb9\u878d": 999}

这是汉字的十六进制数据,直接显示汉字的方法详见下文;这种方式是覆盖写,即删除 全部内容重写; JSON 数据中的字符串必须用双引号,而不能用单引号。

2. 读取数据

JSON 读取数据,代码如下:

```
#//第 5 章/5.22.py
import json
with open('jsonwj/test.json', 'r', encoding = 'UTF - 8') as f: #打开文件
    jg = json.load(f)["张三"] #获取内容
    print(jg)
```

运行结果如下:

666

open('jsonwj/test.json', 'r', encoding='UTF-8')命令的参数'jsonwj/test.json'是读 取的文件,参数'r'表示只读,参数 encoding='UTF-8'表示编码为 UTF-8。读取数据用 load(), 写入数据用 dump()。

3. 修改数据

JSON 添加、删除、修改数据的代码如下:

```
#//第5章/5.23.py
import json
with open("jsonwj/test.json", "r", encoding = 'UTF - 8') as f:
    data = json.load(f) #读取
    data["王五"] = "增加的" #增加
    data["李四"] = "增加的" #增加
    del data["李四"] #删除
    data["张三"] = "修改的" #修改
with open("jsonwj/test.json", 'w', encoding = 'UTF - 8')as f: #保存
    json.dump(data, f, ensure_ascii = False)
```

JSON 增加、修改、删除数据的方法和字典操作相同,打开 test. json 文件,内容如下:

{"张三":"修改的","王五":"增加的"}

参数 ensure_ascii=False 可使打开的 JSON 文件显示汉字。

5.9 视频文件的操作

FFmpeg 是开源的视频处理库,进入 FFmpeg 官网,如图 5-14 所示。





将光标悬停在 Marker 图标上,图标下面会弹出 Windows 版本的下载链接,选择 Windows builds by BtbN 进入下载界面,如图 5-15 所示。

https://gith	ک https://github.com/BtbN/FFmpeg-Builds/releases که ffmpeg-n5.0-latest-linux64-Igpl-5.0.tar.xz		î	8	G
				68	.5 MB
	G ffmpeg-n5.0-latest-linux64-lgpl-shared-5.0.tar.xz			2	28 MB
	G ffmpeg-n5.0-latest-win64-gpl-5.0.zip			10)4 MB
	ffmpeg-n5.0-latest-win64-gpl-shared-5.0.zip	40.7 MB			
	G ffmpeg-n5.0-latest-win64-lgpl-5.0.zip			8	38 MB
	ffmpeg-n5.0-latest-win64-lgpl-shared-5.0.zip			35	.3 MB
	Source code (zip)				
	Source code (tar.gz)				

图 5-15 FFmpeg 下载界面

选择 ffmpeg-n5. 0-latest-win64-gpl-5. 0. zip 进行下载,下载完成后双击 ffmpeg-n5. 0-latest-win64-gpl-5. 0. zip,把 bin 目录下的程序 ffmpeg. exe 复制到项目目录内,例如,笔者的目录是 ffm,再拷人一个 MP4 格式的视频文件并改名为 1. mp4。在该目录命令行中输入 cmd 命令后按 Enter 键,进入 cmd 命令行窗口,如图 5-16 所示。

📕 🛃 📮 ffm	
文件 主页 共享	查看
$\leftarrow \hspace{0.1in} \rightarrow \hspace{0.1in} \bigstar \hspace{0.1in} \uparrow \hspace{0.1in} \left[\hspace{0.1in} \blacksquare \hspace{0.1in} \operatorname{cmd} \right]$	
📕 ZTYdatikashibie 🖈 🔿	▲ 名称
2022-03	5 1 mn4
📕 baid	ffmpeq.exe
📕 ffm	

图 5-16 FFmpeg 目录

输入一条提取声频的命令 ffmpeg-i 1. mp4-q: a 0-map a sa. mp3 后按 Enter 键,如图 5-17 所示。

程序运行完成后,在ffm 目录生成了视频的声频文件 sa. mp3,如何用 Python 实现自动 批量操作呢? 需要用 Python 内置库 os 的 system()函数,它可以执行字符串内的命令,相

î

Microsoft Windows [版本 10.0.19044.1826] (c) Microsoft Corporation。保留所有权利。

D:\arumenpython\5\ffm>ffmpeg -i 1.mp4 -q:a 0 -map a sa.mp3

图 5-17 FFmpeg 命令

当于在 cmd 命令行窗口内执行,示例代码如下:

import os os.system('sa.mp3') #打开 MP3 os.system('notepad') #打开记事本 system()函数也可以打开 Word、视频等任意格式的文件。 FFmpeg 的常用命令如下。 提取视频 1. mp4 里的声频并保存为 1. mp3 文件,代码如下: import os os.system('ffmpeg - i 1.mp4 - q:a 0 - map a 1.mp3') 将声频文件 sa. mp3 切割为 5s 一份的多个声频文件,代码如下: import os os.system('ffmpeq - i sa.mp3 - f segment - segment time 5 - c copy out % 02d.mp3') 将声频文件 sa. mp3 切割为 5s 一份的多个声频文件并保存到 m 文件夹,代码如下: import os os.system('ffmpeg - i sa.mp3 - f segment - segment time 5 - c copy m out % 02d.mp3') 转换为百度能识别的单声道、16位,并且为 16kHz 采样的 PCM 格式,代码如下: import os os.system('ffmpeg - y - i sa.mp3 - acodec pcm_s16le - f s16le - ac 1 - ar 16000 r.pcm') 剪去所有超过 2s 的静音片段,代码如下: import os os.system('ffmpeg - i sample.mp3 - af silenceremove = stop periods\ = -1:stop duration = 2:stop threshold = - 30dB output.mp3') 选择性切割,参数-ss用于设置开始切割的位置,参数-t用于设置切割持续的时间,代 码如下: import os

os.system('ffmpeg - i 1.mp3 - ss 00:00:03 - t 00:00:12 - acodec copy tmp.mp3')

MP4 格式转换为 AVI 格式,代码如下:

```
os.system('ffmpeg - i 1.mp4 output.avi')
```

FFmpeg还有合并声频、视频、录音、录屏等功能,格式化工厂就是由FFmpeg开发出来的。

%02d中的%在字符串中表示占位符,d表示占位符的变量是整数,2表示整数长度为2,0表示不足2位的用0从左边填充。与format()命令的用法相似,其他常用方法如下:

print('%2d-%02d'%(4,1))结果为 4-01。

print('%.3f'% 3.1415926)的结果为 3.142, 3 表示保留 3 位小数, f 表示浮点数。

print('我%s是%s人!'%('就','中国'))的结果为'我就是中国人!',s表示字符串。

当代码中%02d省略了变量时,默认从1开始递增。%占位符还可以控制两对三单 引号'''''表示的多行字符串内的变量,详见17.7.py文件,这是format()命令所不能的。

