



алгоритмика

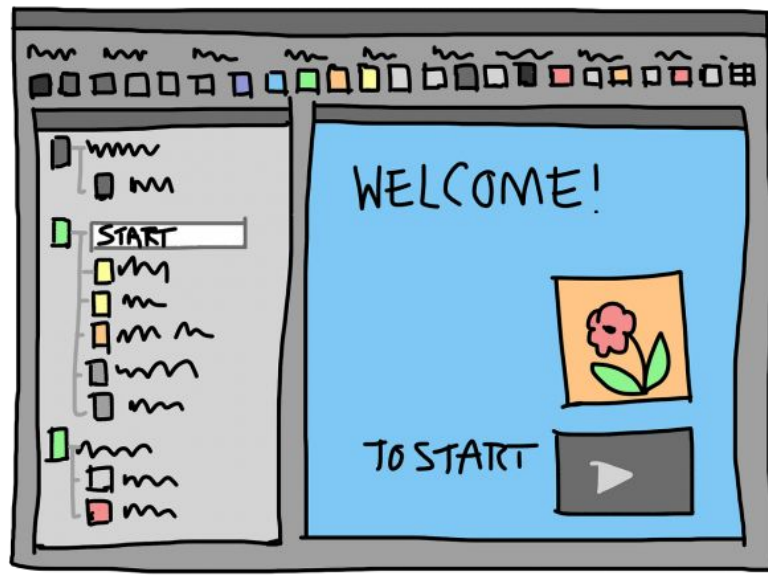
# PYTHON

Занятие 15

# Сегодня на уроке

---

- Создание нового инструмента.
- Случайное значение.



# Вопросы

---

1. Что означает событие B1-Motion?
2. Что означает событие Button-1?
3. Как задать цвет контура фигуры?
4. Какие виджеты вы уже знаете?
5. Как передать значение из виджета в переменную?
6. Что такое глобальная переменная?
7. Как указать в программе, что нужно изменить или использовать глобальную переменную?

# Создание переключателя

---

`var=IntVar()` – переменная `var` указывает, что значения в переключателе будут целыми числами (`IntVar`).

`var.set(0)` – по умолчанию будет выбран первый переключатель.

`r_1=Radiobutton(text="1",variable=var,value=0)`

`text="текст_рядом_с_кнопкой"`

`variable=имя_переменной`

`value=номер_кнопки`

`r_1.pack()`

`var.get()` – получение значения переменной.



# Настройка цвета виджета

---

**configure** – метод объектов tkinter, который позволяет настроить внешний вид виджета.

**объект.configure**(background="цвет")

Background – фон объекта

# Виджет Scale (шкала)

---

С помощью бегунка на шкале можно менять значения от значения `from_` до значения `to`.

```
var_scale=IntVar()
```

```
scale = Scale(root,from_=1, to=50, length=200, orient=HORIZONTAL, resolution=5, variable=var_scale)
```

1. `var_scale=IntVar()` – переменная, которая будет указывать на значение из виджета.
2. `scale = Scale(...)` – создание объекта класса `Scale`.
3. Аргументы класса `Scale`:

`root` – название окна, которому принадлежит виджет;

`from_ =1, to=50` – минимальное и максимальное значение;

`length=200` - длина в пикселях в окне;

`orient=HORIZONTAL` – ориентация шкалы (`VERTICAL` или `HORIZONTAL`);

`resolution=5` – шаг-сдвиг бегунка (минимальное=3);

`variable=var_scale` – соединение виджета и переменной.

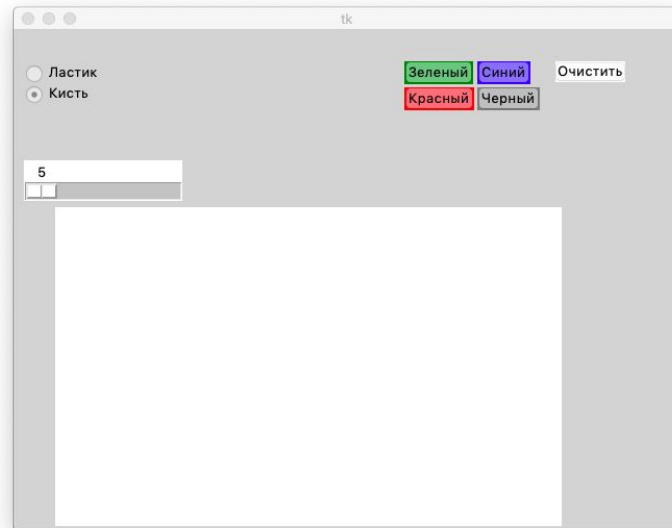


# Задание

---

Скачайте файл и запустите программу m2u6\_student.

- Что изменилось в интерфейсе приложения?
- Какие виджеты располагаются в окне?
- Минимальное и максимальное значение Scale?

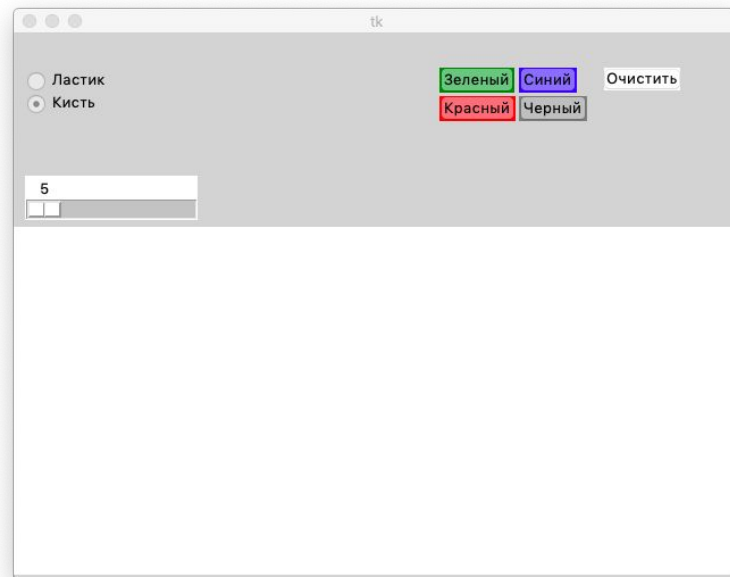


# Задание на повторение

---

Увеличьте размер холста, чтобы он занимал всё пространство по горизонтали окна.

- С какой точки должен начинаться холст?
- Какая ширина будет у холста?



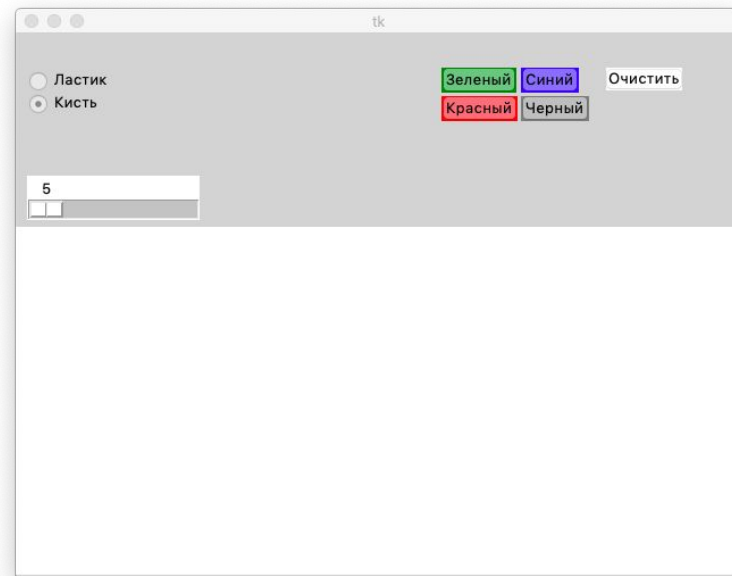


# Задание на повторение. Решение

---

Увеличьте размер холста, чтобы он занимал всё пространство по горизонтали окна.

```
canvas=Canvas(root,width=640,height=300)  
canvas.pack()  
canvas.place(x=0,y=170)
```



