



第1章 概述





广东工程职业技术学院
GUANGDONG ENGINEERING POLYTECHNIC



Python

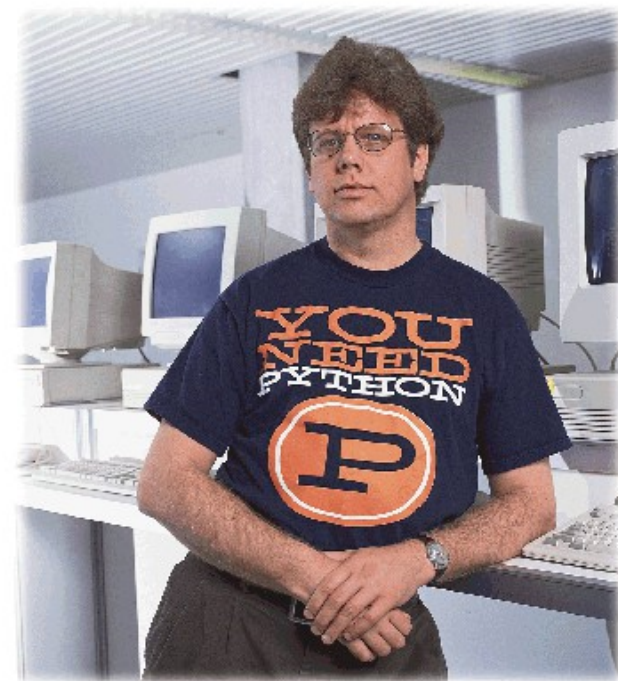


工于建构 成于创造



Guido van Rossum（吉多·范罗苏姆）是 Python 编程语言的创始人，1989 年创立，荷兰人。现居美国，曾就职于 Google 公司。之所以选中 Python 作为程序的名字，是因为他迷上了英国电视喜剧片《Monty Python and the Flying Circus》。

2018 年 7 月，Guido van Rossum 宣布退出决策层，不再领导该语言，后续会作为一个普通的核心开发者。



十大应用领域

WEB 开发、桌面软件、网络编程、爬虫开发、云计算开发、人工智能、自动化运维、金融分析、科学运算、游戏开发



就业方向

Python Web 网站工程师、Python 自动化测试、数据分析、人工智能

☐ 全选	申请职位	收藏职位	智能排序	发布时间	1 / 718	共35864条职位
职位名	公司名	工作地点	薪资	发布时间		
☐ Python开发工程师	北京易诚互动网络技术有限公司	北京-顺义区	1-2万/月	02-24		
☐ Python开发工程师	广州光汇软件科技有限公司	广州	0.8-1.5万/月	02-24		
☐ Python运维工程师	深圳市沃享科技有限公司	深圳-南山区	1-2万/月	02-24		
☐ python开发工程师(业务方向)	广州回头车信息科技有限公司	广州-天河区	0.8-1.5万/月	02-24		
☐ Python全栈工程师实训生 可培训	北游国际科技（北京）有限公司	郑州-高新区	6-8千/月	02-24		
☐ JAVA/ Python讲师(上市外企+双休+...	童程童美济南分公司	济南-市中区	6-8千/月	02-24		
☐ 教学老师(python和scratch)	杭州智一信息技术有限公司	杭州-西湖区	1-1.5万/月	02-24		
☐ 资深Python开发工程师	浙江甬天科技股份有限公司	杭州	1.5-2.5万/月	02-24		
☐ Python开发工程师	深圳市易佰网络科技有限公司	深圳-龙岗区	0.8-1.4万/月	02-24		
☐ Python编程老师	南京蓝骑士文化发展有限公司	南京-建邺区	0.8-1万/月	02-24		
☐ Python数据分析/爬虫 工程师	上海魔曦广告传媒有限公司	上海-徐汇区	2-2.5万/月	02-24		
☐ python/go软件工程师	上海委可得企业管理咨询有限公司	杭州-西湖区	2-4.5万/月	02-24		

课程学习目标

学习 Python 的编程模式，熟练运用 Python 内置函数与运算符、列表、元组、字典、集合等基本数据类型以及相关列表推导式、切片等语法来解决实际问题，熟练掌握 Python 分支结构、循环结构、函数使用，熟练使用正则表达式处理字符串，掌握数据分析、数据爬虫、数据可视化的步骤和方法。

