



алгоритмика

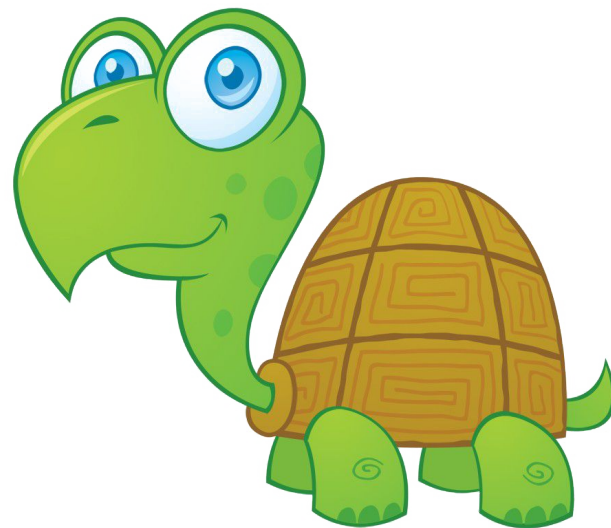
# PYTHON

Занятие 9

# Сегодня на уроке

---

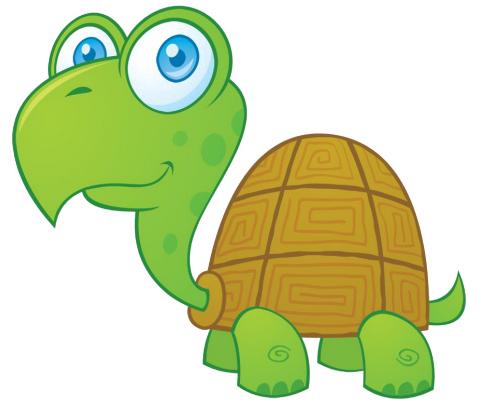
- Черепашьи гонки!!!



# Вопрос

---

1. Что такое условие?
2. Что такое условный оператор?
3. Что такое цикл с условием?
4. Что такое арифметический цикл?
5. Что такое диапазон значений переменной?
6. Как работает функция `range()`?
7. Для чего используется команда `goto()`?
8. Для чего используются команды `penup` и `pendown`?



# Цикл в Python

---

Арифметический цикл:

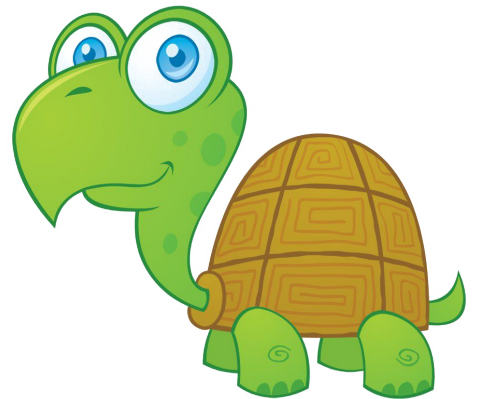
`for переменная_цикла in range(старт,стоп):`

↔ тело цикла

Цикл с условием

`while` условие:

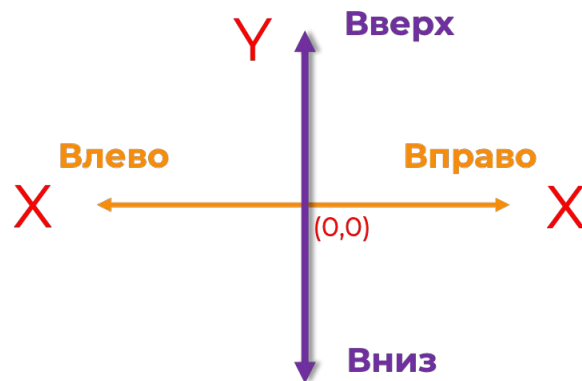
↔ тело цикла



# Команда goto()

Команда goto(X,Y) - перемещает перо черепахи в указанные координаты.

Направление движения	Координата X	Координата Y
направо →	увеличится	не изменится
← налево	уменьшится	не изменится
вверх ↑	не изменится	увеличится
вниз ↓	не изменится	уменьшится

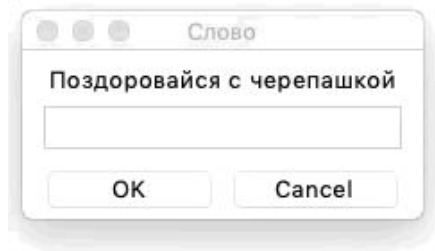


# Команда write() в Python

---

Команда write(text) – выводит текст в месте нахождения черепахи.

```
from turtle import*  
text=textinput("Слово","Поздоровайся с черепашкой")  
write(text)  
exitonclick()
```



# Создание игрового поля

---

1. Назовите первые три действия черепахи после запуска программы.
2. Какой диапазон чисел печатается на экране?
3. Какие команды выполняет черепаха после печати числа 0?
4. Есть ли в программе повторяющиеся действия?



# Создание игрового поля

---

1. Назовите первые три действия черепахи после запуска программы?

**Поднять перо; переместиться; опустить перо.**

2. Какой диапазон чисел печатается на экране?

**От 0 до 14.**

3. Какие команды выполняет черепаха после печати числа 0?

**Поворот направо; идти вперёд; поворот налево; идти вперёд; поворот направо; идти вперёд.**

4. Есть ли в программе повторяющиеся действия?

**Да, печать числа+отрисовка линии вниз, вверх и вправо.**





# Создание игрового поля

---

1. Какой цикл мы будем использовать?
2. С помощью какой функции мы задаём диапазон значений?



# Создание игрового поля

---

1. Какой цикл мы будем использовать?

**Арифметический, цикл for.**

2. С помощью какой функции мы задаём диапазон значений?

**Функция range().**

# Задание

---

Допишите программу, чтобы получилось игровое поле

```
from turtle import*
```

```
penup()
```

```
goto(-100,100)
```

```
pendown()
```



# Задание. Решение

---

Допишите программу, чтобы получилось игровое поле

```
from turtle import*
```

```
penup()
```

```
goto(-100,100)
```

```
pendown()
```

```
for i in range(0,15):
```

```
    write(i)
```

```
    right(90)
```

```
    forward(200)
```

```
    left(180)
```

```
    forward(200)
```

```
    right(90)
```

```
    forward(20)
```

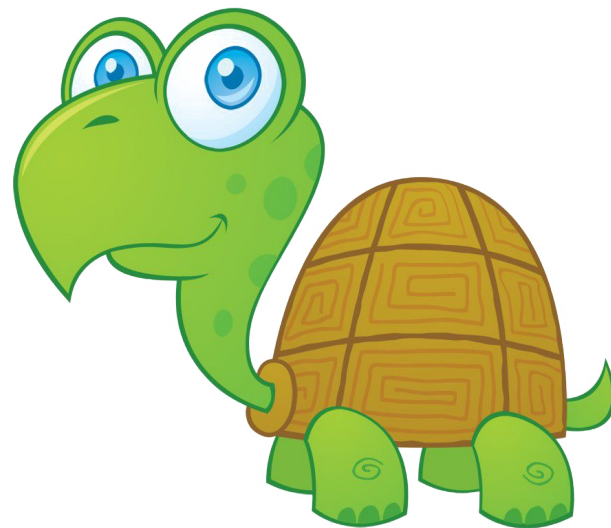
```
exitonclick()
```



# Вопрос

---

Кто выполняет наши команды?

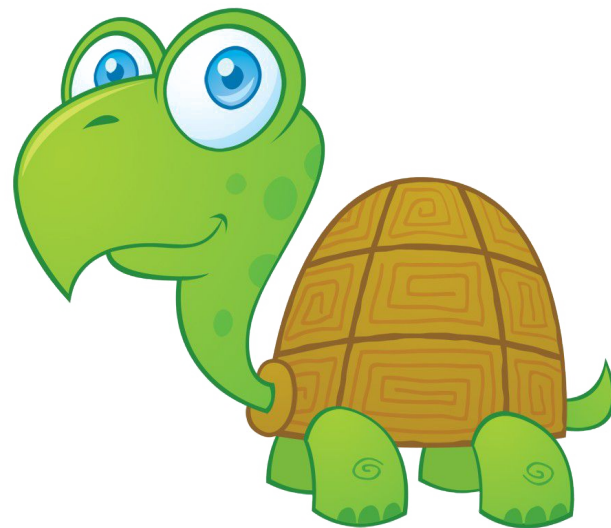


# Вопрос-ответ

---

Кто выполняет наши команды?

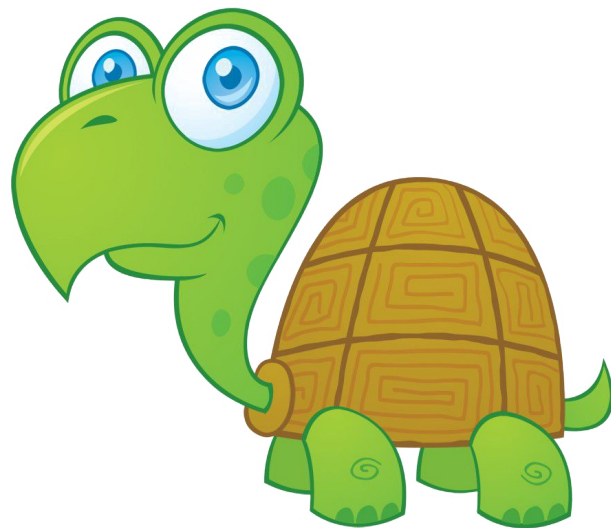
Черепаха



# Вопрос

---

Как отличать черепах, если в программе их будет 2, 3 или даже 5?

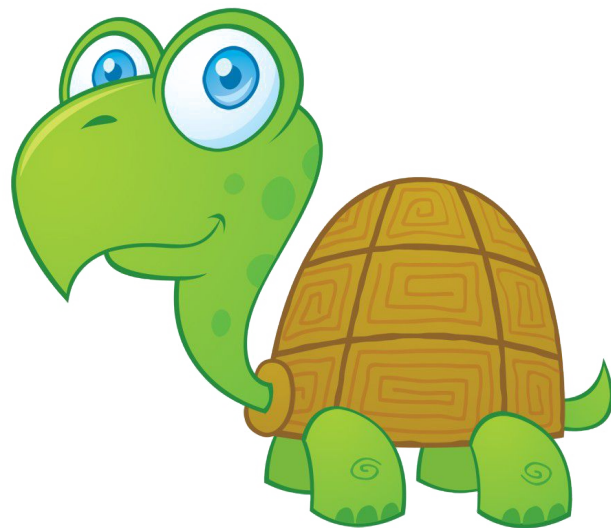


# Вопрос-ответ

---

Как отличать черепах, если их в программе будет 2, 3 или даже 5?

