БОЛЕЕ СОВЕРШЕННАЯ ГРАФИКА С МОДУЛЕМ TKINTER

```
1 from tkinter import *
2 tk = Tk()
3 btn = Button(tk, text="нажми меня")
4 btn.pack()
```

Мы создали переменную и сохранили в ней объект класса `Tk`
Объект `tk` создает пустое окно, в которое можно добавлять кнопки, строки ввода, холсты для рисования и так далее.
Это основной класс модуля `tkinter`.
Без создания объекта класса `Tk` работать с графикой или анимацией с помощью этого модуля невозможно.

Именованные аргументы

```
1  from tkinter import *
2  tk = Tk()
3
4  def hello():
5   print('привет')
6
7  btn = Button(tk, text="нажми меня", command=hello)
8  btn.pack()
```

Именованные аргументы

Допустим, у нас есть функция `person`, принимающая два аргумента:
`width` и `height`.

Person — человек
Width — ширина
Height — высота

```
1  def person(width, height):
2   print('Моя ширина - %s, а высота - %s' % (width, height))
3
4  person(4,3)
5
```

Именованные аргументы позволяют вызвать эту функцию, указывая для каждого значения имя соответствующего аргумента:

```
1  def person(width, height):
2   print('Моя ширина - %s, а высота - %s' % (width, height))
3
4  person(height=3, width=4)
5
```

Создаем холст для рисования

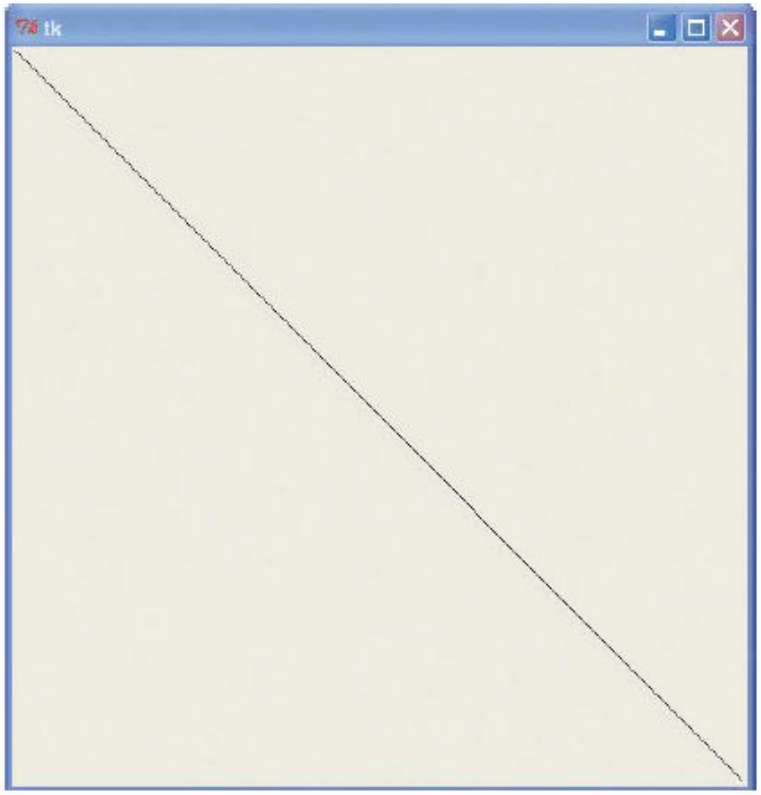
холст — объект класса `Canvas`, который входит в модуль `tkinter`

```
1  from tkinter import *
2
3  tk = Tk()
4  canvas = Canvas(tk, width=500, height=500)
5  canvas.pack()
```

Упаковщик (packer) вызывается методом **pack**, который имеется у всех виджетов-объектов. Мы уже использовали его. Если к элементу интерфейса не применить какой-либо из менеджеров геометрии, то он не отобразится в окне.

