



#### Unit 4: Driving Features

#### Skill Builder 1: Make Rover Move!

In this lesson, you will begin to operate the TI-Innovator Rover robotic vehicle by making 'Rover' move and turn.

#### Objectives:

- Make Rover move
- Make Rover drive in a pattern

Python's TI-Innovator **Rover** commands all begin with **rv**. This syntax is a result of the way that the **ti\_rover** module is imported into your programs (**import ti\_rover as rv**). The same is true of the **ti\_plotlib** module. This technique is called 'aliasing' the module name (replacing it with a *shorter* name).

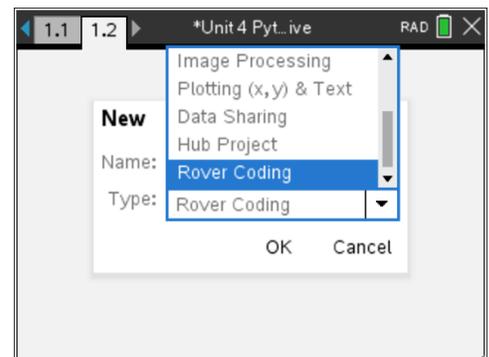
When you write a program that operates Rover, you are controlling the vehicle through easy-to-use commands but there's a lot going on 'behind the scenes' just like other TI-Innovator™ Hub commands.

Be sure your TI-Nspire CX II is attached and connected to Rover and that Rover's power is turned on.



**Teacher Tip:** New to Python? Check out 'aliasing module Python' online.

1. Begin a new Python program and select the **Rover Coding** template from the **Type:** dropdown after entering the name of the program.

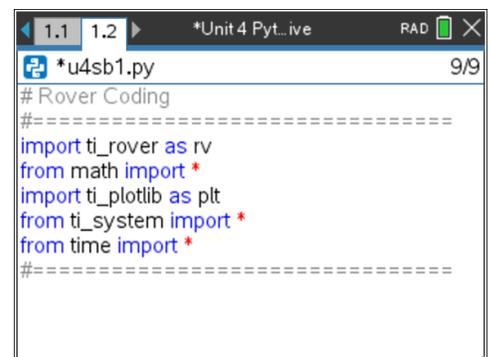


2. The template contains many **import** statements to give easy access to several useful functions.

The most important one right now is the first statement:

**import ti\_rover as rv**

This form of the import statement indicates that you *must* use **rv**. when using any of the methods in this module. When you choose the Rover functions from the **menu**, the **rv**. is automatically included.



**Teacher Tip:** Remember that it is also possible to simply type statements into the Python editor but when controlling the Rover, **rv**. is still required and is a possible source of errors.



3. To make Rover go forward use **menu > TI Rover > Drive** to get this statement:

**rv.forward(distance)**

Notice that the menu does not include **rv.** in front of **forward** but **rv.** is pasted into your program. This is done to save space on the menu.

The distance prompt is measured in 'grid units' and you will be able to determine what that means in a moment. Use the number **1** as the distance and run the program. If Rover moves, hooray!

But...how *far* did Rover move?

**Teacher Tip:** Rover's default unit is 10 cm or 1 decimeter or 0.1 m. Distance units are in meters and angle units are in degrees. If you *delete* the `distance` prompt, Rover moves 1 meter.

```

1.1 1.2 *Unit 4 Pyt...ive RAD
*u4sb1.py 9/9
# Rover Coding
#-----
import ti_rover as rv
from math import *
import ti_plotlib as plt
from ti_system import *
from time import *
#-----
rv.forward(distance)

```

4. Try the basic drive commands: **forward**, **backward**, **left**, and **right**.

- 1 forward(distance)
- 2 backward(distance)
- 3 left(angle\_degrees)
- 4 right(angle\_degrees)

**Teacher Tip:** This is a good place for exploration.

5. To drive a square:  
Erase the drive commands you entered (or make them comments).  
Use a **for** loop to make Rover drive in a square pattern.  
Begin with:

**for index in range(size):**

Use an appropriate `index` and `size` to make a square and add just two statements in the loop block. Try it yourself and run the program.

```

1.1 1.2 *Unit 4 Pyt...ive RAD
*u4sb1.py 9/11
# Rover Coding
#-----
import ti_rover as rv
from math import *
import ti_plotlib as plt
from ti_system import *
from time import *
#-----
for index in range(size):
  <<block

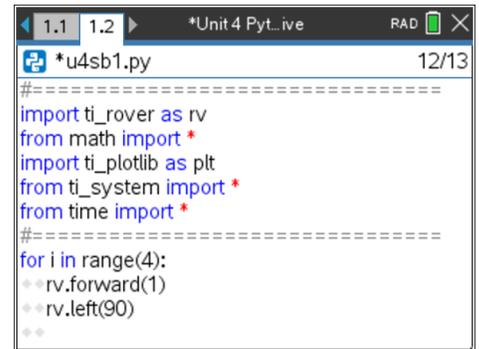
```



6. Did you write something like this?

Congratulations! Remember to save your work.

What other patterns can you make?



```
*Unit 4 Pyt_ive RAD X
*u4sb1.py 12/13
#=====
import ti_rover as rv
from math import *
import ti_plottlib as plt
from ti_system import *
from time import *
#=====
for i in range(4):
    rv.forward(1)
    rv.left(90)
    
```

**Teacher Tip:** The **Drive with Options** menu offers many customized drive statements related to distance, time, and rate using units such as meters, grid units, wheel revolutions (for forward and backward), and degrees, radians or gradians (for angles). These options are not covered in this course but make for interesting exploration opportunities.