
一、threading多线程编程

```
import threading, time

def function():
    time.sleep(seconds) #程序延时seconds秒运行

t = threading.Thread(target = function, args = [], name = "gived name")
t.start()
```

二、tkinter库的GUI编程

1.常用的两种变量类型

```
intV = tkinter.IntVar()
stringV = tkinter.StringVar()

intV.set(num) #给变量设置一个值
stringV.get() #获取该变量的值
# 如果是ListBox组件
stringV.set(item1, item2, item3...)
```

2. Tk()创建的主窗口

```
win = tkinter.Tk()
win.title("mainwindow title") #设置主窗口的标题
win.geometry("1920x1080") #设置主窗口大小（小写字母x）
win.resizable(False, False) #主窗口不允许放缩，默认可以

win.mainloop() #开始监听主窗口

win.destroy() #关闭主窗口
```

3. 几何管理的三种方式

```
共有属性：
padx:x方向的外边距
pady:y方向的外边距
```

ipadx:x方向的内边距
ipady: y方向的内边距

(1). pack()包

1- side: 空间位于容器的方位:
 参数: left, top (默认), right, bottom
2- fill : 填充:
 x:水平方向填充
 y:竖直方向填充
 both:水平和竖直方向填充
none:不填充
3- expand:
 yes:扩展整个空白区, no:不扩展
4- anchor:
 N:北上,E:东右,S:南下,W:西左,CENTER:中间

(2). grid包

column: 组件所在的列起始位置;
columnspan: 组件的列宽;
row: 组件所在的行起始位置;
rowspan: 组件的行宽;

(3). place包

#用坐标来放置组件, 最不可能用到的方式, 因为wfp自己必不可能会
1- 使用绝对位置 place(x=0, y=0)

2- 使用相对坐标 lb.place(relx = 1, rely = 0.5, anchor = CENTER)

3- 同时制定多个组件
lb = Label(root, text='hello Place')
v = IntVar()
for i in range(5):
 Radiobutton(
 root,
 text='Radio' + str(i),
 variable=v,
 value=i
).place(x=80 * i, anchor=NW)

```
root.mainloop()
```

4- 同时使用相对和绝对坐标

```
# 先设置相对坐标为(0.5,0.5),再使用(-200,-200)将坐标作偏移(-200,-200)
lb1.place(relx=0.5, rely=0.5, anchor=CENTER, x=-200, y=-200)
```

5-使用in来指定放置的容器

```
# 创建一个Label
label.place(relx=0.5, rely=0.5, anchor=CENTER)
bt = Button(root, text='button in root', fg='yellow')
bt.place(in_=label)
#指定bt放的位置是label的默认(0,0)处
```

6-深入in用法

```
lb1.place(in_=fm1, relx=0.5, rely=0.5, anchor=CENTER)
bt1.place(in_=fm3, anchor=W)
```

anchor:	组件对齐方式;
x:	组件左上角的x坐标;
y:	组件右上角的y坐标;
relx:	组件相对于窗口的x坐标, 应为0-1之间的小数;
rely:	组件相对于窗口的y坐标, 应为0-1之间的小数;
width:	组件的宽度;
height:	组件的高度;
relwidth:	组件相对于窗口的宽度, 0-1;
relheight:	组件相对于窗口的高度, 0-1;

4. 常用组件

(1). tkinter.Button

anchor:	指定按钮上文本的位置;
background(bg)	指定按钮的背景色;
bitmap:	指定按钮上显示的位图;
borderwidth(bd)	指定按钮边框的宽度;
command:	指定按钮消息的回调函数;
cursor:	指定鼠标移动到按钮上的指针样式;
font:	指定按钮上文本的字体;
foreground(fg)	指定按钮的前景色;
height:	指定按钮的高度;
image:	指定按钮上显示的图片;
state:	指定按钮的状态 (disabled);
text:	指定按钮上显示的文本;
width:	指定按钮的宽度;
padx	设置文本与按钮边框x的距离, 还有pady;