По именам

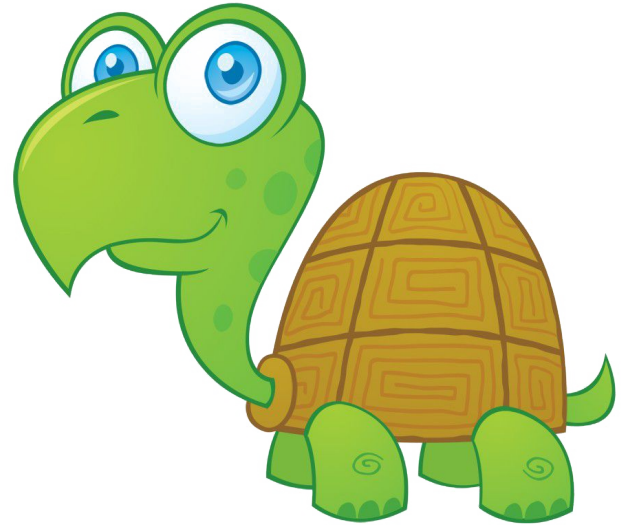




# Объект

---

**Объект** - это тот, **КТО** будет выполняет действие.  
Каждый объект имеет имя, которое позволяет отличать его от других объектов.



# Создание объекта в Python

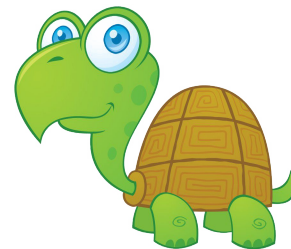
---

Для создания объекта используется следующая конструкция:

```
first=Turtle()
```

↑  
Имя объекта

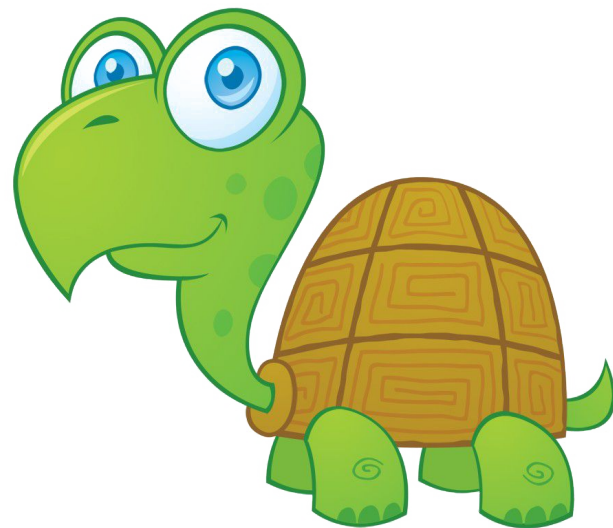
↑  
Кому принадлежит этот объект



# Вопрос

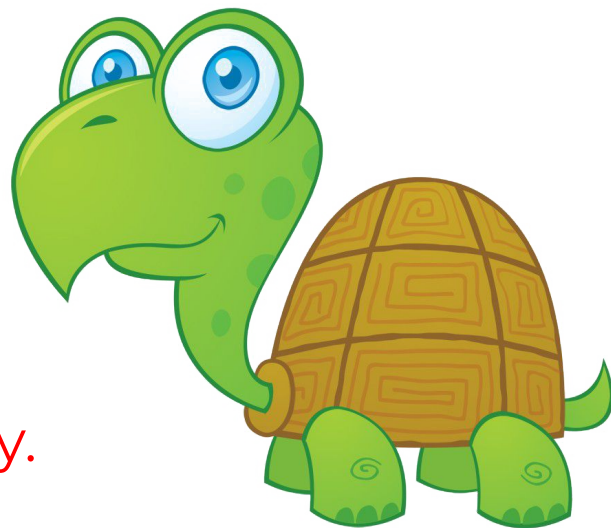
---

**Что** умеет делать черепаха?



## Что умеет делать черепаха?

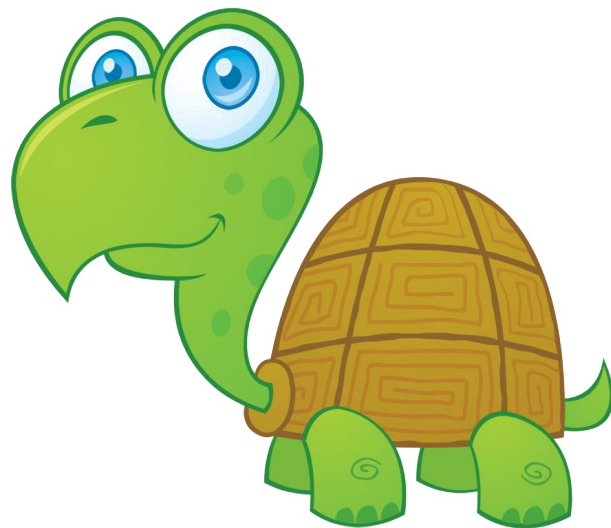
- Идти вперёд;
- Идти назад;
- Поворачиваться налево;
- Поворачиваться направо;
- Менять цвет;
- Поднимать и опускать перо;
- Перемещаться в указанную точку.



# Метод

---

**Метод** – это действие, которое может совершить объект. **Что** делать объекту.



# Вызов метод объекта в Python

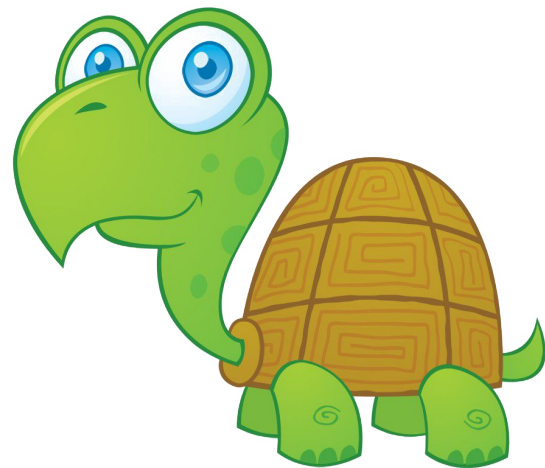
---

Обращение к методу происходит через точку - .

Вызов метода возможен **ТОЛЬКО** после создания объекта.

**Имя\_объекта.Метод**

**first.goto(-100,0)**



# Задание

---

Добавьте в проект создание первого участника гонки с именем first.

```
from turtle import*
```

```
penup()
```

```
goto(-100,100)
```

```
pendown()
```

```
for i in range(0,15):
```

```
    write(i)
```

```
    right(90)
```

```
    forward(200)
```

```
    left(180)
```

```
    forward(200)
```

```
    right(90)
```

```
    forward(20)
```

```
exitonclick()
```

first=Turtle()

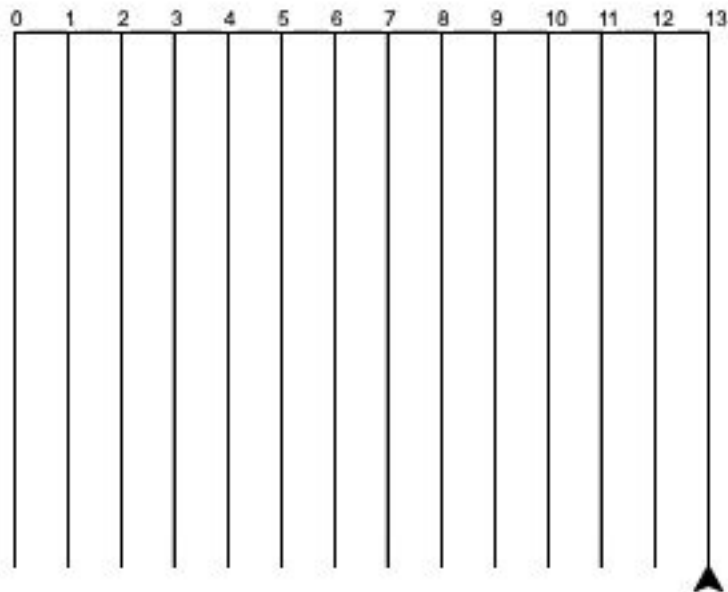


# Задание. Решение

Добавьте в проект создание первого участника гонки с именем first.

```
from turtle import*
```

```
penup()  
goto(-100,100)  
pendown()  
for i in range(0,15):  
    write(i)  
    right(90)  
    forward(200)  
    left(180)  
    forward(200)  
    right(90)  
    forward(20)  
first=Turtle()  
exitonclick()
```

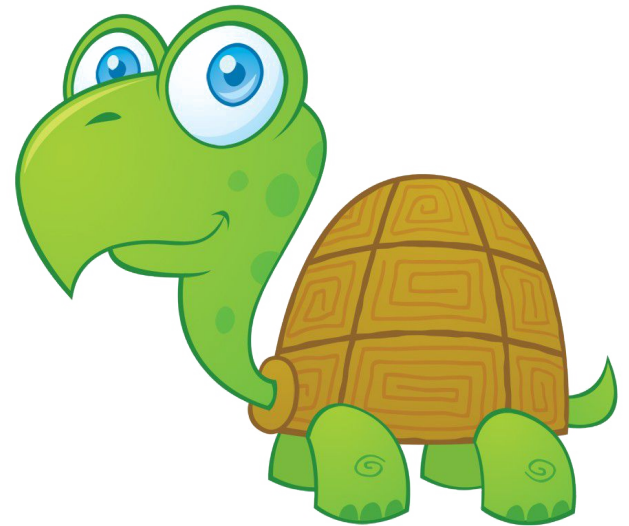




# Вопрос

---

Кто будет участвовать в гонке?

