



# Python程序设计

## 第七章 办公自动化

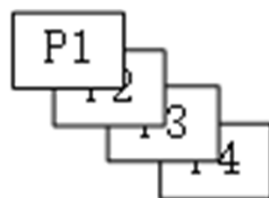
霍佳皓 博士

北京科技大学计算机与通信工程学院

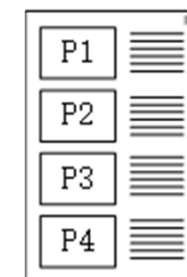


## 作业4：自动完成的笔记本

教师每次下发课件是PDF格式，由若干页（N页）横版的PPT组成，例如：



这样的格式不方便做笔记，于是你希望完成一个软件，将上述PDF文件转成下面的形式的PDF，共计N/4页（如不能整除则向上取整），每页的效果如图所示：



即每页（A4版面）左侧放置4版的课件（若不足4版则下部为空白），右侧放置多条记笔记的横线，右上方为条码，内容为自己的学号（略去学号中的字母，不足补零）。完成上述代码功能，同时完成较为详细的软件设计文档（最终提交pdf即可），文档包括设计思路、关键代码解释、效果图等。以上文件打包，上传到CG系统。 **禁止学术不端行为。**



## 回顾

- OOP的思想
- 类：封装、继承和多态
- 类与实例
- 类的初始化
- 鸭子类型



## »» 目录

---

- Excel自动化
- Word自动化
- PDF自动化
- 图像处理自动化
- 键盘鼠标自动化
  
- 目标: **简化日常办公操作。**



## »» 参考资料

- 基于Python的高效编程，使得繁琐工作自动化





## 作者前言

- 21世纪早期，我的大学室友在一个电子产品零售商店工作。
- 商店偶尔会收到一份电子表格，其中包含竞争对手的数千种产品的价格。
- 由3个员工组成的团队，会将这个电子表格打印在一叠厚厚的纸上，然后3个人分一下。
- 针对每个产品价格，他们会查看自己商店的价格，并记录竞争对手价格较低的所有产品。
- 这通常会花几天的时间。



## 作者前言

- “我会写一个程序来做这件事。” 我的室友告诉他们。
- 几个小时后，他写了一个简短的程序，从文件读取竞争对手的价格，在商店的数据库中找到该产品，并记录竞争对手是否更便宜。
- 他当时还是编程新手，花了许多时间在了一本编程书籍中查看文档。实际上程序只花了几秒钟运行。
- 这就是计算机编程的威力。计算机就像瑞士军刀，可以用来完成数不清的任务。许多人花上数小时点击鼠标和敲打键盘，执行重复的任务，却没有意识到，如果他们给机器正确的指令，机器就能在几秒钟内完成他们的工作。



## 》》》 可能的任务

- 你能将下面这样一些简单的任务自动化：
  - 移动并重命名几千个文件，分类放入文件夹；
  - 填写在线表单；
  - 在网站更新时，从网站下载文件或复制文本；
  - 让计算机向客户发出短信通知；
  - 更新或格式化 Excel 电子表格；
  - 检查电子邮件并发出预先写好的回复。
- 对人来说，这些任务简单，但很花时间。它们通常很琐碎、很特殊，没有现成的软件可以完成。有一点编程知识，就可以让计算机为你完成这些任务。





# Excel自动化

## »» Excel

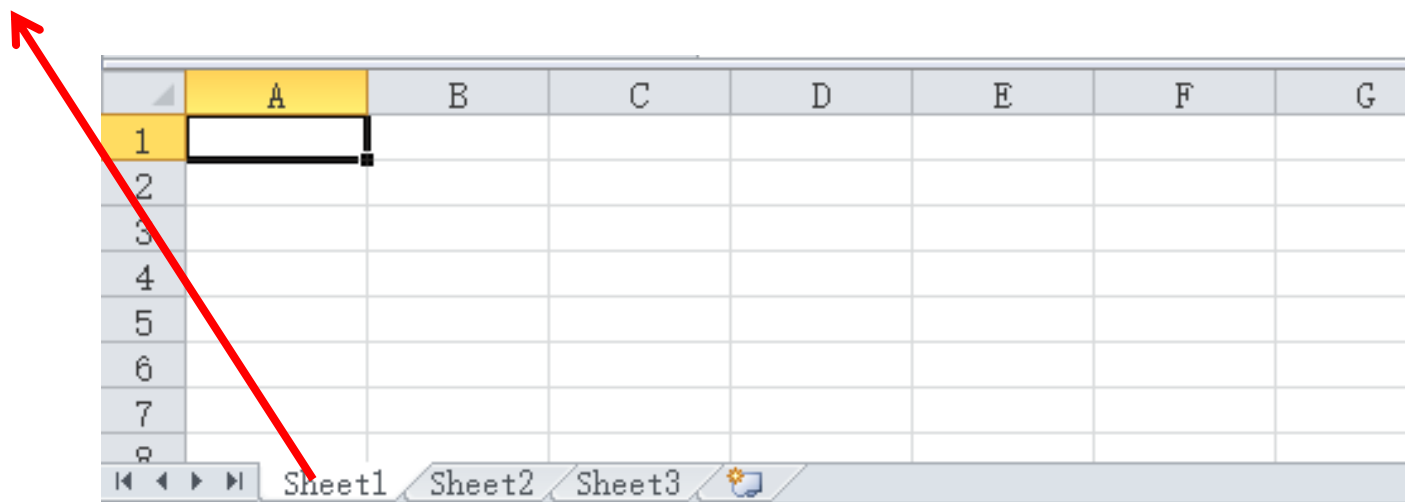


- Excel 是 Windows 环境下流行的、强大的电子表格应用。
  - 包括LibreOffice Calc, OpenOffice Calc, WPS等软件也可以读写xlsx格式。
- openpyxl 模块让 Python 程序能读取和修改 Excel 电子表格文件。
- Excel 是 Windows 环境下流行的、强大的电子表格应用。
- <https://openpyxl.readthedocs.io>
- 安装: `pip install openpyxl`



## Excel基本概念

- 一个Excel 电子表格文档称为一个工作簿 (Workbook) 。
- 一个工作簿保存在扩展名为.xlsx 的文件中。
- 每个工作簿可以包含多个表 (也称为工作表Worksheet) 。





## 读取表格

- 假设存在Excel文件：example.xlsx

1	A	2019-4-1	Li Lei	52	➔
		2019-4-2	Han Meimei	300	
		2019-4-5	Li Lei	230	
		2019-4-8	Li Lei	170	
		2019-4-10	Han Meimei	96	



## 》》 取得工作簿及工作表

```
>>> import openpyxl
>>> wb = openpyxl.load_workbook('example.xlsx')
>>> wb.sheetnames
['Sheet1', 'Sheet2', 'Sheet3']
>>> sheet = wb['Sheet3']
>>> sheet
<Worksheet "Sheet3">
>>> type(sheet)
<class 'openpyxl.worksheet.worksheet.Worksheet'>
>>> sheet.title
'Sheet3'
>>> anotherSheet = wb.active
>>> anotherSheet
<Worksheet "Sheet1">
```

打开工作簿

获取所有sheet  
的名称

按名称或按照活跃  
状态取得sheet对象。





## 》》》 取得单元格

	A	B	C
1	2019-4-1	Li Lei	52
2	2019-4-2	Han Meimei	300
3	2019-4-3	Li Lei	300

```
>>> sheet = wb['Sheet1']
>>> sheet['A1']
<Cell Sheet1.A1>
>>> sheet['A1'].value
datetime.datetime(2019, 4, 1, 0, 0)
>>> c = sheet['B1']
>>> c.value
'Li Lei'
>>> f'Row {c.row}, Column {c.column} = {c.coordinate}'
'Row 1, Column B = B1'
>>> sheet['C1'].value
52
```



## 》》》 取得单元格

	A	B	C
1	2019-4-1	Li Lei	52
2	2019-4-2	Han Meimei	300
3	2019-4-3	Li Lei	52

```
>>> for i in range(1, 5):  
    print(i, sheet.cell(row=i, column=2).value)
```

```
1 Li Lei  
2 Han Meimei  
3 Li Lei  
4 Li Lei
```

```
>>> print(sheet.max_row, sheet.max_column)
```

```
5 3
```

cell对象也可以通过行、列的序号来访问，均从1开始。

通过max\_row属性获得最大的行，通过max\_column属性获得最大的列。



## 》》》 取得矩形区域

	A	B	C
1	2019-4-1	Li Lei	52
2	2019-4-2	Han Meimei	300
3	2019-4-3	...	...

```
>>> for rowOfCellObjects in sheet['A1':'C3']:  
    for cellObj in rowOfCellObjects:  
        print(cellObj.coordinate, cellObj.value)  
    print('--- END OF ROW ---')
```

A1 2019-04-01 00:00:00

--- END OF ROW ---

B1 Li Lei

--- END OF ROW ---

C1 52

--- END OF ROW ---

...

B3 Li Lei

--- END OF ROW ---

C3 230

--- END OF ROW ---

支持按行、列次序访问一组对象。





## 》》 取得行或者列：支持循环迭代

```
>>> for i in sheet.rows:
    print(i)
(<Cell 'Sheet1'.A1>, <Cell 'Sheet1'.B1>, <Cell 'Sheet1'.C1>)
(<Cell 'Sheet1'.A2>, <Cell 'Sheet1'.B2>, <Cell 'Sheet1'.C2>)
(<Cell 'Sheet1'.A3>, <Cell 'Sheet1'.B3>, <Cell 'Sheet1'.C3>)
(<Cell 'Sheet1'.A4>, <Cell 'Sheet1'.B4>, <Cell 'Sheet1'.C4>)
(<Cell 'Sheet1'.A5>, <Cell 'Sheet1'.B5>, <Cell 'Sheet1'.C5>)

>>> for i in sheet.columns:
    print(i)
(<Cell 'Sheet1'.A1>, <Cell 'Sheet1'.A2>, <Cell 'Sheet1'.A3>, <Cell
'Sheet1'.A4>, <Cell 'Sheet1'.A5>)
(<Cell 'Sheet1'.B1>, <Cell 'Sheet1'.B2>, <Cell 'Sheet1'.B3>, <Cell
'Sheet1'.B4>, <Cell 'Sheet1'.B5>)
(<Cell 'Sheet1'.C1>, <Cell 'Sheet1'.C2>, <Cell 'Sheet1'.C3>, <Cell
'Sheet1'.C4>, <Cell 'Sheet1'.C5>)
```

## 综合练习

- 编写程序，统计各位同学（列B）的日常总开销（列C）。

2019-4-1	Li Lei	52
2019-4-2	Han Meimei	300
2019-4-5	Li Lei	230
2019-4-8	Li Lei	170
2019-4-10	Han Meimei	96





## »» 参考答案

```
import openpyxl
wb = openpyxl.load_workbook('example1.xlsx')
sh = wb.active
pay_sum = {}
for row in sh.rows:
    fields = [c.value for c in row]           读出一行的所有值
    name = fields[1]
    pay = fields[2]
    pay_sum[name] = pay_sum.get(name, 0) + pay
print(pay_sum)                             检索name, 默认为0
```



## 数据结果的保存

- 保存在txt文件中
- 保存在csv文件中
- 保存在py文件中

```
import pprint
resultFile = open('result.py', 'w')
resultFile.write(
    'pay_sum = ' + pprint.pformat(pay_sum)
)
resultFile.close()
```

生成的result.py文件如何读取?



