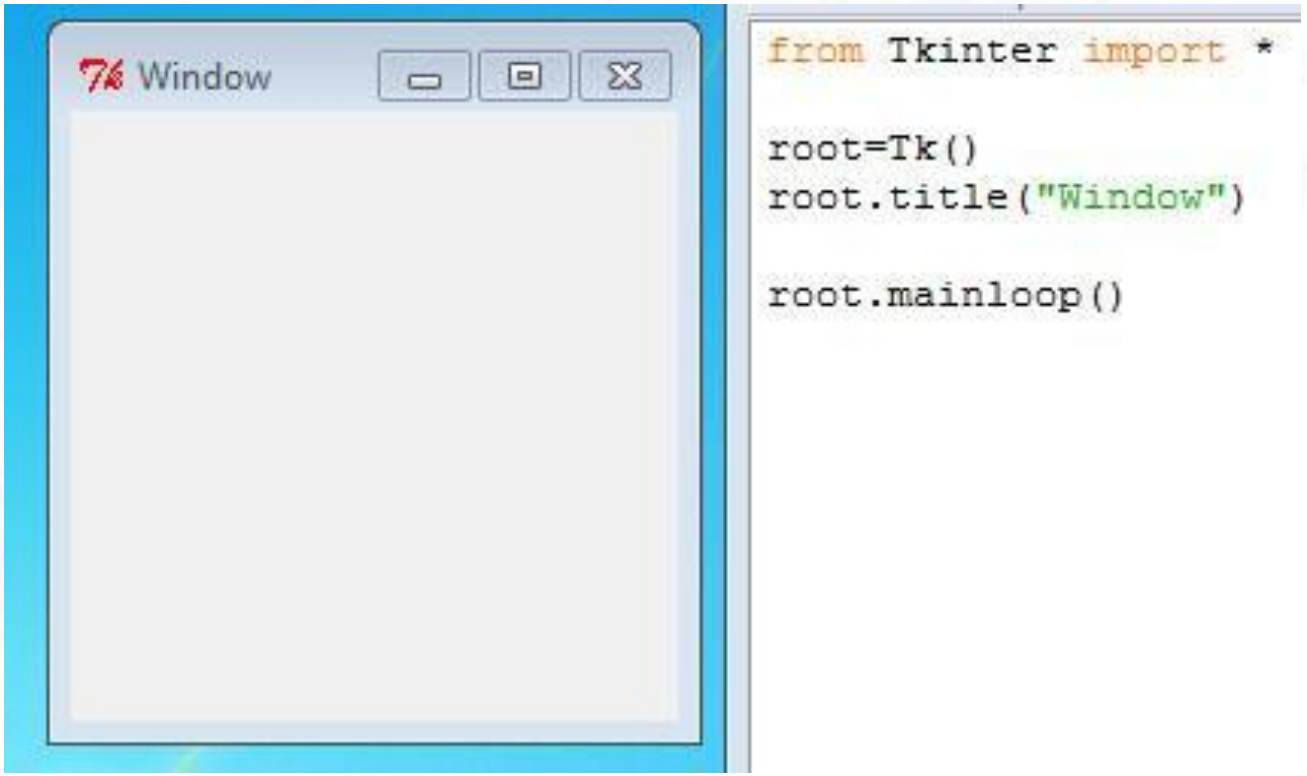
Графические интерфейсы

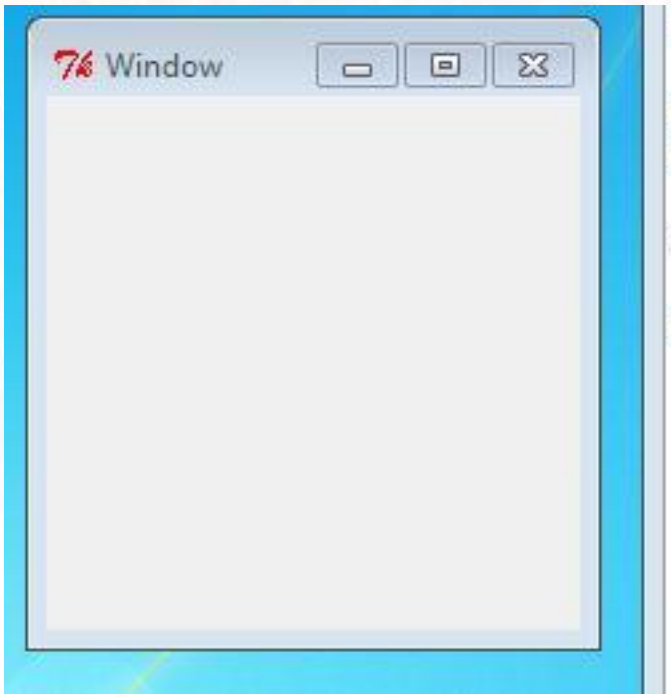
пакет Tkinter

- import Tkinter
- from Tkinter import *



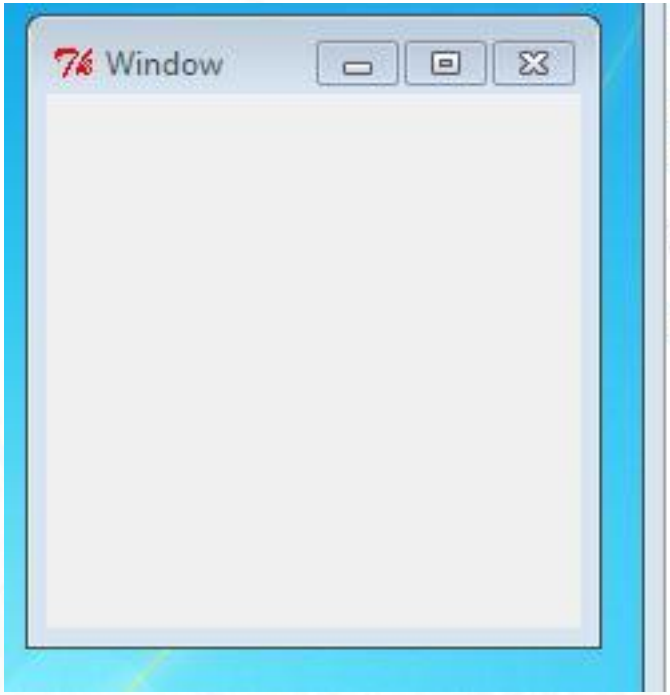
Widgets

- label
- button
- entry
- text
- menu
- canvas
- widget=Widgetclass(master, option=value, ...)




```
from Tkinter import *

root=Tk()
root.title("Window")

l=Label(root,text="label")

root.mainloop()
```