# Правила хорошего интерфейса

---

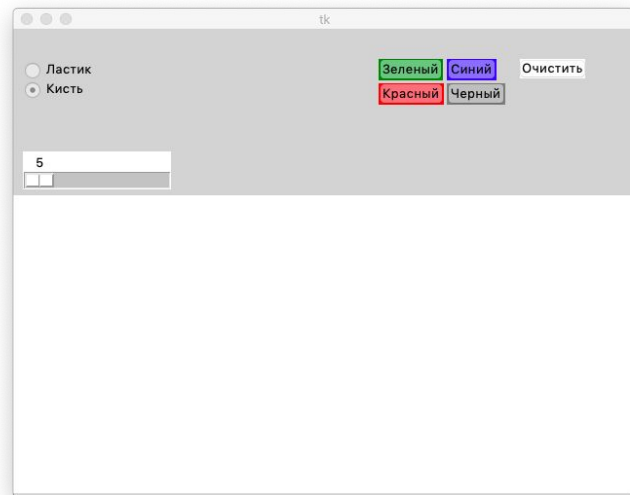
1. Все виджеты сгруппированы по смыслу (цвета, инструменты, меню и т.д.).
2. Виджеты одного назначения имеют одинаковый размер (кнопки, шкалы и т.п.).
3. Группы виджетов имеют название, чтобы пользователь мог понять их назначение.
4. Окно приложения имеет заголовок, который отражает назначение приложения.
5. Пользователю должно быть понятно, с каким инструментом он сейчас работает.

# Вопрос

---

## Что из этого уже учтено в нашем приложении?

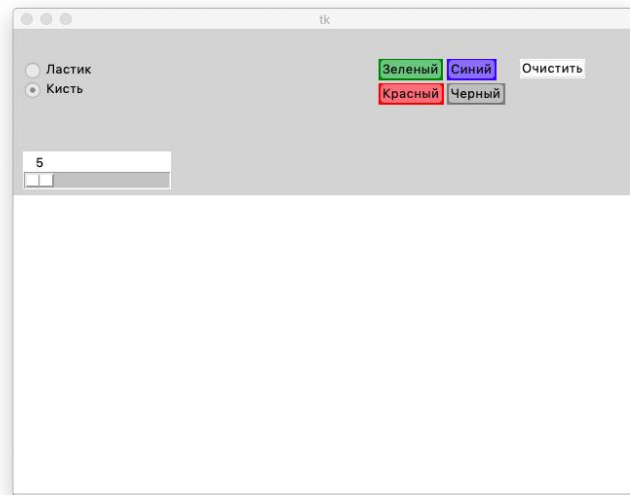
1. Все виджеты сгруппированы по смыслу (цвета, инструменты, меню и т.д.).
2. Виджеты одного назначения имеют одинаковый размер (кнопки, шкалы и т.п.).
3. Группы виджетов имеют название, чтобы пользователь мог понять их назначение.
4. Окно приложения имеет заголовок, который отражает назначение приложения.
5. Пользователю должно быть понятно, с каким инструментом он сейчас работает.



# Вопрос-ответ

## Что из этого уже учтено в нашем приложении?

- + 1. Все виджеты сгруппированы по смыслу (цвета, инструменты, меню и т.д.).
- 2. Виджеты одного назначения имеют одинаковый размер (кнопки, шкалы и т.п.).
- 3. Группы виджетов имеют название, чтобы пользователь мог понять их назначение.
- 4. Окно приложения имеет заголовок, который отражает назначение приложения.
- + - 5. Пользователю должно быть понятно, с каким инструментом он сейчас работает.

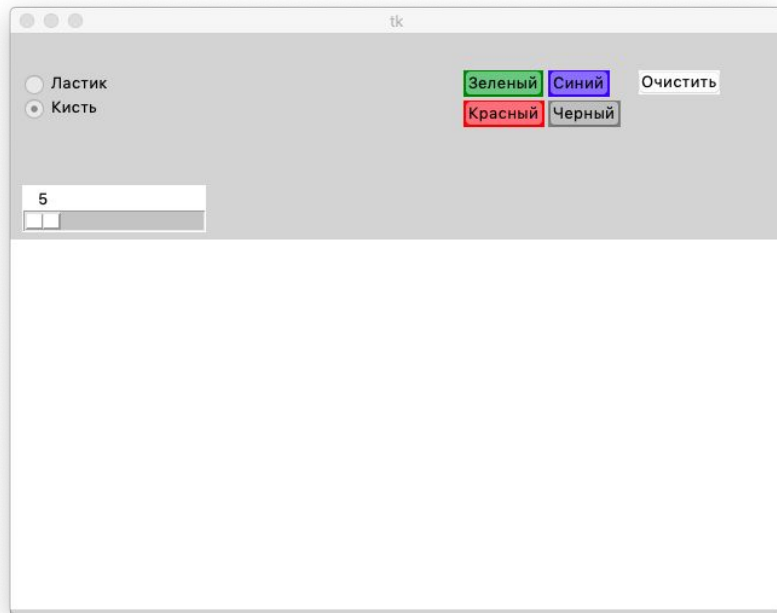


# Вопрос

---

## Какие виджеты имеют разные размеры?

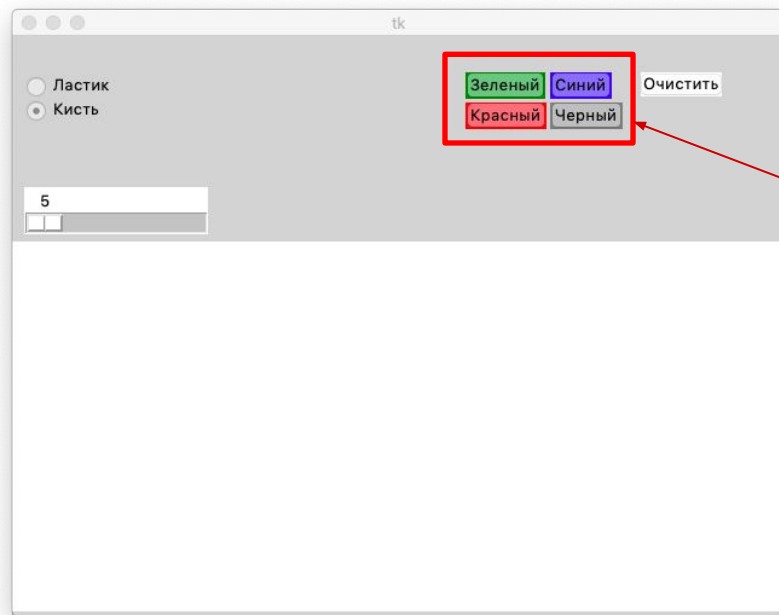
- Виджеты одного назначения имеют одинаковый размер (кнопки, шкалы и т.п.).



# Вопрос-ответ

## Какие виджеты имеют разные размеры?

- Виджеты одного назначения имеют одинаковый размер (кнопки, шкалы и т.п.).



**Кнопки**

# Установка размера виджета

---

Для изменения размера виджета, используются аргументы метода `configure`: `width` – ширина, `height` – высота.

