

Челлендж «Программирование графических объектов на языке Питон. Корабли нашей мечты»

Учимся рисовать на холсте

Наша задача – научиться рисовать фигуры на холсте.

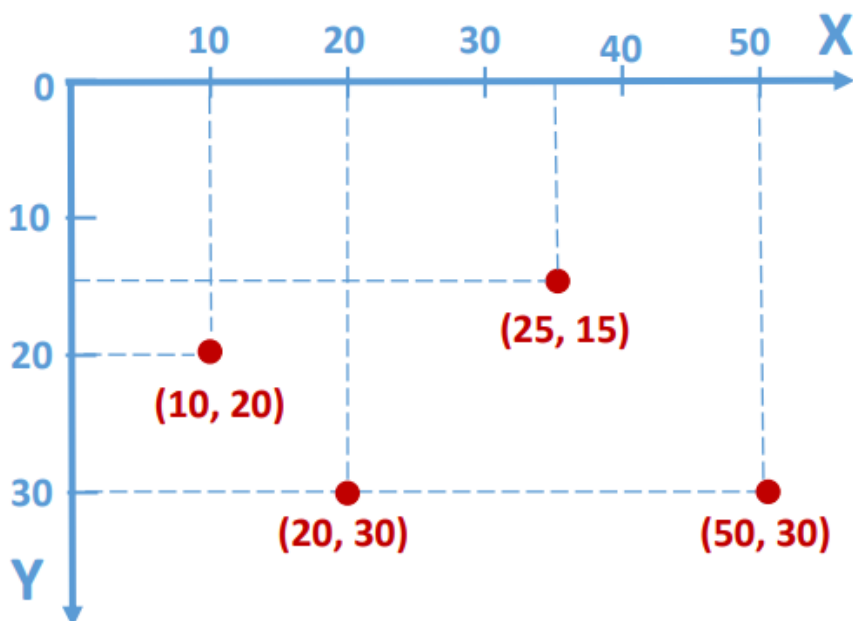
Для того, чтобы нарисовать точку на холсте, используются координаты точки x и y . Отсчет координат точек ведется из левого верхнего угла (начала координат). Координаты x откладываются по оси от начала координат вправо, координаты y откладываются от начала координат вниз.

Для того, чтобы нарисовать отрезок, надо указать координаты начала и конца отрезка.

Перед началом работы предлагается использовать написанную во втором вызове программу, которая создает главное окно и размещает в нем холст. Укажем размеры главного окна и холста, они должны совпадать: ширина окна и холста составляет 800 пикселей, высота окна и холста - 450 пикселей. Укажем белый цвет холста. Цвет окна оставим по умолчанию, т.к. окна видно не будет, его полностью закроет холст.