Python 第3种格式控制 f-string 则是整合了以上两种方法中的优点,示例代码如下:

```
print(f'''结果为{6 * 8}''')
```

运行结果如下:

结果为 48

如果变量是字符串,则使用更灵活,代码如下:

aaa = '光' ss = f""" 床前 明月 {aaa} """ print(ss)

运行结果如下:

床前 明月 光

在代码 17.6. py 文件中也用到了 f-string 格式控制的方法。

5.10 自动发送邮件

Python 内置库 smtplib、email 用于邮件发送。以 QQ 邮箱为例,首先要获取授权码,步骤如下:登录 QQ 邮箱,选择【设置】→【账户】→POP3/SMTP→【已开启】,手机发送短信后 收到授权码。

1. 发送纯文本邮件

smtplib、email 发送纯文本邮件,代码如下:

```
#//第 5 章/5.24.py
import smtplib
```

from email.mime.text import MIMEText msg = MIMEText('这是纯文本邮件内容','plain','UTF-8') #创建邮件 msg['From'] = '发件邮箱' #发件邮箱 msg['To'] = '收件邮箱' #收件邮箱 msg['Subject'] = "标题" #邮件标题 server = smtplib.SMTP SSL('smtp.qq.com',465) #实例化应用 trv: server.login('发件邮箱','授权码') #发件邮箱及授权码 server.send message(msg) #发送 print('已发送') except Exception: print('发送失败') pass server.guit() # 退出

MIMEText()函数的第1个参数'这是纯文本邮件内容'是邮件正文,第2个参数'plain' 表明邮件类型为纯文本邮件,第3个参数表明编码为'UTF-8',否则会出现乱码。

2. 发送带附件的邮件

发送带附件的邮件,代码如下:

```
#//第5章/5.25.py
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
msg = MIMEMultipart()
                                              #创建邮件
msg['From'] = '发件邮箱'
                                              #发件邮箱
msg['To'] = '收件邮箱'
                                              #收件邮箱
msg['Subject'] = "标题"
                                              #邮件标题
                                              #附件1
att1 = MIMEText(open('附件文件路径', 'rb').read(), 'base64', 'UTF-8')
att1["Content - Type"] = 'application/octet - stream'
att1["Content - Disposition"] = 'attachment; filename = "显示的附件文件名"'
msg.attach(att1)
                                              #附件2
att2 = MIMEText(open('附件文件路径', 'rb').read(), 'base64', 'UTF-8')
att2["Content - Type"] = 'application/octet - stream'
att2["Content - Disposition"] = 'attachment; filename = "显示的附件文件名"'
msg.attach(att2)
server = smtplib.SMTP_SSL('smtp.qq.com',465)
                                             #实例化应用
trv:
   server.login('发件邮箱', '授权码')
                                              #发件邮箱及授权码
                                              #发送
   server.send message(msg)
   print('已发送')
except Exception:
   print('发送失败')
   pass
server.guit()
                                              #退出
```

本例中发送了两个附件,如果需要,则可以添加更多附件。smtplib、email还可以发送 html 邮件。

5.11 Pandas 数据分析

Pandas 是专业的数据分析库,功能强大,这里仅对常用功能进行介绍,安装命令如下:

pip install pandas

1. 文件操作

Pandas 对文件的操作有打开、保存、显示等,代码如下:

#//第5章/5.26.py	
import pandas as pd	
df = pd.read_excel('pan/1.xlsx',index_col = '姓名')	
df2 = pd.read_excel('pan/1.xlsx',skiprows = 1,usecols = 'a:b')	#显示 a:b 列数据
print(df)	#默认显示前5行
print(df2)	
<pre>print(df.head(2))</pre>	#显示前2行内容
<pre>print(df.tail(2))</pre>	#显示后2行内容
df.to_excel('pan/2.xlsx')	#保存

运行结果如下:

	语文	数学
姓名		
张三	99	96.0
李四	96	NaN
王五	97	91.0
	张三	99
0	李四	96
1	王五	97
	语文	数学
姓名		
张三	99	96.0
李四	96	NaN
	语文	数学
姓名		
李四	96	NaN
王五	97	91.0

read_excel()函数的参数 index_col='姓名'是指定第1列'姓名'为索引列,如果不指定, Pandas 则会自动在第1列位置插入新的索引列;参数 skiprows=1表示跳过第1行数据, 如果设置为 skiprows=2,则表示跳过前两行数据,可以把需要跳过的行放入列表中,跳过 多列;参数 usecols='a:b'表示只显示 a 到 b 两列数据;默认情况下 print()只显示前5 行数 据, df. head(2)表示只显示前两行数据, df. tail(2)表示只显示后两行数据。

2. 统计列信息

Pandas 常用的统计方法有最大值、最小值、平均值、缺失的空值等,代码如下:

```
#//第 5 章/5.27.py
import pandas as pd
```

```
      df = pd. read_excel('pan/1.xlsx', index_col = '姓名', header = 0)

      print('总数',df['数学'].size)
      #返回值的总数

      print('正缺',df['数学'].count())
      #返回非缺失值的数目

      print('空值',df['数学'].isnull().sum())
      #统计空值的个数

      print('最小',df['数学'].min())
      #返回最小值

      print('最大',df['数学'].max())
      #返回最大值

      print('平均',df['数学'].max())
      #返回最大值

      print('平均',df['数学'].max())
      #返回平均值

      print('总和',df['数学'].sum())
      #返回总和

      df.to excel('pan/2.xlsx')
      #
```

参数 header=0 用来指定题头,默认为 0,即第 1 行为题头,也可以将其他行指定为 题头。

3. 计算列

Pandas 可以直接对列数据进行运算,代码如下:

对列可以直接进行加、减、乘、除等运算,运行结果如下:

	语文	数学
姓名		
张三	99	96.0
李四	96	NaN
王五	97	91.0
	语文	数学
姓名		
张三	99	98.0
李四	96	NaN
王五	97	93.0

