**[Раздел «Язык Си»](http://acm.mipt.ru/twiki/bin/view/Cintro/WebHome).**[**TurtleGeomxy**](http://acm.mipt.ru/twiki/bin/view/Cintro/TurtleGeomxy)**:**

* [Работа с координатами в Turtle](http://acm.mipt.ru/twiki/bin/view/Cintro/TurtleGeomxy#_cdq_c___mqq_fkpc_cok___Turtle)
  + [Пример 1: как работают с координатами](http://acm.mipt.ru/twiki/bin/view/Cintro/TurtleGeomxy#__kog__1_mcm__cdq_c_____mqq_fkpc)
  + [Пример 2: line(x1, y1, x2, y2) - линия по точкам](http://acm.mipt.ru/twiki/bin/view/Cintro/TurtleGeomxy#__kog__2_line_x1_y1_x2_y2_nkpks_)
* [Задачи](http://acm.mipt.ru/twiki/bin/view/Cintro/TurtleGeomxy#_cfc_k)

# Работа с координатами в Turtle

На поле черепахи есть (их не видно) оси Х и У.

Черепаха сначала в точке (0, 0).

Можно узнать координаты где стоит черепаха.

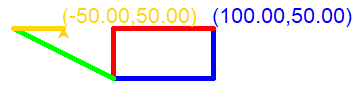
|  |  |
| --- | --- |
| [**Функция**](http://acm.mipt.ru/twiki/bin/view/Cintro/TurtleGeomxy?sortcol=0&table=1&up=0#sorted_table) | [**Что возвращает**](http://acm.mipt.ru/twiki/bin/view/Cintro/TurtleGeomxy?sortcol=1&table=1&up=0#sorted_table) |
| t.xcor() | x координата черепахи |
| t.ycor() | x координата черепахи |
| t.pos() | (x, y) точка, где стоит черепаха |

Можно передвинуть черепаху в другую точку. Если черепаха рисовала (pd), то при движении она будет тоже рисовать.

Направление головы черепахи не поменяется. Она может двигаться вбок или назад, не меняя направление.

|  |  |
| --- | --- |
| [**Функция**](http://acm.mipt.ru/twiki/bin/view/Cintro/TurtleGeomxy?sortcol=0&table=2&up=0#sorted_table) | [**Что делает**](http://acm.mipt.ru/twiki/bin/view/Cintro/TurtleGeomxy?sortcol=1&table=2&up=0#sorted_table) |
| t.setx(x) | не изменяет y, двигается в точку с этим х |
| t.sety(y) | не изменяет x, двигается в точку с этим y |
| t.setpos(p) t.setpos((x,y)) | двигает черепаху в точку p или в точку (x, y) |

## Пример 1: как работают с координатами



import turtle

fb = ("Arial", 16, "normal")

t = turtle.Turtle()

t.width(5)

t.color("blue")

# начинаем движение в точке (0, 0)

t.fd(100)

t.lt(90)

t.fd(50)

# узнать координаты можно следующими методами:

p = t.pos() # координаты черепахи в виде точки (x, y)

x = t.xcor() # x координата черепахи

y = t.ycor() # y координата черепахи

print(x, y) # 100 50 - печатаем

t.write(p, font=fb) # пишем на рисунке координаты

# сдвинем черепаху в точку с той же координатой y, изменим x=0

t.color("red")

t.setx(0)

# сдвинем черепаху в точку с той же координатой x, изменим y = 0

t.color("red")

t.sety(0)

# сдвинем черепаху в точку с координатами (-100, 50)

t.color("green")

t.setpos((-100, 50)) # внимание, координаты пишем со скобками (-100, 50)

# сдвинем черепаху НА +50 по оси Х:

t.color("gold")

x = t.xcor()

t.setx(x + 50)

t.write(t.pos(), font=fb) # пишем на рисунке координаты

turtle.done()

## Пример 2: line(x1, y1, x2, y2) - линия по точкам

Напишем функцию:

* **goto(x, y)** (передвигает черепаху в точку (x,y), ничего не рисует)
* **line(x1, y1, x2, y2)** - рисует отрезок из точки (x1, y1) в точку (x2, y2)

Нарисуем с их помощью треугольник с вершинами в точках (30, -50), (200, 0) и (100, 150). Заметьте, что линии можно рисовать в любом порядке. Не обязательно соединять вершины одну за другой.