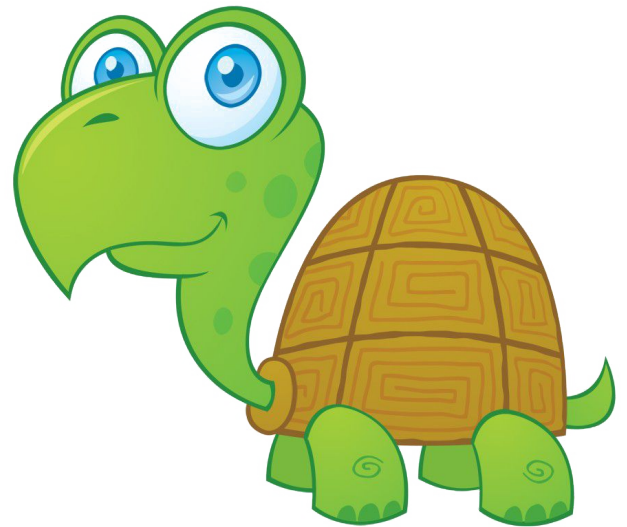


# Вопрос-ответ

---

Кто будет участвовать в гонке?

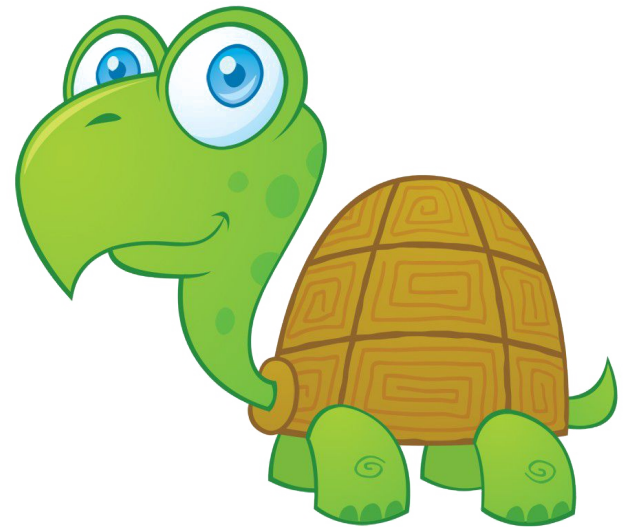
Черепахи



# Вопрос

---

В каком виде мы видим черепаху на экране?

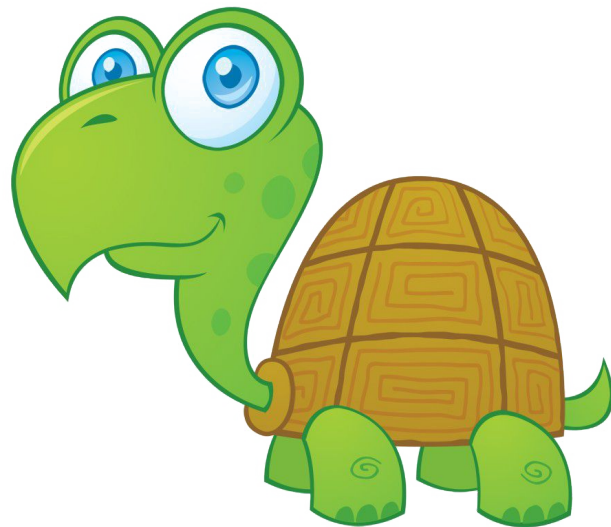


# Вопрос-ответ

---

В каком виде мы видим черепаху на экране?

В виде стрелки



# Метод shape()

---

shape("форма") – устанавливает форму объекта (черепахи)

Доступные формы:

- ▶ "arrow" – стрелка
- ✖ "turtle" – черепаха
- "circle" – круг
- "square" – квадрат
- ▶ "triangle" – треугольник
- ▶ "classic" – классика

```
first=Turtle()  
first.shape("turtle")
```



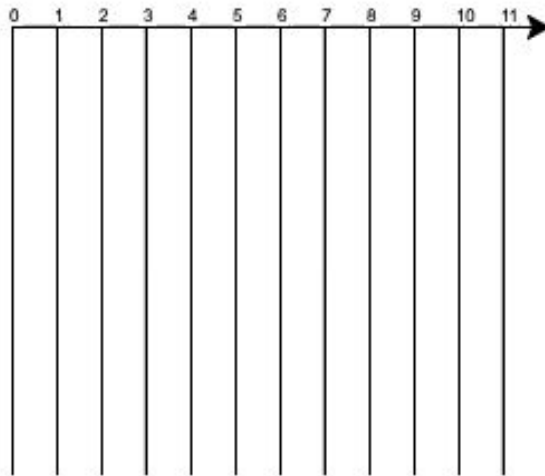
# Задание

---

Измените форму черепахи в проекте

```
from turtle import*
```

```
penup()  
goto(-100,100)  
pendown()  
for i in range(0,15):  
    write(i)  
    right(90)  
    forward(200)  
    left(180)  
    forward(200)  
    right(90)  
    forward(20)  
first=Turtle()  
exitonclick()
```

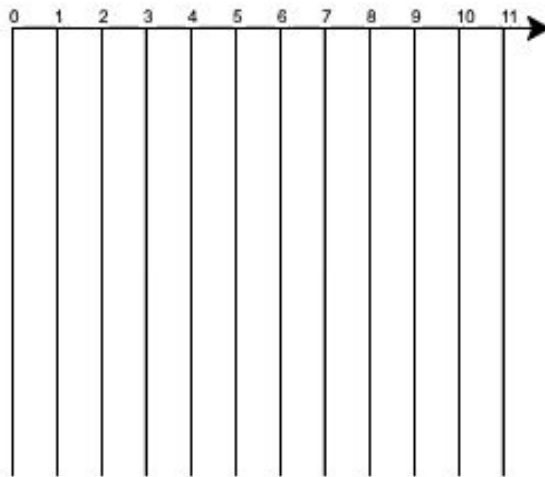


# Задание. Решение

Измените форму черепахи в проекте

```
from turtle import*
```

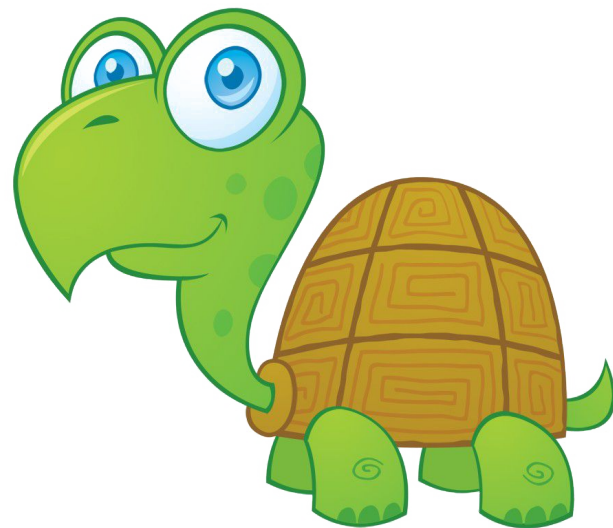
```
penup()
goto(-100,100)
pendown()
for i in range(0,15):
    write(i)
    right(90)
    forward(200)
    left(180)
    forward(200)
    right(90)
    forward(20)
first=Turtle()
first.shape("turtle")
exitonclick()
```



# Вопрос

---

Где в данный момент находится черепаха после появления на поле?



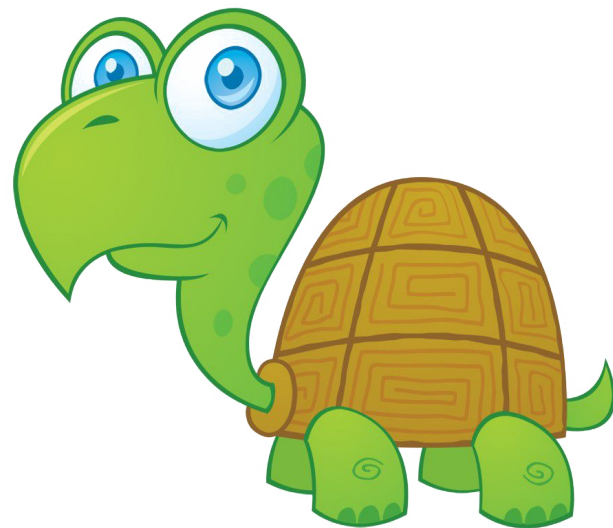


# Вопрос-ответ

---

Где в данный момент находится черепаха после появления на поле?

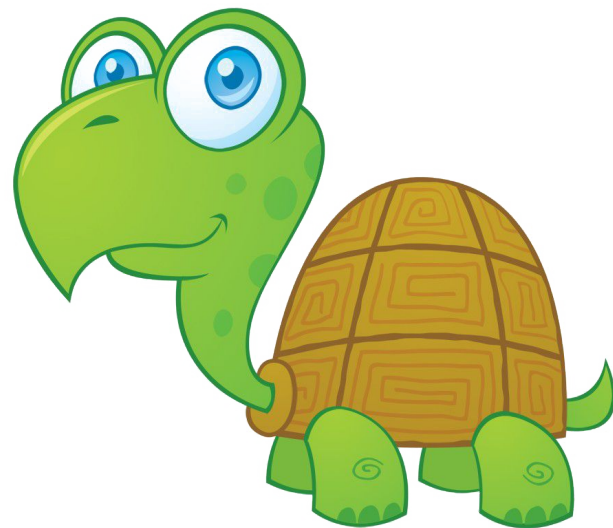
**В центре экрана**



# Вопрос

---

Где должна стоять черепаха перед началом забега?

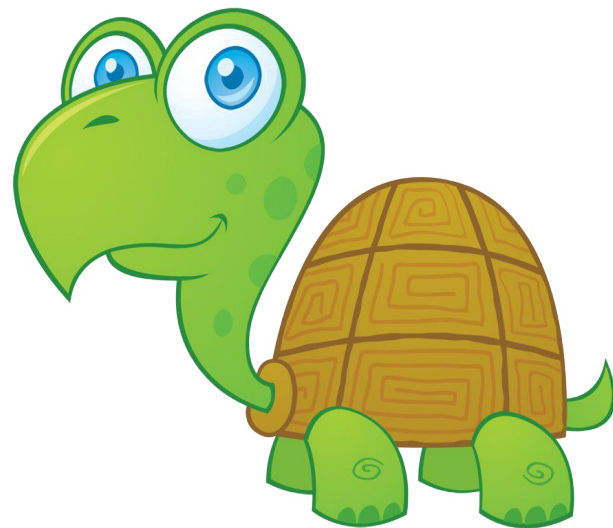


# Вопрос-ответ

---

Где должна стоять черепаха перед началом забега?

