

# Графический интерфейс в Python

# модуль **tkinter**

Алгоритм работы:

***import tkinter as tk*** # импортируем модуль

***win = tk.Tk()*** # создаем окно

***win.mainloop()*** # обязательная инструкция  
для конца программы

# Еще вариант создания окна

```
from tkinter import*  
win = Tk()  
win.geometry('100*100')  
win.config (bg='red')  
win.title ('Красное окно')  
win.mainloop()
```

# Содержимое скрипта

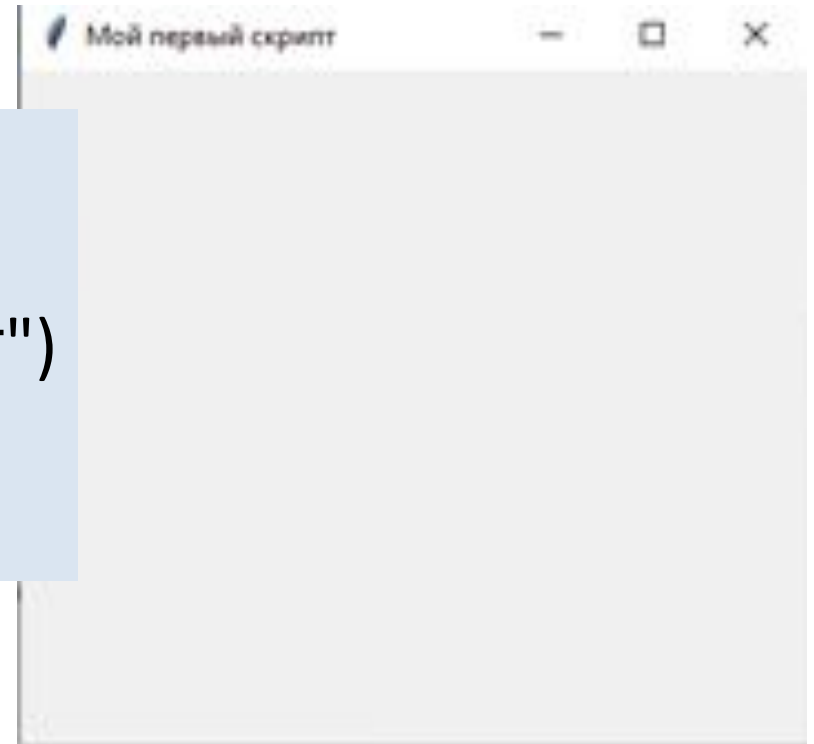
***win.title(«Мой первый скрипт») #***

название окна

***win.geometry("350x300") # ширина\*высота***

окна

```
import tkinter as tk
win = tk.Tk()
win.title("Мой первый скрипт")
win.geometry("350x300")
win.mainloop()
```



# Виджеты (примитивы, элементы)

- основные (метка, кнопка, текстовое поле)
- свойства и методы бывают общие и частные
- способы конфигурирования: в момент создания объекта; с помощью метода `config()` или `configure()`; путем обращения к свойству как к элементу словаря.

# Label – метка

- отображает текст в окне и служит в основном для информационных целей (вывод сообщений, подпись других элементов интерфейса)
- Свойства метки во многом схожи с таковыми у кнопки. Однако у меток нет опции `command`. Клик по метке не обрабатывается Tkinter

