**Функции возвращают значения**

Иногда нужно узнать какие числа или другие значения были у функции, когда она работала.

Тогда можно сказать функции, чтобы она ВЕРНУЛА ( **return** ) это нужное нам значение

Пример. Нарисуем квадрат, вычислим его середину и вернем это значение

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

import turtle

import time

def write(data):

t.write(data, font=("Arial", 14, "normal"))

def carre(size, col):

p0 = t.pos() # запомним от какой точки будем рисовать

t.color(col)

t.down()

# вычислим координаты середины

xc = size/2

yc = size/2

# нарисуем квадрат

for i in range(4):

t.fd(size)

t.left(90)

t.up()

t.goto(xc, yc) # встанем на центр

centr = t.pos() # запомним точку центра

t.goto(p0) # вернемся в начальную точку

return centr # ВЕРНЕМ ЗНАЧЕНИЕ центр

t = turtle.Turtle()

t.shape("turtle")

t.width(3)

pc = carre(100, "blue") # Нарисуем квадрат и получим точку центра pc

# вычислять ее больше не нужно

# pc вычисляет функция carre()

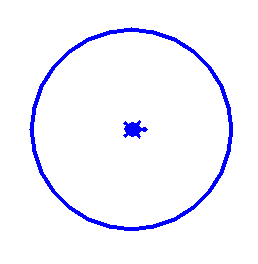
t.goto(pc) # поставим черепаху на этот центр

write(pc)

turtle.done()

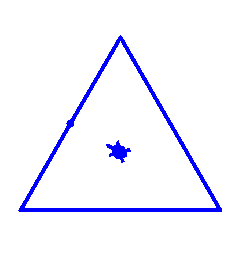
**Задача 1. Центр окружности**

Написать функцию **def circR(radius, color)**, которая ВОЗВРАЩАЕТ точку центра окружности. Получить эту точку в программе и поставить туда черепаху.



**Задача 2 Центр треугольника**

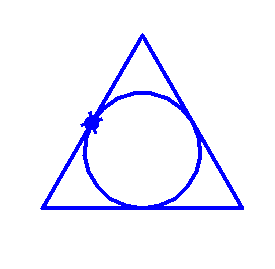
Написать функцию **def trC(size, color)**, которая рисует правильный треугольник и возвращает точку центра этого треугольника. Получить эту точку в программе и поставить туда черепаху.



**Задача 3 Вписанная окружность**

Написать функцию **def trOk(size, color)**, которая рисует правильный треугольник и возвращает радиус вписанной окружности. После работы функции черепаха стоит на половине стороны и повернута по направлению движения.

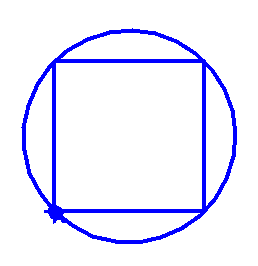
В основной программе нарисовать вписанную окружность



**Задача 4 Описанная окружность**

Написать функцию **def carre(size, color)**, которая рисует квадрат и возвращает радиус описанной окружности. После работы функции черепаха стоит в углу квадрата и повернута перпендикулярно диагонали.

В основной программе нарисовать описанную окружность



**Условия.**

Чтобы выбрать какое действие выполнить есть УСЛОВИЯ

Например. Если размер квадрата равен 0, то не будем его рисовать.

def carre(size, col):

t.down()

t.color(col)

if size == 0:

write("не буду рисовать")

# здесь функция сразу перестанет работать

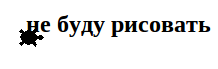
return t.pos()

# а это будет выполнять если не 0

for n in range(4):

t.fd(size)

t.left(90)



Можно сравнить и по-другому

def carre(size, col):

t.down()

t.color(col)

