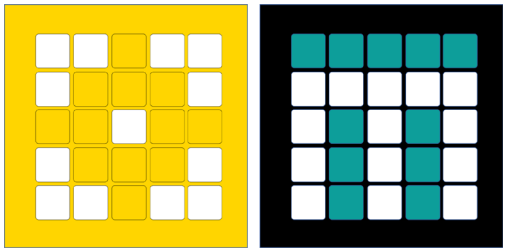
PRIME LESSONS

By the Makers of EV3Lessons



MOVING STRAIGHT

BY SANJAY AND ARVIND SESHAN

LESSON OBJECTIVES

- Learn how to make your robot go forward and backwards
- Learn how to use the Motor Pair Move methods
- Note: Although images in this lessons may show a SPIKE Prime, the code is the same for Robot Inventor

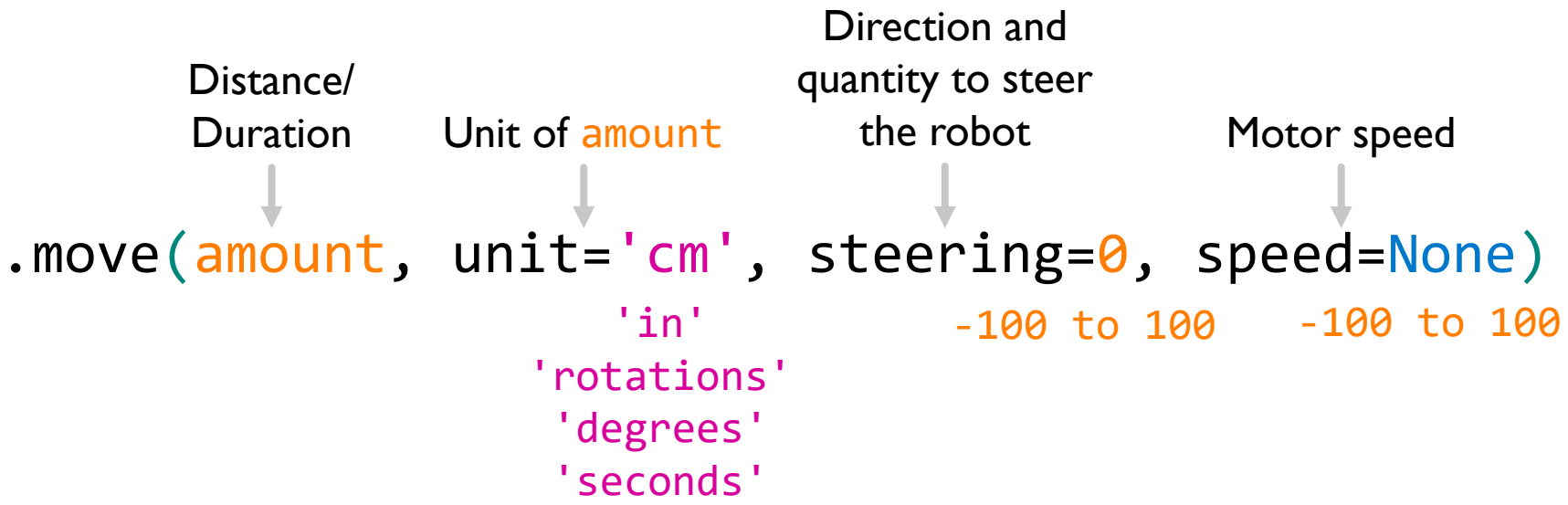
CREATING A MOTOR PAIR OBJECT

- Basic movement is done using a MotorPair object
  - See Configuring Robot Movement lesson for details on creating this object
- The following slides will cover the different methods of this object that are used for movement
  - E.g., `motor_pair.move(5, unit='cm', speed='100')`
- Do not initialize more than one motor pair with the same ports → this is redundant, will only waste memory and may cause undesired conflicts

MOTOR PAIR METHODS

- **get_default_speed**()
- **move**(amount, unit='cm', steering=0, speed=None)
- **move_tank**(amount, unit='cm', left_speed=None, right_speed=None)
- **set_default_speed**(speed)
- **set_motor_rotation**(amount, unit='cm')
- **set_stop_action**(action)
  - `Action is 'brake','hold', or 'coast'`
- **start**(steering=0, speed=None)
- **start_at_power**(power, steering=0)
- **start_tank**(left_speed, right_speed)
- **start_tank_at_power**(left_power, right_power)
- **stop**()
- We will be covering the yellow methods in this lesson

