

2 Python GUI 概述

本章的主要内容是介绍 Python 的 3 种 GUI 框架：tkinter，wxPython 和 PyQt。

2.1 wxPython

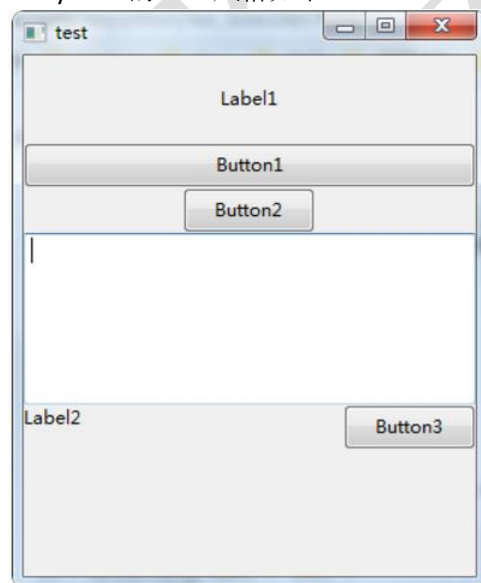
wxPython 是一套基于 wxWidget 的跨平台 GUI 框架。它支持 Linux、Windows 和 Mac OS。wxPython 是基于 wxWidget 的 Python 封装，而 wxWidget 是用 C++ 开发的。wxPython 的一个优点是开源的，开发者可以免费使用这套图形开发工具包。wxPython 的使用非常简单，上手也很容易。最大的缺点可能就是需要安装软件包才能在 Python 中使用。如果有 C++ 和 wxWidget 的经验，在 Python 中使用 wxPython 还是一个很不错的选择。

手动用代码创建好看的 GUI 是几乎不可能完成的任务，因此选择一个可视化 GUI 工具是非常明智的选择。支持 wxPython 的 GUI 工具有：

- wxFormBuilder
- wxDesigner
- wxGlade
- BoaConstructor
- gui2py

关于这些 GUI 设计工具的使用，就不在本书中介绍了。读者可以自行查阅相关的文档。

wxPython 的 GUI 风格如下：



2.2 PyQt

PyQt 是基于 Qt 的 Python 模块集，它有 300 多个类，将近 6000 个函数和方法。它是一个跨平台的工具包，可以运行在所有的主流操作系统上，包括 Linux、Windows 和 Mac OS。PyQt 采用 commercial 和 GPL 的双重许可。PyQt 是 Python GUI 工具包中功能最强大的一个。

PyQt3 支持 Qt1 到 Qt3，PyQt4 支持 Qt4，PyQt5 支持 Qt5。Qt 适合于大型应用，由它自带的 qt designer 可以让我们轻松来构建界面元素。因此 PyQt 不太适合简单的 GUI 应用。

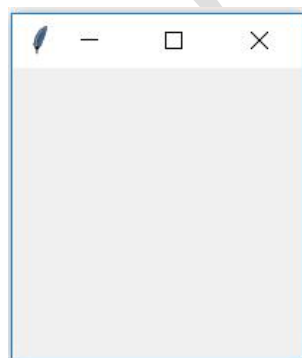
PyQt 的 GUI 风格如下：



2.3 Tkinter

Tkinter 是 Python 的自带 GUI 库。基于 TCL/TK 实现。Tkinter 是集成到 Python 安装包中的，因此只要安装好 Python，在 import tkinter 模块之后，就可以使用 GUI 了。对于简单的图形界面，使用 tkinter 是完全可以胜任的。Tkinter 的优点是内置 Python 中的，缺点是运行速度较慢。