Widgets




```
from Tkinter import *

root=Tk()
root.title("Window")

l=Label(root,text="label")

root.mainloop()
```

```
from Tkinter import *

root=Tk()
root.title("Window")

l=Label(root,text="label")
l.pack()

root.mainloop()
```

Основные методы виджетов

- config(option=value, ...) - изменить опции
- config() → dict – информация об опциях
- cget(option) – возвращает текущее значение опции
- bind(event, handler) – привязать определенное действие к определенному событию виджета
- grid(options), pack(options) – менеджеры расположения

Опции виджетов

- Общие
  - Размер
  - Фон
  - Цвет и стиль текста
  - Наличие и тип границы
  - Тип курсора
  - enabled/disabled

  и т.д.

- Специфические

Label

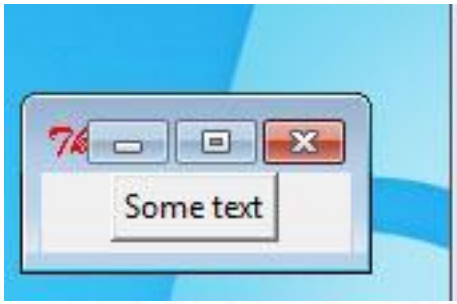
- Текст / картинка
- label = Label(master, text=”Text”)

- Опции:
  - text
  - image
  - width, height
  - textvariable (типа StringVar)

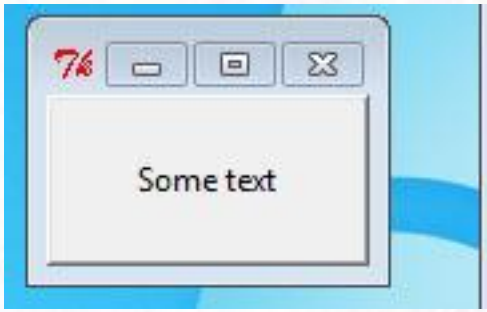
  и т.д.

Button

- b = Button(master, option=value, ...)




```
from Tkinter import *

root=Tk()
b=Button(root,text="Some text")
b.pack()
root.mainloop()
```

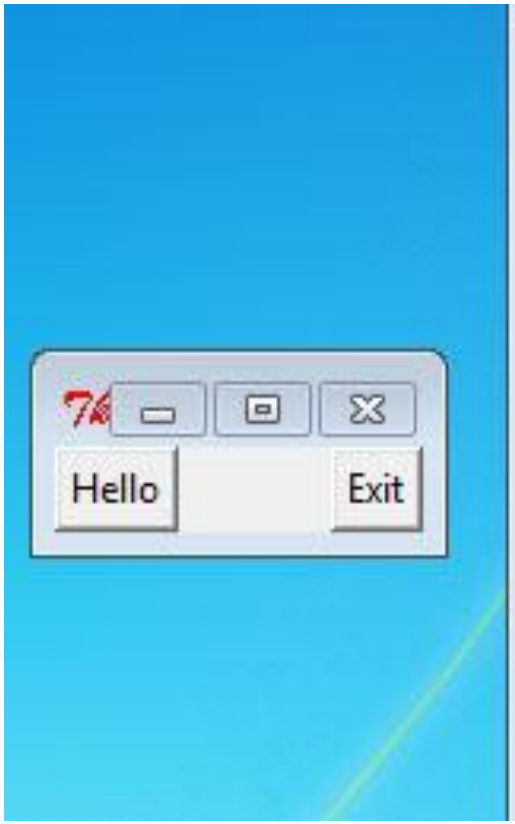



```
from Tkinter import *

root=Tk()
b=Button(root,text="Some text",padx=30,pady=20)
b.pack()
root.mainloop()
```

```
from Tkinter import *

root=Tk()
b=Button(root,text="Some text",
         width=5,heigh=3)
b.pack()
root.mainloop()
```

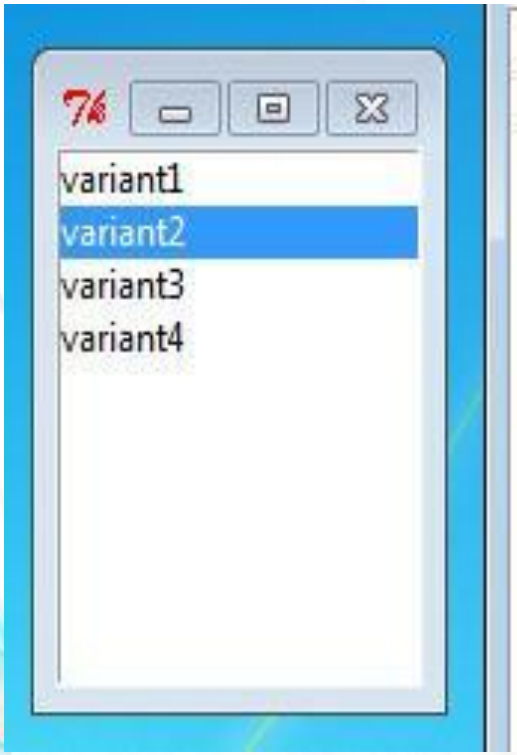
Button

- b = Button(master, option=value, ..., command=callback)




```
from Tkinter import *

def hello_label():
    l=Label(root,text="Hello,world!")
    l.pack(side=LEFT)

root=Tk()

b=Button(root,text="Hello",command=hello_label)
b.pack(side=LEFT)
b_exit=Button(root,text="Exit",command=root.quit)
b_exit.pack(side=RIGHT)


root.mainloop()
```

Listbox




```
from Tkinter import *

root=Tk()

l=Listbox()
l.insert(0,"variant1","variant2","variant3","variant4")
l.select_set(1)

l.pack()

root.mainloop()
```

insert(index,items)

delete(item) / delete(first,last)

get(index) → string

get(first,last) → list – список вариантов

select_set(index) / select_set(first,last)

select_includes(index) → boolean

curselection() → list of indexes – выбранные варианты

select_clear()