**result.py** `pay_sum = {'Han Meimei': 396, 'Li Lei': 452}`



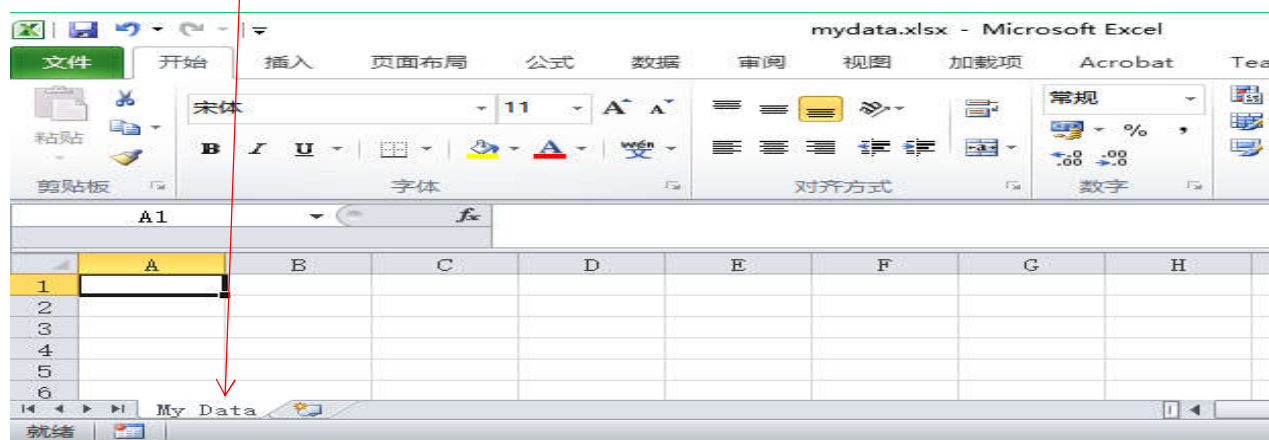
## 创建空白工作簿和表

```
import openpyxl
wb = openpyxl.Workbook()
print(wb.sheetnames)
sh = wb.active
sh.title = "My Data"
wb.save("mydata.xlsx")
```

创建空白工作簿

修改工作表的标题

保存为xlsx文件





## 创建和删除工作表

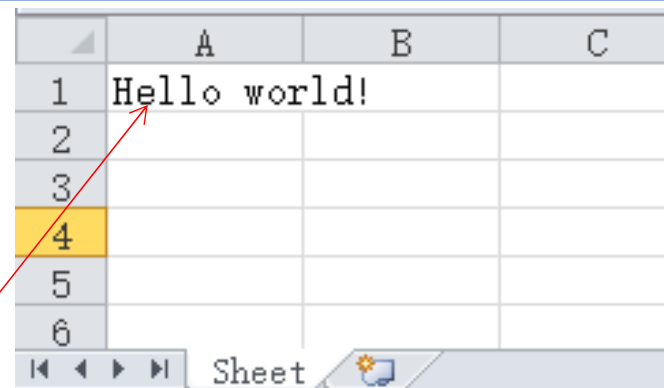
```
import openpyxl
>>> wb = openpyxl.Workbook()
>>> wb.get_sheet_names()
['Sheet']
>>> sh1 = wb.create_sheet()
<Worksheet "Sheet1">
>>> wb.get_sheet_names()
['Sheet', 'Sheet1']
>>> wb.create_sheet(index=0, title='First Sheet')
<Worksheet "First Sheet">
>>> wb.get_sheet_names()
['First Sheet', 'Sheet', 'Sheet1']
>>> wb.remove_sheet(sh1)
>>> wb.get_sheet_names()
['First Sheet', 'Sheet']
```

创建空白工作表

在给定的位置，创建指定名称的工作表

删除sheet对象对应的表

## 将值写入单元格



	A	B	C
1	Hello world!		
2			
3			
4			
5			
6			

```
import openpyxl
wb = openpyxl.Workbook()
sheet = wb.active
sheet['A1'].value = 'Hello world!'
wb.save("hello.xlsx")
```

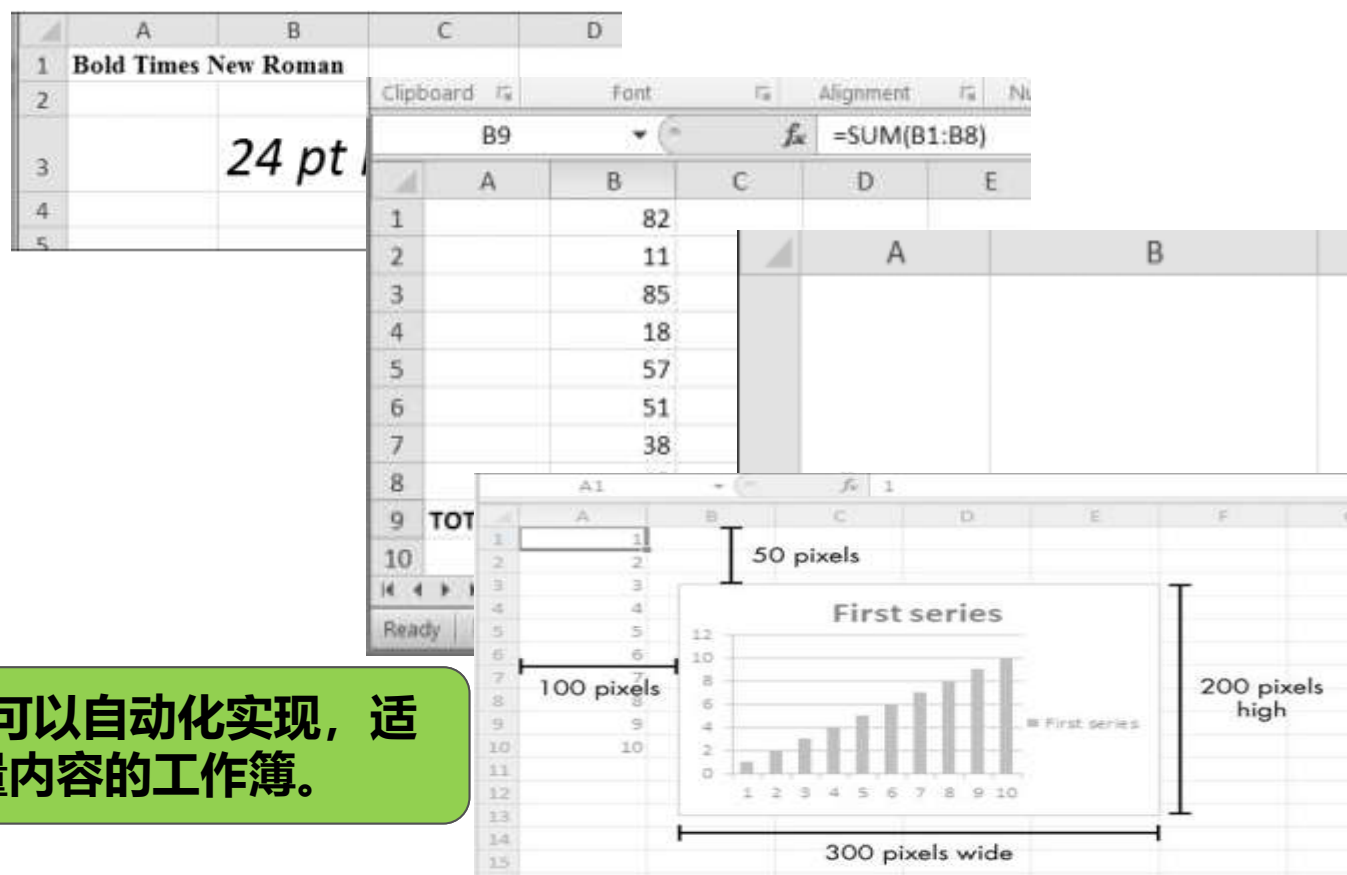
对单元格直接赋值即可。

此处对 `sheet['A1'].value` 赋值或对 `sheet['A1']` 赋值的效果相同，但一般应写上 `.value`



## 其他功能

- 改变字体
- 使用公式
- 设置行高和列宽
- 单元格拆分
- 绘制图表



Excel的大部分功能都可以自动化实现，适合从数据库中生成批量内容的工作簿。



## 综合练习

- 编写程序，生成一个九九乘法表。

	A	B	C	D	E	F	G
1		1	2	3	4	5	6
2	1	1	2	3	4	5	6
3	2	2	4	6	8	10	12
4	3	3	6	9	12	15	18
5	4	4	8	12	16	20	24
6	5	5	10	15	20	25	30
7	6	6	12	18	24	30	36





## »» 参考答案

	A	B	C	D	E	F	G	H	I	J
1		1	2	3	4	5	6	7	8	9
2	1	1	2	3	4	5	6	7	8	9
3	2	2	4	6	8	10	12	14	16	18
4	3	3	6	9	12	15	18	21	24	27
5	4	4	8	12	16	20	24	28	32	36
6	5	5	10	15	20	25	30	35	40	45
7	6	6	12	18	24	30	36	42	48	54
8	7	7	14	21	28	35	42	49	56	63
9	8	8	16	24	32	40	48	56	64	72
10	9	9	18	27	36	45	54	63	72	81

```
import openpyxl
wb = openpyxl.Workbook()
sheet = wb.active
for i in range(1, 10):
    sheet.cell(row=i+1, column=1).value = i
    sheet.cell(row=1, column=i+1).value = i
for i in range(1, 10):
    for j in range(1, 10):
        sheet.cell(row=i+1, column=j+1).value=i*j
wb.save("99.xlsx")
```

边上的数字填入第一行  
和第一列。

乘积填入相应的单元格<sup>26</sup>

## 综合练习

- 还记得刚才python存储的.py文件的数据么?
- 如何把result.py文件存储到xlsx中?

