

Meet TI-Rover

Geometry Challenges Day

TI-84 Plus CE

Python

Texas Instruments

@ticalculators



www.TIstemProjects.com

Contact stem-team@ti.com with questions

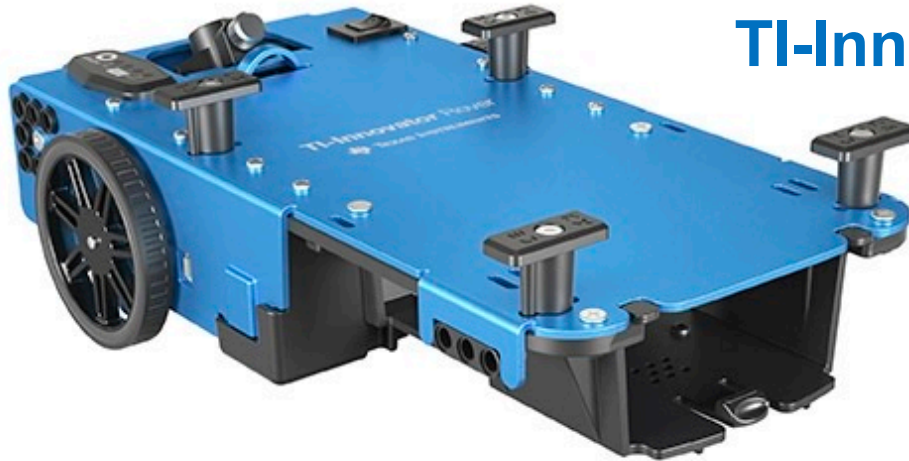
Meet the TI-Innovator™ Rover



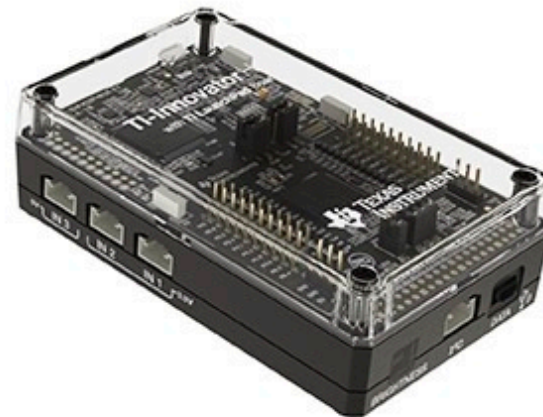
TI Graphing Calculator



TI-Innovator™ Rover



TI-Innovator™ Hub



Rover from the top

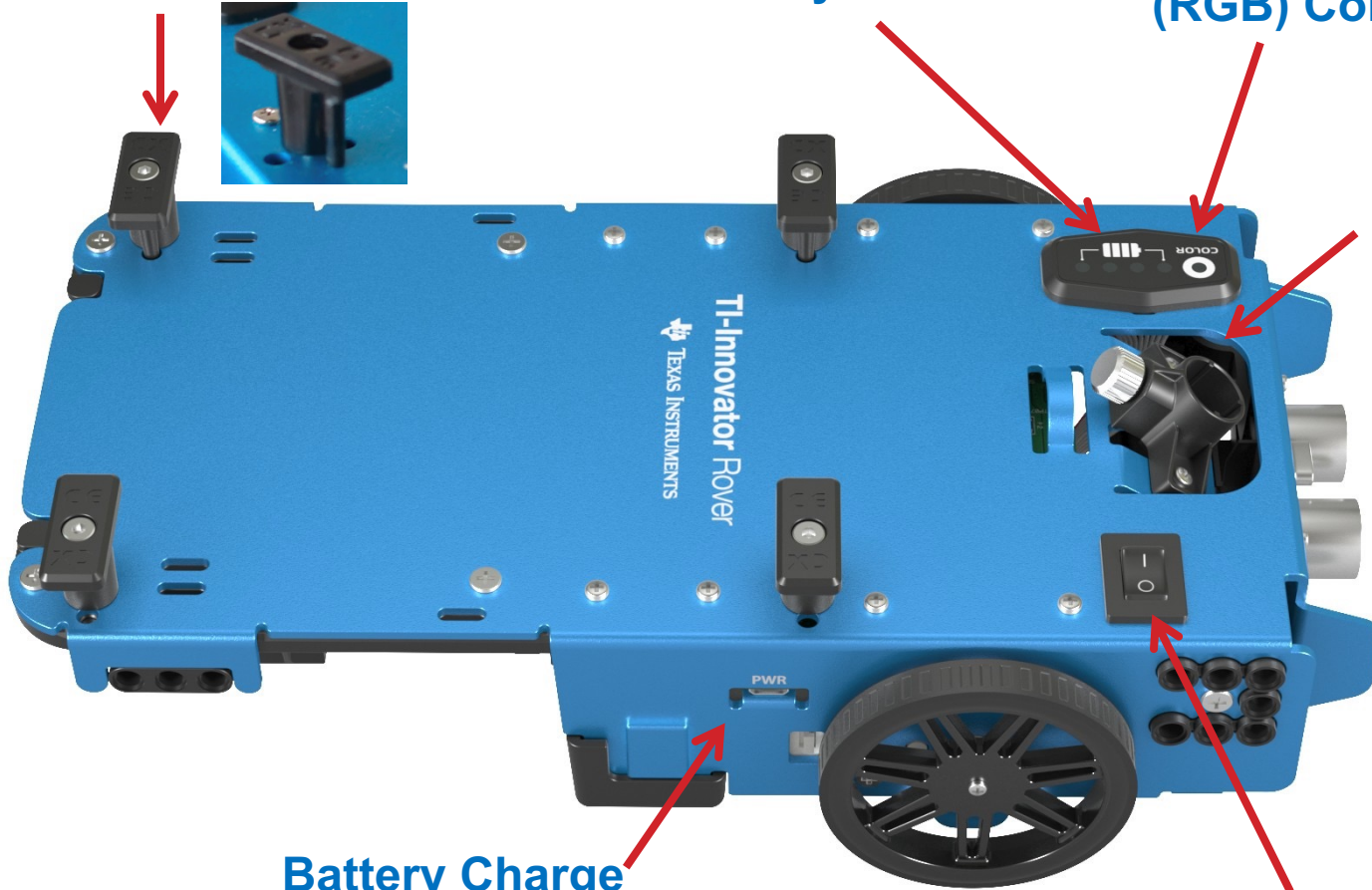
Calculator holder posts.
Lift and twist to CE or CX side.