Frame

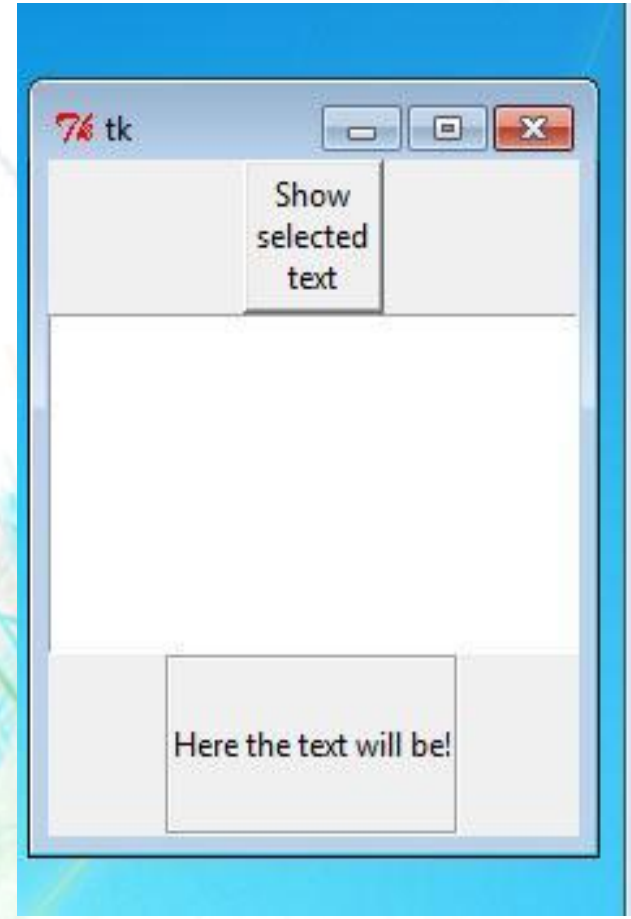
- Frame = Frame(master, option=value, ...)
- Помогает группировать другие виджеты

Entry

```
from Tkinter import *

root=Tk()

Label(root,text="This is an Entry:").pack()
e=Entry(root,width=25)
e.pack()

root.mainloop()
```

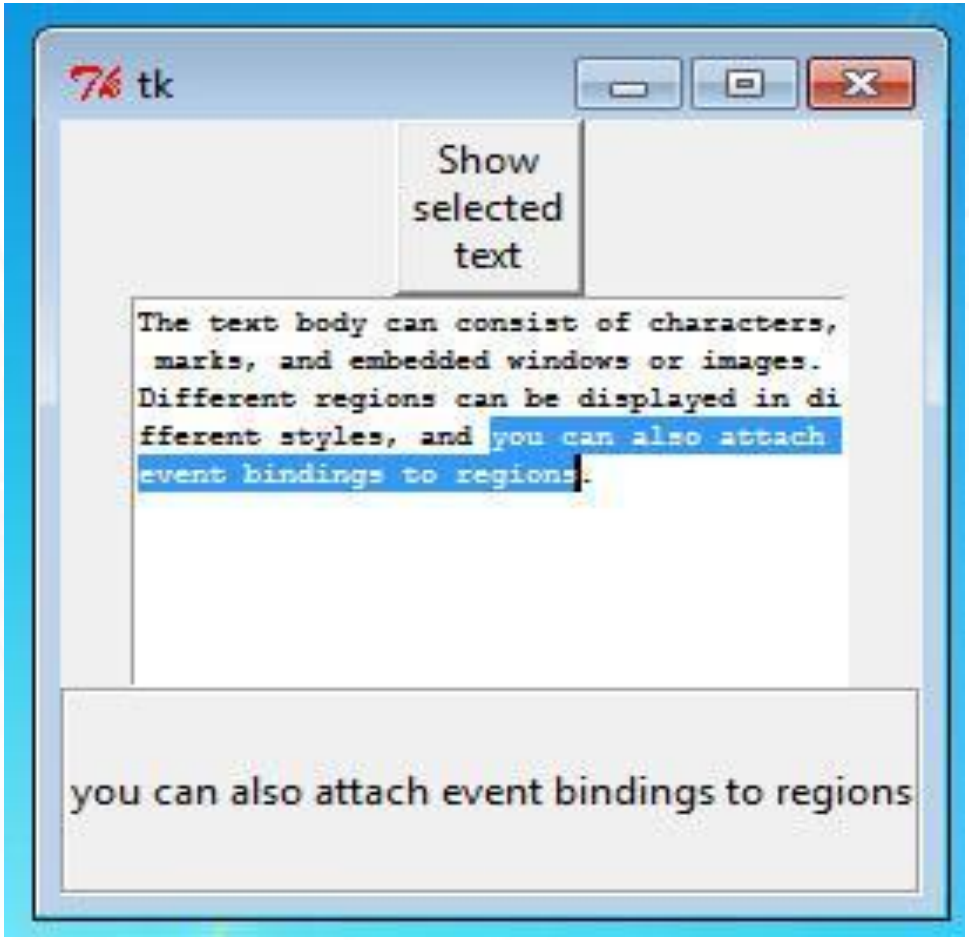
get() → string (получить содержимое поля)

Text




```
from Tkinter import *

def show_text():
    selected_text=t.get(SEL_FIRST,SEL_LAST)
    v.set(selected_text)

root=Tk()

b=Button(root,text="Show\nselected\ntext",command=show_text)
b.pack()

t=Text(root,height=10,width=40)
t.pack()

v=StringVar()
l=Label(root,textvariable=v,height=4,relief=GROOVE)
v.set("Here the text will be!")
l.pack()

root.mainloop()
```