**result.py** `pay_sum = {'Han Meimei': 396, 'Li Lei': 452}`



	A	B
1	Han Meimei	396
2	Li Lei	452



## »» 参考答案

## Meta Programming



```
import openpyxl  
import result
```

用import即可导入，因为存储的文本也是合法的python语句。

```
wb = openpyxl.Workbook()  
sheet = wb.active
```

```
row = 1  
for k,v in result.pay_sum.items():  
    sheet.cell(row=row, column=1).value = k  
    sheet.cell(row=row, column=2).value = v  
    row += 1  
wb.save("result.xlsx")
```

逐行插入key和value。



# Word自动化



## Word

- Microsoft公司的Word是一套功能丰富的办公软件，所提供的docx格式可以存储文本以及字体、颜色等丰富的信息。
- docx的读写比txt文件要复杂，需要第三方库的支持。
- `pip install python-docx` *安装的名称和导入的名称可以不一致。*
- 注意：导入的时候用 `import docx`



## docx文件结构

- docx 文件有很多结构。
- 这些结构在 python-docx 中用 3 种不同的类型来表示。
- Document 对象表示整个文档。
- Document 对象包含一个 Paragraph 对象的列表， Paragraph即文档中的段落。
- 每个 Paragraph 对象都包含一个 Run 对象的列表。



## docx文件结构

- 下图段落有 4 个 Run 对象。

A plain paragraph with some **bold** and some *italic*

Run Run Run Run

- Word 文档中的文本不仅仅是字符串。它包含与之相关的字体、大小、颜色和其他样式信息。在 Word 中，样式是这些属性的集合。
- 一个 Run 对象是相同样式文本的延续。
- 当文本样式发生改变时，就需要一个新的 Run 对象。





## »» 示例：libai.docx

### 铁杵成针

李白是唐代的大诗人，但是小时候读书并不用功。有一天，他的书读到一半，就不耐烦了：“这么厚一本书，什么时候才能读完啊！”于是他干脆不读了，把书一扔就溜出去玩。

李白快乐地跑着，忽然，他看见一位老奶奶正在磨刀石上用力地磨着一根铁棒。李白觉得很奇怪，便蹲了下来，傻傻地看了好一阵。老奶奶也不理会他，只是全神贯注地磨着。后来，李白忍不住了，问道：“奶奶，您这是干什么呢？”

“我在磨一根针来缝衣服。”老奶奶头也不抬，专心地磨。

“磨针？”李白更加奇怪了，“这么粗一根铁棒怎么能磨成针？！”

老奶奶这才抬起头来说：“孩子，铁棒再粗，我天天磨，还怕它磨不成一根针吗！”李白听了，恍然大悟，“对呀！只要有恒心，再难的事情也能做成功的，读书不也是这样吗！”

于是他便立刻转身跑回家去，拾起扔在地上的书本，专心地读，从此也不再敢偷懒了。后来他终于成了中国历史上一位伟大的诗人。



## »» 读取文档

```
import docx
doc = docx.Document('libai.docx')
print(len(doc.paragraphs))
# 8
print(doc.paragraphs[0].text)
# '铁杵成针'
print(doc.paragraphs[2].text)
# '李白是唐代的大诗人，但是小时候读书并不用功。...'
print(doc.paragraphs[2].runs)
#[<docx.text.run.Run at 0x1e0e56a3c50>, ...]
print(doc.paragraphs[2].runs[0].text)
# '李白是唐代的大诗人，'
print(doc.paragraphs[2].runs[1].text)
# '但是小时候读书并不用功'
```

打开文档

获取所有段落的数量。注意，包含空段落。

获取段落的文本。

获取Run对象的文本。



## 》》 获取Word中的纯文本

- 如果不关注样式，只希望获得所有的文本
- 参考代码实现：

```
import docx

def getText(filename):
    doc = docx.Document(filename)
    fullText = []
    for para in doc.paragraphs:
        fullText.append(para.text)
    return '\n'.join(fullText)
```



*join方法可以拼接字符串。*



## »» Word中的样式

- Word有很多样式，可以通过段落或run对象的style属性获得。
- 默认的样式包括：

'Normal'	'Heading5'	'ListBullet'	'ListParagraph'
'BodyText'	'Heading6'	'ListBullet2'	'MacroText'
'BodyText2'	'Heading7'	'ListBullet3'	'NoSpacing'
'BodyText3'	'Heading8'	'ListContinue'	'Quote'
'Caption'	'Heading9'	'ListContinue2'	'Subtitle'
'Heading1'	'IntenseQuote'	'ListContinue3'	'TOCHeading'
'Heading2'	'List'	'ListNumber'	'Title'
'Heading3'	'List2'	'ListNumber2'	
'Heading4'	'List3'	'ListNumber3'	



## Word中的样式

- 可以通过设置下列属性产生不同的样式

属性	描述
bold	文本以粗体出现
italic	文本以斜体出现
underline	文本带下划线
strike	文本带删除线
double_strike	文本带双删除线
all_caps	文本以大写首字母出现
small_caps	文本以大写首字母出现，小写字母小两个点
shadow	文本带阴影
outline	文本以轮廓线出现，而不是实心
rtl	文本从右至左书写
imprint	文本以刻入页面的方式出现
emboss	文本以凸出页面的方式出现



## 修改样式

```
import docx
doc = docx.Document('libai.docx')
doc.paragraphs[2].runs[0].underline = True
doc.save('restyled-libai.docx')
```

保存文件

李白是唐代的大诗人，但是小时候读书

李白是唐代的大诗人，但是小时候读书



## »» 生成Word文档

---

- `doc.add_paragraph(text)` 新建一个段落，返回段落对象。
- `para.add_run(text)` 新建一个run，返回run对象。
- `doc.add_heading(text, level)` 新建一个标题，层级为level
- `doc.add_picture(filename)` 新建一个图片



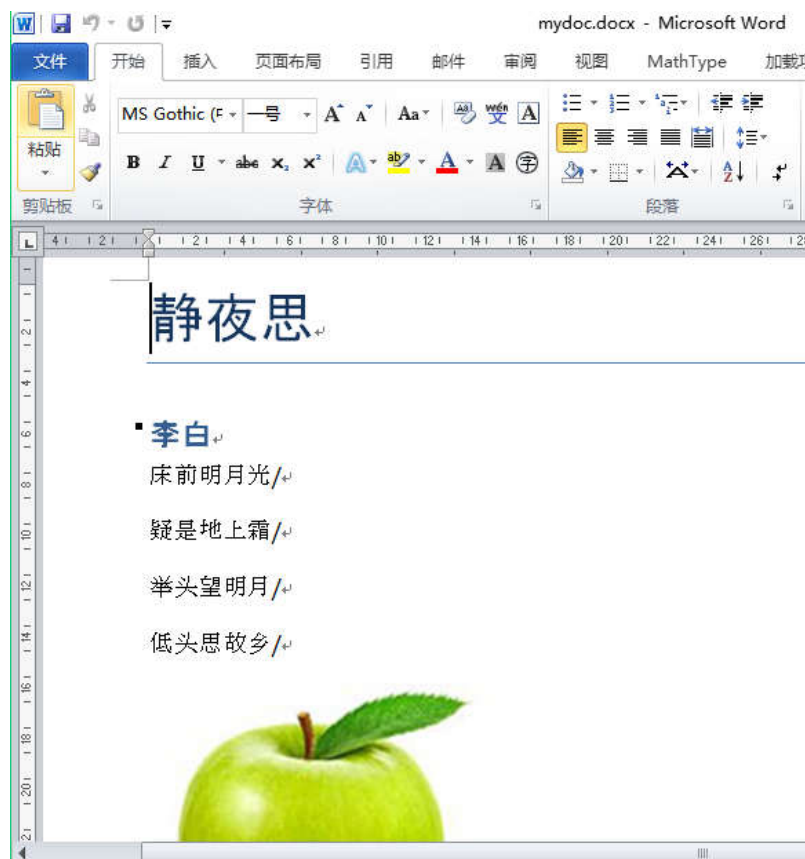
## »» 生成Word文件练习

```
import docx
doc = docx.Document()
doc.add_heading('静夜思', 0)
doc.add_heading('李白', 1)
for s in ["床前明月光", "疑是地上霜", "举头望明月", "低头思故乡"]:
    para = doc.add_paragraph(s)
    para.add_run("/")
doc.add_picture("green.jpg")
doc.save('mydoc.docx')
```





## 生成Word文件练习



运行效果如左图。

通过进一步的定制，可以  
基于你的数据，实现批量  
的Word排版。



## 综合练习

- 在Excel表格存储了班里同学的姓名和性别;
- 现在召开聚会, 要求批量生成一组Word文件, 对应于每个同学的邀请函

	A	B
1	李雷	男
2	韩梅梅	女
3	张雪	女
4	王强	男
5		
6		
7		
8		

李雷先生:  
恭请您参加2023年3月31日晚上举办的假期聚会!  
北京科技大学Python课程聚会筹备组

## »» 参考答案



	A	B
1	李雷	男
2	韩梅梅	女
3	张雪	女
4	王强	男
5		
6		
7		
8		

```
import openpyxl
import docx

wb = openpyxl.load_workbook('名单.xlsx')
sh = wb.active
name_gender = []
for row in sh.rows:
    fields = [c.value for c in row]
    name = fields[0]
    gender = fields[1]
    name_gender.append([name, gender])
```

