Programming the RI5 hub with Python

Hi, I'm Shah. I'm a professional trainer specialising in MacOS and iOS software and hardware, and I also conduct classes on iOS App Development and LEGO Mindstorms.

In the last lesson, you learned how to create your first Python program. Today, you'll learn more about the Python programming user interface, learn about the program you created, and view the programming documentation in the Robot Inventor app. You'll also learn how to make your robot move forward and backwards. Are you ready? Great! Let's go!

What you'll need:

- LEGO Mindstorms Robot Inventor (set no. 51515)
- A computer (Mac/PC), tablet (iOS/Android) or phone (iOS/Android) with the Mindstorms Robot Inventor app installed

Before you begin

For this lesson you will need to build the Tricky robot. Make sure the batteries in your programmable brick and device are charged.

The Python programming user interface

Launch the Robot Inventor app and choose the same program that you worked on last time. It should look like the screen shot at the below.

Let's learn more about the Python programming user interface.

- You type code in the **Editing area**.
- Clicking the **Programming Guide icon** displays details of the Python commands you can use.
- The **Zoom controls** allow you to set the default zoom level, zoom out and zoom in.
- The **Undo/Redo** buttons allow you to undo or redo your actions.
- The **Console button** brings up a panel which displays error messages and other information.
- The **Program slot** button allows you to choose which slot on the hub will be used to store the program.
- The **Stop** button stops any currently running program.
- The **Play** button copies your program to the selected slot and runs it.

About your program

Let's learn a little more about the program you created in the last lesson. There you ran the program without learning much about it. Let's now look at it in more detail.

The first four lines of your program imports the various libraries and other code needed to make your robot work.

The hub = MSHub() command creates a representation of your robot, Tricky in this case. You need to do this so that you can give it commands.

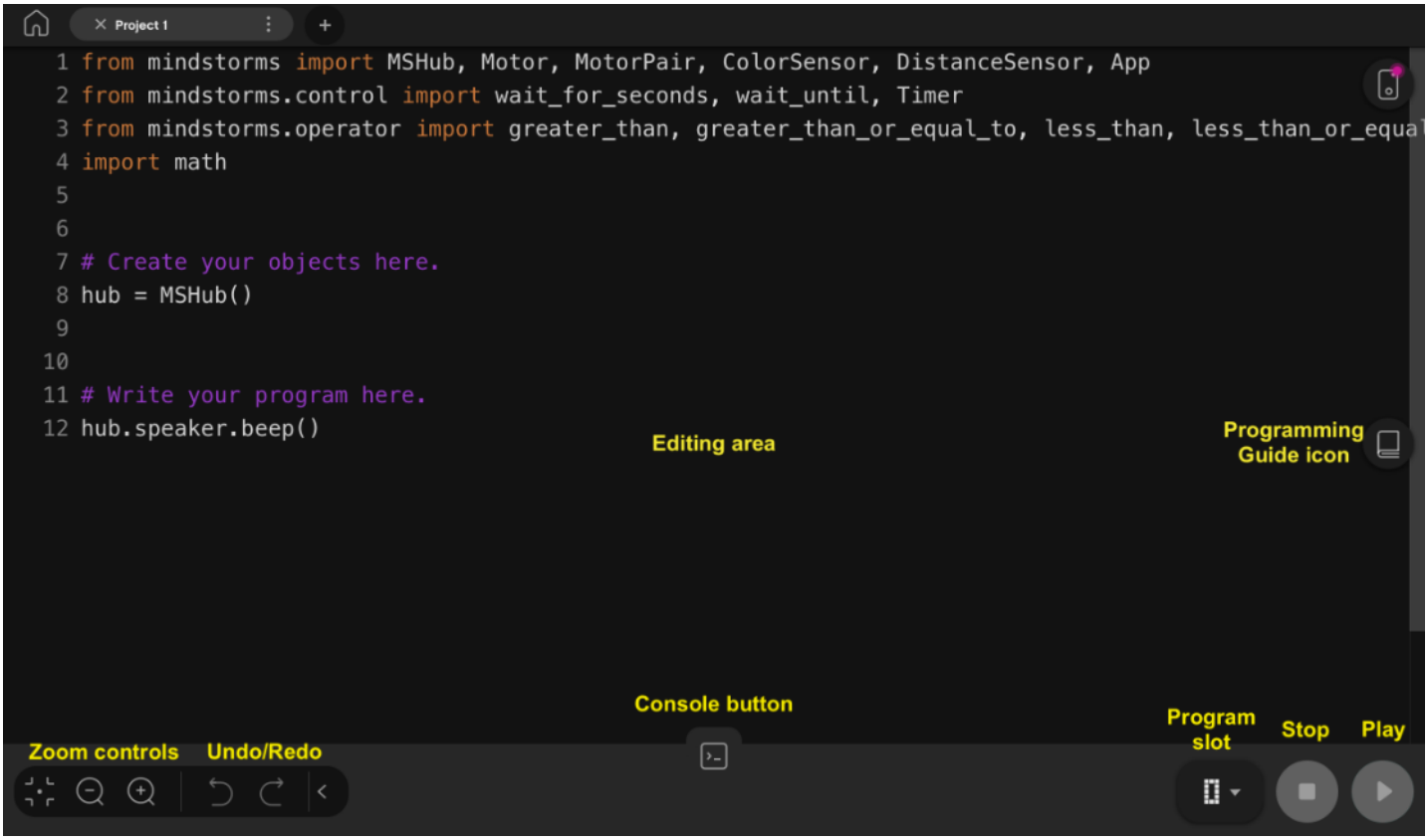
The final line of code tells Tricky to beep.