Рисование линий

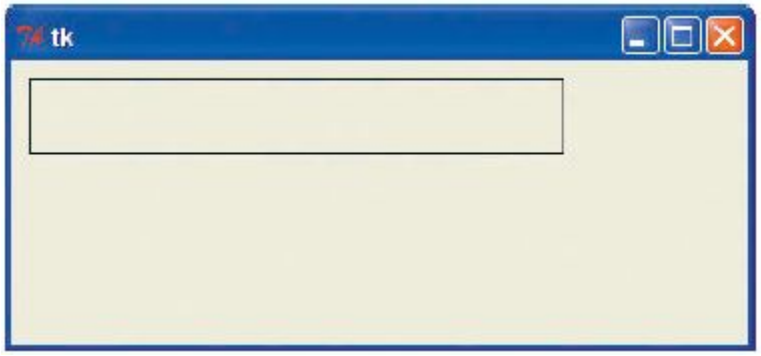
```
1  from tkinter import *
2  tk = Tk()
3  canvas = Canvas(tk, width=500, height=500)
4  canvas.pack()
5  canvas.create_line(0, 0, 500, 500)
```



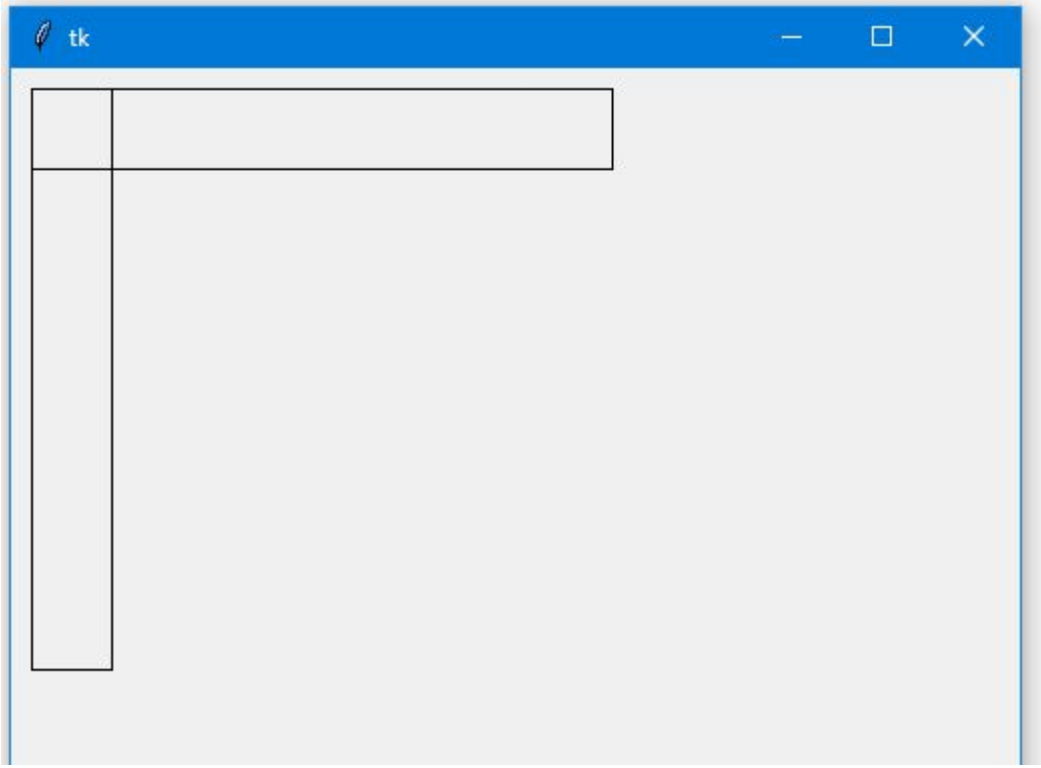
Рисование прямоугольников

```
1  from tkinter import *
2  tk = Tk()
3  canvas = Canvas(tk, width=500, height=500)
4  canvas.pack()
5  canvas.create_rectangle(10, 10, 50, 50)
```

