

Виджеты и их свойства



Кнопки

- ▣ Объект-кнопка создается вызовом класса `Button` модуля `tkinter`. При этом обязательным аргументом является лишь родительский виджет (например, окно верхнего уровня). Другие свойства могут указываться при создании кнопки или задаваться (изменяться) позже.
- ▣ Синтаксис:

```
переменная = Button (родит_виджет, [свойство=значение, ... ..])
```

Некоторые свойства кнопок

```
from tkinter import *
```

```
root = Tk()
```

```
but = Button(root,  
             text="Это кнопка", #надпись на кнопке  
             width=30,height=5, #ширина и высота  
             bg="white",fg="blue") #цвет фона и надписи
```

```
but.pack()  
root.mainloop()
```

bg и fg – это сокращения от background (фон) и foreground (передний план). Ширина и высота измеряются в знаках (количество символов).

Метки

- ▣ Метки (или надписи) — это достаточно простые виджеты, содержащие строку (или несколько строк) текста и служащие в основном для информирования пользователя.

```
lab = Label(root, text="Это метка! \n Из двух строк.", font="Arial 18")
```

Однострочное текстовое поле

- ▣ Такое поле создается вызовом класса `Entry` модуля `tkinter`. В него пользователь может ввести только одну строку текста.

```
ent = Entry(root,width=20,bd=3)
```

`bd` – это сокращение от `borderwidth` (ширина границы)

Многострочное текстовое поле

- Text предназначен для предоставления пользователю возможности ввода не одной строки текста, а существенно больше.

```
tex = Text(root,width=40,  
font="Verdana 12",  
wrap=WORD)
```

- Свойство (wrap) в зависимости от своего значения позволяет переносить текст, вводимый пользователем либо по символам, либо по словам, либо вообще не переносить, пока пользователь не нажмет Enter.

Радиокнопки (переключатели)

- ▣ Объект-радиокнопка никогда не используется по одному. Их используют группами, при этом в одной группе может быть «включена» лишь одна кнопка.

```
var=IntVar()
var.set(1)
rad0 = Radiobutton(root,text="Первая",
    variable=var,value=0)
rad1 = Radiobutton(root,text="Вторая",
    variable=var,value=1)
rad2 = Radiobutton(root,text="Третья",
    variable=var,value=2)
```

Флажки

- ▣ Объект `checkboxbutton` предназначен для выбора не взаимоисключающих пунктов в окне (в группе можно активировать один, два или более флажков или не один). В отличие от радиокнопок, значение каждого флажка привязывается к своей переменной, значение которой определяется опциями `onvalue` (включено) и `offvalue` (выключено) в описании флажка.

```
c1 = IntVar()
c2 = IntVar()
che1 = Checkbutton(root, text="Первый флажок",
variable=c1, onvalue=1, offvalue=0)
che2 = Checkbutton(root, text="Второй флажок",
variable=c2, onvalue=2, offvalue=0)
```


Списки

- Вызов класса `Listbox` создает объект, в котором пользователь может выбрать один или несколько пунктов в зависимости от значения опции `selectmode`. В примере ниже значение `SINGLE` позволяет выбирать лишь один пункт из списка.

```
r = ['Linux','Python','Tk','Tkinter']  
lis = Listbox(root,selectmode=SINGLE,height=4)  
for i in r: lis.insert(END,i)
```

- Изначально список (`Listbox`) пуст. С помощью цикла `for` в него добавляются пункты из списка (тип данных) `r`. Добавление происходит с помощью специального метода класса `Listbox` — `insert`. Данный метод принимает два параметра: куда добавить и что добавить.

Практическая работа

- Создайте два скрипта на языке программирования Python и с использованием модуля Tkinter генерирующие шаблоны представленные ниже.



Ваш адрес:

Комментарий:

Отправить

Сколько штук?

0-10

11-20

21-30

31-40

Какого цвета?

RED

BLUE

GREEN

YELLOW

Frame (рамка)

```
from tkinter import *
```

```
root = Tk()
```

```
fra1 = Frame(root,width=500,height=100,bg="darkred")
```

```
fra2 = Frame(root,width=300,height=200,bg="green",bd=20)
```

```
fra3 = Frame(root,width=500,height=150,bg="darkblue")
```

```
fra1.pack()
```

```
fra2.pack()
```

```
fra3.pack()
```

```
root.mainloop()
```

Данный скрипт создает три фрейма разного размера. Свойство `bd` (сокращение от `boderwidth`) определяет расстояния от края рамки до заключенных в нее виджетов (если они есть).