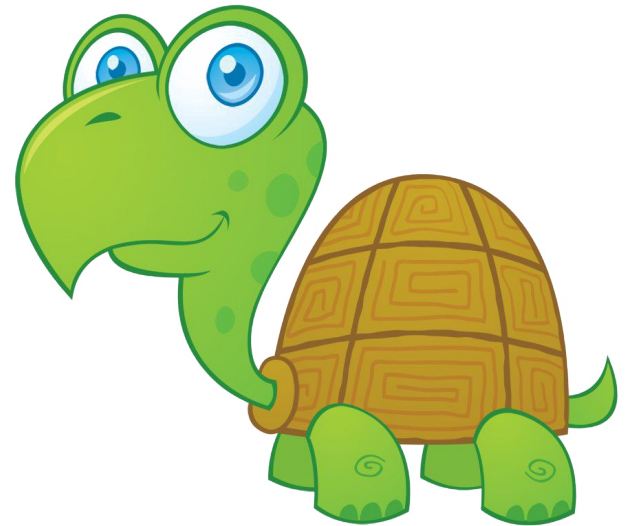
Перед линией старта



# Вопрос

---

Какое число соответствует  
линии старта?

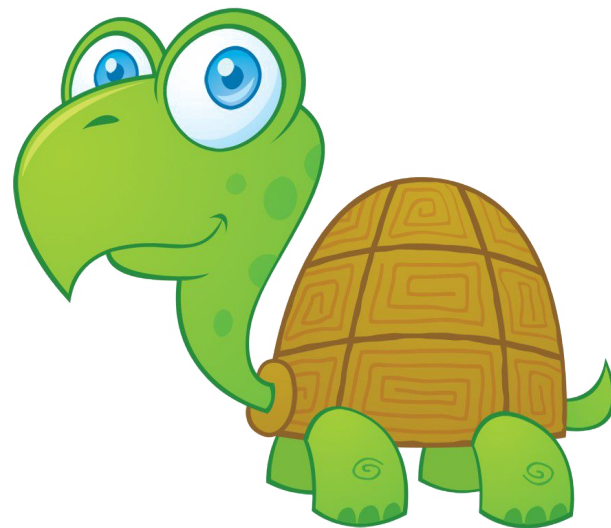
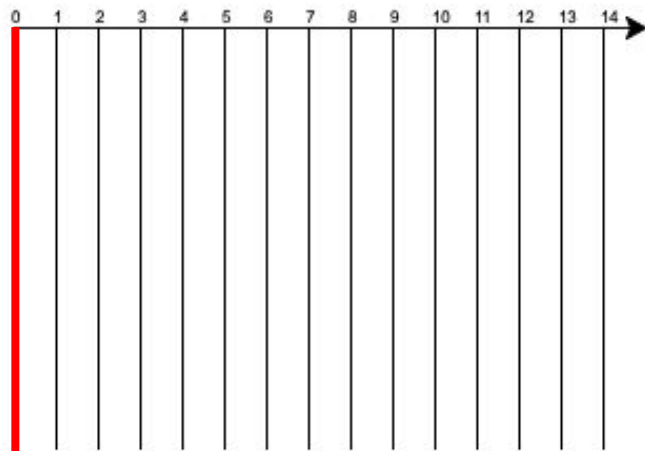


# Вопрос-ответ

---

Какое число соответствует  
линии старта?

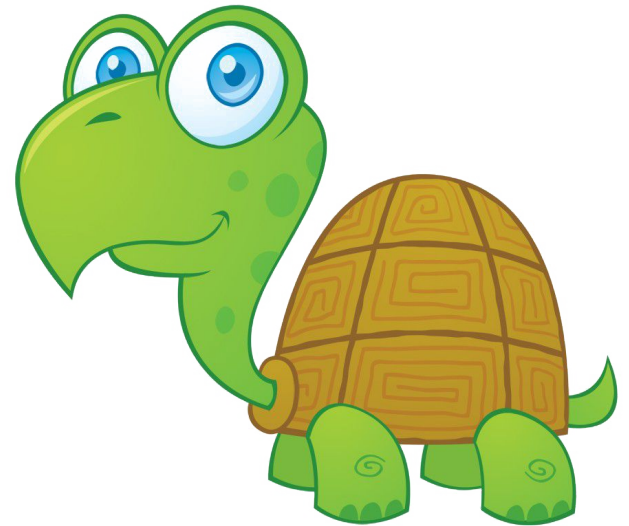
0



# Вопрос

---

Каким действием мы можем переместить черепаху ?

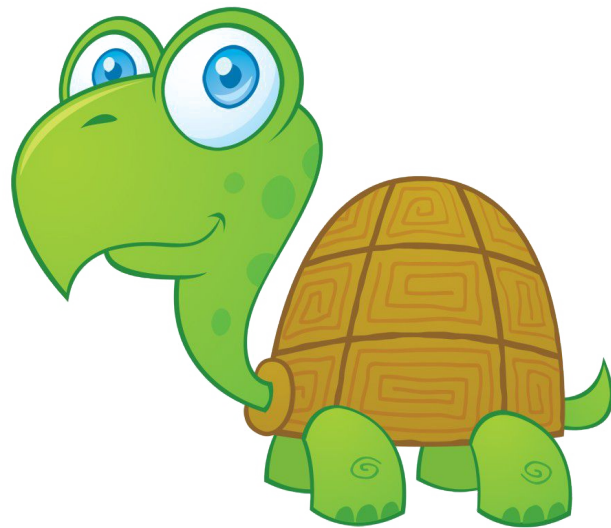


# Вопрос-ответ

---

Каким действием мы можем переместить черепаху ?

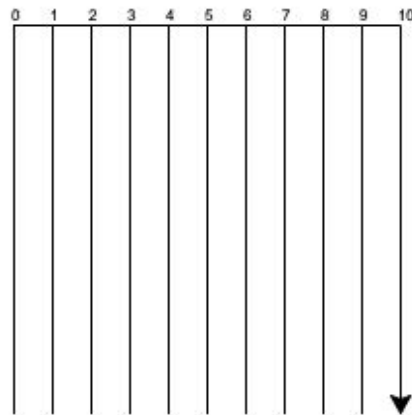
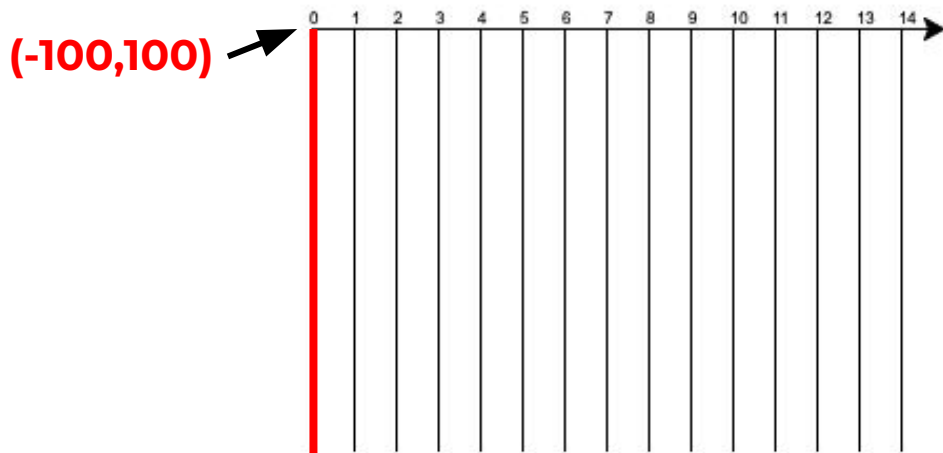
`goto(x,y)`



# Вопрос

---

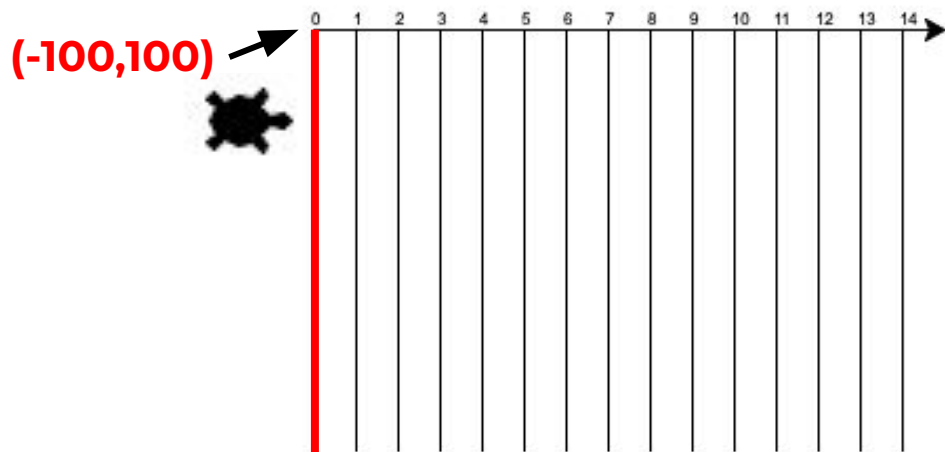
Какие координаты мы должны указать в `goto()`, чтобы черепаха оказалась перед линией? В какую сторону и от какой линии мы должны отступить?





# Вопрос-ответ

Какие координаты мы должны указать в `goto()`, чтобы черепаха оказалась перед линией? В какую сторону и от какой линии мы должны отступить?



`goto(-120,70)`

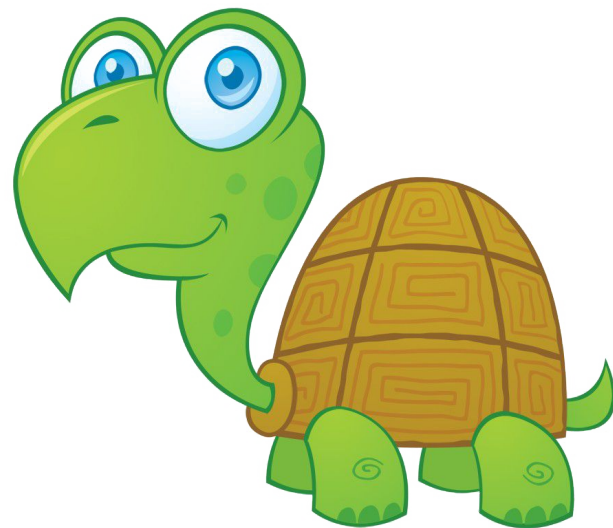
**-120** – отступ от линии старта влево

**70** – отступ от линии с числами вниз

# Вопрос

---

Какая линия на поле является финишем для черепахи?

