

Графика в Python.

С помощью **графики в Python** можно рисовать фигуры и изображения, создавать анимацию, визуализировать математические вычисления в Python. В программах python можно использовать элементы графики в компьютерных играх.

Для работы с **графикой в Python** нужно импортировать модуль `graphics.py`. [Как установить модуль graphics в Python.](#)

Чтобы начать работу с **графикой в Python**, нужно создать окно для графики

Графический объект = `GraphWin("Название окна для графики", ширина окна для графики в пикселях, высота окна для графики в пикселях)`  
`GraphWin` это ключевое слово, которое задаёт окно графической области, в котором будут отображаться графические объекты.

В качестве параметров этой функции указывается название окон для графики, ширина и высота окон в пикселях.

После запуска программы откроется окно для графики, где будут отображаться графические объекты.

Вся работа с графикой будет осуществляться нами через графические объекты.

Общая структура работы с графическими объектами в Python

Графический\_объект.Вызов\_команды()

Общая структура графической программы в Python.

```
# импортируем библиотеку graphics
```

```
from graphics import *
```

```
# создаём окно для графики
```

```
win = GraphWin("Окно для графики", 400, 400)
```

```
# ...рисуем все объекты...
```

```
win.getMouse() # ждём нажатия кнопки мыши
```

```
win.close() # закрываем окно для графики
```

В этой программе мы определили объект графическое окно `win` и открыли его с размерами 400 на 400 пикселей.

Команда `win.getMouse()` ожидает нажатие на любую кнопку мыши, наведённую на область окна `win`.

`win.close()` закрывает окно для графических объектов `win`.

С помощью модуля `graphics.py` в программах на Python можно отобразить точку, линию, окружность, прямоугольник, эллипс и многоугольник, вывести текст на экран.

Чтобы задать расположение объекта в графическом окне Python, необходимо указать его координаты в системе координат Python. Начало

координат находится в левом верхнем углу окна для графики. Положительное направление оси X определяется слева направо, оси Y определяется сверху вниз. Чем больше значение координаты X, тем правее точка, чем больше значение координаты Y, тем точка ниже. Чтобы нарисовать заданный объект `obj` в окне для графики `win`, нужно использовать процедуру `obj.draw(win)`

Перед тем, как рисовать графические объекты в заданном графическом окне, нужно их задать.

Для задания точки в Python используется функция `Point(x, y)`

```
obj = Point(x, y)
```

`x, y` – координаты точки.

Пример программы на Python, которая задаёт и отображает точку в графическом окне.

```
from graphics import * # импортируем библиотеку graphics
win = GraphWin("Окно для графики", 400, 400) # создаём окно для графики
размером 400 на 400 пикселей
obj = Point(50, 50) # создаём точку в координатах (50, 50)
obj.draw(win) # отображаем точку в окне для графики
win.getMouse() # ждём нажатия кнопки мыши
win.close() # закрываем окно для графики
```

Для задания отрезка в Python используется функция `Line(объект точка первого конца, объект точка второго конца)`

```
obj = Line(Point(x1, y1), Point(x2, y2))
```

`x1, y1` – координаты начала отрезка линии,

`x2, y2` – координаты конца отрезка линии.

Чтобы задать цвет рисования линий в Python используется команда `obj.setOutline("цвет")`

Пример программы на Python, которая отображает линию в графическом окне.

```
from graphics import *
win = GraphWin("Окно для графики", 400, 400)
obj = Line(Point(50, 50), Point(350, 350))
obj.setOutline("blue")
obj.draw(win)
win.getMouse()
win.close()
```

Для отображения окружности в Python используется

```
obj = Circle(Point(x, y), R)
```

`x, y` – координаты центра окружности,

R – радиус окружности.

Пример программы на Python, которая отображает окружность в графическом окне.

```
from graphics import *
win = GraphWin("Окно для графики", 400, 400)
obj = Circle(Point(200, 200), 50)
obj.draw(win)
win.getMouse()
win.close()
```