<code>activeforeground</code>	按下时前景色
<code>textvariable</code>	可变文本，与StringVar等配合着用

(2). tkinter.Entry/Text

<code>background(bg)</code>	文本框背景色；
<code>foreground(fg)</code>	前景色；
<code>selectbackground</code>	选定文本背景色；
<code>selectforeground</code>	选定文本前景色；
<code>borderwidth(bd)</code>	文本框边框宽度；
<code>font</code>	字体；
<code>show</code>	文本框显示的字符，若为*，表示文本框为密码框；
<code>state</code>	状态；
<code>width</code>	文本框宽度
<code>textvariable</code>	可变文本，与StringVar等配合着用

(3). tkinter.Label

<code>Anchor</code>	标签中文本的位置；
<code>background(bg)</code>	背景色；
<code>foreground(fg)</code>	前景色；
<code>borderwidth(bd)</code>	边框宽度；
<code>width</code>	标签宽度；
<code>height</code>	标签高度；
<code>bitmap</code>	标签中的位图；
<code>font</code>	字体；
<code>image</code>	标签中的图片；
<code>justify</code>	多行文本的对齐方式；
<code>text</code>	标签中的文本，可以使用'\n'表示换行
<code>textvariable</code>	显示文本自动更新，与StringVar等配合着用

(4). tkinter.ListBox

```
import tkinter

root = tkinter.Tk()
root.title("hello world")
root.geometry()

def print_item(event):
    res = lb.get(lb.curselection())
    print(res)
```

```

var = tkinter.StringVar()
lb = tkinter.Listbox(root, listvariable=var)
# 放入列表的值
list_item = [1, 2, 3, 4] # 控件的内容为1 2 3 4
for item in list_item:
    lb.insert('end', item)
# lb.delete(2, 4) # 删除指定此时控件的内容为1 3

# 放入元组的值
# var.set(('a1', 'a2', 'A3', 'A4')) # 重新设置了, 这时控件的内容就编程var的内容了
print(var.get())
# 绑定事件函数
lb.bind('<ButtonRelease-1>', print_item)
lb.pack()
root.mainloop()

```

(5). tkinter.Scrollbar

```

import tkinter

# Listbox与Scrollbar绑定
root = tkinter.Tk()
lb = tkinter.Listbox(root)
sl = tkinter.Scrollbar(root)
sl.pack(side='right', fill='y')

# side指定Scrollbar为居右; fill指定填充整个剩余区域, 到WM在时候再详细介绍这几个属性。
# 下面的这句是关键: 指定Listbox的yscrollbar的回调函数为Scrollbar的set
lb['yscrollcommand'] = sl.set
for i in range(100):
    lb.insert('end', str(i))

# side指定Listbox为居左
lb.pack(side='left')

# 下面的这句是关键: 指定Scrollbar的command的回调函数是Listbar的yview
sl['command'] = lb.yview
root.mainloop()

```

三、flask包web编程

1. 创建并运行一个app

```

import flask *

app = Flask(__name__)
app.secret_key = "123"

```

```
if __name__ == "__main__":  
    app.run(host = "127.0.0.1", port = 8081)
```