Tkinter 的窗口风格



下表是三种 GUI 工具包的比较：

GUI	简介	内置	操作系统	版本	优点	缺点
Tkinter	Python 的标准 Tk GUI 工具包的接口	是	Linux Windows Mac OS	Python2.x Python3.x	标准库创建的 GUI 简单，学习成本低	功能比较弱，只适合开发简单的界面 性能不好 执行速度慢
wxPython	wxWidgets(C++ 编写)的包装	否	Linux Windows Mac OS	Python2.x	开源跨平台	不支持 Python3.x 需要学习 wx 组件
PyQt	PyQt 提供 GPL 版和商业版	否	Linux Windows Mac OS	Python2.x (PyQt4) Python3.x (PyQt5)	可使用大多数为 QT 开发的组件 文档丰富 辅助工具丰富	注意避免内存泄漏 运行时庞大 需要有一点 C++ 知识

2.4 Tkinter 简介

2.4.1 Tkinter 标准控件

Tkinter 提供了多种的标准控件(Widget)，比如按钮、输入框等。下表是 Tkinter 控件的列表。后面的章节会详细介绍各个控件的功能以及属性、方法等。

Tk 提供如下控件：

控件	描述
Button	按钮控件：按钮一般与一个回调函数绑定，完成指定的功能
Canvas	画布控件：显示图形元素如线条或文本
Checkbutton	多选按钮控件：有 on-off 两种状态，一般用于选中或者未选中。
Entry	输入控件：用于输入或者显示文本内容
Frame	框架控件：在屏幕上显示一个矩形区域，多用来作为容器
Label	标签控件：可以显示文本和位图。多用来显示提示信息
LabelFrame	标签框架控件：带有标签的框架，可以显示标题。
Listbox	列表框控件：可以显示显示一组字符串（选项），用户可以选择一个或者多个选项。

Menubutton	菜单按钮控件：用于在菜单上显示按钮风格的菜单条。
Menu	菜单控件：显示菜单栏,下拉菜单和弹出菜单
Message	消息控件：用来显示多行文本，是标签控件的升级版。不过到底会怎么显示你输入的文本，好像要看 Message 控件的心情。开个玩笑，和你设定的 width 参数有关。如果没有设定，怎么显示就是系统计算的结果了。
OptionMenu	可选菜单控件：类似多选框控件，创建一个弹出菜单，可以选择相应的选项，选择完毕后，选中的文字会显示按钮旁边
PanedWindow	PanedWindow 是一个窗口布局管理的控件，可以包含一个或者多个子控件（Pane）。
Radiobutton	单选按钮控件：显示一个单选的按钮状态，实现多选一的功能
Scale	范围控件或刻度控件：显示一个数值刻度，用户拖动滑块选择数值。可以设定最大值与最小值。
Scrollbar	滚动条控件：当内容超过可视化区域时使用，如列表框、文本框等。。
Spinbox	数值调整控件：与 Entry 类似，但是可以指定数值的输入范围
Text	文本控件：用于显示多行文本
Toplevel	容器控件：用来提供一个单独的对话框，和 Frame 比较类似

2.4.2 ttk

ttk 中扩展了下拉选择框((Combobox)、树形列表(Treeview)、属性页标签(Notebook)、进度条(Progressbar)等。这些控件扩展了 tkinter 模块，功能更为完善和丰富。

2.4.3 Tkinter 标准对话框

对话框是比较常用的人机交互窗口。Tkinter 中也提供很多种定制好的对话框，开发者可以直接调用，省去了自己开发的麻烦。

Tkinter 内置的对话框有：

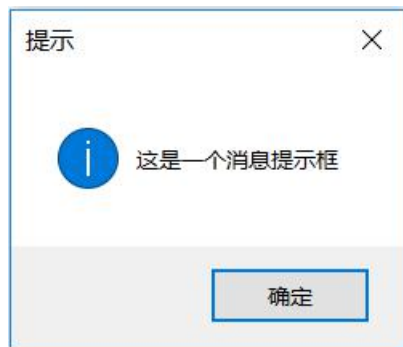
2.4.3.1 消息框

消息框有很多种，比如提示框、警告框等。具体的看下面的说明。下面的对话框都需要加上这 2 条程序：

```
import tkinter
import tkinter.messagebox
```