Для отображения прямоугольника в Python используется процедура

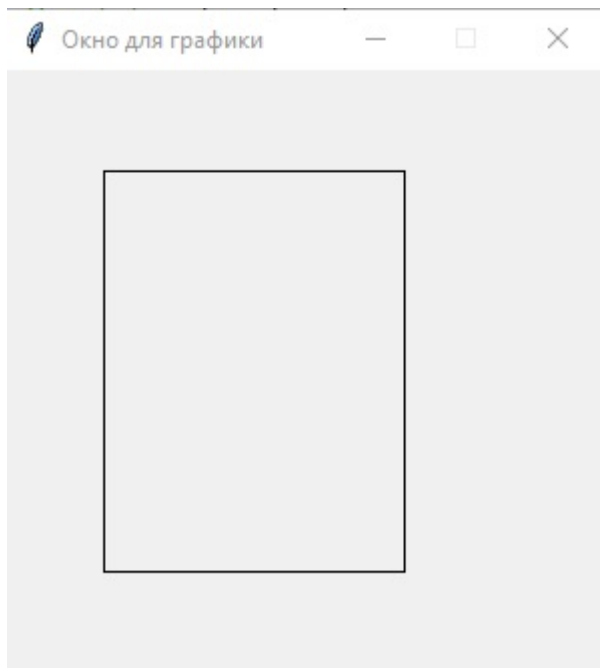
```
obj = Rectangle(Point(x1, y1), Point(x2, y2))
```

x1, y1 – координаты левого верхнего угла прямоугольника,

x2, y2 – координаты правого верхнего угла прямоугольника

Пример программы на Python, которая отображает прямоугольник в графическом окне.

```
from graphics import *
win = GraphWin("Окно для графики", 300, 300)
obj = Rectangle(Point(50, 50), Point(200, 250))
obj.draw(win)
win.getMouse()
win.close()
```



Для отображения эллипса в Python используется процедура

```
obj = Oval(Point(x1, y1), Point(x2, y2))
```

$x_1, y_1$  – координаты первого фокуса эллипса,

$x_2, y_2$  – координаты второго фокуса эллипса.

Пример программы на Python, которая отображает эллипс в графическом окне.

```
from graphics import *
```

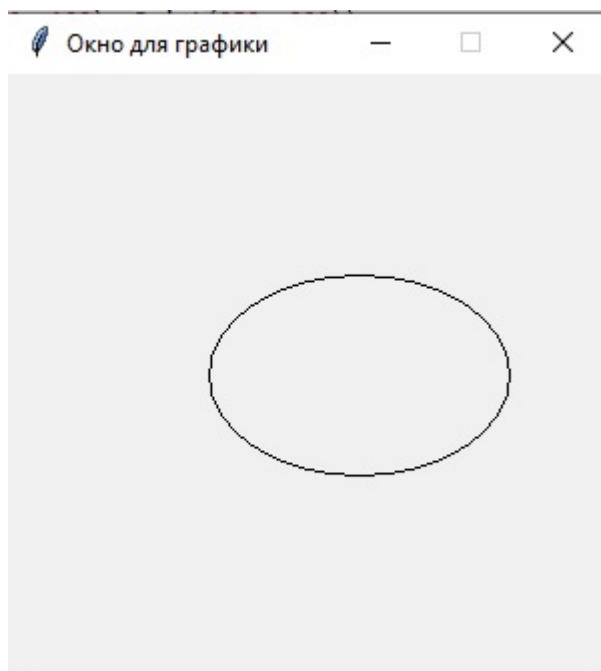
```
win = GraphWin("Окно для графики", 300, 300)
```

```
obj = Oval(Point(100, 100), Point(250, 200))
```

```
obj.draw(win)
```

```
win.getMouse()
```

```
win.close()
```



Для отображения многоугольника в Python используется процедура

```
obj = Polygon(Point(x1, y1), Point(x2, y2),..., Point(xn, yn))
```

$x_1, y_1, x_2, y_2, \dots, x_n, y_n$  – координаты вершин многоугольника.

Пример программы на Python, которая отображает пятиугольник в графическом окне.

```
from graphics import *
```

```
win = GraphWin("Окно для графики", 400, 400)
```

```
obj = Polygon(Point(10, 10), Point(300, 50), Point(200, 300), Point(150, 150),  
Point(70, 70))
```

```
obj.draw(win)
```

```
win.getMouse()
```

```
win.close()
```

Определение цвета закрашивания графического объекта в Python

Чтобы задать цвет закрашивания графического объекта в python используется команда `obj.setFill("цвет")`

**Пример программы на Python, которая рисует закрашенную синюю окружность**

```
from graphics import *
win = GraphWin("Окно для графики", 400, 400)
obj = Circle(Point(200, 200), 50)
obj.setFill("blue")
obj.draw(win)
win.getMouse()
win.close()
```

Для редактирования границ объектов в Python используются процедуры `setOutline("цвет границы")` и `setWidth(ширина границы)`.