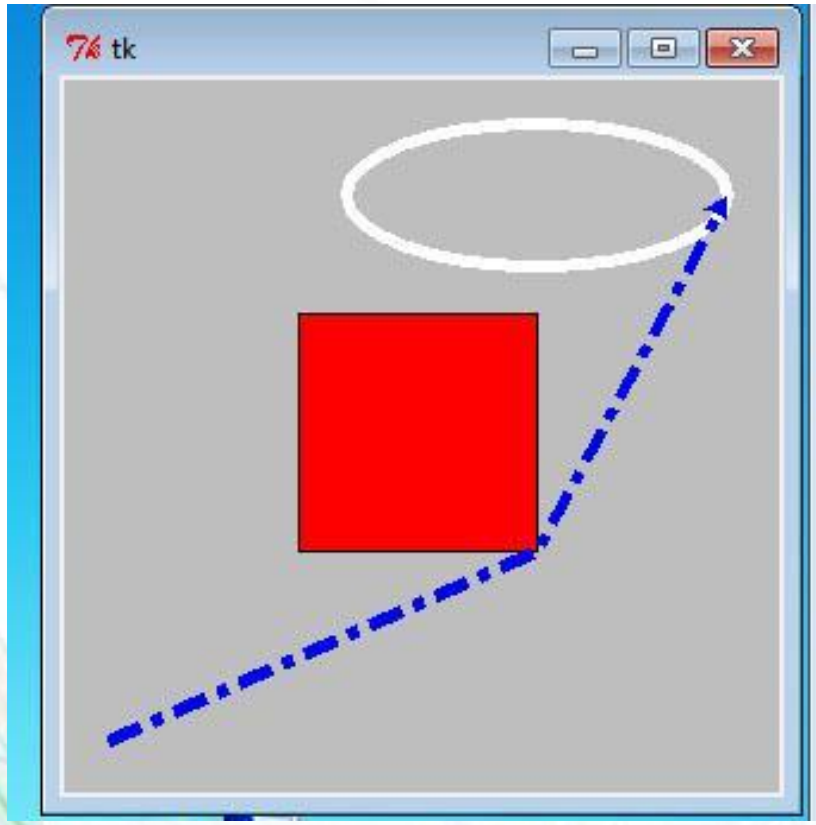
Text


tk
Show selected text
The text body can consist of characters, marks, and embedded windows or images. Different regions can be displayed in different styles, and you can also attach event bindings to regions.
you can also attach event bindings to regions

Canvas

- Рисунки, графики, диаграммы...
- Создаются элементы виджета, которые на нем размещаются
  - arc
  - bitmap
  - image
  - line
  - oval
  - polygon
  - rectangle
  - text
  - window – как контейнер для других виджетов, располагаемых поверх canvas

Canvas




```
File  Edit  Format  Run  Options  Windows  Help
from Tkinter import *

root=Tk()

canvas=Canvas(width=300,height=300,background="grey")
canvas.pack()

a=canvas.create_rectangle(100,100,200,200,fill="red")
b=canvas.create_oval(120,20,280,80,outline="white",width=5)
c=canvas.create_line(20,280,200,200,280,50,
                     fill="blue",
                     width=5,arrow="last",capstyle="round",
                     dash=[1,1,1,1])

root.mainloop()
```

create_название_элемента(options) → id

delete(items)

lift(item)

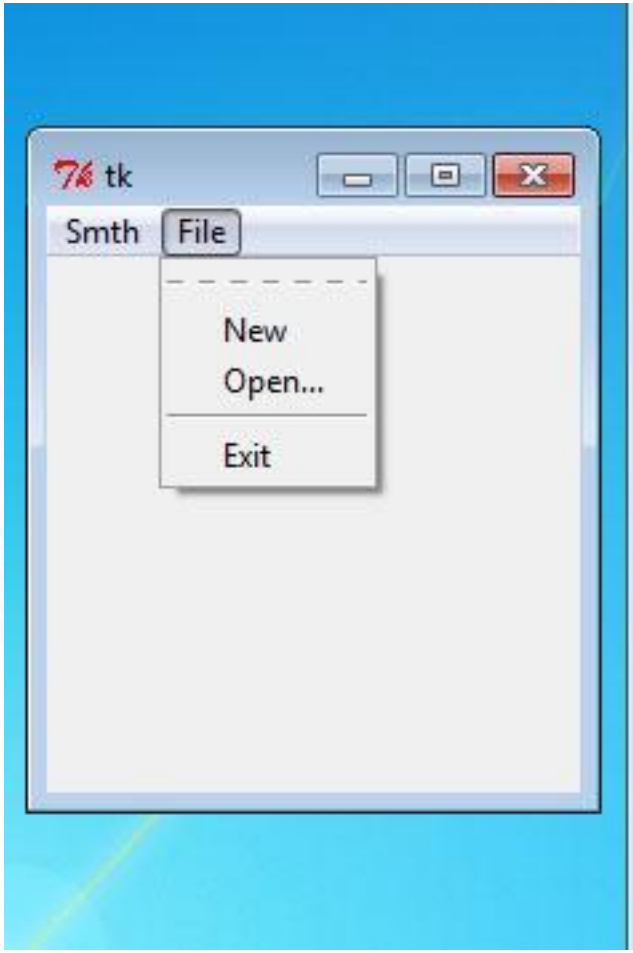
lower(item)

- itemcget(item, option) → string – значение опции
- itemconfig(item, options)

Menu

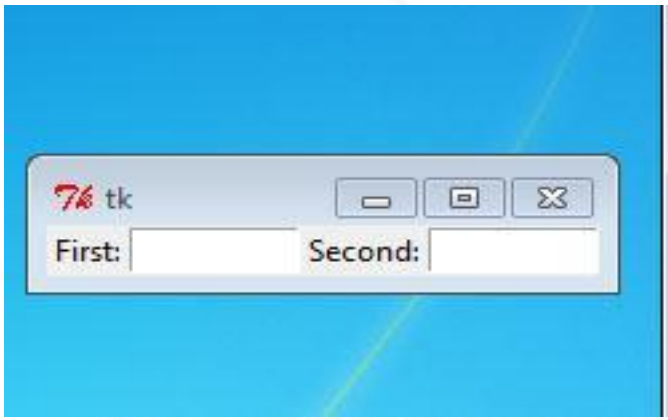
```
from Tkinter import *

def callback():
    print "called the callback!"

root = Tk()

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

menubar.add_command(label="Smth",command=callback)

root.mainloop()
```

Menu




```
from Tkinter import *

def callback():
    print "called the callback!"

root = Tk()

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

menubar.add_command(label="Smth",command=callback)

filemenu = Menu(menubar)

filemenu.add_command(label="New", command=callback)
filemenu.add_command(label="Open...", command=callback)
filemenu.add_separator()
filemenu.add_command(label="Exit", command=callback)

menubar.add_cascade(label="File", menu=filemenu)

root.mainloop()
```

