EDU

CSDN学院　IT实战派

# 第一章 Python基础

## 第17节 图形用户界面实战

讲师：张 涛

# 本节目标



1. 丰富的图形界面平台
2. wxPython的安装和运行
3. GUI应用案例开发
    - 3.1 开始
    - 3.2 窗口和组件
    - 3.3 标签、标题和位置
    - 3.4 更智能的布局
    - 3.5 事件处理
    - 3.6 完成

# 1. 丰富的图形界面平台



- 目前支持Python的所谓”GUI工具包”有很多，但没有一个被认为是标准的GUI工具包。这样也好（自由选择空间大），本节介绍最成熟的跨平台—wxPython
- GUI工具包：

| | | |
|---|---|---|
| Tkinter | 使用Tk平台。很容易得到。半标准 | http://wiki.python.org/moin/TkInter |
| wxpython | 基于wxWindows。跨平台越来越流行 | http://wxpython.org |
| PythonWin | 只能在Windows上使用。 | http://starship.python.net/crew/mhammond |
| Java Swing | 只能用于Python。使用本机的Java GUI | http://java.sun.com/docs/books/tutorial/uiswing |
| PyGTK | 使用GTK平台，在Linux上很流行 | http://pygtk.org |
| PyQt | 使用Qt平台，跨平台 | http://wiki.python.org/moin/PyQt |

