ВИДЖЕТЫ

АйтиландиЯ
страна талантов

Библиотека tKINTER

- Tkinter (от англ. tk interface) - это графическая библиотека, позволяющая создавать программы с оконным интерфейсом. Эта библиотека является интерфейсом к популярному языку программирования и инструменту создания графических приложений tcl/tk. Для ее использования ее нужно импортировать

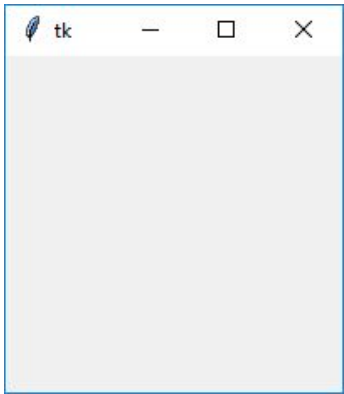
```
from tkinter import*
```

Библиотека tKINTER

Минимальная программа при использовании библиотеки tkinter выглядит следующим так:

```
#импортирует библиотеку
from tkinter import *
root = Tk() #создает графическое
  окно
root.mainloop() #отображает
  содержимое окна
```

```
from tkinter import *
root = Tk()
root.mainloop()
```



Виджет Button

Виджет Button – самая обыкновенная кнопка, которая используется в тысячах программ. Пример кода:

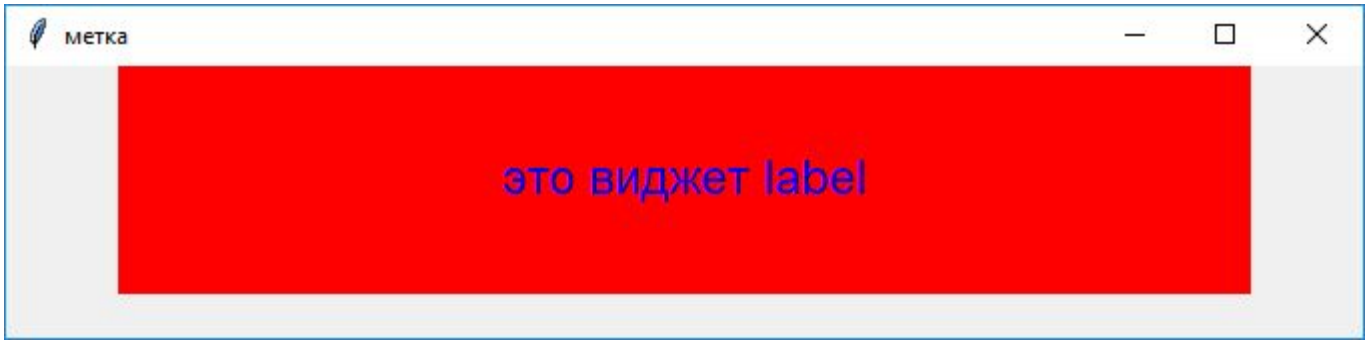
- **text** - какой текст будет отображён на кнопке
- **bg** - цвет кнопки (сокращенно от background)
- **width,height** - соответственно, ширина и длина
- **fg** - цвет текста на кнопке (foreground)
- **font** - шрифт и его размер

```
from tkinter import *
root=Tk()
button1=Button(root,text='Кнопка',width=25,height=5,
               bg='black',fg='yellow',font='arial 14')
button1.pack()
root.mainloop()
```

Виджет label

- **Label** - это виджет, предназначенный для отображения какой-либо надписи без возможности редактирования пользователем. Имеет те же свойства, что и перечисленные свойства кнопки.

```
from tkinter import*
root=Tk()
root.title("метка")

lab=Label(root,text="это виджет label",
          widt=40,height=4,bg="red",
          fg="blue",font="arial 18")

lab.pack()
root.mainloop()
```



Виджет Entry

- **Entry** - это виджет, позволяющий пользователю ввести одну строку текста. Имеет дополнительное свойство bd (сокращённо от borderwidth), позволяющее регулировать ширину границы.
- **borderwidth** - ширина бордюра элемента
- **bd** - сокращение от borderwidth
- **width** - задаёт длину элемента в знакоместах.
- **show** - задает отображаемый символ.