Pack




```
from Tkinter import *

root=Tk()

Label(root,text="First:").pack(side=LEFT)
Entry(root,width=10).pack(side=LEFT)
Label(root,text="Second:").pack(side=LEFT)
Entry(root,width=10).pack(side=LEFT)

root.mainloop()
```

pack(option=value, …)

pack_configure(option=value, …)

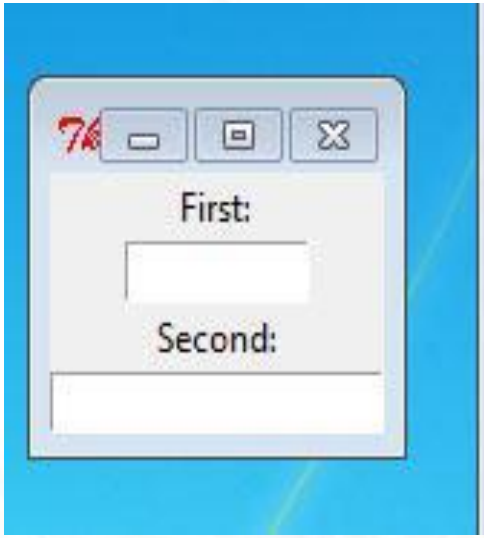
pack_forget()

pack_info() → dict – словарь с имеющимися опциями

side (TOP, BOTTOM, LEFT, RIGHT)

fill (X, Y, BOTH)

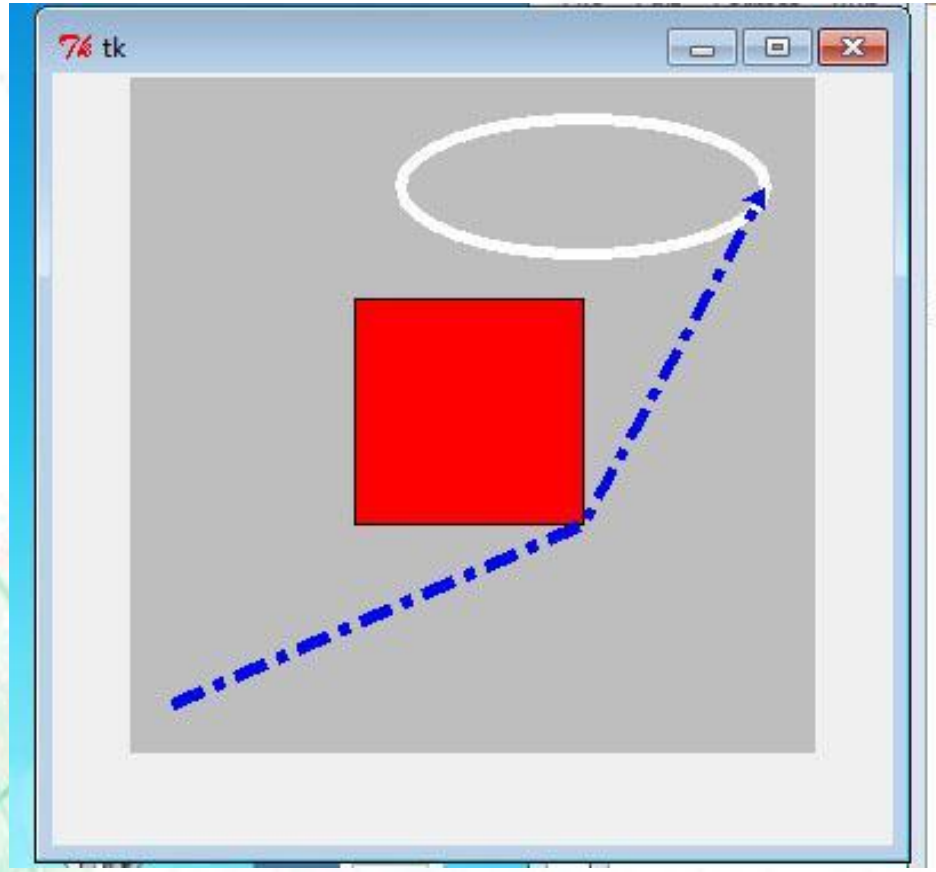
Pack




```
from Tkinter import *

root=Tk()

Label(root,text="First:").pack(side=TOP)
Entry(root,width=10).pack(side=TOP)
Label(root,text="Second:").pack(side=TOP)
Entry(root,width=10).pack(side=TOP,fill=X)

root.mainloop()
```

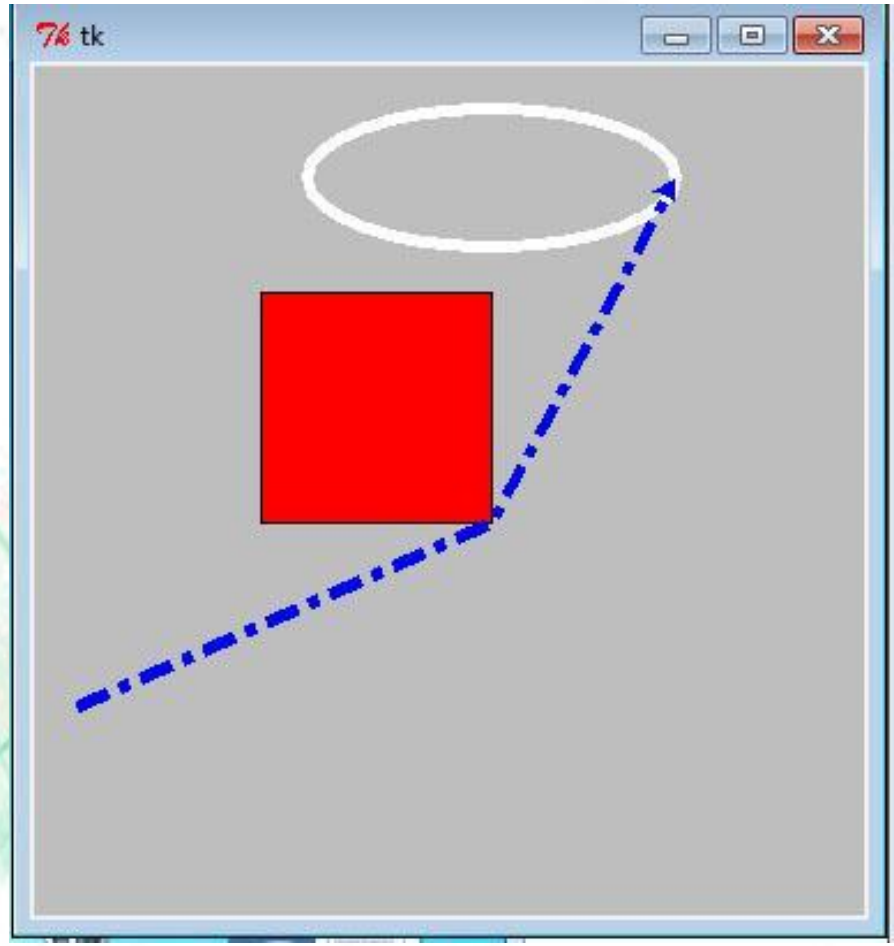
```
from Tkinter import *

root=Tk()

Label(root,text="First:").pack(side=TOP)
Entry(root,width=10).pack(side=LEFT)
Label(root,text="Second:").pack(side=TOP)
Entry(root,width=10).pack(side=LEFT)

root.mainloop()
```