Battery indicator

Red-Green-Blue
(RGB) Color LED

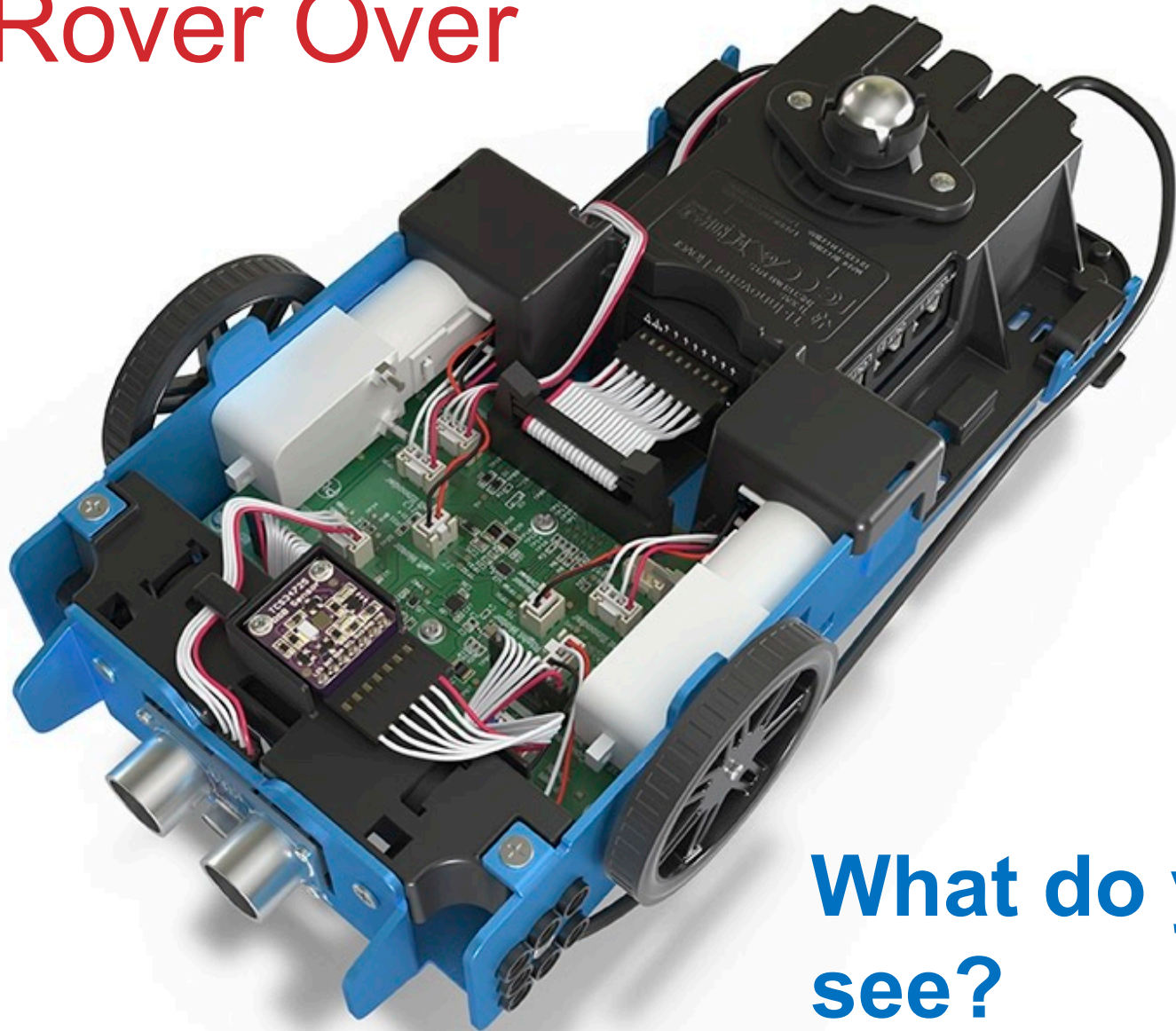
Marker holder
(Expo Fine and
Ultra Fine sizes)



Battery Charge
with USB micro
to wall adapter.