1.1 Python 语言简介

- Python 是一门跨平台、开源、免费的解释型高级动态编程语言。
- Python 支持命令式编程（ How to do ）、函数式编程（ What to do ），完全支持面向对象程序设计，拥有大量扩展库。
- 胶水语言：可以把多种不同语言编写的程序融合到一起实现无缝拼接，更好地发挥不同语言和工具的优势，满足不同应用领域的需求。

1.1 Python 语言简介

- **问题解决：**把列表中的所有数字都加 5，得到新列表。（命令式编程）

创建列表

```
>>> x = list(空列表)
```

```
>>> x
```

循环，遍历 x 中的每个元素

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
>>> y = []
```

列表方法，在尾部追加元素

```
>>> for num in x:
```

```
    y.append(num+5)
```

列表推导式

```
>>> y
```

```
[5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
```

1.1 Python 语言简介

- **问题解决**：把列表中的所有数字都加 5，得到新列表。（函数式编程）

```
>>> x = list(range(10))
```

```
>>> x
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

定义函数，接收
个数字，加 5 后返
回

```
>>> def add5(num):
```

把函数 add5 映射到 x 中的每个元
素

```
    return num+5
```

```
>>> list(map(add5, x))
```

lambda 表达式，等价于函数
add5

```
[5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
```

```
>>> list(map(lambda num: num+5, x))
```

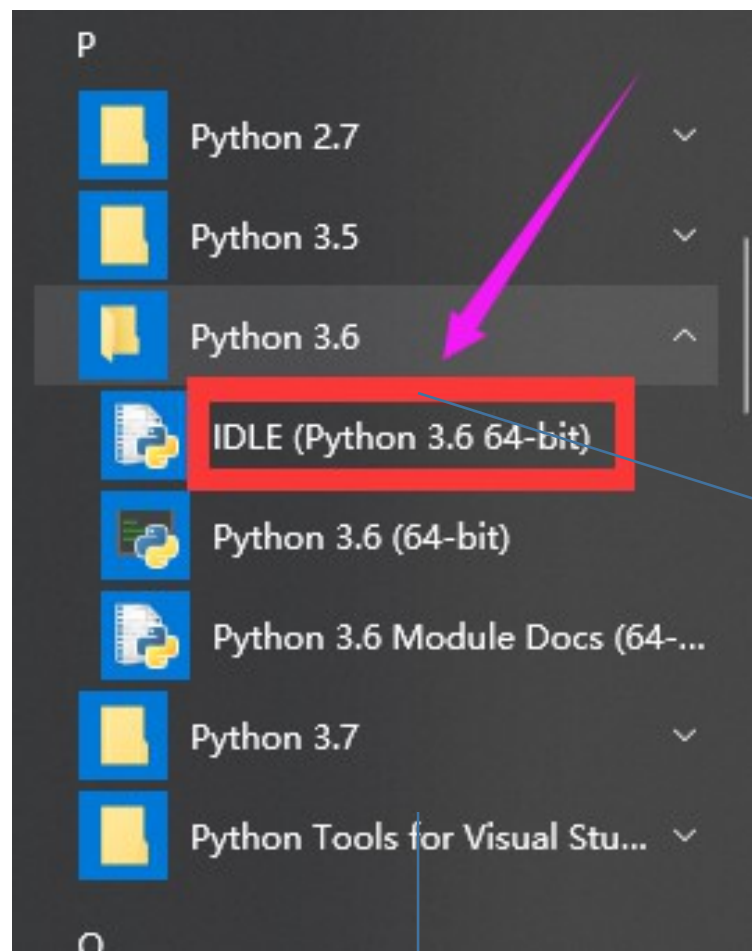
1.2 Python 版本简介

- Python 目前存在 2.x 和 3.x 两个系列的版本，互相之间不兼容。
- Python 2.x 系列最迟将于 2020 年全面放弃维护和更新。

1.3 Python 开发环境安装与配置

- 默认编程环境：IDLE：<https://www.python.org/>
- 其他常用开发环境：
 - Eclipse+PyDev
 - pyCharm：<http://www.jetbrains.com/pycharm/>
 - wingIDE
 - Eric
 - PythonWin
 - Anaconda（内含 Jupyter 和 Spyder）：<https://www.anaconda.com/download>
 - zwPython