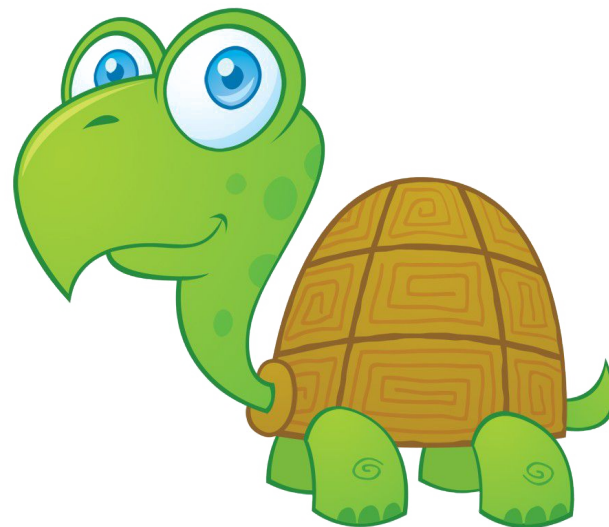
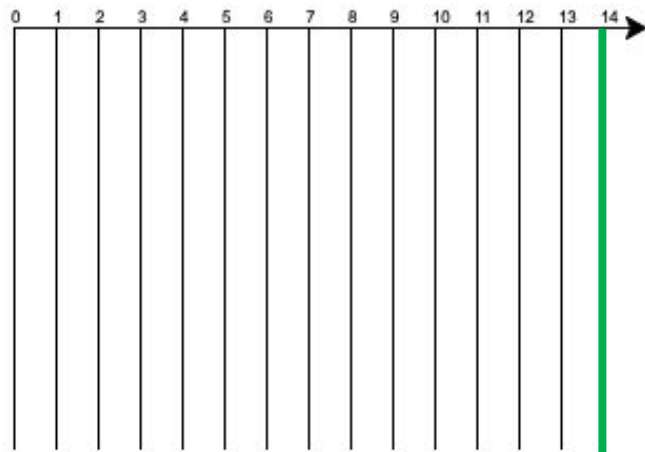


# Вопрос-ответ

---

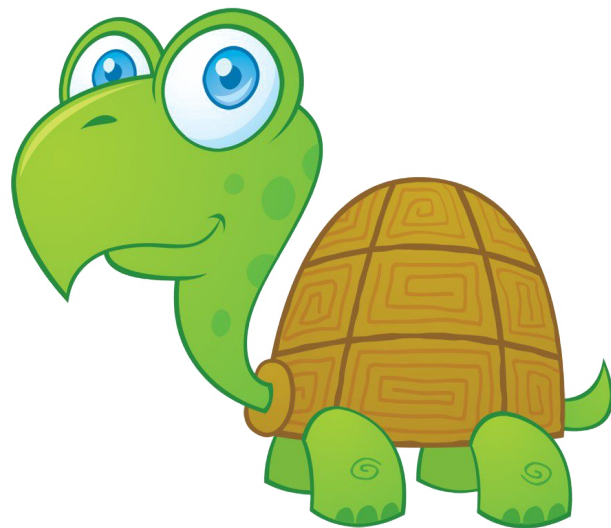
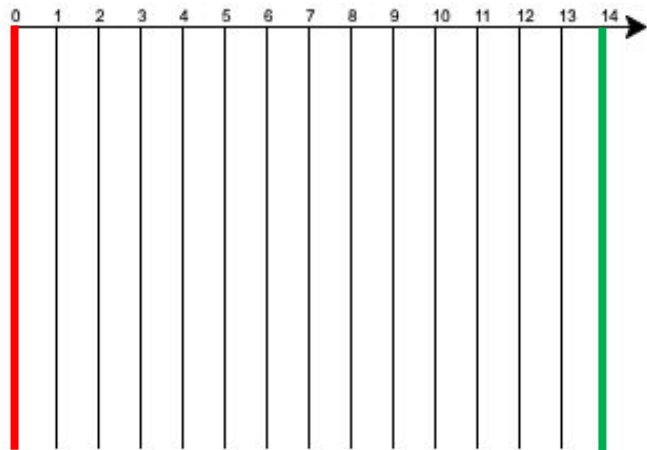
Какая линия на поле является финишем для черепахи?

14



# Вопрос

Какая координата будет изменяться у черепахи при движении от линии старта до линии финиша? Расстояние между каждой линией равно 20.

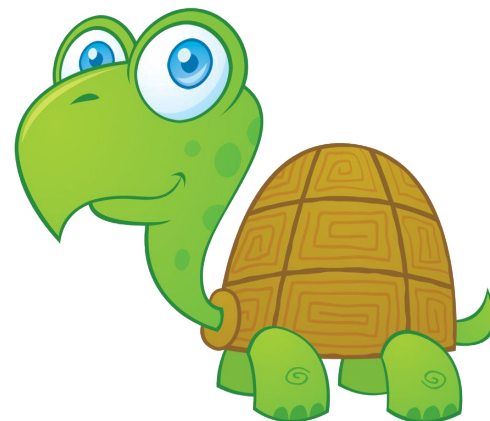
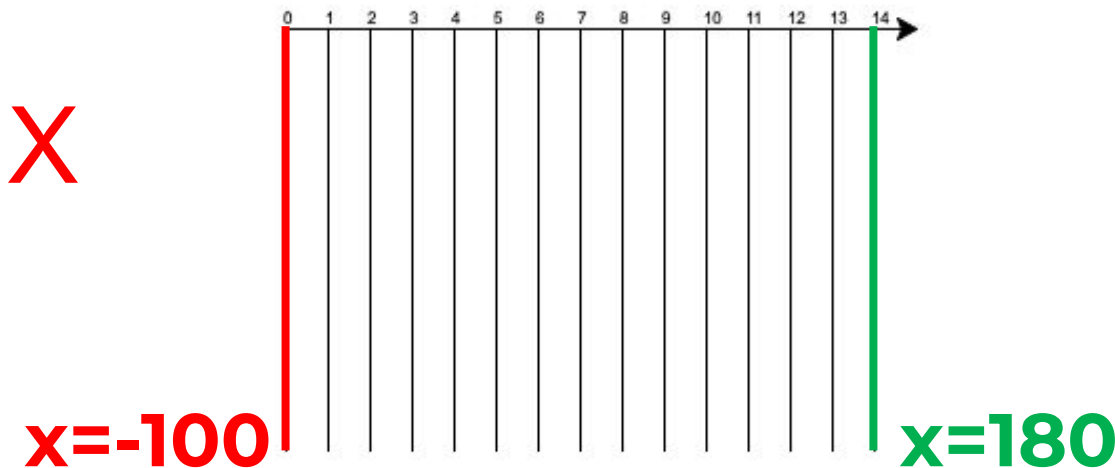


# Вопрос-ответ

Расстояние между каждой линией равно 20.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
-100	-80	-60	-40	-20	0	20	40	60	80	100	120	140	160	180

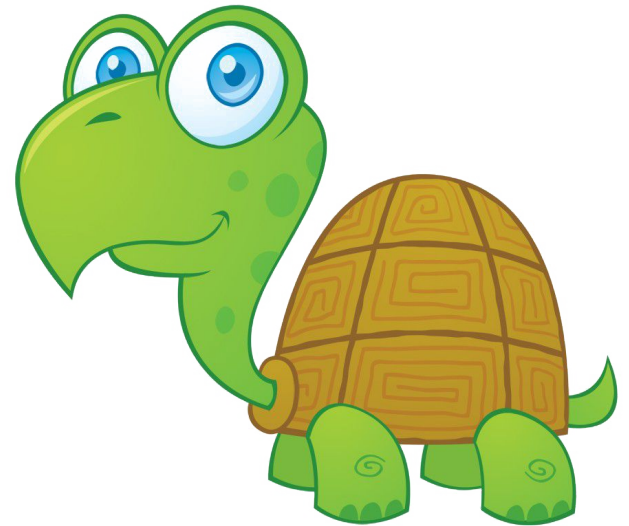
280



# Вопрос

---

Какое действие будет повторять черепаха?

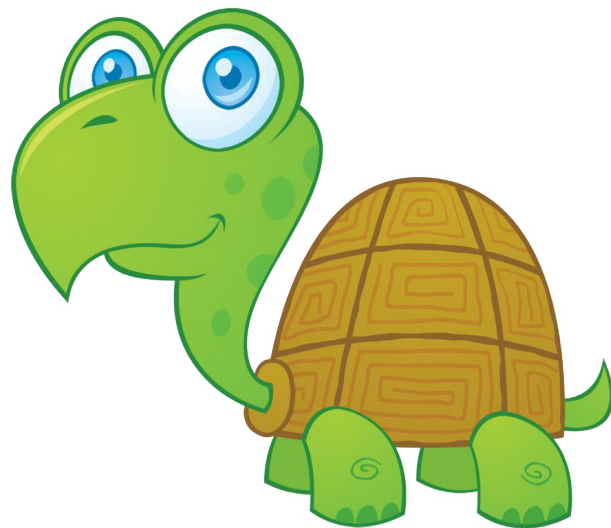


# Вопрос

---

Какое действие будет повторять черепаха?

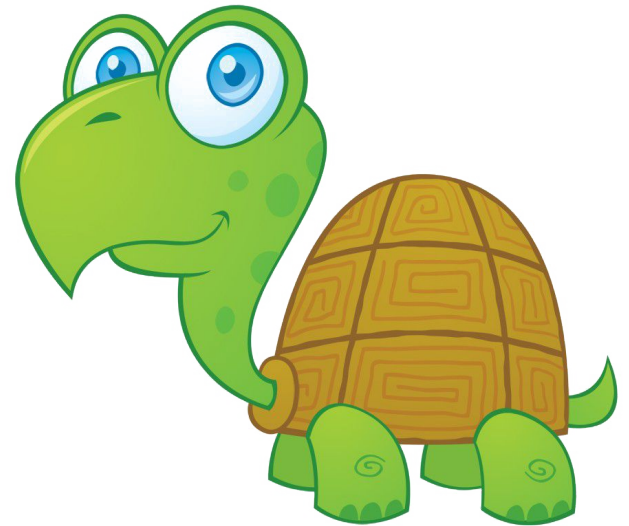
Идти вперёд



# Вопрос

---

Когда черепаха должна прекратить движение?