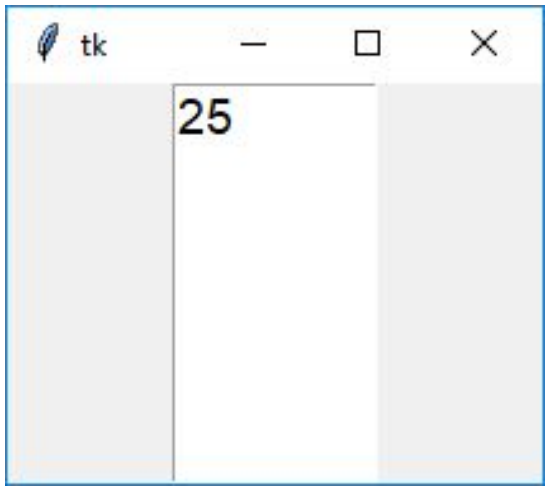
label.py - F:/Айтиландия/Python/виджеты/label.py (3.7.1)

File Edit Format Run Options Window Help

```
from tkinter import*
root=Tk()
root.title("entry")

ent=Entry(root,width=10,font='Arial 14')

ent.pack()

root.mainloop()
```

Виджет text

- **Text** - это виджет, который позволяет пользователю ввести любое количество текста. Имеет дополнительное свойство wrap, отвечающее за перенос (чтобы, например, переносить по словам, нужно использовать значение WORD).Например:

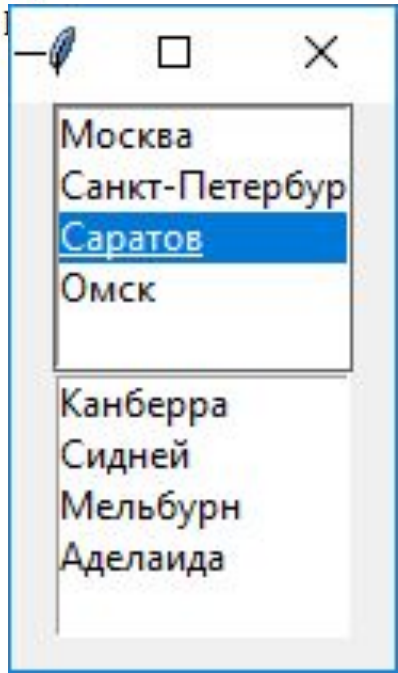



```
from tkinter import *
root=Tk()
text1=Text(root,height=7,width=7,
           font='Arial 14',wrap=WORD)
text1.insert('1.0',"2")
text1.insert(END,"5")
#text1.delete('1.0', END)    # Удалить все
a=text1.get('1.0',END)       # Извлечь все
print(a)
text1.pack()
root.mainloop()
```

Виджет listbox

- **Listbox** - это виджет, который представляет собой список, из элементов которого пользователь может выбирать один или несколько пунктов. Имеет дополнительное свойство selectmode, которое, при значении SINGLE, позволяет пользователю выбрать только один элемент списка, а при значении EXTENDED - любое количество. Пример:

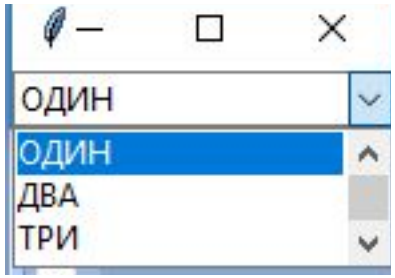
```
from tkinter import *
root=Tk()
listbox1=Listbox(root,height=5,width=15,selectmode=EXTENDE
listbox2=Listbox(root,height=5,width=15,selectmode=SINGLE)
list1=["Москва","Санкт-Петербург","Саратов","Омск"]
list2=["Канберра","Сидней","Мельбурн","Аделаида"]
for i in list1:
    listbox1.insert(END,i)
for i in list2:
    listbox2.insert(END,i)
listbox1.pack()
listbox2.pack()
root.mainloop()
```



Виджет Combobox

- **Виджет Combobox** предназначен для отображения списка значений, их выбора или изменения пользователем. В версии tk ему подобен виджет Listbox. Разница заключается в том, что Combobox имеет возможность сворачиваться подобно свитку, а Listbox будет отображаться всегда открытым. Чтобы отобразить Combobox с заранее заданными значениями в форме, достаточно сделать следующее:

```
import tkinter as tk
import tkinter.ttk as ttk
root = tk.Tk()
frame = tk.Frame(root)
frame.grid()
combobox = ttk.Combobox(
    frame,values = ["ОДИН","ДВА","ТРИ",
                    "четыре","пять","шесть"],height=3)
combobox.set("ОДИН")#с помощью этой строчки мы установим
                    #Combobox в значение ОДИН изначально
combobox.grid(column=0,row=0)#Позиционируем Combobox на форме
root.mainloop()
```