On/Off Switch

Turn Rover Over



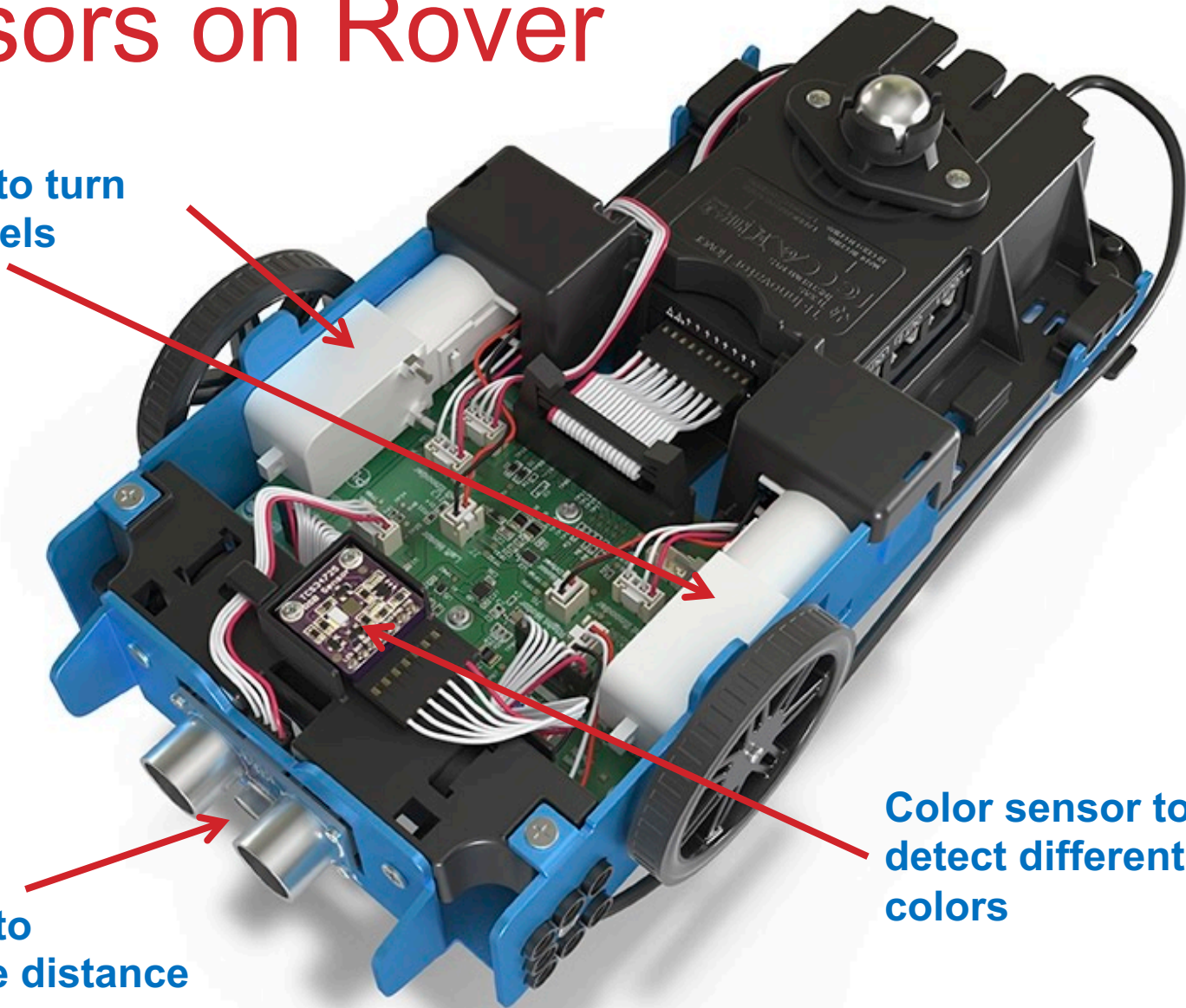
**What do you
see?**

Sensors on Rover

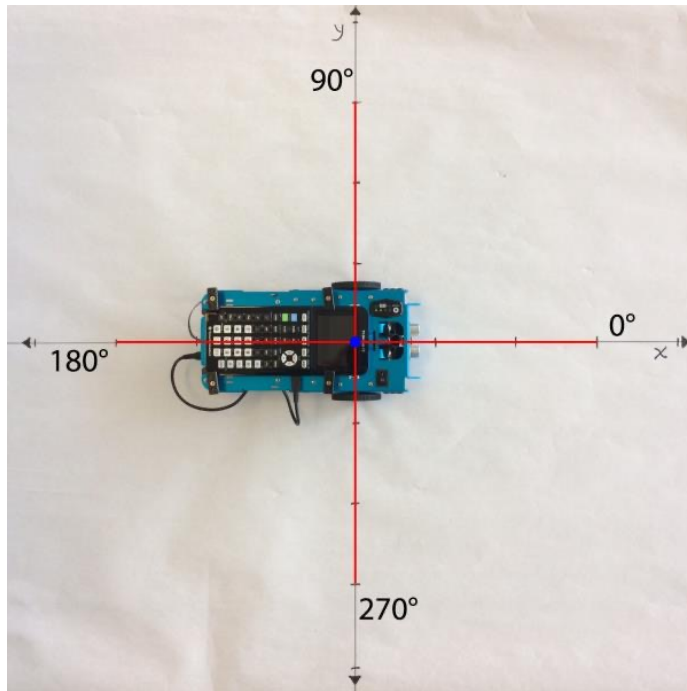
Motors to turn the wheels

Ranger to measure distance

Color sensor to detect different colors



TI-Rover orientation and virtual grid



Rover programs set the initial position as the origin and the heading as 0 degrees measured from the x-axis.

Note: The Rover tracks its position on a virtual coordinate grid with a unit value of 10 cm. The coordinate grid position applies to the `to_xy(x,y)`, `to_polar(r,theta_degrees)` and `to_angle(angle, "unit")` functions on the Rover Drive menu. The virtual grid also applies to Path menu functions.