## »» 参考答案

2



```
def makeDoc(name, gender):  
    doc = docx.Document()  
    para = doc.add_paragraph(name)  
    if gender=="男":  
        para.add_run("先生:")  
    else:  
        para.add_run("女士:")  
    doc.add_paragraph("恭请您参加2023年3月31日晚上举办的假期聚会！")  
    doc.add_paragraph("北京科技大学Python课程聚会筹备组")  
    doc.save(f'{name}.docx')  
  
for name, gender in name_gender:  
    makeDoc(name, gender)
```

李雷先生：  
恭请您参加2023年3月31日晚上举办的假期聚会！  
北京科技大学Python课程聚会筹备组

 韩梅梅.docx

 李雷.docx

 名单.xlsx

 王强.docx

 张雪.docx



# PDF自动化

# PDF



- PDF 表示 Portable Document Format 便携式文件格式，由Adobe Systems在1993年用于文件交换所发展出的文件格式。
- 本部分专注两件事：
  - 从 PDF 读取文本内容
  - 从已有的文档页面生成新的 PDF。
- `pip install pyPDF2`

## »» 典型的PDF视图

70

IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 37, NO. 1, FEBRUARY 2007

# Correspondence

## Wrapper–Filter Feature Selection Algorithm Using a Memetic Framework

Zexuan Zhu, Yew-Soon Ong, and Manoranjan Dash

**Abstract**—This correspondence presents a novel hybrid wrapper and filter feature selection algorithm for a classification problem using a memetic framework. It incorporates a filter ranking method in the traditional genetic algorithm to improve classification performance and accelerate the search in identifying the core feature subsets. Particularly, the method adds or deletes a feature from a candidate feature subset based on the univariate feature ranking information. This empirical study on commonly used data sets from the University of California, Irvine repository and microarray data sets shows that the proposed method outperforms existing

### Procedure of WFFSA

```
1 Begin
2   Initialize: Randomly generate an initial population of feature subsets;
3   While (Stopping conditions are not satisfied)
4     Evaluate all feature subsets encoded in the population;
5     For each subset chosen to undergo the local improvement process
6       Perform local search and replace it with locally improved
        solution in the spirit of Lamarckian learning;
7   End For
8   Perform evolutionary operators based on selection, crossover, and
    mutation;
9   End While
10 End
```

Fig. 1. Procedure of WFFSA.





## »» PDF读取文字

pyPDF2库不能很好的处理  
中文编码的pdf。

```
import PyPDF2
with open('04067093.pdf', 'rb') as pdfFile:
    pdfReader = PyPDF2.PdfFileReader(pdfFile)
    print("Total pages: ", pdfReader.numPages)
    page0 = pdfReader.getPage(0)
    print(page0.extractText())
```

提取所有的文本

总页数

页面对象

Total pages: 7

70IEEE TRANSACTIONSONSYSTEMS,MAN,ANDCYBERNETICSŠPARTB:CY  
BERNETICS,VOL.37,NO.1,FEBRUARY2007





## »» 合并PDF

```
import PyPDF2
pdf1File = open('00294849.pdf', 'rb')
pdf2File = open('04067093.pdf', 'rb')
pdf1Reader = PyPDF2.PdfFileReader(pdf1File)
pdf2Reader = PyPDF2.PdfFileReader(pdf2File)
pdfWriter = PyPDF2.PdfFileWriter()

for pageNum in range(pdf1Reader.numPages):
    pageObj = pdf1Reader.getPage(pageNum)
    pdfWriter.addPage(pageObj)

for pageNum in range(pdf2Reader.numPages):
    pageObj = pdf2Reader.getPage(pageNum)
    pdfWriter.addPage(pageObj)

pdfOutputFile = open('combined.pdf', 'wb')
pdfWriter.write(pdfOutputFile)
pdfOutputFile.close(); pdf1File.close(); pdf2File.close()
```



## »» 页面旋转和叠加

- rotateClockwise(0, 90, 180, 270)
- rotateCounterClockwise(...)
  - 可以将页面对象旋转
- mergePage(anotherPage)
  - 可以将页面叠加（水印）
- 综合上述功能，可以实现pdf文件的合并、页面抽取、旋转、叠印等功能。



## 综合练习

- 能否把所有的课件合并成一个大的pdf文件?
  - 提示: `glob.glob("CH*.pdf")`





## »» PDF中文编码问题

---

- pyPDF2不能读取中文编码的文字;
- 可以使用pdfminer库或其他类似的库。



## »»» PDF中文编码问题

```
import io
from pdfminer.converter import TextConverter
from pdfminer.pdfinterp import PDFPageInterpreter
from pdfminer.pdfinterp import PDFResourceManager
from pdfminer.pdfpage import PDFPage
def extract_text_from_pdf(pdf_path):
    resource_manager = PDFResourceManager()
    fake_file_handle = io.StringIO()
    converter = TextConverter(resource_manager, fake_file_handle)
    page_interpreter = PDFPageInterpreter(resource_manager, converter)
    with open(pdf_path, 'rb') as fh:
        for page in PDFPage.get_pages(fh, caching=True):
            page_interpreter.process_page(page)
            text = fake_file_handle.getvalue()
        converter.close()
        fake_file_handle.close()
    if text:
        return text

if __name__ == '__main__':
    print(extract_text_from_pdf('CH07办公自动化.pdf'))
```



## 运行效果



Python程序设计第七章办公自动化霍佳皓博士北京科技大学计算机与通信工程学院...



# 图像处理自动化



## »» 图像处理

- 图像处理已经是深入日常生活，Photoshop的缩写已经成为图像处理的代名词。
- 如果需要编辑大量的图像，手工编辑可能是漫长、枯燥的工作。
- Pillow 是一个第三方 Python 模块，用于处理图像文件。该模块包含一些函数，可以很容易地裁剪图像、调整图像大小，以及编辑图像的内容，从而轻松地自动编辑成千上万的图像。





## 颜色 and RGBA 值

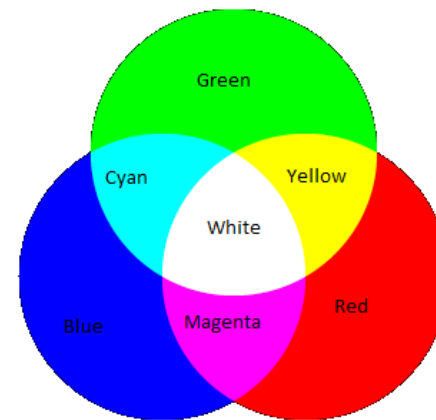
- 计算机程序通常将图像中的颜色表示为 RGBA 值。RGBA 值是一组数字，指定颜色中的红、绿、蓝和 alpha（透明度）的值。
- 这些值是从 0 到 255（最高）的整数。
- 这些 RGBA 值分配给单个像素，像素是计算机屏幕上能显示一种颜色的最小点。
- 像素的 RGB 设置准确地告诉它应该显示哪种颜色的色彩。



## Alpha通道

- 图像也有一个 alpha 值，用于生成 RGBA 值。如果图像显示在屏幕上，遮住了背景图像或桌面墙纸，alpha 值决定了“透过”颜色可以看到多少程度的背景。
- 如果alpha 值为 0，表示完全透明。
- 如果alpha 值为 255，表示完全不透明。

# RGB



名称	RGBA 值	名称	RGBA 值
White	(255, 255, 255, 255)	Red	(255, 0, 0, 255)
Green	(0, 128, 0, 255)	Blue	(0, 0, 255, 255)
Gray	(128, 128, 128, 255)	Yellow	(255, 255, 0, 255)
Black	(0, 0, 0, 255)	Purple	(128, 0, 128, 255)



## 获取颜色

```
>>> from PIL import ImageColor
>>> ImageColor.getcolor('red', 'RGBA')
(255, 0, 0, 255)
>>> ImageColor.getcolor('RED', 'RGBA')
(255, 0, 0, 255)
>>> ImageColor.getcolor('Black', 'RGBA')
(0, 0, 0, 255)
>>> ImageColor.getcolor('chocolate', 'RGBA')
(210, 105, 30, 255)
>>> ImageColor.getcolor('CornflowerBlue', 'RGBA')
(100, 149, 237, 255)
```

注: Pillow库的导入用import PIL

## »» 图像坐标系

- 左上角坐标为(0,0)
- y正方向向下
- x正方向向右

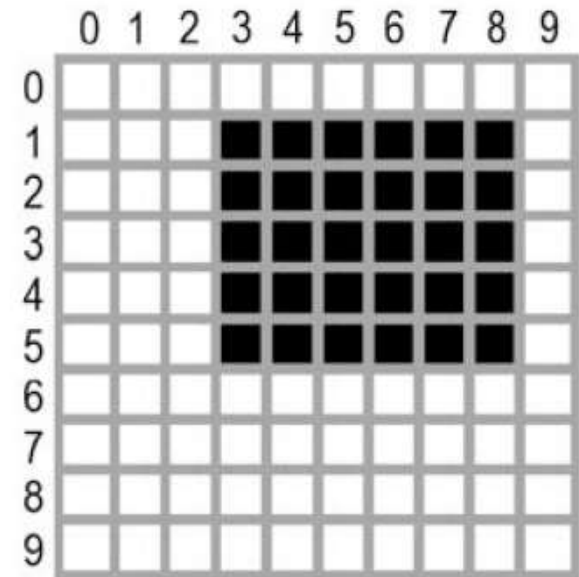




## Box

- 一个矩形元组参数，表示图像中的一个矩形区域。四个整数按顺序分别是：
  - 左：该矩形的最左边的 x 坐标。
  - 顶：该矩形的顶边的 y 坐标。
  - 右：该矩形的最右边右面一个像素的 x 坐标。
  - 底：该矩形的底边下面一个像素的 y 坐标。
- 矩形元组 (3, 1, 9, 6)

*注意：前闭后开的区间。*





## »» 图像的打开和存储

```
>>> from PIL import Image
>>> img = Image.open('horse.jpg')
>>> img.size
      (1674, 1116)
>>> width, height = img.size
>>> img.format
'JPG'
>>> img.save('horse.png')
```

打开图像文件

获取图片的尺寸

获取图片的格式

另存为另一种格式

**本例实现不同格式图像的相互转换。**

## 生成新图像

```
from PIL import Image  
im = Image.new('RGBA', (100, 200), 'purple')
```

*Image.new*可以生成指定大小和背景色的图像。

```
im.save('purpleImage.png')  
im2 = Image.new('RGBA', (20, 20))  
im2.save('transparentImage.png')
```