Pack




```
from Tkinter import *

root=Tk()

canvas=Canvas(width=300,height=300,background="grey")
canvas.pack()

a=canvas.create_rectangle(100,100,200,200,fill="red")
b=canvas.create_oval(120,20,280,80,outline="white",width=5)
c=canvas.create_line(20,280,200,200,280,50,
                     fill="blue",
                     width=5,arrow="last",capstyle="round",
                     dash=[110,1,1,100])

root.mainloop()
```

Pack



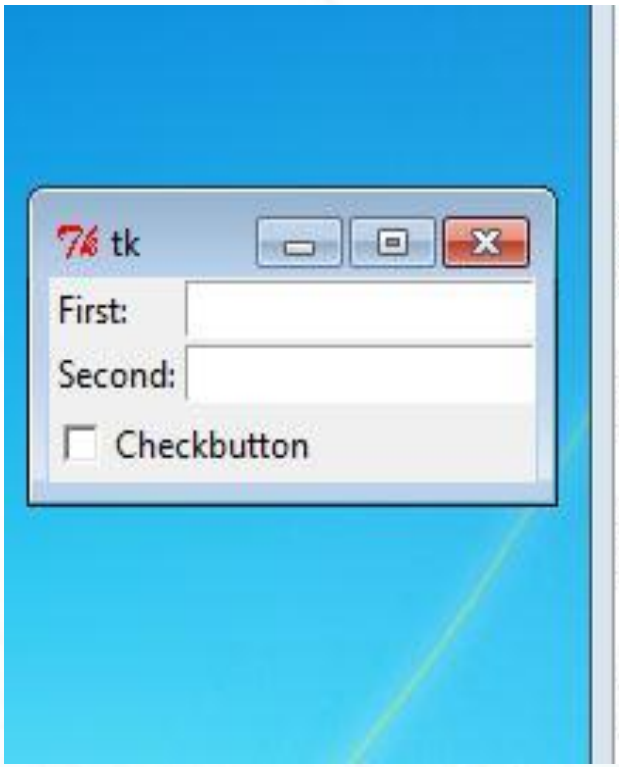
File Edit Format Run Options Windows Help

```
from Tkinter import *

root=Tk()

canvas=Canvas(width=300,height=300,background="grey")
canvas.pack(fill=BOTH,expand=True)

a=canvas.create_rectangle(100,100,200,200,fill="red")
b=canvas.create_oval(120,20,280,80,outline="white",width=5)
c=canvas.create_line(20,280,200,200,280,50,
                     fill="blue",
                     width=5,arrow="last",capstyle="round",
                     dash=[110,1,1,100])

root.mainloop()
```

Grid

- Создает таблицу
- Метод grid, вызванный у виджетов, задает расположение в таблице
  - widget.grid(column=1,row=1)
- Пустые строки и столбцы игнорируются
- По умолчанию виджеты в ячейках центрируются

Grid



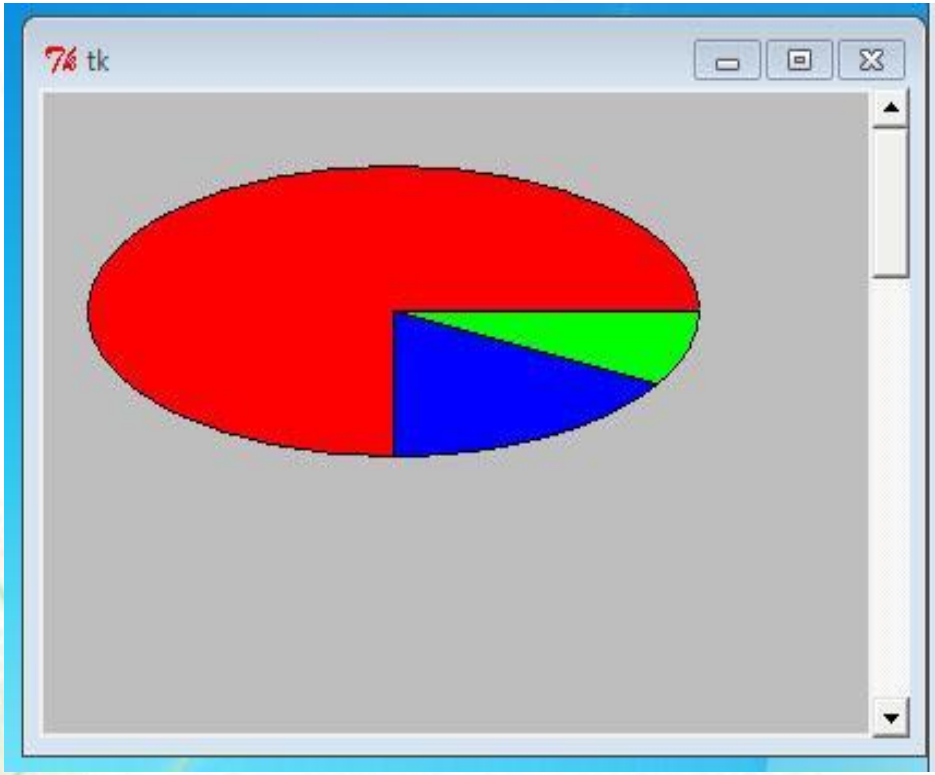

```
from Tkinter import *

root=Tk()

Label(root,text="First:").grid(row=0,sticky=W)
Label(root,text="Second:").grid(row=1,sticky=W)

e1=Entry(root)
e2=Entry(root)

e1.grid(row=0,column=1)
e2.grid(row=1,column=1)

cb=Checkbutton(root,text="Checkbutton")
cb.grid(row=2,columnspan=2,sticky=W)
```

