

Python程序语言入门与应用



python

Life is short, use Python
人生苦短，我用Python



新乡医学院

Python程序语言入门与应用

第十一章 Python 图形化界面开发基础

王海蛟

新乡医学院





下周课程&课后作业



第9章 科学计算和可视化

课后练习

修改实例24，按照省份输出大学排名结果。

代码文件命名：10-1pachongshengfen

修改实例24，编写软科世界一流学科排名2019-化学排名结果。

代码文件命名：10-2pachongchem

.py代码文件打包(10.学号
+姓名)发送到
python_xxmu@163.com



下周课程&课后作业

代码文件命名: 10-1pachongshengfen



修改实例24，按照省份输出大学排名结果。

24-按照省市排名.py - E:\课程\python\第11次课\上次习题\24-按照省市排名.py (3.7.3)

File Edit Format Run Options Window Help

```
#e24CrawUnivRanking.py
import requests
from bs4 import BeautifulSoup
allUniv = []
def getHTMLText(url):
    try:
        r = requests.get(url, timeout=30)
        r.raise_for_status()
        r.encoding = 'utf-8'
        return r.text
    except:
        return ""
def fillUnivList(soup):
    data = soup.find_all('tr')
    for tr in data:
        ltd = tr.find_all('td')
        if len(ltd)==0:
            continue
        singleUniv = []
        for td in ltd:
            singleUniv.append(td.string)
        allUniv.append(singleUniv)
def printUnivList(shen):
    print("{:~4}{:~10}{:~5}{:~8}".format("排名","学校名称","省市","总分"))
    for i in range(len(allUniv)):
        u=allUniv[i]
        if u[2]==shen:
            print("{:~4}{:~10}{:~5}{:~8}".format(u[0],u[1],u[2],u[3]))
def main():
    url = 'http://www.zuihaodaxue.com/zuihaodaxuepaiming2019.html'
    sheng=input('请输入需要排名的省市')
    html = getHTMLText(url)
    soup = BeautifulSoup(html, "html.parser")
    fillUnivList(soup)
    printUnivList(sheng)
main()
```

```
=====
请输入需要排名的省市河南
排名 学校名称 省市 总分
126 郑州大学 河南 33.8
143 河南大学 河南 32.5
165 河南师范大学 河南 30.8
216 河南理工大学 河南 28.6
226 河南工业大学 河南 28.2
234 郑州师范学院 河南 28.0
246 河南财经政法大学 河南 27.6
260 河南农业大学 河南 27.1
279 河南科技大学 河南 26.4
313 郑州轻工业大学 河南 25.4
324 华北水利水电大学 河南 25.2
343 信阳师范学院 河南 24.8
357 南阳师范学院 河南 24.2
379 洛阳理工学院 河南 23.7
384 河南科技学院 河南 23.5
384 洛阳师范学院 河南 23.5
430 安阳师范学院 河南 22.6
434 河南工程学院 河南 22.5
443 郑州航空工业管理学院 河南 22.1
463 新乡学院 河南 21.6
471 河南牧业经济学院 河南 21.4
473 河南城建学院 河南 21.3
478 安阳工学院 河南 21.2
494 中原工学院 河南 20.6
497 周口师范学院 河南 20.4
507 南阳理工学院 河南 19.8
512 黄淮学院 河南 19.5
514 许昌学院 河南 19.4
520 商丘师范学院 河南 19.2
548 平顶山学院 河南 16.0
>>>
```



下周课程&课后作业

代码文件命名: 10-1pachongshengfen



修改实例24，按照省份输出大学排名结果。

24-按照省市排名.py - E:\课程\python\第11次课\上次习题\24-按照省市排名.py (3.7.3)

File Edit Format Run Options Window Help

```
#e24CrawUnivRanking.py
import requests
from bs4 import BeautifulSoup
allUniv = []
def getHTMLText(url):
    try:
        r = requests.get(url, timeout=30)
        r.raise_for_status()
        r.encoding = 'utf-8'
        return r.text
    except:
        return ""
def fillUnivList(soup, shen):
    data = soup.find_all('tr')
    for tr in data:
        ltd = tr.find_all('td')
        if len(ltd)==0:
            continue
        singleUniv = []
        for td in ltd:
            singleUniv.append(td.string)
        if singleUniv[2]==shen:
            allUniv.append(singleUniv)
def printUnivList(num=0):
    print("{:4}{:10}{:5}{:8}".format("排名", "学校名称", "省市", "总分"))
    if num==0:
        num=len(allUniv)
    for i in range(num):
        u=allUniv[i]
        print("{:4}{:10}{:5}{:8}".format(u[0],u[1],u[2],u[3]))
def main():
    url = 'http://www.zuihaodaxue.com/zuihaodaxuepaiming2019.html'
    sheng=input('请输入需要排名的省市: ')
    num=input('请输入需要获取该省排名大学的个数(0代表全部获取): ')
    print('\n\n\n')
    html = getHTMLText(url)
    soup = BeautifulSoup(html, "html.parser")
    fillUnivList(soup, sheng)
    printUnivList(eval(num))
main()
```

=====

请输入需要排名的省市: 河南

请输入需要获取该省排名大学的个数(0代表全部获取): 2

排名	学校名称	省市	总分
126	郑州大学	河南	33.8
143	河南大学	河南	32.5

>>>

#总列表添加相关查询省市的大学信息



下周课程&课后作业

代码文件命名: 10-2pachongchem



修改实例24, 编写软科世界一流学科排名2019-化学排名结果。

http://www.zuihaodaxue.com/BCSR/huaxue2019.html

国足输球, 里皮辞职!

最好大学网
ZUIHAODAXUE.COM

English | 订阅我们

网站首页

中国大学排名

世界大学排名

原创分析

要闻资讯

院校信息

会议

联系我们



首页 / 中国大学排名 / 软科中国最好学科排名 / 软科中国最好学科排名2019-化学

软科中国最好学科排名2019-化学

2019

大学排名

门类: 理学

一级学科: 化学

排名方法

FAQ

2019 排名	2018 排名	百分位段	院校名称*	博士点	重点学科	总分
1	2	前1%	中国科学技术大学	博 ¹	重 ¹	1463
2	1	前1%	北京大学	博 ¹	重 ¹	1310
3	3	前2%	复旦大学	博 ¹	重 ¹	1126
4	7	前2%	南开大学	博 ¹	重 ¹	979
5	4	前3%	清华大学	博 ¹	重 ¹	973
6	5	前3%	厦门大学	博 ¹	重 ¹	926

软科世界一流学科排名

软科全球院系排名

软科中国最好大学排名

软科世界大学学术排名

软科世界大学学科领域排名

软科中国两岸四地大学排名

软科中国最好学科排名

各国大学排名

澳大利亚

奥地利

比利时

巴西

加拿大

中国

丹麦

芬兰

法国

德国



下周课程&课后作业

代码文件命名: 10-2pachongchem

修改实例24, 编写软科世界一流学科排名2019-化学排名结果。

http://www.zuihaodaxue.com/BCSR/wulixue2019.html

国足输球, 里皮辞职!

最好大学网
ZUIHAODAXUE.COM

English | 订阅我们

网站首页

中国大学排名

世界大学排名

原创分析

要闻资讯

院校信息

会议

联系我们



首页 / 中国大学排名 / 软科中国最好学科排名 / 软科中国最好学科排名2019-物理学

软科中国最好学科排名2019-物理学

2019

大学排名

门类: 理学

一级学科: 物理学

排名方法

FAQ

2019 排名	2018 排名	百分位段	院校名称*	博士点	重点学科	总分
1	1	前1%	中国科学技术大学	博 ¹	重 ¹	1408
2	2	前2%	清华大学	博 ¹	重 ¹	1198
3	3	前2%	北京大学	博 ¹	重 ¹	1014
4	4	前3%	南京大学	博 ¹	重 ¹	735
5	5	前3%	上海交通大学	博 ¹	重 ¹	626
6	6	前4%	复旦大学	博 ¹	重 ¹	558
7	7	前4%	吉林大学	博 ¹	重 ¹	459

软科世界一流学科排名

软科全球院系排名

软科中国最好大学排名

软科世界大学学术排名

软科世界大学学科领域排名

软科中国两岸四地大学排名

软科中国最好学科排名

各国大学排名

澳大利亚

奥地利

比利时

巴西

加拿大

中国

丹麦

芬兰

法国

德国

中国香港

爱尔兰

以色列

意大利

日本

荷兰



代码文件命名: 10-2pachongchem

 **修改实例24，编写软科世界一流学科排名2019-化学排名结果。**

1	2	前 1%	中国科学技术大学
博士学位授权点			
			1463
前 1%		北京大学	
1310	3	3	前 2%
一级学科博士学位授权点			
			1126
前 2%			
		南开大学	
	5	4	前 3%
一级学科博士学位授权点			
			2
6	5	前 3%	厦门大学
一级学科国家重点学科			
			926
7			



下周课程&课后作业



代码文件命名: 10-2pachongchem

✚ 修改实例24, 编写软科世界一流学科排名2019-化学排名结果。

```
<tr>
<th style="width:10%">2019 排名</th>
<th style="width:10%">2018 排名</th>
<th style="width:15%">百分位段</th>
<th style="width:25%">院校名称*</th>
<th style="width:15%">博士点</th>
<th style="width:15%">重点学科</th>
<th style="width:10%">总分</th>
</tr>
</thead>
<tbody>
```



下周课程&课后作业



24-化学学科排名.py - E:\课程\python\第11次课\上次习题\24-化学学科排名.py (3.7.3)