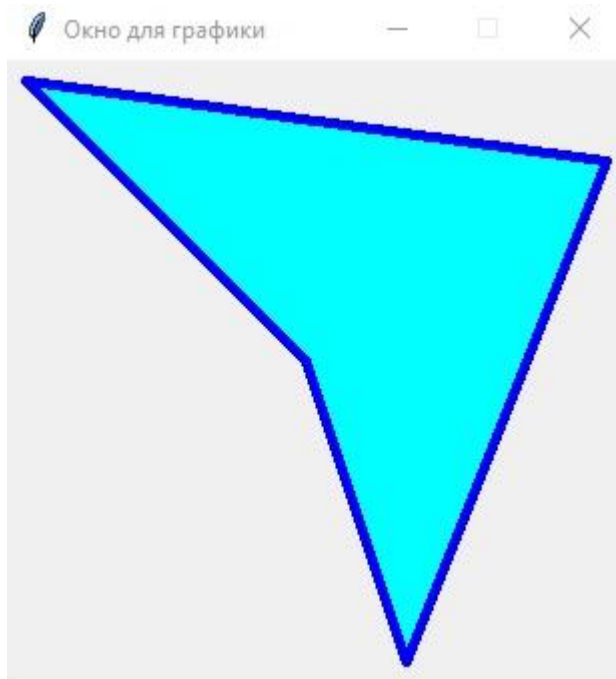
`obj.setOutline("blue")` – объект `obj` отображается с границей синего цвета.

`obj.setWidth(5)` – объект `obj` отображается с шириной границы 5 пикселей.

По умолчанию графический объект в Python будет отображаться с границами чёрного цвета шириной 1 пиксель.

Пример программы на Python, которая отображает фигуру с синей границей и заливкой в графическом окне.

```
from graphics import *
win = GraphWin("Окно для графики", 310, 310)
obj = Polygon(Point(10, 10), Point(300, 50), Point(200, 300), Point(150, 150),
Point(70, 70))
obj.setOutline("blue")
obj.setWidth(5)
obj.setFill("cyan")
obj.draw(win)
win.getMouse()
win.close()
```



**Чтобы переместить графический объект в Python**, используется процедура `move(dx, dy)`, которая перемещает объект на `dx` пикселей вправо и `dy` пикселей вниз.

`obj.move(50, 50)` смещает объект `obj` на 50 пикселей вправо и 50 пикселей вниз.

Для клонирования объектов используется процедура `clone()`

```
newObj = obj.clone()
```

С помощью этой команды создаётся новый графический объект `newObj`, который идентичен объекту `obj`.

Для удаления фигур с экрана используется процедура `undraw()` Объект удаляется с графического окна, но не удаляется из памяти.

```
obj.undraw()
```

Пример программы на Python, которая удаляет, перемещает и копирует объект в графическом окне.

```
from graphics import *
win = GraphWin("Окно для графики", 400, 400)
obj = Polygon(Point(30, 10), Point(30, 50), Point(20, 30), Point(15, 30),
Point(7, 7))
obj.setOutline("blue")
obj.setWidth(2)
obj.setFill("cyan")
obj.draw(win)
win.getMouse()
obj.undraw()
win.getMouse()
```

```
obj.draw(win)
obj.move(100, 100)
win.getMouse()
shape = obj.clone()
shape.move(-100, -100)
shape.draw(win)
win.getMouse()
win.close()
```

Для создания текста в графическом окне в Python используется команда

текстовый объект=Text(координаты точки размещения текста, "Текст")  
msg = Text(Point(50, 100), "Hello World!")

На экран в точке с координатами (50, 100) выведется текст со строкой Hello World!

Для изменения размера текста используется команда текстовый объект.setSize(размер текста)

```
msg.setSize(12)
```

Цвет текста изменяется с помощью метода setTextColor(цвет)

```
msg.setTextColor("black")
```

Текст в графическом объекте можно заменить с помощью метода setText("Текст")

```
msg.setText("Другой текст")
```

Стиль текста изменяется с помощью процедуры setStyle(стиль)

```
msg.setStyle("bold")
```

Стиль normal изменяет стиль текста на обычный, bold меняет стиль на полужирный, italic меняет стиль на курсив, bold italic меняет стиль текста на полужирный курсив.

Пример программы на Python, которая отображает текст в графическом окне.

```
from graphics import *
win = GraphWin("Окно для графики", 400, 400)
obj = Polygon(Point(10, 10), Point(300, 50), Point(200, 300), Point(150, 150),
Point(70, 70))
obj.setOutline("blue")
obj.setWidth(5)
obj.setFill("cyan")
obj.draw(win)
win.getMouse()
obj.undraw()
msg = Text(Point(200, 200), "Фигура удалилась с экрана")
```

```
msg.setSize(12)
msg.setTextColor("black")
msg.setStyle("bold italic")
msg.draw(win)
win.getMouse()
win.close()
```

Задание рассчитано на один полный урок 13.04.2020