Connecting Rover to your calculator



1 Make sure that your Rover is switched on.

3 Plug A side into port on calculator the Rover Hub.

2 Plug B side into USB B port of the Rover Hub.



Unit-to-unit cable



Connecting your calculator to the Rover

Initial Connection

- » Step 1: Make sure that Rover is switched ON
- » Step 2: plug unit-to-unit cable into Hub
 - » (Use end of cable labeled B)
- » Step 3: plug unit-to-unit cable into the calculator
 - » (Use end of cable labeled A)

Troubleshooting

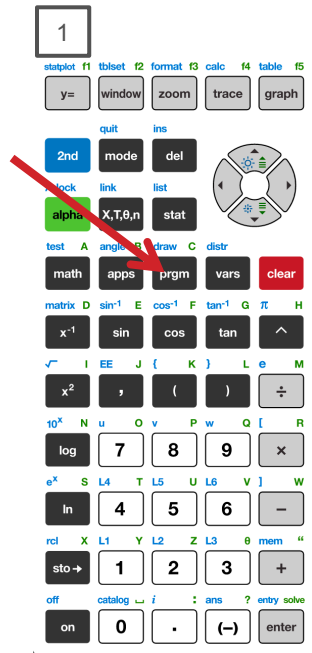
Try the following as a fast fix:

- » Step 1: unplug the unit-to-unit cable from the hub and the calculator.
- » Step 2: re-connect in the order of first Hub then calculator second.

If the fast fix does not work, try these steps:

- » Step 1: unplug unit-to-unit cable from both the hub and the calculator
- » Step 2: Switch Rover OFF
- » Wait a second
- » Step 3: Switch Rover ON
- » Step 4: plug unit-to-unit cable into Hub
- » Step 5: plug unit-to-unit cable into the calculator

Creating a new Python Program



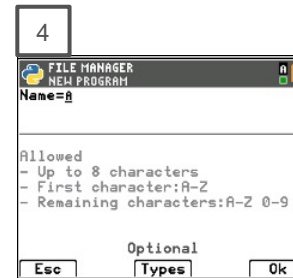
Press the **[prgm]** key to create, edit and execute TI-Python programs.



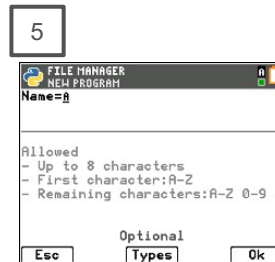
Press down arrow **[enter]** or Press **[2]** to select 2: Python App



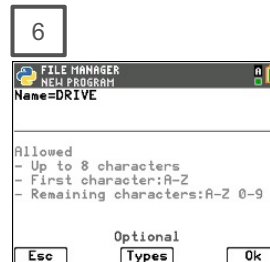
You have the option to run, edit, create or manage programs.



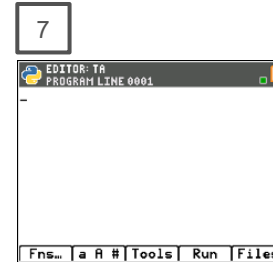
Press **[New]** softkey (zoom button)



You are prompted to enter a program name. The blinking A cursor shows that you are in alpha entry mode. The green alpha labels on the keys are active.



Type your program name and press **[Ok]**.

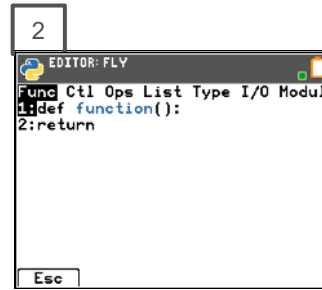


You are now in position to begin entering statements to your program.

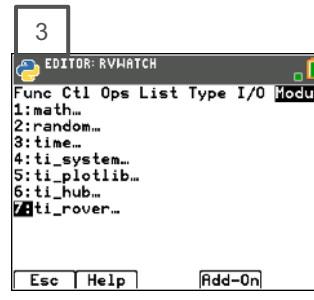
Entering a TI-Rover Program – importing the TI-Rover module and connecting to a Rover



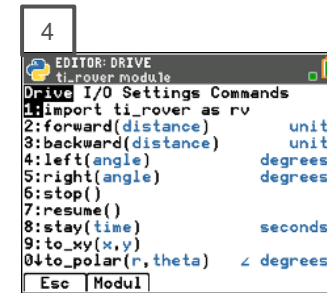
The Python program editor uses an insert cursor and a backspace delete. Press **[Fns...]** softkey to see functions to use in your program.



Press **right arrow** repeatedly or **left arrow** to move to the Modul menu.



You will see a menu of installed modules available to use functions from. Select **7:ti_rover**.



Select **1:import ti_rover as rv**.



The `ti_rover` module import statement is pasted to your program. The `ti_rover` import statement is required at the beginning of every Rover program. This import statement brings in Rover functions to use in your program, sets Rover's initial position and sets up communication between the Rover and the Hub.

Entering a TI-Rover Program

1

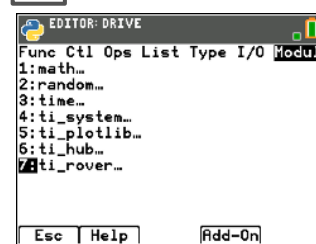


```
EDITOR: DRIVE
PROGRAM LINE 0002
import ti_rover as rv
_
```

Fns... a A # | Tools | Run | Files

You are now ready to enter functions to control your Rover. Navigate to the Rover menus by pressing [Fns...] then arrow to the Modul menu.

2

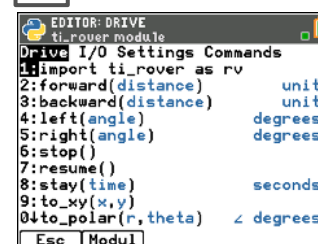


```
EDITOR: DRIVE
ti_rover module
Drive I/O Settings Commands
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
7:ti_rover...
```

Esc Help | Add-On

Then **select ti_rover...** to see options.

