



алгоритмика

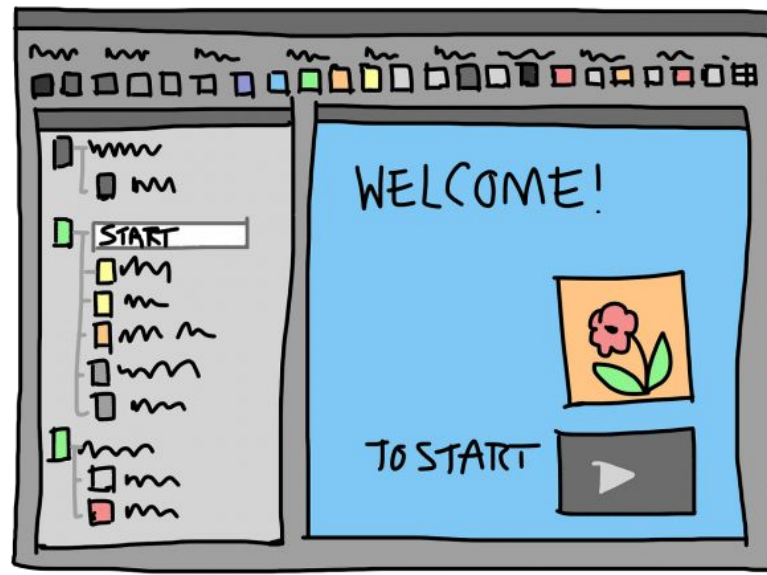
# PYTHON

Занятие 11

# Сегодня на уроке

---

- Классы и объекты
- Работа с кнопками
- Создание функций
- Первое мини-приложение



# Вопрос

---

1. Как подключается модуль tkinter?
2. Что такое объект?
3. Что такое метод?
4. Что такое класс?
5. Как создается окно приложения?
6. Как закрепляется объект на окне?
7. Для чего используется метод mainloop?

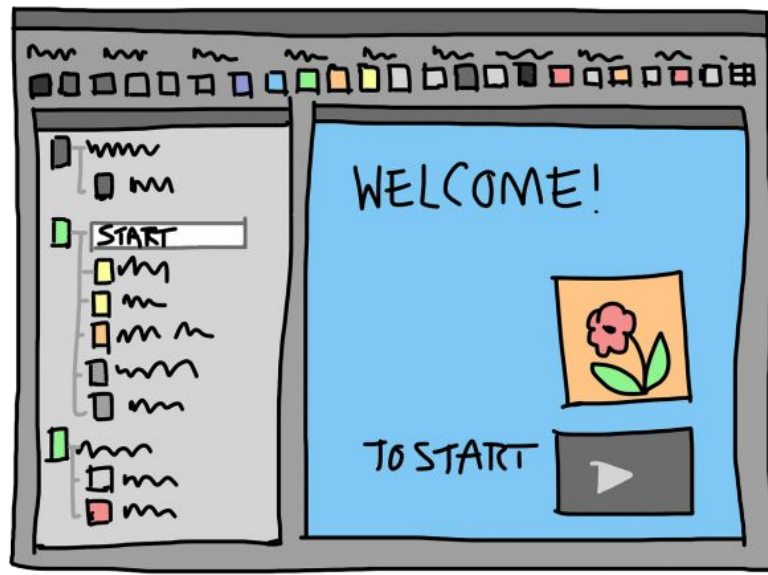
# Модуль tkinter

---

**tkinter** – **tool**kit **inter**face – инструментарий для создания пользовательского интерфейса

Подключение модуля:

```
from tkinter import*
```



# Класс в Python

---

**Класс** – это тип объекта, который мы создаем в программе

**root** = **Имя\_класса()**

имя объекта

класс объекта

**Tk()** – базовый класс модуля. При создании объекта создается окно приложения.

**Canvas()** - позволяет располагать на самом себе другие графические объекты.



# Объект и метод

---

Обращение к методу происходит через точку - .

Вызов метода возможен **ТОЛЬКО** после создания объекта.

Имя\_объекта.Метод

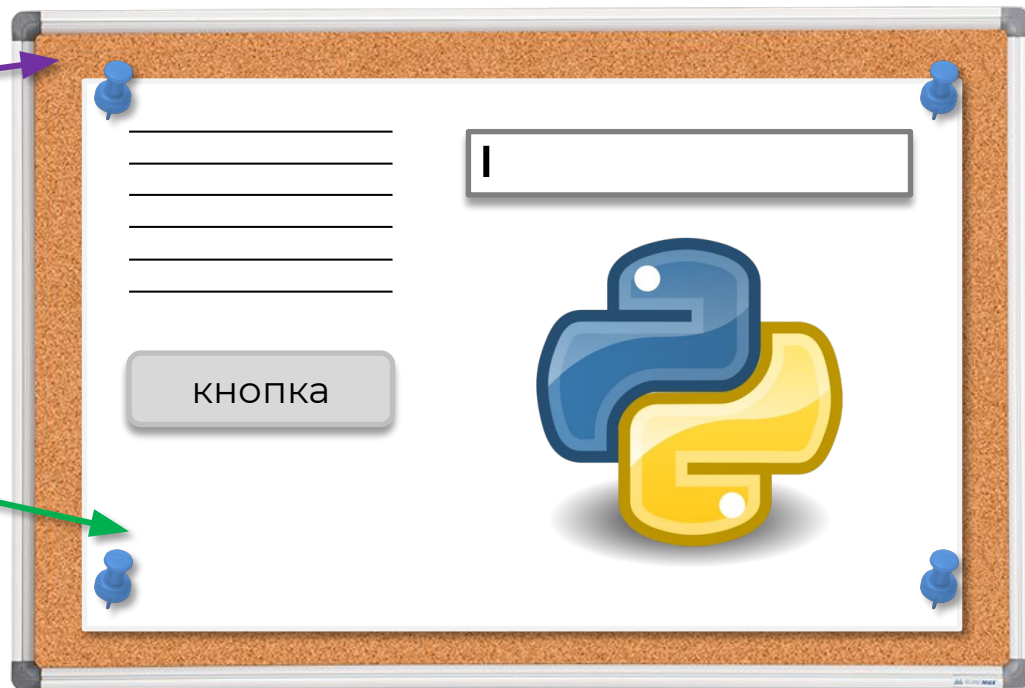
root.mainloop()