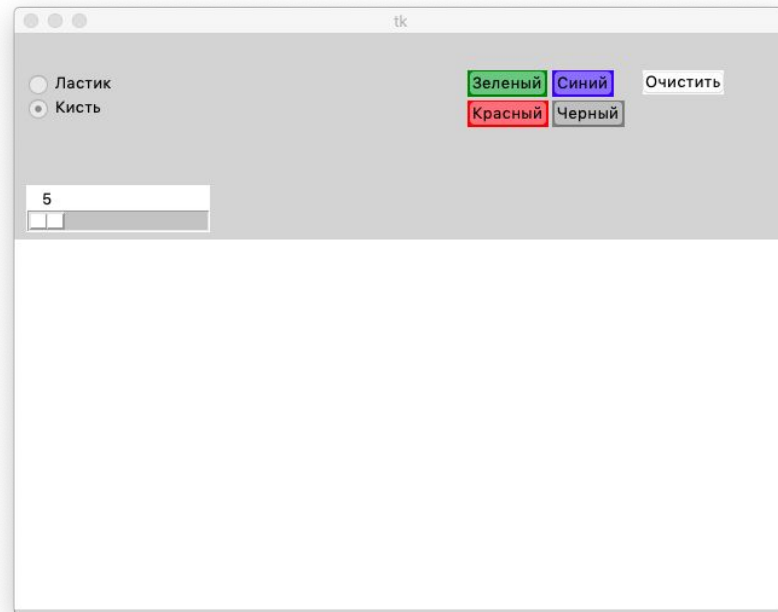
**Замечание: ширина и высота задаются не в пикселях, а в условных единицах.**

```
btn.configure(background="red", width=7, height=1)
```

# Задание

---

Задайте всем кнопкам в методе `configure`: `width=7`, `height=1`.  
Сравните внешний вид окна.





# Задание. Решение

---

Задайте всем кнопкам в методе `configure`: `width=7`, `height=1`.  
Сравните внешний вид окна.

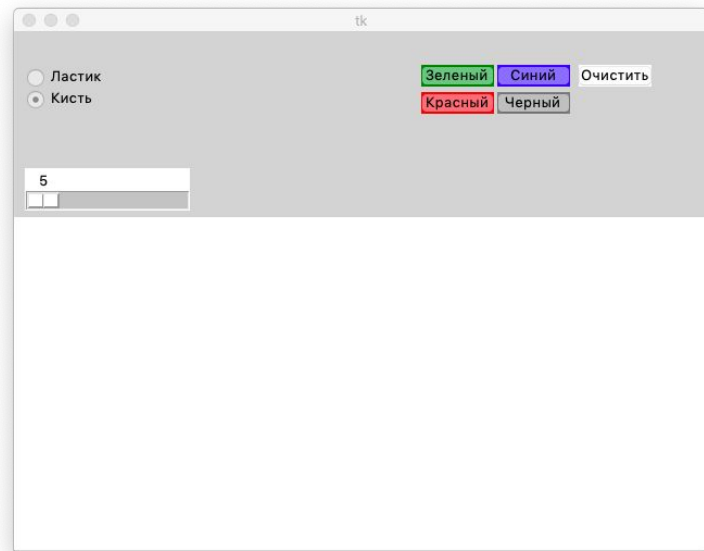
```
btn_1.configure(width=7,height=1)
```

```
btn_2.configure(background="green",width=7,height=1)
```

```
btn_3.configure(background="red",width=7,height=1)
```

```
btn_4.configure(background="blue",width=7,height=1)
```

```
btn_5.configure(background="gray",width=7,height=1)
```



# Правила хорошего интерфейса

---

- + 1. Все виджеты сгруппированы по смыслу (цвета, инструменты, меню и т.д.).
- + 2. Виджеты одного назначения имеют одинаковый размер (кнопки, шкалы и т.п.).
- 3. Группы виджетов имеют название, чтобы пользователь мог понять их назначение.
- 4. Окно приложения имеет заголовок, который отражает назначение приложения.
- 5. Пользователю должно быть понятно, с каким инструментом он сейчас работает.

# Виджет Label

---

Label – метка, которая позволяет добавлять названия над виджетом или группой виджетов.

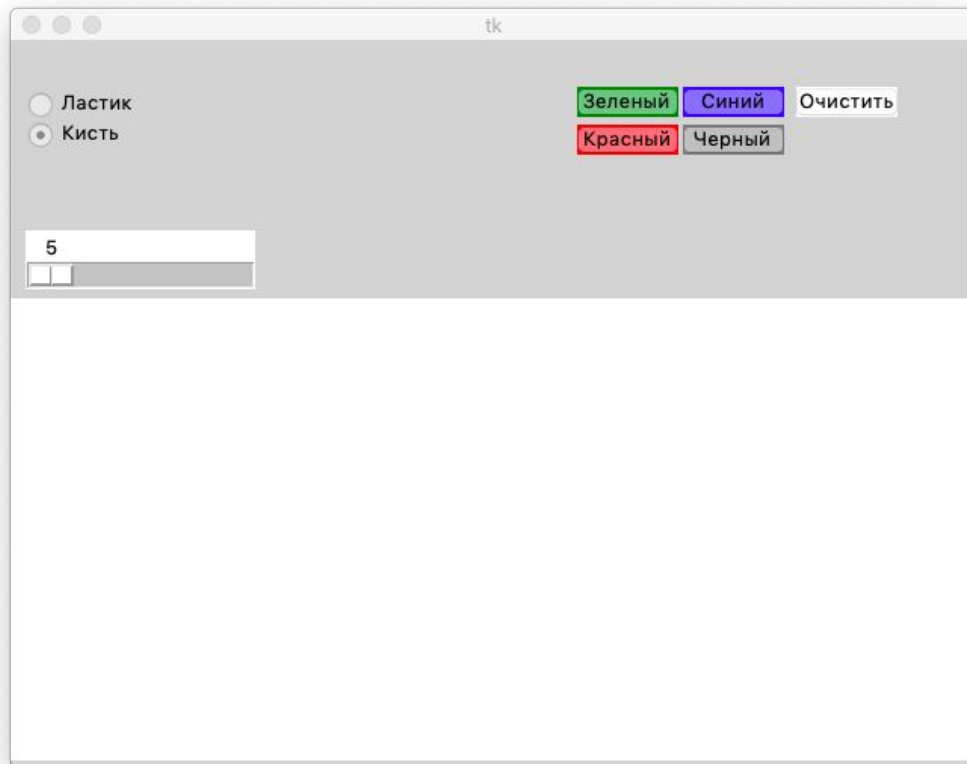
Название\_виджета = Label(название\_окна, text="Текст")

lab = Label(root, text="Инструменты")

# Вопрос

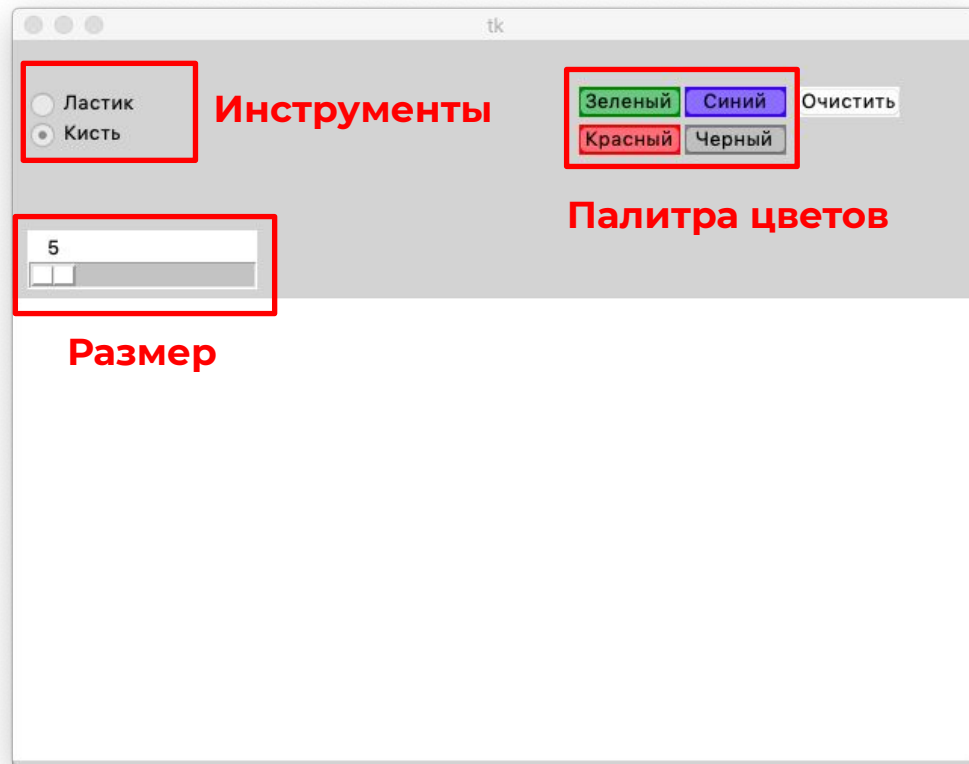
---

Какие группы виджетов  
можно выделить в нашем  
приложении?



