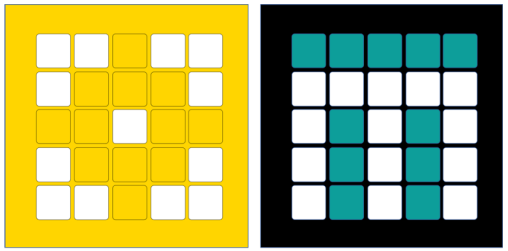
PRIME LESSONS

By the Makers of EV3Lessons



MOVING STRAIGHT

BY SANJAY AND ARVIND SESHAN

This lesson uses SPIKE 3 software

LESSON OBJECTIVES

Learn how to make your robot go forward and backwards

Learn how to use the Motor Pair Move methods

CREATING A MOTOR PAIR

Basic movement is done using a Motor Pair

- See Configuring Robot Movement lesson for details on creating a motor pair

The following slides will cover the different methods of this pair that are used for movement

- E.g., `motor_pair.move_for_degrees(<parameters>)`

Note that unlike SP2, SP3 does not have objects. You use functions to operate on the motor pair, and have to pass in the pair slot each time.

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

move

move_for_degrees

move_for_time

move_tank

move_tank_for_degrees

move_tank_for_time

pair

stop

unpair

We will be covering the yellow methods in this lesson

MOTOR_PAIR.PAIR()/UNPAIR()

To set up a motor pair, you have to pair two motors and assign them one of three available pair names.

Since there are only 6 ports on the hub, you can't have more than 3 pairs of motors

For Drive Base 1

```
motor_pair.pair(motor_pair.PAIR_1, port.C, port.D)
```

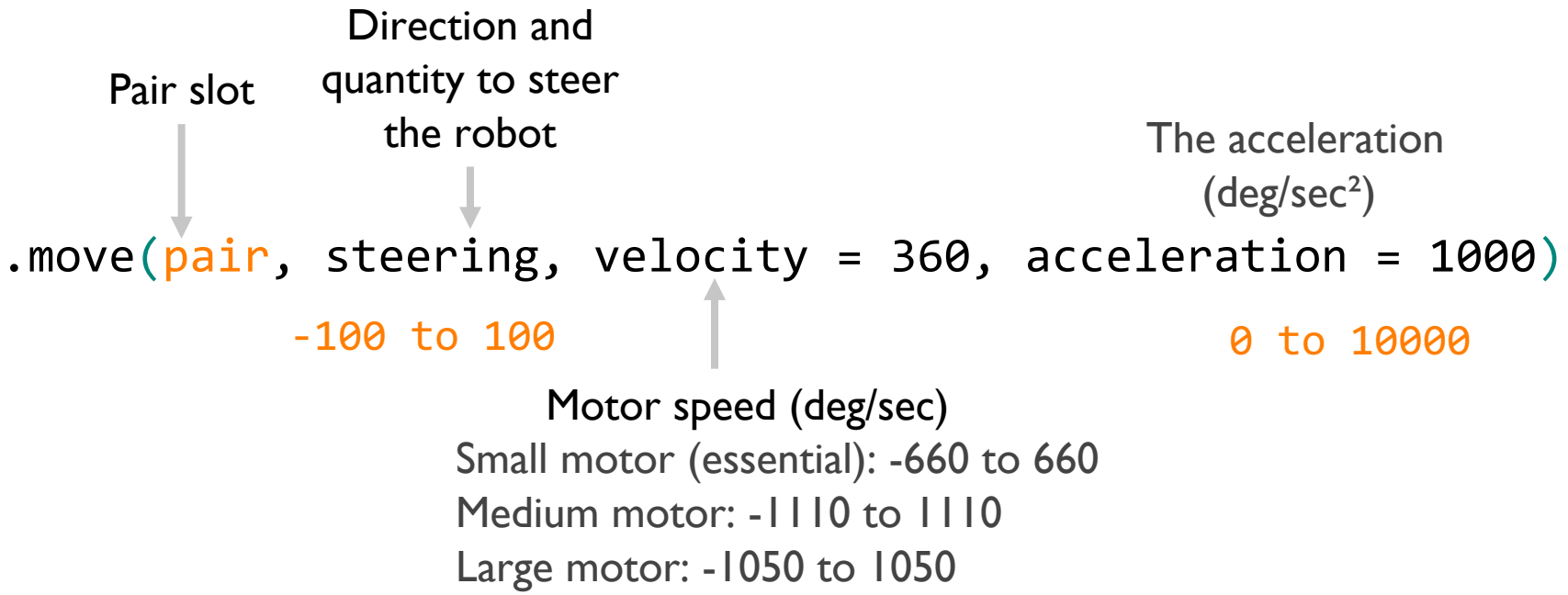
Once a pair is set up, you can use it in move commands it by its Pair name. You don't have to use the port names.