# Элементы интерфейса

---

**Окно** – основа интерфейса

**Холст** – основа для графических элементов

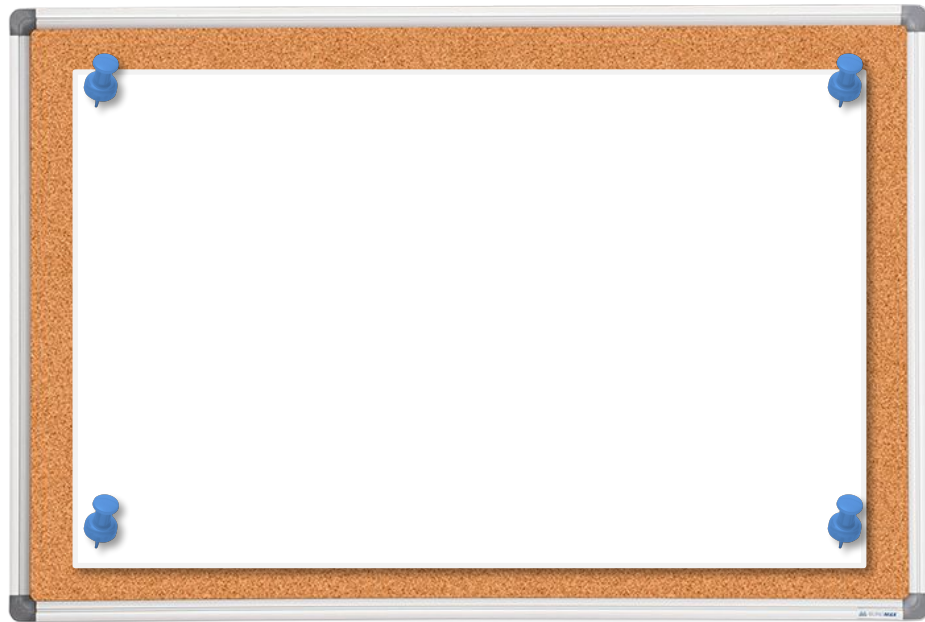


# Закрепление элемента

---

**pack()** – метод для закрепления объекта

**canvas.pack()**





# Вопрос

---

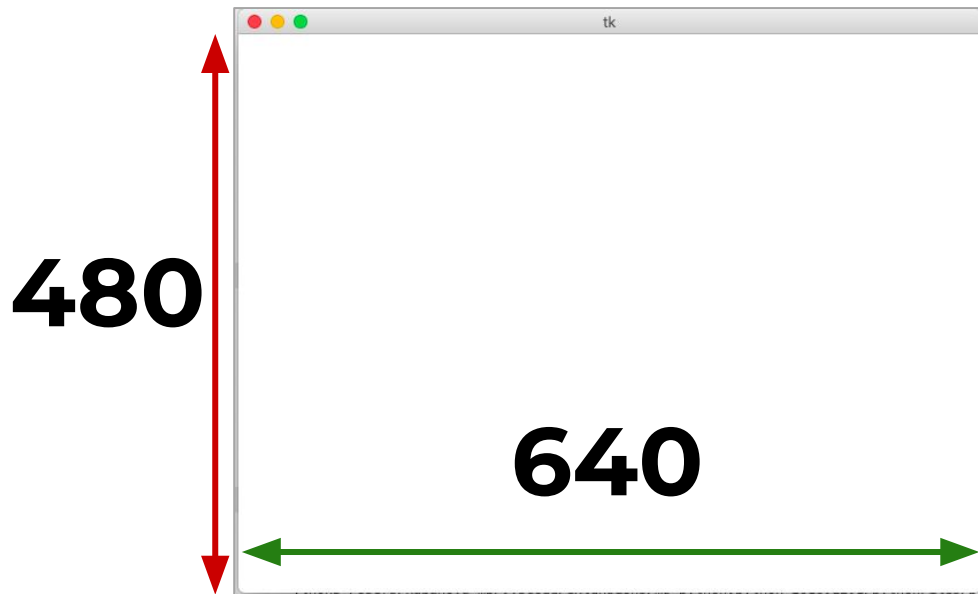
Опишите работу следующей программы:

```
from tkinter import*  
root=Tk()  
canvas = Canvas(root,width=640,height=480)  
canvas.pack()  
root.mainloop()
```

# Вопрос-ответ

---

```
from tkinter import*  
root=Tk()  
canvas = Canvas(root,width=640,height=480)  
canvas.pack()  
root.mainloop()
```



# Вопрос

---

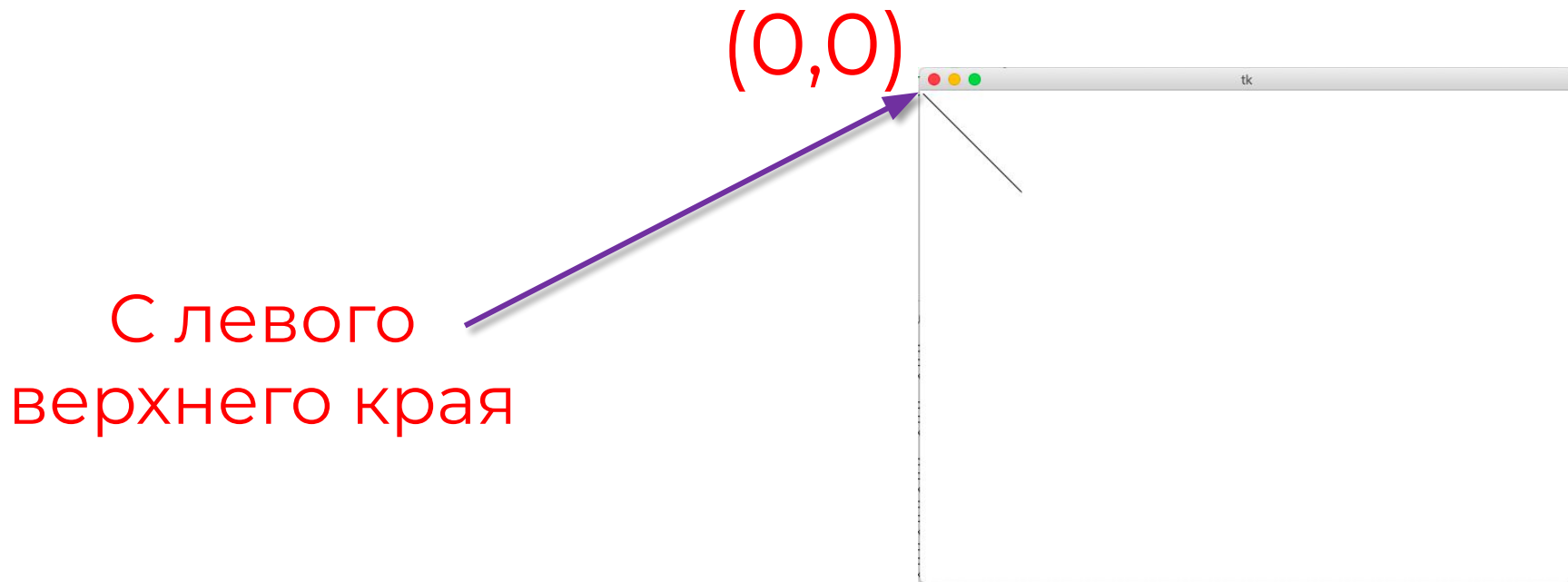
Откуда начинается отсчет координат в окне tkinter?



# Вопрос

---

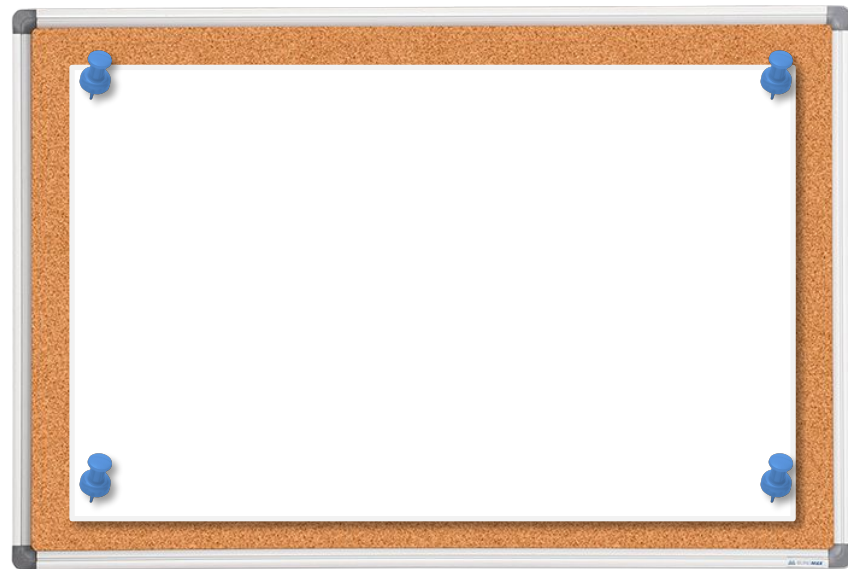
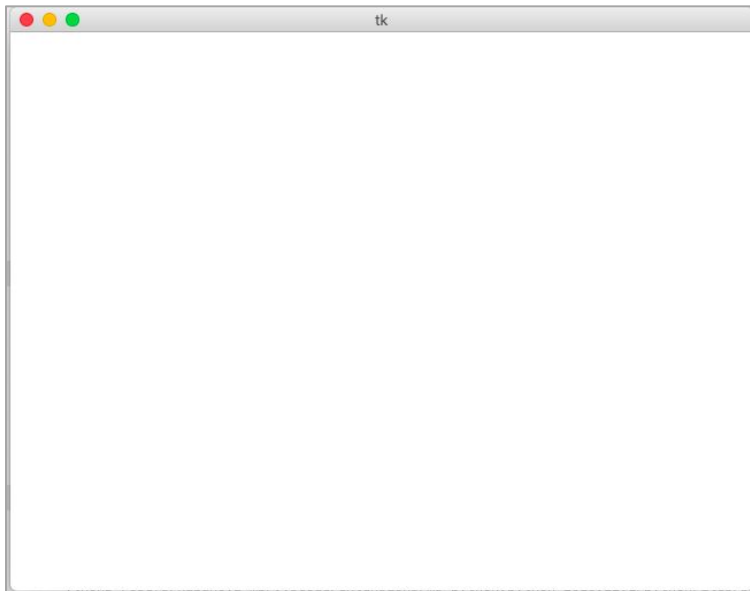
Откуда начинается отсчет координат в окне tkinter?