3



```
EDITOR: DRIVE
ti_rover module
Drive I/O Settings Commands
1:import ti_rover as rv
2:forward(distance) unit
3:backward(distance) unit
4:left(angle) degrees
5:right(angle) degrees
6:stop()
7:resume()
8:stay(time) seconds
9:to_xy(x,y)
04to_polar(r,theta) < degrees
```

Esc Modul

You begin on the Drive menu. **Select the 2:forward()** function.

4



```
EDITOR: DRIVE
PROGRAM LINE 0002
import ti_rover as rv
rv.forward(_
```

Fns... a A # | Tools | Run | Files

Enter a value for the number of Rover units to drive forward. **Arrow to the end of the statement** and press [enter] to move to the next statement.

A faster approach is to use [2nd] [enter] from any place on a line to complete the statement and move the cursor to the beginning of a blank line below.

Note: It is important that each statement begin on a new line.

5



```
EDITOR: DRIVE
PROGRAM LINE 0003
import ti_rover as rv
rv.forward(3)
_
```

Fns... a A # | Tools | Run | Files

Navigate to the Drive menu again by press [fns...], left arrow, 7:ti_rover...,4:left() to select the **left turn function**.

6



```
EDITOR: DRIVE
PROGRAM LINE 0003
import ti_rover as rv
rv.forward(3)
rv.left(_
```

Fns... a A # | Tools | Run | Files

Enter a value for the angle to turn in degrees. Press [2nd] [enter] to move to the next statement.

7



```
EDITOR: DRIVE
PROGRAM LINE 0004
import ti_rover as rv
rv.forward(3)
rv.left(180)
_
```

Fns... a A # | Tools | Run | Files

Navigate to the Drive menu again, then select **2:forward()**. After the function is pasted **enter the Rover units** to drive. Press [2nd] [enter] to move to the next statement.

8



```
EDITOR: DRIVE
PROGRAM LINE 0005
import ti_rover as rv
rv.forward(3)
rv.left(180)
rv.forward(3)
_
```

Fns... a A # | Tools | Run | Files

You are now ready to run your TI-Rover program.

Running a TI-Rover Program

1

```
EDITOR: DRIVE
PROGRAM LINE 0005
import ti_rover as rv
rv.forward(3)
rv.left(180)
rv.forward(3)
_
```

Fns... a A # Tools Run Files

You are now ready to run your program.

Before pressing **[Run]** go through the pre-drive checklist.

1. Make sure that TI-Rover is turned ON.
2. Make sure that the calculator unit-to-unit cable is connected to the Hub inside the Rover. Plug the B end of the cable into the Data USB B port of the Hub. Plug the A end of the cable into the calculator.
3. Press **[Run]**.

2

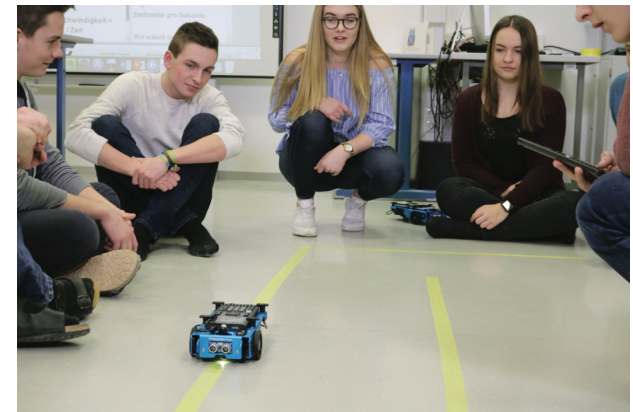
```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running DRIVE
>>> from DRIVE import *
>>> |
```

Fns... a A # Tools Editor Files

The program will run in the Python shell. You will receive messages on the status of the program.

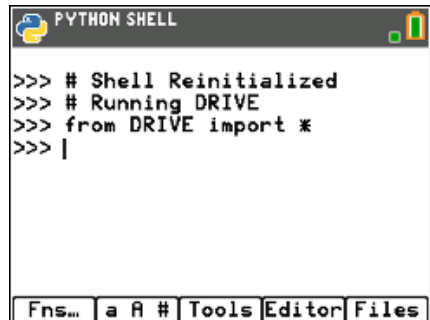
You can run the program again by pressing **[Tools]** and selecting **1:Rerun Last Program** from the menu.

You can return to the program editor by pressing **[Editor]**.



Editing a Rover Program

1



```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running DRIVE
>>> from DRIVE import *
>>> |
```

Fns... a A # Tools Editor Files

Press **[Editor]** to go back to your Python editor page.

2



```
EDITOR: DRIVE
PROGRAM LINE 0001
import ti_rover as rv
rv.forward(3)
rv.left(180)
rv.forward(3)
```

Fns... a A # Tools Run Files

Use the arrow keys to position the cursor to change the value of the forward distance.

3



```
EDITOR: DRIVE
PROGRAM LINE 0002
import ti_rover as rv
rv.forward(3_
rv.left(180)
rv.forward(3)
```

Fns... a A # Tools Run Files

Press **[del]** to backspace over the 3.

4



```
EDITOR: DRIVE
PROGRAM LINE 0002
import ti_rover as rv
rv.forward(_
rv.left(180)
rv.forward(3)
```

Fns... a A # Tools Run Files

Type in a new value for distance, **right arrow** to the end of the line, then **down arrow** to position the cursor to change the value of the second forward() function.

5



```
EDITOR: DRIVE
PROGRAM LINE 0004
import ti_rover as rv
rv.forward(5)
rv.left(180)
rv.forward(3_
```

Fns... a A # Tools Run Files

Press **[del]** to backspace over the current distance value. Type in a new value for distance, Press **[2nd]** **[enter]** to move to the next statement.

