

# Челлендж «Программирование анимированных объектов на языке Питон. Потешные огни»

Наша задача – научиться изменять существующие объекты.

Для выполнения работы будем использовать то, чему научились в первом вызове.

## Создаем графические объекты

Вспомним, что надо было сделать до рисования фигур.

```
PA_items1.py - C:\ID\!Петровские асссамблеи\Фейерверки\PA_items1.py (3.10.2)
File Edit Format Run Options Window Help
import time
from tkinter import *

tk = Tk()          # создаем объект главного окна
tk.title('Потешные огни') # задаем заголовок окна
tk.resizable(width=False, height=False) # делаем невозможным изменение ширины и высоты окна

w=1024 # ширина холста
h=576 # высота холста
clr='#000000' # цвет фона холста

# создаем объект - экземпляр класса холста
# все остальные объекты затем будут размещены на холсте
c=Canvas(tk,width=w,height=h, bg=clr)
c.pack() # вызываем метод отображения объекта холста

tk.update() # обновляем информацию в главном окне

# задаем цвет линии
line_w=4
clr1='#cc3333' # красный
clr2='#cccc00' # желтый
clr3='#cc00cc' # маджента
clr4='#0099ff' # циан
```

Главное окно будет называться *tk*, а холст будет называться *c* (от слова *Canvas*).

Создаем холст размером 1024 на 576.

Для того, чтобы было проще работать с цветами, создайте несколько переменных для хранения цвета – как в данном примере переменные *clr1*, *clr2*, *clr3*, *clr4*.

Для будущих рисунков фон (*clr*) лучше сделать темным – наша задача создавать потешные огни, а на темном фоне фейерверк будет более заметным. Для изучения изменения объектов цвет фона не важен.

Не забывайте после каждого рисования изменения состояния объекта вызывать два метода – метод обновления окна *tk.update()* и метод задержки (в данном случае на 1 секунду) *time.sleep(1)* – для того, чтобы видеть изменения.

Ранее мы рисовали фигуры следующим образом:

```
c.create_oval((50,50),(100,100), width=0, fill="#ff0000")
c.create_oval((60,60),(90,90), width=0, fill="#ffff00")
```

Результатом работы будут два круга – первый красный диаметром 50 пикселей, второй, нарисованный поверх первого, – желтый диаметром 30 пикселей.



### Как можно менять графические объекты?

Нам не надо было запоминать ссылку на созданный объект, потому что мы не собирались ничего в нем изменять. Но чтобы изменить какие-то свойства объекта или переместить его, необходимо запомнить объект – присвоить значение метода создания фигуры какой-либо переменной. Изменим предыдущий код так:

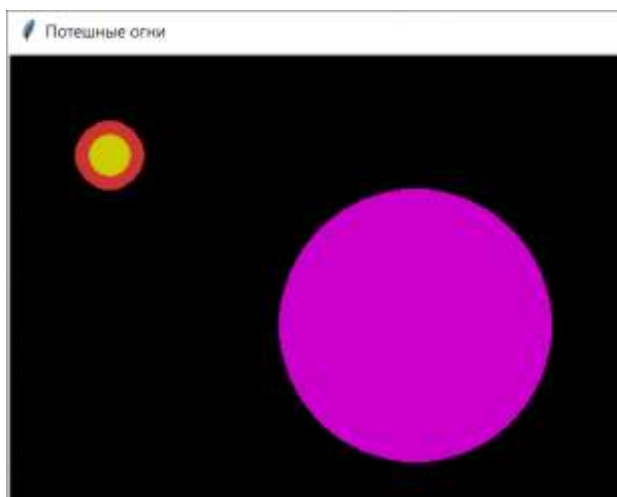
```
f1 = c.create_oval((50,50),(100,100), width=0, fill="#ff0000")
f2 = c.create_oval((60,60),(90,90), width=0, fill="#ffff00")
```

Для изменения свойств существующего объекта используется метод холста `itemconfig`. Первым параметром этого метода должна быть переменная – ссылка на тот объект, который изменяем. Затем идут те параметры, что вы хотите изменить.

### холст.itemconfig(объект, параметры)

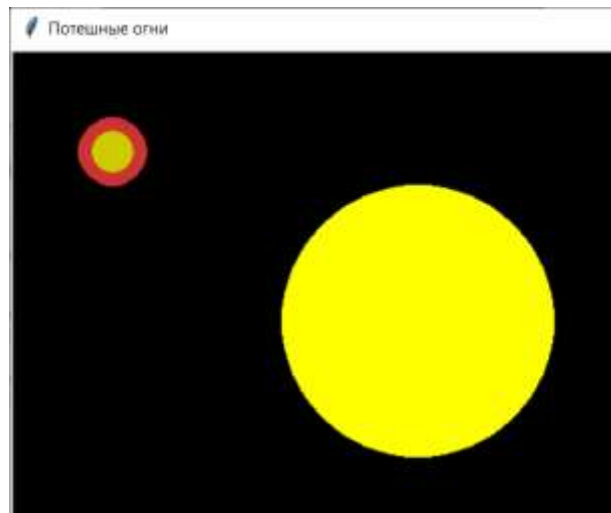
Создадим круг с заливкой цветом маджента, без обводки и диаметром 200 пикселей.

```
f3 = c.create_oval((200,100),(400,300), width=0, fill="#ff00ff")
time.sleep(1)
tk.update()
```