# Размещение холста

---

Холст занимает все пространство окна. Но что, если холст должен занимать определенное место в окне?



# Задание размера окна

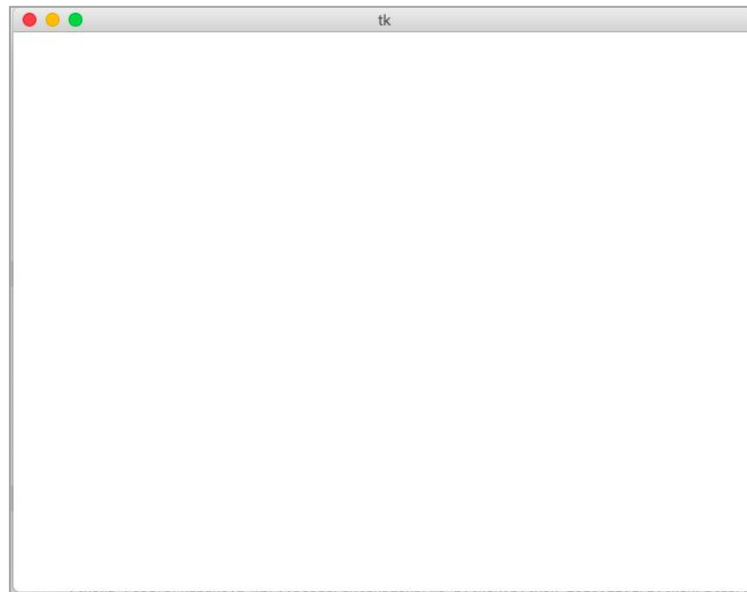
---

**geometry** - метод класса Tk(), который позволяет задать размер окна. **Размер окна должен быть больше размера холста.**

```
.geometry('ширинаxвысота')
```

```
root=Tk()
```

```
root.geometry('800x800')
```

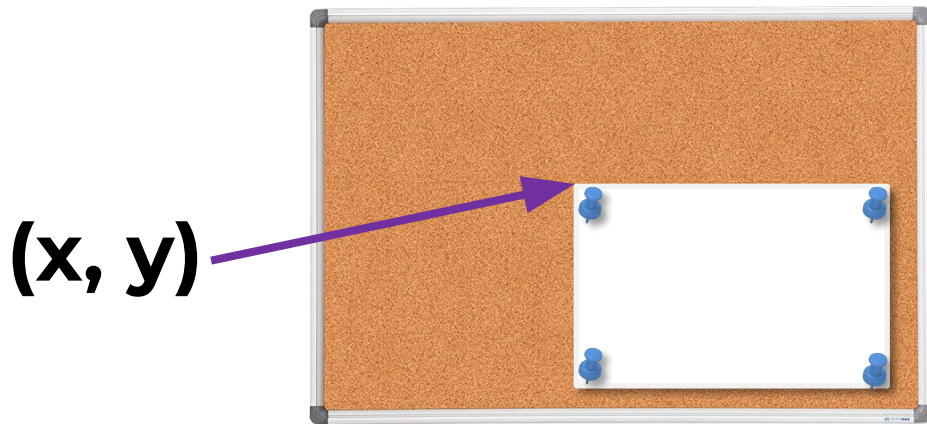


# Перемещение холста

---

**place** - метод, позволяющий закрепить объект в указанных координатах.

объект.place (x=100, y=100)



# Задание

---

Составьте и запустите следующую программу. В каком месте появится квадрат?

```
from tkinter import*  
root=Tk()  
root.geometry('800x800')  
canvas = Canvas(root,width=640,height=480)  
canvas.pack()  
canvas.place(x=100,y=100)  
canvas.create_rectangle(5,5,640,480)  
root.mainloop()
```



# Задание. Решение

Составьте и запустите следующую программу. В каком месте появится квадрат?

```
from tkinter import*
```

```
root=Tk()
```

```
root.geometry('800x800')
```

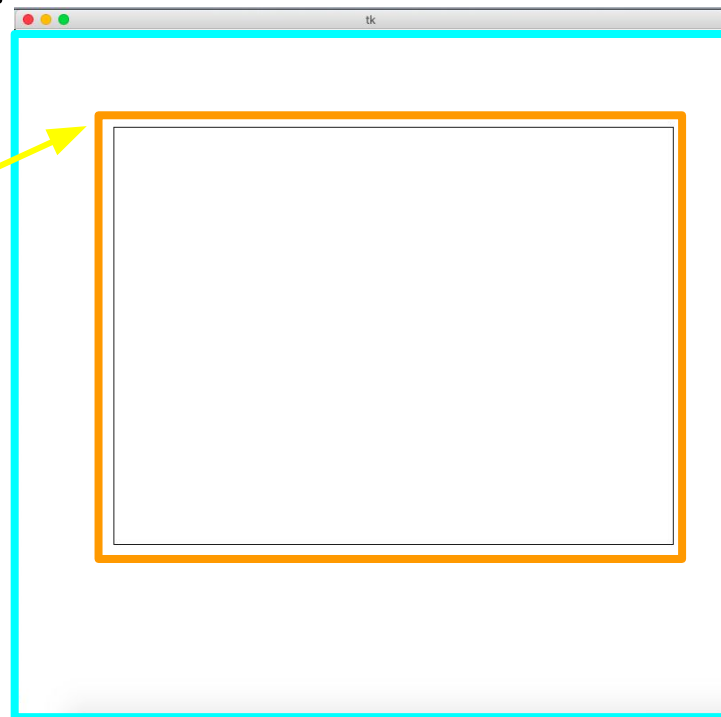
```
canvas = Canvas(root,width=640,height=480)
```

```
canvas.pack()
```

```
canvas.place(x=100,y=100)
```

```
canvas.create_rectangle(5,5,640,480)
```

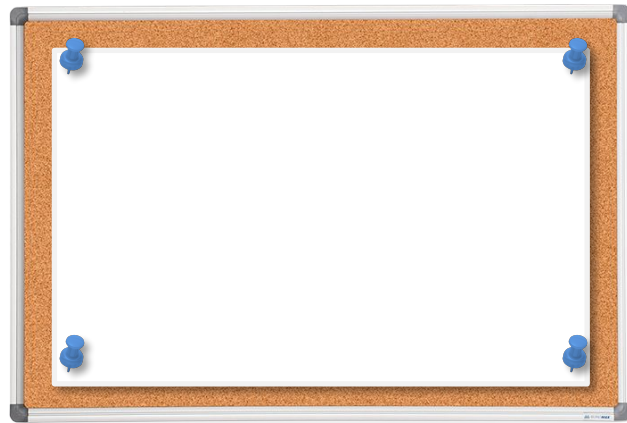
```
root.mainloop()
```



# Вопрос



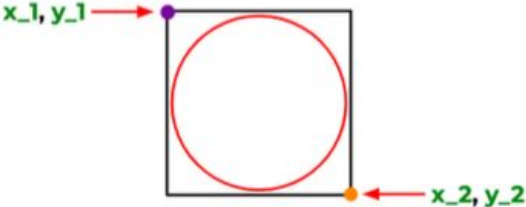
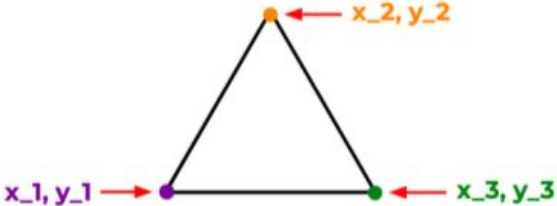
---

Какие фигуры мы научились создавать на холсте? С помощью каких методов объекта Canvas мы это делали?