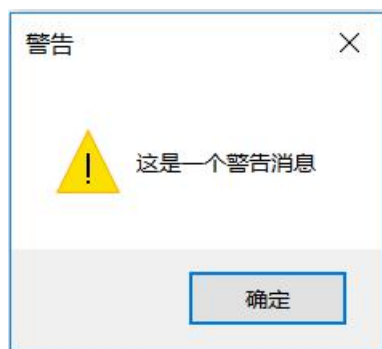
2.4.3.1.1 提示消息框

代码: `tkinter.messagebox.showinfo('提示','我是一个提示消息框')`



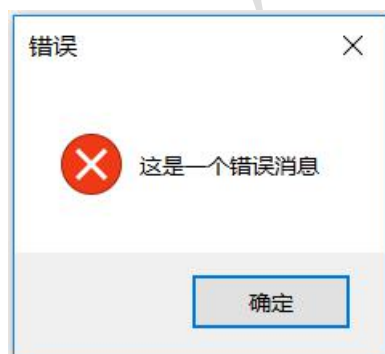
2.4.3.1.2 警告消息对话框

代码: `tkinter.messagebox.showwarning('警告','这是一个警告消息')`



2.4.3.1.3 错误消息对话框

代码: `tkinter.messagebox.showerror('错误','这是一个错误消息')`

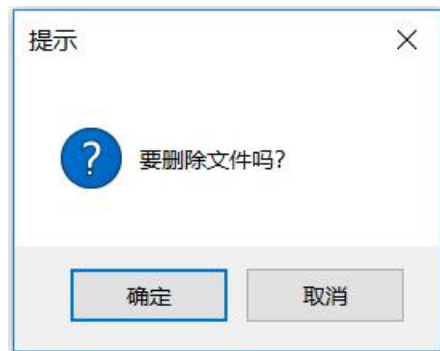


注: 如果系统支持, 会有报警的声音。

2.4.3.1.4 确认取消对话框

代码: `ok=tkinter.messagebox.askokcancel('提示','要删除文件吗?')`

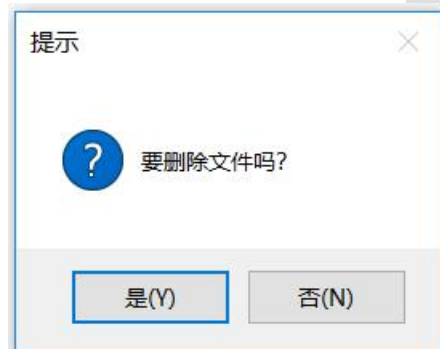
返回值是 True 或者 False



2.4.3.1.5 询问对话框

代码: `a=tkinter.messagebox.askquestion('提示','要删除文件吗?')`

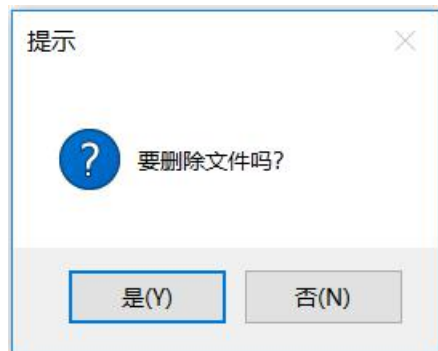
返回值是 yes 或 no



2.4.3.1.6 是否对话框

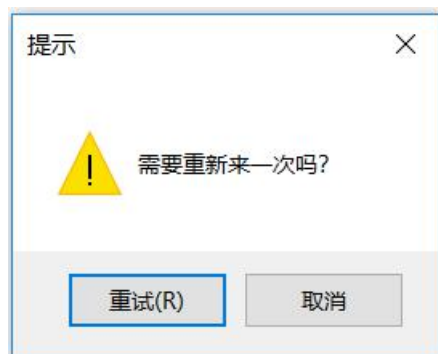
代码: `a=tkinter.messagebox.askyesno('提示','要删除文件吗?')`