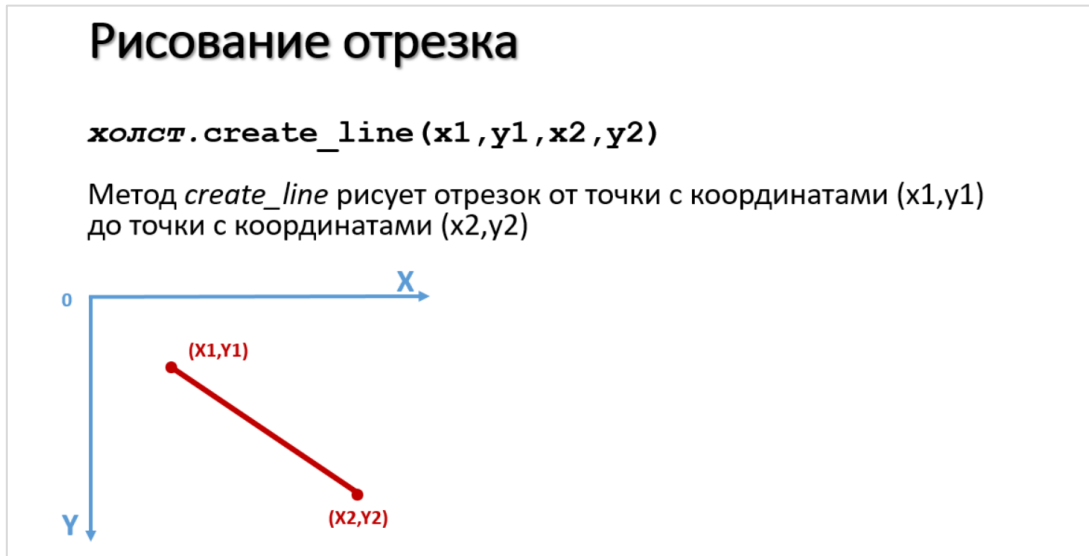
Система координат холста

Для размещения объектов (линий, прямоугольников, многоугольников и т.п.) указываются координаты их вершин на холсте. Точкой отсчета является левый верхний угол. У нас по оси x отсчет будет вестись до 800, по оси y – до 450.



Рисование линий (отрезков)

Выберем две точки, задав их координаты (x_1, y_1) и (x_2, y_2) . Линия на холсте рисуется с помощью метода холста `холст.create_line` – линия будет нарисована от точки (x_1, y_1) до точки (x_2, y_2) . Холст задается именем, мы используем имя `c`.



Кроме координат начала и конца отрезка можно указать и другие параметры: `холст.create_line(x1, y1, x2, y2, параметры)`. Параметры перечисляются через запятые, пробелы между ними можно ставить, можно не ставить. Пары координат можно взять в круглые или квадратные скобки. В качестве параметров можно указывать ширину линии, ее цвет.

В окне программы наберем следующий текст, где `c` – имя холста:

```
c.create_line(x1, y1, x2, y2, width=10, fill='red')
```

Кроме координат начала и конца линии указывается ширина (толщина) линии, это команда `width`, и цвет линии, это команда `fill`. Цвет можно указать в шестнадцатеричном коде или задать английским словом, обозначающим цвет, например, в нашей программе используется `red` – красный.

Выберем следующие точки начала и конца отрезка $(100, 100)$ и $(400, 400)$.

Программа будет иметь вид:

```
proba.py - D:/Лаборатория/Мероприятия_2022-1/06_Конкурс Питон кораблики/програ...
File Edit Format Run Options Window Help
from tkinter import *
window = Tk()
window.title('Корабли нашей мечты')
window.configure(background = '#ffffff')
window.geometry("800x450")
window.resizable(width=False, height=False)
c = Canvas(window, width=800, height=450, bg='#ffffff') #создает объект холста
c.pack() #отображает холст в окне
window.update()
c.create_line(100, 100, 400, 400, width=10, fill='red')
window.mainloop()
|
Ln: 12 Col: 0
```

В результате работы данной программы создается холст, на котором нарисована красная линия толщиной 10 пикселей.



Поэкспериментируйте с данной программой, задавая конкретные значения координат начала и конца отрезка (помните, что размер холста 800x450 и надо не «выскочить» за его пределы!), толщины и цвета линии.

Попробуйте нарисовать несколько линий с помощью следующих кодов:

```
c.create_line(150, 0, 300, 450, width=10, fill='red')
c.create_line(0, 0, 0, 450//2, width=20, fill='blue')
c.create_line(10, 450//2, 10, 450, width=20, fill='blue')
```

Каждый код задает рисование одной линии, попробуйте рисование линий, используя по очереди каждый из указанных кодов. В тексте программ для задания координат точек используется операция деления, она обозначается знаком //. Помните, что 450 – это высота нашего холста. В переводе с английского blue означает синий цвет.

Обратите внимание, что толщина линии имеет значение, линия рисуется в обе стороны по толщине, именно поэтому лучше задавать четную толщину линии и начинать рисовать не от нулевой координаты, а отступив половину ширины линии, в нашем случае толщина линии 10 и координата начала линии (5,5) вместо (0,0), если хотим рисовать линию из левого верхнего угла и чтобы вся линия была отображена.

Для изменения внешнего вида линий можно использовать параметры, определяющие вид линии (например, пунктирная), наличие стрелок на концах.

Более подробно о командах, задающих указанные параметры можно посмотреть в дополнительных материалах.