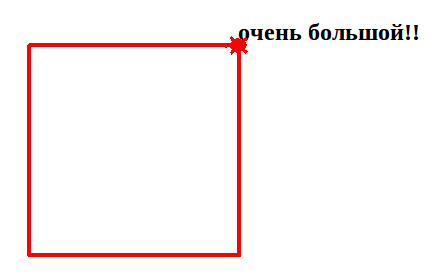
if size > 200 :

write("очень большой!")

for n in range(4):

t.fd(size)

t.lt(90)



Можно и так написать:

def carre(size, col):

t.down()

t.color(col)

if 0 < size and size < 200 :

write("правильный")

else:

write("плохие размеры")

for n in range(4):

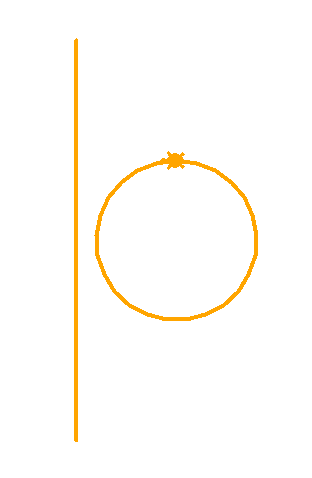
t.fd(size)

t.lt(90)

|  |  |
| --- | --- |
| http://acm.mipt.ru/twiki/pub/Cintro/PythonTurtleIfFor1/if3.png | http://acm.mipt.ru/twiki/pub/Cintro/PythonTurtleIfFor1/if4.png |

**Задача 5 Круг слева**

Написать функцию **def line(x)**, которая рисует вертикальную линию с координатой **x**. Написать функцию **def circRight(центр, радиус, х)**, которая рисует круг, если он будет полностью находится СПРАВА от линии. Если круг находится справа от линии или пересекает линию, пишет **Не буду рисовать**.



**Задача 6 Разные квадраты**

Функция **carre(size, color)** рисунт квадрат цвета color. Функция **carres(size1, size2, color1, color2)** рисует два квадрата цветом color1 и color2. Если цвета совпадают, второй квадрат будет не видно.

Напишем функцию так, чтобы поменять красный и желтый цвета:

def carre2(size1,size2,color1,color2):

if color1 == color2:

if color1 == "red":

color2 = "yellow"

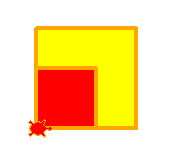
else :

color2 = "red"

carre(size1, color1)

carre(size2, color2)

Дописать функцию так, чтобы поменять все цвета. Проверить ее работу



**Задача 7 Черточки**

Написать функцию **def dash(size, step)**, которая рисует черточки размером step на полоске размером size. Цвет черточек меняется: нечетная - красная, четная - зеленая



**Задача 8 Вложенные квадраты**

Написать функцию def carreInf(size, skolko), которая рисует skolko вложенных цветных квадратов. Размер стороны квадрата каждый раз уменьшается на 20. Если размер становится равен 0, то рисование прекращается. Написать сколько получилось квадратов

|  |  |
| --- | --- |
| вызов | взов |
| carreInf(150,14) | carreInf(100,5) |
| http://acm.mipt.ru/twiki/pub/Cintro/PythonTurtleIfFor1/carreinf.png | http://acm.mipt.ru/twiki/pub/Cintro/PythonTurtleIfFor1/carreinf2.png |

**Задача 9 "Шахматная" доска**

Написать функцию **def chessLine(size, step, firstColor)** , которая рисует size/step клеточек размера step чередуя цвета клеток. Функция должна возвращать цвет последнего квдратика.

|  |  |
| --- | --- |
| http://acm.mipt.ru/twiki/pub/Cintro/PythonTurtleIfFor1/lp1.png | http://acm.mipt.ru/twiki/pub/Cintro/PythonTurtleIfFor1/lp2.png |

Написать функцию **chessPole(h,w,step)** , которая заполняет прямоугольник размером h x w клеточками размера size. При этом цвета должны чередоваться в шахматном порядке