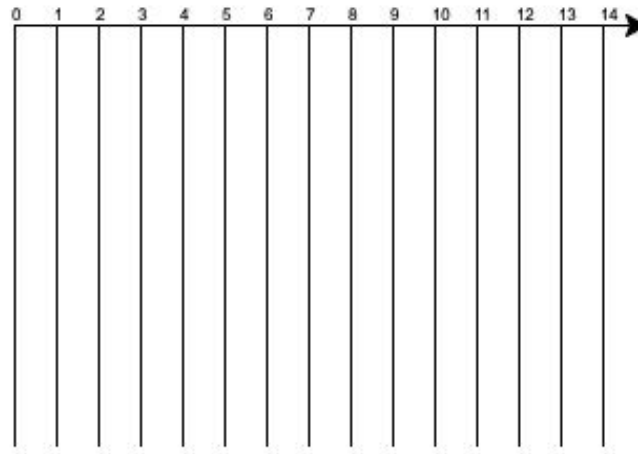


# Вопрос

---

Когда черепаха должна прекратить движение?

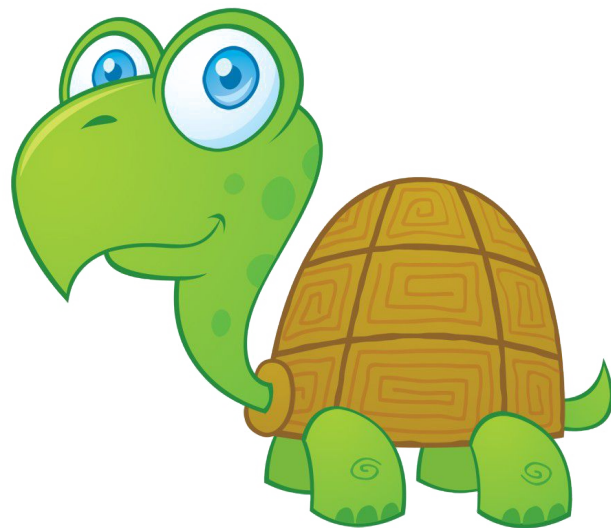
Когда дойдёт до  
финиша  
координата  $x \geq 180$



# Вопрос

---

Какой цикл используется, когда есть условие для завершения действий?



# Вопрос

---

Какой цикл используется, когда есть условие для завершения действий?

Цикл с условием.  
`while`

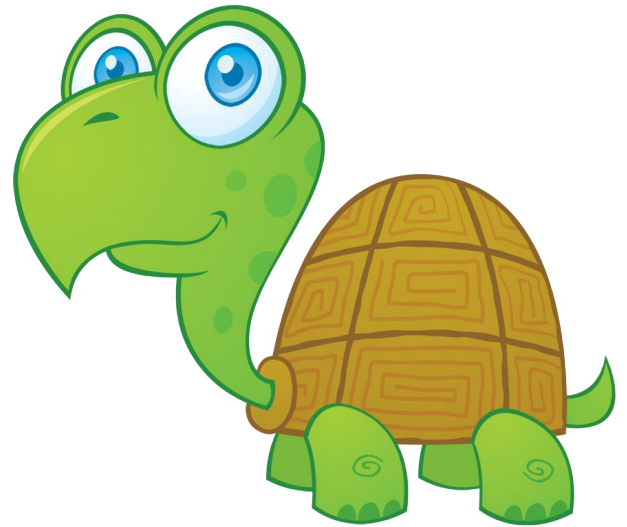
`while` условие:



# Вопрос

---

За что будет отвечать переменная цикла?



# Вопрос

---

За что будет отвечать переменная цикла?

За координату X  
у черепахи

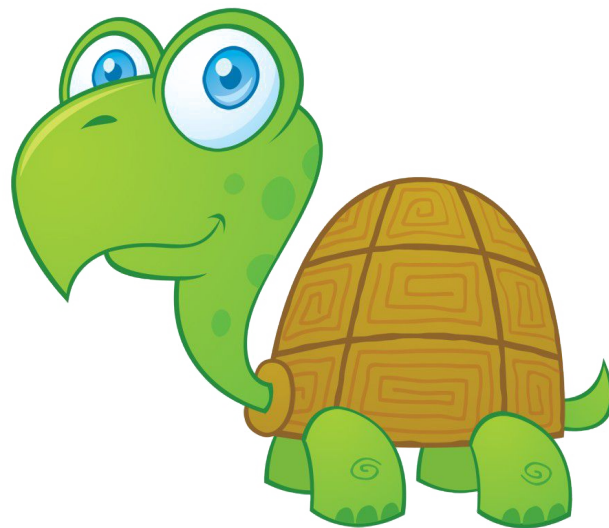
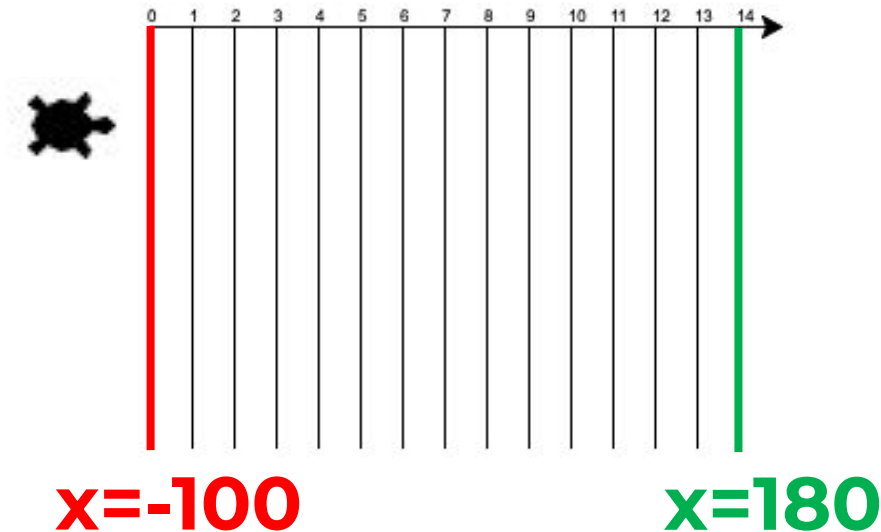
`x_first=-120`

`while` условие:



# Вопрос

Какое условие будет проверяться в заголовке цикла?



# Вопрос

---

Какое условие будет проверяться в заголовке цикла?

Пока переменная  
меньше 180

```
x_first=-120
```

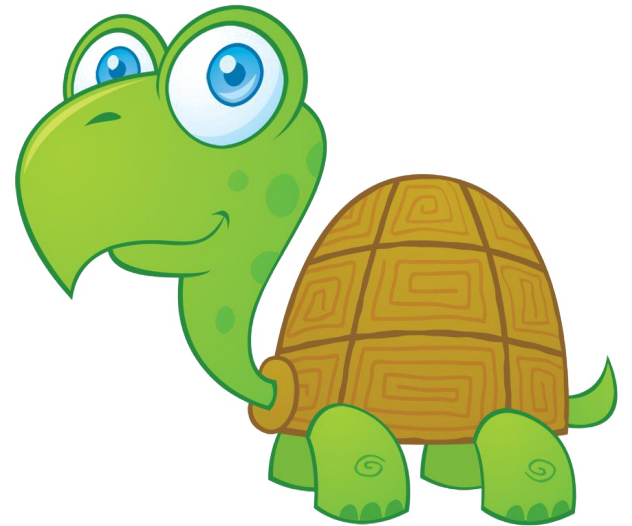
```
while x_first<180:
```



# Вопрос

---

Какие команды будут в теле цикла?





# Вопрос

---

Какая команда будет в теле цикла?

Увеличение  
координаты X и  
перемещение черепахи

```
x_first=-120
```

```
while x_first<180:
```

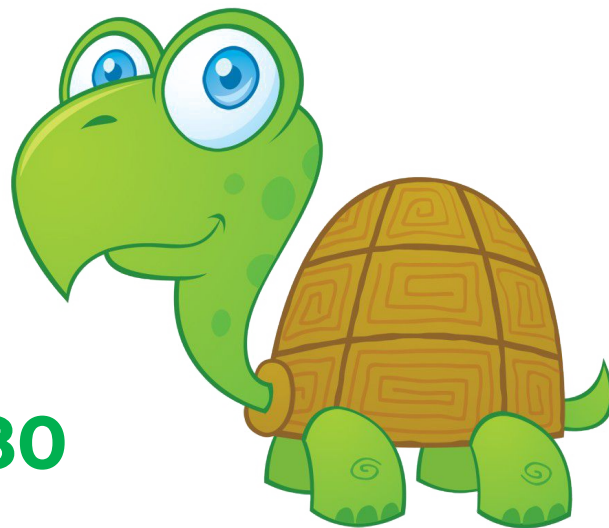
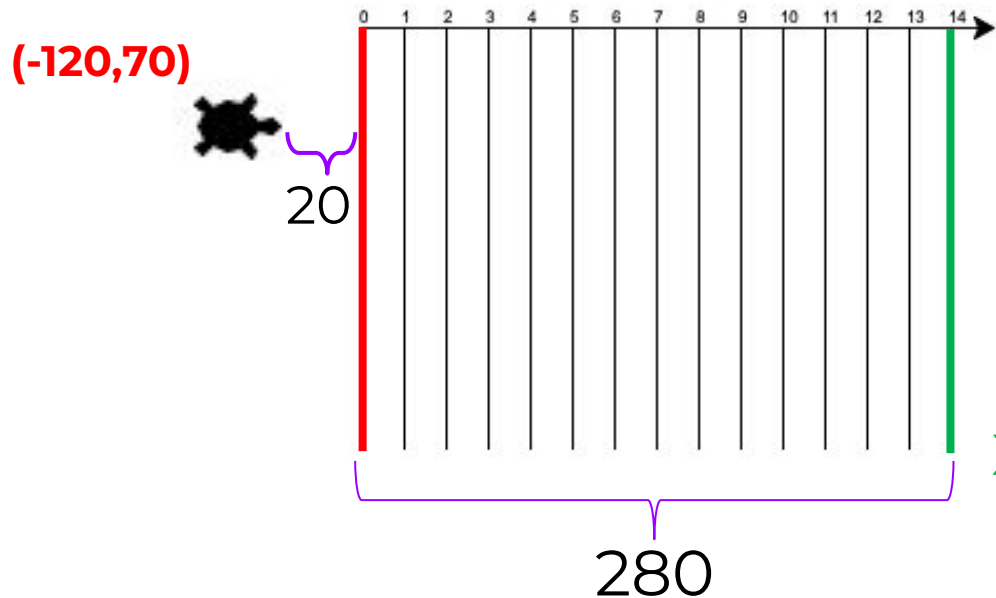
```
    x_first=x_first+1
```

```
    first.goto(x_first,70)
```



# Вопрос

Сколько шагов сделает черепаха до финиша?