6

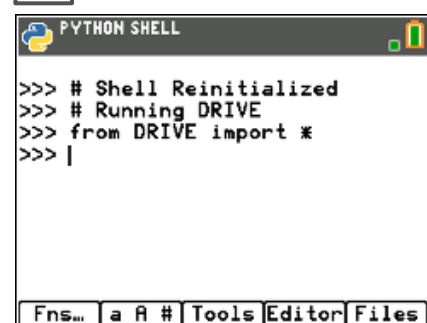


```
EDITOR: DRIVE
PROGRAM LINE 0005
import ti_rover as rv
rv.forward(5)
rv.left(180)
rv.forward(5)
-
```

Fns... a A # Tools Run Files

Press **[Run]** to run the program in the Python shell.

7



```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running DRIVE
>>> from DRIVE import *
>>> |
```

Fns... a A # Tools Editor Files

TI-Rover Module Menus

Drive

```
EDITOR: DRIVE
ti_rover module
Drive I/O Settings Commands
1:import ti_rover as rv
2:forward(distance) unit
3:backward(distance) unit
4:left(angle) degrees
5:right(angle) degrees
6:stop()
7:resume()
8:stay(time) seconds
9:to_xy(x,y)
0↓to_polar(r,theta) < degrees
A:to_angle(angle) degrees
B:forward_time(time) seconds
C:backward_time(time) seconds
D:forward(distance,"unit") ▶
E:backward(distance,"unit") ▶
F:left(angle,"unit") ▶
G:right(angle,"unit") ▶
H:forward_time(T,S,"unit") ▶
I↓backward_time(T,S,"unit") ▶
J:forward(D,"unit",S,"unit") ▶
K:backward(D,"unit",S,"unit") ▶
L:disconnect_rv() Disconnect
```

Input/Output (I/O)

```
EDITOR: DRIVE
ti_rover module
Drive I/O Settings Commands
1:Inputs...
2:Outputs...
3:Path...

EDITOR: DRIVE
ti_rover module
Inputs
1:ranger_measurement() meters
2:color_measurement() 1-9
3:red_measurement() 0-255
4:green_measurement() 0-255
5:blue_measurement() 0-255
6:gray_measurement() 0-255
7:encoders_gyro_measurement()
8:gyro_measurement() degrees
9:ranger_time() seconds

EDITOR: DRIVE
ti_rover module
Outputs
1:color_rgb(r,g,b) 0-255
2:color_blink(freq,time)
3:color_off()
4:motor_left(speed,time) ±255
5:motor_right(speed,time) ±255
6:motors("ldir",L,"rdir",R,T) ▶

EDITOR: DRIVE
ti_rover module
Path
1:waypoint_xythdrn()
2:waypoint_prev()
3:waypoint_eta()
4:path_done()
5:pathlist_x()
6:pathlist_y()
7:pathlist_time()
8:pathlist_heading()
9:pathlist_distance()
0↓pathlist_revs()
A:pathlist_cmdnum()
B:waypoint_x()
C:waypoint_y()
D:waypoint_time()
E:waypoint_heading()
F:waypoint_distance()
G:waypoint_revs()
```

Settings

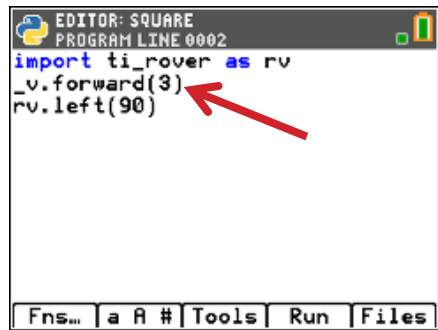
```
EDITOR: DRIVE
ti_rover module
Drive I/O Settings Commands
1:units/s
2:m/s
3:revs/s
4:units
5:m
6:revs
7:degrees
8:radians
9:grads
0:clockwise
A:counterclockwise
```

Commands

```
EDITOR: DRIVE
ti_system module
Drive I/O Settings Commands
1:from ti_system import *
2:sleep(seconds)
3:disp_at(row,"text","align") ▶
4:disp_clr() clear text screen
5:disp_wait() [clear]
6:disp_cursor() 0=off 1=on
7:while not escape(): [clear]
8:wait_until_done()
9:while not path_done():
0↓position(x,y)
A:position(x,y,heading,"unit") ▶
B:grid_origin()
C:grid_m_unit(scale_value)
D:path_clear()
E:zero_gyro()
```

Copying and Pasting a Line of Code

1

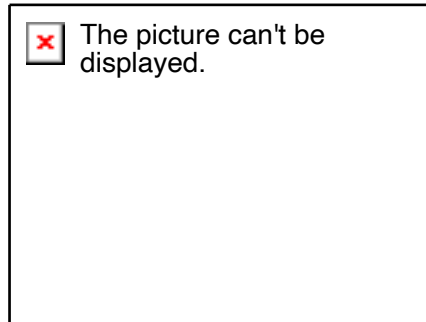


```
EDITOR: SQUARE
PROGRAM LINE 0002
import ti_rover as rv
rv.forward(3)
rv.left(90)
```

Fns... | a A # | Tools | Run | Files

Use **arrow keys** to move the cursor to a position anywhere on the line that you would like to copy.