Pandas 不仅可以对列直接进行四则运算,还可以对两个相同结构的表进行四则运算,示例代码如下:

pf = pf + pf2

详见 6.21 节内容。

4. 空值处理

从以上示例可以看出,空值不能参与运算,所以数据运算之前要对空值进行处理,代码 如下:

```
#//第5章/5.29.py
import pandas as pd
df = pd.read excel('pan/1.xlsx', index col = '姓名', header = 0)
```

```
print(df)
df['数学']=df['数学'].fillna(0)
print(df)
# df.to_excel('pan/2.xlsx')
```

#用0替换缺失值

运行结果如下:

	语文	数学
姓名		
张三	99	96.0
李四	96	NaN
王五	97	91.0
	语文	数学
姓名		
张三	99	96.0
李四	96	0.0
王五	97	91.0

fillna(0)可以把空值换成 0,也可以把空值处理成其他值,例如 fillna(1)可把空值都换成 1。

5. 插入列

Pandas 可以直接插入新列,并赋值或计算,代码如下:

```
#//第 5 章/5.30.py
import pandas as pd
df = pd.read_excel('pan/1.xlsx', index_col = '姓名')
print(df)
# df["总分"] = '' # 后面建一列,其值为空
df['数学'] = df['数学'].fillna(0) # 后面建一列,其值为 0
df["总分"] = df["数学"] + df["语文"] # 插入新列并计算
print(df)
# df.to_excel('pan/2.xlsx')
```

例中有 3 行代码用于插入新列,即赋值为空、赋值为 0、赋值为两列之和,当然也可以赋 值为其他。

6. 排序

用 openpyxl 库对数据进行排序是非常麻烦的,用 Pandas 排序就简单多了,代码如下:

```
#//第5章/5.31.py
import pandas as pd
df = pd.read_excel('pan/1.xlsx', index_col = '姓名')
print(df)
df.sort_values(by = '数学', inplace = True, ascending = True) # '数学'列升序
# df.sort_values(by = ['数学', '语文'], inplace = True, ascending = [True, False])
print(df)
df.to_excel('pan/2.xlsx')
```

对"数学"这一列按升序进行排序,运行结果如下:

	语文	数学	
姓名			
张三	99	96.0	
李四	96	NaN	
王五	97	91.0	
	语文	数学	
姓名	语文	数学	
姓名 王五	语文 97	数学 91.0	
姓名 王五 张三	语文 97 99	数学 91.0 96.0	
姓名 王五 张三 李四	语文 97 99 96	数学 91.0 96.0 NaN	

sort_values()排序函数的参数 inplace=True 表示在原数据中修改,inplace=False 表示在新建数据中修改; ascending=True 表示升序,ascending=False 表示降序; 将多列数据排序放入列表中,例如 by=['数学','语文'],排序方法也是列表,例如 ascending=[True, False],表示先按'数学'升序排序,再按'语文'降序排序。

7. 筛选

Pandas 的筛选功能就是筛选出符合条件的行,代码如下:

```
#//第5章/5.32.py
import pandas as pd
df = pd.read_excel('pan/1.xlsx', index_col = '姓名')
print(df)
df = df.loc[df['数学'].apply(lambda a:95 <= a < 100)] # '数学'列筛选
print(df)
df.to_excel('pan/2.xlsx')</pre>
```

程序只对"数学"这一列进行筛选,运行结果如下:

	语文	数学
姓名		
张三	99	96.0

本例中用到了 Python 内置的匿名函数 lambda,读者可百度"匿名函数 lambda 教程"进行学习。

打开保存的文件可以看到,文件中只有筛选数据,这一点与 Excel 的筛选功能不同。 Pandas 也可以实现多列筛选。

5.12 Matplotlib

Matplotlib 是专业的数据可视化工具,安装命令如下:

pip install matplotlib

下面对常见的几个绘图命令进行介绍。

1. 散点图

scatter()函数用于绘制散点图,代码如下:

运行结果如图 5-18 所示。

再加一组数据 y2,指定颜色、标记符号及图例,代码如下:

#//第5章/5.34.py	
import matplotlib.pyplot as plt	
$\mathbf{x} = [1, 2, 3, 4, 5]$	#1 x 坐标值
y = [2.3, 3.4, 1.2, 6.6, 7.0]	#2y坐标值
y2 = [3.3, 3.4, 4.2, 10.6, 13.0]	#2 y2 坐标值
<pre>plt.scatter(x, y, color = 'r', marker = ' + ')</pre>	#3 绘制红色+号
plt.scatter(x, y2)	
plt.legend([r'\$Y\$',r'\$N\$'])	#添加图例
<pre>plt.savefig('plt/sdt.png')</pre>	#4 保存图像
plt.show()	#5 展示图像

运行结果如图 5-19 所示。



2. 折线图

Matplotlib 并没有直接提供绘制折线图的函数,但是可以用 plot()函数实现,它既可以 画点图,也可以画线图,代码如下:

#//第 5 章/5.35.pv	
import matplotlib.pyplot as plt	
$\mathbf{x} = [1, 2, 3, 4, 5]$	#1x坐标值
y = [2.3, 3.4, 1.2, 6.6, 7.0]	#2y坐标值
y2 = [3.3, 3.4, 4.2, 10.6, 13.0]	
<pre>plt.plot(x, y, color = "g", marker = 'D', markersize = 5)</pre>	#3 绘图
plt.scatter(x, y2)	
plt.legend([r'\$Y\$',r'\$N\$'],loc = "lower right")	#图例
<pre>plt.rcParams['font.sans - serif'] = 'simhei'</pre>	#显示中文

plt.xlabel("时间")	♯x轴标签
plt.ylabel("活跃度")	♯y轴标签
plt.title("用户活跃度")	#标题
<pre>plt.savefig('plt/zxt.png')</pre>	#4 保存
plt.show()	#5 展示

运行结果如图 5-20 所示。



plt.plot(x, y, color="g", marker='D', markersize=5)函数的参数 x 和 y 是坐标数 值, color="g"表示线条是绿色, marker='D'表示坐标点的线型为菱形, markersize=5 表示 坐标点大小,还可以用 linestyle='--'表示线型为虚线, linewidth=2.5 表示线的粗细等。 线型类别见表 5-1。

松 31 以主大加

符号(Specifier)	线型名称(Line Style)
-	实线(默认样式)
	虚线(短画线)
:	点线
	点画线

颜色代码见表 5-2。

表 5-2 颜色代码

符号(Specifier)	颜色名称(Color)
у	黄色
m	品红
с	蓝绿色
r	红色
g	绿色
b	蓝色
W	白色
k	黑色

坐标点样式见表 5-3。

符号(Specifier)	坐标点样式名称(Marker)
0	圆
+	加号
*	星号
	点
x	十字
s	正方形
d	菱形
٨	上指向三角形
v	下指向三角形
>	右指向三角形
<	左指向三角形
р	五角星
h	六角形