# Вопрос-ответ

Какие группы виджетов можно выделить в нашем приложении?



# Задание

---

Добавьте соответствующие метки для виджетов, как показано на рисунке.

Название\_виджета = Label(название\_окна, text="Текст")



# Задание. Решение

Добавьте соответствующие метки для виджетов, как показано на рисунке.

```
lab_1 = Label(root, text="Инструмент")  
lab_1.configure(background='light gray')  
lab_1.pack()  
lab_1.place(x=10,y=5)
```

```
lab_2 = Label(root, text="Размер")  
lab_2.configure(background='light gray')  
lab_2.pack()  
lab_2.place(x=10,y=100)
```

```
lab_3 = Label(root, text="Палитра цветов")  
lab_3.configure(background='light gray')  
lab_3.pack()  
lab_3.place(x=370,y=5)
```



# Правила хорошего интерфейса

---

- + 1. Все виджеты сгруппированы по смыслу (цвета, инструменты, меню и т.д.).
- + 2. Виджеты одного назначения имеют одинаковый размер (кнопки, шкалы и т.п.).
- + 3. Группы виджетов имеют название, чтобы пользователь мог понять их назначение.
- 4. Окно приложения имеет заголовок, который отражает назначение приложения.
- 5. Пользователю должно быть понятно, с каким инструментом он сейчас работает.

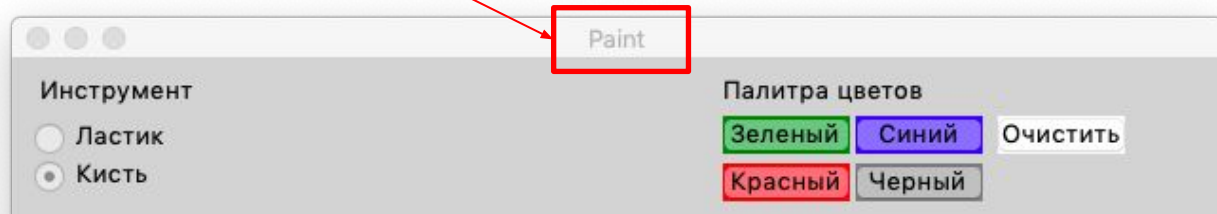


# Заголовок окна

---

Title – метод окна, который позволяет задавать название окна-приложения.

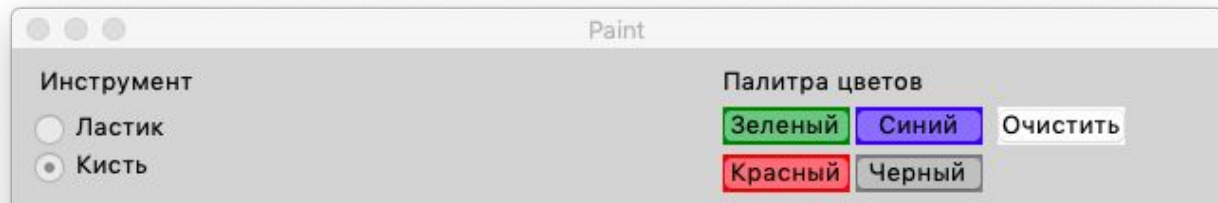
```
root.title("Paint")
```



# Задание

---

Добавьте заголовок (title) главному окну приложения.

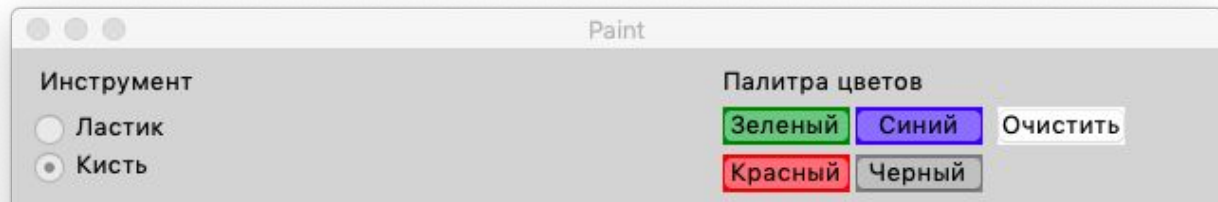


# Задание. Решение

---

Добавьте заголовок (title) главному окну приложения.

```
root=Tk()  
root.geometry('640x480')  
root.configure(background="light gray")  
root.title("Paint")
```



# Правила хорошего интерфейса

---

- + 1. Все виджеты сгруппированы по смыслу (цвета, инструменты, меню и т.д.).
- + 2. Виджеты одного назначения имеют одинаковый размер (кнопки, шкалы и т.п.).
- + 3. Группы виджетов имеют название, чтобы пользователь мог понять их назначение.
- + 4. Окно приложения имеет заголовок, который отражает назначение приложения.
- 5. Пользователю должно быть понятно, с каким инструментом он сейчас работает.

# Вопрос

---

Понятно ли сейчас пользователю, до того, как он начал рисовать, какого цвета будет линия?



# Вопрос-ответ

---

Понятно ли сейчас пользователю, до того, как он начал рисовать, какого цвета будет линия?

**Нет**



# Вопрос

---

Как обычно в приложениях выделяются элементы, которые неактивные или недоступные?



# Вопрос

---

Как обычно в приложениях выделяются элементы, которые неактивные или недоступные?

**Становятся серыми,  
а активные остаются  
яркими.**





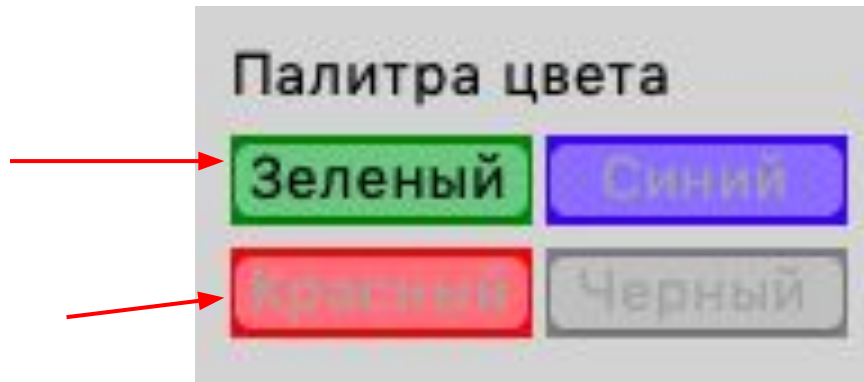
# Состояние кнопки

---

Кнопка имеет 2 основных состояния: активная или неактивная. Аргумент `state` метода `configure` позволяет изменять состояние кнопки.

```
btn.configure(state="normal")
```

```
btn.configure(state="disabled")
```

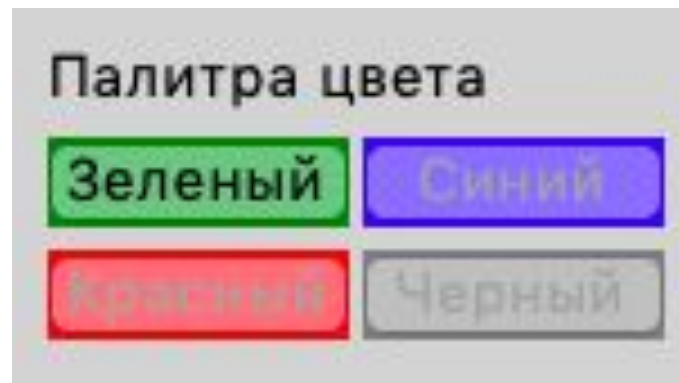


# Вопрос

---

Если пользователь выбрал «Зелёный» цвет, то какая кнопка должна быть активной, а какие неактивными?