File Edit Format Run Options Window Help

```
#e24CrawUnivRanking.py
import requests
from bs4 import BeautifulSoup
allUniv = []
n=0
def getHTMLText(url):
    try:
        r = requests.get(url, timeout=30)
        r.raise_for_status()
        r.encoding = 'utf-8'
        return r.text
    except:
        return ""
def fillUnivList(soup):
    data = soup.find_all('tr', {'class': 'bgfd'}) #其它地方也有tr标签，为了确定位置正确，属性值也写上。
    for tr in data:
        n=0
        ltd = tr.find_all('td')
        if len(ltd)==0:
            continue
        singleUniv = []
        for td in ltd:
            n=n+1
            if (n == 5) or (n==6):
                if td.contents==[]: #如果标签为空，说明没有Img选项，说明没有博士点或者重点学科。
                    td.string='无'
                else:
                    td.string='有'
            singleUniv.append(td.string)
        allUniv.append(singleUniv)
def printUnivList(num):
    print("{:^10}{:^10}{:^10}{:^10}{:^10}{:^10}{:^10}{:^10}".format("2019排名", "2018排名", "百分段位", "院校排名", "博士点", "重点学科", "总分"))
    for i in range(num):
        u=allUniv[i]
        print("{:^10}{:^10}{:^10}{:^10}{:^10}{:^10}{:^10}{:^10}".format(u[0], u[1], u[2], u[3], u[4], u[5], u[6]))

def main():
    url = 'http://www.zuihaodaxue.com/BCSR/huaxue2019.html'
    html = getHTMLText(url)
    soup = BeautifulSoup(html, "html.parser")
    fillUnivList(soup)
    printUnivList(20)
main()
```



下周课程&课后作业

代码文件命名: 10-2pachongchem

修改实例24, 编写软科世界一流学科排名2019-化学排名结果。

```
==== RESTART: E:\课程\python\第11次课\上次习题\24-化学学科排名.py =====
2019排名 2018排名 百分段位 院校排名 博士点 重点学科 总分
1 2 前1% 中国科学技术大学 有 有 1463
2 1 前1% 北京大学 有 有 1310
3 3 前2% 复旦大学 有 有 1126
4 7 前2% 南开大学 有 有 979
5 4 前3% 清华大学 有 有 973
6 5 前3% 厦门大学 有 有 926
7 6 前4% 南京大学 有 有 828
8 8 前4% 吉林大学 有 有 755
9 9 前4% 浙江师范大学 有 有 597
10 10 前5% 中山大学 有 有 556
11 12 前5% 四川大学 有 有 522
12 12 前6% 兰州大学 有 有 500
13 11 前6% 苏州大学 有 有 494
14 14 前7% 武汉大学 有 有 482
15 15 前7% 湖南大学 有 有 465
16 16 前8% 上海交通大学 有 无 447
17 18 前8% 天津大学 有 无 362
18 20 前8% 华南理工大学 有 无 346
19 17 前9% 华东理工大学 有 无 302
20 19 前9% 华东师范大学 有 无 296
>>>
```



下周课程&课后作业

代码文件命名: 10-2nachongchem



```
#e24CrawUnivRanking.py
import requests
from bs4 import BeautifulSoup
allUniv = []
n=0
def getHTMLText(url):
    try:
        r = requests.get(url, timeout=30)
        r.raise_for_status()
        r.encoding = 'utf-8'
        return r.text
    except:
        return ""
def fillUnivList(soup):
    data = soup.find_all('tr', {'class': 'bgfd'}) #其它地方也有tr标签，为了确定位置正确，属性值也写上。
    for tr in data:
        n=0
        ltd = tr.find_all('td')
        if len(ltd)==0:
            continue
        singleUniv = []
        for td in ltd:
            n=n+1
            if (n == 5) or (n==6):
                if td.contents==[]: #如果标签为空，说明没有Img选项，说明没有博士点或者重点学科。
                    td.string='无'
                else:
                    td.string='有'
            singleUniv.append(td.string)
        allUniv.append(singleUniv)
def printUnivList(num):
    print("{:^10}{:^10}{:^10}{:^10}{:^10}{:^10}{:^10}{:^10}".format("2019排名", "2018排名", "百分段位", "院校排名", "博士点", "重点学科", "总分"))
    for i in range(num):
        u=allUniv[i]
        print("{:^10}{:^10}{:^10}{:^10}{:^10}{:^10}{:^10}{:^10}".format(u[0], u[1], u[2], u[3], u[4], u[5], u[6]))

def main():
    xueke=input('请输入需要排名的学科')
    url = 'http://www.zuihaodaxue.com/BCSR/'+xueke+' 2019.html'
    html = getHTMLText(url)
    soup = BeautifulSoup(html, "html.parser")
    fillUnivList(soup)
    printUnivList(20)
main()
```

改成任意学科都可以查询



下周课程&课后作业

代码文件命名: 10-2pachongchem

修改实例24, 编写软科世界一流学科排名2019-化学排名结果。

请输入需要排名的学科wulixue

2019排名 总分	2018排名	百分段位	院校排名	博士点	重点学科
1	1	前1%	中国科学技术大学	有	有 1408
2	2	前2%	清华大学	有	有 1198
3	3	前2%	北京大学	有	有 1014
4	4	前3%	南京大学	有	有 735
5	5	前3%	上海交通大学	有	有 626
6	6	前4%	复旦大学	有	有 558
7	7	前4%	吉林大学	有	有 459
8	8	前5%	浙江大学	有	有 353
9	9	前5%	华中科技大学	有	无 346
10	10	前6%	中山大学	有	有 315
11	11	前6%	山东大学	有	有 300
12	15	前7%	北京航空航天大学	有	无 239
13	32	前7%	北京理工大学	有	无 227
14	18	前8%	山西大学	有	有 218
15	20	前8%	武汉大学	有	有 215
16	12	前9%	南开大学	有	有 210
16	14	前9%	西安交通大学	有	无 210
18	13	前10%	苏州大学	有	无 199
19	16	前10%	哈尔滨工业大学	有	有 186
20	19	前11%	华东师范大学	有	有 177

>>>



新乡医学院

Python程序语言入门与应用

第十一章 Python 图形化界面开发基础

王海蛟

新乡医学院





学习目标



Talk is cheap. Show me the code.

— *Linus Torvalds* —

AZ QUOTES

Linux 的创始人 Linus Torvalds



本课概要



🔗 第11章 Python 图形化界面开发基础

🔗 11.1 GUI介绍

🔗 11.2 Tkinter开发基础

🔗 11.3 Tkinter组件开发详解

🔗 11.4 Tkinter库的事件

🔗 11.5 实例26：编程实现简易聊天窗口GUI



学习目标



基本要求



掌握



Tkinter库进行图形用户界面开发方法。



理解&了解



Python GUI编程的概念与应用。



Python GUI 编程的各种生态库

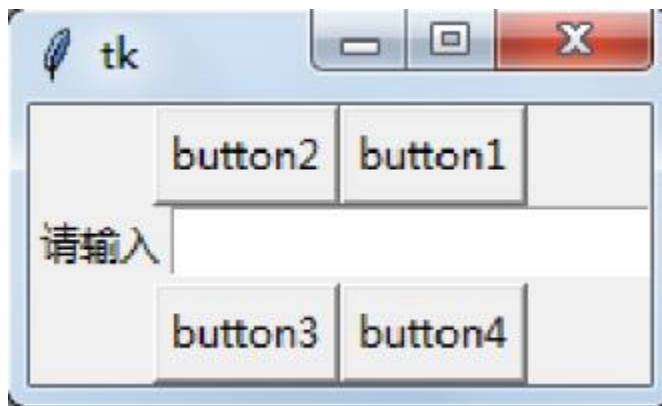


第11章 Python 图形化界面开发基础



11.1 GUI介绍

- Gui即图像用户界面的英文简写。使用图形用户界面（GUI），可以使程序更友好。
- Python作为一种“胶水性”语言，提供了众多GUI开发库的绑定，适合快速开发GUI。
- 目前常用的开发库有：**Python内置的Tkinter、Graphics、Turtle等，以及非常强大的PyQT等。**





第11章 Python 图形化界面开发基础



- 11.1 GUI介绍
- Tkinter—Python 标准GUI**
- Turtle—python内置的图形库。**

日常使用的大量客户端程序都属于GUI程序，即在一个界面上有很多功能块，包括：标签、按钮、输入框、菜单等。

开发GUI程序，1首先需要有一块空白画面，2然后在其上划分出不同的区域，放上不同的模块，3最后完成每一个模块的功能。



第11章 Python 图形化界面开发基础



11.1 GUI介绍

GUI程序开发基本步骤和过程：

- 首先要有底层的**根窗口对象**，在其基础上创建一个个**小窗口对象**。每一个窗口都是一个容器，可将所需的组件置于其中。每种GUI开发库都拥有大量的组件，一个GUI程序就是由各种不同功能的组件组成的，而**根窗口对象则包含了所有组件**。
- 组件本身也可以作为一个容器**，它可以包含其他组件，如下拉框。这种包含其他组件的称为父组件，反之，包含在其他组件中的组件称为子组件。这是一种相对的概念，对于有着多层包含的情况，某组件的父组件一般指的是直接包含它的组件。



第11章 Python 图形化界面开发基础



11.1 GUI介绍

- 构建出了GUI程序的每一个组件，只完成了程序的界面，**但此时只能看不能用，需要给每一个组件添加对应的功能。**
- 使用GUI程序时，会进行各种操作，如鼠标移动，按下或松开鼠标键，按下键盘按键等，**这些操作称为事件**。每个组件对应着一些行为，如在文本框中输入文本，单击按钮等，这些也称为事件。
- GUI程序启动的时候就一直**监控这些事件**，当某个事件发生的时候，就进行对应的处理并返回相应的结果。因此，GUI程序是由这一整套事件驱动的，这个过程称为**事件驱动处理**。



第11章 Python 图形化界面开发基础







11.1 GUI介绍

- ☞ 一个事件发生后，**GUI程序捕获该事件**、作出对应的处理并返回结果的过程称为**回调**。如计算器程序，单击了“=”按钮之后，便产生了一个事件，需要计算最终的结果，程序便开始对算式进行计算，返回最终结果并显示出来。这个计算并显示结果的过程即为回调。
- ☞ 当为程序需要的每一个事件都添加完相应的回调处理之后，整个GUI就完成了。