MOTOR_PAIR.MOVE()

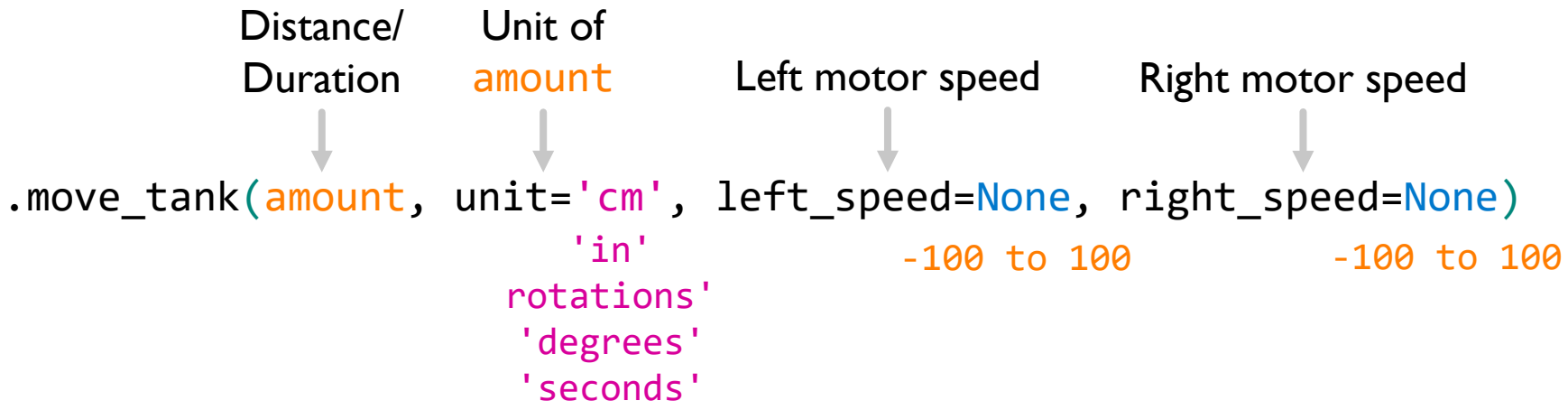


`unit='cm'`, `steering=0`, and `speed=None`, are the default values if nothing is set. When `speed=None`, the speed value used is the default speed set by `set_default_speed()`. Positive steering values turn the robot right, negative turn left. Larger values turn more sharply.

Set in Configuration

To use this method, you will set the speed, stop mode, motor ports, wheel size (see Configuring Robot Movement Lesson)

MOTOR_PAIR.MOVE_TANK()



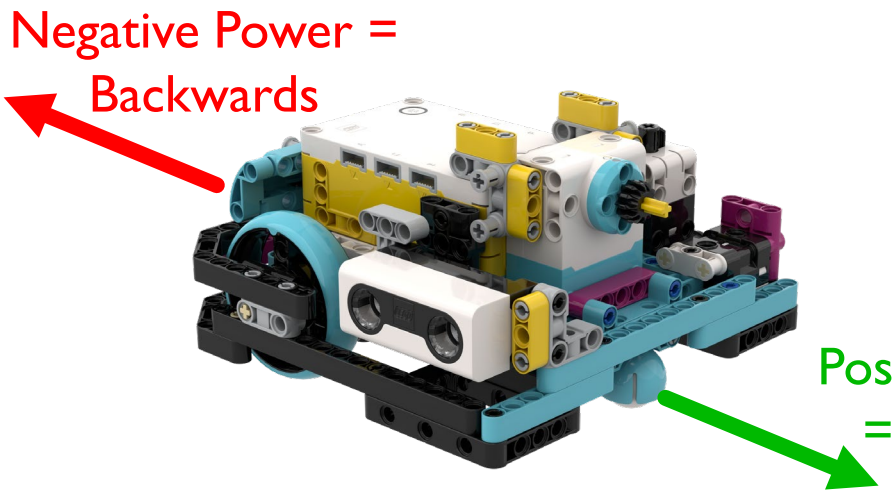
`unit='cm'`, `left_speed=None`, and `right_speed=None`, are the default values if nothing is set. When `left_speed=None` and/or `right_speed=None`, the speed value used is the default speed set by `set_default_speed()`.