# Элементы окна

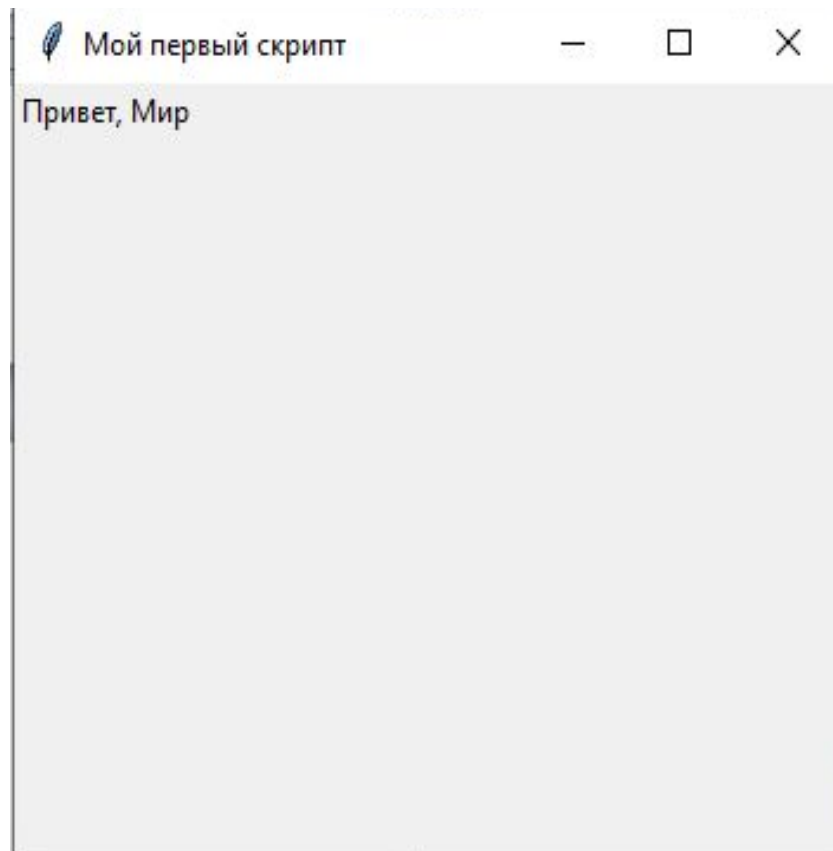
***txt = tk.Label(win, text = "Комментарий") #***

txt это название элемента-метки, Label это название метки, win это окно, в котором отображается метка, text это параметр, отвечающий за текст в метке, “Комментарий” это текст который будет отображаться в метке.

***txt.grid(row = m, column = n) # место***  
расположение метки

# Label - метка

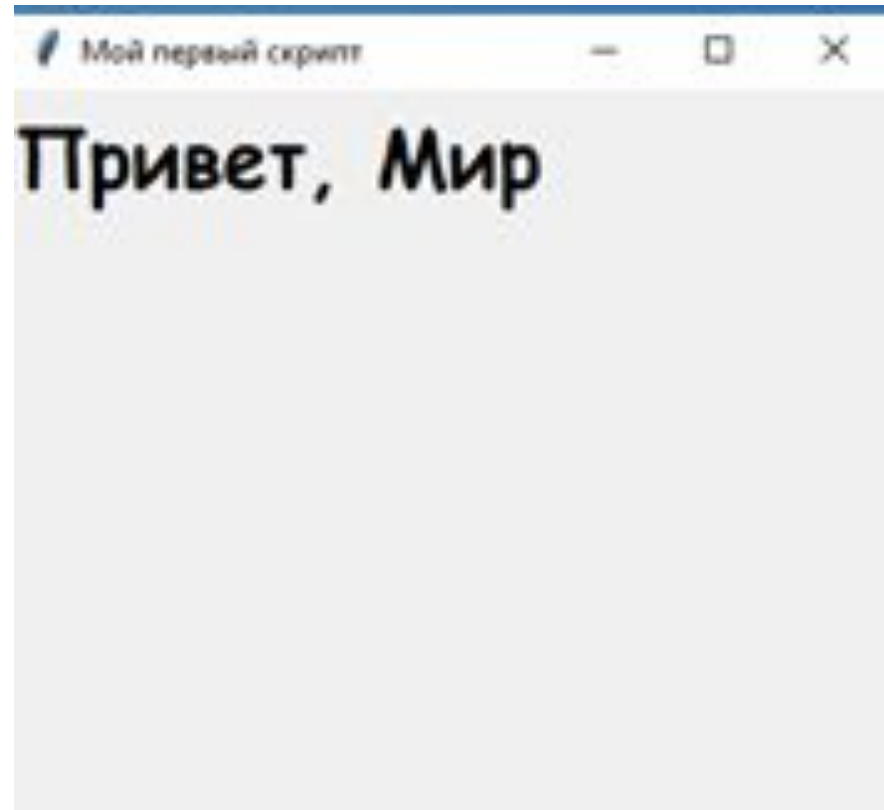
```
import tkinter as tk
win = tk.Tk()
win.title("Мой первый скрипт")
win.geometry("350x300")
txt = tk.Label(win, text =
"Привет, Мир")
txt.grid(row = 1, column = 0)
win.mainloop()
```