1.3.1 IDLE

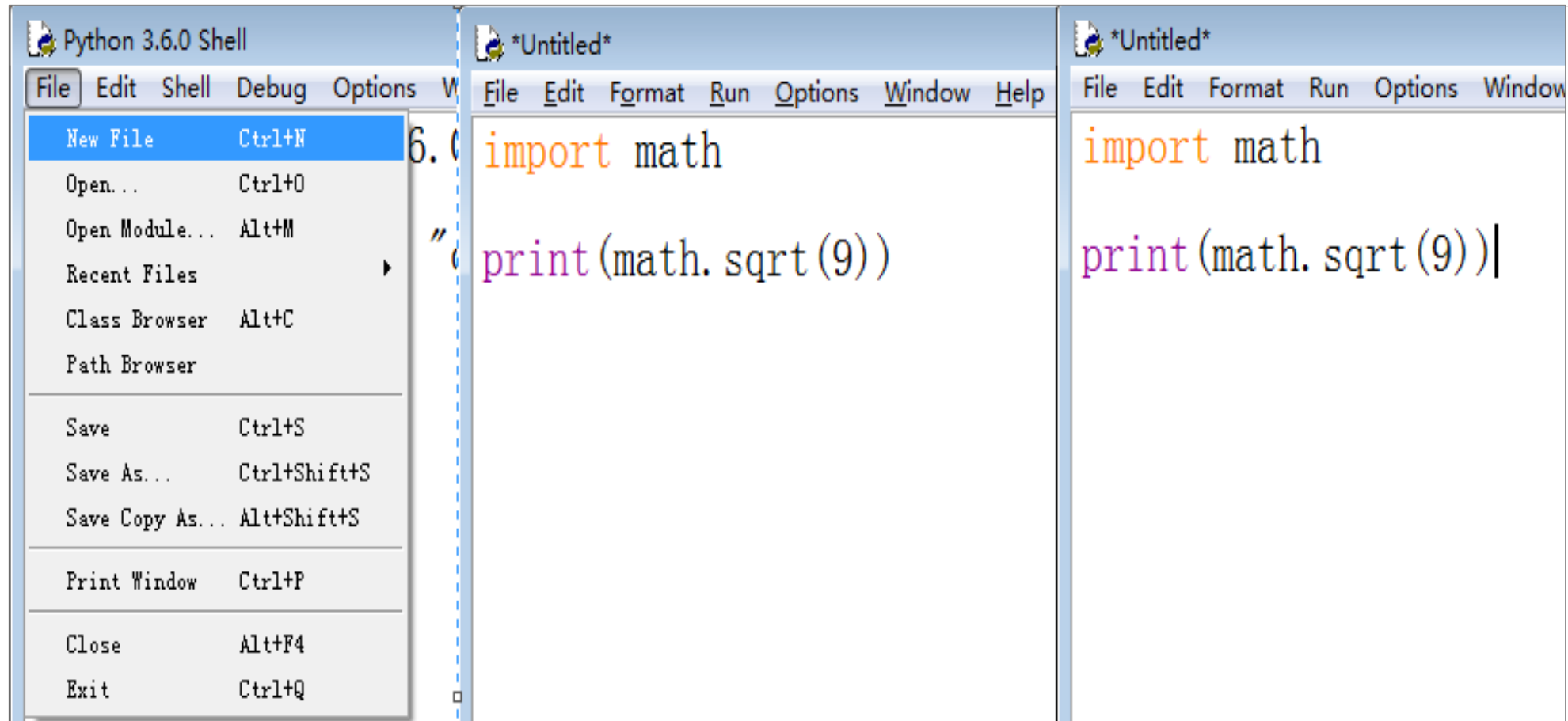


```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more
information.
>>> print("Hello World!")
Hello World!
>>> import math
>>> print(math.sqrt(9))
3.0
>>>
```