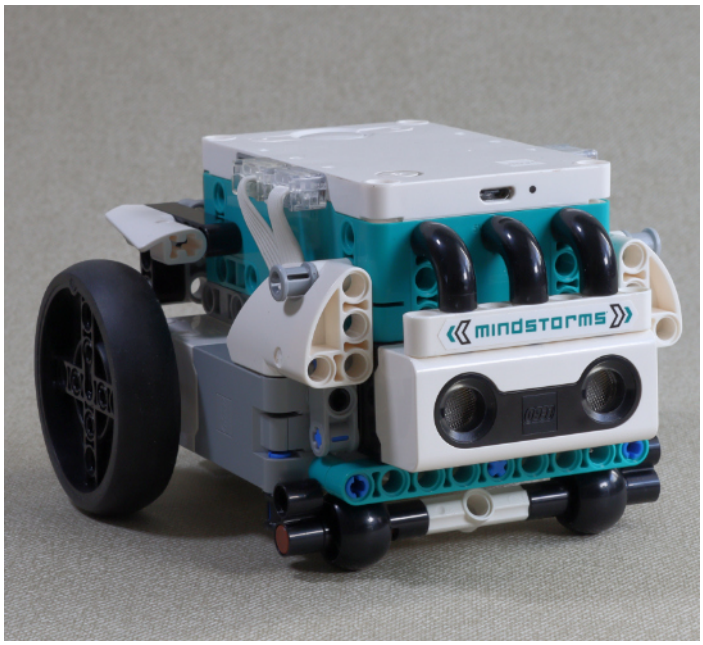
Making your robot move straight

Now let's modify your program to make Tricky move forward and backwards.

You'll be modifying the program that you used in the last lesson to make Tricky move forward and backwards. Follow these steps:

1. Click the Programming Guide icon and search the documentation for details about Motor Pairs. Read about how to initialise motor pairs and how to issue motor pair commands. It's ok if you don't yet understand this as it will be explained in subsequent steps.





2. Remove the line of code that makes the robot beep. Add the following line of code to your program to create a motorPair object and assign it to a motorPair variable:

```
motorPair = motorPair( A , B )
```

The motorPair object represents the motor configuration on Tricky. Note that Tricky has two motors with a wheel connected to each one. The left motor is connected to port A and the right motor is connected to port B. Once the motorPair object has been created, you can program it to make your robot move and turn.

3. Add the following lines of code to your program to make Tricky drive forward:

```
motorPair.set_default_speed(30)
motorPair.move(2, rotations , steering=0)
```

The first line sets the speeds of the motors to 30. You can set the speed to any value between -100 and 100. Setting the default_speed to a positive value makes Tricky drive forwards; setting it to a negative value makes Tricky drive backwards. The second line makes Tricky drive until the wheels have completed two rotations. Setting the steering to 0 makes both motors move at the same speed.

4. Add the following line of code to make Tricky pause for a second:

```
wait_for_seconds(1)
```

5. Add the following lines of code to make Tricky drive backwards:

```
motorPair.set_default_speed(-30)
motorPair.move(720, degrees , steering=0)
wait_for_seconds(1)
```

The first line sets the speeds of the motors to -30. Because this is a negative value it makes Tricky drive backwards. The second line instructs Tricky to move until the wheels have rotated 720 degrees (2 complete rotations). As before, setting the steering to 0 makes both motors move at the same speed. The third line makes Tricky pause for one second as it did before.

6. Add the following lines of code to make Tricky drive forwards again:

```
motorPair.set_default_speed(30)
motorPair.move(2, seconds , steering=0)
```

The first line sets the speeds of the motors back to 30, and the second line instructs Tricky to move forward for two seconds.

7. Verify that the complete program looks like the screenshot below:
Run your program. Tricky should move forward for two rotations, backward for two rotations and move forward again for two seconds.

You have just programmed Tricky to move forward and backwards! Yay!
In the next lesson, you'll make Tricky turn while moving. See you then!