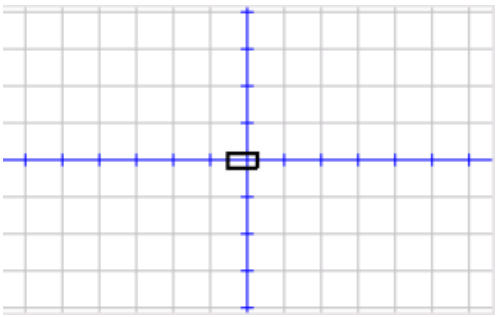
Unit 6: Coordinates | **Skill Builder 1: Introduction to Coordinates**

In this first lesson for Unit 6, you will learn about the TI-Innovator Rover's coordinate system and movement to coordinates.

Objectives:

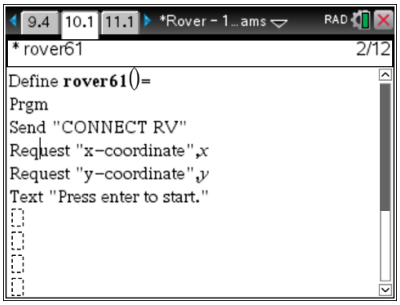
- Understand the Rover coordinate system and initial position and heading
- Make the Rover move to a certain point on the coordinate plane
- Use mathematics to determine distance

The Rover has a 'built-in' coordinate system just like a graphing system. When you **Send** "**CONNECT RV**", the Rover's position on the coordinate grid is set to (0,0) and its heading is 0 degrees (points toward the positive x-axis).



FORWARD moves the Rover to the right.
LEFT turns the Rover 90 degrees counterclockwise.

Our program will tell the Rover to move to a point on its coordinate grid. We'll use **Request** statements to enter values for *x* and *y* and then make the Rover drive to the point (x, y) and then move back to the origin.



1. Start your program with the usual instructions.

2. Include **Request** statements for *x* and *y*.

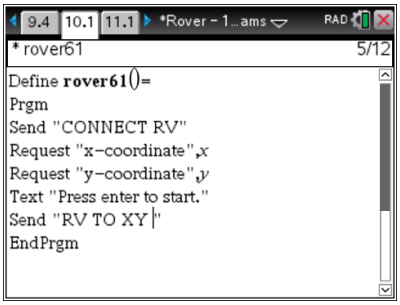
   Recall that **Request** will display the message entered in quotes (the 'prompt').

3. The **Text** statement gives you time to place the Rover at the origin and face the Rover in the 'right' direction.

4. Add the Rover command to drive **TO XY**. The command is found in the **menu > Hub > Rover (RV) > Drive RV** menu. and will appear in your program as an incomplete statement.

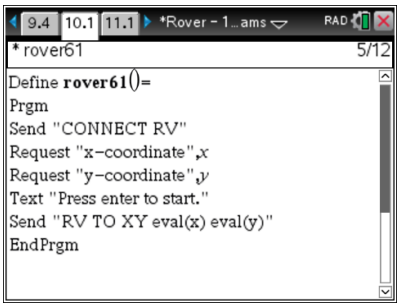
   Send "RV TO XY "

   The command will appear in your program as an incomplete statement.



©2018 Texas Instruments Incorporated

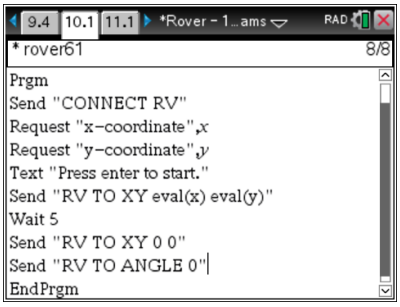
5. The x- and y-coordinates must be added and will be stored in the variables *x* and *y,* respectively. In order for the TI-Innovator™ Hub to use these values, you must use the **eval( )** function twice.



6. Add **eval(x) eval(y)** to the command.
   Send "RV TO XY eval(x) eval(y)"

7. Test your program now. Depending on the coordinate values you enter, the Rover will move to that position.

8. Add a **Wait** command to let the Rover move to your point, and then tell the Rover to return to the origin. Simply use the numbers 0 and 0 separated by a space. We also add a statement to make the Rover point in its original direction (**TO ANGLE 0**).



9. Test your program again. This time, the Rover should travel to your entered point and then return to the origin and face in the original direction.