2. 门户信息显示

(1). 一级路由

```
@app.route('/')  
def index():  
    title1 = ""  
    content1 = ""  
    contents2 = ['content1', 'content2', 'content3']  
    return render_template('index.html', title = title1, content = content1,  
contents = contents2)  
  
# index.html  
<!DOCTYPE html>  
<html>  
    <head> //head里包含 meta 和 title  
        <meta charset="UTF-8">  
        <title>  
            {{title}} //传参  
        </title>  
        <style>  
            *{  
                margin:0 auto;  
                padding:0;  
                border:0;  
            }  
            #header  
            {  
                width:100%;  
                height:100%;  
                border-bottom:1px dashed gray;  
                text-align: center;  
                line-height: 80px;  
                vertical-align: middle;  
                font-size: 60%;  
                padding-top: 10px;  
            }  
            #content  
            {  
                width: 100%; height: 730px;  
                border: 1px dashed grey;  
                text-align: center;  
                line-height: 80px;  
                font-size: 60px;  
                padding-top: 10px;
```

```

    }
    #footer
    {
        width: 100%; height: 100px;
        border-top: 1px dashed gray;
        text-align: center;
        line-height: 25px;
        vertical-align: middle;
        font-size: 20px;
        padding-top: 20px;
        font-weight: bold;
    }
</style>
<script>
</script>
</head>
<body>
    <div>
        <div id="header">
            {{header}} //传参
        </div>
        <div id="content">
            <br/>
            {{content}}
            <br/>
            <a href="http://127.0.0.1:8081/test1">please click
part 1, enter show_test1.html</a><br/>
            <a href='http://127.0.0.1:8081/test2'>please click
part 2, enter show_test2.html</a><br/>
            <a href='http://127.0.0.1:8081/test3'>please click
part 3, enter show_test2.html</a><br/>
            {% for content in contents %}
                {{contets}}&nbsp;<br/>
            {% endfor %}
            //for循环进行内容嵌入
        </div>
        <div id="footer">
            technology support and all rights reserve
@wpf<br/>
            college of informatics, huazhong agricultural
university
        </div>
    </div>
</body>
</html>

#html模板套用
{% extends "show.html" %}

{% block content %}
    {{content}}

```

```
{% endblock %}
```

```
#show.html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>{{ title }}</title>
  <style>
    *{
      margin:0 auto; /*div等元素居中方法*/
      padding:0;
      border:0;
    }
    #header{
      width:100%; height:100px;
      border-bottom:1px dashed gray;
      text-align: center; /*水平居中*/
      line-height: 80px;vertical-align: middle;
      font-size: 60px;
      padding-top:10px
    }
    #content{
      width:100%; height:730px;
      border:1px dashed gray;
      text-align: center; /*水平居中*/
      line-height: 50px;vertical-align: middle;
      color:blue;
      padding-top: 20px;
      font-size: 40px;
    }
    #footer{
      width:100%; height:60px;
      border-top:1px dashed gray;
      text-align: center; /*水平居中*/
      line-height: 25px;vertical-align: middle;
      font-size: 20px;
      padding-top: 20px;
      font-weight: bold;
    }
  </style>
  <script>
  </script>
</head>
<body>
  <div>
    <div id="header">
      {{ header }}
```



```

        </div>
        <div id="content">
            {% block content %}

            {% endblock %}
        </div>
        <div id="footer">
            technology support and all rights reserve @wpf<br/>
            colloge of informatics, huazhong agricultural university
        </div>
    </div>
</body>
</html>

```

(2). 反向路由

```

@app.route('/user/<uid>', methods = ['GET', 'POST'])
def query_uesr(uid):
    pass

@app.route('/query_user', methods = ['GET', 'POST'])
def query_user_uid():
    try:
        uid = request.args.get('uid') //获取uid属性的值
        uname = request.args.get('uname')
        if int(uid):
            title1 = "query_user"
            header1 = "welcome to query_user page"
            content1 = "Hello wpf, this is the query user page"
            if uname:
                content2 = "query user: id=" + uid + "," + "name = " + uname
            else:
                content2 = "query user: id=" + uid
            return render_template("query_user.html", title = title1, content1 =
content1, content2 = content2)
        else:
            abort(500) //产生错误
    except:
        abort(500)

```

(3). 错误路由

```

@app.errorhandler(500)// 错误路由
def not_found2(e):
    title1="error:500"
    header1="Welcome to error:500 page"

```

```
        content1="This page has been aborted!"
        return render_template('500.html',title=title1, header=header1,
                               content=content1)
```

(4). 获取地址

```
@app.route('/login')
def LOGIN():
    pass

print("url: " + url_for(LOGIN))
# url: /login
```