11.2Tkinter开发基础

-  Tk是Python默认的工具集（即图形库），Tkinter是Tk的Python接口，通过Tkinter可以方便地调用Tk进行图形界面开发。
-  Tk与其他开发库相比，不是最强大的，模块工具也不是非常的丰富。但它非常简单，所提供的功能开发一般的应用也完全够用了，且能在大部分平台上运行。
-  Python自带的IDEL也是用Tkinter开发的。
-  Tkinter的不足之处是缺少合适的可视化界面设计工具，需要通过代码来完成窗口设计和元素布局。



11.2Tkinter开发基础

-  Tkinter中提供了较为丰富的控件，完全能满足基本的GUI程序的需求。
-  由于Tkinter模块已经在Python中内置，所以在使用之前，只需将其导入即可。
-  两种导入方式为：

```
import tkinter as tk
```

导入tkinter，但没引入任何组件，在使用时需要使用tk前缀，如需要引入按钮，则表示为：tk.Button。






```
from tkinter import *
```

将tkinter中的所有组件一次性引入。



11.2Tkinter开发基础

 **利用Tkinter模块来引用Tk构建和运行GUI程序，通常需要5步：**

-  **导入Tkinter模块；**
-  **创建一个顶层窗口；**
-  **在顶层窗口的基础上构建所需要的GUI模块和功能；**
-  **将每一个模块与底层程序代码关联起来；**
-  **执行主循环。**



第11章 Python 图形化界面开发基础



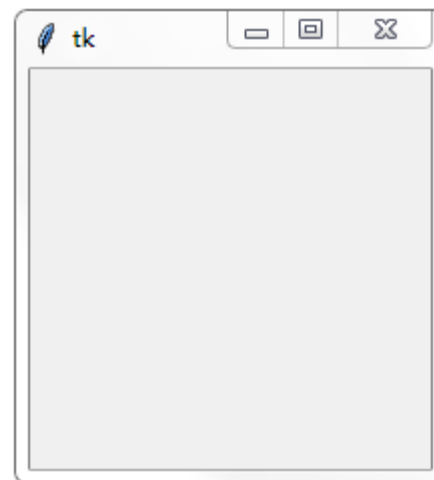
11.2Tkinter开发基础

微实例11.1 第一个GUI程序

```
import tkinter as tk
```

```
top=tk.Tk()
```

```
top.mainloop()
```





第11章 Python 图形化界面开发基础



11.2Tkinter开发基础

微实例11.2向第一个GUI程序添加组件

```
import tkinter as tk

top=tk.Tk()

label1=tk.Label(top,text='第一个标签')

label1.pack()

bn1=tk.Button(top,text='按钮1')

bn1.pack(side=tk.LEFT)

bn2=tk.Button(top,text='按钮2')

bn2.pack(side=tk.RIGHT)


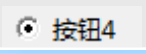
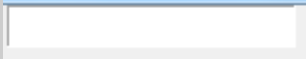
top.mainloop()
```





第11章 Python 图形化界面开发基础



组件	功能
Button	按钮。类似标签，但提供额外功能，如鼠标按下、释放及键盘操作事件
Canvas	画布。提供绘图功能（直线、椭圆、多边形、矩形），可以包含图形或位图
Checkbutton	选择按钮。一组方框，可以选择其中的任意个 
Radiobutton	单选按钮。一组方框，其中只有一个可被选择 
Entry	文本框。单行文字域，用来收集键盘输入 
Frame	框架。包含其他组件的纯容器
Label	标签。用来显示文字或图片
Listbox	列表框。一个选项列表，用户可以从中选择
Menu	菜单。单击后弹出一个选项列表，用户可以从中选择
Menubutton	菜单按钮。用来包含菜单的组件（有下拉式、层叠式）
Message	消息框。类似于标签，但可以显示多行文本
Scale	进度条。线性“滑块”组件，可设定起始值和结束值，显示当前位置的精确值
Scrollbar	滚动条。对其支持的组件（文本域、画布、列表框、文本框）提供滚动功能
Text	文本域。多行文字区域，可用来收集（或显示）用户输入的文字
Toplevel	顶级。类似框架，但提供一个独立的窗口容器。



11.2Tkinter组件开发详解

组件的共同属性

- dimensions : 尺寸**

- colors: 颜色**

- fonts: 字体**

- anchors: 锚点** 锚是用来定义文本的相对位置参考点

- relief styles: 浮雕式**

- bitmaps: 显示位图**

- cursors: 光标的外形**

说明：每种组件还有其各自特有的属性



❧ 11.2Tkinter组件开发详解

❧ 组件的几何管理方法

- ❧ Tk使用了一种包管理器来管理所有的组件，当定义完组件之后，需要调用`pack()`方法来控制组件的显示方式，若不调用`pack()`方法，组件将不会显示。
- ❧ 在交互环境下，编写Tkinter测试代码时，运行过`Tk()`（创建顶层窗口的函数）之后即进入主循环，可以看到顶层窗口。而若是运行py文件，一定要调用`mainloop()`方法进入主循环，方可看到顶层窗口。



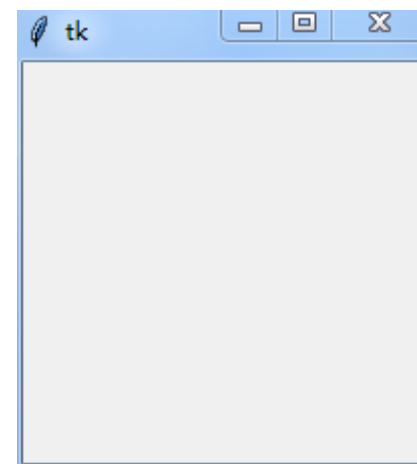
第11章 Python 图形化界面开发基础



11.2Tkinter组件开发详解

 创建GUI应用程序窗口代码模板：

```
from tkinter import *  
tk = Tk()  
#此处添加控件代码  
...  
tk.mainloop()
```





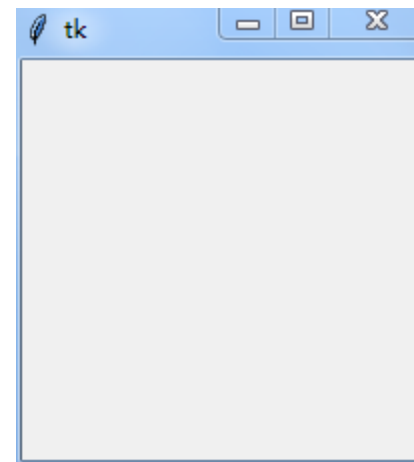
第11章 Python 图形化界面开发基础



11.2Tkinter组件开发详解

创建GUI应用程序窗口代码模板：

```
import tkinter as tk  
Gui=tk.Tk()  
#此处添加按钮  
  
• • • •  
Gui.mainloop()
```





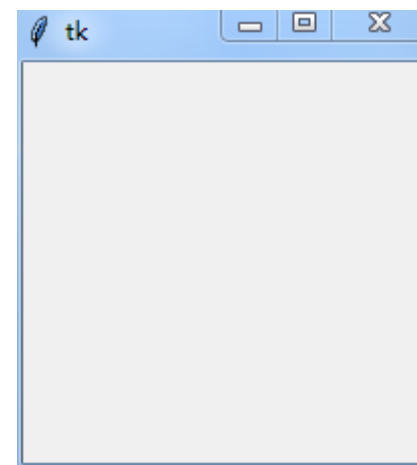
11.2 Tkinter组件开发详解

顶层窗口

顶层窗口也称为根窗口。

顶层窗口实际上是一个普通窗口，包括一个标题栏和窗口管理器所提供的窗口装饰部分，如最大化按钮等。

在一个Tkinter开发的应用程序中，只需要创建一个顶层窗口即可，且此窗口的创建必须是在其他窗口创建之前。





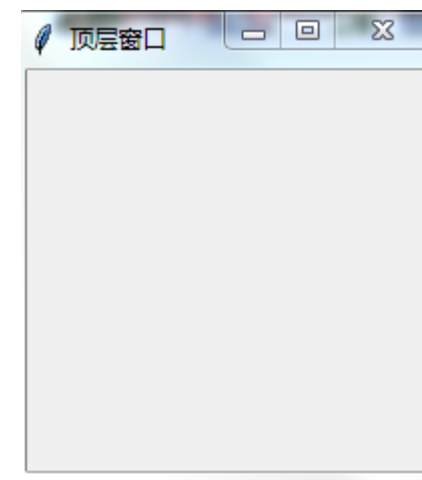
第11章 Python 图形化界面开发基础



11.2Tkinter组件开发详解

顶层窗口的创建

```
import tkinter as tk
top=tk.Tk()
top.title('顶层窗口') #不设置默认是TK
top.mainloop()
```





11.2Tkinter组件开发详解

按钮组件

- ✓按钮组件（**Button**）是tkinter最常用的图形组件之一，通过Button可以方便地与用户进行交互。
- ✓严格地说，按钮（**Button**）也可被看作标签，只是它可以捕获键盘和鼠标事件。
- ✓按钮可以禁用，禁用之后的按钮不能进行单击等任何操作。
- ✓如果将按钮放进**TAB**群中，就可以使用**TAB**键来进行跳转和定位。



第11章 Python 图形化界面开发基础



11.2Tkinter组件开发详解

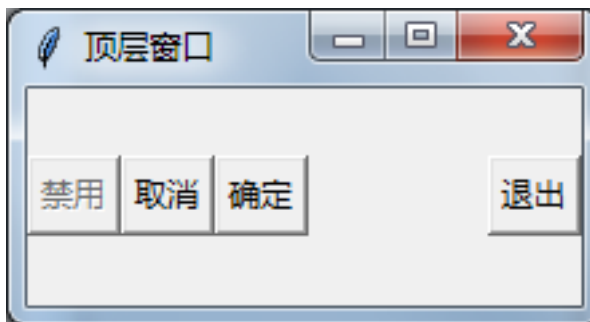
按钮组件创建微实例11.3

#创建按钮

```
from tkinter import *  
root=Tk()  
root.title("顶层窗口")
```

#使用state参数来设定按钮的状态

```
Button(root, text="禁用", state=DISABLED).pack(side=LEFT)  
Button(root, text="取消").pack(side=LEFT)  
Button(root, text="确定").pack(side=LEFT)  
Button(root, text="退出", command=root.quit).pack(side=RIGHT) #绑定了系统内置的退出回调  
  
root.mainloop()
```