# -\*- coding: utf-8 -\*-

import turtle

fb = ("Arial", 16, "normal")

# передвигаем черепаху в точку (x, y), НЕ рисуем

def goto(x, y):

t.pu()

t.setpos((x, y))

# рисуем отрезок с концами (x1, y1) и (x2, y2)

def line(x1, y1, x2, y2):

t.pu()

t.setpos((x1, y1))

t.pd()

t.setpos((x2, y2))

t = turtle.Turtle()

t.width(5)

t.color("blue")

# рисуем треугольник

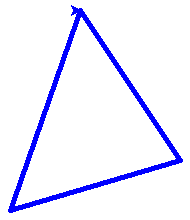
# линии можно рисовать в любом порядке

line(30, -50, 200, 0)

line(30, -50, 100, 150)

line(200, 0, 100, 150)

turtle.done()

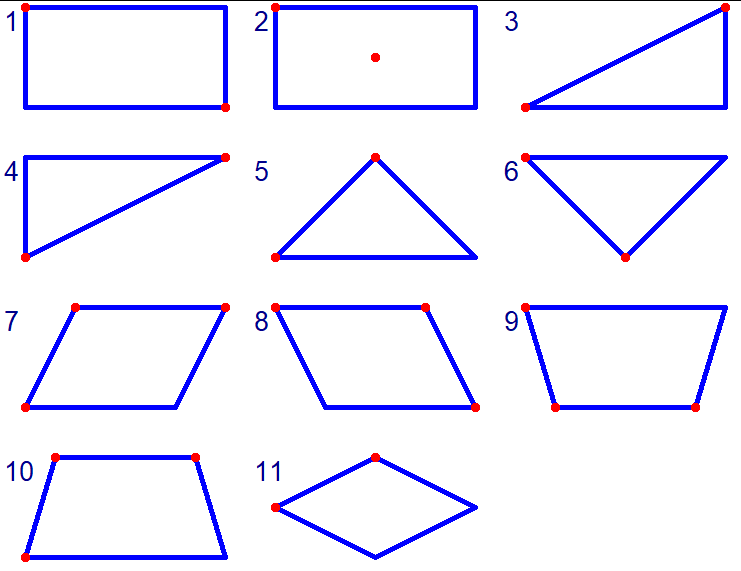


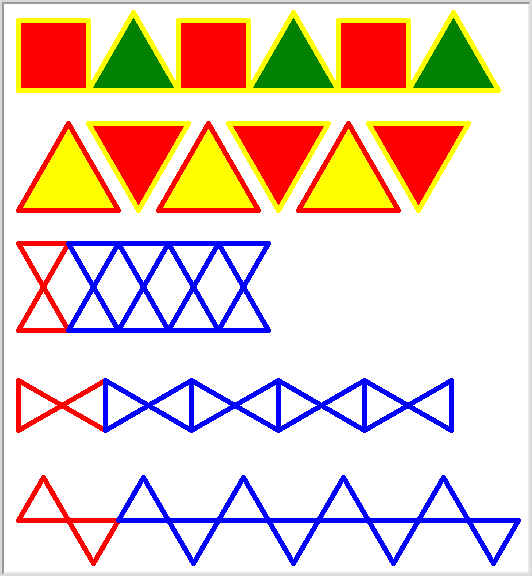
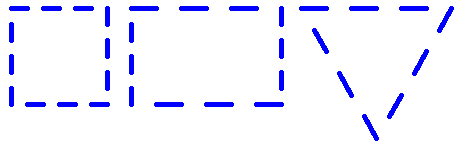
# Задачи

Даны координаты нескольких точек. Написать функцию, которая по этим точкам рисует фигуру, как на рисунке. Точки рисовать не надо.

Варианты:

1. **rect(x1, y1, x2, y2)** - прямоугольник по левой верхней (x1, y1) и правой нижней точкам.
2. **rect(x1, y1, xс, yс)** - прямоугольник по левой верхней (x1, y1) точке и пересечению диагоналей (xc, yc).
3. **tri90(x1, y1, x2, y2)** - прямоугольный треугольник по вершинам гипотенузы (x1, y1) и (x2, y2). Катеты параллельны осям Х и У.
4. **tri90(x1, y1, x2, y2)** - прямоугольный треугольник по вершинам гипотенузы (x1, y1) и (x2, y2). Катеты параллельны осям Х и У.
5. **tri2(x1, y1, x2, y2)** - равнобедренный треугольник (боковые стороны равны); вершины боковой стороны в точках (x1, y1) и (x2, y2). Основание параллельно оси Х.
6. **tri2(x1, y1, x2, y2)** - равнобедренный треугольник (боковые стороны равны); вершины боковой стороны в точках (x1, y1) и (x2, y2). Основание параллельно оси Х.
7. **par(x1, y1, x2, y2, x3, y3)** - параллелограмм (противоположные стороны параллельны). 2 стороны параллельны оси Х.
8. **par(x1, y1, x2, y2, x3, y3)** - параллелограмм (противоположные стороны параллельны). 2 стороны параллельны оси Х.
9. **trap(x1, y1, x2, y2, x3, y3)** - равнобедренная трапеция. Боковые стороны равны. Основания параллельны оси Х.
10. **trap(x1, y1, x2, y2, x3, y3)** - равнобедренная трапеция. Боковые стороны равны. Основания параллельны оси Х.
11. **romb(x1, y1, x2, y2, x3, y3)** - ромб, все стороны равны. Диагонали параллельны осям Х и У.



* узоры, вариант 1-5:   
  
* узоры, вариант 6-8:   
  
* узоры, вариант 9-11:   
  