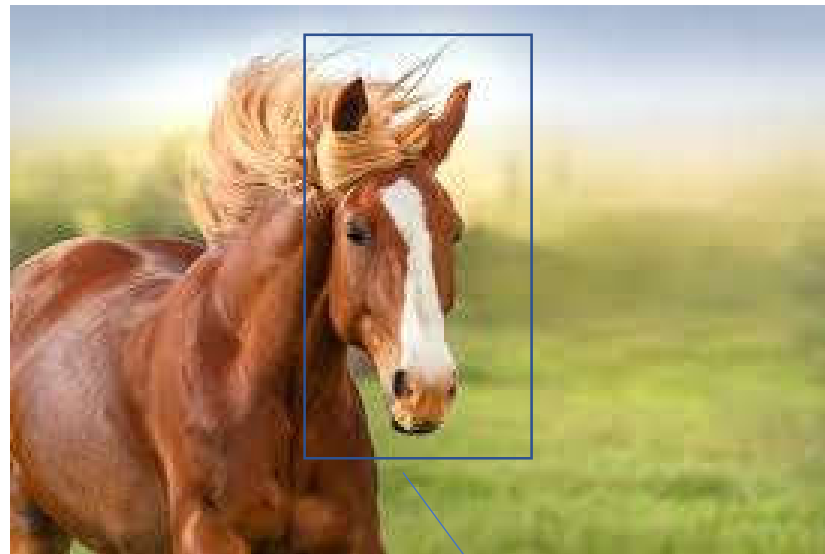
注意，此图像是全透明的，此处额外加了边框。



## ➤➤➤ 裁剪图片

```
croppedIm = img.crop(  
    (600, 200, 1000, 800))  
croppedIm.save('cropped-horse.png')
```

返回一个裁剪后的图像对象，参数是一个Box。



## 复制和粘贴

```
from PIL import Image
img = Image.open('horse.jpg')
croppedIm = img.crop((600, 200, 1000, 800))
horsehead = croppedIm.copy()
img.paste(horsehead, (1200, 200))
img.save('horse2.jpg')
```



*copy()函数返回图像的一个副本，两者可以独立进行修改。  
paste()函数将另一个图像对象放置在本图像的指定位置。*

*尽管名称是 copy()和 paste(), 但 Pillow 中的方法不使用计算机的剪贴板。*

## 调整图像大小及旋转图像

```
from PIL import Image
img = Image.open('horse.jpg')
width, height = img.size
img44 = img.resize((width//4, height//4), Image.NEAREST)
img42 = img.resize((width//4, height//2), Image.NEAREST)
imgr30 = img.rotate(30)
img44.save('horse44.jpg')
img42.save('horse42.jpg')
imgr30.save('horse_r30.jpg')
```

等比例缩放

不等比例, 高度拉长

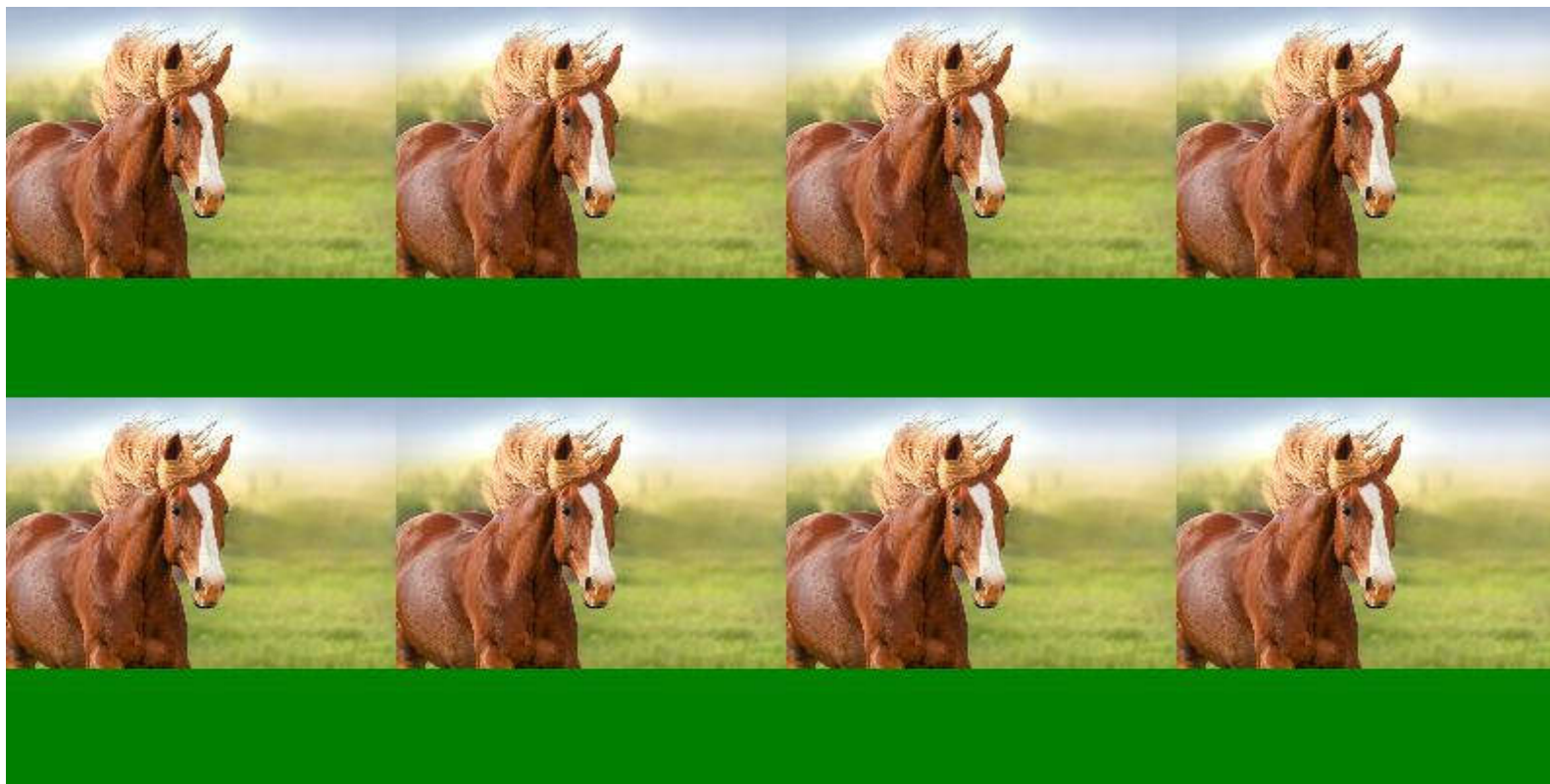
旋转图像



*rotate*有一个`expand`关键字参数, 如果设置为 `True`, 就会放大图像的尺寸以便保证原有像素都在图像中。

*transpose()*函数可以镜像翻转, 此处略。

## 综合练习：八骏图





## »» 参考答案

```
from PIL import Image
img = Image.open('horse.jpg')
width, height = img.size
img_s = img.resize((width//8, height//8), Image.NEAREST)

im = Image.new('RGB', (800, 400), 'green')
for i in range(4):
    for j in range(2):
        im.paste(img_s, (i*200, j*200))

im.save('horse8.jpg')
```

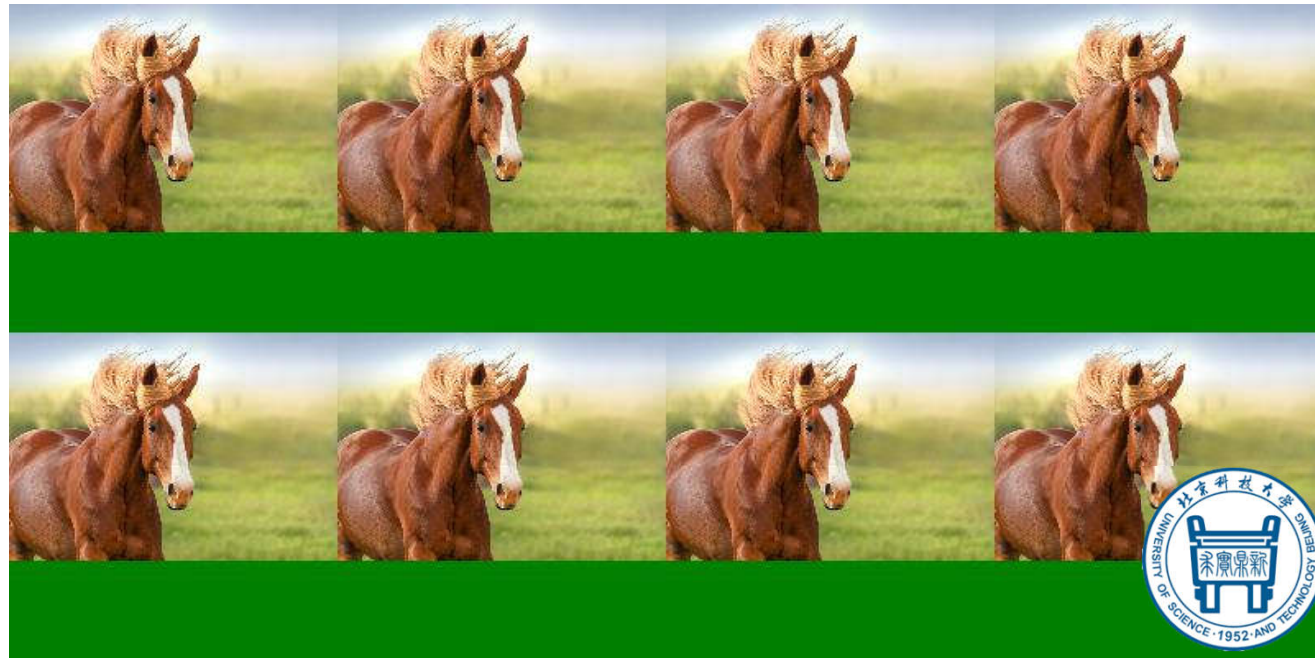
生成新图片

逐个贴入马的图片。



## 添加水印

- 如何在图片的右下角增加水印?