第11章 Python 图形化界面开发基础



属性	描述
activebackground	按钮按下时背景颜色。默认是系统指定的颜色。
activeforeground	按钮按下时前景颜色。默认是系统指定的颜色。
anchor	采用何种方式锚定文字或者图片。默认是CENTER（居中模式）。可以选择如下方式： N, NE, E, SE, S, SW, W, NW, or CENTER.
background bg	按钮的背景颜色。默认是系统指定颜色
bitmap	bitmap形式显示按键。如果设置了images属性，则忽略bitmap属性。
borderwidth bd	按钮的边缘宽度。一般是1~2个像素值。
command	设置回调函数。当按钮被按下时，会调用该函数。如果该属性没有设置，按下按钮时，不会有任何动作发生。
compound	在按钮上同时显示文字和图片。默认的模式是如果提供了图片，会只显示图片。但是，如果将选项设为： CENTER：在图片中间叠加显示文字 BOTTOM：在图片下方显示文字 LEFT：在图片左边显示文字 RIGHT：在图片右边显示文字 TOP：在图片顶部显示文字 NONE：不显示文字



第11章 Python 图形化界面开发基础



cursor	当鼠标移动经过按钮的时候，显示光标
default	取值有normal,active和disabled三个。
disabledforeground	按钮被禁止使用时，按钮上的文字的颜色
font	按钮上文字的字体。只能选择一种字体显示。
foreground fg	按钮上文字或者位图的颜色
height	设置按钮的高度。如果是显示文字，数值是文字单位。如果是显示图片，数值单位为像素。如果没有设置，系统自动计算按钮的高度。
highlightbackground	当按钮失去焦点的时候，显示按钮边框的高亮颜色
highlightcolor	当按钮获得焦点的时候，显示按钮边框的高亮颜色
highlightthickness	设置高亮边框的宽度
image	设置按钮显示的图片。如果该选项被设置，会取代text或bitmap选项。
justify	当按钮有多行文字时，设置文字的对齐方式。可设的数值有： LEFT,RIGHT,CENTER
overrelief	当鼠标移动经过的时候，按钮显示浮雕效果。如果没有设定，会使用relief中的值。
padx	在水平方向上，按钮边框和文字或图像之间的填充（pad）
pady	在垂直方向上，按钮边框和文字或图像之间的填充（pad）
relief	按钮3D美化效果。通常情况下，按钮被按下时，是SUNKEN效果，释放时是RAISED效果。其他的可选项包括：GROOVE， RIDGE以及FLAT
state	按钮的状态，包括：NORMAL, ACTIVE 或者 DISABLED
takefocus	标识用户是否能够使用Tab键选中按钮。取值是True，False或者None
text	显示在按钮上的文字。如果使用了bitmap或者image，该选项被忽略

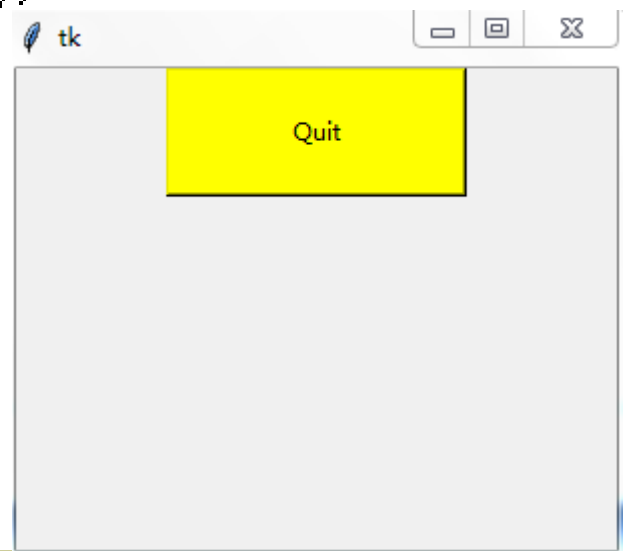


11.2Tkinter组件开发详解

按钮组件 微实例11.4

File Edit Format Run Options Window Help




```
import tkinter as tk
root=tk.Tk()
root.geometry('300x240')
b1=tk.Button(root,text='Quit',height=3,width=20,
              bg='yellow',command=root.destroy)
b1.pack()
root.mainloop()
```





11.2Tkinter组件开发详解

文本控件

-  一个GUI程序，接收用户的输入几乎是必不可少的。输入框（Entry）组件就是用来接收用户输入的最基本的组件。
-  可以为输入框设置默认值，也可以禁止用户输入。如果禁止输入，用户就不能改变输入框中的值了。
-  当用户输入的内容一行显示不下的时候，输入框会自动生成滚动条。



第11章 Python 图形化界面开发基础



11.2Tkinter组件开发详解

文本控件 创建微实例11.5

#创建输入框

```
from tkinter import *  
root=Tk()  
root.title("顶层窗口")
```

f1=Frame(root) #定义框架

Label(f1, text="标准输入框：").pack(side=LEFT, padx=5, pady=10)

e1=StringVar() #定义输入框内容

Entry(f1, width=50, textvariable=e1).pack(side=LEFT) #基本输入框

e1.set('输入框默认内容') #设置一般输入框默认内容

f1.pack()

f2=Frame(root) #定义框架

e2=StringVar()

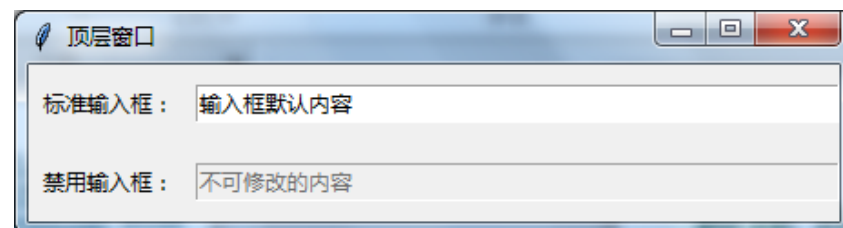
Label(f2, text="禁用输入框：").pack(side=LEFT, padx=5, pady=10)

Entry(f2, width=50, textvariable=e2, state=DISABLED).pack(side=LEFT) #禁用的输入框

e2.set("不可修改的内容") #设置禁用的输入框内容

f2.pack()

root.mainloop()





11.2Tkinter组件开发详解

文本控件

Entry参数组件常用参数

参数	描述
height	组件的高度（所占行数）
width	组件的宽度（所占字符个数）
fg	前景字体颜色
bg	背景颜色
show	将Entry框中的文本替换为指定字符，用于输入密码等，如设置 <code>show="*"</code>
state	设置组件状态，默认为 NORMAL ，可设置为： DISABLED —禁用组件， READONLY —只读



第11章 Python 图形化界面开发基础



11.2Tkinter组件开发详解

综合微实例11.6 将摄氏度转换为华氏度

#将摄氏度转换为华氏度

```
import tkinter as tk
```

```
def cToFClicked():
```

```
    cd = float(entryCd.get())
```

```
    labelcToF.config(text = "%.2f°C = %.2f°F" % (cd, cd*1.8+32))
```

```
top = tk.Tk()
```

```
top.title("Entry Test")
```

```
labelcToF = tk.Label(top, text = "Convert °C to °F...", height = 5,\n                      width = 20, fg = "blue")
```

```
labelcToF.pack()
```

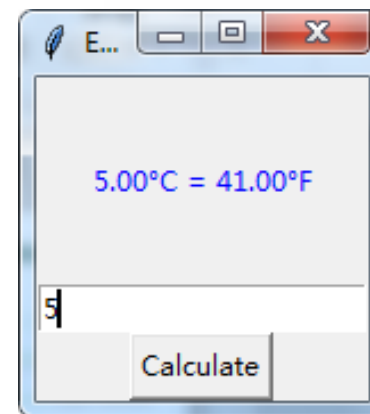
```
entryCd = tk.Entry(top, text = "0")
```

```
entryCd.pack()
```

```
btnCal = tk.Button(top, text = "Calculate", command = cToFClicked)
```

```
btnCal.pack()
```

```
top.mainloop()
```





11.2Tkinter组件开发详解

标签

- 标签组件可以用来显示图片和文本，通过在文本中添加换行符来控制换行，也可以通过控制组件的大小实现自动换行。
- 例：使用标签（Label）编写一个程序，在程序主体中显示“Hello World!”。

```
#使用Label实现Hello World程序
from tkinter import *
root=Tk()
root.title("顶层窗口")
label=Label(root, text="Hello World!") #定义标签
label.pack() #调用pack方法
root.mainloop()
```





11.2Tkinter组件开发详解

标签

Label组件常用参数

参数	描述
height	组件的高度（所占行数）
width	组件的宽度（所占字符个数）
fg	前景字体颜色
bg	背景颜色
justify	多行文本的对齐方式，可选参数为： LEFT 、 CENTER 、 RIGHT
padx	文本左右两侧的空格数（默认为1）
pady	文本上下两侧的空格数（默认为1）



❖ 11.2Tkinter组件开发详解

❖ 单选按钮&复选按钮

- ❖ **单选按钮 (Radiobutton)** 是一组排他性的选择框，只能从该组中选择一个选项，当选择了其中一项之后便会取消其他选项的选择。
- ❖ 要想使用单选按钮，必须将这一组单选按钮与一个相同的变量关联起来，由用户为这个变量选择不同的值。
- ❖ **与单选按钮相对的是复选按钮 (Checkbutton)**。复选按钮之间没有互斥作用，可以一次选择多个。
- ❖ 同样地，每一个按钮都需要与一个变量相关联，且每一个复选按钮关联的变量都是不同的。若像单选按钮一样，关联的是同一个按钮，则当选中其中一个的时候，会将所有按钮都选上。
- ❖ 可以给每一个复选按钮绑定一个回调，当该选项被选中时，执行该回调。