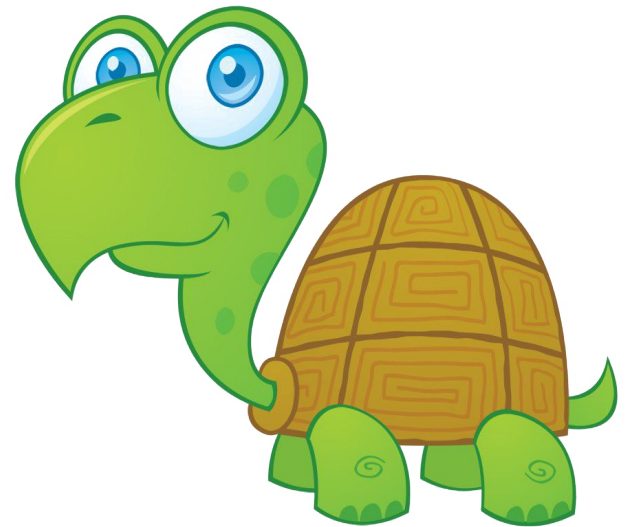


# Вопрос

---

Сколько шагов сделает черепаха от старта до финиша?

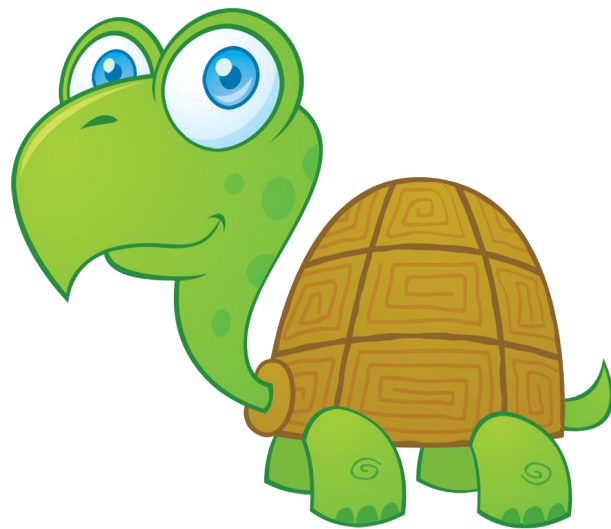
300



# Вопрос

---

Как изменится количество шагов, если минимальный шаг черепахи будет 1, а максимальный 5?



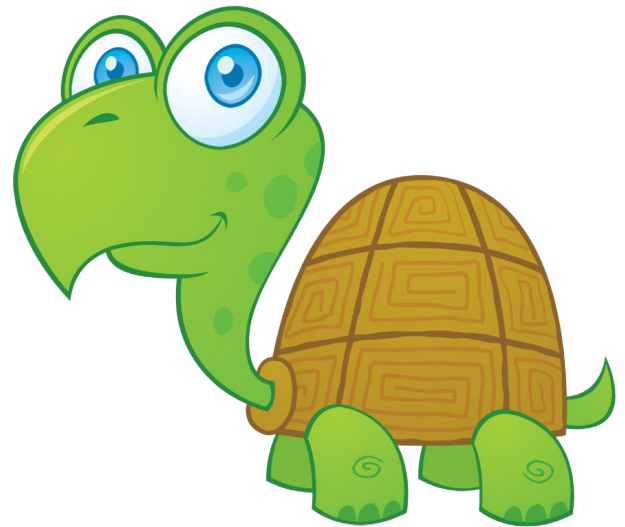
# Вопрос

---

Как изменится количество шагов, если минимальный шаг черепахи будет 1, а максимальный 5?

Максимальное  
количество шагов = 300

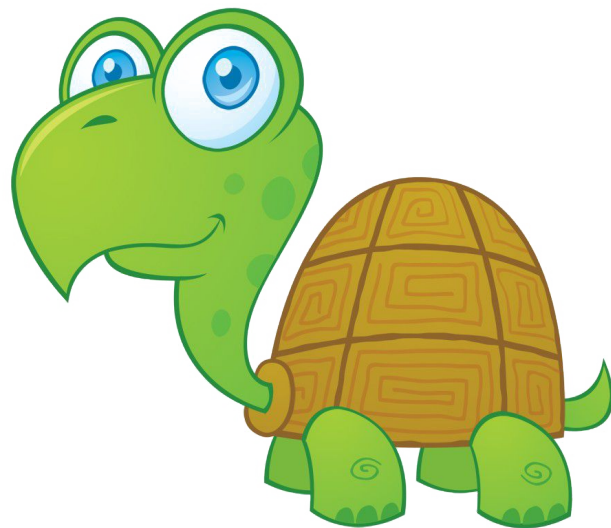
Минимальное  
количество шагов = 60



# Вопрос

---

Сможем ли мы заранее указать количество шагов, если длина каждого шага будет заранее не известна?

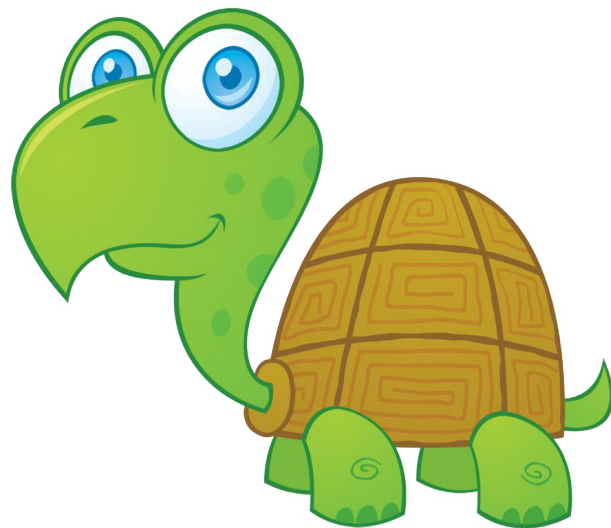


# Вопрос

---

Сможем ли мы заранее указать количество шагов, если длина каждого шага будет заранее не известна?

Нет



# Вопрос

---

Какое значение может выпасть при подбрасывании кубика?





# Вопрос-ответ

---

Какое значение может выпасть при подбрасывании кубика?

1, 2, 3, 4, 5, 6



# Вопрос

---

Знаем ли мы заранее, какое число  
выпадет?



# Вопрос

---

Знаем ли мы заранее, какое число  
выпадет?

Нет



# Случайное значение

---

Это значение, которое выпадает из диапазона возможных значений.

1, 2, 3, 4, 5, 6

диапазон возможных значений

5

случайное значение



# Модуль random в Python

---

random – случайный.

Модуль random управляет созданием случайных значений.

**Подключение модуля:**

```
from random import*
```

**randint(начало,конец)** - возвращает случайное число из диапазона от начала до конца.

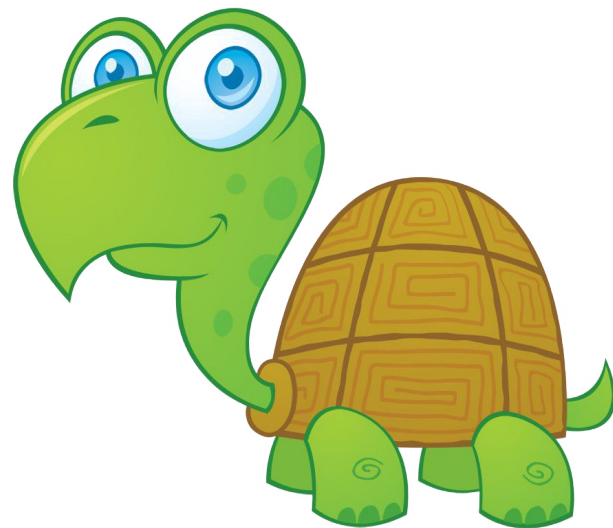
randint(1,5) - вернёт случайное значение от 1 до 5.



# Вопрос

---

Какое значение должно стать случайным в программе?



# Вопрос

---

Какое значение должно стать случайным в программе?

## Шаг черепахи

`x_first=x_first+1` → `x_first=x_first+randint(1,5)`

# Задание

---

Измените программу, чтобы каждый шаг черепахи был случайным.

```
from turtle import*
from random import*
|
x_first=-120
while x_first<180:
    x_first=x_first+1
    first.goto(x_first,70)
```

`randint(1,5)`- вернёт случайное значение от 1 до 5





# Задание. Решение

---

Измените программу, чтобы каждый шаг черепахи был случайным.

```
x_first=-120
```

```
while x_first<180:
```

```
    x_first=x_first+randint(1,5)
```

```
    first.goto(x_first,70)
```

# Задание

---

Измените цвет первой черепахи на красный.

```
first=Turtle()  
first.shape("turtle")
```



# Задание. Решение

---

Измените цвет первой черепахи на красный.

```
from turtle import*  
from random import*
```

```
penup()  
goto(-100,100)  
pendown()  
for i in range(0,15):  
    write(i)  
    right(90)  
    forward(200)  
    left(180)  
    forward(200)  
    right(90)  
    forward(20)
```

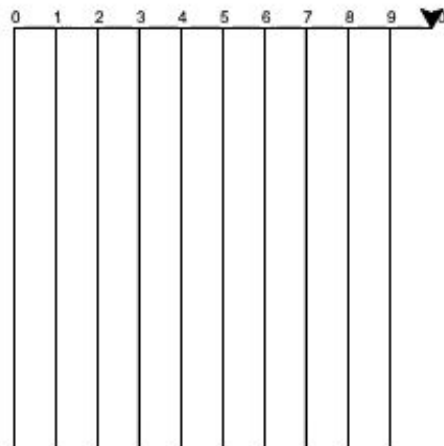
```
first=Turtle()  
first.shape("turtle")  
first.color("red")  
first.penup()  
first.goto(-120,70)  
first.pendown()  
x_first=-120  
  
while (x_first<180):  
    x_first=x_first+randint(1,5)  
    first.goto(x_first,70)  
  
exitonclick()
```



# Задание

---

Добавьте в игру вторую черепаху: имя – `second`, место старта – `(-120,40)`, цвет – синий, переменная для координаты – `x_second`.