表 5-3 坐标点样式

当用 Matplotlib 绘制多组数据时,如果没有指定样式,则 Matplotlib 会按上述样式顺序 赋值。

#1 x 坐标值

#2 y 坐标值

#4 保存 #5 展示

3. 柱状图

Matplotlib 提供了 bar()函数来绘制柱状图,代码如下:

#//第5章/5.36.py import matplotlib.pyplot as plt x = [1, 2, 3, 4, 5]y = [2.3, 3.4, 1.2, 6.6, 7.0]#3 绘制柱状图 plt.bar(x, y) # plt. savefig('plt/zzt. png') plt.show()

运行结果如图 5-21 所示。 多组数据的柱状图绘制,代码如下:

```
#//第5章/5.37.py
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5]
                                        #1x坐标值
y = [2.3, 3.4, 1.2, 6.6, 7.0]
                                        #2y坐标值
x2 = [1.4, 2.4, 3.4, 4.4, 5.4]
y_2 = [3.3, 3.4, 4.2, 10.6, 13.0]
fig = plt.figure()
                                        #添加子图区域
ax = fig.add_axes([0,0,1,1])
                                        #子图比例系数
ax.bar(x, y, color = 'r', width = 0.4)
                                        #3 绘制柱状图
ax.bar(x2, y2, color = 'g', width = 0.4)
# plt. savefig('plt/zzt37.png')
                                        #4 保存
plt.show()
                                         #5 展示
```

运行结果如图 5-22 所示。



ax = fig.add_axes([0,0,1,1])函数的参数[0,0,1,1]表示子图在原图中距左边、底 边、宽度、高度的比例系数。读者可以把参数[0,0,1,1]换成[0.2,0.2,0.8,0.8],对比效果, 加深理解。

Matplotlib 绘制多柱状图时需要添加子图,并且需要指定子图在原图中的位置及缩放比例,相当于多张图叠加在一起,所以每个柱状图都有各自的 *x*、*y*数据,各柱状图 *x* 坐标值的差值应为柱状图的宽度,本例中柱状图的宽度为 width =0.4,差 1.4-1 为 0.4,表示画完一个紧接着画下一个。如果 *x* 轴数据差大于柱宽,则会出现柱子分离现象,示例代码如下:

```
#//第5章/5.38.py
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5]
                                         #1x坐标值
                                         #2 y 坐标值
y = [2.3, 3.4, 1.2, 6.6, 7.0]
x2 = [1.5, 2.5, 3.5, 4.5, 5.5]
y^2 = [3.3, 3.4, 4.2, 10.6, 13.0]
                                         #添加子图区域
fig = plt.figure()
ax = fig.add axes([0,0,1,1])
                                         #子图比例系数
                                         #3 绘制柱状图
ax. bar(x, y, color = 'r', width = 0.3)
ax.bar(x2, y2, color = 'g', width = 0.3)
# plt. savefig('plt/zzt38.png')
                                         #4 保存
plt.show()
                                         #5 展示
```

运行结果如图 5-23 所示。

差距越大,柱之间的距离就越大。

4. 饼图

Matplotlib 的 pie()函数用于绘制饼图,代码如下:

```
#//第5章/5.39.py
from matplotlib import pyplot as plt
x = [23, 17, 35, 29, 12]
                                                #1x坐标值
fig = plt.figure()
                                                #添加子图区域
ax = fig.add_axes([0,0,1,1])
                                               #子图比例系数
langs = ['C', 'C++', 'Java', 'Python', 'PHP']
                                               # 外标签
ax.pie(x, labels = langs, autopct = '%1.2f% %')
                                               #3 绘制饼图
plt.savefig('plt/zzt39.png')
                                               #4 保存
                                                #5 展示
plt.show()
```



ax. pie(x, labels = langs, autopct = '%1. 2f%%')函数的参数 x 为一维数组,参数 labels 是饼图的外标签,参数 autopct 是饼图的内标签,'%1. 2f%%'是饼图面积的百分比, 包含两位小数。

5. 多子图并列

当用 Matplotlib 绘制多子图时,可用 subplot()指定行数、列数及画图位置,代码如下:

#//第 5 章/5.40.py	
from matplotlib import pyplot as plt	
<pre>x = ['group_a', 'group_b', 'group_c']</pre>	#1 x 坐标值
y = [1, 10, 100]	#2y坐标值
<pre>plt.figure(figsize = (9, 3))</pre>	#图宽9英寸,高3英寸
plt.subplot(131)	#1行3列放在第1列
plt.bar(x, y)	#3 绘图
plt.subplot(132)	#1行3列放在第2列
<pre>plt.scatter(x, y)</pre>	#3 绘图
plt.subplot(133)	#1行3列放在第3列
plt.plot(x, y)	#3 绘图
<pre>plt.savefig('plt/zzt40.png')</pre>	#4 保存
plt.show()	#5 展示

运行结果如图 5-25 所示。

plt. subplot(131)函数的参数 131 表示绘图的位置在 1 行 3 列的第 1 列位置。

还可以用 facecolor 参数指定背景颜色,用 edgecolor 参数指定边框颜色,用 frameon 参数指定是否显示边框,用 axis('off') 参数关闭子图外轮廓。

还有其他设置,如设置坐标刻度值的大小,代码如下:

plt.tick_params(labelsize = 25)

设置图例字体及大小,代码如下:



5.13 百度 AI

由于百度 AI 提供了很方便的接口,所以可以很容易地调用百度 AI 实现人工智能。进 入百度 AI 官网后,单击右上角的【控制台】按钮进行登录,使用已有的百度账号即可,如果 没有百度账号,则需要注册一个并完成实名认证。

安装百度 AI 库的命令如下:

pip install baidu - aip

5.13.1 图像文字识别

1. 创建文字识别应用

将光标移动到控制台左上角蓝色导航按钮 ■上,网页会自动弹出导航栏,如图 5-26 所示。

依次选择【导航栏】→【文字识别】→【创建应用】,弹出的界面如图 5-27 所示。

【应用名称】可以任意填写,例如 wzocr,选择【接口选择】右边红色的【去领取】,进入【领取页面】界面,【接口名称】选择【全部】→【0元领取】,然后关闭本界面,回到创建应用界面, 【文字识别 HTTP SDK】不用选择,【应用归属】选择【个人】,【应用描述】可以任意填写,单击 【立即创建】→【返回列表】。