Кнопка	Состояние
Зелёный	
Синий	
Красный	
Чёрный	

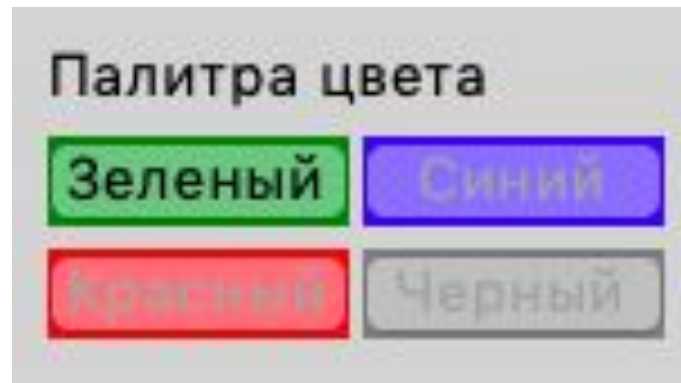


# Вопрос-ответ

---

Если пользователь выбрал «Зелёный» цвет, какая кнопка должна быть активной, а какие неактивными?

Кнопка	Состояние
<b>Зелёный</b>	<b>normal</b>
Синий	disabled
Красный	disabled
Чёрный	disabled

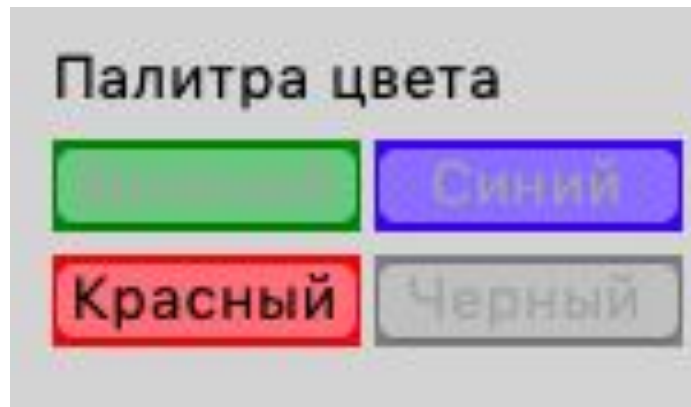


# Вопрос

---

Если пользователь выбрал «Красный» цвет, какая кнопка должна быть активной, а какие неактивными?

Кнопка	Состояние
Зелёный	
Синий	
Красный	
Чёрный	

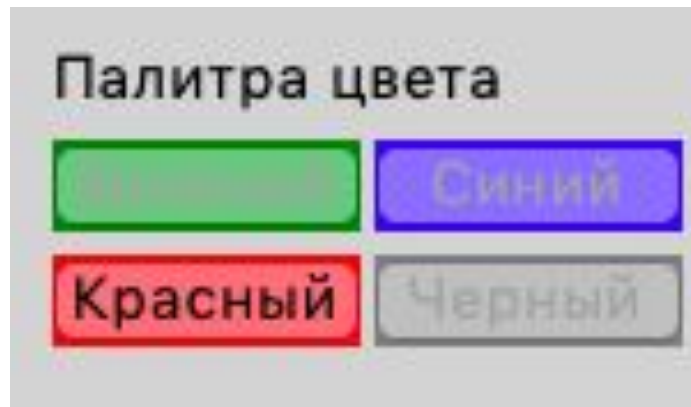


# Вопрос-ответ

---

Если пользователь выбрал «Красный» цвет, какая кнопка должна быть активной, а какие неактивными?

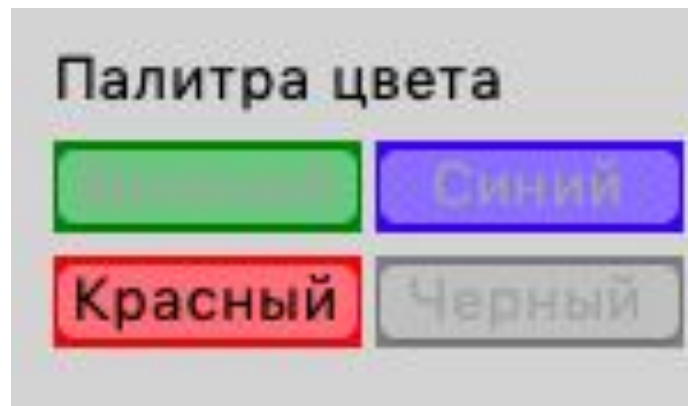
Кнопка	Состояние
Зелёный	disabled
Синий	disabled
<b>Красный</b>	<b>normal</b>
Чёрный	disabled



# Вопрос

---

После какого действия пользователя кнопка должна стать активной?

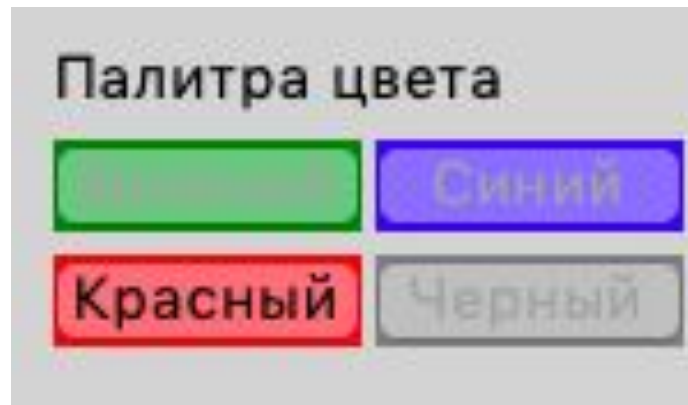


# Вопрос-ответ

---

После какого действия пользователя кнопка должна стать активной?

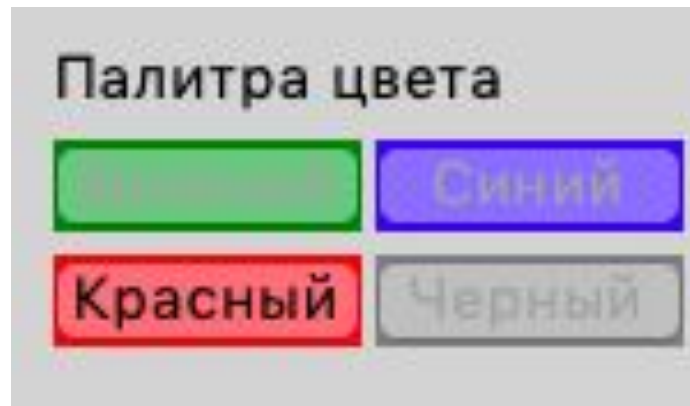
**После нажатия  
левой кнопкой  
мышы по кнопке.**



# Вопрос

---

Какая часть кода отвечает за событие после нажатия на кнопку?