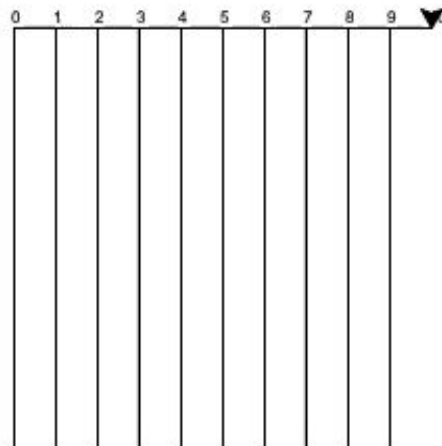


# Задание. Решение

---

Добавьте в игру вторую черепаху: имя – second, место старта – (-120,40), цвет – синий, переменная для координаты – x\_second.

```
second=Turtle()  
second.shape("turtle")  
second.color("blue")  
second.penup()  
second.goto(-120,40)  
second.pendown()  
x_second=-120
```



# Задание

---

Аналогично движению первой черепахи добавьте в цикл движение для второй черепахи.

```
while (x_first < 180):  
    x_first = x_first + randint(1, 5)  
    first.goto(x_first, 70)
```



# Задание. Решение

---

Аналогично движению первой черепахи добавьте в цикл движение для второй черепахи.

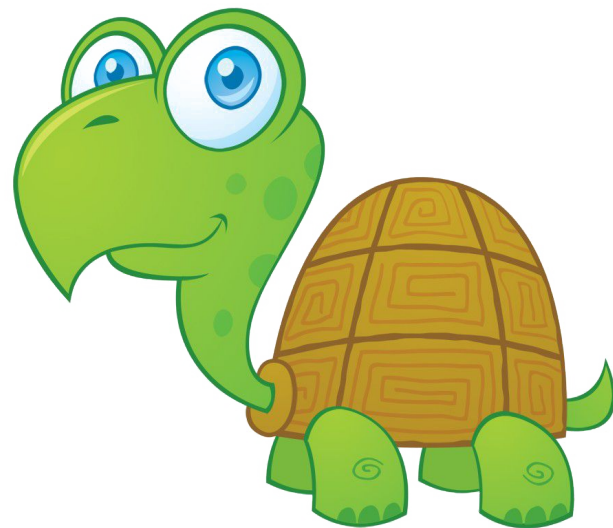
```
while (x_first < 180):  
    x_first = x_first + randint(1, 5)  
    first.goto(x_first, 70)  
    x_second = x_second + randint(1, 5)  
    second.goto(x_second, 40)
```



# Вопрос

---

Что сейчас является условием окончания игры?



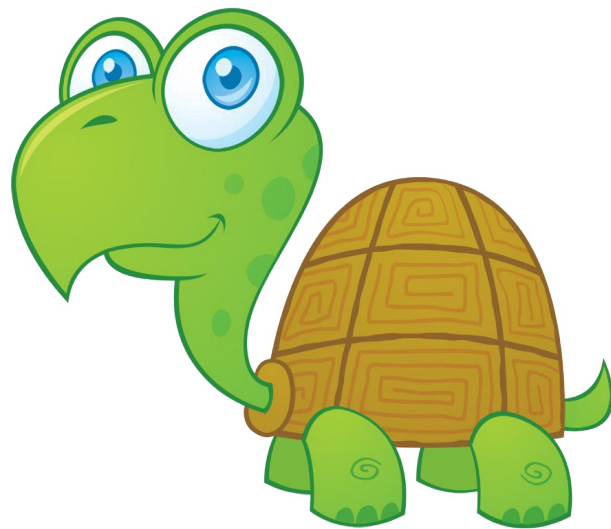


# Вопрос

---

Что сейчас является условием окончания игры?

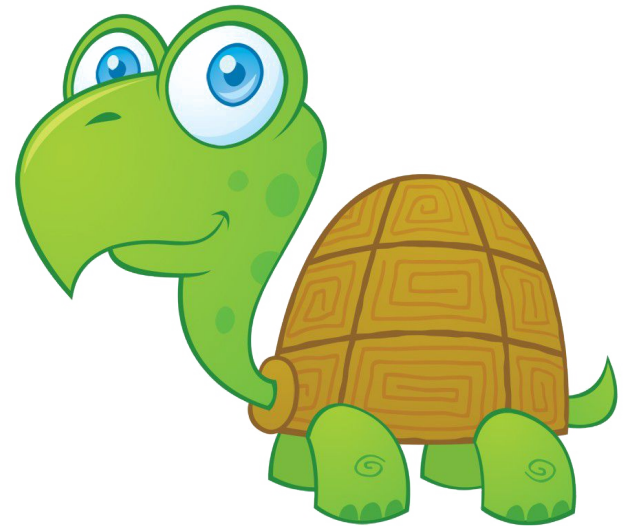
Первая черепаха  
(красная) дошла  
до финиша



# Вопрос

---

Что должно являться условием окончания игры?

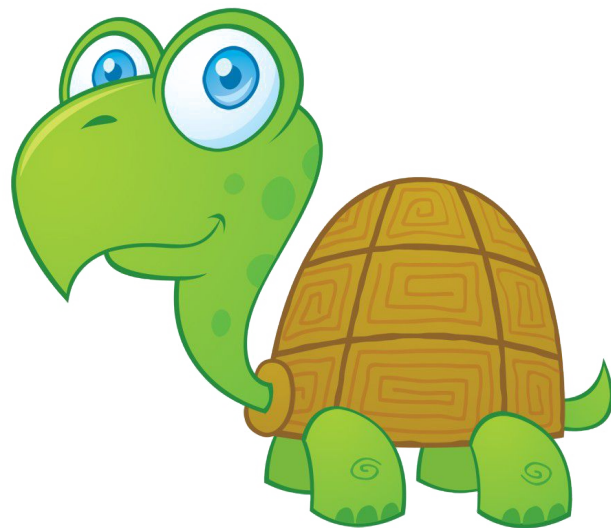


# Вопрос

---

Что должно являться условием окончания игры?

Одна из черепах  
дошла до финиша

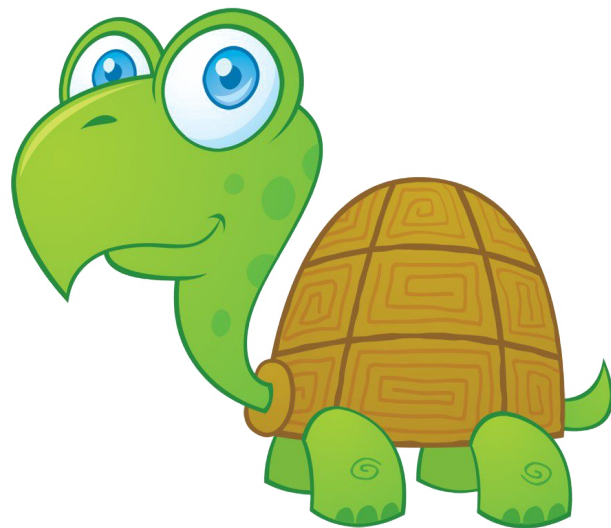


# Вопрос

---

Как изменить условие цикла, чтобы учитывать движение **и** второй черепахи?

```
while (x_first < 180):
```



# Составное условие

---

Это условие, состоящее из нескольких простых условий, связанных с помощью логических операций:

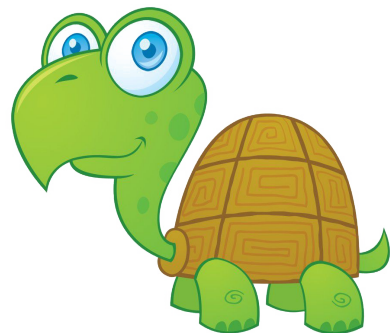
Знак операции	Название	Описание
or	ИЛИ	Для истинности выражения достаточно выполнение хотя бы одного из условий.
and	И	Для истинности выражения необходимо одновременное выполнение условий.

# Вопрос

---

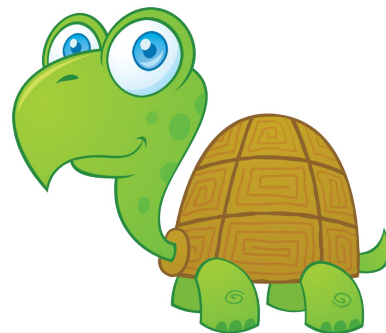
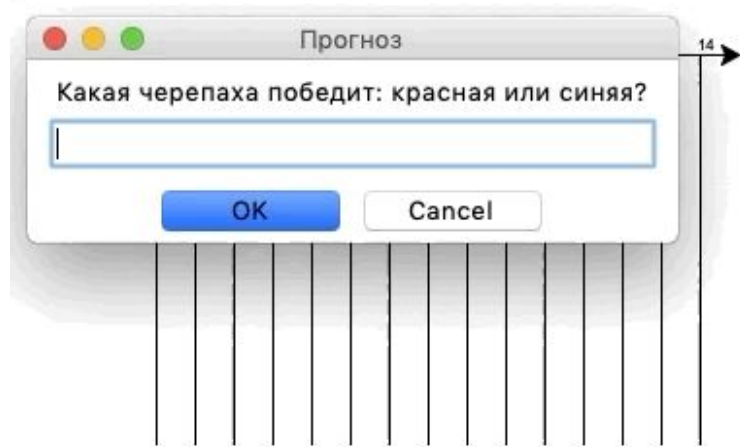
Как изменить условие цикла, чтобы учитывать движение **и** второй черепахи?

```
while ((x_first < 180) and (x_second < 180)):
```



# Дополнительные задания

1. Добавьте запрос прогноза на выигрыш черепахи по цвету
2. Значение прогноза вывести в окно черепахи



# Условный оператор в Python

---

if условие\_1 :

↔ набор команд\_1

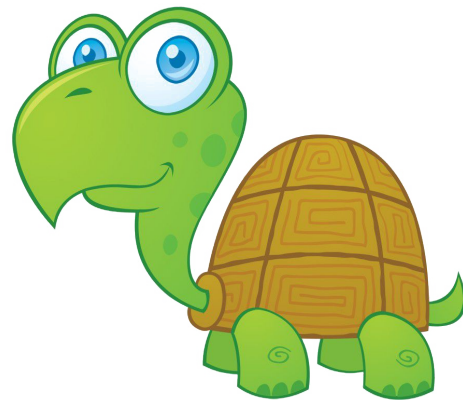
elif условие\_2 :

↔ набор команд\_2

else:

↔ набор команд\_3

**elif** позволяет указать альтернативное условие, которое требуется проверить.





# Дополнительные задания

---

3. Добавить проверку прогноза, если пользователь угадал, вывести в консоль (print) «Ты угадал», иначе «Ты не угадал»
4. Добавить ввод длины игрового поля через диалоговое окно

