

Инструкции



Только просмотр: 11 / 11 блоки

при запуске

здать цвет

здать ширину

1

для counter от 15 до 300 с шагом 2

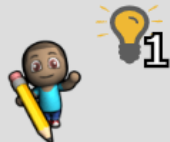
передвинуться вперед на counter пикселей

повернуть вправо на 89 градусов



Инструкции

Вспомни предыдущий этап, когда мы использовали переменную для управления числом сторон фигуры? Теперь мы собираемся выполнить аналогичное действие с переменной «счетчик». В основе этого шаблона лежит треугольник, к которому добавляется по одной стороне за раз, пока фигура не станет десятиугольником (10 сторон). Длина каждой стороны должна быть 100 пикселей. Подсказка: Разделив 360 градусов на число сторон фигуры, ты получишь количество градусов, на которое нужно поворачивать.



Только просмотр: 13 / 13 блоки

</> Показать код

при запуске

для counter от 3 до 10 с шагом 1

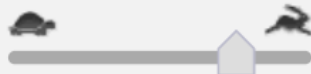
повторить counter раз

выполнить

передвинуться вперед на 100 пикселей

повернуть вправо на $\frac{360}{\text{counter}}$ градусов

▶ Выполнить



Русский ▼

Программирование на языке Python

Процедуры



Процедура – вспомогательный алгоритм, который выполняет некоторые действия.

- *текст (расшифровка) процедуры записывается **до** её вызова в основной программе*
- *в программе может быть **много** процедур*
- *чтобы процедура заработала, нужно **вызвать** её по имени из основной программы или из другой процедуры*

Задача. Вывести на экран запись целого числа (0..255) в 8-битном двоичном коде.

много раз!

Алгоритм:

178 \Rightarrow 10110010₂

? Как вывести первую цифру?

n := ^{7 6 5 4 3 2 1 0}
1 0 1 1 0 0 1 0₂ разряды

n // 128

n % 128

? Как вывести вторую цифру?

n1 // 64

Задача. Вывести на экран запись целого числа (0..255) в 8-битном двоичном коде.

Решение:

```
k = 128
while k > 0:
    print ( n // k,
            end = "" )
    n = n % k
    k = k // 2
```

178 \Rightarrow 10110010



Результат зависит
от **n**

n	k	ВЫВОД
178	128	1
50	64	0
50	32	1
18	16	1
2	8	0
2	4	0
2	2	1
0	1	0
0	0	

Параметры – данные, изменяющие работу процедуры.

локальная
переменная

```
def printBin( n ):
    k = 128
    while k > 0:
        print ( n // k, end = "" )
        n = n % k;
        k = k // 2
```

`printBin (99)`

значение параметра
(аргумент)

Несколько параметров:

```
def printSred( a, b ):
    print ( (a + b)/2 )
```

Вывести все простые числа на отрезке [a, b]:

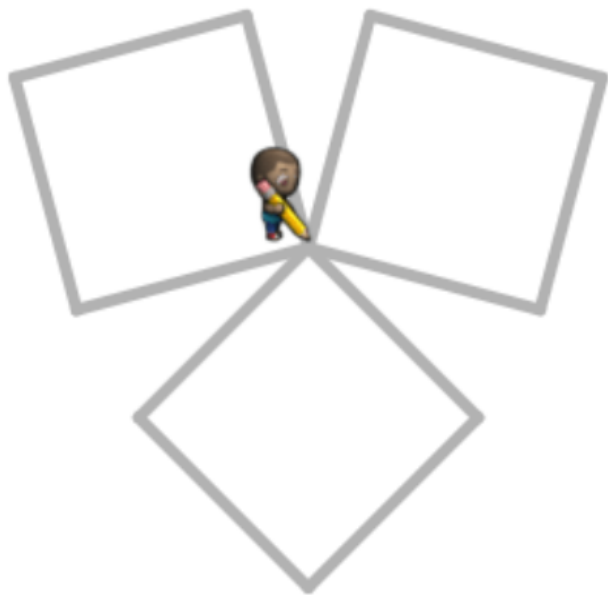
```
def prost_chislo(n):  
    k=2  
    while k*k<=n and n%k!=0:  
        k=k+1  
    if k*k>n:  
        print(n, end=", ")
```

```
a=int(input())  
b=int(input())  
for x in range(a, b+1):  
    prost_chislo(x)
```


Рисуем ряд из 10 квадратов (урок 12 code.org Художник функции)



```
import turtle
def square(n):
    for i in range(4):
        turtle.forward(n)
        turtle.left(90)
for i in range(10):
    square(20)
    turtle.penup()
    turtle.forward(40)
    turtle.pendown()
turtle.mainloop()
```



Инструкции



Используй функцию «нарисовать квадрат», чтобы нарисовать этот узор.
Подсказка: Тебе необходимо будет использовать блок «повернуть направо на 120 градусов» после рисования каждого квадрата.