# Вопрос-ответ

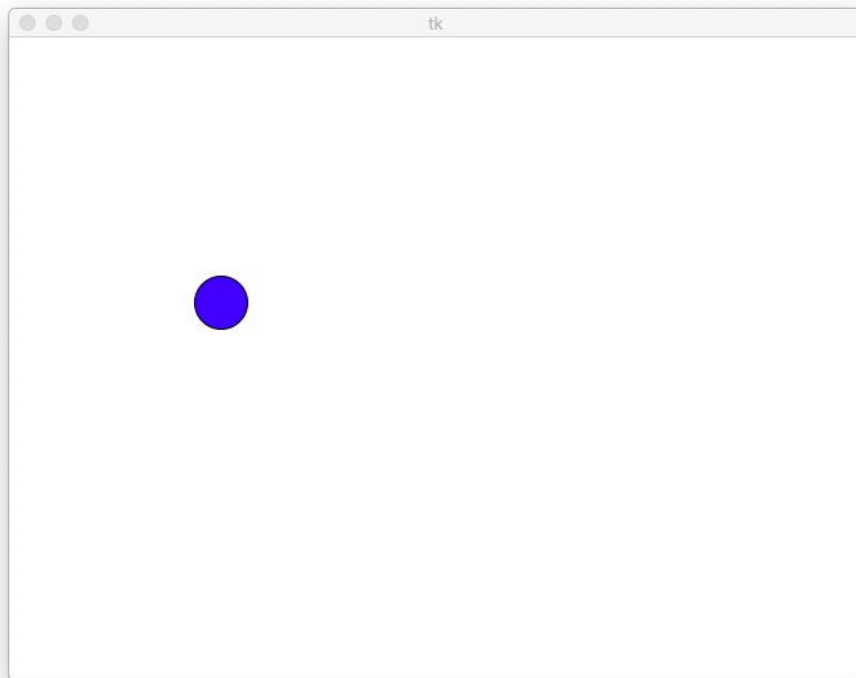
Какие фигуры мы научились создавать на холсте? С помощью каких методов объекта Canvas мы это делали?

Метод	Фигура
<code>create_line(x_1, y_1, x_2, y_2)</code>	
<code>create_rectangle(x_1, y_1, x_2, y_2)</code>	
<code>create_oval(x_1, y_1, x_2, y_2)</code>	
<code>create_polygon(x_1, y_1, x_2, y_2, x_3, y_3, ...)</code>	

# Вопрос

---

Как закрасить фигуру цветом?

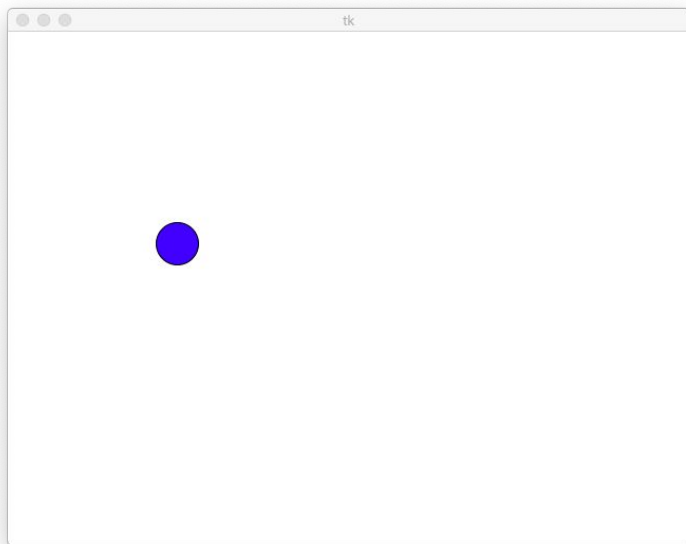


# Вопрос-ответ

---

Как закрасить фигуру цветом?

```
canvas.create_oval(x_1, y_1, x_2, y_2, fill="color")
```

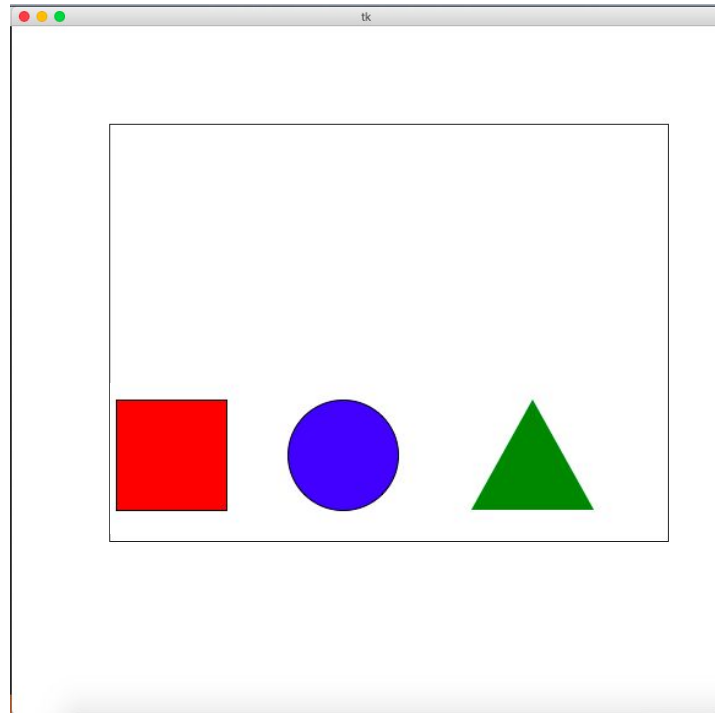


**Указать fill="цвет" при  
создании фигуры**

# Задание на повторение

---

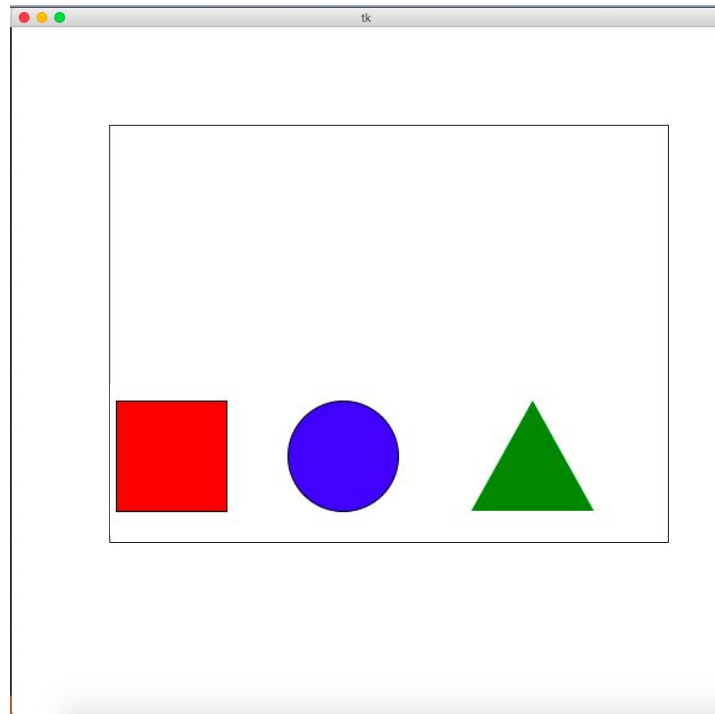
На листе определите координаты фигур и напишите программу для их создания в созданном окне.



# Вопрос

---

Умеем ли мы контролировать порядок появления объектов в окне?

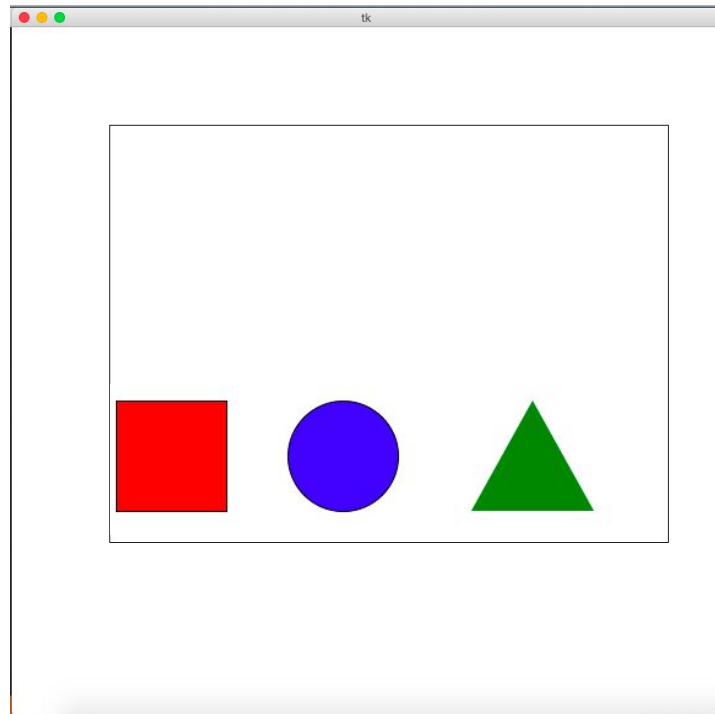


# Вопрос-ответ

---

Умеем ли мы контролировать порядок появления объектов в окне?