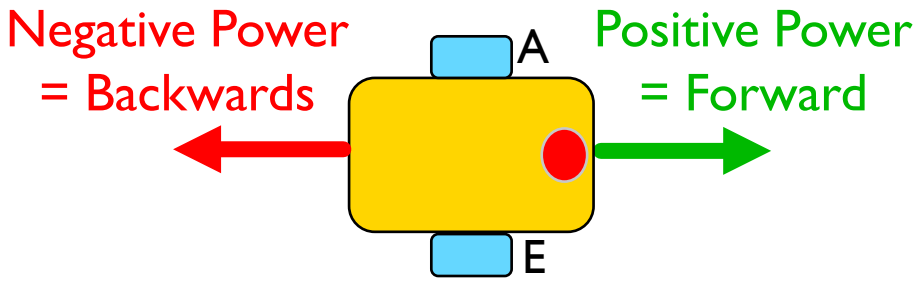
Set in Configuration

To use this method you will set the speed, stop mode, motor ports, wheel size (see Configuring Robot Movement Lesson)

NEGATIVE VALUES

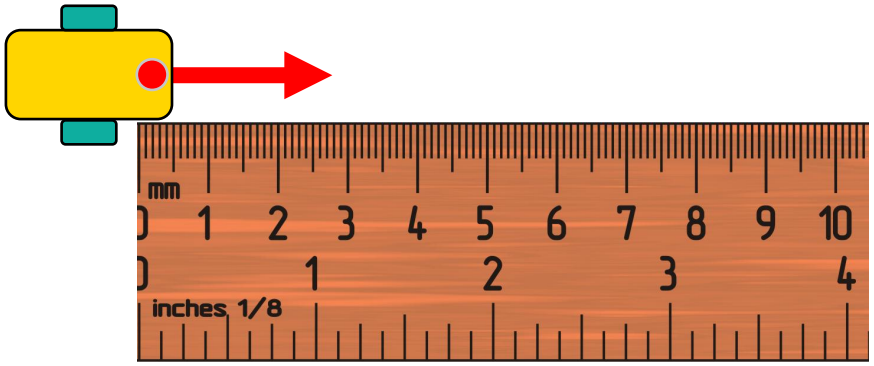
- You can enter negative values for power or distance
- This will make the robot move backwards
- If you negate two values (e.g., speed and distance negative), the robot will move forward.





CHALLENGE 1: MOVE 10 CM

- Move the robot 10 centimeters forward
- Basic steps:
  - Configure your robot
  - Use a MotorPairs method (move() or move_tank()) to move forward for 10cm