Только просмотр: 13 / 13 блоки

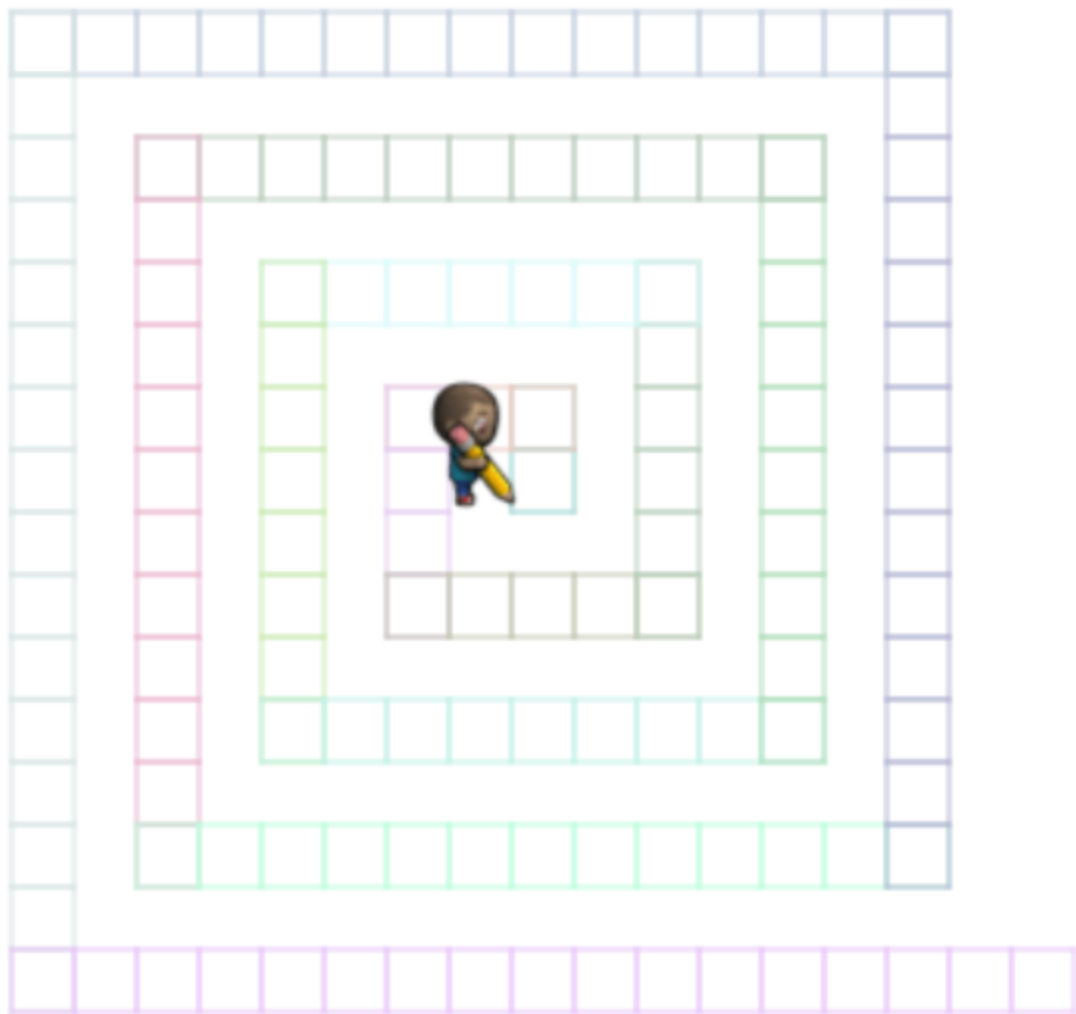
при запуске

повторить 3 раз

выполнить

draw a square [редактировать](#)

повернуть вправо на 120 градусов



Инструкции



Нажми здесь, чтобы еще раз увидеть г

Только просмотр:

при запуске

для counter от 1 до 17 с шагом 1

здать цвет случайный цвет

повторить counter раз

выполнить draw a square 20 редактировать

прыгнуть вперед на 20 точек

повернуть влево на 90 градусов