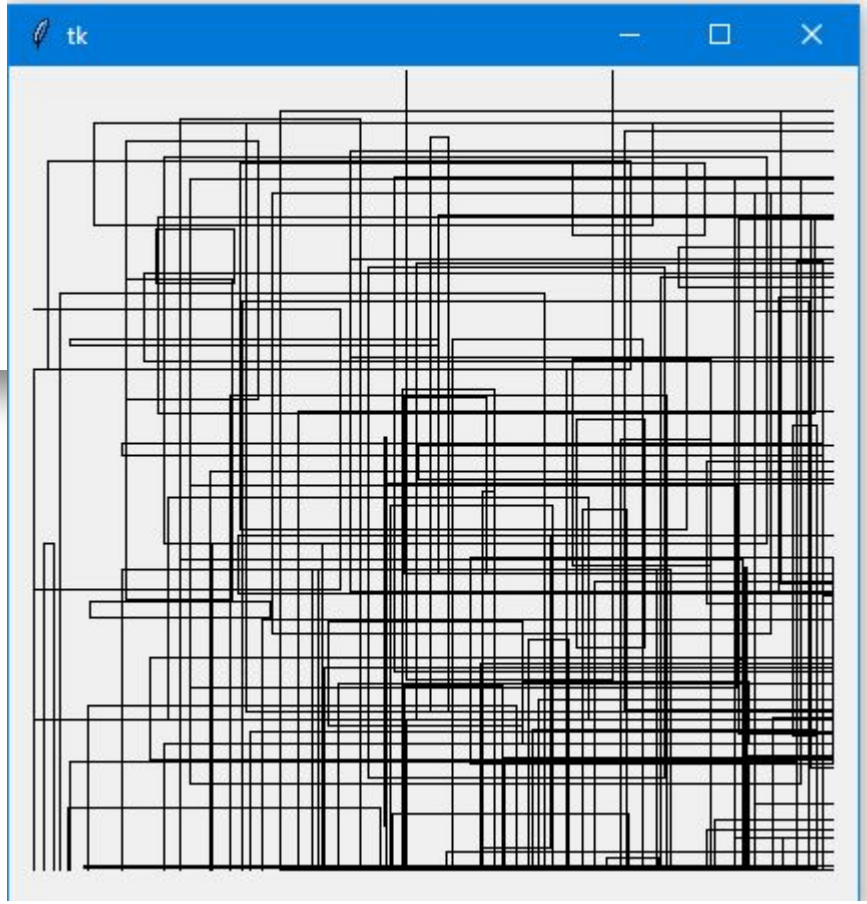
```
1  from tkinter import *
2  tk = Tk()
3  canvas = Canvas(tk, width=500, height=500)
4  canvas.pack()
5  canvas.create_rectangle(10, 10, 300, 50)
```



```
1 from tkinter import *
2 tk = Tk()
3 canvas = Canvas(tk, width=500, height=500)
4 canvas.pack()
5 canvas.create_rectangle(10, 10, 300, 50)
6 canvas.create_rectangle(10, 10, 50, 300)
```

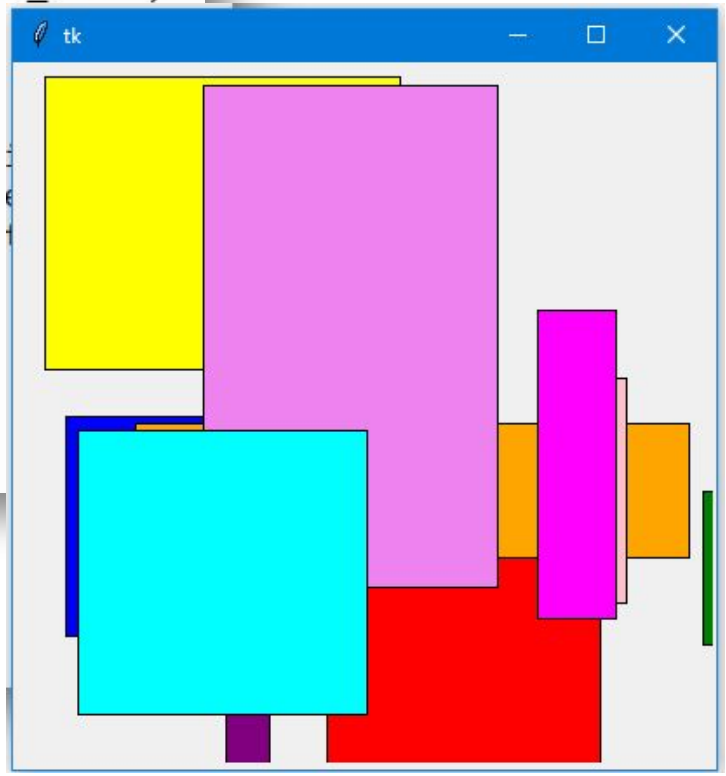


```python
from tkinter import *
import random

tk = Tk()
canvas = Canvas(tk, width=400, height=400)
canvas.pack()

def random_rectangle(width, height):
    x1 = random.randrange(width)
    y1 = random.randrange(height)
    x2 = x1 + random.randrange(width)
    y2 = y1 + random.randrange(height)
    canvas.create_rectangle(x1, y1, x2, y2)

for x in range(0, 100):
    random_rectangle(400, 400)