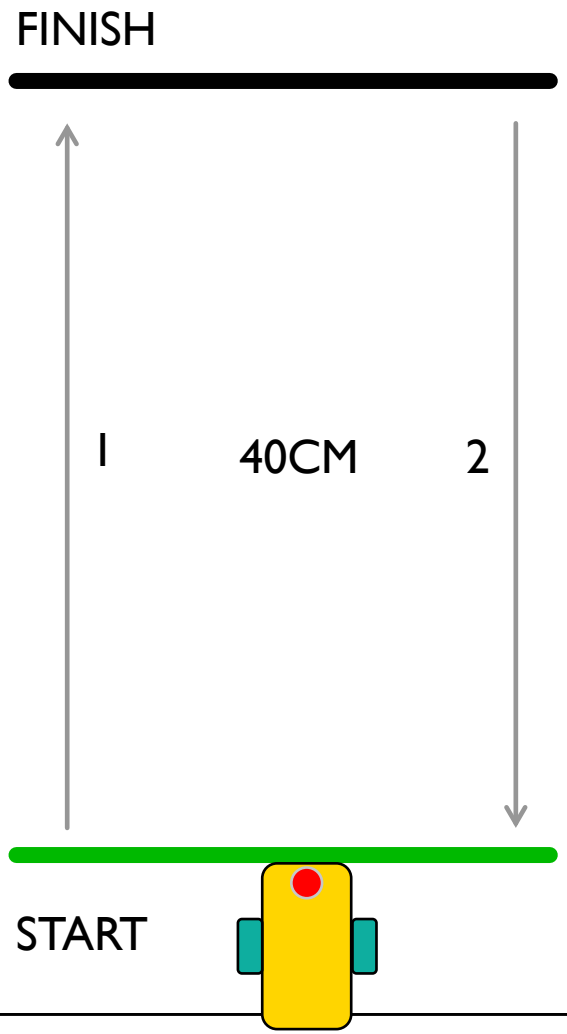
CHALLENGE 1 SOLUTION

- Configure your robot
- If you are using the smaller SPIKE Prime wheels on Droid Bot IV, set the one rotation to 17.5cm (in the code shown)
- If you are using the larger SPIKE Prime wheels on ADB, remember to set one rotation to 27.6cm
- Move forward for 10 cm. The same cm mode is available in the move_tank() method as well.

```
motor_pair = MotorPair('A', 'E')
motor_pair.set_stop_action('brake')
motor_pair.set_motor_rotation(17.5, 'cm')
motor_pair.set_default_speed(50)
motor_pair.move(10, 'cm')
```

CHALLENGE II: MOVE FORWARD AND BACK

- Move your robot forward from the start line to the finish line (1) and back to the start (2)
- Basic steps:
  - Configure your robot
  - Use a MotorPair method and move forward for the desired amount (40cm)
  - Use the same MotorPair method to move backwards (40cm)



CHALLENGE II SOLUTION

- Configure your robot
- If you are using the smaller SPIKE Prime wheels on Droid Bot IV, set the one rotation to 17.5cm (in the code shown)
- If you are using the larger SPIKE Prime wheels on ADB, you will set one rotation to 27.6cm
- Robot moves forward 40cm and backwards 40cm by setting the distance to -40.

```
motor_pair = MotorPair('A', 'E')
motor_pair.set_stop_action('brake')
motor_pair.set_motor_rotation(17.5, 'cm')
motor_pair.set_default_speed(50)
motor_pair.move(40, 'cm')
motor_pair.move(-40, 'cm')
```

START MOVING AND STOP MOVING METHODS

- There are 5 more move methods
- The start method will turn **on** your drive motors at the given speed (and steering if given).
- These methods have no duration/distance. After turning the motor on, the program instantly moves to the next line
- The motor will continue running until stopped or controlled by another method
- stop() method will halt your drive motors no matter what action they are running.
- There are also methods that allow you to control motor power instead of speed.

```
start(steering=0, speed=None)
stop()
start_tank(left_speed, right_speed)
start_at_power(power, steering=0)
start_tank_at_power(left_power, right_power)
```

ANALYSIS: MOTOR_PAIR.START_TANK()

- Turns on motors at given powers (for left and right motor)
- Power can be anywhere between -100% to 100%
- This method is called "asynchronous", meaning that it schedules the motor to run in the background
- `.start_tank(left_speed, right_speed)`
- Motors will move until stopped by the .stop() method

- Note: you can use the `.set_stop_action(action)` method to set how the motors will stop
  - The action can be `'brake'`,`'hold'`, or `'coast'`

WAIT/SLEEP

- Since Start and Stop Moving methods execute asynchronously, they need to be used with other code to be made useful. One common way they are used is with Wait Functions. Wait Functions hold up the program execution until some event occurs. The lessons on sensors cover Wait Functions in more detail.
- There are two ways to wait for a duration:
  - Option I (Recommended): Use the standard Python API. Place `import time` at the beginning of your program once. Use `time.sleep(seconds)` to wait for a specified duration
  - Option II: Use the LEGO API: Place `wait_for_seconds(seconds)` anywhere in your program to wait for a specified duration

CHALLENGE III

- Use Start Moving, Stop Moving, and Wait to make the robot move forward for 3 seconds
- Some hints:
  - Start the robot moving at 50, 50 speed
  - After turning on the motors, make the program wait 3 seconds using the time.sleep() function.
  - Use the stop() method makes the robot stop

CHALLENGE III: MOVING FOR 3 SECONDS

- Can you Move 3 Seconds using just the Start Moving and Wait blocks?

```
Import time
motor_pair = MotorPair('A', 'E')
motor_pair.set_stop_action('brake')
motor_pair.start_tank(50, 50)
time.sleep(3)
motor_pair.stop()
```

- Start the robot moving at 50, 50 speed
- After turning on the motors, the program begins running the time.sleep() function. This takes 3 seconds to run.
- The stop() method makes the robot stop

CREDITS

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