**Нет, все фигуры  
появляются  
одновременно**

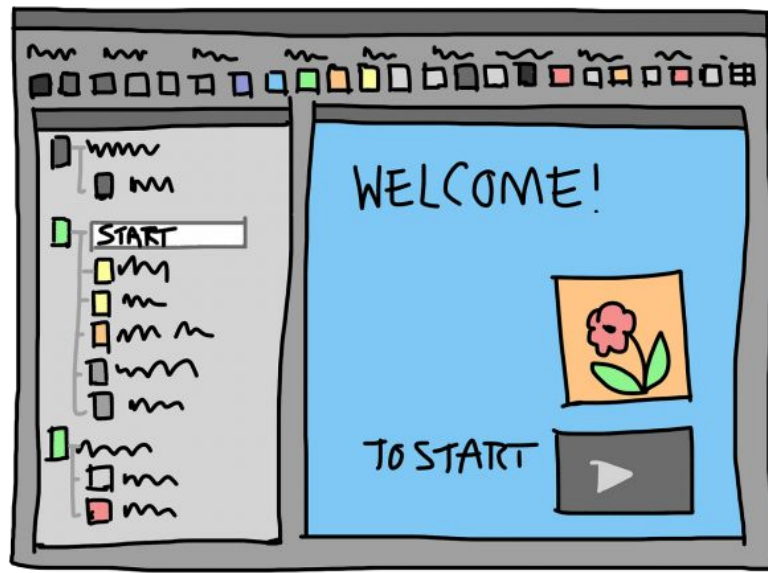




# Вопрос

---

Зачем нужны кнопки в приложениях?

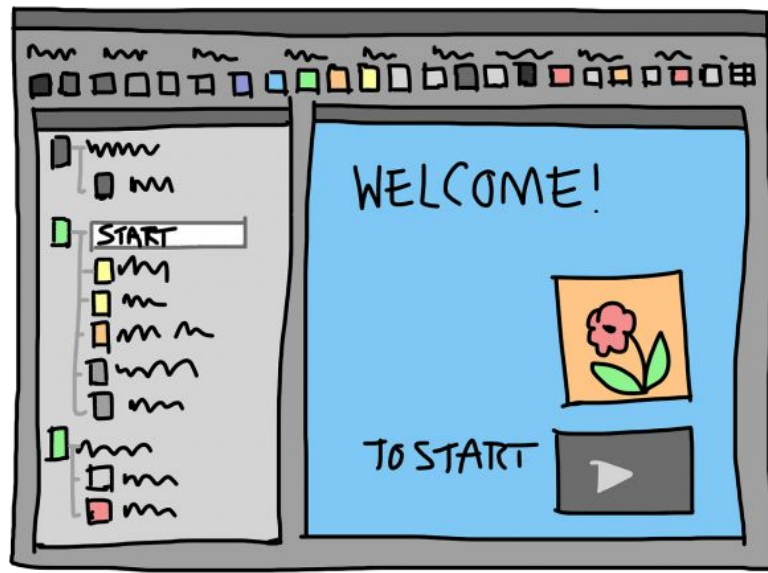


# Вопрос-ответ

---

Зачем нужны кнопки в приложениях?

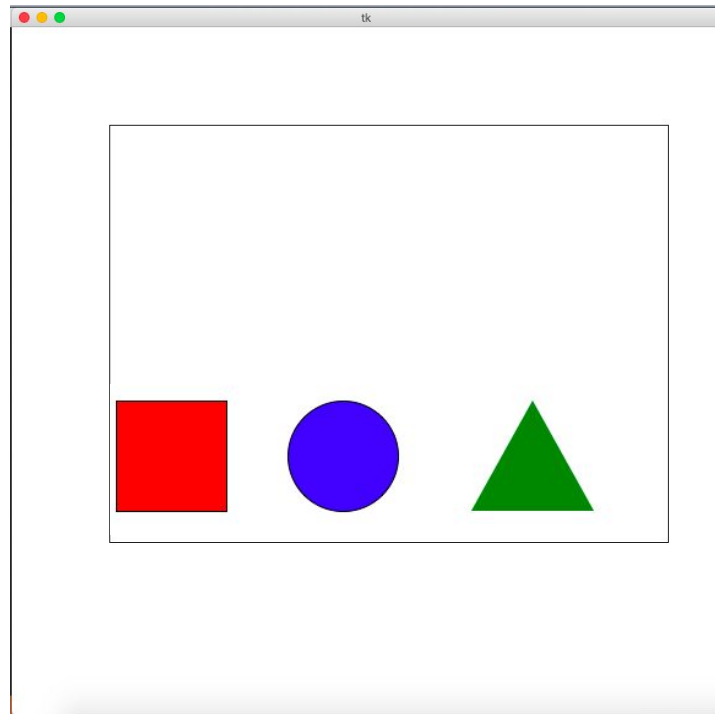
- Для подтверждения действия;
- Для выполнения действий.



# Вопрос

---

Для чего мы можем использовать кнопки в своем приложении?

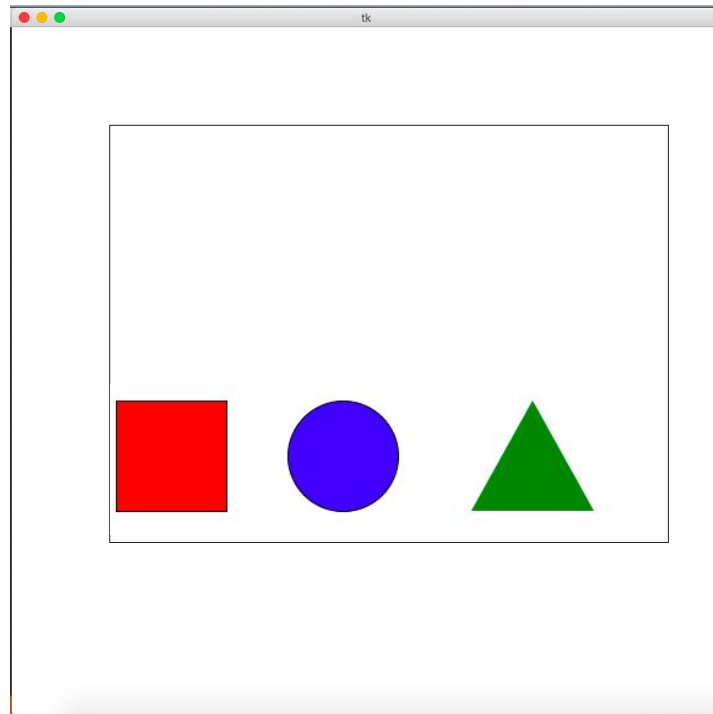


# Вопрос-ответ

---

Для чего мы можем использовать кнопки в своем приложении?

**Для создания  
определенной фигуры**



# Кнопка в tkinter

---

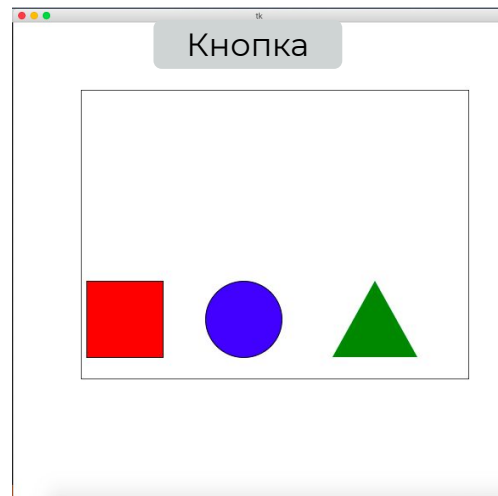
Создание кнопки:

```
btn_1 = Button(root, text="Кнопка")
```

```
Имя_кнопки = Класс_кнопок(имя_окна, text="Надпись_на_кнопке")
```