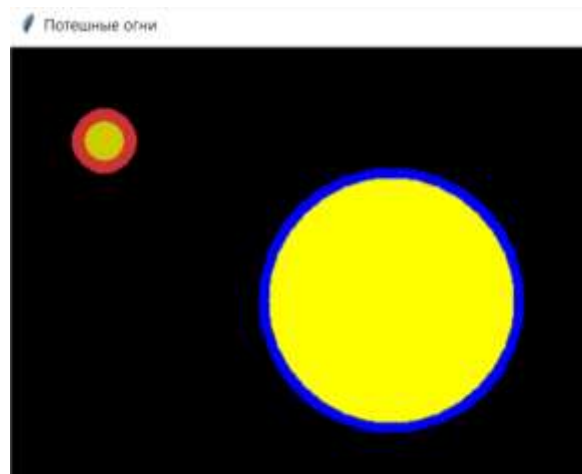
Изменим цвет заливки этого объекта на желтую:

```
c.itemconfig(f4, fill="#ffff00")
time.sleep(1)
tk.update()
```



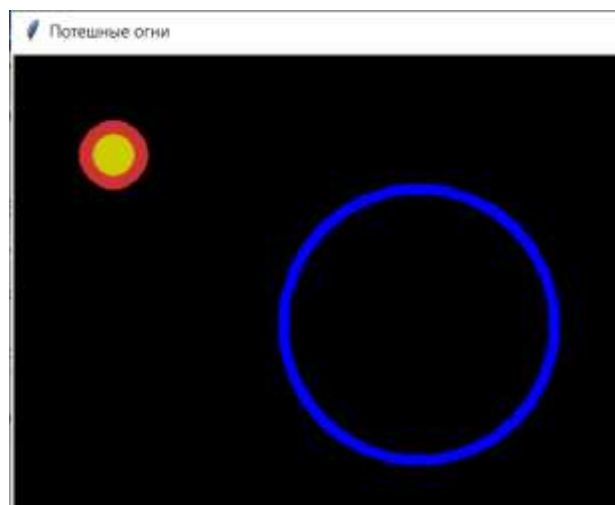
Добавим к измененному объекту синюю обводку шириной 8 пикселей:

```
c.itemconfig(f4, width=8, outline="#0000ff")
time.sleep(1)
tk.update()
```



Уберем заливку, оставляя только обводку:

```
c.itemconfig(f4, fill="")
time.sleep(1)
tk.update()
```



Для изменения координат существующего объекта используется метод холста *coords*. Первым параметром этого метода должна быть переменная – ссылка на тот объект, который изменяем. Затем идут новые координаты.

## холст.coords (объект, новые координаты)

С помощью этого метода можно и перемещать объект, и изменять размеры и пропорции объекта.

Попробуем изменить размер и пропорции фигуры. Создадим квадрат с циановой заливкой со стороной 400 пикселей.

```
f4=c.create_rectangle((500,100),(900,500), width=0, fill="#0099ff")
time.sleep(1)
tk.update()
```

Изменим его размер, оставив прежними координаты верхнего левого угла, и уменьшив на 100 координаты правого нижнего угла – таким образом мы получим прямоугольник шириной 300 и высотой 200 пикселей.

```
c.coords(f4, 500,100,800,300)
time.sleep(1)
tk.update()
```

Изменим координаты только верхнего левого угла, увеличив координату X на 200, а координату Y на 100 – мы получили квадрат со стороной 100 пикселей. Обратите внимание, что если при создании объекта можно было указывать координаты попарно в скобках, то в методе *coords* необходимо указывать их отдельно друг от друга.

```
c.coords(f4, 700,200,800,300)
time.sleep(1)
tk.update()
```

С помощью этого же метода можно и перемещать объект. Создадим красный квадрат со стороной 100 пикселей.

```
f5=c.create_rectangle((100,300),(200,400), width=0, fill="#ff0000")
time.sleep(1)
tk.update()
```

Вызовем для него метод *coords*, увеличив каждую координату на 100 – таким образом мы переместим этот объект, не меняя его пропорций, на 100 пикселей вниз и 100 пикселей вправо.

```
c.coords(f5, 200,400,300,500)
time.sleep(1)
tk.update()
```

## Использование метода холста move для перемещения объектов

Гораздо удобнее перемещать объекты, используя метод холста *move*. Первый параметр метода – ссылка на перемещаемый объект, второй – смещение по оси X, третий – смещение по оси Y. Напомним, что положительное смещение – это смещение вправо или вниз, а отрицательное смещение – это смещение влево или вверх, в зависимости от того, для какой координаты это смещение указано.

### холст.move (объект, А, В)

Этот метод перемещает объект на А пикселей по оси X и на В пикселей по оси Y. Переместим созданный ранее квадрат *f5* на 100 пикселей вправо и на 200 вверх:

```
c.move(f5, 100, -200)
time.sleep(1)
tk.update()
```

Теперь переместим его на 600 пикселей вправо и на 200 вниз.

```
c.move(f5, 600, 200)
time.sleep(1)
tk.update()
```

Если для линии, овала и прямоугольника не так сложно пересчитать координаты для перемещения методом *coords*, то дуги и многоугольники удобнее перемещать методом *move*, не делая сложных пересчетов новых координат объекта.

Попробуйте нарисовать треугольник в верхнем левом углу холста, поменять его заливку и обводку, а затем переместить его в правый нижний угол.