```



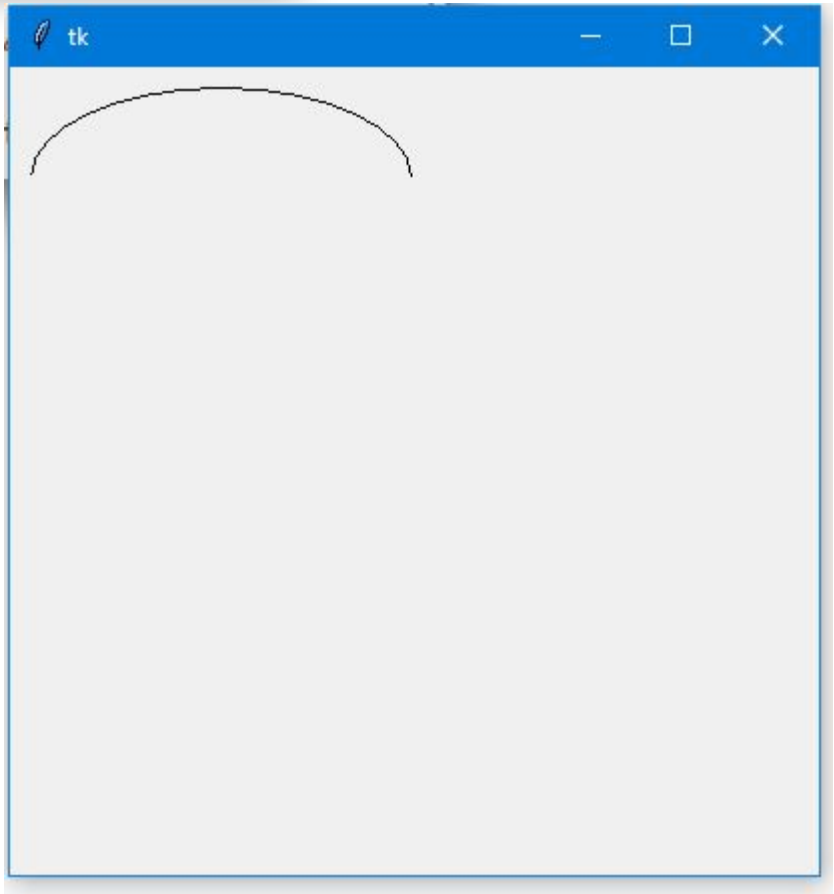
Рисование в цвете

```
from tkinter import *
import random
tk = Tk()
canvas = Canvas(tk, width=400, height=400)
canvas.pack()
def random_rectangle(width, height, fill_color):
    x1 = random.randrange(width)
    y1 = random.randrange(height)
    x2 = random.randrange(x1 + random.randrange(width))
    y2 = random.randrange(y1 + random.randrange(height))
    canvas.create_rectangle(x1, y1, x2, y2, fill=fill_color)
random_rectangle(400, 400, 'green')
random_rectangle(400, 400, 'red')
random_rectangle(400, 400, 'blue')
random_rectangle(400, 400, 'orange')
random_rectangle(400, 400, 'yellow')
random_rectangle(400, 400, 'pink')
random_rectangle(400, 400, 'purple')
random_rectangle(400, 400, 'violet')
random_rectangle(400, 400, 'magenta')
random_rectangle(400, 400, 'cyan')
```

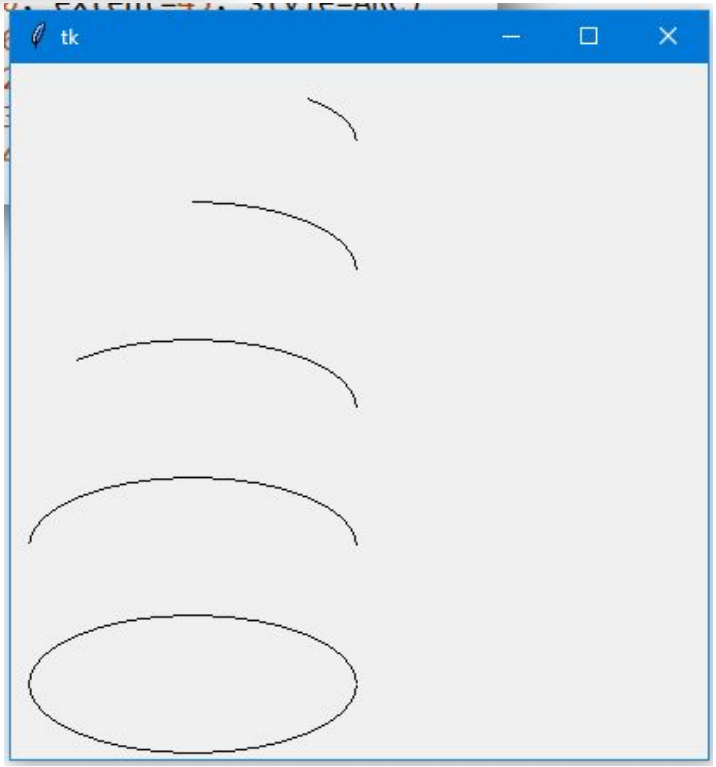


Рисование дуг

```
1  from tkinter import *
2
3
4  tk = Tk()
5  canvas = Canvas(tk, width=400, height=400)
6  canvas.pack()
7  canvas.create_arc(10, 10, 200, 100, extent=180, style=ARC)
```