11.2Tkinter组件开发详解

单选按钮&复选按钮

#创建单选按钮

```
from tkinter import *  
root=Tk()  
root.title("顶层窗口")
```

foo=IntVar() #定义变量

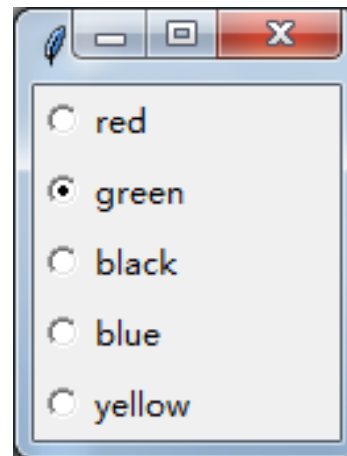
```
for text, value in [('red', 1), ('green', 2), ('black', 3),\  
                    ('blue', 4), ('yellow', 5)]:
```

```
    r=Radiobutton(root, text=text, value=value, variable=foo)
```

```
    r.pack(anchor=W) #向西对齐(可选参数:NSEW)
```

```
foo.set(2) #设定默认选项
```

```
root.mainloop()
```





11.2Tkinter组件开发详解

单选按钮&复选按钮

Radiobutton组件常用参数

参数	描述
variable	单选按钮索引变量，通过变量的值确定哪个单选按钮被选中。一组单选按钮使用同一个索引变量
value	单选按钮选中时变量的值
command	单选按钮选中时执行的命令（函数）



第11章 Python 图形化界面开发基础



11.2Tkinter组件开发详解

单选按钮&复选按钮

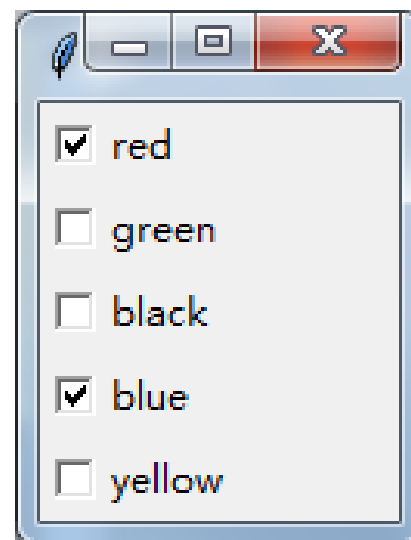
#创建基本复选按钮

```
from tkinter import *  
root=Tk()  
root.title("顶层窗口")
```

```
l=[('red', 1), ('green', 2), ('black', 3, ), ('blue', 4),\  
   ('yellow', 5)] #设定按钮的值
```

```
for text, value in l:  
    foo=IntVar()  
    c=Checkbutton(root, text=text, variable=foo)  
    c.pack(anchor=W)
```

```
root.mainloop()
```





11.2Tkinter组件开发详解

单选按钮&复选按钮

Checkbutton组件常用参数

参数	描述
variable	复选按钮索引变量，通过变量的值确定哪些复选按钮被选中。每个复选按钮使用不同的变量，使复选按钮之间相互独立
onvalue	复选按钮选中（有效）时变量的值
offvalue	复选按钮未选中（无效）时变量的值
command	复选按钮选中时执行的命令（函数）



第11章 Python 图形化界面开发基础



11.2Tkinter组件开发详解

绘图组件

绘图组件（**Canvas**—画布）可以在**GUI**中实现**2D**图形的绘制，相当于画板。组件内置了多种绘图函数，可以通过简单的**2D**坐标绘制直线、矩形、圆形、多边形等。



第11章 Python 图形化界面开发基础



11.2 Tkinter组件开发详解

绘图组件

#绘图应用

```
import tkinter as tk
```

```
def drawCircle(self, x, y, r, **kwargs):  
    print(kwargs) #两个星号的参数传进来的是字典形式，一个星号的参数传进来的是元组  
    return self.create_oval(x-r, y-r, x+r, y+r, **kwargs)
```

```
top = tk.Tk()  
top.title("Canvas Test")
```

```
cvs = tk.Canvas(top, width = 600, height = 400)  
cvs.pack()
```

```
cvs.create_line(50, 50, 50, 300)
```

```
cvs.create_line(100, 50, 200, 300, fill = "red", dash = (4, 4), arrow = tk.LAST)
```

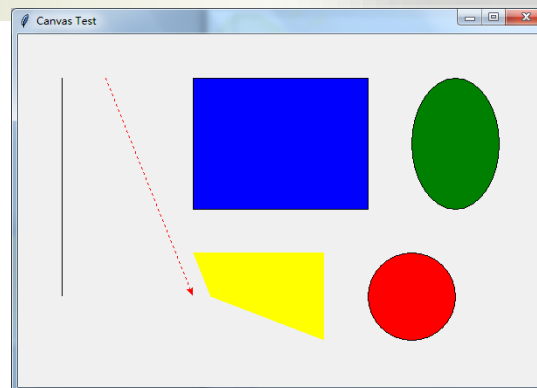
```
cvs.create_rectangle(200, 50, 400, 200, fill = "blue")
```

```
cvs.create_oval(450, 50, 550, 200, fill = "green" )
```

```
drawCircle(cvs, 450, 300, 50, fill = "red")
```

```
cvs.create_polygon(200, 250, 350, 250, 350, 350, 220, 300, fill="yellow")
```



```
top.mainloop()
```





11.2Tkinter组件开发详解

框架

-  **框架 (Frame) 相对于其他组件而言, 它只是个容器, 因为它没有方法, 但它可以捕获键盘和鼠标的事件来进行回调。**
-  **框架一般用作包含一组控件的主体, 且可以定制外观。**

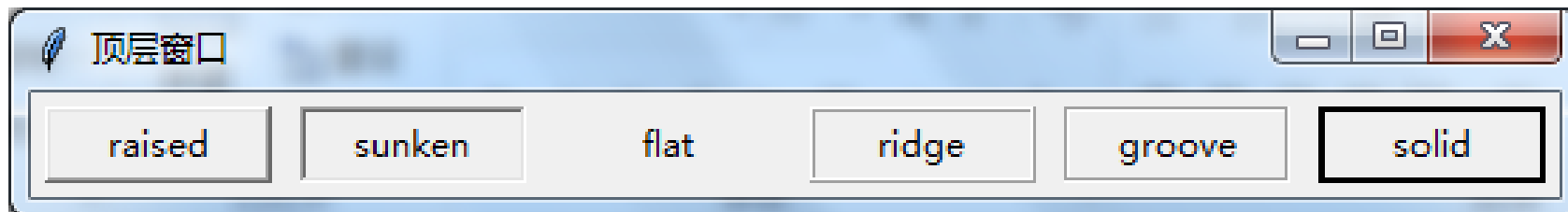


11.2Tkinter组件开发详解

框架

#创建不同样式的框架

```
from tkinter import *
root=Tk()
root.title("顶层窗口")
for relief in [RAISED, SUNKEN, FLAT, RIDGE, GROOVE, SOLID]:
    f=Frame(root, borderwidth=2, relief=relief) #定义框架
    #定义标签, 且使用side参数设定排列方式
    Label(f, text=relief, width=10).pack(side=LEFT)
    #显示框架, 并设定向左排列, x和y轴的宽度均为5个像素
    f.pack(side=LEFT, padx=5, pady=5)
root.mainloop()
```





11.2Tkinter组件开发详解

框架

padx	水平方向上的边距
pady	垂直方向上的边距
relief	<ol style="list-style-type: none">1. 指定边框样式2. 默认值是 FLAT3. 另外你还可以设置 SUNKEN , RAISED , GROOVE 或 RIDGE4. 注意，如果你要设置边框样式，记得设置 borderwidth 或 bd 选项不为 0，才能看到边框
takefocus	<ol style="list-style-type: none">1. 指定该组件是否接受输入焦点（用户可以通过 tab 键将焦点转移上来）2. 默认值是 False
visual	<ol style="list-style-type: none">1. 为新窗口指定视觉信息2. 该选项没有默认值
width	<ol style="list-style-type: none">1. 设置 Frame 的宽度2. 默认值是 0



第11章 Python 图形化界面开发基础




11.2Tkinter组件开发详解

框架

padx	水平方向上的边距
pady	垂直方向上的边距
relief	<ol style="list-style-type: none">1. 指定边框样式2. 默认值是 FLAT3. 另外你还可以设置 SUNKEN , RAISED , GROOVE 或 RIDGE4. 注意, 如果你要设置边框样式, 记得设置 borderwidth 或 bd 选项不为 0 , 才能看到边框
takefocus	<ol style="list-style-type: none">1. 指定该组件是否接受输入焦点 (用户可以通过 tab 键将焦点转移上来)2. 默认值是 False
visual	<ol style="list-style-type: none">1. 为新窗口指定视觉信息2. 该选项没有默认值
width	<ol style="list-style-type: none">1. 设置 Frame 的宽度2. 默认值是 0



11.3Tkinter库的事件

-  用户通过鼠标、键盘、游戏控制设备在与图形界面交互时，就会触发事件。tkinter事件通常采用了将事件名称放置于尖括号内的字符串表示，尖括号中的内容我们称之为事件类型。事件类型有其通用的定义方式。如下

`<[modifier-]...type[-detail]>`

其中方括号内的内容为可选参数； modifier为组合键的定义，例如，同时按下Ctrl键；type为通用类型，例如，键盘按键（KeyPress）；detail用于具体信息，如按下键盘中 ‘B’ 键。