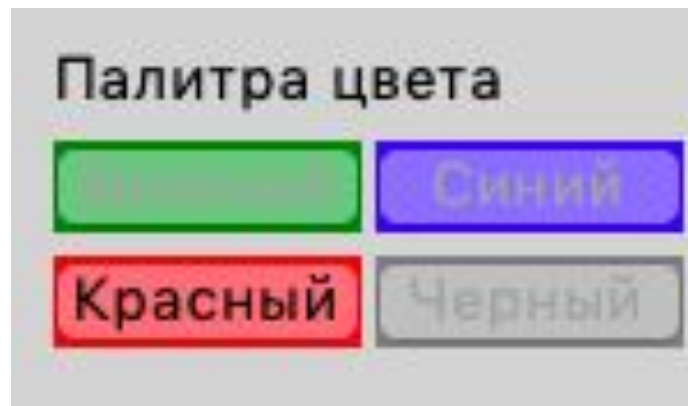
# Вопрос

---

Какая часть кода отвечает за событие после нажатия на кнопку?

## Функция

```
def color_red(event):  
    global color_fill  
    color_fill="red"
```



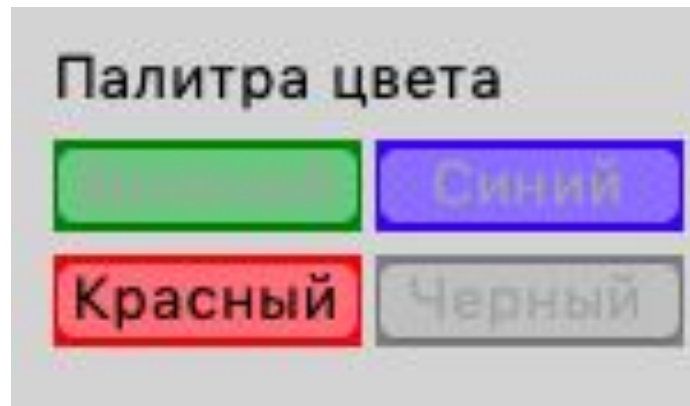
# Задание

---

Добавьте в программу изменение состояние кнопки. Если кнопку нажали (выбрали), она активная, иначе неактивная.

```
btn.configure(state="normal")
```

```
btn.configure(state="disabled")
```



# Задание. Решение

---

Добавьте в программу изменение состояние кнопки. Если кнопку нажали (выбрали), она активная, иначе неактивная.

```
def color_green(event):  
    global color_fill  
    color_fill="green"  
    btn_2.configure(state="normal")  
    btn_3.configure(state="disabled")  
    btn_4.configure(state="disabled")  
    btn_5.configure(state="disabled")
```

```
def color_red(event):  
    global color_fill  
    color_fill="red"  
    btn_2.configure(state="disabled")  
    btn_3.configure(state="normal")  
    btn_4.configure(state="disabled")  
    btn_5.configure(state="disabled")
```

```
def color_blue(event):  
    global color_fill  
    color_fill="blue"  
    btn_2.configure(state="disabled")  
    btn_3.configure(state="disabled")  
    btn_4.configure(state="normal")  
    btn_5.configure(state="disabled")
```

```
def color_black(event):  
    global color_fill  
    color_fill="black"  
    btn_2.configure(state="disabled")  
    btn_3.configure(state="disabled")  
    btn_4.configure(state="disabled")  
    btn_5.configure(state="normal")
```



# Правила хорошего интерфейса

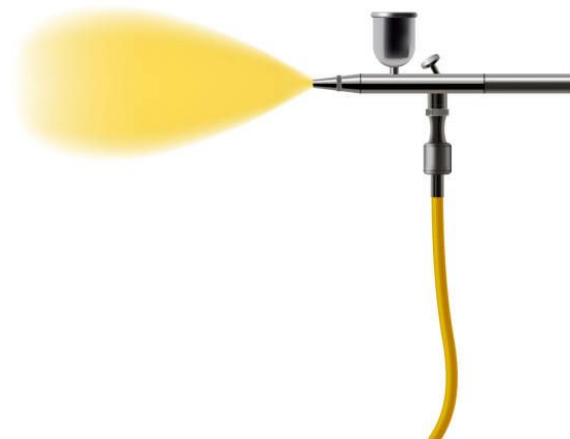
---

- + 1. Все виджеты сгруппированы по смыслу (цвета, инструменты, меню и т.д.).
- + 2. Виджеты одного назначения имеют одинаковый размер (кнопки, шкалы и т.п.).
- + 3. Группы виджетов имеют название, чтобы пользователь мог понять их назначение.
- + 4. Окно приложения имеет заголовок, который отражает назначение приложения.
- + 5. Пользователю должно быть понятно, с каким инструментом он сейчас работает.

# Вопрос

---

Как работает распылитель?



# Вопрос

---

Как работает распылитель?

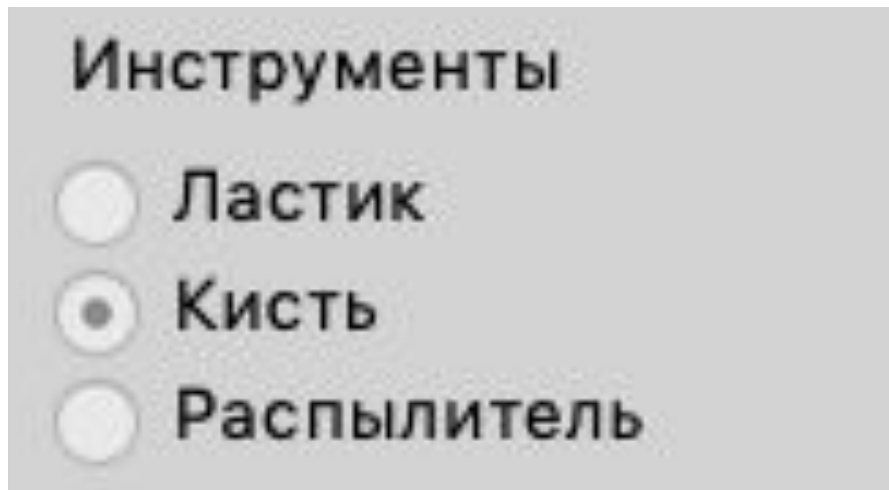
**Закрашиваем  
мелкими частицами  
область вокруг себя.**



# Создание инструмента «Распылитель»

---

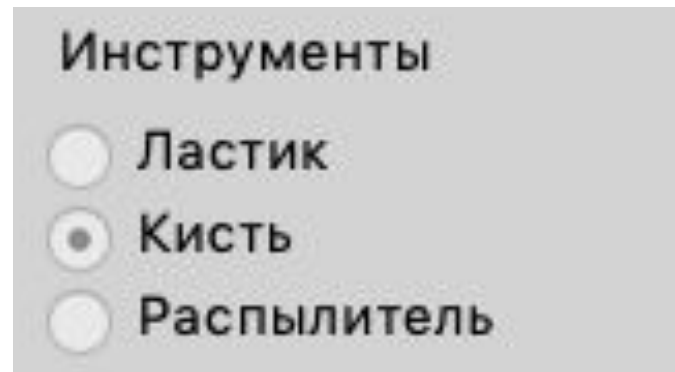
**1. Инструмент «Распылитель»** - рассеивает частицы краски вокруг нажатия кнопки.



# Задание

---

Добавьте в Переключатель «Инструменты» кнопку «Распылитель».