## »» 更改单个像素

- 单个像素的颜色通过 `getpixel()`和`putpixel()`方法取得和设置。它们都接受一个元组，表示像素的 `x` 和 `y`坐标。
- `putpixel()`方法还接受一个元组，作为该像素的颜色。这个颜色参数是四整数 `RGBA` 元组或三整数 `RGB` 元组。
- 注：1. 能改一个像素就能进行任意的图像编辑；2. 图片像素很多时，本方法很慢。



## »»» 更改单个像素

```
img = Image.new('RGB', (200, 100), 'black')
print(img.getpixel((0, 0)))
for x in range(100):
    for y in range(50):
        img.putpixel((x, y), (210, 210, 210))
img.save('putPixel.png')
```







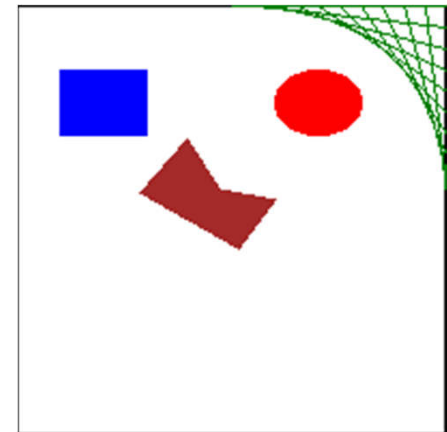
## ➤➤➤ 绘制形状

- ImageDraw 方法在图像上绘制各种形状。
- 点：point(xy, fill)方法绘制单个像素。xy 参数表示要画的点的列表。该列表可以是 x和 y 坐标的元组的列表，例如[(x, y), (x, y), ...]，或是没有元组的 x 和 y 坐标的列表，例如[x1, y1, x2, y2, ...]。
- 线：line(xy, fill, width)方法绘制一条线或一系列的线。xy 是一个元组的列表，例如[(x, y), (x, y), ...]，或整数列表，例如[x1, y1, x2, y2, ...]。每个点都是正在绘制的线上的一个连接点。
- 矩形、椭圆、多边形、文字等。



## 画图示例

```
from PIL import Image, ImageDraw
im = Image.new('RGBA', (200, 200), 'white')
draw = ImageDraw.Draw(im)
draw.line([(0, 0), (199, 0), (199, 199), (0, 199), (0, 0)], fill='black')
draw.rectangle((20, 30, 60, 60), fill='blue')
draw.ellipse((120, 30, 160, 60), fill='red')
draw.polygon(((57, 87), (79, 62), (94, 85), (120, 90), (103, 113)),
fill='brown')
for i in range(100, 200, 10):
    draw.line([(i, 0), (200, i - 100)], fill='green')
im.save('drawing.png')
```





# 键盘鼠标自动化



## 自动化几个层次

- 白盒：完成了解docx/xlsx格式细节，通过python直接生成符合格式的文档。
- 灰色：Word/Excel提供某种控制接口（如COM接口），通过COM接口操作Word/ Excel软件达到预期目标。
- 黑色：缺乏控制接口，仅能通过模拟键盘和鼠标事件驱动GUI软件。

## 自动化几个层次

- 超级黑：无法模拟键盘和鼠标，只能通过外部设备操作真正的鼠标键盘。



左图：模拟人体点击屏幕。



## 自动化终极工具

- 计算机自动化任务的终极工具，就是写程序直接控制键盘和鼠标。
- 这些程序可以控制其他应用，向它们发送虚拟的击键和鼠标点击，就像人类坐在计算机前与软件交互一样。这种技术被称为“图形用户界面自动化”，或“GUI 自动化”。
- 基于 GUI 自动化，就像对一个机械臂编程，让它敲键盘或移动鼠标。



## »» pyautogui

- pyautogui模块可以模拟鼠标移动、按键和滚动鼠标滚轮等。
- 本节仅介绍pyautogui的部分功能。
- <http://pyautogui.readthedocs.org/>
- 安装：
  - pip install PyGetWindow==0.0.1
  - pip install pyautogui

高版本有  
BUG

## »» 注意事项

本课程仅讲述和讨论技术问题。由于使用GUI自动化造成的损失由用户自行负责，本课程不承担任何责任。



- Python 能以想象不到的高速移动鼠标并击键。它可能太快，导致被控程序跟不上。
- 如果出了问题，而程序仍然继续到处移动鼠标，计算机就会失去控制（因为你的键盘和鼠标被人控制了）。
  - 如果程序自己在移动鼠标，停止它可能很难。你可能无法点击“关闭”按钮
- Windows和 Linux 上，注销的热键是 Ctrl-Alt-Del，有可能会失去目前没有保存的文档。





## ➤➤➤ 预防措施

- 设置脚本在每次函数调用后暂停一段时间，让用户有很短的时间窗口来控制鼠标和键盘。
  - 将pyautogui.PAUSE 变量设置为要暂停的秒数。如设置：  
pyautogui.PAUSE = 1.5。
- 开启pyautogui 自动防故障功能。
  - 将鼠标移到屏幕的左上角，将导致pyautogui. FailSafeException 异常。
  - 故障时应尽可能快地向左上移动鼠标，以触发异常。
  - 该异常可以用try语句捕获。

## »» 屏幕坐标系

- 左上角为原点
- x向右为正方向
- y向下为正方向

```
import pyautogui as ag
width, height = ag.size()
print(width, height)
#1366, 768
```

*size()方法返回屏幕尺寸*

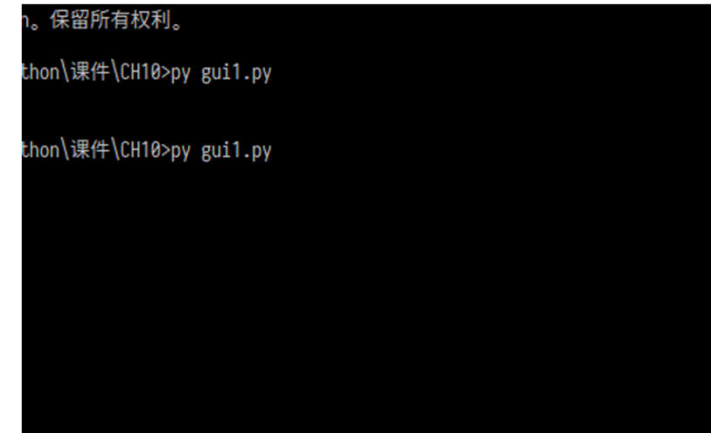




## 移动鼠标

- moveTo() 函数将鼠标移动到屏幕的指定位置，x、y 坐标的整数值分别构成了函数的前两个参数。
- 可选的 duration 整数或浮点数关键字参数，指定了将鼠标移到目的位置所需的秒数。
- 如果不指定，默认值是零，表示瞬间移动。
- moveRel()类似，相对于当前鼠标位置进行移动。

```
import pyautogui
for i in range(10):
    ag.moveTo(100, 100, duration=0.25)
    ag.moveTo(200, 100, duration=0.25)
    ag.moveTo(200, 200, duration=0.25)
    ag.moveTo(100, 200, duration=0.25)
```





## 获取鼠标位置

- position()函数获得鼠标当前的位置。
- 为了持续跟踪获得鼠标位置，可以采用下述代码：

```
print('Press Ctrl-C to quit.')
try:
    while True:
        x, y = pyautogui.position()
        print(f"mouse at {x:4d}, {y:4d}")
except KeyboardInterrupt:
    print('\nDone.')
```

通知用户退出的方式，Ctrl-C  
将引发键盘中断异常。

获取鼠标位置。

在出现键盘中断时触发，退出  
无限循环。



## »» 点击鼠标1/2

- 当前位置点击 click() 方法。
- 若在位置(x,y) 点击，采用click(x,y)方法。
- 默认情况下点击左键。若指定鼠标按键，则需要button关键字参数，取值为 'left'、'middle'或'right'。
- 例：
  - click(100, 150, button='left')
  - click(200, 250, button='right')



## »» 点击鼠标2/2

- 模拟鼠标按下: `mousedown()`
- 释放鼠标按键: `mouseup()`
- 函数参数与 `click()`
  - `click = mousedown + mouseup`
- 鼠标双击: `doubleClick()`



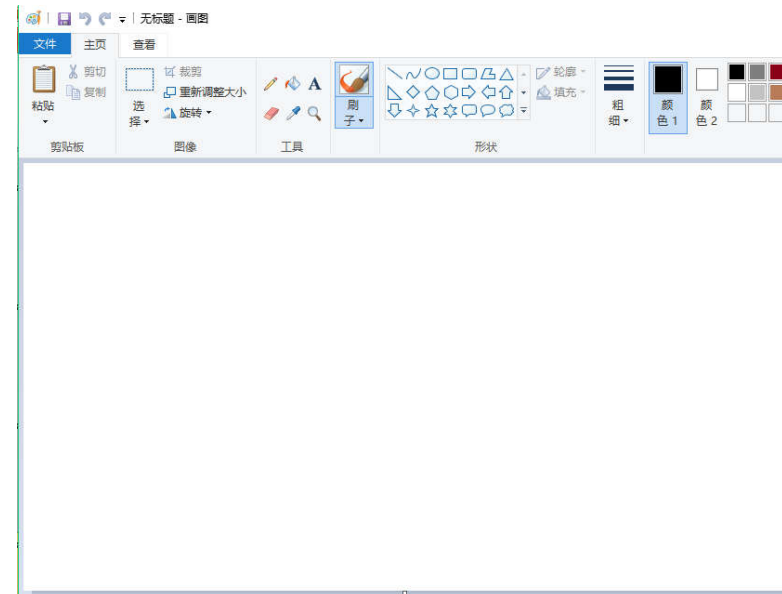
## ➤➤➤ 拖动鼠标

---

- Drag即按住一个按键不放同时移动鼠标。
  - 通过拖动文件图标，在文件夹之间移动文件；
  - 可以拖动绘图。
- dragTo() 和 dragRel() 函数将鼠标拖动到一个新的位置，或相对当前的位置。
  - 函数的参数与moveTo() 和 moveRel() 相似。