# 2. wxPython的安装和运行



- 安装：wxpython

  pip install -U wxpython

  --Installing collected packages: six, wxpython

  --Successfully installed six-1.11.0 wxpython-4.0.1

- wxPython的执行过程：

  1 导入wx模块

  2 定义一个应用程序对象

  3 创建wx.Frame主窗口对象，设置窗口标题和大小

  4 创建组件、布局、添加事件处理等操作

  5 通过Frame的show()方法显示窗体

  6 进入应用程序事件主循环

# 3. GUI应用案例开发



3.1 开始

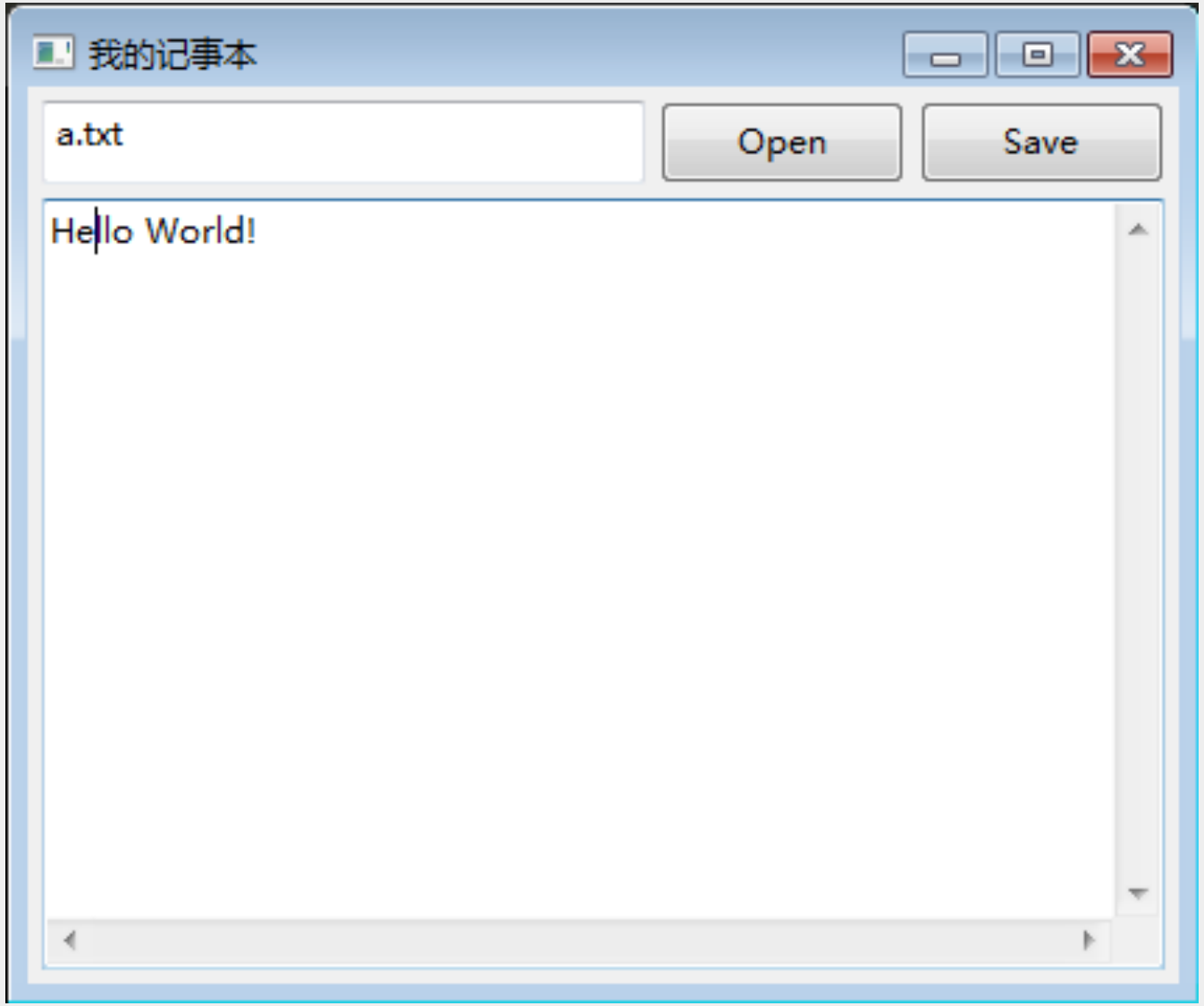
3.2 窗口和组件

3.3 标签、标题和位置

3.4 更智能的布局

3.5 事件处理

3.6 完成



# 3.1 开始



```
# 导入wxPython模块
import wx

# 创建应用程序对象
app = wx.App()

#进入应用程序事件主循环
app.MainLoop()
```

```
# 导入wxPython模块
import wx
# 创建应用程序对象
app = wx.App()
win = wx.Frame(None)    #创建一个单独的窗口
btn=wx.Button(win)      #创建一个按钮组件
win.Show()              #设置可见
#进入应用程序事件主循环
app.MainLoop()
```

# 3.3 标签、标题和位置



```
import wx
app = wx.App()
win = wx.Frame(None,title="我的记事本",size=(410,335)) #创建一个单独的窗口
#参数： label：标签字； pos：定位 ； size：大小
loadButton = wx.Button(win,label="Open",pos=(225,5),size=(80,25))

saveButton = wx.Button(win,label="Save",pos=(315,5),size=(80,25))

filename = wx.TextCtrl(win,pos=(5,5),size=(210,25))
# style表示样式： TE_MULTILINE表示多行文本；wx.HSCROLL表示水平滚动条
contents = wx.TextCtrl(win,pos=(5,35),size=(390,260),style=wx.TE_MULTILINE | wx.HSCROLL)

win.Show()
app.MainLoop() #进入应用程序事件主循环
```

# 3.4 更智能的布局



```
bkg = wx.Panel(win) #创建一个面板
#创建组件
loadButton = wx.Button(bkg,label="Open")
saveButton = wx.Button(bkg,label="Save")
filename = wx.TextCtrl(bkg)
contents = wx.TextCtrl(bkg,style=wx.TE_MULTILINE | wx.HSCROLL)
#布局容器
hbox=wx.BoxSizer() #默认水平布局
hbox.Add(filename,proportion=1,flag=wx.EXPAND)
hbox.Add(loadButton,proportion=0,flag=wx.LEFT,border=5)
hbox.Add(saveButton,proportion=0,flag=wx.LEFT,border=5)
#布局容器
vbox=wx.BoxSizer(wx.VERTICAL) #垂直布局
vbox.Add(hbox,proportion=0,flag=wx.EXPAND|wx.ALL,border=5)
vbox.Add(contents,proportion=1,flag=wx.EXPAND|wx.LEFT|wx.BOTTOM|wx.RIGHT,border=50)
bkg.SetSizer(vbox) 布局容器放入到面板中
```