Параметры для рисования замкнутых фигур

При создании различных замкнутых фигур на холсте используются следующие параметры метода холста *create*:

Ширина обводки *width* = ширина обводки

Цвет обводки *outline* = цвет обводки

Цвет заливки *fill* = цвет заливки

Вид пунктирной обводки *dash* = (число1, число2), где число1 – длина штриха в пикселях, число2 – длина промежутка между штрихами в пикселях. Вид штриха (тире) определяется операционной системой компьютера.

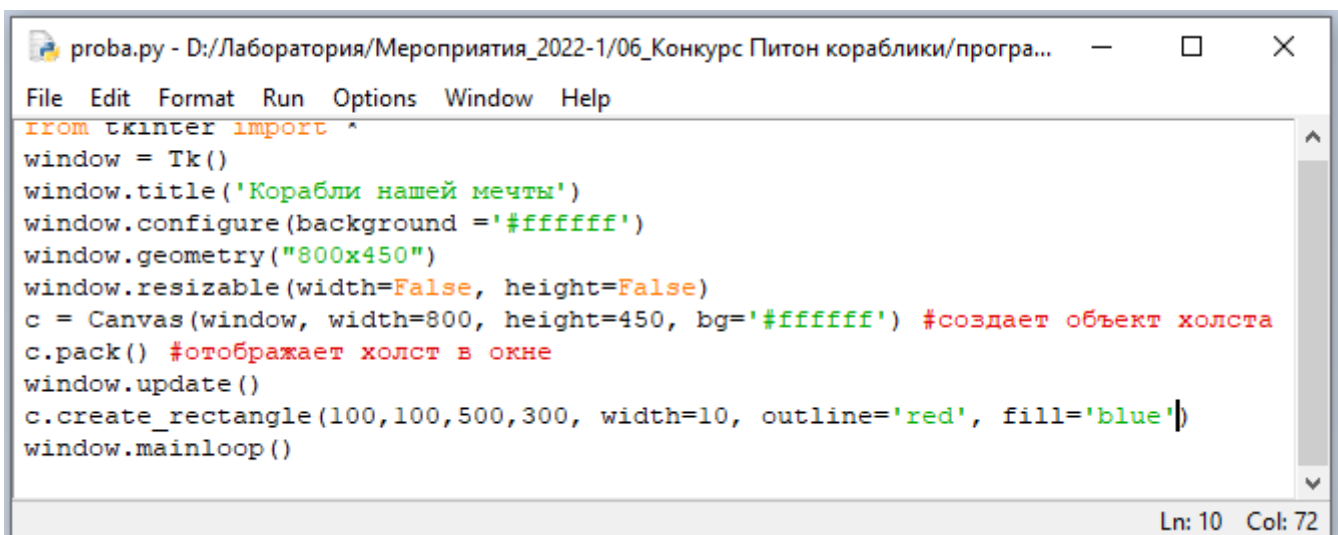
Рисование прямоугольников

Выберем две точки – левая верхняя вершина прямоугольника и правая нижняя вершина прямоугольника, задав их координаты (*x1*, *y1*) и (*x2*, *y2*). Прямоугольник на холсте рисуется с помощью метода холста *холст.create_rectangle(x1,y1,x2,y2, параметры)* – прямоугольник будет нарисован с вершинами в точках (*x1*, *y1*) и (*x2*, *y2*). Параметры могут отсутствовать. Могут использоваться параметры, описанные выше – ширина обводки границы, цвет обводки, цвет заливки, вид пунктирной обводки.