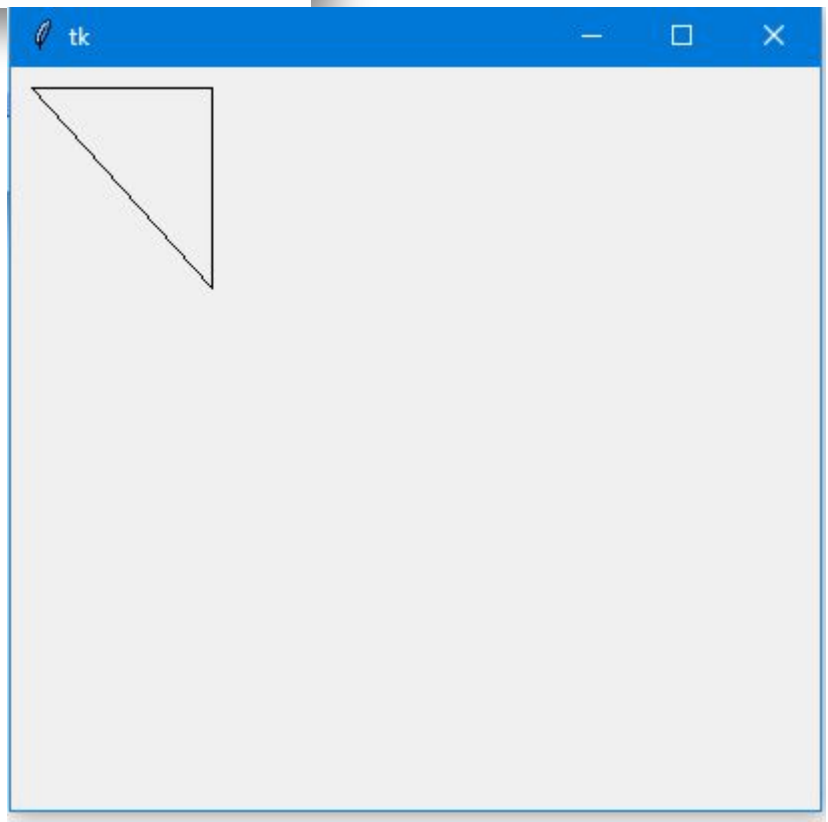


```python
from tkinter import *


tk = Tk()
canvas = Canvas(tk, width=400, height=400)
canvas.pack()
canvas.create_arc(10, 10, 200, 80, extent=45, style=ARC)
canvas.create_arc(10, 80, 200, 160, extent=90, style=ARC)
canvas.create_arc(10, 160, 200, 240, extent=135, style=ARC)
canvas.create_arc(10, 240, 200, 320, extent=180, style=ARC)
canvas.create_arc(10, 320, 200, 400, extent=359, style=ARC)
```