На фреймах также можно размещать виджеты как на основном окне (`root`).

- Здесь текстовое поле находится на рамке fra2.

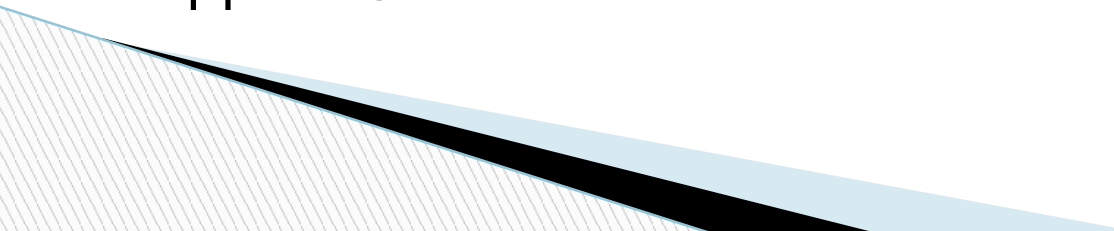
```
ent1 = Entry(fra2,width=20)  
ent1.pack()
```

Scale (шкала)

- ▣ Назначение шкалы — это предоставление пользователю выбора какого-то значения из определенного диапазона. Внешне шкала представляет собой горизонтальную или вертикальную полосу с разметкой, по которой пользователь может передвигать движок, осуществляя тем самым выбор значения.

```
sca1 = Scale(fra3,orient=HORIZONTAL,length=300,  
            from_=0,to=100,tickinterval=10,resolution=5)  
sca2 = Scale(root,orient=VERTICAL,length=400,  
            from_=1,to=2,tickinterval=0.1,resolution=0.1)
```

Свойства:

- `orient` определяет направление шкалы;
 - `length` – длина шкалы в пикселях;
 - `from_` и `to` – с какого значения шкала начинается и каким заканчивается (т. е. диапазон значений);
 - `tickinterval` – интервал, через который отображаются метки для шкалы;
 - `resolution` - минимальная длина отрезка, на которую пользователь может передвинуть движок.
- 

Scrollbar (полоса прокрутки)

- Данный виджет позволяет прокручивать содержимое другого виджета (например, текстового поля или списка). Прокрутка может быть как по горизонтали, так и по вертикали.

```
from tkinter import *  
root = Tk()  
  
tx = Text(root,width=40,height=3,font='14')  
scr = Scrollbar(root,command=tx.yview)  
tx.configure(yscrollcommand=scr.set)  
  
tx.grid(row=0,column=0)  
scr.grid(row=0,column=1)  
root.mainloop()
```


Toplevel (окно верхнего уровня)

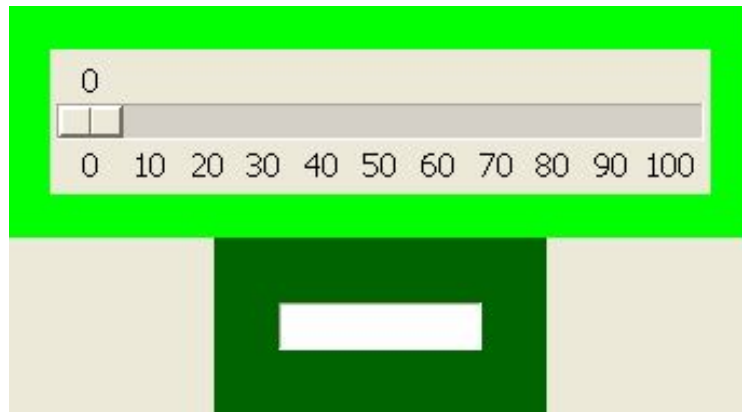
- С помощью класс `Toplevel` создаются дочерние окна, на которых также могут располагаться виджеты.

```
win = Toplevel(root,relief=SUNKEN,bd=10,bg="lightblue")  
win.title("Дочернее окно")  
win.minsize(width=400,height=200)
```

- Метод `title` определяет заголовок окна. Метод `minsize` конфигурирует минимальный размер окна (есть метод `maxsize`, определяющий максимальный размер окна). Если значение аргументов `minsize = maxsize`, то пользователь не сможет менять размеры окна.

практическая работа

- 1. Создайте два скрипта на языке программирования Python и с использованием модуля Tkinter генерирующие шаблоны представленные ниже.



Это многострочное текстовое поле с движком и вертикальной полосой прокрутки. Если сюда ввести много

- 2. Создайте приложение, состоящее из главного и двух дочерних окон. На каждом из трех окон должны располагаться один или два любых графических объекта.