11.3 Tkinter库的事件

常用事件类型如下：

【鼠标单击事件】

- <Button-1>：单击鼠标左键
- <Button-2>：单击鼠标中间键（如果有）
- <Button-3>：单击鼠标右键
- <Button-4>：向上滚动滑轮
- <Button-5>：向下滚动滑轮

【鼠标双击事件】

- <Double-Button-1>：鼠标左键双击
- <Double-Button-2>：鼠标中键双击
- <Double-Button-3>：鼠标右键双击

【鼠标释放事件】

- <ButtonRelease-1>：鼠标左键释放
- <ButtonRelease-2>：鼠标中键释放
- <ButtonRelease-3>：鼠标右键释放

【鼠标按下并移动事件（即拖动）】

- <B1-Motion>：左键拖动
- <B2-Motion>：中键拖动
- <B3-Motion>：右键拖动

【鼠标其他操作】

- <Enter>：鼠标进入控件（放到控件上面）
- <FocusIn>：控件获得焦点
- <Leave>：鼠标移出控件
- <FocusOut>：控件失去焦点



11.3Tkinter库的事件

常用事件类型如下：

【键盘按下事件】

- `<Key>`：键盘按下，事件event中的keycode,char都可以获取按下的键值
- `<Return>`：键位绑定，回车键，其它还有`<BackSpace>`,`<Escape>`,
`<Left>`,`<Up>`,`<Right>`,`<Down>`等等

【控件属性改变事件】

`<Configure>`：控件大小改变，新的控件大小会存储在事件event对象中的 width 和 height 属性传递，部分平台上该事件也代表控件位置改变。


【组合使用】


- `<Control-Shift-Alt-KeyPress-A>`：同时按下Ctrl+Shift+Alt+A等4个键
- `<KeyPress-A>`：按下键盘中的'A'键

游戏设备使用参见设备供应商提供的API函数，这里不再赘述，常用的事件类型就是这些了。大家没有必要全部记住，这些事件都是一种类英文的描述，只要记住主要的几个，按照`<[modifier-]...type[-detail]>`进行组合就可以了。



11.3Tkinter库的事件

 事件的绑定：


 事件需要绑定具体的对象才能够响应。事件的绑定可以通过如下4个方式实现：


- 1、创建组件对象实例时，可以通过其命名参数command指定事件处理函数，如为Button控件绑定单击时间，当控件被单击时执行clickhandler函数处理。

b = Button(root, text='按钮', command=clickhandler)



11.3Tkinter库的事件

 事件的绑定:

 事件需要绑定具体的对象才能够响应。事件的绑定可以通过如下4个方式实现:

2、调用组件对象实例方法bind, 可以为指定组件实例绑定事件。


w.bind('<event>', eventhandler, add='')


其中, <event>为事件类型, eventhandler为事件处理函数, 可选参数add默认为", 表示事件处理函数替代其他绑定, 如果为 '+', 则加入事件处理队列。
如绑定组件对象, 使得Canvas组件实例c可以处理鼠标右键单击事件(处理函数名称为eventhandler), 代码如下:

c=Canvas(); c.bind('Button-3', eventhandler)



11.3Tkinter库的事件

 事件的绑定:

 事件需要绑定具体的对象才能够响应。事件的绑定可以通过如下4个方式实现:

3、**类绑定**: 调用组件对象实例方法bind_class函数, 可以为特定类绑定事件


w.bind_class('Widget', '<event>', eventhandler, add='')


其中, Widget为组件类; <event>为事件; eventhandler为事件处理函数.如绑定组件类, 使得所有Canvas组件实例可以处理鼠标中键事件(事件处理函数为eventhandler):

c = Canvas(); c.bind_class('Canvas', '<Button-2>', eventhandler)



11.3Tkinter库的事件

 事件的绑定:

 事件需要绑定具体的对象才能够响应。事件的绑定可以通过如下4个方式实现:

4、**程序界面绑定**: 调用组件对象实例方法bind_all函数, 可以为所有组件类型绑定事件:

w.bind_all('<event>', eventhandler, add='')


同上, 其中<event>为事件; eventhandler为事件处理函数.如将PrintScreen键与程序中所有组件对象绑定, 使得程序界面能处理打印屏幕的键盘事件:

```
c = Canvas(); c.bind('<Key-Print>', printscreen)
```



11.3Tkinter库的事件

事件处理函数的编写

-  对于能通过command传入的函数，其参数没有必须指定第一个参数为event的要求。但是通过bind（含bind_class、bind_all，当然，bind_class第一个参数为className，但其后必须是Event实例）绑定的事件，在定义函数方法时（事件处理可以定义为函数，也可以定义为对象的方法），两者都带一个参数event。触发事件调用处理函数时，将传递Event对象实例。

函数定义:

```
def handlerName(event):
```


类中定义:

```
def handlerName(self, event):
```



11.3Tkinter库的事件

Event事件对象参数属性

 通过传递Event事件对象的属性，可以获取相关参数备程序使用。常用的Event事件参数有以下几种。

- widget : 产生该事件的控件
- x, y : 当前鼠标位置
- x_root, y_root : 当前鼠标相对于屏幕左上角 (0 , 0) 的位置，以像素px为单位。
- char : 字符代码 (限键盘事件) ，作为字符串返回
- keysym : 关键符号 (限键盘事件)
- keycode : 关键代码 (限键盘事件)
- num : 按钮号码 (限鼠标按钮事件)
- width, height : 小部件的新大小 (以像素px为单位) (限配置事件) 。
- type : 事件类型



第11章 Python 图形化界面开发基础



11.3Tkinter库的事件



事件实例演示。

```
import tkinter as tk
def motder(event):
    s='【鼠标事件】\n\n产生事件的控件：{0}\n鼠标位置： (x={1},y={2})'\
      '\n鼠标事件相对屏幕左上角位置：(x={3},y={4})\n'\
      '鼠标事件类型：{5}'.format(event.widget,event.x,event.y,event.x_root,
                                  event.y_root,'鼠标再移动' if event.type=='6' else '')
    la['text']=s

top=tk.Tk()
top.title='事件测试'
top.geometry('800x600')
top.resizable(width=False,height=False)
la=tk.Label(top,bg='red',width=330,fg='yellow',
            font=('times',14,'bold'))
la.pack(side=tk.LEFT,fill=tk.BOTH,expand=tk.YES,anchor=tk.E,
        padx=10,pady=10)
la.bind('<Motion>',motder)
top.mainloop()
```



第11章 Python 图形化界面开发基础



11.3Tkinter库的事件



事件实例演示。

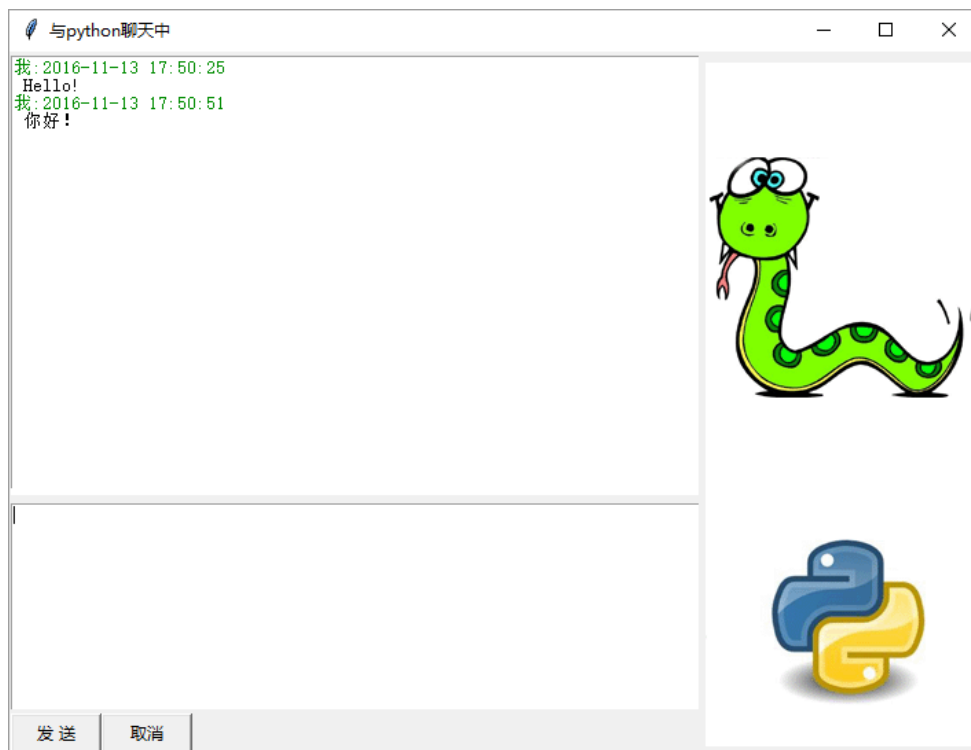




第11章 Python 图形化界面开发基础



11.5实例26：编程实现简易聊天窗口GUI



1.聊天窗口布局:

左上：聊天历史信息显示

左中：当前信息编辑区域

左下：按钮区域

右侧：信息显示区域



第11章 Python 图形化界面开发基础



11.5实例26：编程实现简易聊天窗口GUI

2.Frame控件规划：

容器区域布局：

frmLT、frmLC、frmLB、frmRT

#创建frame容器

```
frmLT = Frame(width=500, height=320, bg='white')  
frmLC = Frame(width=500, height=150, bg='white')  
frmLB = Frame(width=500, height=30)  
frmRT = Frame(width=200, height=500)
```



第11章 Python 图形化界面开发基础



❧ 11.5实例26：编程实现简易聊天窗口GUI

3.统一控件对象命名规则：

❧ “控件类型” + “功能”