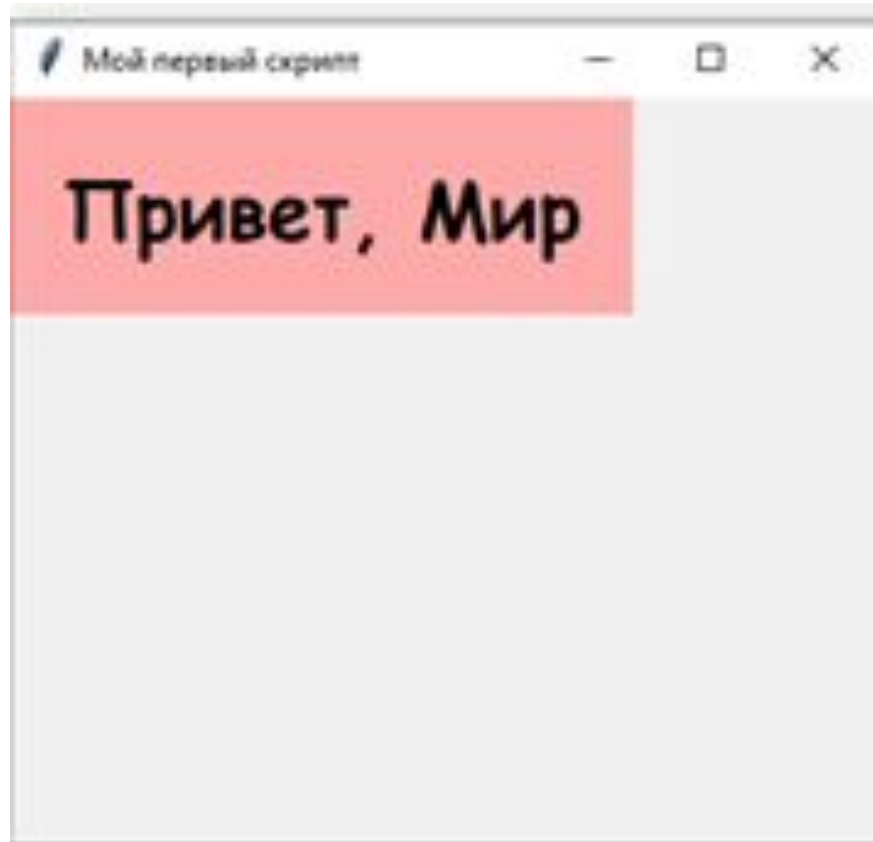
# СВОЙСТВО font

```
txt = tk.Label(win, text = "Привет,  
Мир", font=("Comic Sans MS", 24, "bold"))
```



# ФОН

- `txt.config(bd=20, bg='#ffaaaa')`



# Button – кнопка

У класса Button есть обязательный параметр – объект, которому кнопка принадлежит (чаще используют root).

fg - меняет цвет текста кнопки или любого другого виджета

bg - цвет фона любого виджета

# Создание кнопки

```
from tkinter import *
```

```
window = Tk()
```

```
window.title("Добро пожаловать в приложение  
PythonRu")
```

```
window.geometry('400x250')
```

```
lbl = Label(window, text="Привет", font=("Arial Bold", 50))
```

```
lbl.grid(column=0, row=0)
```

```
btn = Button(window, text="Не нажимать!")
```

```
btn.grid(column=1, row=0)
```

```
window.mainloop()
```

# Событие

- Действия(алгоритм) часто оформляют в виде функции, а затем вызывают необходимым событием. Создадим функцию:

```
def hello(event):  
    print("Привет, программист!")
```

Параметр event – это какое-либо событие. Событие нажатия левой кнопки мыши выглядит так: <Button-1>. Требуется связать это событие с обработчиком (функцией hello). Для связи предназначен метод bind. Синтаксис:

```
but1.bind("<Button-1>",hello)
```

# Привязка события (простой вывод)

```
from tkinter import *

def hello(event):
    print("Привет, программист!")

root = Tk()
root.title("Заголовок окна программы")
root.geometry("400x200")
but1 = Button(root)
but1["text"] = "Название кнопки"
but1.bind("<Button-1>",hello)

but1.pack()
root.mainloop()
input()
```

# Привязка события (посложнее)

```
from tkinter import *
```

```
def clicked():  
    lbl.configure(text="Я же просил...")
```

```
window = Tk()  
window.title("Добро пожаловать в приложение PythonRu")  
window.geometry('400x250')  
lbl = Label(window, text="Привет", font=("Arial Bold", 50))  
lbl.grid(column=0, row=0)  
btn = Button(window, text="Не нажимать!", command=clicked)  
btn.grid(column=1, row=0)  
window.mainloop()
```