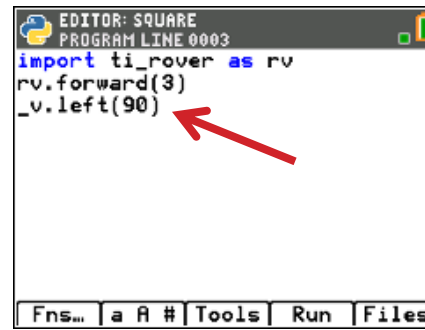
2



Press **[Tools]** then select **6:Copy Line** from the menu.

After you select you will be returned to the editor.

3

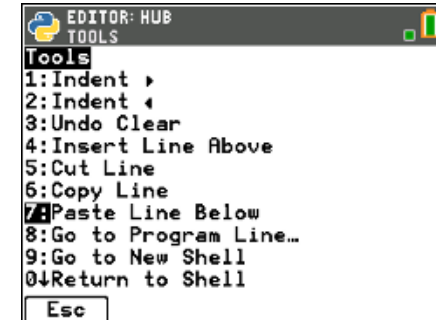


```
EDITOR: SQUARE
PROGRAM LINE 0003
import ti_rover as rv
rv.forward(3)
rv.left(90)
```

Fns... | a A # | Tools | Run | Files

Use **arrow keys** to move the cursor to any location on the line above where you would like to insert the copied line.

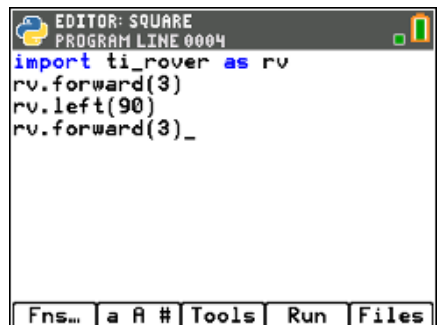
4



```
EDITOR: HUB
TOOLS
Tools
1:Indent >
2:Indent <
3:Undo Clear
4:Insert Line Above
5:Cut Line
6:Copy Line
7:Paste Line Below
8:Go to Program Line...
9:Go to New Shell
0↓Return to Shell
Esc
```

Press **[Tools]** then select **7:Paste Line Below** from the menu. The copied line will be pasted.

5

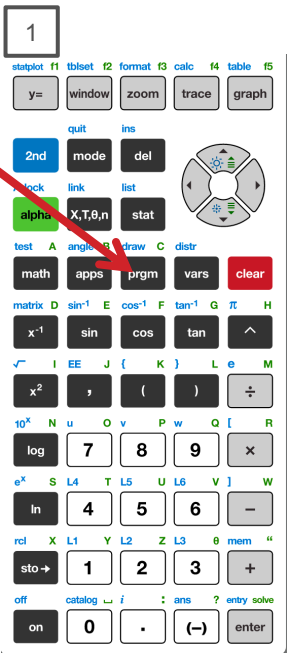


```
EDITOR: SQUARE
PROGRAM LINE 0004
import ti_rover as rv
rv.forward(3)
rv.left(90)
rv.forward(3)_
```

Fns... | a A # | Tools | Run | Files

You can paste again by returning to the **[Tools]** menu and selecting **7:Paste Line Below**.

Opening an existing Python Program File



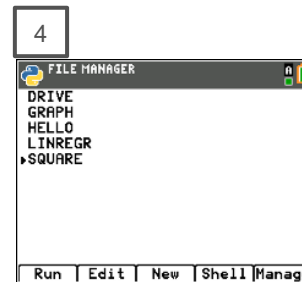
Press the **[prgm]** key to create, edit and execute TI-Python programs.



Press **[enter]** or Press **[2]** to select 2: Python App



To edit an existing program, use the **Up and Down Arrow** keys to select a program.



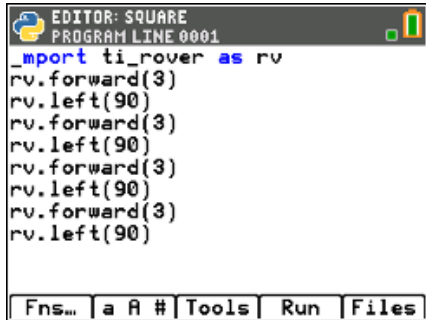
Press **[Edit]** to open with Python Editor with the selected program.



You can now make changes to the program or run the program.

Copying/Replicating a Python Program File

1




```
EDITOR: SQUARE
PROGRAM LINE 0001
import ti_rover as rv
rv.forward(3)
rv.left(90)
rv.forward(3)
rv.left(90)
rv.forward(3)
rv.left(90)
rv.forward(3)
rv.left(90)
```

Fns... a A # Tools Run Files

Press **[Files]** to return to the file management screen.

2

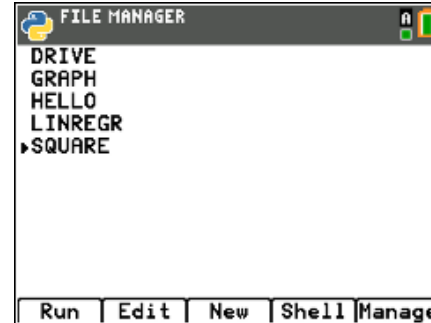


```
FILE MANAGER
DRIVE
GRAPH
HELLO
LINREGR
SQUARE
```

Run Edit New Shell Manage

Use the **Up and Down Arrow** keys to select a program.

3



```
FILE MANAGER
DRIVE
GRAPH
HELLO
LINREGR
SQUARE
```

Run Edit New Shell Manage

Press **[Manage]** to see file options.