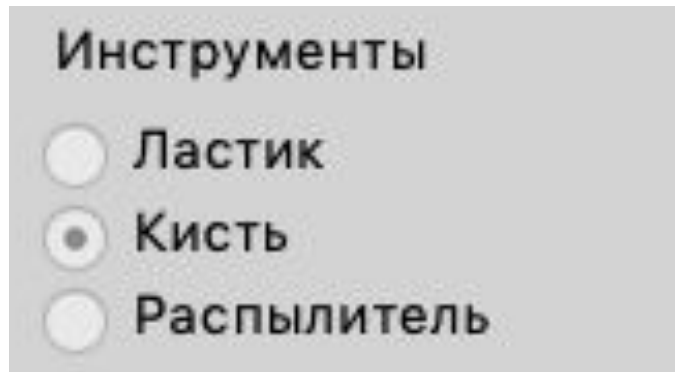


# Задание. Решение

---

Добавьте в Переключатель «Инструменты» кнопку «Распылитель».

```
r_3=Radiobutton(text="
Распылитель",variable=var_radio,value=2)
r_3.configure(background='light gray')
r_3.pack()
r_3.place(x=10,y=70)
```



# Создание инструмента «Распылитель»

---

## 2. Настройка распыления:

**Размер/Распыление** – величина одной частицы.

**Интенсивность распыления** – это количество частиц краски на участок холста.

Размер/Распыление	Интенсивность
<div>5</div> <div><div></div><div></div></div>	<div>3</div> <div><div></div><div></div></div>

# Задание

---

Переименуйте виджет Размер в Размер/Распыление.  
Добавьте ещё одну шкалу и метку с названием.

```
var_scale_2=IntVar()  
scale = Scale(root,from_=3, to=20, length=150, orient=HORIZONTAL, resolution=3,  
variable=var_scale_2)
```

```
scale.pack()  
scale.place(x=200,y=125)
```



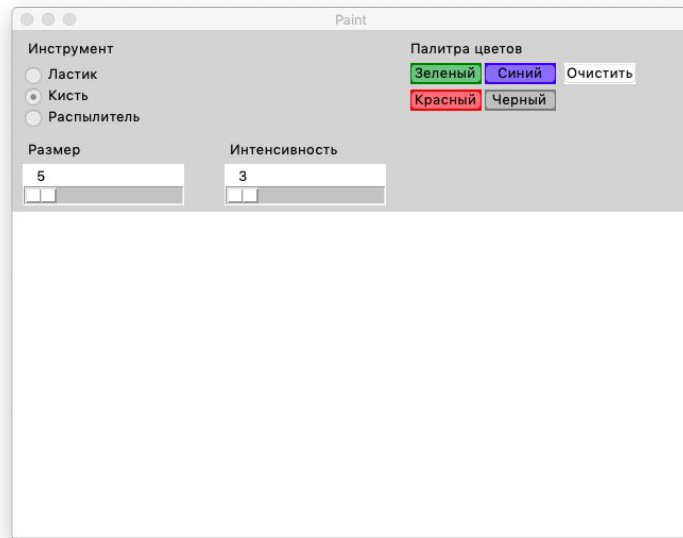
# Задание. Решение

Переименуйте виджет Размер в Размер/Распыление.  
Добавьте ещё одну шкалу и метку с названием.

```
var_scale_2=IntVar()  
scale = Scale(root,from_=3, to=20, length=150, orient=HORIZONTAL, resolution=3,  
variable=var_scale_2)
```

```
scale.pack()  
scale.place(x=200,y=125)
```

```
lab_4 = Label(root,text="Интенсивность")  
lab_4.configure(background='light gray')  
lab_4.pack()  
lab_4.place(x=200,y=100)
```



# Вопрос

---

Как должна измениться функция `click_left`, чтобы учесть появление нового инструмента?

```
def click_left(event):  
    x = event.x  
    y = event.y  
    r=var_scale.get()  
    if var_radio.get() == 0:  
        global color_fill  
        color_fill = "white"  
        canvas.create_oval(x-r,y-r,  
                           x+r,y+r,fill=color_fill, outline="")  
    else:  
        canvas.create_oval(x-r,y-r,  
                           x+r,y+r,fill=color_fill, outline="")
```

# Вопрос-ответ

---

Как должна измениться функция `click_left`, чтобы учесть появление нового инструмента?

**Добавить еще одну  
проверку выбора  
кнопки.**

```
def click_left(event):  
    x = event.x  
    y = event.y  
    r=var_scale.get()  
    if var_radio.get() == 0:  
        global color_fill  
        color_fill="white"  
        canvas.create_oval(x-r,y-r,  
                           x+r,y+r,fill=color_fill, outline="")  
    else:  
        canvas.create_oval(x-r,y-r,  
                           x+r,y+r,fill=color_fill, outline="")
```

# Вопрос

---

Знаем ли мы заранее, где упадёт частица краски?



# Вопрос

---

Знаем ли мы заранее, где упадёт частица краски?

**Нет, это случайность!**





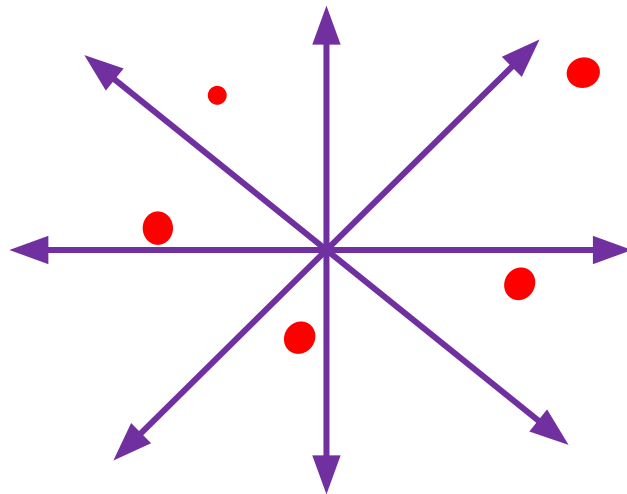
# Создание инструмента «Распылитель»

---

Настройка распыления:

$x + \text{randint}(-r, r), y + \text{randint}(-r, r), x - \text{randint}(-r, r), y - \text{randint}(-r, r)$

$(x, y)$  – точка нажатия  
пользователем.



# Создание инструмента «Распылитель»

Настройка распыления:

$x + \text{randint}(-r, r), y + \text{randint}(-r, r), x - \text{randint}(-r, r), y - \text{randint}(-r, r)$

$(x, y)$  – точка нажатия  
пользователем.

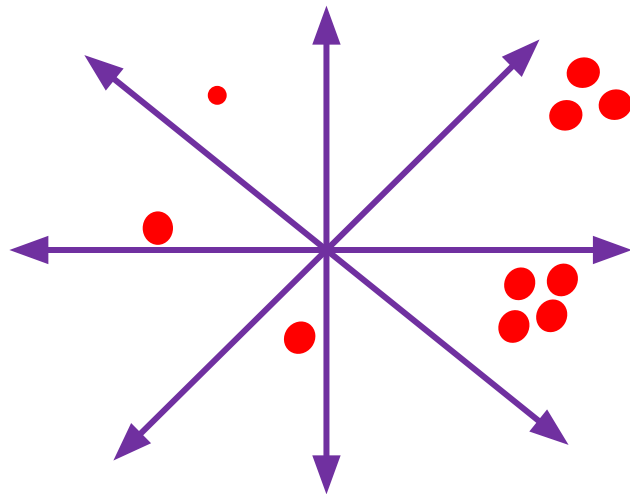
$dR$  – величина количества  
точек в одном месте.

Интерфейс настройки интенсивности:

Интенсивность

3

График с ползунком, показывающим текущее значение 3.