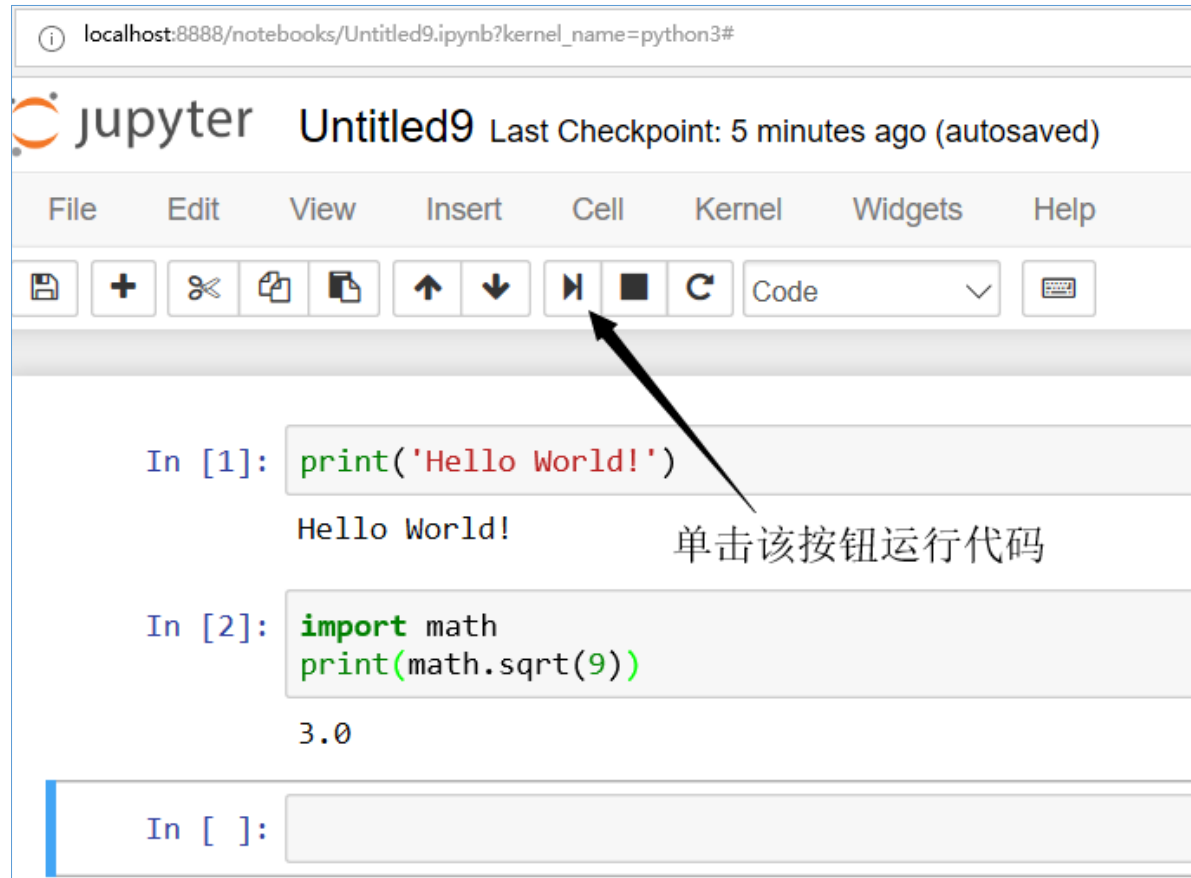
命令提示符

每次只能执行一条语句

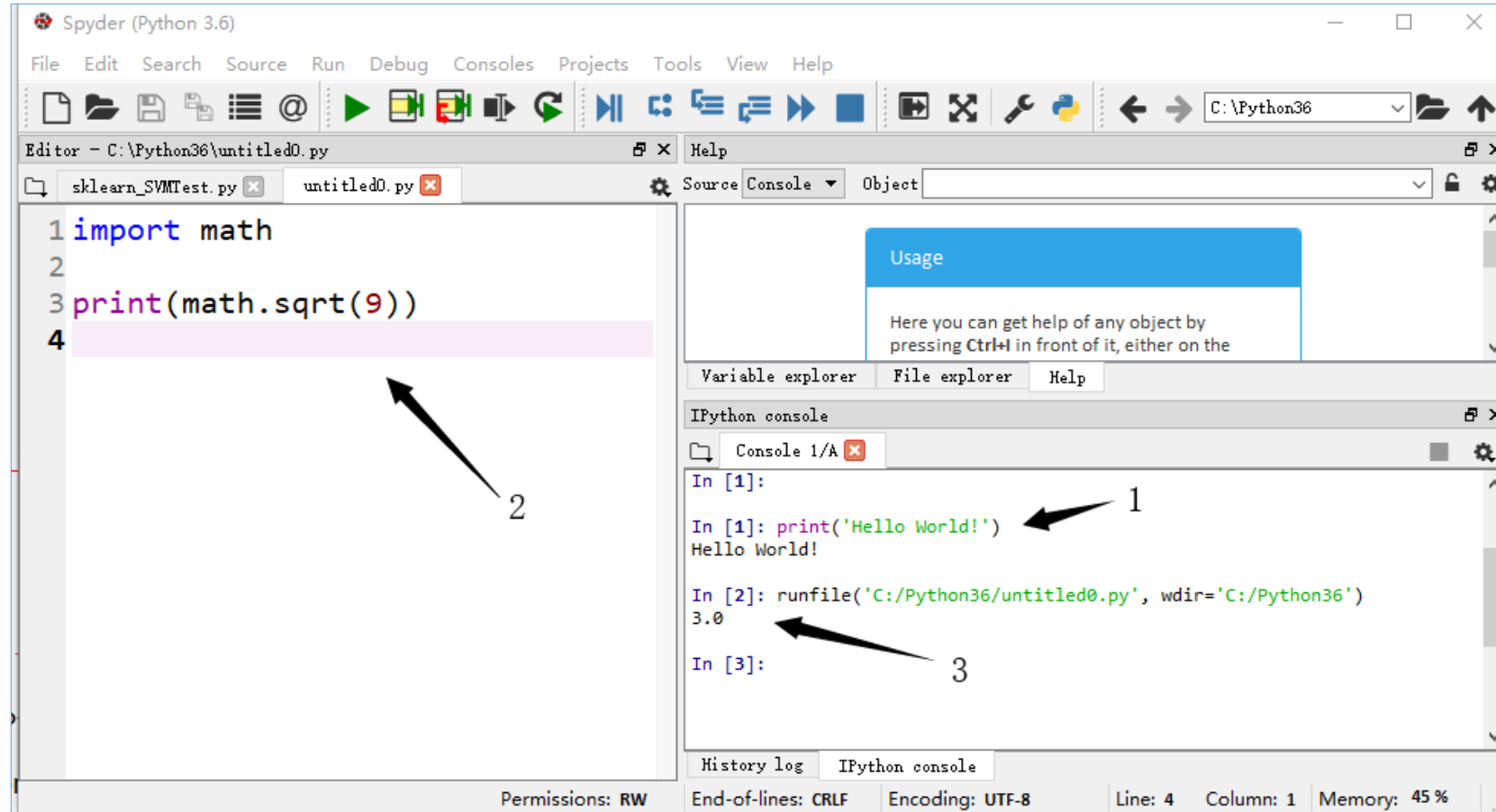
1.3.1 IDLE



1.3.2 Anaconda



1.3.2 Anaconda



1.4 Python 编程规范与代码优化建议

(1) 缩进

- ✓ 类定义、函数定义、选择结构、循环结构、 with 块，行尾的冒号表示缩进的开始。
- ✓ python 程序是依靠代码块的缩进来体现代码之间的逻辑关系的，缩进结束就表示一个代码块结束了。
- ✓ 同一个级别的代码块的缩进量必须相同。
- ✓ 一般而言，以 4 个空格为基本缩进单位。

```
with open(fn) as fp:
    for line in csv.reader(fp):
        if line:
            print(*line)
```

1.4 Python 编程规范与代码优化建议

(2) 每个 import 语句只导入一个模块，最好按标准库、扩展库、自定义库的顺序依次导入。

```
import csv
import random
import datetime
import pandas as pd
import matplotlib.pyplot as plt
```

1.4 Python 编程规范与代码优化建议

(3) 最好在每个类、函数定义和一段完整的功能代码之后增加一个空行，
在运算符两侧各增加一个空格，逗号后面增加一个空格。

```
sockDianming.close()
sockDianming = None

# 开始点名
def buttonStartDianmingClick():
    # 如果还没有注册，拒绝运行
    if int_zhuce.get() == 0:
        tkinter.messagebox.showerror('很抱歉', '请联系作者进行软件注册!')
        return
    if int_zuoye.get() == 1:
        tkinter.messagebox.showerror('很抱歉', '现在正在收作业')
        return
    if int_canDianming.get() == 1:
        tkinter.messagebox.showerror('很抱歉', '现在正在点名')
        return
    tkinter.messagebox.showinfo('恭喜', '设置成功，现在开始点名')
    #开始点名
    int_canDianming.set(1)
    global tDianming_id
    t = threading.Thread(target=thread_Dianming)
    t.start()
    tDianming_id = t.ident
    buttonStartDianming = tkinter.Button(root, text='开始点名', command=buttonStartD
    buttonStartDianming.place(x=20, y=60, height=30, width=100)

def buttonStopDianmingClick():
    # 如果还没有注册，拒绝运行
    if int_zhuce.get() == 0:
        tkinter.messagebox.showerror('很抱歉', '请联系作者进行软件注册!')
```

空行

有空格

没有空格

1.4 Python 编程规范与代码优化建议

(4) 尽量不要写过长的语句。如果语句过长，可以考虑拆分成多个短一些的语句，以保证代码具有较好的可读性。如果语句确实太长而超过屏幕宽度，最好使用续行符（ line continuation character ） “ \ ”，或者使用圆括号将多行代码括起来表示是一条语句。

```
x = 1 + 2 + 3\  
    + 4 + 5\  
    + 6  
  
y = (1 + 2 + 3  
     + 4 + 5  
     + 6)
```