新建一个 keyword.txt 文件,将 AppID、API Key、Secret Key 内容复制到 keyword.txt 内,Secret Key 是隐藏的,单击【显示】按钮后再复制,如图 5-28(a)所示,新建文本内容如 图 5-28(b)所示。

2. 图像文字识别

图像文字识别只需简单传入 key 和图像,代码如下:

∃ ○ 百度智能云 🗄	常制台总览 全局 ~ Q	盟 🗊 🛱 餐業
詣 产品服务	请输入关键词	
♀ 语音能力引擎	计算	存储和CDN
① 文字识别	☐ 云服务器 BCC	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
● 人脸识别	□ 轻量应用服务器 LS □ 专属服务器 DCC	🔛 云磁盘 CDS
(♣) 人体分析	圓 弹性裸金属服务器 BBC	■ 数据流转平台
[4] 图像识别	◎ 弹性伸缩	計 内容分发网络 計 动态加速 DRCDN
◎ 图像搜索	● 云手机 BAC	智能视频
[4] 图像增强与特效	二百十	〕 音视频处理 MCP
~ 内容审核		□1 音视频直播 LSS
◎ 自然语言处理	◇ 容器引擎 ○ 容器镜像服务 CCR	▶ 视频刨作分发半台 ▶ 媒体内容审核 VCF

图 5-26 产品服务导航

文字识别	く 返回 创建新	应用
概览		
公有云服务	↑ 部分接口务	b.费额度还未领取,请先去领取再创建应用,确保应用可以正常调用 去领取
・ 应用列表	* 应用名称:	wzocr
• 监控报表	* 接口选择:	勾选以下接口,使此应用可以请求已勾选的接口服务,注意文字识别服务已默
・ 技术文档 🖉		您已开通付费的接口为:通用文字识别(高精度版)
• API在线调试 🖉	* 应用归属:	公司 个人
• HTTP SDK 🖉	* 应用描述:	文字识别
私有化部署服务		2
离线SDK		
图文转换器		
	立即创建	取消

图 5-27 创建文字识别应用

				keyword.txt - 记事本 文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
应用名称	AppID	API Key	Secret Key	201152 pebc2umo2c2oTW4XQqbMfO
wzocr	20115245	pebc2umo2c2oTW4XQqbMfO4u	显示	8aNLyDqx3GEcOYNLZUx4LFYVmQfg9y
	(a) 文字	识别key		(b)将key保存到文件
		图 5-28 文字识别	key	

#//第5章/5.41.py	
from aip import AipOcr	
client = AipOcr('AppID', 'API Key','Secret Key')	井加载 key
with open('baid/1.png', 'rb') as f:	#读取图像
<pre>image = f.read()</pre>	
<pre>text = client.basicAccurate(image)</pre>	# 调 用
<pre>result = text['words_result']</pre>	#获取结果
for i in result:	
<pre>print(i['words'])</pre>	#打印结果

识别结果如下:

展示目录.txt

文字识别还有生僻字版、通用文字识别 enhancedGeneralUrl()、网络图像文字识别 webImageUrl()、身份证识别 idcard()、表格文字识别 form()、二维码识别 qrcode()、数字识别 numbers()、手写文字识别 handwriting()等功能,需要把代码中的 basicAccurate()函数 换成相应的函数,准确率更高、速度更快,如果不换,则用通用方法识别。

5.13.2 语音与文字互转

1. 创建语音识别应用

依次选择【导航栏】→【语音技术】→【创建应用】,其他步骤与创建文字识别应用的步骤 类似,最后将语音 key 保存为 keyvoice.txt 文件。

2. 文字转语音

文字转语音 MP3,代码如下:

```
#//第5章/5.42.py
from aip import AipSpeech
client = AipSpeech('AppID', 'API Key', 'Secret Key')
result = client.synthesis('百度 AI 开放平台', 'zh', 1)
if not isinstance(result, dict):
    with open('baid/auido.mp3', 'wb') as f:
    f.write(result)
```

#加载 key #请求 #保存

运行结果是在文件夹内生成 auido. mp3 语音文件。

client.synthesis('百度 AI 开放平台','zh',1)的参数'百度 AI 开放平台'是文本内容,参数'zh'表示转换为中文,参数"1"表示 Web 端请示。更多设置(例如改变语速、音调、音量、发声人物等)详见 10.10 节。

3. 语音文件转文字

语音文件转文字,代码如下:

#//第 5 章/5.43.py	
from aip import AipSpeech	
<pre>client = AipSpeech('AppID', 'API Key', 'Secret Key')</pre>	♯加载 key
with open('baid/record.wav', 'rb') as fp:	#读取声频文件
au = fp.read()	
res = client.asr(au, 'wav', 16000, {'dev_pid': 1537, })	#请求
print('识别结果:' + "".join(res['result']))	

识别结果如下:

识别结果:现在进行测试。

asr(au,'wav',16000,{'dev_pid':1537,})命令的第1个参数 au 是上传的二进制声音 文件; 'wav'是上传的文件格式,百度支持 mp3、pcm、m4a、amr 等单声道、16 位深、16000Hz 采样率的声频文件,dev_pid 是语言类型,1537 表示中文普通话。

5.13.3 人脸识别

1. 创建人脸识别应用

依次选择【导航栏】→【人脸识别】→【创建应用】→【应用名称】,可以任意填写,例如 rlsb→【去领取】→【接口类别】全选,如果之前没有完成实名认证,则无法领取,实名认证完 成后刷新网页,再选择【0元领取】,其他步骤与创建文字识别应用的步骤类似,最后将人脸 识别 key 保存为 keyface.txt 文件。

2. 本地图像人脸的颜值打分

对计算机上的图像人物进行"颜值"打分,代码如下:

#//第5章/5.44.py	
from aip import AipFace	
import base64	
<pre>aipFace = AipFace('AppID', 'API Key','Secret Key')</pre>	♯加载 key
with open('baid/2.png', "rb") as fp:	
<pre>base64_data = str(base64.b64encode(fp.read()),'UTF - 8')</pre>	#编码
<pre>r = aipFace.detect(base64_data, "BASE64", {"face_field":"beauty"})</pre>	#调用
<pre>print(r["result"]["face_list"][0]['beauty'])</pre>	#结果

运行结果如下:

79.86

3. 网络图像人脸颜值打分

对网络上的图像人脸进行"颜值"打分,代码如下:

```
#//第 5 章/5.45.py
from aip import AipFace
aipFace = AipFace('AppID', 'API Key', 'Secret Key') #加载 key
im = "http://n1.itc.cn/img8/wb/recom/2017/04/19/149256623627782055.JPEG"
op = {"face_field":"age,beauty", "max_face_num": 4}
result = aipFace.detect(im, "URL", op) # 请求
face_num = result['result']['face_num']
for num in range(0, int(face_num)):
    location = result['result']['face_list'][num - 1]
    beauty = location['beauty']
    age = location['age"]
    print(age,beauty) # 年龄、颜值
```

运行结果如下:

23 70.89 22 80.09 16 53.57

op={"face_field":"age, beauty", "max_face_num":4}中的参数"face_field":"age, beauty"表示年龄、颜值, "max_face_num":4表示1幅图像中最多检测4张人脸。

程序默认按脸的大小顺序进行打分。学完第 11 章 OpenCV 后,分值可以直接绘制到 图像上。

人脸检测还有性别、人脸数量、发笑程度、戴眼镜类型、情绪、口罩等55项检测内容。

4. 人脸对比

现在的化妆技术十分强大,男性能化妆成女性,就让百度 AI 来分辨一下吧,代码如下:

#//第 5 章/5.46.py	
from aip import AipFace	
import base64	
client = AipFace('AppID', 'API Key','Secret Key')	井加载 Key
<pre>def openfile(file):</pre>	#打开图像函数
with open(file, 'rb') as f:	
<pre>data = base64.b64encode(f.read())</pre>	
<pre>image = str(data, 'UTF - 8')</pre>	
return image	
<pre>def face detect(filepath1, filepath2):</pre>	#人脸对比函数
<pre>img1 = openfile(filepath1)</pre>	#打开图像1
img2 = openfile(filepath2)	出打开图像 2
result = client.match([{	#人脸对比
'image': imgl.	
'image type': 'BASE64'.	
}	
1	
l 'image': img2	
image turne'. 'BASE6/'	
linge_type . Dromov4 ,	
if rogult["orror mag"] in "SUCCESS".	世加里近回结里或功
ii iesuit[eiioi_msg] iii Success .	# 如禾返酉垍禾成功
score - result result j[score]	+
11 Score > /U:	+11-17-11年
print("相似度:{} 定问一人iormat(score))	
print('相似度:{} 个走问一人.'.tormat(score))	
<pre>face detect('baid/f1.ipg'.'baid/f2.ipg')</pre>	#调用程序

运行结果如下:

相似度:78.56130981 是同一人。

图像还可以是网上的 URL 链接指向的图像,还可以指定照片类型等。

5. 人脸搜索

进行人脸搜索需要先向人脸库上传人脸数据,然后用程序对照片与人脸库的照片进行 对比,返回结果。选择【人脸检测】→【应用列表】→【查看人脸库】→【新建组】→【新建用户】, 这样就可以上传照片了。人脸搜索,代码如下:

```
#//第5章/5.47.pv
import base64
from aip import AipFace
client = AipFace('AppID', 'API Key', 'Secret Key')
                                                         #加载 Kev
with open('baid/s2.jpg', 'rb') as f:
                                                         #图像转码
    data = str(base64.b64encode(f.read()), 'UTF - 8')
result = client.search(data, "BASE64", 'mr')
                                                         #提交
if result["error msg"] in "SUCCESS":
                                                         #提取结果
    score = result["result"]["user_list"][0]["score"]
                                                        #获取分值
    user_id = result["result"]["user_list"][0]["user_id"] # 获取 id
    if score > 80:
        print(user id)
        if user id == '7':
                                                         #转换为人名
           print('潘建伟')
        if user_id == '8':
           print('爱因斯坦')
    else:
        print("没有找到此人")
```

查询结果如下:

8 爱因斯坦

search()函数的第1个参数 data 是照片数据,第2个参数 ABSE64 是编码类型,第3个 参数是在哪个组查询,笔者此处的查询组是 mr。返回查询到的人脸库编号,这里简单地对 返回的7号和8号进行了判断。

分组、上传、修改、删除等功能都可以通过程序完成,而不需要登录网站。

5.13.4 文本纠错

百度自然语言分析功能有词法分析、词法分析(定制版)、中文词向量表示、词义相似度、 短文本相似度、依存句法分析、中文 DNN 语言模型、情感倾向分析、情感倾向分析(定制版)、评论观点抽取、评论观点抽取(定制版)、对话情绪识别、文本纠错、文章分类、文章标签、 新闻摘要、地址识别、智能作诗、智能对联等。

1. 创建自然语言应用

依次选择【导航栏】→【机器翻译】→【创建应用】,其他步骤与创建文字识别应用的步骤 类似,最后将自然语言 key 保存为 keyyuyan. txt 文件。

2. 文本纠错

文本纠错功能可以提供正确示例参考,代码如下:

#//第5章/5.48.py	
from aip import AipNlp	
client = AipNlp('AppID', 'API Key','Secret Key')	♯加载 key
result = client.ecnet('百度是一家人工只能公斯')	#请求
<pre>for item in result.items():</pre>	
<pre>if item[0] == 'item':</pre>	
<pre>if float(item[1]['score']) > 0 and float(item[1]['score']</pre>]) < 1:
print('百度是一家人工只能公斯')	#打印原文
<pre>print(item[1]['correct_query'])</pre>	#打印纠错后

运行结果如下:

百度是一家人工只能公斯百度是一家人工智能公司

文本纠错的更多用法,详见10.7节。

百度的自然语言处理还有词法分析 lexer()、情感倾向分析 sentimentClassify()、文章 分类 topic()、对话情绪识别 emotion()、新闻摘要 newsSummary()等功能,只要把 ecnet() 换成相应方法即可。

5.13.5 图像增强与特效

图像增强与特效技术能对质量较低的图像进行去雾、对比度增强、无损放大等多种优化 处理,重建高清图像,并提供黑白图像上色、图像风格转换、人物动漫化风格等多个图像特效 API能力。

1. 创建图像增强与特效应用

依次选择【导航栏】→【图像增强与特效】→【创建应用】,其他步骤与创建文字识别应用的步骤类似,最后将图像增强与特效 key 保存为 keying.txt 文件。

2. 黑白图像上色

黑白图像上色功能可以把黑白照片处理成彩色照片,代码如下:

#//第5章/5.49.py	
import base64	
from aip import AipImageProcess	
<pre>client = AipImageProcess('AppID', 'API Key', 'Secret</pre>	Key')
with open('baid/hb.jpg', 'rb') as f:	
<pre>image = f.read()</pre>	#打开图像
<pre>img = client.colourize(image)</pre>	#黑白上色
<pre># img = client.imageQualityEnhance(image)</pre>	#无损放大
<pre># img = client.contrastEnhance(image)</pre>	#对比度
<pre># img = client.imageDefinitionEnhance(image)</pre>	#清晰度
<pre>imgData = base64.b64decode(img['image'])</pre>	#转码
with open('baid/s3.jpg',mode = 'wb') as t:	#保存
t.write(imgData)	

图像处理结果如图 5-29 所示。



(a) 处理前

(b) 处理后

图 5-29 图像处理

图像增强与特效应用还有很多其他功能,例如图像无损放大 imageQualityEnhance()、 图像去雾 dehaze()、图像对比度增强 contrastEnhance()、拉伸图像恢复 stretchRestore() 等,只要把 5.49.py 文件中的 colourize()换成相应的方法即可。

5.14 语音与文件互转

本节介绍免费的语音技术。

1. 播放声频文件

Python 内置库 winsound 用于播放声频文件,代码如下:

import winsound winsound.PlaySound('baid/record.wav', winsound.SND_FILENAME)

winsound 只能播放标准格式的 WAV 声频文件,不能播放 MP3 格式的声频文件,播放 更多声频格式及播放控制,详见 10.10 节。

2. 语音播报文字

Python 内置库 win32com 可以用语音形式播报文字内容,代码如下:

```
from win32com.client import Dispatch
speaker = Dispatch('SAPI.SpVoice') 
speaker.Speak('100 分') 
del speaker
```

#传入参数实例化
#播放语音
#释放资源

3. 生成语音文件

pyttsx3 库可以播报文本内容,也可以生成语音文件,安装 pyttsx3 库的命令如下:

pip install pyttsx3

示例代码如下:

```
#//第5章/5.50.py
import pyttsx3 as pyttsx
engine = pyttsx.init()
                                         #初始化
engine.setProperty('rate', 115)
                                         #设置语音播报速度
engine.setProperty('volume', 1.0)
                                         #设置音量 0~1
voices = engine.getProperty('voices')
                                         #获取当前音量
engine.setProperty('voice', voices[0].id)
                                         #设置为女中音
engine.say('测试结束')
                                         #播报文本
engine.save_to_file('测试结束', 'test.mp3')
                                         #保存为文件
engine.runAndWait()
```

运行程序后先播报语音,然后生成语音文件 test. mp3。

4. 用话筒录音

录音需要 Python 的 Pyaudio 库和 Wave 库, Wave 是内置库, Pyaudio 是第三方库, 安装 命令如下:

pip install pyaudio

```
代码如下:
```

#//第 5 章/5.51.py	
import wave	
import time	
from pyaudio import PyAudio, paInt16	
<pre>def savefile(data):</pre>	#保存参数设置函数
<pre>wf = wave.open('baid/record.wav', 'wb')</pre>	
wf.setnchannels(1)	#声道
wf.setsampwidth(2)	#采样宽度2字节
wf.setframerate(16000)	#采样率
<pre>wf.writeframes(b''.join(data))</pre>	
wf.close()	
def record():	#录音函数
pa = PyAudio()	#初始化
<pre>stream = pa.open(format = paInt16, channe</pre>	ls = 1, rate = 16000, input = True,
<pre>frames_per_buffer = 2)</pre>	#加载参数录音
buf = []	
<pre>t = time.time()</pre>	
print('正在录音')	
<pre>while time.time() < t + 4:</pre>	#4s空白,结束录音
audiodata = stream.read(2000)	#读取 2000 字节
buf.append(audiodata)	
print('录音结束.')	
<pre>savefile(buf)</pre>	#调用保存函数
<pre>stream.close()</pre>	
record()	#调用录音函数

运行程序开始录音,如果超过4s空白,则停止录音。

5. 语音文件转文字

语音文件转文字有一个开源的库 SpeechRecognition,首先要安装两个库,命令如下:

```
pip install PocketSphinx
pip install SpeechRecognition
```

然后,下载中文模型,网址为 https://sourceforge.net/projects/cmusphinx/files/Acoustic% 20and%20Language%20Models/,如图 5-30 所示。

\rightarrow C $\widehat{\odot}$ https://sourceforge.net/p	jects/cmusphinx/files/Acoustic%20and%20Language%20Models/
SOURCE FORGE	Open Source Software Business So
Download Latest Version sphinx4-Sprealpha-src.zip (41.3 MB)	iet Updates
Home / Acoustic and Language Models	
Name 🗢	Modified 🗢 🛛 Size 🗢
J Parent folder	
🗀 Mexican Spanish	2019-08-24
D Portuguese	2019-08-23
🗅 Mandarin	2019-02-15
Archive	2019-02-15

图 5-30 SpeechRecognition 官网

选择 Mandarin 文件夹,进入如图 5-31 所示的界面。

Download Latest Version sphinx4-5prealpha-src.zip (41.3 MB)	Set Updates	
Home / Acoustic and Language Models / Mandarin		
Name 🗢	Modified 🖨	Size ≑
J Parent folder		
LICENSE	2019-02-15	1.5 kB
cmusphinx-zh-cn-5.2.tar.gz	2019-02-15	54.0 ME
Totals: 2 Items		54.0 ME

图 5-31 SpeechRecognition 下载

选择 cmusphinx-zh-cn-5.2. tar. gz 进行下载。在 Python 安装目录下找到 Lib\sitepackages\speech_recognition\pocketsphinx-data 目录,进入 pocketsphinx-data 文件夹,并 新建文件夹 zh-CN,在这个文件夹内添加刚刚解压的文件,把解压出来的 zh_cn. cd_cont_ 5000 文件夹重命名为 acoustic-model,将 zh_cn. lm. bin 重命名为 language-model. lm. bin, 并将 zh_cn. dic 重命名为 pronounciation-dictionary. dict,如图 5-32 所示(或者直接将本章 目录内 zh-CN 文件夹复制到 pocketsphinx-data 目录下)。

^		
名称	修改日期	类型
🖹 acoustic-model	2022/4/24 10:16	文件夹
language-model.Im.bin	2019/10/8 10:07	BIN 文件
LICENSE.txt	2019/10/8 10:04	文本文档
pronounciation-dictionary.dict	2019/10/8 10:04	DICT 文件