# 3.5 事件处理

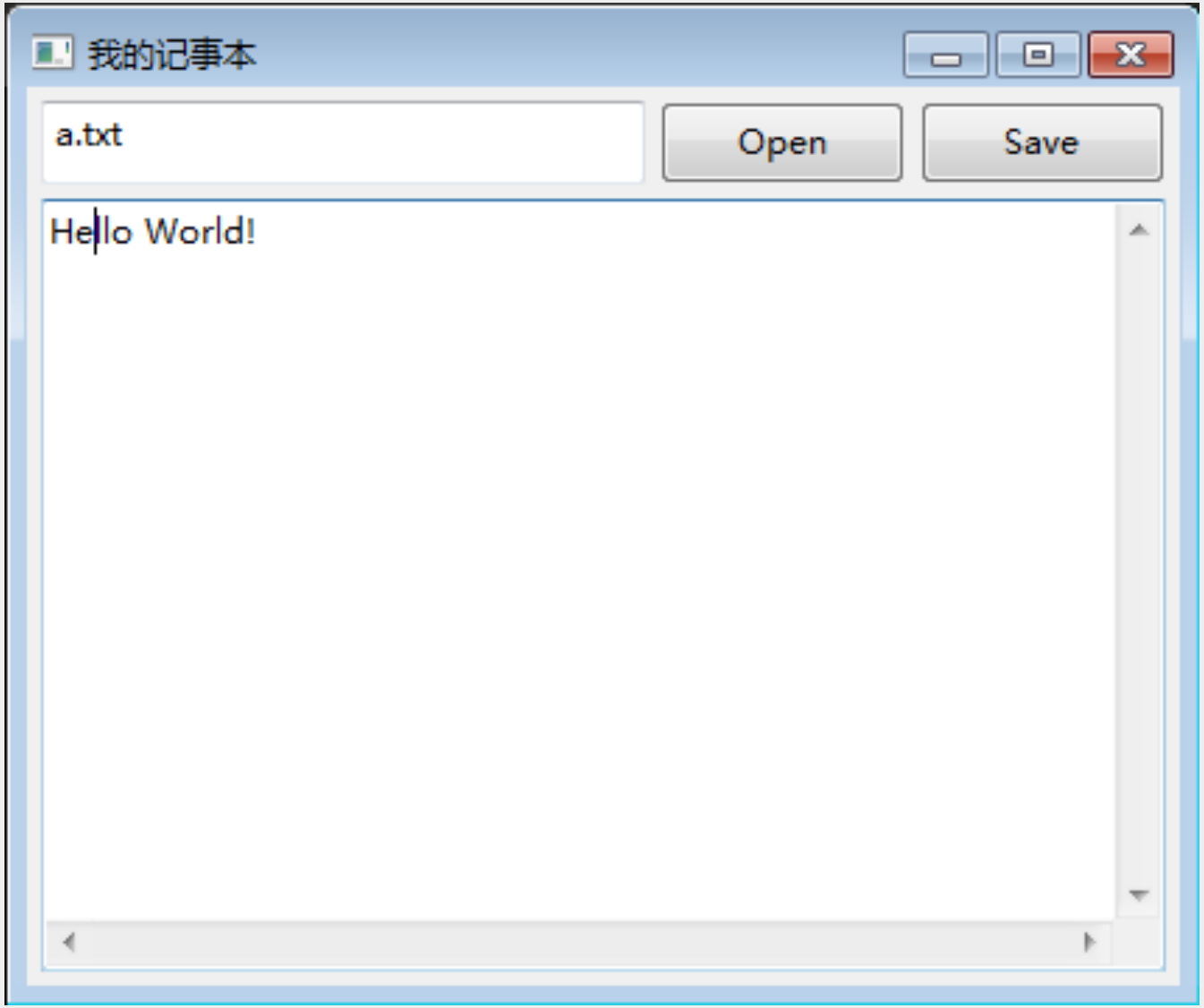


```
#按钮事件处理函数
def load(event):
    '''加载文件内容'''
    file=open(filename.GetValue(),"r")
    contents.SetValue(file.read())
    file.close()

def save(event):
    '''保持文件内容'''
    file=open(filename.GetValue(),"w")
    file.write(contents.GetValue())
    file.close()

# 绑定按钮事件处理按时
loadButton.Bind(wx.EVT_BUTTON,load)
saveButton.Bind(wx.EVT_BUTTON,save)
```

# 3.6 完成



# 小结



1. 丰富的图形界面平台
2. wxPython的安装和运行
3. GUI应用案例开发
    - 3.1 开始
    - 3.2 窗口和组件
    - 3.3 标签、标题和位置
    - 3.4 更智能的布局
    - 3.5 事件处理
    - 3.6 完成

# 布置作业

- 跟着老师自己动手练一练
- 自己动手将界面布局改为更智能的布局


EDU
CSDN学院 IT实战派