1.4 Python 编程规范与代码优化建议

(5) 虽然 Python 运算符有明确的优先级，但对于复杂的表达式建议在适当的位置使用括号使得各种运算的隶属关系和顺序更加明确、清晰。

1.4 Python 编程规范与代码优化建议

(6) 注释

- ✓ 以符号 # 开始，表示本行 # 之后的内容为注释。
- ✓ 包含在一对三引号 `"""..."""` 或 `"""..."""` 之间且不属于任何语句的内容将被解释器认为是注释。

```
# 读取并显示上面代码生成的csv文件内容
with open(fn) as fp:
    for line in csv.reader(fp):
        if line:
            print(*line)
```

1.5 扩展库安装方法

- pip 在线安装（命令提示符环境，建议切换至 Python 安装目录中的 scripts 文件夹执行）
- pip 离线安装：<https://www.lfd.uci.edu/~gohlke/pythonlibs/>
- exe 安装，不是每个扩展库都支持
- conda 在线安装
- 如果机器上安装了多个 Python 开发环境，那么在一个环境下安装的扩展库无法在另一个环境下使用，需要分别安装。

1.5 扩展库安装方法

pip 命令示例	说明
<code>pip download SomePackage[==version]</code>	下载扩展库的指定版本，不安装
<code>pip freeze [> requirements.txt]</code>	以 requirements 的格式列出已安装模块
<code>pip list</code>	列出当前已安装的所有模块
<code>pip install SomePackage[==version]</code>	在线安装 SomePackage 模块的指定版本
<code>pip install SomePackage.whl</code>	通过 whl 文件离线安装扩展库
<code>pip install package1 package2 ...</code>	依次（在线）安装 package1、package2 等扩展模块
<code>pip install -r requirements.txt</code>	安装 requirements.txt 文件中指定的扩展库
<code>pip install --upgrade SomePackage</code>	升级 SomePackage 模块
<code>pip uninstall SomePackage[==version]</code>	卸载 SomePackage 模块的指定版本

把 SomePackage 替换
为实际要安装或卸载
的扩展库名

<http://www.lfd.uci.edu/~gohlke/pythonlibs/>
下载时选择合适版本，并且不要修改文件名

1.6 标准库与扩展库对象的导入与使用

- Python 默认安装仅包含基本或核心模块，启动时也仅加载了基本模块，在需要时再显式地导入和加载标准库和第三方扩展库（需正确安装），这样可以减小程序运行的压力，并且具有很强的可扩展性。
- 从“木桶原理”的角度来看，这样的设计与安全配置时遵循的“最小权限”原则的思想是一致的，也有助于提高系统安全性。

1.6.1 import 模块名 [as 别名]

```
>>> import math
```

```
# 导入标准库 math
```

```
>>> math.sin(0.5)  
正弦
```

```
# 求 0.5 ( 单位是弧度 ) 的
```

```
0.479425538604203
```

```
>>> import random
```

```
# 导入标准库 random
```

```
>>> n = random.random()
```

```
# 获得 [0,1) 内的随机小数
```

```
>>> n = random.randint(1,100)  
机整数
```

```
# 获得 [1,100] 区间上的随
```

```
>>> n = random.randrange(1, 100)  
机整数
```

```
# 返回 [1, 100) 区间中的随
```

1.6.2 from 模块名 import 对象名 [as 别名]

```
>>> from math import sin          # 只导入模块中的指定对象，访问速度略快
>>> sin(3)
0.1411200080598672
>>> from math import sin as f     # 给导入的对象起个别名
>>> f(3)
0.1411200080598672
>>> from os.path import isfile
>>> isfile(r'C:\windows\notepad.exe')
True
```

1.6.3 from 模块名 import *

```
>>> from math import *           # 导入标准库 math 中所有对象
```

```
>>> sin(3)                       # 求正弦值
```

```
0.1411200080598672
```

```
>>> gcd(36, 18)                 # 最大公约数
```

```
18
```

```
>>> pi                          # 常数  $\pi$ 
```

```
3.141592653589793
```

```
>>> e                          # 常数 e
```

```
2.718281828459045
```

```
>>> log2(8)                     # 计算以 2 为底的对数值
```

```
3.0
```

```
>>> log10(100)                 # 计算以 10 为底的对数值
```