grid_configure(option=value, …)

grid_forget() / grid_remove()

grid_info() → dict – словарь имеющихся опций

Scrollbar

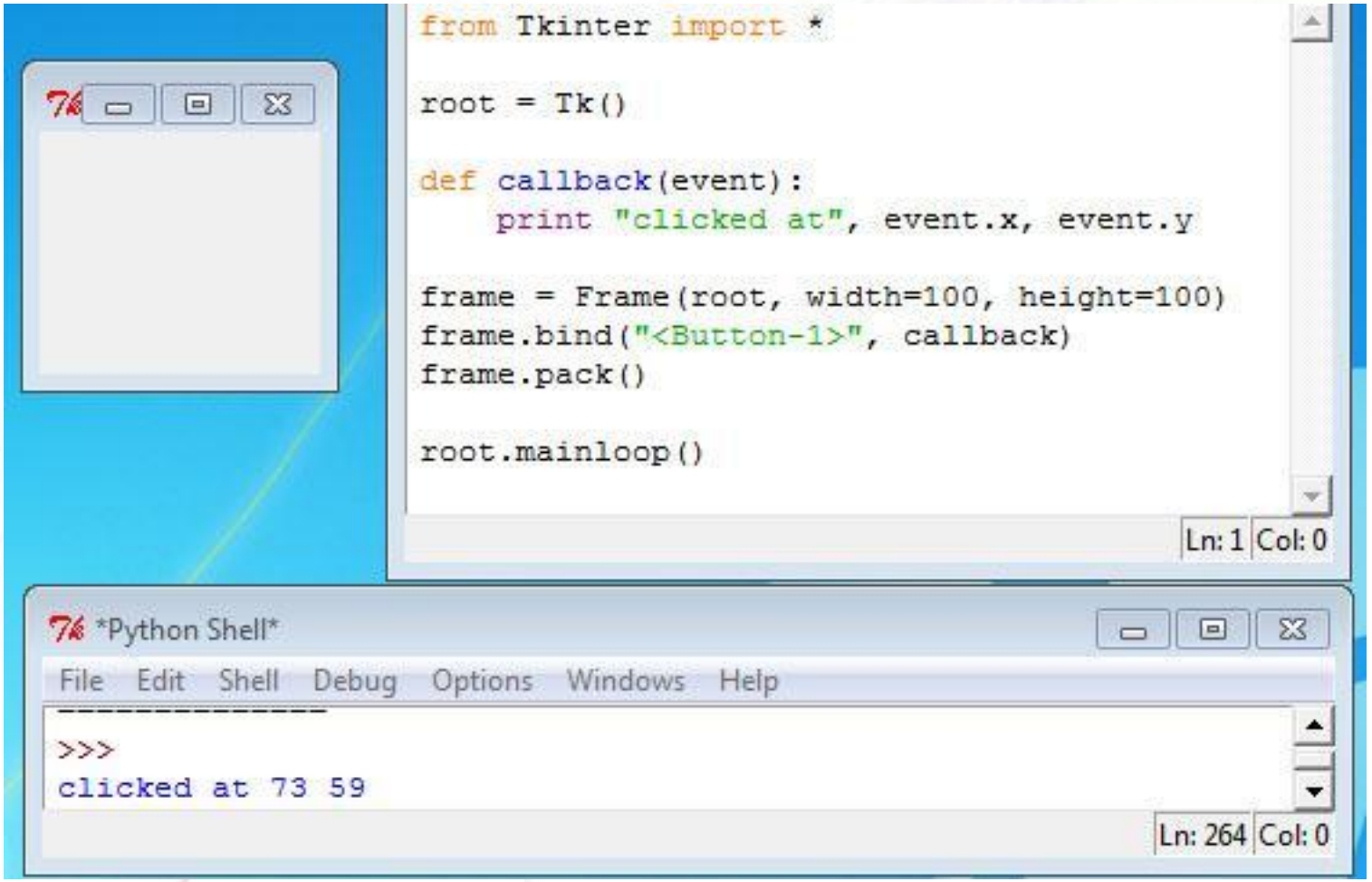



```
from Tkinter import *

root=Tk()

canvas=Canvas(scrollregion=(0,0,100,1000),background="grey")
canvas.grid(row=0,column=0,sticky="nsew")

xy=20,30,300,150
canvas.create_arc(xy,start=0,extent=270,fill="red")
canvas.create_arc(xy,start=270,extent=60,fill="blue")
canvas.create_arc(xy,start=330,extent=30,fill="green")

scroll_y=Scrollbar(orient="vertical",command=canvas.yview)
scroll_y.grid(row=0,column=1,sticky="ns")
canvas['yscrollcommand']=scroll_y.set

root.mainloop()
```

Для Listbox, Text, Canvas, Entry

Связывание (bind)

- Связь между виджетом, событием и действием
- widget.bind(event, handler)



События

- <Button-1> = <ButtonPress-1> = <1>
  - Нажатие на кнопку мыши
  - <1> - левая кнопка, <2> - средняя, <3> - правая
  - event.x, event.y – положение курсора
- <B1-Motion>
  - Перемещение мышки при нажатой кнопке
  - event.x, event.y – положение курсора
- <ButtonRelease-1>
  - Отпустить кнопку
  - event.x, event.y – положение курсора
- <Double-Button-1>
  - Двойное нажатие
  - event.x, event.y – положение курсора
- <Enter>
  - Указатель мыши входит в область виджета
- <Leave>
  - Указатель мыши покидает область виджета