# ТЕКСТОВОЕ ПОЛЕ

## entry

```
txt = Entry(window, width=10)
```



# Пример

```
from tkinter import *
```

```
def clicked():  
    lbl.configure(text="Я же просил...")
```

```
window = Tk()  
window.title("Добро пожаловать в приложение PythonRu")  
window.geometry('400x250')  
lbl = Label(window, text="Привет")  
lbl.grid(column=0, row=0)  
txt = Entry(window,width=10)  
txt.grid(column=1, row=0)  
btn = Button(window, text="Не нажимать!", command=clicked)  
btn.grid(column=2, row=0)  
window.mainloop()
```

# Получение данных

можно получить текст ввода, используя функцию **get**. Мы можем записать код для выбранной функции таким образом:

```
def clicked():  
    res = "Привет {}".format(txt.get())  
    lbl.configure(text=res)
```

# Получение данных

```
from tkinter import *
```

```
def clicked():  
    res = "Привет {}".format(txt.get())  
    lbl.configure(text=res)
```

```
window = Tk()  
window.title("Добро пожаловать в приложение PythonRu")  
window.geometry('400x250')  
lbl = Label(window, text="Привет")  
lbl.grid(column=0, row=0)  
txt = Entry(window,width=10)  
txt.grid(column=1, row=0)  
btn = Button(window, text="Клик!", command=clicked)  
btn.grid(column=2, row=0)  
window.mainloop()
```

# Добавление виджета Combobox

```
from tkinter import *  
from tkinter.ttk import Combobox  
  
window = Tk()  
window.title("Добро пожаловать в приложение PythonRu")  
window.geometry('400x250')  
combo = Combobox(window)  
combo['values'] = (1, 2, 3, 4, 5, "Текст")  
combo.current(1) # установите вариант по умолчанию  
combo.grid(column=0, row=0)  
window.mainloop()
```

# Добавление виджета Checkbutton

```
from tkinter import *  
from tkinter.ttk import Checkbutton  
  
window = Tk()  
window.title("Добро пожаловать в приложение PythonRu")  
window.geometry('400x250')  
chk_state = BooleanVar()  
chk_state.set(True) # задайте проверку состояния чекбокса  
chk = Checkbutton(window, text='Выбрать', var=chk_state)  
chk.grid(column=0, row=0)  
window.mainloop()
```

# Установка значения

- Здесь мы создаем переменную типа BooleanVar, которая не является стандартной переменной Python, это переменная Tkinter, затем передаем ее классу Checkbutton, чтобы установить состояние чекбокса как True в приведенном выше примере.
- Вы можете установить для BooleanVar значение false, что бы чекбокс не был отмечен.  
Так же, используйте IntVar вместо BooleanVar и установите значения 0 и 1.

```
chk_state = IntVar()
```

```
chk_state.set(0) # False
```

```
chk_state.set(1) # True Э
```

- Эти примеры дают тот же результат, что и BooleanVar.

# Добавление виджетов Radio Button

```
from tkinter import *  
from tkinter.ttk import Radiobutton  
  
window = Tk()  
window.title("Добро пожаловать в приложение PythonRu")  
window.geometry('400x250')  
rad1 = Radiobutton(window, text='Первый', value=1)  
rad2 = Radiobutton(window, text='Второй', value=2)  
rad3 = Radiobutton(window, text='Третий', value=3)  
rad1.grid(column=0, row=0)  
rad2.grid(column=1, row=0)  
rad3.grid(column=2, row=0)  
window.mainloop()
```

# СВОЙСТВО command

```
rad1 = Radiobutton(window,text='Первая',  
value=1, command=clicked)
```

```
def clicked():  
    # Делайте, что нужно
```



# Получение значения Radio Button (Избранная Radio Button)

```
from tkinter import *
from tkinter.ttk import Radiobutton

def clicked():
    lbl.configure(text=selected.get())

window = Tk()
window.title("Добро пожаловать в приложение PythonRu")
window.geometry('400x250')
selected = IntVar()
rad1 = Radiobutton(window,text='Первый', value=1, variable=selected)
rad2 = Radiobutton(window,text='Второй', value=2, variable=selected)
rad3 = Radiobutton(window,text='Третий', value=3, variable=selected)
btn = Button(window, text="Клик", command=clicked)
lbl = Label(window)
rad1.grid(column=0, row=0)
rad2.grid(column=1, row=0)
rad3.grid(column=2, row=0)
btn.grid(column=3, row=0)
lbl.grid(column=0, row=1)
window.mainloop()
```