Кнопка

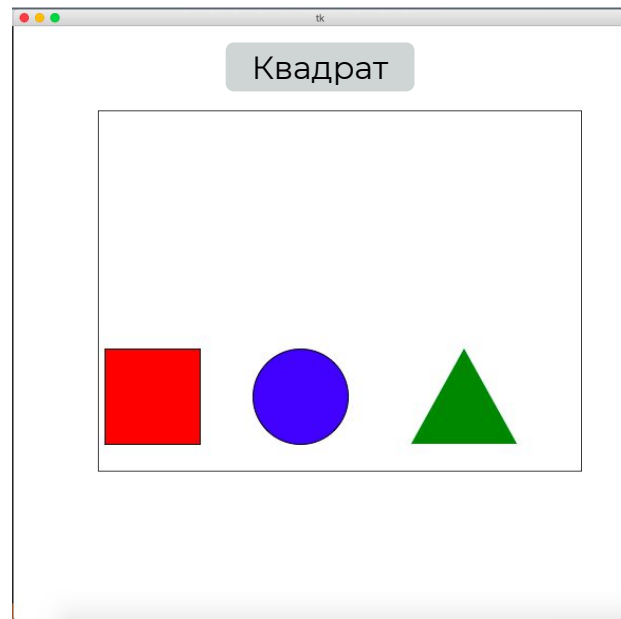


# Задание

---

Добавьте в программу создание кнопки с надписью **Квадрат**

```
btn_1 = Button(root,text="Кнопка")
```



# Задание. Решение

Добавьте в программу создание кнопки с надписью **Квадрат**

```
from tkinter import*
```

```
root=Tk()
```

```
root.geometry('800x800')
```

```
btn_1 = Button(root,text="Квадрат")
```

```
btn_1.pack()
```

```
canvas = Canvas(root,width=640,height=480)
```

```
canvas.pack()
```

```
canvas.place(x=100,y=100)
```

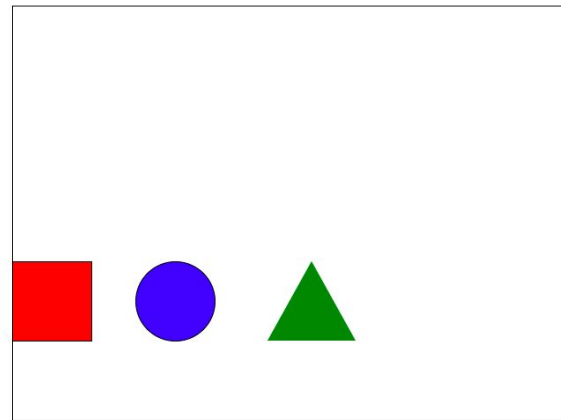
```
canvas.create_rectangle(5,5,640,480)
```

```
canvas.create_rectangle(10,300,100,390,fill="red")
```

```
canvas.create_oval(150,300,240,390,fill="blue")
```

```
canvas.create_polygon(300,390,350,300,400,390,fill="green")
```

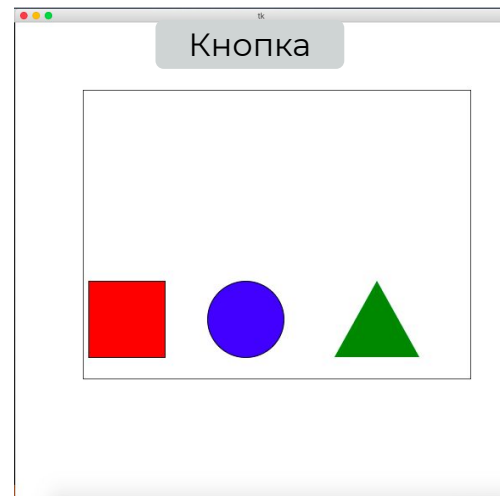
```
root.mainloop()
```



# Вопрос

---

Что происходит, когда мы нажимаем на кнопку?



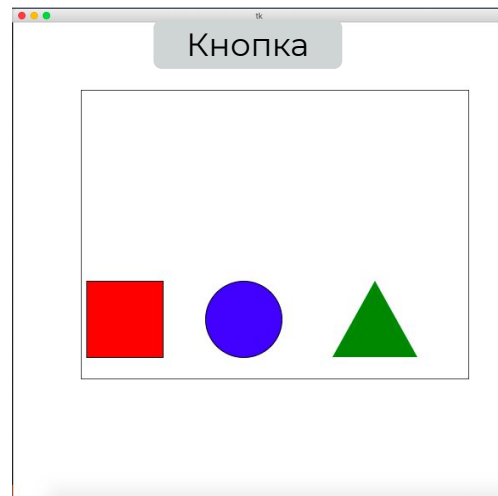


# Вопрос-ответ

---

Что происходит, когда мы нажимаем на кнопку?

**Ничего**



# Событие

---

Событие – это ответное действие, которое происходит после действия пользователя.

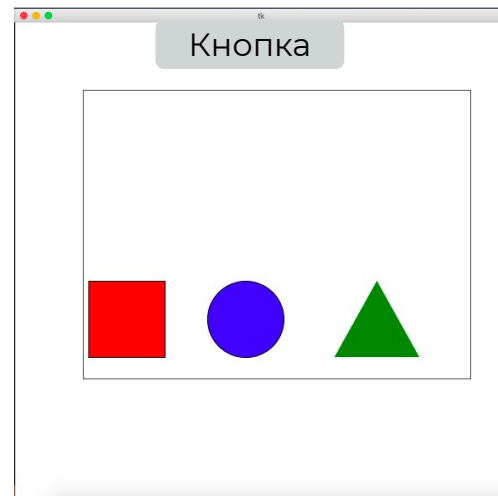
**Действие**

Пользователь нажал кнопку



**Событие**

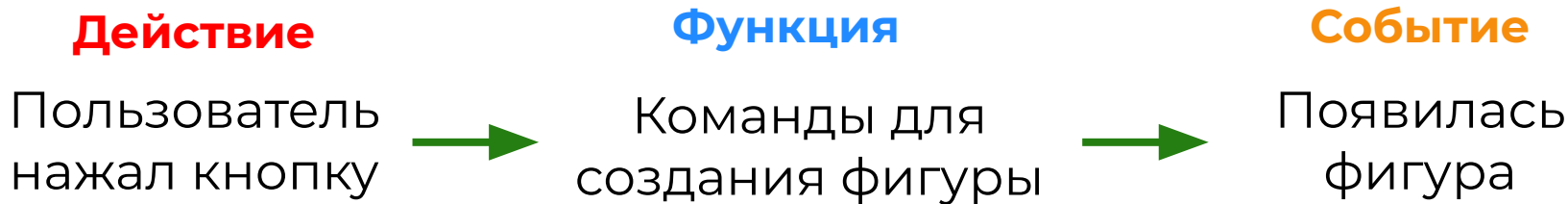
Появилась фигура



# Функция

---

Функция – это правило, которое описывает событие. Функция вызывается столько раз, сколько раз совершит действие пользователь.



# Функция в Python

---

**def** имя\_функции **(event):**

↔ тело функции

**Tab**

**def – definition** – правило. Сокращение, с которого начинается каждая функция

имя\_функции – отражает назначение функции (печать, квадрат и т.п.)

**(event)** – обозначает ожидание события.



# Вопрос

---

Как же программе понять, какую функцию вызывать после нажатия кнопки?

Кнопка



Функция  
kvadrat

Функция  
polygon

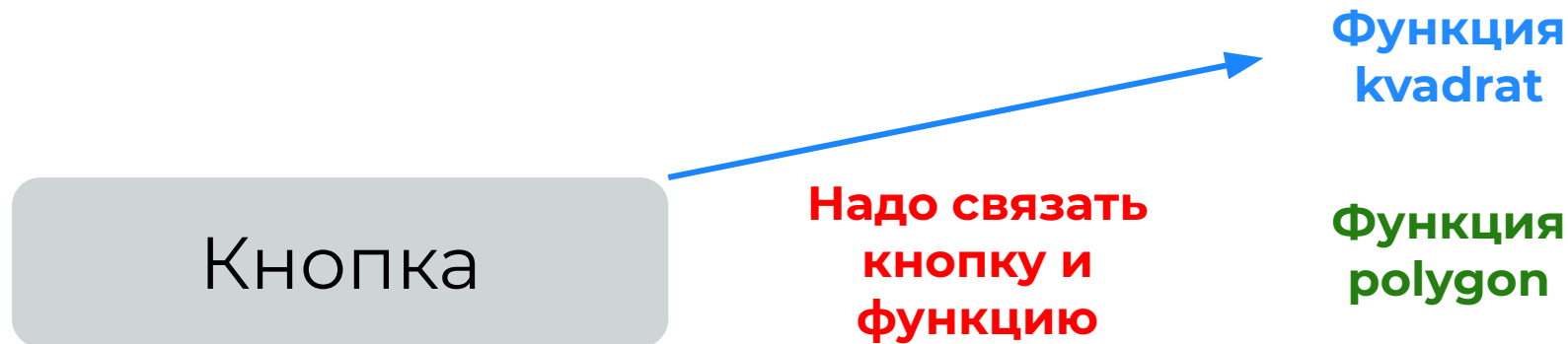
Функция  
oval



# Вопрос-ответ

---

Как же программе понять, какую функцию вызывать после нажатия кнопки?



