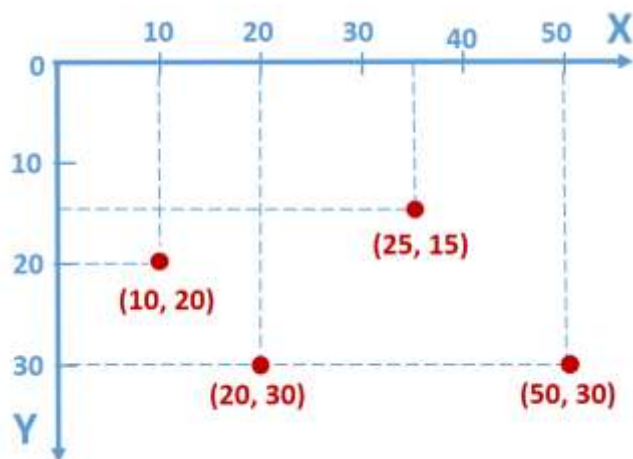


При размещении на холсте геометрических и других объектов указываются их координаты на холсте. Точкой отсчета является верхний левый угол холста.



Создание экземпляра изображения

Чтобы добавить изображение в любой контейнер, сначала надо создать этот объект как объект-экземпляр класса *PhotoImage*.

```
объект_изображение = PhotoImage(file="имя_файла")
```

```
f = PhotoImage(file="img1.gif")
```

Указанный файл *img1.gif* должен находиться в той же папке, что и сама программа.

tkinter поддерживает эти форматы графических файлов:

- PNG
- GIF

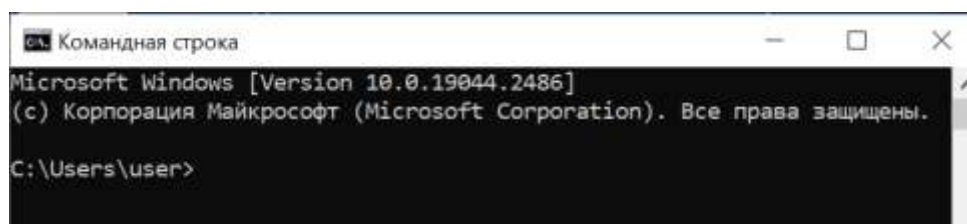
ВНИМАНИЕ! Старайтесь называть файлы так, чтобы в имени файла были только латиница, цифры и знаки подчеркивания.

Дополнительная информация

Поддержка формата JPG

Чтобы загружать изображения формата JPG, необходимо установить библиотеку PIL (*Python Imaging Library*).

Для установки библиотеки вызовите командную строку Windows – в окне поиска наберите *cmd*. Появится следующее окно



В командной строке наберите команду *pip install pillow* и нажмите <Enter>. После установки библиотеки выйдите из командной строки командой *exit*

Теперь добавьте сразу после строки с импортом из *tkinter* строку
from PIL import Image, ImageTk

Для создания экземпляра класса изображения в отличие от предыдущего способа надо добавить две строки:

```
p = Image.open("img1.jpg")
f = ImageTk.PhotoImage(p)
```

Теперь разместим экземпляр изображения с именем *f* (неважно, созданный первым или вторым способом) в окне. Мы будем делать двумя способами: с использованием холста и с использованием виджета *Label*

Вставка изображения с помощью метода холста *create_image*

Метод холста *create_image* размещает объект-изображение в точке с координатами (x,y) с точкой привязки (*anchor*)

```
рисунок = холст.create_image(x,y,
                             image=объект_изображение, anchor="значение")
```

Обратите внимание, что координаты (x,y) – это местоположение указанной точки привязки изображения.

Приведем полный текст программы:

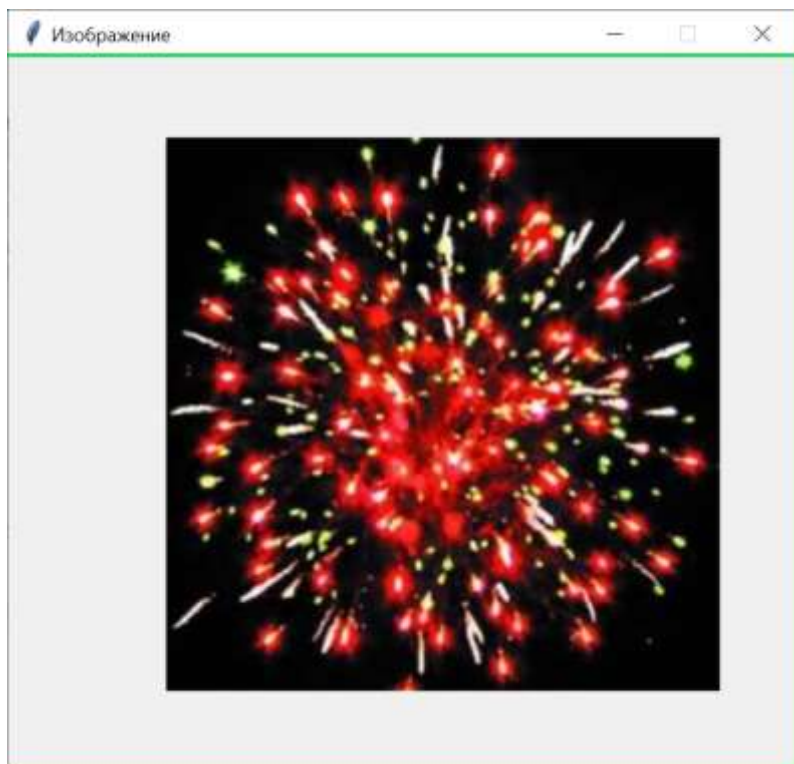
```
from tkinter import *                                # подключаем модуль tkinter
tk = Tk()                                           # создаем объект главного окна
tk.title('Изображение')                             # задаем заголовок окна
tk.geometry("500x450+0+0")                           # задаем размеры и смещение окна
tk.resizable(width=False, height=False)             # запрещаем менять размеры окна

c = Canvas(tk, width=500, height=450)
c.pack()

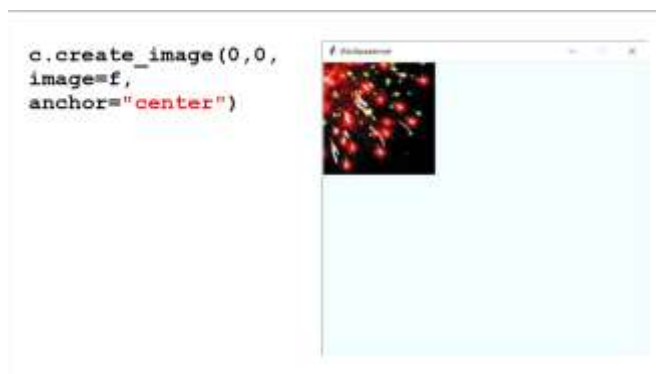
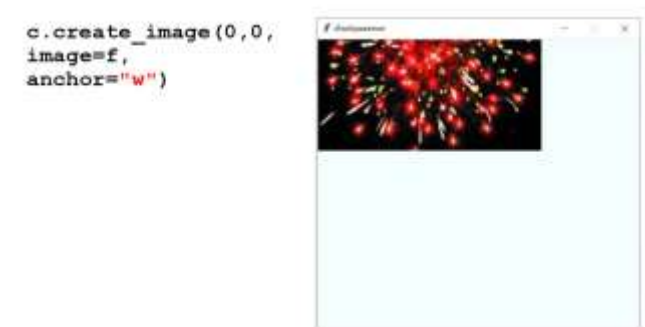
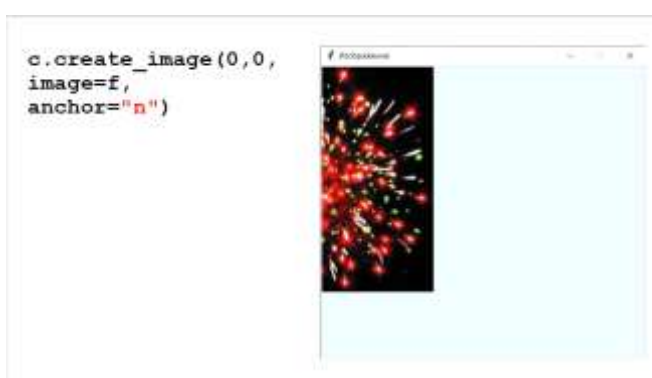
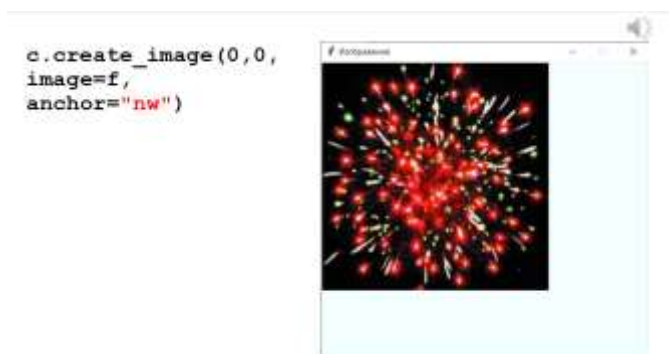
f=PhotoImage(file="fire.png")
img=c.create_image(100,50,image=f,anchor="nw")
tk.update()                                         # обновляем содержимое окна

tk.mainloop()
```

Размеры файла *fire.png* – 350x350. Результат работы программы:



В дальнейших примерах использования параметра *anchor* будет изменен только выделенный красным текст программы:



```

c.create_image(0,0,
image=f,
anchor="ne")

c.create_image(0,0,
image=f,
anchor="e")

c.create_image(0,0,
image=f,
anchor="sw")

c.create_image(0,0,
image=f,
anchor="s")

c.create_image(0,0,
image=f,
anchor="se")

```



```

c.create_image(500,450,
image=f,
anchor="se")

```



```

c.create_image(500,450,
image=f,
anchor="center")

```



```

c.create_image(500,450,
image=f,
anchor="e")

```



```

c.create_image(500,450,
image=f,
anchor="s")

```



```

c.create_image(500,450,
image=f,
anchor="nw")

c.create_image(500,450,
image=f,
anchor="n")

c.create_image(500,450,
image=f,
anchor="ne")

c.create_image(500,450,
image=f,
anchor="w")

c.create_image(500,450,
image=f,
anchor="sw")

```