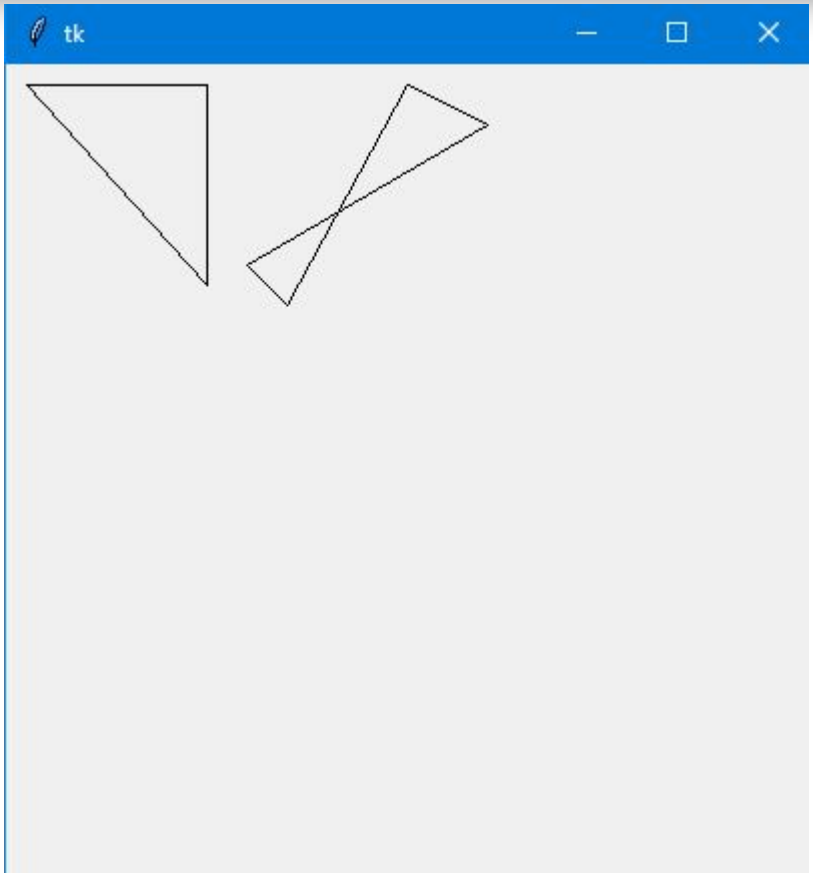


Рисование многоугольников

```
1  from tkinter import *
2
3
4  tk = Tk()
5  canvas = Canvas(tk, width=400, height=400)
6  canvas.pack()
7
8  canvas.create_polygon(10, 10, 100, 10, 100, 110, fill="",
9  outline="black")
```

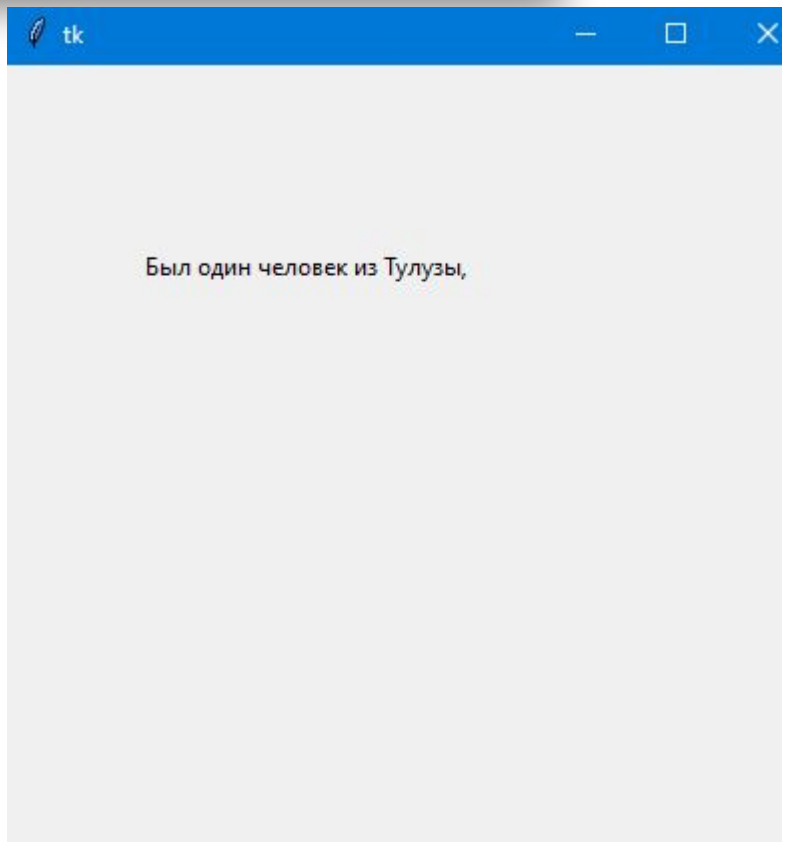


```python
from tkinter import *


tk = Tk()
canvas = Canvas(tk, width=400, height=400)
canvas.pack()

canvas.create_polygon(10, 10, 100, 10, 100, 110, fill="", outline="black")
canvas.create_polygon(200, 10, 240, 30, 120, 100, 140, 120,fill="", outline="black")
```