返回值是 True 或 False



2.4.3.1.7 重试对话框

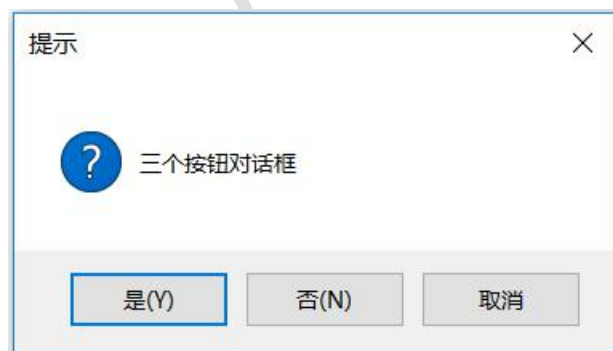
代码: `a=tkinter.messagebox.askretrycancel('提示','需要重新来一次吗?')`
返回值是 True 或者 False



注: 如果系统支持, 会有报警的声音。

2.4.3.1.8 三按钮对话框

代码: `a=tkinter.messagebox.askyesnocancel('提示','三个按钮对话框')`
返回值: True、False 和 None



2.4.3.2 SimpleDialog

使用 SimpleDialog 可以比较轻松的实现自定义对话框、输入整数、浮点数和字符串的功能。
下面的代码都需要加上下面两行代码：

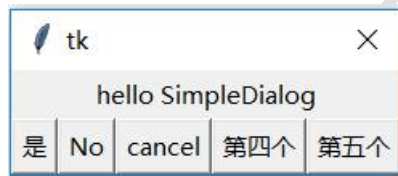
```
import tkinter
from tkinter import simpledialog
```

2.4.3.2.1 自定义对话框

代码：

```
root=tkinter.Tk()
dlg= tkinter.simpledialog.SimpleDialog(root,
                                     text = 'hello SimpleDialog',
                                     buttons = ['是','No','cancel','第四个','第五个']
                                     )
print(dlg.go())
root.mainloop()
```

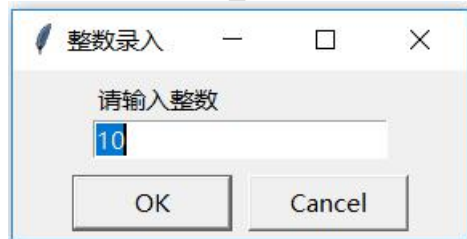
使用 SimpleDialog 可以轻松的实现按钮自定义。自定义的按钮放在 **buttons** 这个列表中。返回值从 0 开始，按照位置递增。
显示的文本通过字符串 **text** 传递。