❧ frmLT, frame+LeftTop

❧ txtMsg, text控件+消息

❧ btnSend, button控件+发送

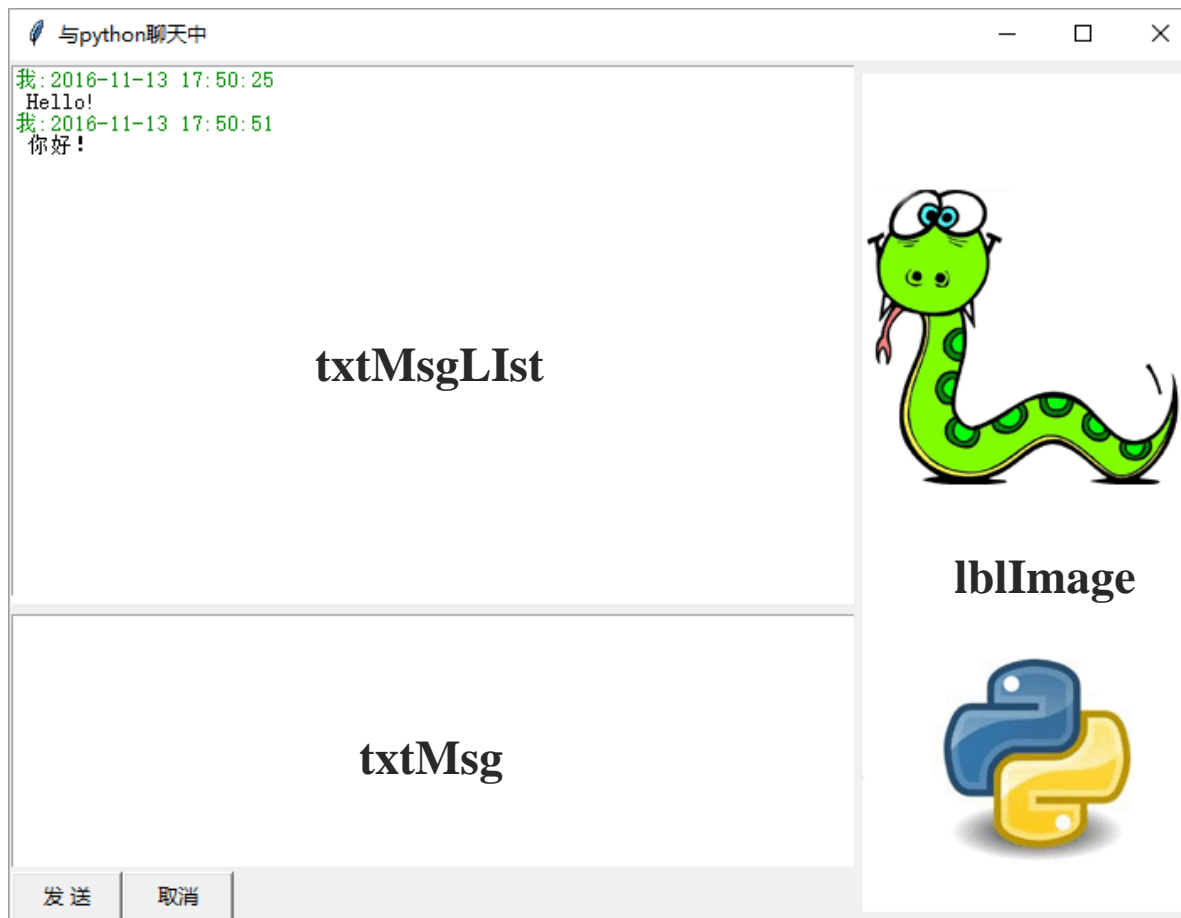


第11章 Python 图形化界面开发基础



11.5实例26：编程实现简易聊天窗口GUI

3.统一控件对象命名规则：



btnSend

btnCancel



第11章 Python 图形化界面开发基础



11.5实例26：编程实现简易聊天窗口GUI

4.创建各窗口控件

#创建控件

```
txtMsgList = Text(frmLT)
txtMsgList.tag_config('greencolor', foreground='#008C00') #创建tag
txtMsg = Text(frmLC);
txtMsg.bind("<KeyPress-Up>", sendMsgEvent)
btnSend = Button(frmLB, text='发 送', width = 8, command=sendMsg)
btnCancel = Button(frmLB, text='取消', width = 8, command=cancelMsg)
imgInfo = PhotoImage(file = "python.gif")
lblImage = Label(frmRT, image = imgInfo)
lblImage.image = imgInfo
```



11.5实例26：编程实现简易聊天窗口GUI

5. sendMsg()

回调函数

- 功能：在消息框列表输出历史消息，同时将当前时间、当前信息作为最后一条历史消息。然后删除消息发送框的内容。

```
def sendMsg():#发送消息
    strMsg = '我:' + time.strftime("%Y-%m-%d %H:%M:%S",
                                    time.localtime()) + '\n '
    txtMsgList.insert(END, strMsg, 'greencolor')
    txtMsgList.insert(END, txtMsg.get('0.0', END))
    txtMsg.delete('0.0', END)
```



第11章 Python 图形化界面开发基础



11.5实例26：编程实现简易聊天窗口GUI

5. sendMsgEvent()

```
def sendMsgEvent(event): #发送消息事件
    if event.keysym == "Up":
        sendMsg()
```

cancelMsg()

```
def cancelMsg():#取消消息
    txtMsg.delete('0.0', END)
```

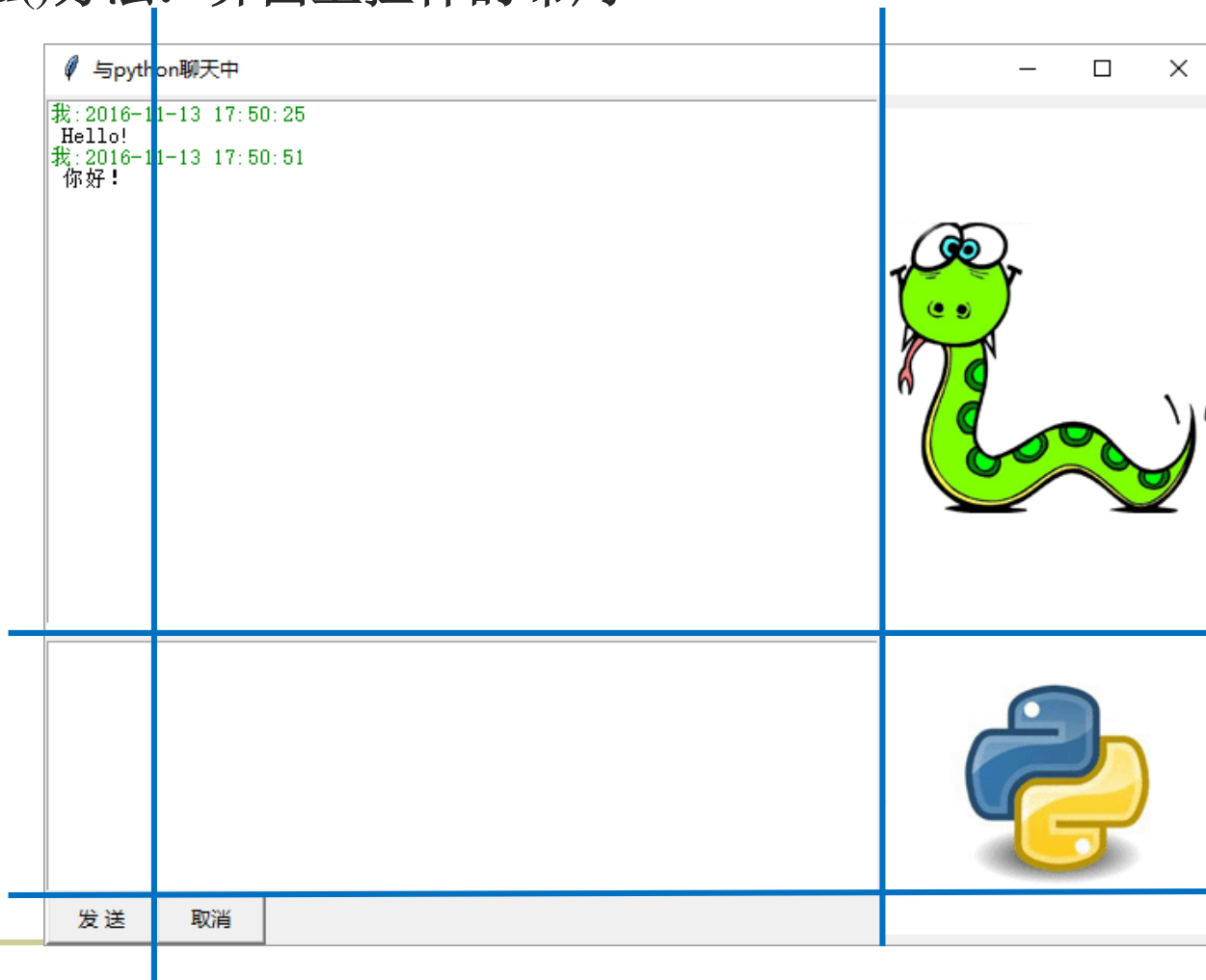


第11章 Python 图形化界面开发基础



11.5实例26：编程实现简易聊天窗口GUI

6.grid()方法：界面上控件的布局





第11章 Python 图形化界面开发基础



11.5实例26：编程实现简易聊天窗口GUI

6.grid()方法：界面上控件的布局

#窗口布局

```
frmLT.grid(row=0, column=0, columnspan=2, padx=1, pady=3)
frmLC.grid(row=1, column=0, columnspan=2, padx=1, pady=3)
frmLB.grid(row=2, column=0, columnspan=2)
frmRT.grid(row=0, column=2, rowspan=3, padx=2, pady=3)
```

#固定大小

```
frmLT.grid_propagate(0)
frmLC.grid_propagate(0)
frmLB.grid_propagate(0)
frmRT.grid_propagate(0)
```

```
btnSend.grid(row=2, column=0)
btnCancel.grid(row=2, column=1)
lblImage.grid()
txtMsgList.grid()
txtMsg.grid()
```