Например, следующая команда программы создаст на холсте с именем *c* прямоугольник с левой верхней вершиной в точке (100,100) и правой нижней вершиной в точке (500,300), красным цветом обводки толщиной 10 и синим цветом заливки:

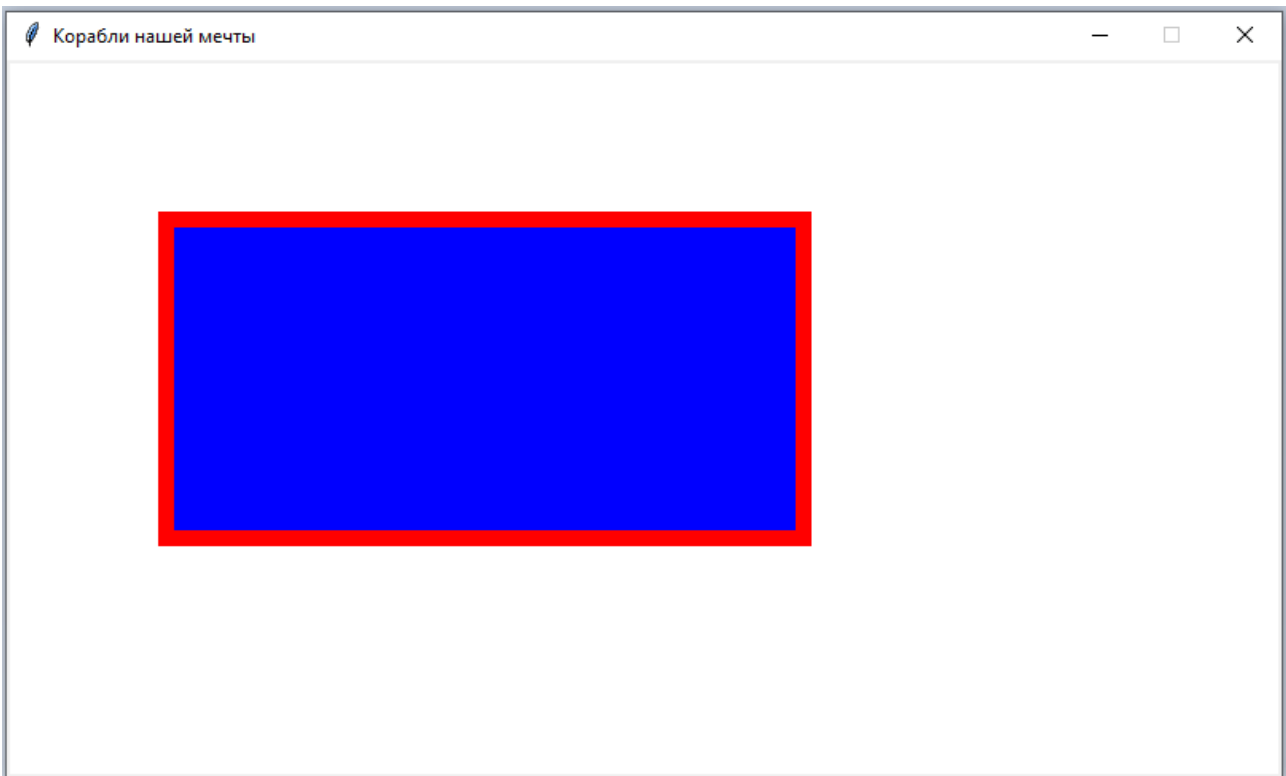
```
c.create_rectangle(100,100,500,300, width=10, outline='red', fill='blue')
```

Программа имеет вид:



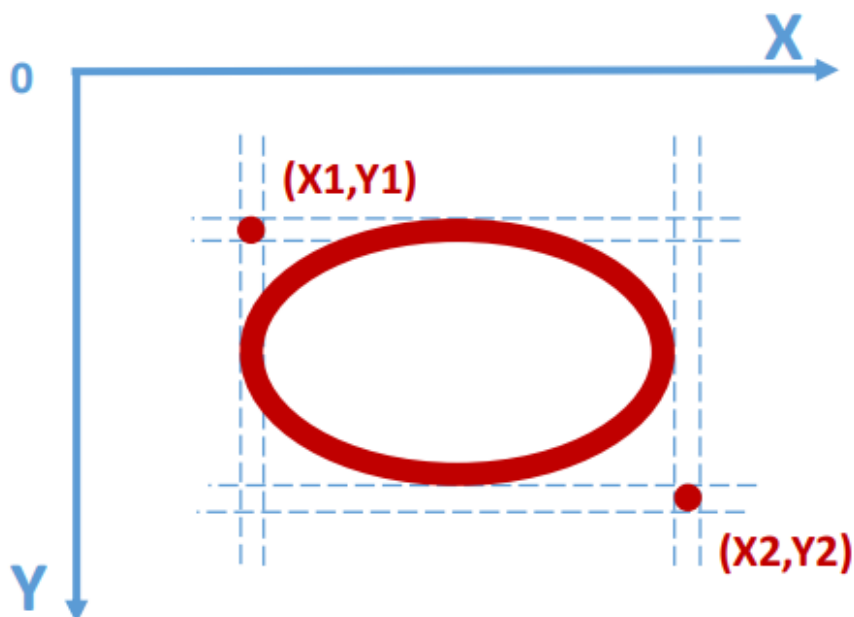
```
proba.py - D:/Лаборатория/Мероприятия_2022-1/06_Конкурс Питон кораблики/програ...
File Edit Format Run Options Window Help
from tkinter import *
window = Tk()
window.title('Корабли нашей мечты')
window.configure(background = '#ffffff')
window.geometry("800x450")
window.resizable(width=False, height=False)
c = Canvas(window, width=800, height=450, bg='#ffffff') #создает объект холста
c.pack() #отображает холст в окне
window.update()
c.create_rectangle(100,100,500,300, width=10, outline='red', fill='blue')
window.mainloop()
Ln: 10 Col: 72
```

В результате ее работы создается окно с холстом белого цвета размером 800x450 и нарисованным на нем прямоугольником с красной границей толщиной 10 пикселей и синей заливкой:



Рисование овалов

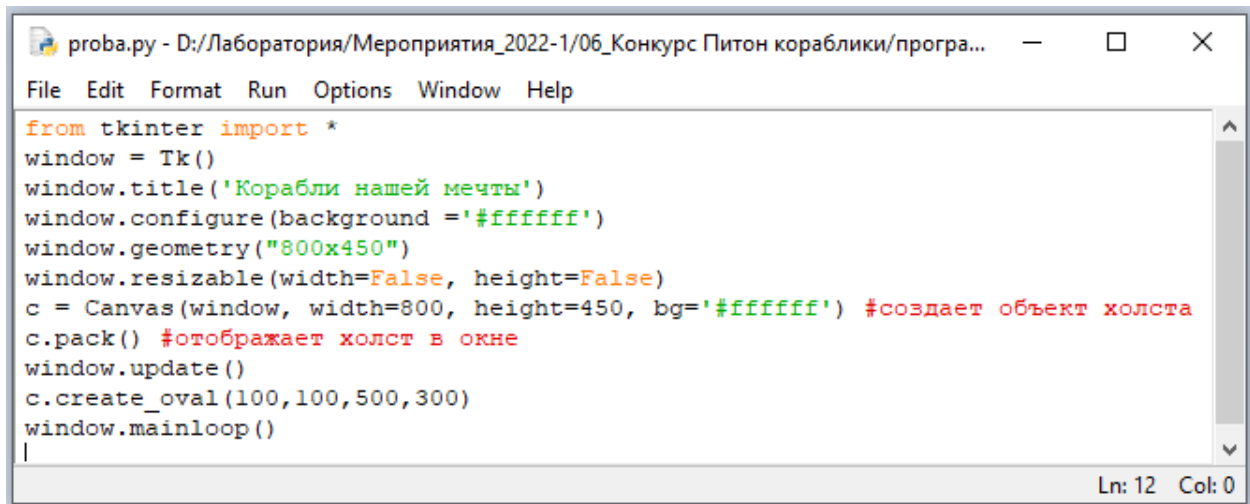
Овал на холсте рисуется с помощью метода холста `холст.create_oval(x1,y1,x2,y2, параметры)`. Овал вписывается в прямоугольник с координатами верхнего левого угла $(x1,y1)$ и нижнего правого угла $(x2,y2)$. Параметры могут быть указаны, например, цвет границы овала, а могут отсутствовать.