To reset the name, use unpair:

```
motor_pair.unpair(motor_pair.PAIR_1)
```

MOTOR_PAIR.MOVE()

Move is a **synchronous** function that moves the pair until a stop is received



`velocity=360` and `acceleration=1000` are the default values if nothing is set. Positive steering values turn the robot right, negative turn left. Larger values turn more sharply, e.g., 0 moves straight, +/-50 moves one wheel only, +/-100 spins the robot.

MOTOR_PAIR.STOP()

Stop is a **synchronous** function that stops a motor pair

```
.stop(pair, stop=motor.BRAKE)
```

See the "Configuring Robot Movement" lesson for a description of the Stop modes. The Knowledge Base also has good information on it.

MOTOR_PAIR.MOVE_FOR_DEGREES()

move_for_degrees moves the motor pair by a certain number of degrees

It is **asynchronous** – use **await** if you want to wait for it to finish

Usually, we want to move the robot by a distance

See the "Configuring Robot Movement" lesson for a description on how to write a function to take in a distance and return the degrees the motor pair needs to turn to move the robot by that distance. It depends on the physical characteristics of your robot.

```
.move_for_degrees(pair, degrees, steering=0, velocity = 360,
          stop = motor.BRAKE, acceleration = 1000)
```

MOTOR_PAIR.MOVE_TANK/MOVE_TANK_FOR_DEGREES()

These functions are similar to the move and move_for_degrees, except they let you set individual velocities for the left and right motors

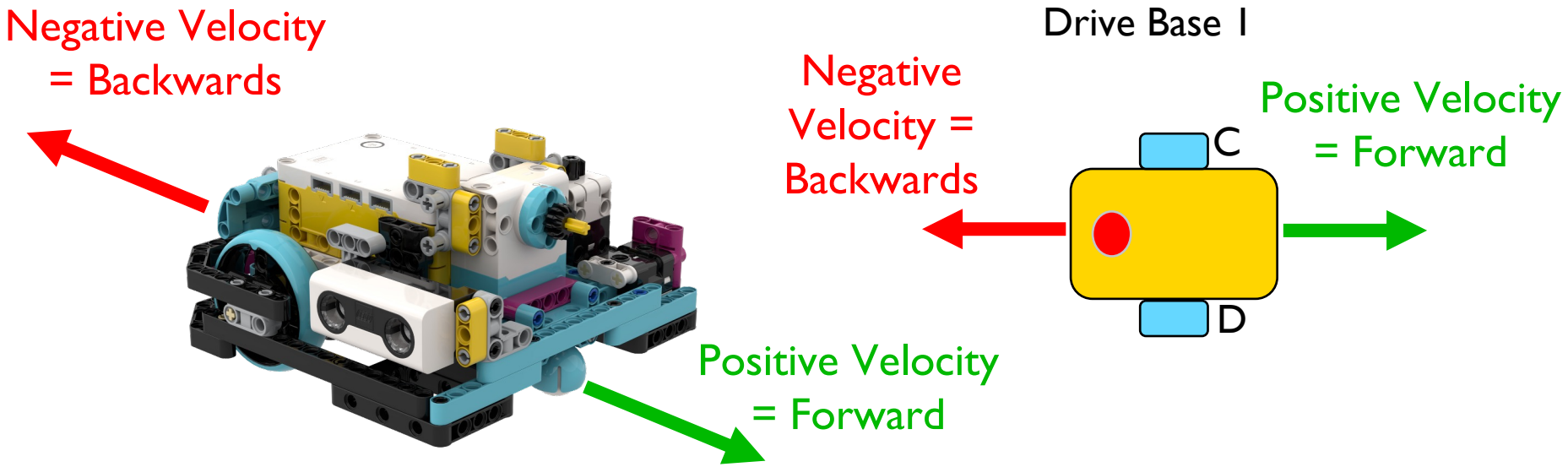
Can be useful for moving each wheel independently, e.g for turning or squaring on a line

NEGATIVE VALUES

You can enter negative values for velocity or degrees

This will make the robot move backwards

If you negate two values (e.g., velocity and degrees negative), the robot will move forward.