## »» 画图软件

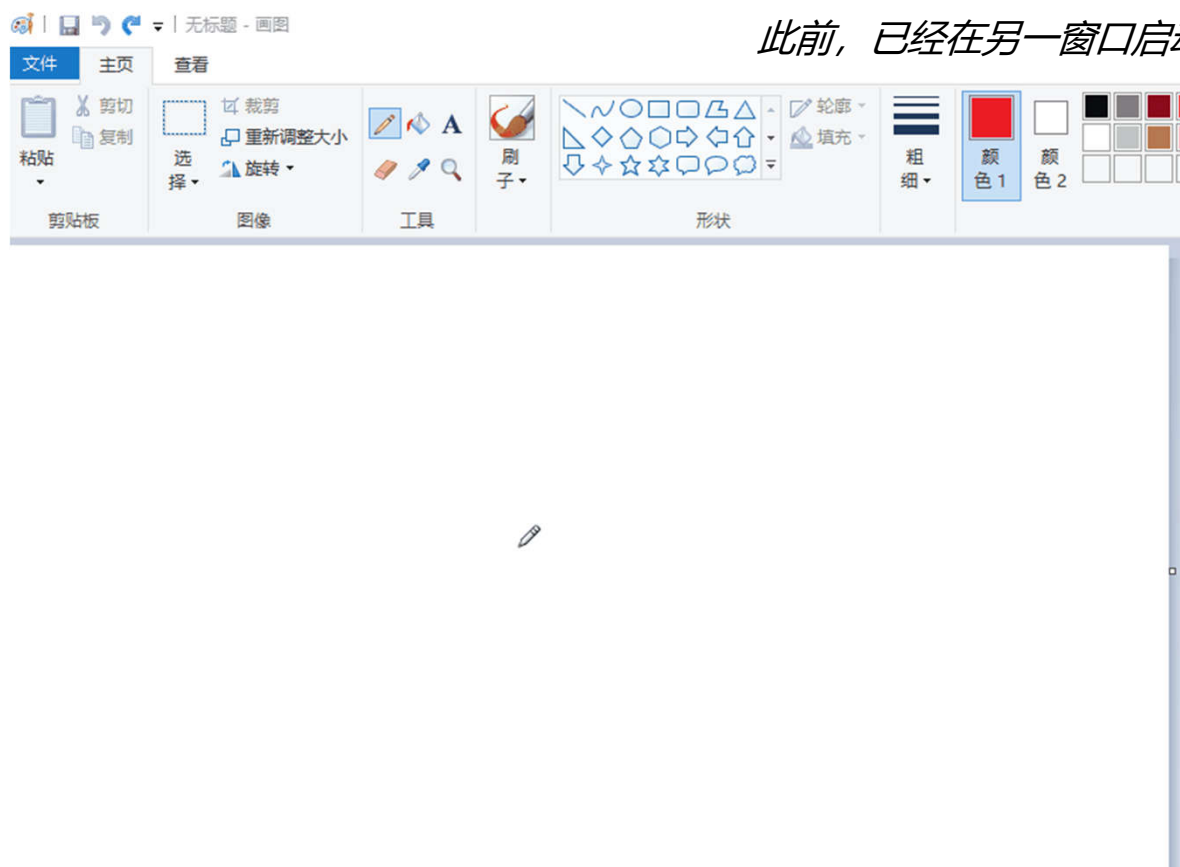
- 绘图应用：
  - Windows 上的 Paint
  - OS X 上的 Paintbrush
  - Linux 上的 GNU Paint
- 选择画笔，拖动鼠标，可以实现绘图。
- 如何模拟鼠标操作来绘图？







## 预期效果



此前, 已经在另一窗口启动了python代码运行。



## »» 示例代码

此处的延时，用于保证在执行本文件之后，有一定的时间可以把鼠标移动到画图软件的画布上方。

```
import pyautogui as ag
import time
time.sleep(5)
ag.click()
distance = 200
while distance > 0:
    ag.dragRel(distance, 0, duration=0.2) # move right
    distance = distance - 5
    ag.dragRel(0, distance, duration=0.2) # move down
    ag.dragRel(-distance, 0, duration=0.2) # move left
    distance = distance - 5
    ag.dragRel(0, -distance, duration=0.2) # move up
```

随后，沿四个方向连续拖曳鼠标。



## 操作步骤

- 打开两个窗口：
  - 画图软件Paint;
  - 命令行或其他可以执行python代码的软件。
- 设置好画图软件的笔刷颜色、粗细等;
- 首先，在命令行中运行上述代码。
- 其次，在5秒内将鼠标移动到画图软件的画布上方，等待。



## 获取屏幕内容

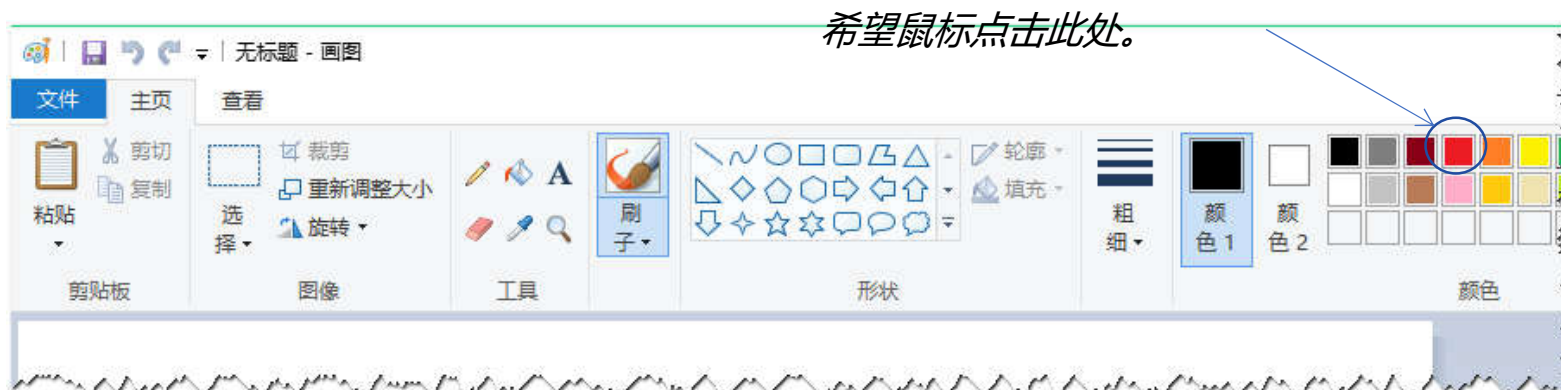
- 做GUI自动测试时，可以利用截图分析按钮是否正常或者变灰（利用Pillow库的getpixel方法）。
- 可以做一个软件，随着鼠标移动，自动取得不同位置的颜色。

```
import pyautogui  
im = pyautogui.screenshot()  
im.save("capture.png")
```

返回屏幕截图，是一个Pillow库中的图像对象。



## 屏幕图像识别



- 提前准备一幅（截图后取出）的局部图像：
- `locateOnScreen('want.png')`
- 找到屏幕上完全匹配的图像所在的box，然后可在box的中心（或其他位置）进行点击。



## 键盘

- `typewrite()`方法模拟发送按键
  - `typewrite('Hello world!')`
  - `typewrite(['a', 'b', 'left', 'left', 'X', 'Y'])`
  - 前者均为正常按键
  - 后者的'left'表示 “左方向键”



## 键盘：特殊键

键盘键字符串	含义
'a', 'b', 'c', 'A', 'B', 'C', '1', '2', '3', '!', '@', '#', 等等	单个字符的键
'enter' (or 'return' or '\n')	回车键
'esc'	Esc 键
'shiftright', 'shiftright'	左右 Shift 键
'altleft', 'altright'	左右 Alt 键
'ctrlleft', 'ctrlright'	左右 Ctrl 键
'tab' (or '\t')	Tab 键
'backspace', 'delete'	Backspace 和 Delete 键
'pageup', 'pagedown'	Page Up 和 Page Down 键
'home', 'end'	Home 和 End 键
'up', 'down', 'left', 'right'	上下左右箭头键
'f1', 'f2', 'f3', 等等	F1 至 F12 键



## 键盘模拟

- `keyDown()`和`keyUp()`将向计算发送虚拟的按键和释放。
- `press()` 函数模拟完整的击键。

```
keyDown('shift');  
press('1');  
keyUp('shift')
```



相当于Shift-1, 即“!”键

- `hotkey()` 按参数自左向右依次模拟按键, 倒序模拟释放。
  - `hotkey('ctrl', 'alt', 'shift', 's')`





## 弹钢琴

- FreePiano是一款免费的钢琴软件
- 通过控制鼠标可以弹琴
- 可以作曲了~
- 万能公式：Python+自动控制+相应软件



## 弹钢琴的演示

代码略，此处是精确控制鼠标点击位置和时间。





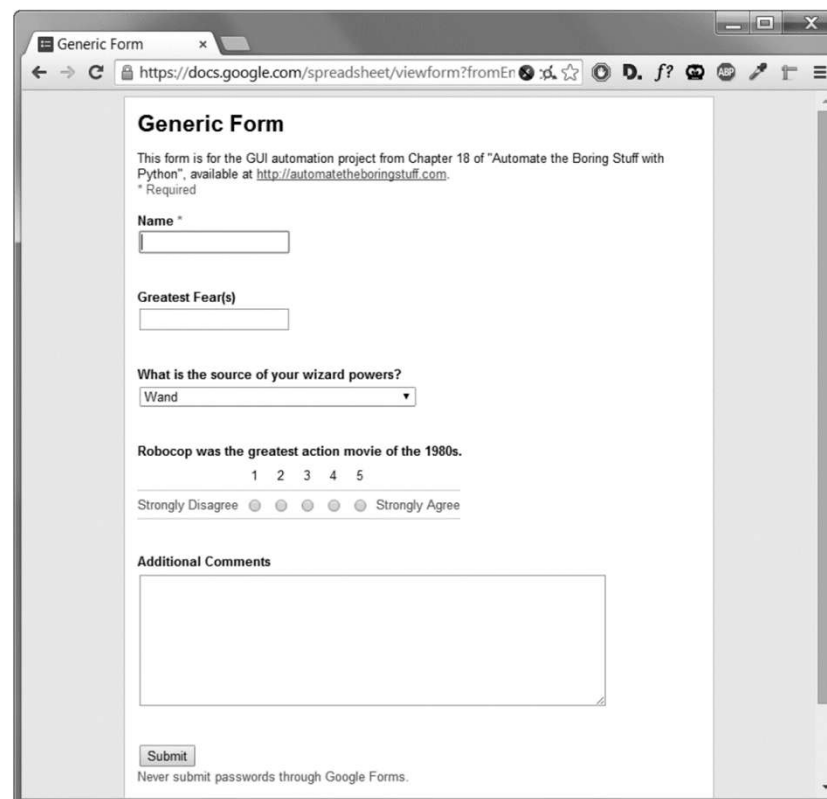
## 》》》 其他自动化软件

---

- pyautogui是一款入门级的库，相对简单且跨平台
- 类似的库还有
  - pywinauto
  - autoit(pyautoit)
  - autohotkey(pyahk), autohotkey是Windows操作系统神器之一