# Связка кнопки и функции

---

Метод bind – связывает виджет и функцию.

виджет.bind("<действие\_пользователя>", имя\_функции)

## Действие пользователя:

- <Button-1> - щелчок левой кнопкой мыши
- <Button-2> - щелчок средней кнопкой мыши
- <Button-3> - щелчок правой кнопкой мыши
- <Double-Button-1> - двойной клик левой кнопкой мыши
- <Motion> - движение мыши

# Вопрос

---

Что изменилось в нашей программе?

```
from tkinter import*
root=Tk()
root.geometry('800x800')

def kvadrat(event):
    canvas.create_rectangle(10,300,100,390,fill="red")

btn_1 = Button(root,text="Квадрат")
btn_1.pack()
btn_1.bind("<Button-1>",kvadrat)

canvas=Canvas(root,width=640,height=480)
canvas.pack()
canvas.place(x=100,y=100)
canvas.create_rectangle(5,5,640,480)
canvas.create_oval(150,300,240,390,fill="blue")
canvas.create_polygon(300,390,350,300,400,390,fill="green")

root.mainloop()
```





# Вопрос-ответ

Что изменилось в нашей программе?

```
from tkinter import*
```

```
root=Tk()
```

```
root.geometry('800x800')
```

```
def kvadrat(event):
```

```
    canvas.create_rectangle(10,300,100,390,fill="red")
```

```
btn_1 = Button(root,text="Квадрат")
```

```
btn_1.pack()
```

```
btn_1.bind("<Button-1>",kvadrat)
```

```
canvas=Canvas(root,width=640,height=480)
```

```
canvas.pack()
```

```
canvas.place(x=100,y=100)
```

```
canvas.create_rectangle(5,5,640,480)
```

```
canvas.create_oval(150,300,240,390,fill="blue")
```

```
canvas.create_polygon(300,390,350,300,400,390,fill="green")
```

```
root.mainloop()
```



# Вопрос

---

Что мы увидим на экране, после запуска программы?

```
from tkinter import*
root=Tk()
root.geometry('800x800')

def kvadrat(event):
    canvas.create_rectangle(10,300,100,390,fill="red")

btn_1 = Button(root,text="Квадрат")
btn_1.pack()
btn_1.bind("<Button-1>",kvadrat)

canvas=Canvas(root,width=640,height=480)
canvas.pack()
canvas.place(x=100,y=100)
canvas.create_rectangle(5,5,640,480)
canvas.create_oval(150,300,240,390,fill="blue")
canvas.create_polygon(300,390,350,300,400,390,fill="green")

root.mainloop()
```



# Вопрос-ответ

Что мы увидим на экране, после запуска программы?

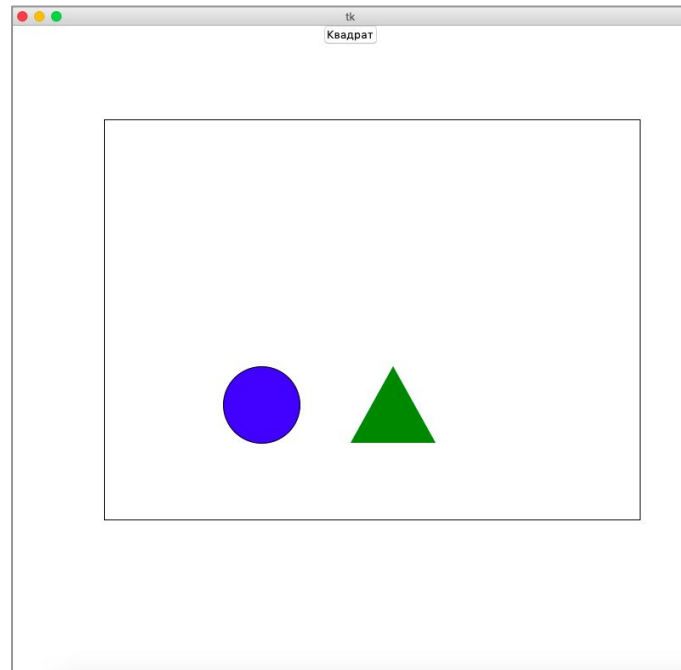
```
from tkinter import*
root=Tk()
root.geometry('800x800')

def kvadrat(event):
    canvas.create_rectangle(10,300,100,390,fill="red")

btn_1 = Button(root,text="Квадрат")
btn_1.pack()
btn_1.bind("<Button-1>",kvadrat)

canvas=Canvas(root,width=640,height=480)
canvas.pack()
canvas.place(x=100,y=100)
canvas.create_rectangle(5,5,640,480)
canvas.create_oval(150,300,240,390,fill="blue")
canvas.create_polygon(300,390,350,300,400,390,fill="green")

root.mainloop()
```



# Вопрос

Что необходимо сделать пользователю, чтобы появился квадрат?

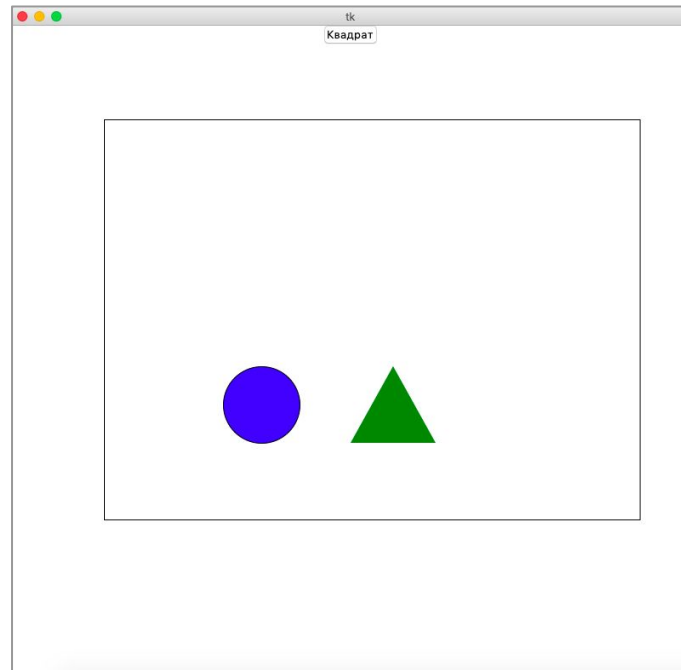
```
from tkinter import*
root=Tk()
root.geometry('800x800')

def kvadrat(event):
    canvas.create_rectangle(10,300,100,390,fill="red")

btn_1 = Button(root,text="Квадрат")
btn_1.pack()
btn_1.bind("<Button-1>",kvadrat)

canvas=Canvas(root,width=640,height=480)
canvas.pack()
canvas.place(x=100,y=100)
canvas.create_rectangle(5,5,640,480)
canvas.create_oval(150,300,240,390,fill="blue")
canvas.create_polygon(300,390,350,300,400,390,fill="green")

root.mainloop()
```



