

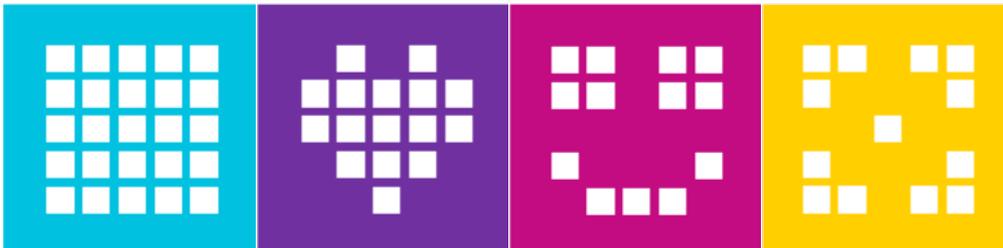
УРОКИ ПО SPIKE PRIME

By the Makers of EV3Lessons



ЦИКЛЫ

BY ARVIND SESHAN



ЦЕЛЬ УРОКА

- Узнаем, как повторить действие используя циклы.

ПОВТОРЕНИЕ КОДА

- Мы хотим, чтобы робот повторял действие несколько раз:
 - Мы бы скопировали нужные функции несколько раз?
 - Что, если Мы хотим повторить действие всегда?
- Мы можем использовать циклы, чтобы повторить действие несколько раз или пока не будет выполнено условие.
- Циклы делают задачу многократного повторения простой.
- Дополнительное преимущество - то, что цикл может закончиться когда Мы захотим (определенное количество раз, повторять всегда, особое условие, и т.д.).
- Есть два типа циклов: цикл `for` и цикл `while`.

ЦИКЛ FOR

- Цикл for используется для последовательного повторения.

```
numbers = [1, 4, 9, 13]
```

```
for number in numbers:
```

```
    print(number)
```

Вывод:

1

4

9

13

Примечание:
Не забудь сделать отступ
в коде, который
находится в цикле.

ЦИКЛ FOR С ДИАПАЗОНОМ()

- Если Вы хотите чтобы цикл выполнялся определённое число раз, Вы можете использовать функцию с диапазоном ().

```
for x in range(4):  
    print(x)
```

Вывод:
0
1
2
3

- Обратите внимание, что диапазон функции начинается с 0. Вы также можете установить начальное значение.

```
for x in range(2, 4):  
    print(x)
```

Вывод:
2
3

- Заметьте, что число 4 не включено. Диапазон () функции исключает максимум, который Вы устанавливаете.

- Наконец, Вы можете увеличить шаг.

```
for x in range(2, 7, 2):  
    print(x)
```

Шаг

Вывод:
2
4
6

ЦИКЛ WHILE

- Цикл While будет выполняться пока условие будет верно.
- Это полезно для повторения действия до определенного значения датчика.
- Вы также можете использовать цикл определенное число раз.

`i = 0` Начальная точка

`while (i < 10):` Конечная точка

`i = i + 1` Шаг

- Этот цикл повторится 10 раз.
- Точно так же, как в функции с диапазоном (), Вы можете установить начальную точку, конечную точку и шаг.
- Вы можете использовать этот цикл как бесконечный.

`while True:`

- Задайте условие, которое будет всегда верно и цикл будет повторяться всегда.

ИСПОЛЬЗОВАНИЕ ЦИКЛА WHILE

В этом примере робот регулирует скорость моторов на основе значений датчика Силы, пока датчик Силы не будет отпущен. Этот тип цикл отличается от функции ожидания, так как Вы можете выполнять различные действия пока ждете.

```
motor_pair = MotorPair('A', 'E')
motor_pair.set_stop_action('brake') Настраиваем моторы и датчик Силы
force = ForceSensor('E')
force.wait_until_pressed() Ждем пока датчик силы не будет нажат
while (force.is_pressed()): Выполняем цикл пока датчик нажат
    motor_pair.start(0, force.get_force_percentage())
motor_pair.stop() Регулируем скорость моторов на основе значений датчика Силы, считывая значения через цикл.
```

ЗАДАЧА: ДВИЖЕНИЕ ВОКРУГ КОРОБКИ

- Необходимо объехать коробку.
- Чтобы сделать это, Мы должны будем двигаться на 20 см вперед и повернуть направо.
- Повторим это действие 4 раза, пока мы не объедем коробку.
- Повторяем эти два действия с помощью двух типов циклов.



20 см

```
for x in range(4):
```

or

```
i = 0  
while (i < 4):  
    i = i + 1
```

РЕШЕНИЕ ЗАДАЧИ

- На предыдущих уроках Мы изучили, как настроить нашего робота. Первый блок настраивает движение моторов и действие при остановке. Эта программа для Droid Bot IV
- Код движения робота на 20 см и поворот на 90 градусов.

```
from spike.operator import greater_than_or_equal_to
motor_pair = MotorPair('A', 'E')
motor_pair.set_stop_action('brake')
motor_pair.set_motor_rotation(17.5, 'cm')
for x in range(4):
    motor_pair.move(20, 'cm')
    motor_pair.start_tank(20, 0)
    hub.motion_sensor.reset_yaw_angle()
    wait_until(hub.motion_sensor.get_yaw_angle, greater_than_or_equal_to, 90)
motor_pair.stop()
```

Примечание:
Не забудьте
импортировать
необходимые
функции

CREDITS

- This lesson was created by Arvind Seshan for SPIKE Prime Lessons
- More lessons are available at www.primelessons.org



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).