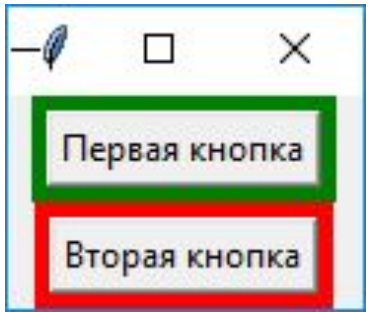


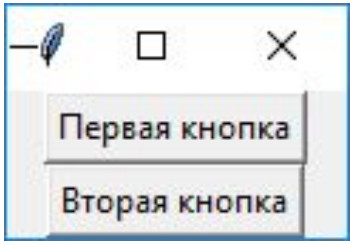
АйтиландиЯ
страна талантов

Виджет frame

- **Виджет Frame** (рамка) предназначен для организации виджетов внутри окна. Рассмотрим пример:

```
from tkinter import *
root=Tk()
button1=Button(root,text='Первая кнопка')
button2=Button(root,text='Вторая кнопка')
button1.pack()
button2.pack()
root.mainloop()
```



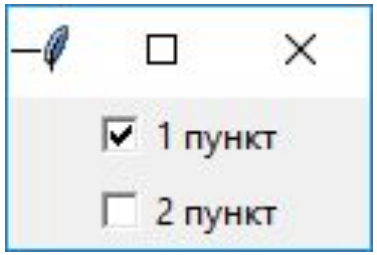



```
from tkinter import *
root=Tk()
frame1=Frame(root,bg='green',bd=5)
frame2=Frame(root,bg='red',bd=5)
button1=Button(frame1,text='Первая кнопка')
button2=Button(frame2,text='Вторая кнопка')
frame1.pack()
frame2.pack()
button1.pack()
button2.pack()
root.mainloop()
```

Виджет checkbutton

- **Checkbutton** - это виджет, который позволяет отметить «галочкой» определенный пункт в окне. При использовании нескольких пунктов нужно каждому присвоить свою переменную. Разберем пример:

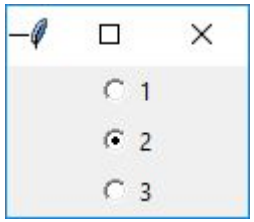
```
from tkinter import *
root=Tk()
var1=IntVar()
var2=IntVar()
check1=Checkbutton(root,text=u'1 пункт',
                   variable=var1,onvalue=1,offvalue=0)
check2=Checkbutton(root,text=u'2 пункт',
                   variable=var2,onvalue=1,offvalue=0)
check1.pack()
check2.pack()
root.mainloop()
```



Виджет radiobutton

- Виджет Radiobutton выполняет функцию, схожую с функцией виджета Checkbutton. Разница в том, что в виджете Radiobutton пользователь может выбрать лишь один из пунктов. Реализация этого виджета несколько иная, чем виджета Checkbutton:

```
from tkinter import *
root=Tk()
var=IntVar()
rbutton1=Radiobutton(root,text='1',variable=var,value=1)
rbutton2=Radiobutton(root,text='2',variable=var,value=2)
rbutton3=Radiobutton(root,text='3',variable=var,value=3)
rbutton1.pack()
rbutton2.pack()
rbutton3.pack()
root.mainloop()
```

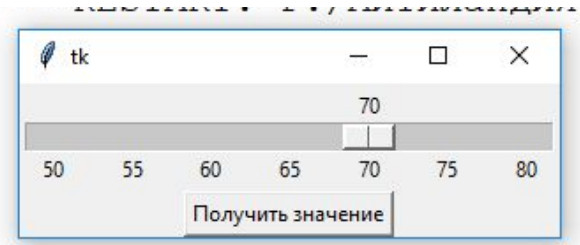


Виджет scale

- **Scale** (шкала) - это виджет, позволяющий выбрать какое-либо значение из заданного диапазона. Свойства:
- **orient** - как расположена шкала на окне. Возможные значения: HORIZONTAL, VERTICAL
- **length** - длина шкалы.
- **from_** - с какого значения начинается шкала.
- **to** - каким значением заканчивается шкала.
- **tickinterval** - интервал, через который отображаются метки шкалы.
- **resolution** - шаг передвижения (минимальная длина, на которую можно передвинуть движок)

```
from tkinter import *
root = Tk()
def getV(root):
    a = scale1.get()
    print ("Значение", a)
scale1 = Scale(root,orient=HORIZONTAL,length=300,
               from_=50,to=80,tickinterval=5,
               resolution=5)
button1 = Button(root,text=u"Получить значение")
scale1.pack()
button1.pack()
button1.bind("<Button-1>",getV)
root.mainloop()
```

```
Значение 50
Значение 50
Значение 55
Значение 70
```