# Вопрос-ответ

Что необходимо сделать пользователю, чтобы появился квадрат?

```
from tkinter import*
root=Tk()
root.geometry('800x800')

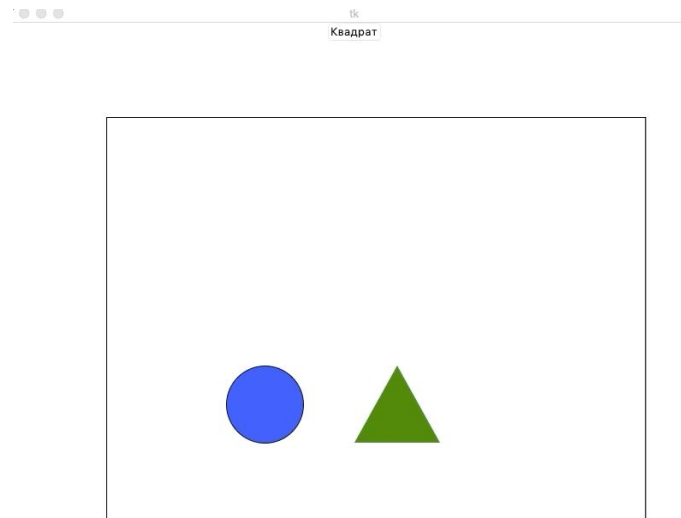
def kvadrat(event):
    canvas.create_rectangle(10,300,100,390,fill="red")

btn_1 = Button(root,text="Квадрат")
btn_1.pack()
btn_1.bind("<Button-1>", kvadrat)

canvas=Canvas(root,width=640,height=480)
canvas.pack()
canvas.place(x=100,y=100)
canvas.create_rectangle(5,5,640,480)
canvas.create_oval(150,300,240,390,fill="blue")
canvas.create_polygon(300,390,350,300,400,390,fill="green")

root.mainloop()
```

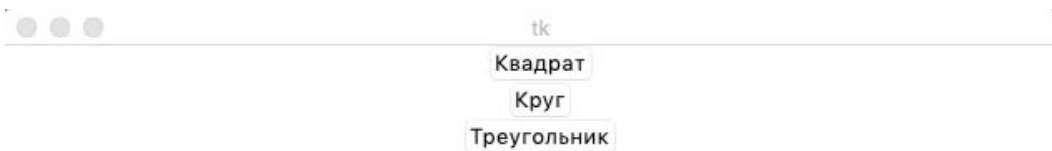
**Нажать левой кнопкой  
мыши на кнопку Квадрат**



# Задание

---

Добавьте в программу функции `oval` и `polygon` для появления фигур круга и треугольника после нажатия соответствующие кнопки.



# Задание. Решение

Добавьте в программу функции `oval` и `polygon` для появления фигур круга и треугольника после нажатия соответствующие кнопки.

```
from tkinter import*
```

```
root=Tk()
```

```
root.geometry('800x800')
```

```
def kvadrat(event):
```

```
    canvas.create_rectangle(10,300,100,390,fill="red")
```

```
def oval(event):
```

```
    canvas.create_oval(150,300,240,390,fill="blue")
```

```
def polygon(event):
```

```
    canvas.create_polygon(300,390,350,300,400,390,fill="green")
```

```
btn_1 = Button(root,text="Квадрат")
```

```
btn_1.pack()
```

```
btn_1.bind("<Button-1>",kvadrat)
```

```
btn_2 = Button(root,text="Круг")
```

```
btn_2.pack()
```

```
btn_2.bind("<Button-1>",oval)
```

```
btn_3 = Button(root,text="Треугольник")
```

```
btn_3.pack()
```

```
btn_3.bind("<Button-1>",polygon)
```

```
canvas=Canvas(root,width=640,height=480)
```

```
canvas.pack()
```

```
canvas.place(x=100,y=100)
```

```
canvas.create_rectangle(5,5,640,480)
```

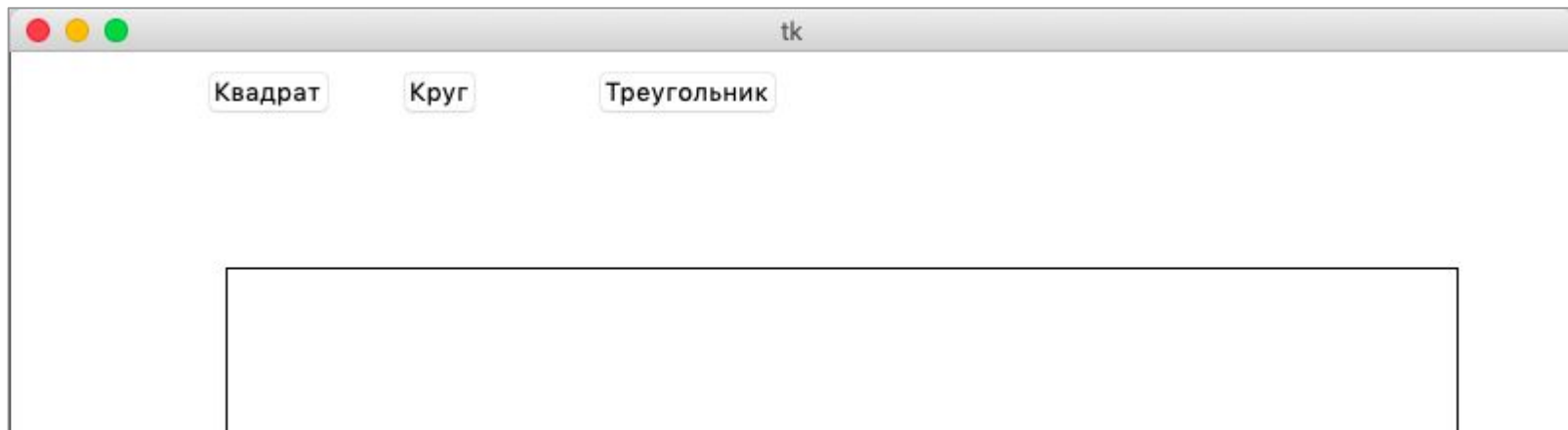
```
root.mainloop()
```



# Задание

---

Используйте метод `place` для кнопок и расположите их в ряд, как показано на рисунке





# Задание. Решение

---

Используйте метод `place` для кнопок и расположите их в ряд, как показано на рисунке

```
btn_1 = Button(root,text="Квадрат")
```

```
btn_1.pack()
```

```
btn_1.place(x=100,y=10)
```

```
btn_1.bind("<Button-1>",kvadrat)
```

```
btn_2 = Button(root,text="Круг")
```

```
btn_2.pack()
```

```
btn_2.place(x=200,y=10)
```

```
btn_2.bind("<Button-1>",oval)
```

```
btn_3 = Button(root,text="Треугольник")
```

```
btn_3.pack()
```

```
btn_3.place(x=300,y=10)
```

```
btn_3.bind("<Button-1>",polygon)
```



# Вопросы

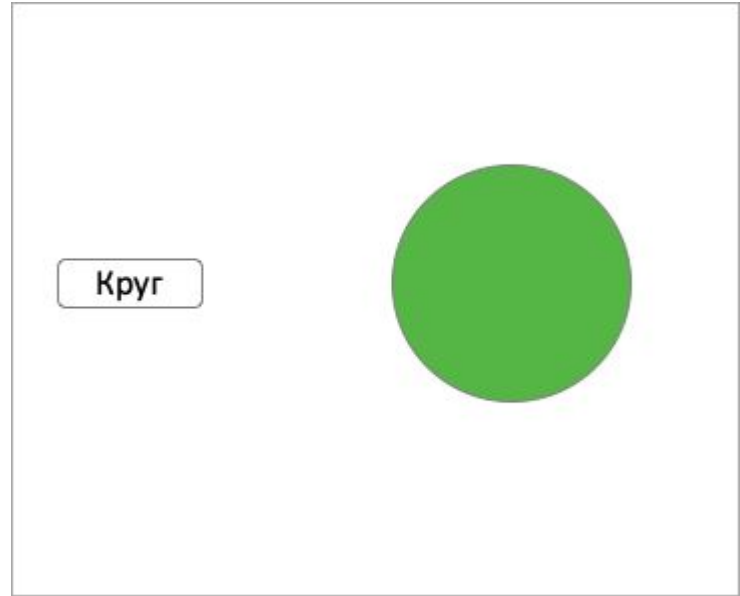
---

1. Как изменить размер окна?
2. Что такое виджет?
3. Как работает метод place?
4. Какой класс работает с кнопками?
5. Что такое функция?
6. Что такое тело функции?
7. Как соединить объект и функцию?

# Задание на дом. Уровень 1

---

Создайте окно размером 500x500 с холстом 200x200. Начало холста в точке (200;10). Добавьте кнопку Круг. После нажатия левой кнопкой мыши на кнопку, должен появиться зеленый круг на весь холст.



# Задание на дом. Уровень 2

---

Создайте окно размером 500x500 с холстом 200x200. Начало холста в точке (200;10). Добавьте кнопку Круг. После нажатия левой кнопкой мыши на кнопку, должен появиться зеленый круг на весь холст. При нажатии на кнопку Очистить, холст очищается. Для очистки используйте метод объекта canvas `delete('all')`