#主事件循环

```
t.mainloop()
```




第11章 Python 图形化界面开发基础



11.5实例26：编程实现简易聊天窗口GUI

最终代码：

```
#ChatWin.py
from tkinter import *
import time

def main():

    def sendMsg():#发送消息
        strMsg = '我:' + time.strftime("%Y-%m-%d %H:%M:%S",
                                         time.localtime()) + '\n '
        txtMsgList.insert(END, strMsg, 'greencolor')
        txtMsgList.insert(END, txtMsg.get('0.0', END))
        txtMsg.delete('0.0', END)

    def cancelMsg():#取消消息
        txtMsg.delete('0.0', END)

    def sendMsgEvent(event): #发送消息事件
        if event.keysym == "Up":
            sendMsg()

    #创建窗口
    t = Tk()
    t.title('与python聊天中')

    #创建frame容器
    frmLT = Frame(width=500, height=320, bg='white')
    frmLC = Frame(width=500, height=150, bg='white')
    frmLB = Frame(width=500, height=30)
    frmRT = Frame(width=200, height=500)
```



第11章 Python 图形化界面开发基础



11.5实例26：编程实现简易聊天窗口GUI

最终代码：

```
#创建控件
txtMsgList = Text(frmLT)
txtMsgList.tag_config('greencolor', foreground='#008C00') #创建tag
txtMsg = Text(frmLC);
txtMsg.bind("<KeyPress-Up>", sendMsgEvent)
btnSend = Button(frmLB, text='发送', width = 8, command=sendMsg)
btnCancel = Button(frmLB, text='取消', width = 8, command=cancelMsg)
imgInfo = PhotoImage(file = "python.gif")
lblImage = Label(frmRT, image = imgInfo)
lblImage.image = imgInfo

#窗口布局
frmLT.grid(row=0, column=0, columnspan=2, padx=1, pady=3)
frmLC.grid(row=1, column=0, columnspan=2, padx=1, pady=3)
frmLB.grid(row=2, column=0, columnspan=2)
frmRT.grid(row=0, column=2, rowspan=3, padx=2, pady=3)
#固定大小
frmLT.grid_propagate(0)
frmLC.grid_propagate(0)
frmLB.grid_propagate(0)
frmRT.grid_propagate(0)

btnSend.grid(row=2, column=0)
btnCancel.grid(row=2, column=1)
lblImage.grid()
txtMsgList.grid()
txtMsg.grid()

#主事件循环
t.mainloop()

if __name__ == '__main__':
    main()
```

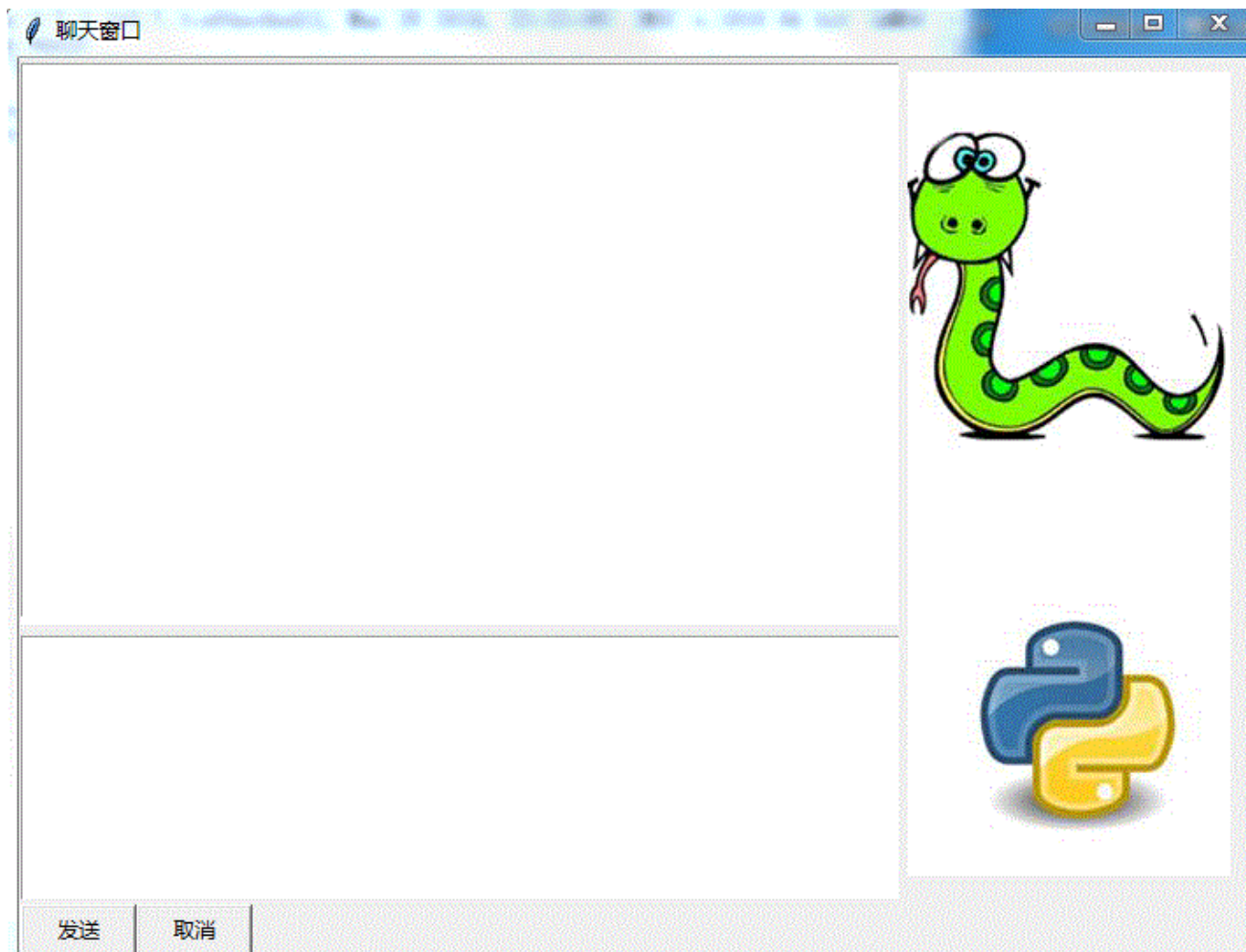


第11章 Python 图形化界面开发基础



11.5 实例26：编程实现简易聊天窗口GUI

运行结果：





下周课程&课后作业



第11章 Python 图形化界面开发基础

课后练习

编写一个具有一个按钮空间的GUI。

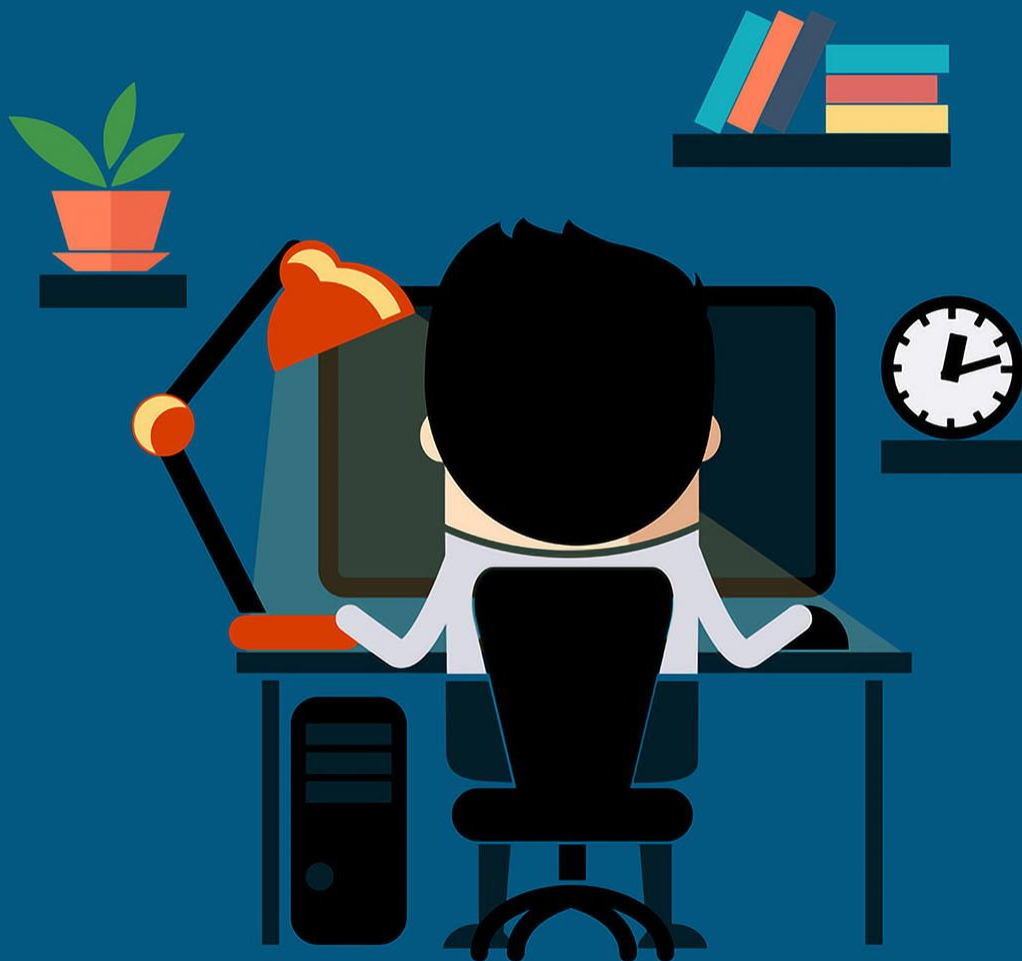
代码文件命名: 11-1BTnGui

在GUI窗体上面显示鼠标当前的位置

代码文件命名: 11-2Mospos

.py代码文件打包(11.学号
+姓名)发送到
python_xxmu@163.com

编程辣么好，还等什么？开始学习吧！



Programing is an Art