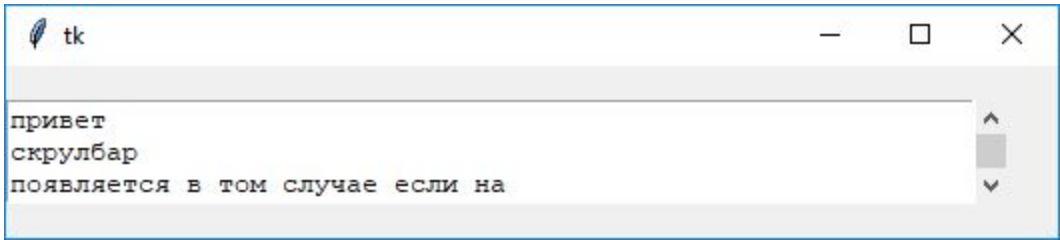


АйтиландиЯ
страна талантов

Виджет scrollbar

- Этот виджет даёт возможность пользователю "прокрутить" другой виджет (например текстовое поле) и часто бывает полезен.

```
from tkinter import *
root = Tk()
text = Text(root, height=3, width=60)
text.pack(side='left')
scrollbar = Scrollbar(root)
scrollbar.pack(side='left')
# первая привязка
scrollbar['command'] = text.yview
# вторая привязка
text['yscrollcommand'] = scrollbar.set
root.mainloop()
```



Progressbar

- Виджет отображает уровень загрузки.
- length - длина полосы.
- Start Запускает бесконечный цикл загрузки. Шаг длиною 1 выполняется один раз в указанное время (в миллисекундах).
- Stop Останавливает цикл загрузки.
- Step Продвигает загрузку на заданное количество шагов.

```
import tkinter as tk
import tkinter.ttk as ttk
root = tk.Tk()
pb = ttk.Progressbar(root, length=100)
pb.pack()
pb.start(100)
root.mainloop()
```

Упаковщик pack()

- Упаковщик pack() является самым интеллектуальным (и самым непредсказуемым). При использовании этого упаковщика с помощью свойства side нужно указать к какой стороне родительского виджета он должен примыкать. Как правило этот упаковщик используют для размещения виджетов друг за другом (слева направо или сверху вниз).

```
from tkinter import *
root=Tk()
button1 = Button(text="1")
button2 = Button(text="2")
button3 = Button(text="3")
button4 = Button(text="4")
button5 = Button(text="5")
button1.pack(side='left')
button2.pack(side='top')
button3.pack(side='left')
button4.pack(side='bottom')
button5.pack(side='right')
root.mainloop()
```

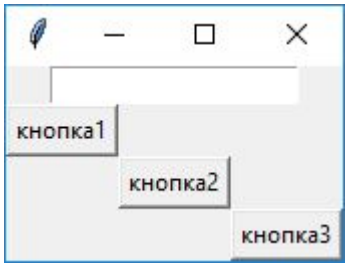
- side ("left"/"right"/"top"/"bottom") - к какой стороне должен примыкать размещаемый виджет.
- fill (None/"x"/"y"/"both") - необходимо ли расширять пространство предоставляемое виджету.
- expand (True/False) - необходимо ли расширять сам виджет, чтобы он занял всё предоставляемое ему пространство.
- in_ - явное указание в какой родительский виджет должен быть

Упаковщик grid()

- Этот упаковщик представляет собой таблицу с ячейками, в которые помещаются виджеты.

```
from tkinter import *
root=Tk()
entry=Entry(root)
button1 = Button(text="кнопка1")
button2 = Button(text="кнопка2")
button3 = Button(text="кнопка3")
entry.grid(row=0,column=0,columnspan=3)
button1.grid(row=1,column=0)
button2.grid(row=2,column=1)
button3.grid(row=3,column=2)
```