## ➤➤➤ 可能完成的任务

- 模拟键盘及鼠标，实现自动填表；
- 填写Excel表格；
- 编写Word文件。



The screenshot shows a web browser window displaying a Google Form titled "Generic Form". The form is part of a GUI automation project from Chapter 18 of "Automate the Boring Stuff with Python". It contains the following fields and elements:

- Name \***: A text input field.
- Greatest Fear(s)**: A text input field.
- What is the source of your wizard powers?**: A dropdown menu with "Wand" selected.
- Robocop was the greatest action movie of the 1980s.**: A rating scale from 1 to 5, with radio buttons for "Strongly Disagree" and "Strongly Agree".
- Additional Comments**: A large text area for additional input.
- Submit**: A button at the bottom of the form.
- Footer**: A note stating "Never submit passwords through Google Forms."



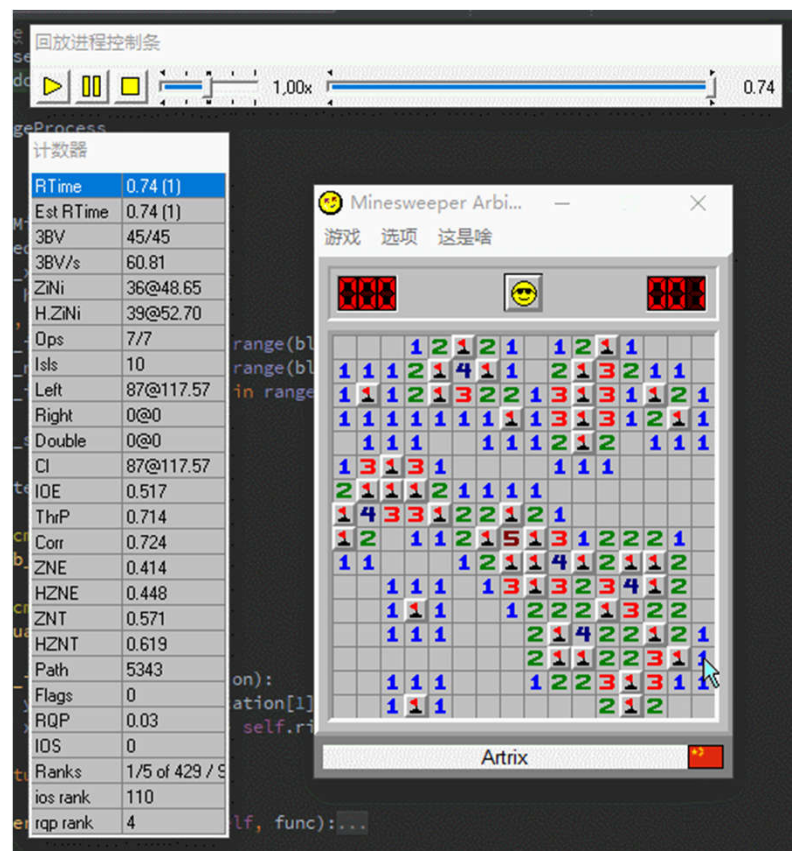
## 可能完成的任务

如何识别数字?

- 自动玩游戏;
- 右图使用了多个库实现了自动扫雷。
- 目前我们有能力完成!

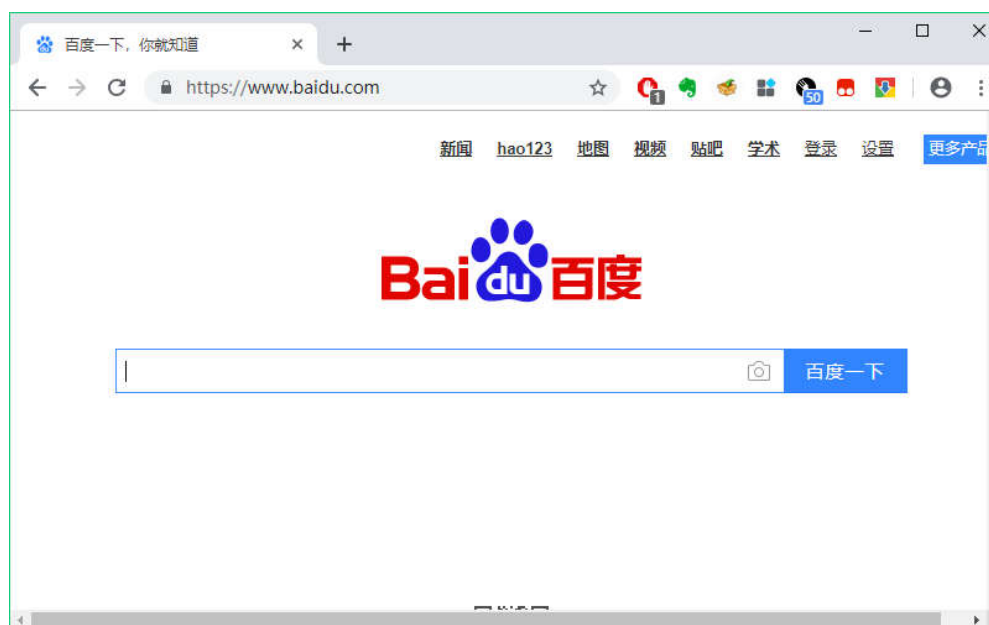
原图来自:

<https://zhuanlan.zhihu.com/p/35755039>



## 》》 可能完成的任务

- 网络爬虫：
- 模拟键盘输入关键字，发送“回车”键，获得页面；
- 利用键盘另存当前页面；
- 点击“下一页”，继续另存当前页面。



## 》》 可能完成的任务

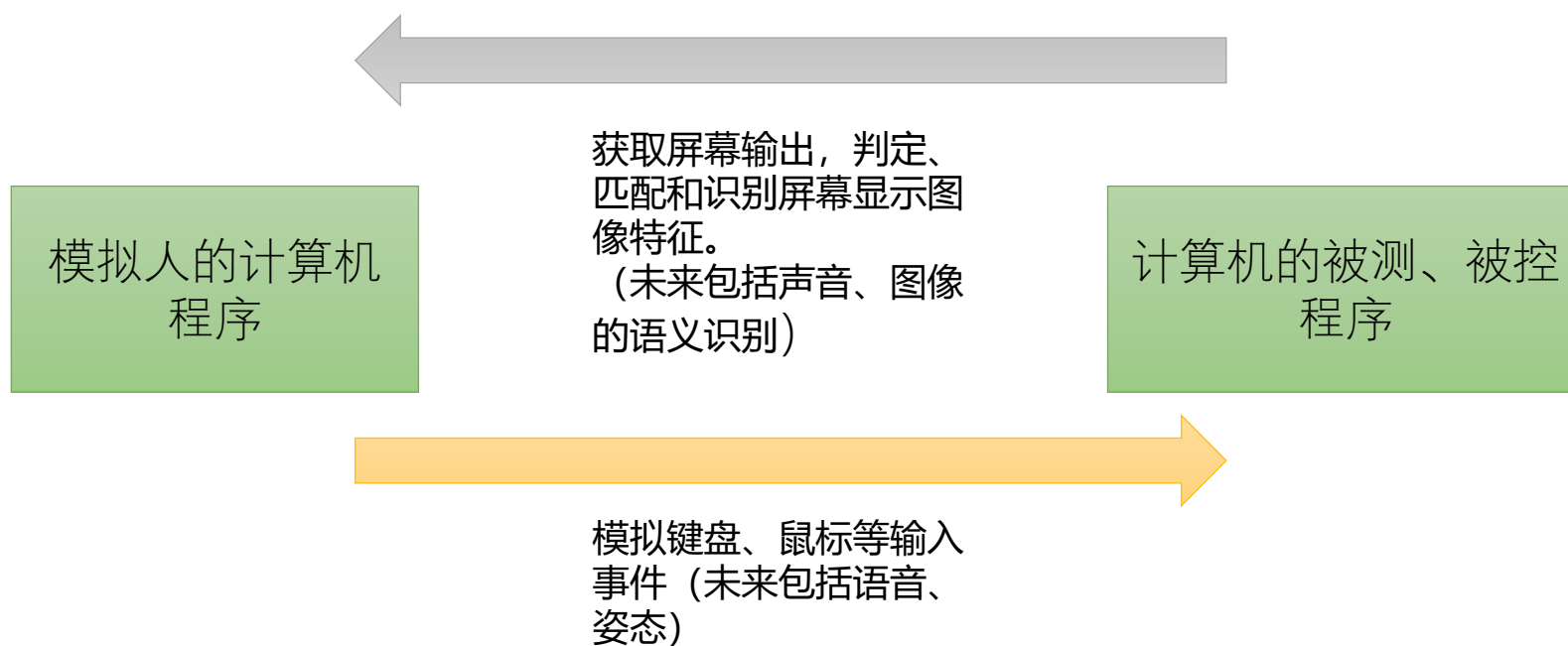
- 聊天机器人
- 模拟键盘输入微信、QQ等
- 在AI时代，视觉、听觉和模拟发声、模拟形象都可以实现。

在CCTV-13的特别报道《直播长江》中，虚拟主持人“康辉”与记者在现场进行对话互动；新华社宣布与搜索引擎搜狗合作创建由世界上第一个人工智能（AI）运营的虚拟女播音员。





## GUI自动化模型





## »» 小结

- 充分挖掘办公自动化的能力，能够有效提升工作效率。