Для рисования овала на холсте необходимо добавить в программу следующую строку, где `s` – имя холста:

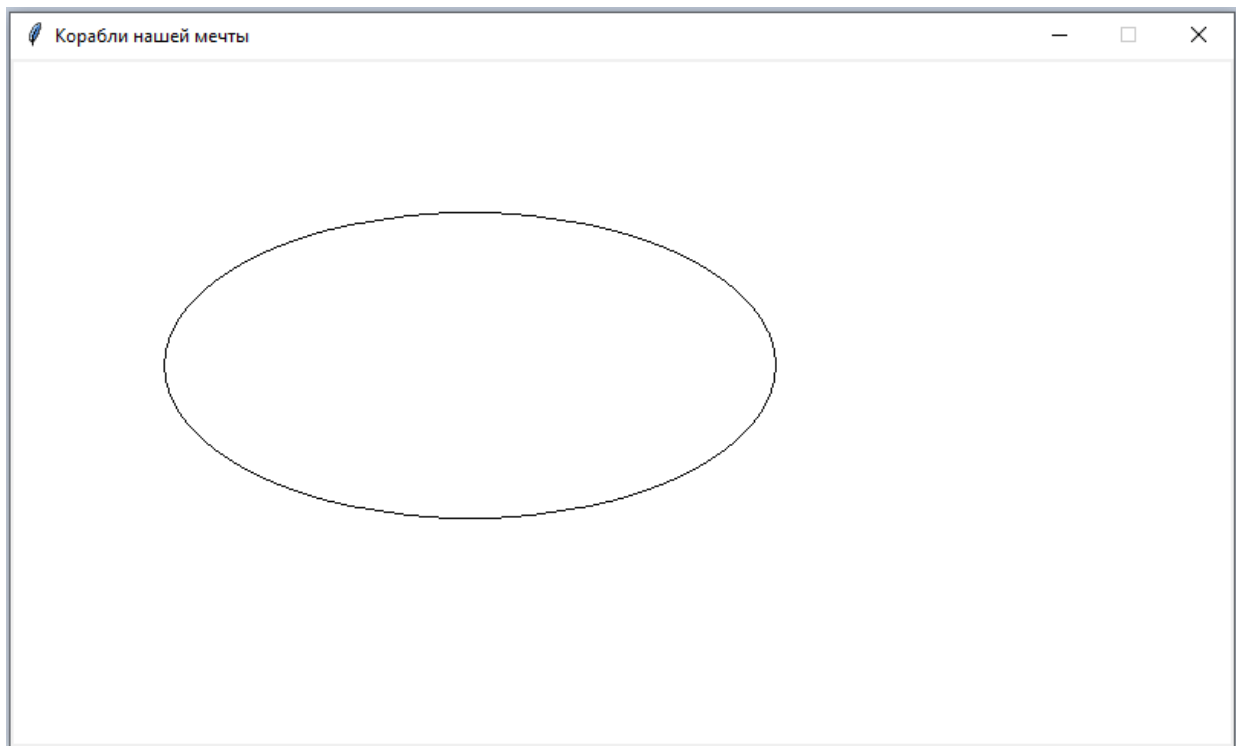
```
c.create_oval(100,100,500,300)
```

Программа имеет вид:



```
proba.py - D:/Лаборатория/Мероприятия_2022-1/06_Конкурс Питон кораблики/програ...
File Edit Format Run Options Window Help
from tkinter import *
window = Tk()
window.title('Корабли нашей мечты')
window.configure(background = '#ffffff')
window.geometry("800x450")
window.resizable(width=False, height=False)
c = Canvas(window, width=800, height=450, bg='#ffffff') #создает объект холста
c.pack() #отображает холст в окне
window.update()
c.create_oval(100,100,500,300)
window.mainloop()
|
Ln: 12 Col: 0
```

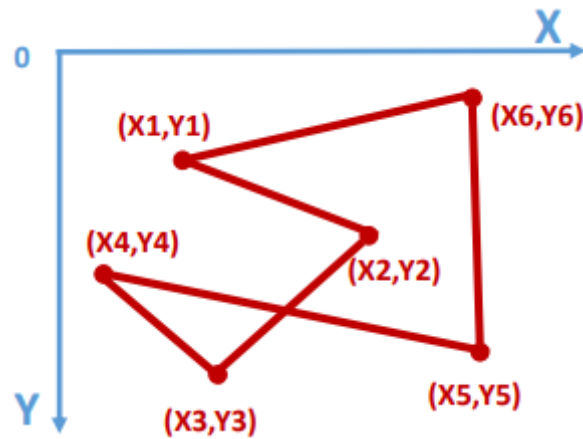
В результате работы программы получится окно с нарисованным овалом:



Овал получился без заливки, граница представляет собой тонкую черную линию, это параметры по умолчанию. Для изменения внешнего вида овала надо указать параметры. Поэкспериментируйте с данной программой, меняя параметры и запуская программу на выполнение. Например, можно добавить `width=10` (толщина границы фигуры, `outline='red'` (цвет границы), `fill='blue'` (цвет заливки).

Рисование многоугольников

Многоугольник на холсте рисуется с помощью метода холста `холст.create_polygon(x1,y1,x2,y2, параметры)`. Метод `create_polygon` рисует многоугольник по координатам. Чтобы получилась фигура, таких точек должно быть минимум три, каждая точка задается координатами `x` и `y`.

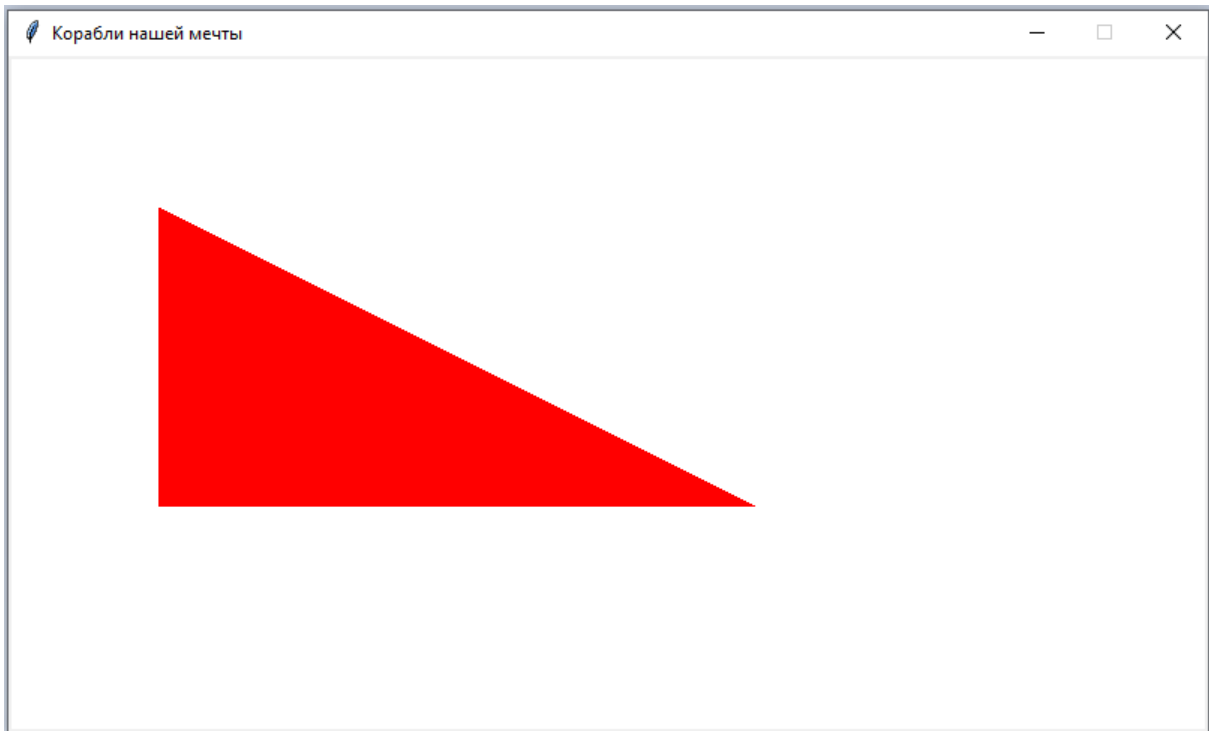


В качестве параметров можно указывать цвет границы, толщину границы, цвет заливки. Параметры можно не указывать, тогда будет нарисован многоугольник с черной границей толщиной 1 пиксель и белой заливкой по умолчанию.

