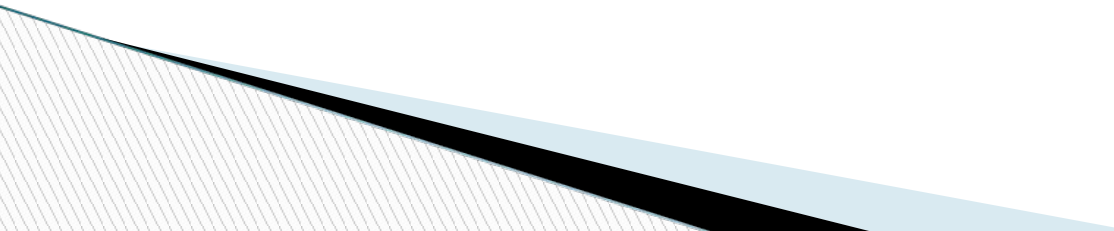


Диалоговые окна



Диалоговые окна, как элементы графического интерфейса, предназначены для вывода сообщений пользователю, получения от него какой-либо информации, а также управления.

Чтобы запрограммировать с помощью Tkinter вызов диалоговых окон открытия и сохранения файлов и работу с ними потребуется дополнительно импортировать "**подмодуль**" **Tkinter - tkinter.filedialog**, в котором описаны классы для окон данного типа.



```
from tkinter import *  
from tkinter.filedialog import *  
root = Tk()  
op = askopenfilename()  
sa = asksaveasfilename()  
root.mainloop()
```

Здесь создаются два объекта (op и sa): один вызывает диалоговое окно "Открыть", а другой "Сохранить как...". При выполнении скрипта, они друг за другом выводятся на экран после появления главного окна.

Если не создать root, то оно все-равно появится на экране, однако при попытке его закрытия в конце возникнет ошибка.

- Метод `input` модуля `fileinput` может принимать в качестве аргумента адрес файла, читать его содержимое, формируя список строк. Далее с помощью цикла `for` можно извлекать строки последовательно и помещать их, например, в текстовое поле.

```
..... import fileinput  
  
.....  
for i in fileinput.input(op):  
    txt.insert(END,i)  
  
.....
```

```
from tkinter import *
from tkinter.filedialog import *
import fileinput

def _open():
    op = askopenfilename()
    for l in fileinput.input(op):
        txt.insert(END,l)

root = Tk()

m = Menu(root)
root.config(menu=m)

fm = Menu(m)
m.add_cascade(label="File",menu=fm)
fm.add_command(label="Open...",command=_open)

txt = Text(root,width=40,height=15,font="12")
txt.pack()

root.mainloop()
```

Теперь попробуем сохранять текст, набранный в текстовом поле. Добавим в код пункт меню и следующую функцию:

```
def _save():  
    sa = asksaveasfilename()  
    letter = txt.get(1.0,END)  
    f = open(sa,"w")  
    f.write(letter)  
    f.close()
```

В переменной `sa` храниться адрес файла, куда будет производиться запись. В переменной `letter` – текст, "полученный" из текстового поля. Затем файл открывается для записи, в него записывается содержимое переменной `letter`, и файл закрывается (на всякий случай).

Еще одна группа диалоговых окон описана в модуле `tkinter.messagebox`. Это достаточно простые диалоговые окна для вывода сообщений, предупреждений, получения от пользователя ответа "да" или "нет" и т. п.

Дополним нашу программу пунктом Exit в подменю File и пунктом About program в подменю Help.

```
from tkinter.messagebox import *
...
def close_win():
    if askyesno("Exit", "Do you want to quit?"):
        root.destroy()

def about():
    showinfo("Editor", "This is text editor.\n(test version)")
...
fm.add_command(label="Exit", command=close_win)
....
hm = Menu(m)
m.add_cascade(label="Help", menu=hm)
hm.add_command(label="About", command=about)
...
```

Практическая работа

1. Напишите программу, описанную в уроке.
2. Измените программу: пусть после нажатия пункта Exit пользователю выводилось не окно с вопросом "выйти или нет", а окно с вопросом "сохранить или нет". В случае положительного ответа должна вызываться функция `_save` и только затем завершаться приложение.
3. Если в текстовом поле что-то содержится, то при открытии файла оно не удаляется, а содержимое файла просто дописывается. Исправьте этот недостаток (перед открытием файла содержимое текстового поля должно удаляться).