# Диалоговые окна (открыть)

```
from tkinter import *  
from tkinter.filedialog import *  
  
root = Tk()  
root.title("Пример работы диалоговых окон")  
root.minsize(width=500, height=400)  
  
op = askopenfilename()  
sa = asksaveasfilename()  
  
root.mainloop()
```

# Диалоговое окно с открытием текстового файла

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

root = Tk()
root.title("Пример работы диалоговых окон")
root.minsize(width=500, height=400)

txt = Text(root, width=40, height=15, font="14")
txt.pack()

op = askopenfilename()

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

root.mainloop()
```

# Редактор

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

# открытие файла
def _open():
    op = askopenfilename()
    print(op)
    f = open(op, "r", encoding='utf-8')
    content = f.read()
    txt.delete(1.0,END)
    txt.insert(END, content)

# сохранение файла
def _save():
    sa = asksaveasfilename()
    content = txt.get(1.0,END)
    f = open(sa, "w", encoding='utf-8')
    f.write(content)
    f.close()

root = Tk()
root.title("Текстовый редактор")
root.minsize(width=500, height=400)
m = Menu(root)
root.config(menu=m)

fm = Menu(m)
m.add_cascade(label="Файл", menu=fm)
fm.add_command(label="Открыть...", command=_open)
fm.add_command(label="Сохранить как...", command=_save)

txt = Text(root, width=40, height=15, font=" Courier 14")
txt.pack()

root.mainloop()
```

# Метод grid()

- менеджер геометрии, использующий табличный способ размещения
  - Родительский контейнер разделяется на ячейки (номер строки и номер столбца)
  - Ячейки можно объединять (colspan – по горизонтали, rowspan – по вертикали)

0, 0		
1, 0	1, 1	1, 2
2, 0		2, 2
3, 0		3, 2
4, 0		

# Пример

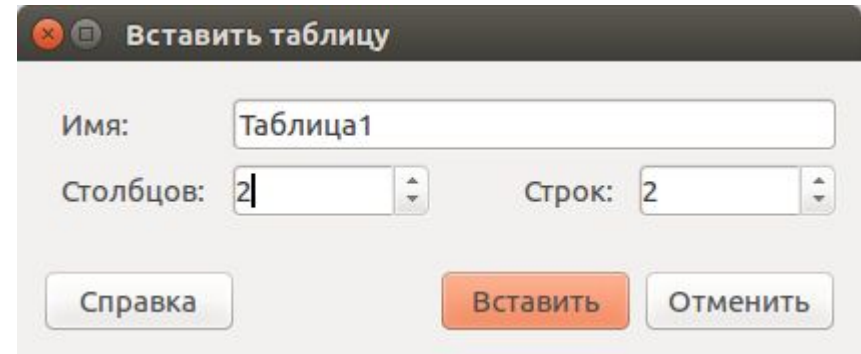
```
from tkinter import *  
root = Tk()
```

```
Label(text="Имя:").grid(row=0, column=0)  
table_name = Entry(width=30)  
table_name.grid(row=0, column=1, colspan=3)
```

```
Label(text="Столбцов:").grid(row=1, column=0)  
table_column = Spinbox(width=7, from_=1, to=50)  
table_column.grid(row=1, column=1)  
Label(text="Строк:").grid(row=1, column=2)  
table_row = Spinbox(width=7, from_=1, to=100)  
table_row.grid(row=1, column=3)
```

```
Button(text="Справка").grid(row=2, column=0)  
Button(text="Вставить").grid(row=2, column=2)  
Button(text="Отменить").grid(row=2, column=3)
```

```
root.mainloop()
```



# Использование внешних и внутренних отступов (padx, pady, ipadx, ipady)

```
from tkinter import *  
root = Tk()  
  
Label(text="Имя:").grid(row=0, column=0, sticky=W, padx=10, pady=10)  
table_name = Entry()  
table_name.grid(row=0, column=1, colspan=3, sticky=W+E, padx=10)  
  
Label(text="Столбцов:").grid(row=1, column=0, sticky=W, padx=10, pady=10)  
table_column = Spinbox(width=7, from_=1, to=50)  
table_column.grid(row=1, column=1, padx=10)  
Label(text="Строк:").grid(row=1, column=2, sticky=E)  
table_row = Spinbox(width=7, from_=1, to=100)  
table_row.grid(row=1, column=3, sticky=E, padx=10)  
  
Button(text="Справка").grid(row=2, column=0, padx=10, pady=10)  
Button(text="Вставить").grid(row=2, column=2)  
Button(text="Отменить").grid(row=2, column=3, padx=10)  
  
root.mainloop()
```