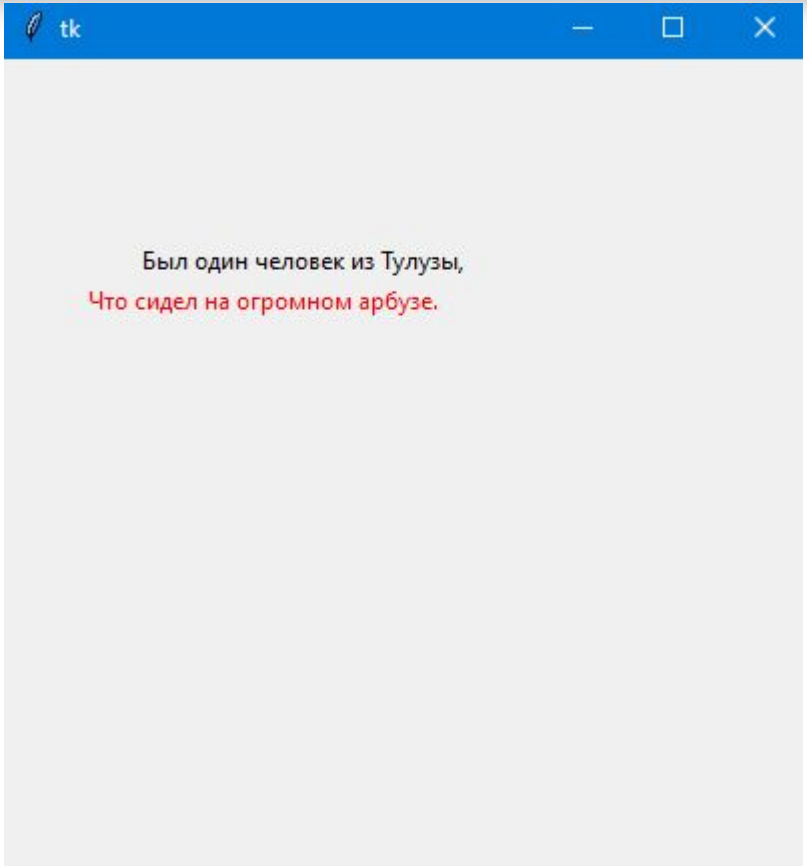


Отображение текста

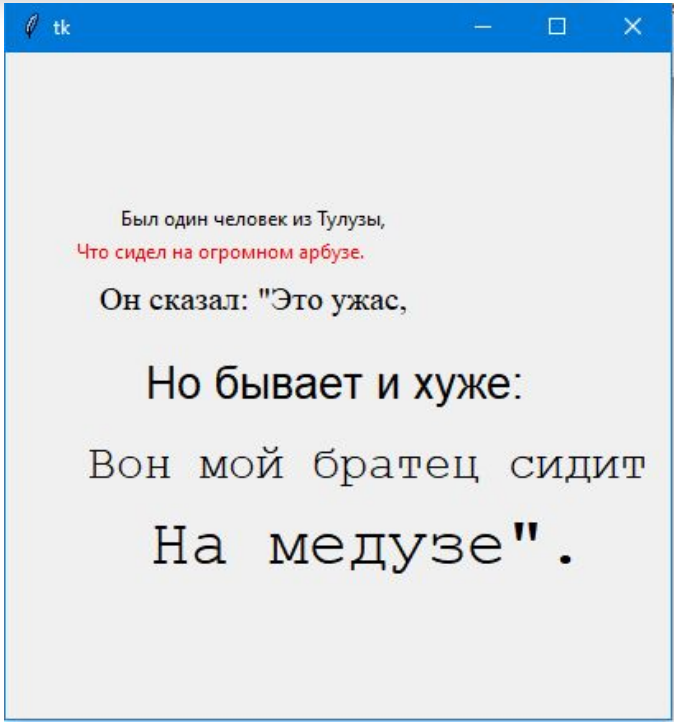
```
1  from tkinter import *
2
3
4  tk = Tk()
5  canvas = Canvas(tk, width=400, height=400)
6  canvas.pack()
7
8  canvas.create_text(150, 100, text='Был один человек из Тулузы,')
```



```python
from tkinter import *


tk = Tk()
canvas = Canvas(tk, width=400, height=400)
canvas.pack()

canvas.create_text(150, 100, text='Был один человек из Тулузы,')
canvas.create_text(130, 120, text='Что сидел на огромном арбузе.',fill='red')
```