Давайте добавим в программу следующую строку:

```
c.create_polygon(100, 100, 500, 300, 100, 300, fill='red')
```

В результате будет нарисован прямоугольный треугольник красного цвета.

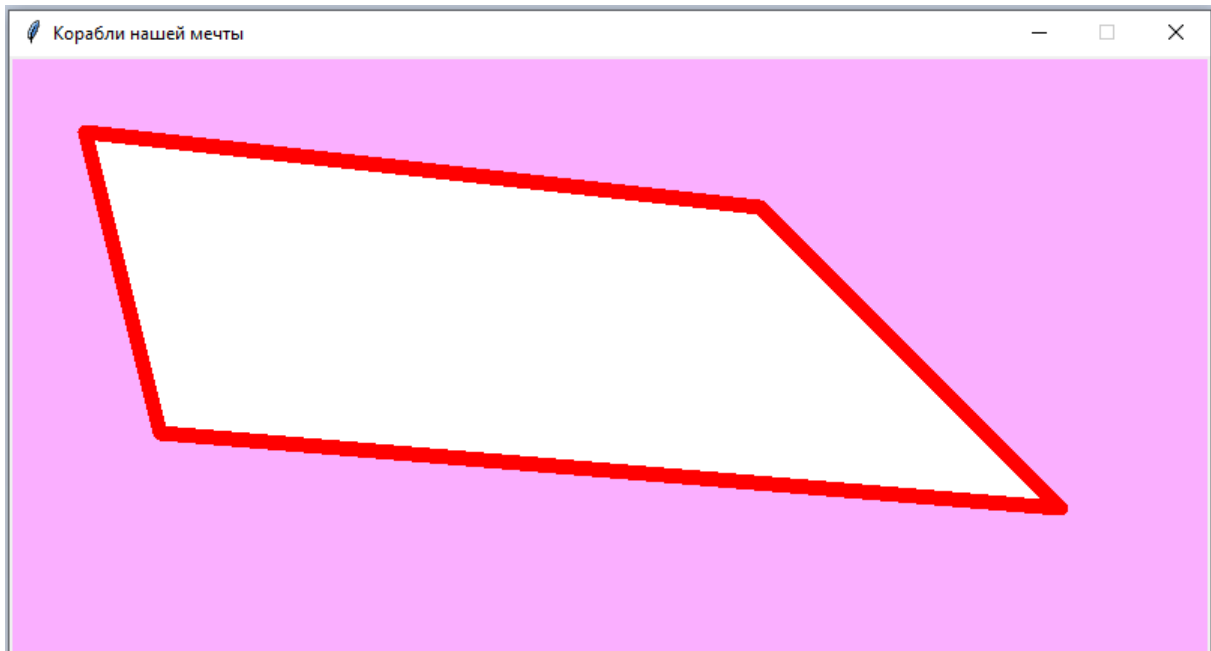


Дополнительную информацию о параметрах, которые можно применять при создании многоугольников, можно изучить в дополнительных материалах.

Задание для самопроверки

Ответьте на следующий **вопрос**.

Программа в результате работы выдала следующий результат:



Далее дан текст этой программы, часть текста программы исчезла. Проанализируйте текст программы и определите, что пропущено в тексте программы, которая выдала представленный выше результат.

```
*proba.py - D:/Лаборатория/Мероприятия_2022-1/06_Конкурс Питон кораблики/программы/про...
File Edit Format Run Options Window Help
from tkinter import *
window = Tk()
window.title('Корабли нашей мечты')
window.configure(background = '#ffffff')
window.geometry("800x450")
window.resizable(width=False, height=False)
c = Canvas(window, width=800, height=450, bg='#faafff')
c.pack()
window.update()
c.create_(50, 50, 500, 100, 700, 300, 100, 250, fill='white', outline='red', width=10)
window.mainloop()
|
Ln: 12 Col: 0
```

Удачной работы!