SimpleDialog 中有一个小的 bug。需要修改 `t.quit` 为 `root.destroy` 去退出。

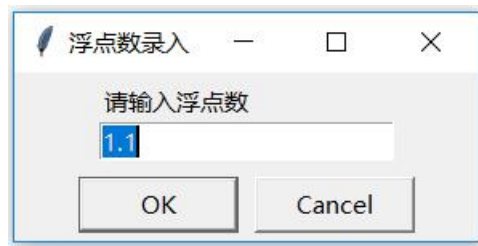
2.4.3.2.2 输入整形 askinteger

代码： `r = simpledialog.askinteger('整数录入','请输入整数', initialvalue=10)`



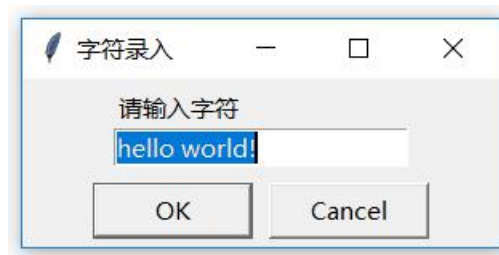
2.4.3.2.3 输入浮点数 askfloat

代码： `r = simpledialog.askfloat('浮点数录入','请输入浮点数', initialvalue=1.1)`



2.4.3.2.5 输入浮点数 askstring

代码: `r = simpledialog.askstring('字符录入', '请输入字符', initialvalue='hello world!')`



2.4.3.2.6 文件对话框

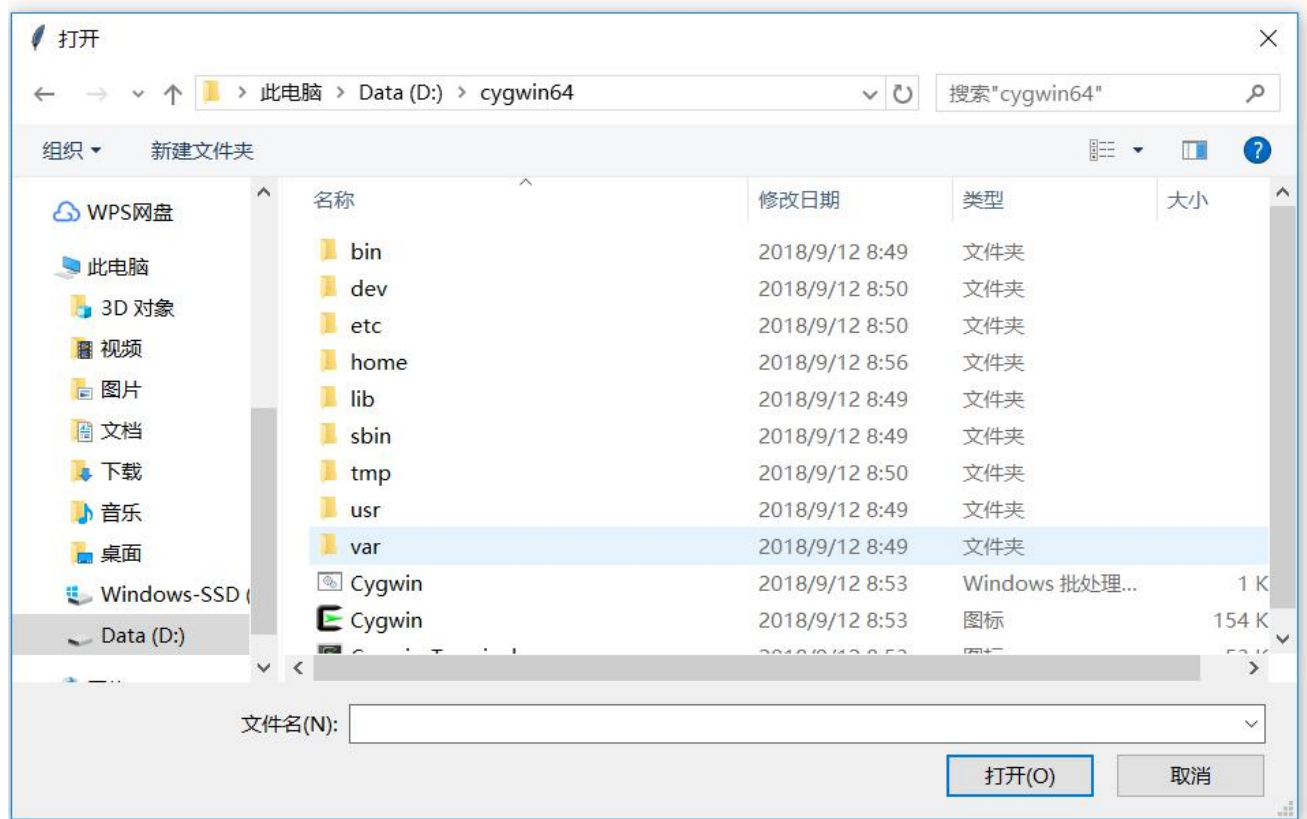
文件对话框的功能有三种:

1. 打开一个文件: `askopenfilename()`
2. 打开一组文件: `askopenfilenames()`
3. 保存文件: `asksaveasfilename()`

使用文件对话框需要引入 `tkinter.filedialog`。下面是详细的介绍:

1. 打开一个文件

代码: `filename = tkinter.filedialog.askopenfilename()`

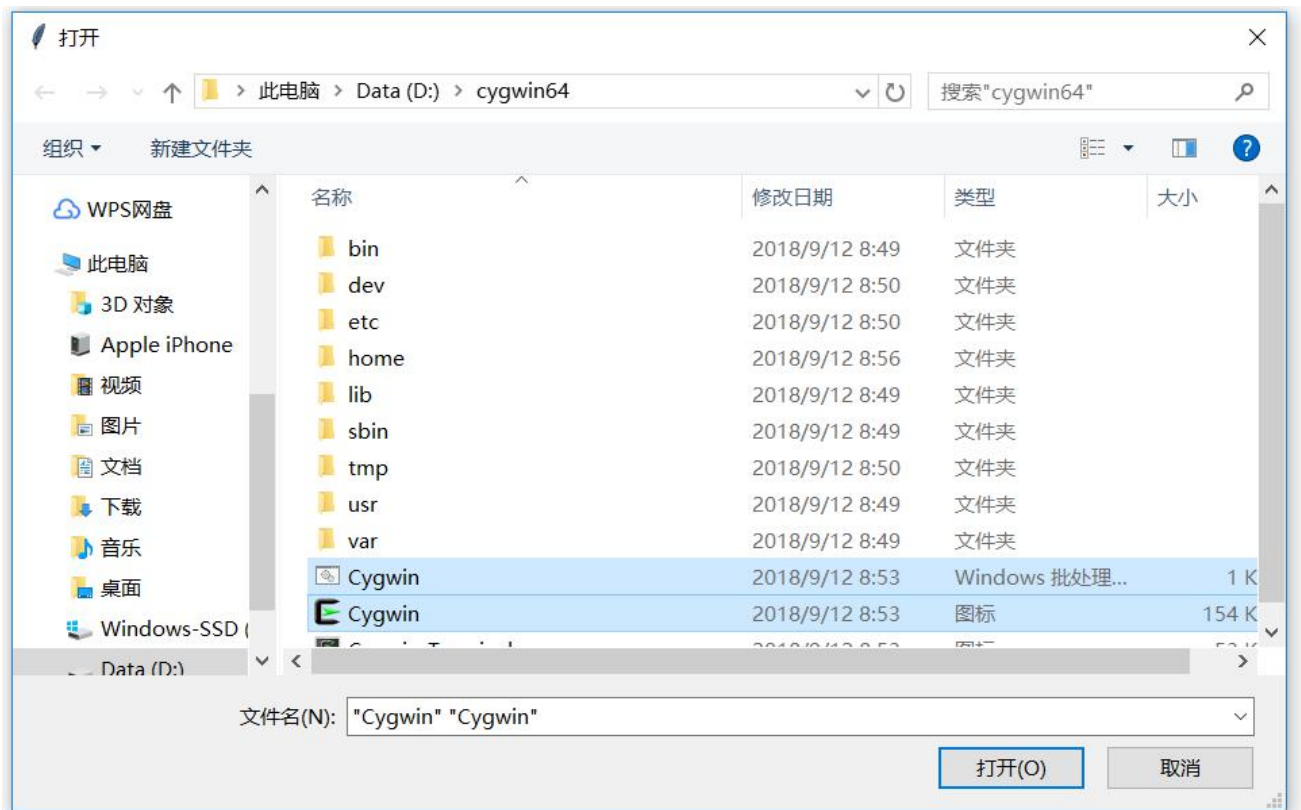


2. 打开一组文件

就是可以打开多个文件。

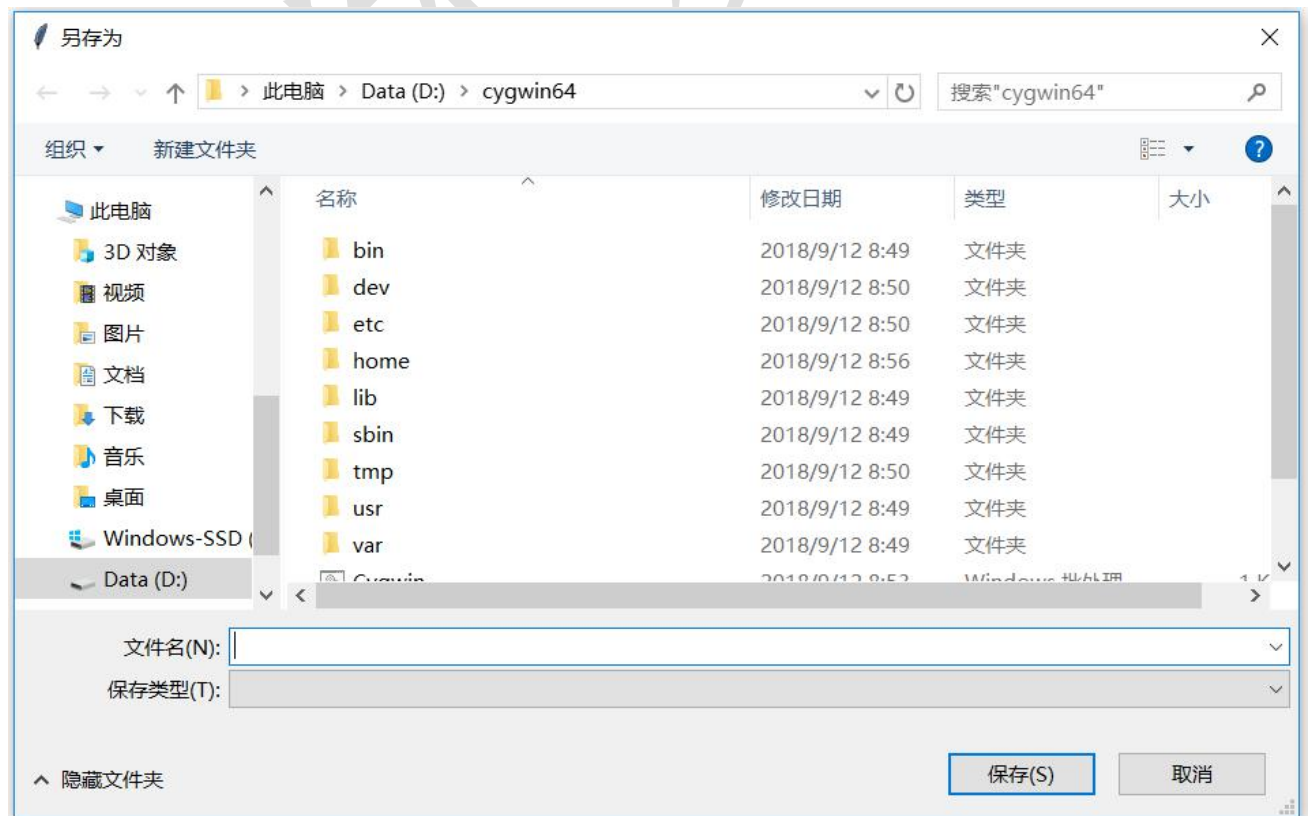
代码：`filenames = tkinter.filedialog.askopenfilenames()`

和打开一个文件非常的类似，只不过打开多个文件可以同时选择多个文件。返回的是一个元组，包含一组文件名。



3. 保存文件

代码: `filename = tkinter.filedialog.asksaveasfilename()`



2.4.3.3 颜色选择对话框

颜色选择对话框需要引入的模块是：

```
import tkinter.colorchooser
```

代码：color = tkinter.colorchooser.askcolor()