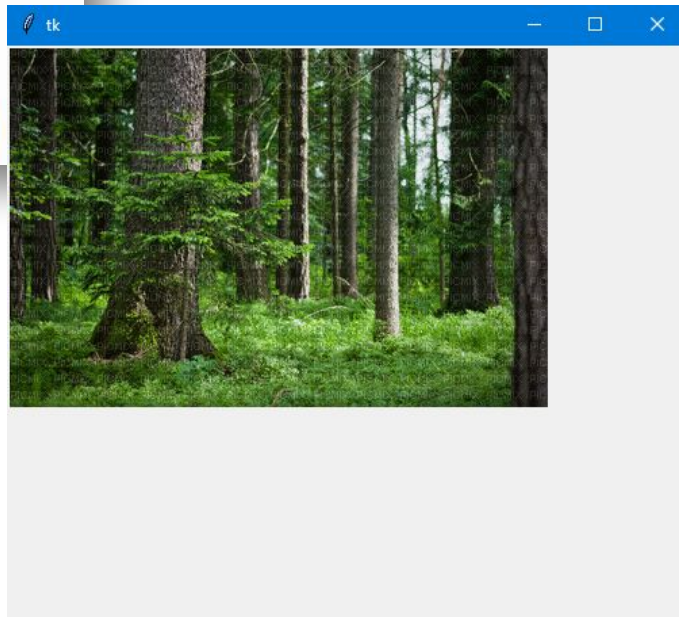
```python
from tkinter import *


tk = Tk()
canvas = Canvas(tk, width=400, height=400)
canvas.pack()

canvas.create_text(150, 100, text='Был один человек из Тулузы,')
canvas.create_text(130, 120, text='Что сидел на огромном арбузе.',fill='red')
canvas.create_text(150, 150, text='Он сказал: "Это ужас,',font=('Times', 15))
canvas.create_text(200, 200, text='Но бывает и хуже:',font=('Helvetica', 20))
canvas.create_text(220, 250, text='Вон мой братец сидит',font=('Courier', 22))
canvas.create_text(220, 300, text='На медузе".', font=('Courier',30))
```



Вывод изображений

Средствами `tkinter` *можно загружать только изображения формата GIF (файлы с расширением .gif). Чтобы вывести изображение другого типа, скажем, PNG (.png) или JPG (.jpg), придется воспользоваться другим модулем, например Python Imaging Library (http://www.pythonware.com/products/pil/).*

```
1  from tkinter import *
2
3
4  tk = Tk()
5  canvas = Canvas(tk, width=500, height=500)
6  canvas.pack()
7
8  my_image = PhotoImage(file='C:\\1220961_233e7.gif')
9  canvas.create_image(0, 0, anchor=NW, image=my_image)
```



Создание простой анимации

```
1  import time
2  from tkinter import *
3  tk = Tk()
4  canvas = Canvas(tk, width=400, height=200)
5  canvas.pack()
6  canvas.create_polygon(10, 10, 10, 60, 50, 35)
7  for x in range(0, 60):
8      canvas.move(1, 5, 0)
9      tk.update()
10     time.sleep(0.05)
11
```

```
1  import time
2  from tkinter import *
3  tk = Tk()
4  canvas = Canvas(tk, width=400, height=400)
5  canvas.pack()
6  canvas.create_polygon(10, 10, 10, 60, 50, 35)
7  for x in range(0, 60):
8      canvas.move(1, 5, 5)
9      tk.update()
10     time.sleep(0.05)
```

```
import time
from tkinter import *
tk = Tk()
canvas = Canvas(tk, width=400, height=400)
canvas.pack()
canvas.create_polygon(10, 10, 10, 60, 50, 35)
for x in range(0, 60):
    canvas.move(1, 5, 5)
    tk.update()
    time.sleep(0.05)

for x in range(0, 60):
    canvas.move(1, -5, -5)
    tk.update()
    time.sleep(0.05)
```

Реакция объектов на события

```
from tkinter import *
tk = Tk()
canvas = Canvas(tk, width=400, height=400)
canvas.pack()
canvas.create_polygon(10, 10, 10, 60, 50, 35)
def movetriangle(event):
    canvas.move(1, 5, 0)
canvas.bind_all('<KeyPress-Return>', movetriangle)
```

```
from tkinter import *
tk = Tk()
canvas = Canvas(tk, width=400, height=400)
canvas.pack()
canvas.create_polygon(10, 10, 10, 60, 50, 35)
def movetriangle(event):
    if event.keysym == 'Up':
        canvas.move(1, 0, -3)
    elif event.keysym == 'Down':
        canvas.move(1, 0, 3)
    elif event.keysym == 'Left':
        canvas.move(1, -3, 0)
    else:
        canvas.move(1, 3, 0)
canvas.bind_all('<KeyPress-Up>', movetriangle)
canvas.bind_all('<KeyPress-Down>', movetriangle)
canvas.bind_all('<KeyPress-Left>', movetriangle)
canvas.bind_all('<KeyPress-Right>', movetriangle)
```