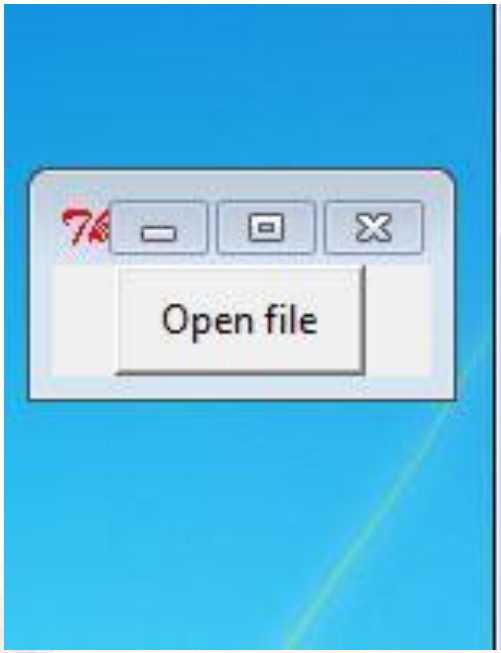
- <Key> = <KeyPress>, <KeyRelease>
  - Значение клавиши передается как event.char
- a = <KeyPress-a>
  - нажатие клавиши a
- <Return>
  - Клавиша Enter
- <Control-x> = <Control-KeyPress-x>, <Shift-x>, <Shift_L-x>, <B1-x>

After

- after(delay_ms,callback,args…) – callback вызывается через delay_ms миллисекунд
- after_cancel(id)
- after(delay_ms) – останавливает петлю событий

```
from Tkinter import *

def change():
    l.configure(text="2nd")

root=Tk()

l=Label(root,text="1st")
l.pack()
root.after(1000,change)

root.mainloop()
```

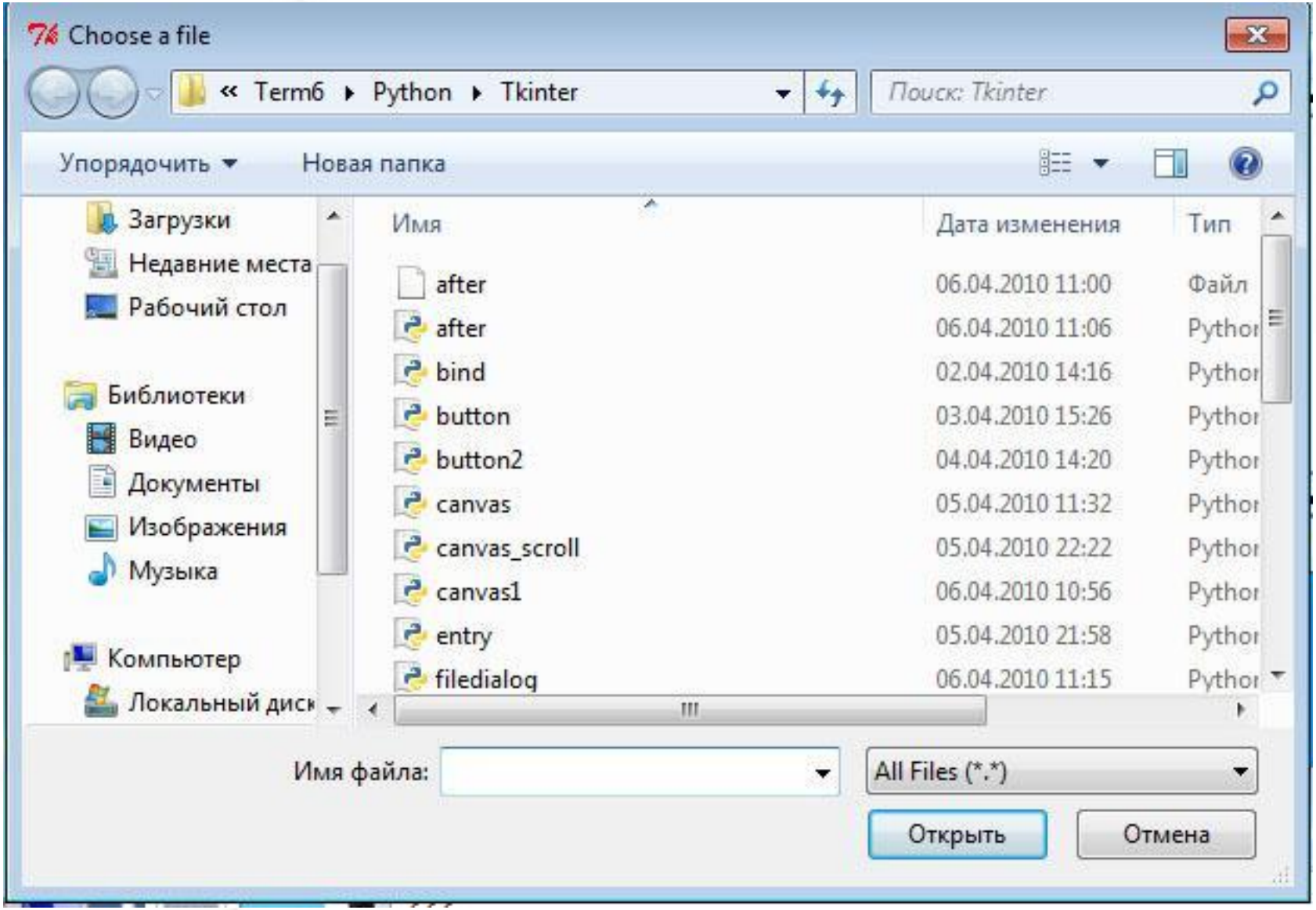
tkFileDialog




```
from Tkinter import *
import tkFileDialog

def open_file():
    options={'parent':root,'title':'Choose a file'}
    file=tkFileDialog.askopenfile(mode='r',**options)

root=Tk()
b=Button(root,text="Open file",command=open_file,
         padx=10,pady=5)
b.pack()
root.mainloop()
```

- tkFileDialog.askopenfile(mode='r', **options)
- tkFileDialog.askopenfiles(mode='r', **options)
- tkFileDialog.askopenfilename(**options)
- tkFileDialog.askopenfilenames(**options)

tkFileDialog



tkFileDialog

- tkFileDialog.askdirectory(**options)
- tkFileDialog.asksaveasfile(mode='w', **options)
- tkFileDialog.asksaveasfilename(**options)

Tcl/Tk & Tkinter

```
button .fred -fg red -text "hi there"
   ^          ^   \_____________________/
   |          |              |
 class       new          options
command  widget    (-opt val -opt val ...)
```

fred=Button(root, fg="red")

Имя виджета само становится командой:

.fred someAction someOptions fred.someAction(someOptions)

Документация

- http://www.pythonware.com/library/tkinter/introduction/index.htm

(An Introduction to Tkinter by Fredrik Lundh)

- Документация Tck/Tk

  Например, http://www.tcl.tk/doc/

- Google)