图 5-32 重命名文件

```
示例代码如下:
```

```
#//第5章/5.52.py
import speech_recognition as sr
r = sr.Recognizer() #实例化对象
with sr.AudioFile('test.mp3') as source: #打开语音文件
    audio = r.record(source)
print('文本内容:',r.recognize_sphinx(audio, language = 'zh-CN'))
```

运行结果如下:

文本内容:测试结束 进程已结束,退出代码 0

SpeechRecognition 库对于计算机合成的语音识别率极高,对于一般录音识别率一般, 提高识别率需要搭建人工智能框架,初学者可以首先从开源项目学习,详见 18.6节。

5.15 Python 压缩文件和文件夹

工作中经常需要把许多文件、文件夹打包,最简单的压缩方法是直接调用 Windows 程^{2min} 序 WinRAR 进行压缩。

1. Path 路径添加

首先要保证计算机内安装了 WinRAR. exe 程序,然后把安装目录加入 path 路径中。 方法如下:在╋【开始】菜单中找到 WinRAR,打开它的子菜单,如图 5-33 所示。

右击 WinRAR,选择【更多】,选择【打开文件所在位置】,右击 WinRAR 后弹出菜单,选择【更多】,选择【打开文件所在位置】,右击 WinRAR,选择【属性】,如图 5-34 所示。

		🜆 WinRAR.exe 🛛	属性	×
		常规 兼容性	数字签名 安全 详细信息 以前的版本	
		: 5	WinRAR.exe	
		文件类型:	应用程序 (.exe)	
		描述:	WinRAR 压缩文件管理器	
		位置:	D:\Program Files\WinRAR	
		大小:	2.22 MB (2,331,080 字节)	
	Windows 权松/法田	占用空间:	2.22 MB (2,334,720 字节)	
R	Windows 系统	创建时间: 修改时间:	2021年10月3日, 10:29:56 2019年12月17日, 18:48:36	
3	WinRAR	访问时间:	2022年3月12日, 18:26:09	
×i	WinRAR	属性:	□只读(R) □ 隐藏(H) 高级(D)	
	WinRAR 中文帮助			
Š3	控制台 RAR 中文手册			
12	最新版本里有哪些新功能			
	00 1 4		确定取消	过用(A)
	图 5-33 开始菜单		图 5-34 WinRAR 属性	

复制【位置】内容,例如笔者此处为 D:\Program Files\WinRAR,打开控制面板,搜索 path,如图 5-35 所示。

单击【编辑账户的环境变量】按钮后,出现的界面如图 5-36 所示。

双击 Path,如图 5-37 所示。

单击【新建】按钮,将 WinRAR 的路径粘贴到这里,然后单击【确定】按钮。

2. 压缩程序所在目录的所有文件

压缩程序所在目录的所有文件,代码如下:

```
import os
os.system("WinRAR.exe a data.rar * ")
```

system()的参数 WinRAR. exe 用于指定压缩程序,参数 a 表示指定压缩的参数, data. rar 是压缩后的文件, * 表示压缩当前目录的所有文件。运行结果是将当前目录下的所有文

Q置						Windows 设置	
				path	М		×
					编辑系统	环境变量	
旦	系统 显示、声音、通知、电源		设备 蓝牙、打印机、鼠标		编辑账户	的环境变量 连接 Android 设备和 iPhone	€
	放用 卸板、默认应用、可选功能	8	院户 你的陈户、电子邮件、同步设 置、工作、家庭		色 A字	时间和语言 语言、区域、日期	⊗
Q	搜索 直线我的文件、权限	A	隐私 位置、相机、麦克风		C	更新和安全 Windows 更新、恢复、备份	

图 5-35 控制面板

变量	值
OneDrive	C:\Users\LENOVO\OneDrive
OneDriveConsumer	C:\Users\LENOVO\OneDrive
Path	C:\Users\LENOVO\AppData\Local\Programs\Python\Python36-32
PyCharm Community Editio	n C:\Program Files\JetBrains\PyCharm Community Edition 2021.3.2\
TEMP	C:\Users\LENOVO\AppData\Local\Temp
FR 100	Cillingen I ENOVOLAnn Datallaca B Tamp

图 5-36 环境变量

訂环境变量	
C:\Users\LENOVO\AppData\Local\Programs\Python\Python36-32\Sc	新建(N)
C:\Users\LENOVO\AppData\Local\Programs\Python\Python36-32\	
C:\python36-32\Scripts\	编辑(E)
C:\python36-32\ P(USEDDDOEILEP()App Data)Lass/NAissass(NAissass(NAissass)Apps	
%USERPROFILE%\AppData\Local\microsoft\windowsApps	浏览(B)
Cillsercil ENOVO)anaconda3\Scrints	
D/Program Files/WinPAP	删除(D)
	下移(0)
	编辑文本(T)
确定	取消

图 5-37 编辑环境变量

件压缩为 data. rar 文件。

3. 压缩指定目录

压缩指定目录,代码如下:

import os
os.system("WinRAR.exe a data1.rar pic")

把 pic 整个目录压缩成 data1. rar 文件。

4. 压缩多目标

压缩多目标,代码如下:

```
import os
os.system("WinRAR.exe a data.rar * pic pic2")
```

os. system("WinRAR. exe a data. rar * pic pic2")命令的参数"*"表示程序所在目录的所有文件,参数 pic 表示 pic 文件夹、参数 pic2 表示 pic2 文件夹,参数 data. rar 表示压缩 成 data. rar 文件。

5. 绝对路径与相对路径

绝对路径就是带盘符的完整路径,例如 4.5 节的"C:/Windows/Fonts/SimHei.ttf"。 相对路径就是相对于程序所在目录的路径,常用的相对路径有以下几种形式。

1) 同一级

"1. txt"或"./1. txt"表示程序所在目录的 1. txt 文件,例如 5. 52. py 文件中的"test. mp3"。

2) 下一级

"./t/1.txt"或者"t/1.txt"表示下一级的t目录下的1.txt文件,例如5.2节的"txt/1.txt"。

3) 上一级

".../t/1. txt"表示上一级的 t 目录下的 1. txt 文件。

4) 上上一级

"../../t/1.txt"表示上一级目录的上一级 t 目录下的 1.txt 文件。

上例中,Python 查找相对路径 WinRAR. exe 程序的顺序是:应用程序所在目录→ Windows SYSTEM 目录→Windows 目录→Path 环境变量指定的路径,所以我们经常把程 序需要的文件复制到程序所在的目录,例如 5.52. py 文件中的 test. mp3;很多项目会将引 用的程序写入 Path 环境变量,例如本节的 WinRAR. exe,9.10 节的 sqlite3. exe。