CHALLENGE 1: MOVE 10 CM

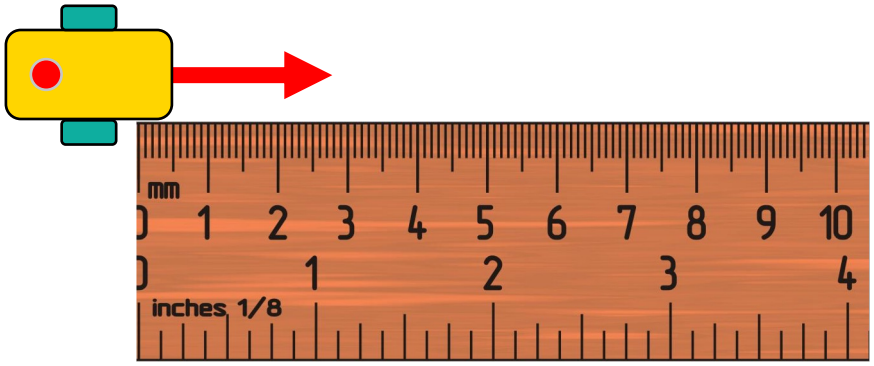
Move the robot 10 centimeters forward

Basic steps:

Configure your robot

Use a MotorPairs method (move_for_degrees() or move_tank_for_degrees()) to move forward for 10cm

Drive Base 1



CHALLENGE 1 SOLUTION

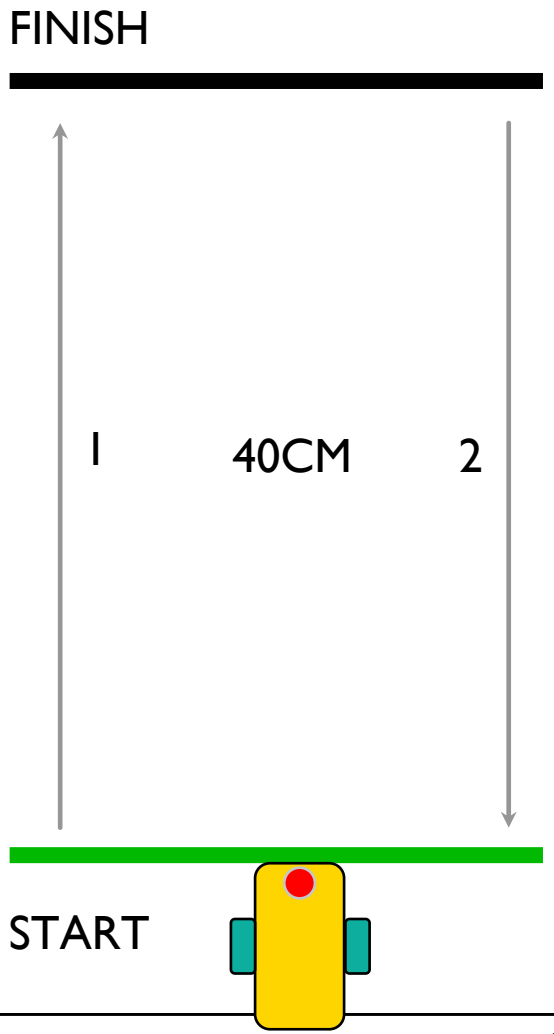
```
from hub import port
import runloop, motor_pair, sys

# cm, this is a constant for your robot
WHEEL_CIRCUMFERENCE = 17.5

# input must be in the same unit as WHEEL_CIRCUMFERENCE
def degreesForDistance(distance_cm):
    # Add multiplier for gear ratio if needed
    return int((distance_cm/WHEEL_CIRCUMFERENCE) * 360)

async def main():
    # Drive Base 1
    motor_pair.pair(motor_pair.PAIR_1, port.C, port.D)
    await motor_pair.move_for_degrees(motor_pair.PAIR_1, degreesForDistance(10), 0)
    sys.exit("Finished")

runloop.run(main())
```

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 11 SOLUTION

```
from hub import port
import runloop, motor_pair, sys

# cm, this is a constant for your robot
WHEEL_CIRCUMFERENCE = 17.5

# input must be in the same unit as WHEEL_CIRCUMFERENCE
def degreesForDistance(distance_cm):
    # Add multiplier for gear ratio if needed
    return int((distance_cm/WHEEL_CIRCUMFERENCE) * 360)

async def main():
    # Drive Base 1
    motor_pair.pair(motor_pair.PAIR_1, port.C, port.D)
    await motor_pair.move_for_degrees(motor_pair.PAIR_1, degreesForDistance(40), 0)
    await motor_pair.move_for_degrees(motor_pair.PAIR_1, degreesForDistance(-40), 0)
    sys.exit("Finished")

runloop.run(main())
```

CREDITS

This lesson was created by Arvind and Sanjay Seshan for Prime Lessons

Additional contributions by FLL Share & Learn community members

More lessons are available at www.primelessons.org