- **row** - номер строки
- **rowspan** - сколько строк занимает виджет
- **column** - номер столбца
- **columnspan** - сколько столбцов занимает виджет.
- **sticky** ("n", "s", "e", "w" или их комбинация) - указывает к какой границе "приклеивать" виджет. Позволяет расширять виджет в указанном направлении.
- "n" (север) , "s" (юг) "w" (запад) - "e" (восток)

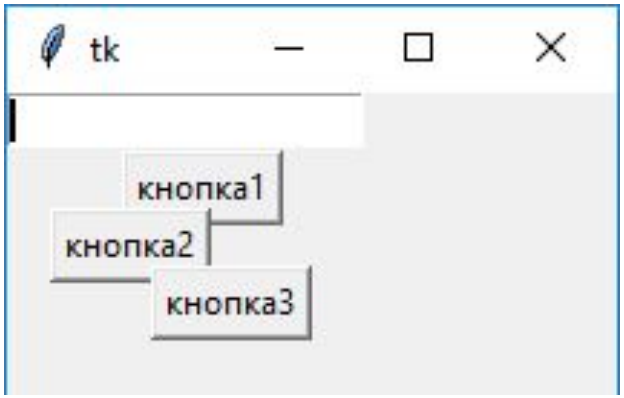


АйтиландиЯ
страна талантов

Упаковщик place()

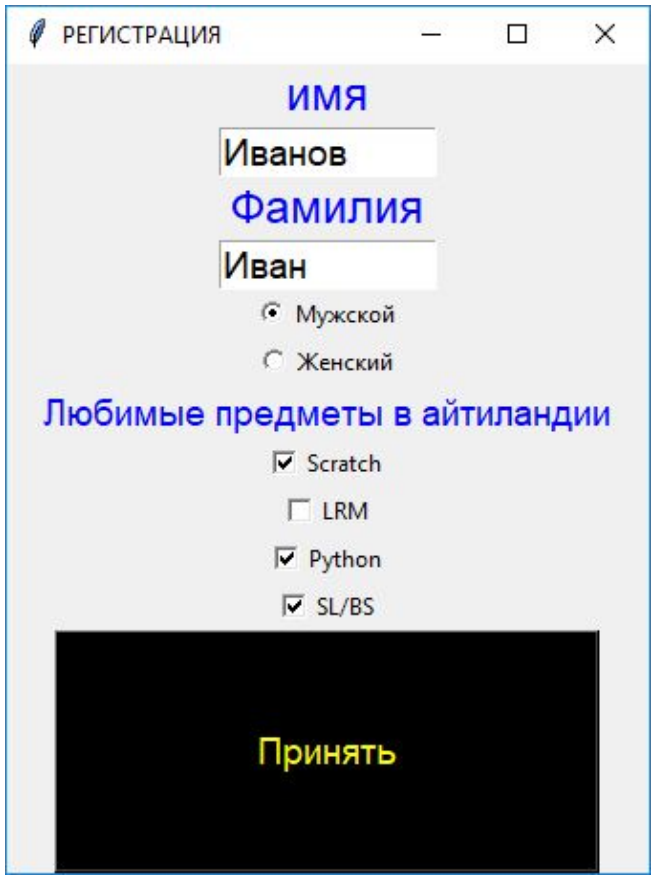
- place представляет собой простой упаковщик, позволяющий размещать виджет в фиксированном месте с фиксированным размером. Также он позволяет указывать координаты размещения в относительных единицах для реализации "резинового" размещения. При использовании этого упаковщика, нам необходимо указывать координаты каждого виджета.

```
from tkinter import *
root=Tk()
entry=Entry(root)
button1 = Button(text="кнопка1")
button2 = Button(text="кнопка2")
button3 = Button(text="кнопка3")
entry.place(x=0,y=0)
button1.place(x=40,y=20)
button2.place(x=15,y=40)
button3.place(x=50,y=60)
```



ПРАКТИЧЕСКАЯ ЧАСТЬ

- Написать программу которая будет выводить окно следующего вида. По нажатии на кнопку вся введенная информация выводится в консоли




```
Имя -  Иванов
Фамилия -  Иван

Пол - мужской
Любимые предметы в айтиландии:
Scratch
Python
Школа лидерства
------------------------------
```

АйтиландиЯ
страна талантов

Домашнее задание

- Доработать программу:
- больше предметов для выбора любимых.
- возможность ввести возраст
- возможность выбора факультета (Teen, Jun), в зависимости от факультета предмет «Школа лидерства» (teen) может поменяться при выводе в консоль на «Бизнес школа»(jun).