# Задание

---

Измените программу: добавьте модуль `random`, взятие значения из шкалы «Интенсивность» и событие после выбора инструмента «Распылитель».

```
from random import*
```

```
dR=var_scale_2.get()
```

```
elif var_radio.get() == 2:  
    for i in range(randint(3,dR)):  
        canvas.create_oval(x+randint(-r,r),y+randint(-r,r),x+randint(-r,r),  
                            y+randint(-r,r),fill=color_fill, outline="black")
```

# Задание. Решение

---

Измените программу: добавьте модуль random, взятие значения из шкалы «Интенсивность» и событие после выбора инструмента «Распылитель».

```
from random import*

def click_left(event):
    x = event.x
    y = event.y
    r=var_scale.get()
    dR=var_scale_2.get()
    if var_radio.get() == 0:
        global color_fill
        color_fill="white"
        canvas.create_oval(x-r,y-r,x+r,y+r,fill=color_fill, outline="")
    elif var_radio.get() == 2:
        for i in range(randint(3,dR)):
            canvas.create_oval(x+randint(-r,r),y+randint(-r,r),x-randint(-r,r),
                               y-randint(-r,r),fill=color_fill, outline="black")
    else:
        canvas.create_oval(x-r,y-r,x+r,y+r,fill=color_fill, outline="")
```

# Задание

---

Измените функцию `move_left` также, как и `click_left`.

```
def click_left(event):
    x = event.x
    y = event.y
    r=var_scale.get()
    dR=var_scale_2.get()
    if var_radio.get() == 0:
        global color_fill
        color_fill = "white"
        canvas.create_oval(x-r,y-r,x+r,y+r,fill=color_fill, outline="")
    elif var_radio.get() == 2:
        for i in range(randint(3,dR)):
            canvas.create_oval(x+randint(-r,r),y+randint(-r,r),x-randint(-r,r),
                               y-randint(-r,r),fill=color_fill, outline="black")
    else:
        canvas.create_oval(x-r,y-r,x+r,y+r,fill=color_fill, outline="")
```

# Вопрос

---

С какой формой кисти мы работаем при рисовании?



# Вопрос-ответ

---

С какой формой кисти мы работаем при рисовании?

**С круглой.**



# Вопрос

---

Создание каких фигур мы ещё знаем?





# Вопрос-ответ

---

Создание каких фигур мы ещё знаем?

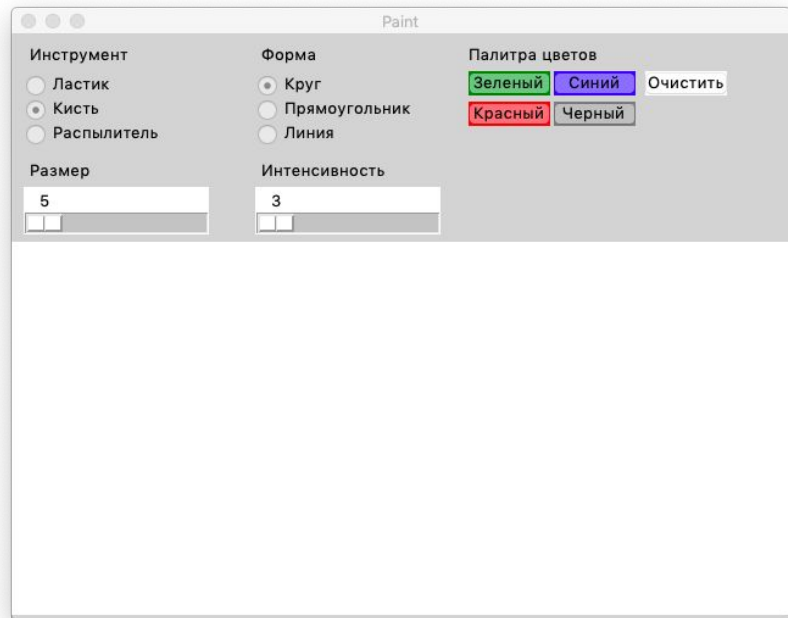
**Линия,  
прямоугольник,  
треугольник.**



# Задание

---

Добавьте переключатель «Форма» с кнопками: круг, прямоугольник, линия. Название переменной `var_radio_shape`. Значения кнопок: 0, 1, 2.



# Задание. Решение

---

Добавьте переключатель «Форма» с кнопками: круг, прямоугольник, линия. Название переменной `var_radio_shape`. Значения кнопок: 0, 1, 2.

```
lab_5 = Label(root,text="Форма")
lab_5.configure(background='light gray')
lab_5.pack()
lab_5.place(x=200,y=5)
```

```
var_radio_shape=IntVar()
var_radio_shape.set(0)
r_4=Radiobutton(text="Круг",variable=var_radio_shape,value=0)
r_4.configure(background='light gray')
r_4.pack()
r_4.place(x=200,y=30)
```

```
r_5=Radiobutton(text="Прямоугольник",variable=var_radio_shape,value=1)
r_5.configure(background='light gray')
r_5.pack()
r_5.place(x=200,y=50)
```

```
r_6=Radiobutton(text="Линия",variable=var_radio_shape,value=2)
r_6.configure(background='light gray')
r_6.pack()
r_6.place(x=200,y=70)
```



# Вопрос

---

Какой след должен остаться при движении мыши, если выбрана «Кисть» и «Круг»?

Какой след должен остаться при движении мыши, если выбрана «Кисть» и «Прямоугольник»?

Какой след должен остаться при движении мыши, если выбрана «Кисть» и «Линия»?

# Вопрос-ответ

Какой след должен остаться при движении мыши, если выбрана «Кисть» и «Круг»?

Какой след должен остаться при движении мыши, если выбрана «Кисть» и «Прямоугольник»?

Какой след должен остаться при движении мыши, если выбрана «Кисть» и «Линия»?

Инструменты	Форма
<input type="radio"/> Ластик	<input checked="" type="radio"/> Круг
<input checked="" type="radio"/> Кисть	<input type="radio"/> Прямоугольник
<input type="radio"/> Распылитель	<input type="radio"/> Линия
Размер/Распыление	Интенсивность
5	3

Инструменты	Форма
<input type="radio"/> Ластик	<input type="radio"/> Круг
<input checked="" type="radio"/> Кисть	<input checked="" type="radio"/> Прямоугольник
<input type="radio"/> Распылитель	<input type="radio"/> Линия
Размер/Распыление	Интенсивность
5	3

Инструменты	Форма
<input type="radio"/> Ластик	<input type="radio"/> Круг
<input checked="" type="radio"/> Кисть	<input type="radio"/> Прямоугольник
<input type="radio"/> Распылитель	<input checked="" type="radio"/> Линия
Размер/Распыление	Интенсивность
20	3

# Задание

---

Добавьте в программу зависимость от выбора в двух переключателях.

**Подсказка: рисование выполняется только когда выбрано И одно И другое.**

# Задание. Решение

Добавьте в программу зависимость от выбора в двух переключателях.

**Подсказка: рисование выполняется только когда выбрано И одно И другое.**

```
if var_radio.get()==0:
    global color_fill
    color_fill="white"
    canvas.create_oval(x-r,y-r,x+r,y+r,fill=color_fill, outline="")
elif ((var_radio.get()==1) and (var_radio_shape.get()==1)):
    canvas.create_rectangle(x+r,y+r,x-r,y-r,fill=color_fill,outline="")
elif ((var_radio.get()==1) and (var_radio_shape.get()==2)):
    canvas.create_rectangle(x,y+r,x,y-r,fill=color_fill,outline="")
elif ((var_radio.get()==2) and (var_radio_shape.get()==0)):
    for i in range(randint(3,dR)):
        canvas.create_oval(x+randint(-r,r),y+randint(-r,r),x-randint(-r,r),y-randint(-r,r),
                           fill=color_fill, outline="black")
elif ((var_radio.get()==2) and (var_radio_shape.get()==1)):
    for i in range(randint(3,dR)):
        canvas.create_rectangle(x+randint(-r,r),y+randint(-r,r),x-randint(-r,r),
                               y-randint(-r,r),fill=color_fill, outline="black")
else:
    canvas.create_oval(x-r,y-r,x+r,y+r,fill=color_fill, outline="")
```



# Вопросы

---

1. Что такое random?
2. Какие виджеты вы уже знаете?
3. Как передать значение из виджета в переменную?
4. Как задать название окну?
5. Как создать надпись над виджетом?
6. Что такое составное условие?



# Задание на дом. Уровень 1

---

Измените размер окна на 800\*800, расположите элементы относительно нового размера окна.



## Задание на дом. Уровень 2

---

Измените размер окна на 800\*800, расположите элементы относительно нового размера окна.  
Добавьте в группу формы: треугольник и измените соответствующие функции.