4

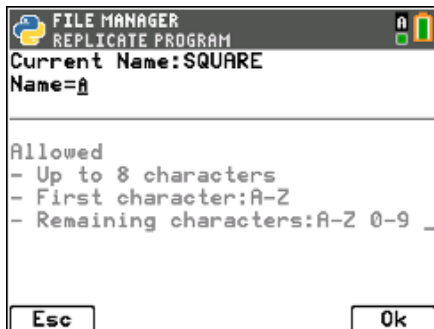


```
FILE MANAGER
MANAGE
Python App:v5.7.1.0022
1:Replicate Program...
2:Delete Program... [del]
3:Rename Program...
4>About...
5:Quit Python
```

Esc

Select **1:Replicate Program** to receive a prompt.

5




```
FILE MANAGER
REPLICATE PROGRAM
Current Name:SQUARE
Name=a
```

Allowed
- Up to 8 characters
- First character:A-Z
- Remaining characters:A-Z 0-9 _

Esc Ok

Type in the name of the new program using the green alpha key labels.

6



```
FILE MANAGER
REPLICATE PROGRAM
Current Name:SQUARE
Name=SQ2_
```

Allowed
- Up to 8 characters
- First character:A-Z
- Remaining characters:A-Z 0-9 _

Esc Ok

To use a number in the name, exit alpha mode by pressing **[2nd] [alpha]** then a **number** key.
Press **[Ok]** to finish the dialogue.

7



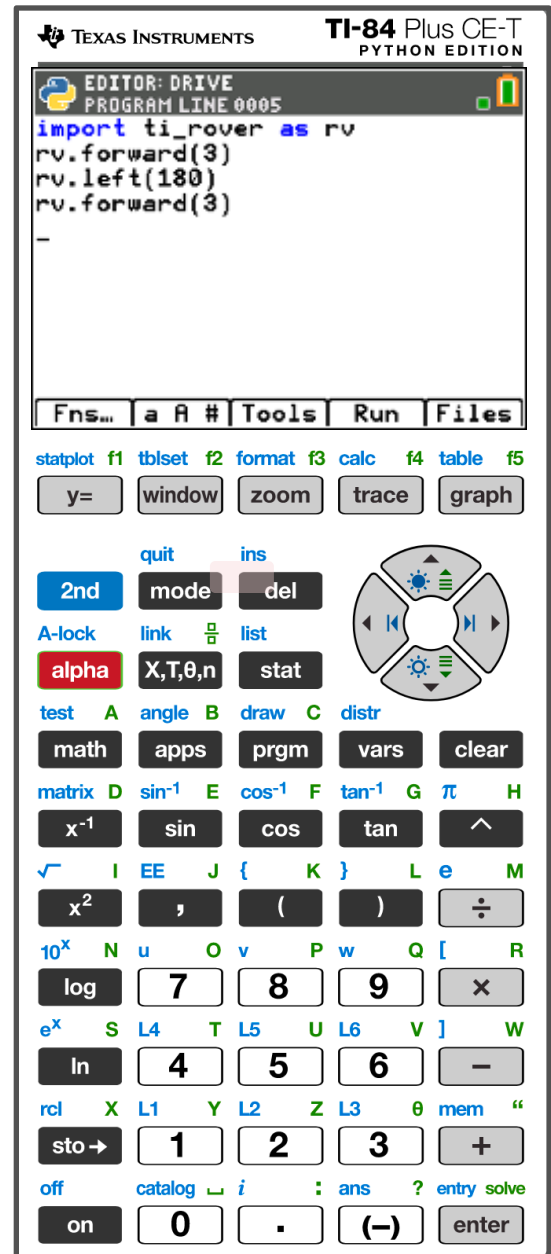
```
EDITOR: SQ2
PROGRAM LINE 0001
import ti_rover as rv
rv.forward(3)
rv.left(90)
rv.forward(3)
rv.left(90)
rv.forward(3)
rv.left(90)
rv.forward(3)
rv.left(90)
```

Fns... a A # Tools Run Files

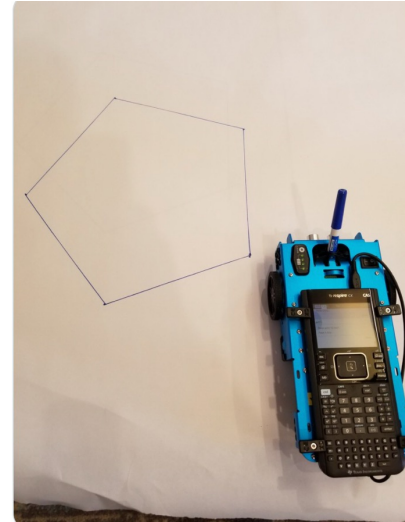
You are now in the editor ready to make changes or to run the new program.

Entry and Edit Tips

- » Use **number key shortcuts** or **arrow keys** and **[enter]** to select from menus
- » Use **arrow keys** to move the cursor around the screen.
- » Use **[alpha]** **repeatedly** to cycle from numeric, to lower case alpha to upper case alpha entry mode. The cursor indicates the current mode.
- » Use **[2nd] [A-lock]** to lock to alpha entry or to return to numeric entry.
- » Use **[Fns...]** **softkey** to bring up Python function menus, including the **Modul (modules)** menu.
- » Use **[clear]** or **[Esc]** **softkey** to back out of a menu.
- » Use **[del]** as a destructive backspace
- » Use **[2nd] [enter]** from any place on a line to complete the statement and move the cursor to the beginning of a blank line below.
- » Use **[Tools]** **softkey** menu to undo a clear and to copy, cut, paste and more.
- » Use **[Editor]** **softkey** to return to the editor from the Shell.
- » Use **[2nd] [quit]** to leave the Python app and return to the calculator.



Drawing with the TI-Rover



Use Expo Fine or Extra Fine dry erase markers.

The markers drop into a slot on the front of the Rover.

Note: The Texas Instruments Workshop Loan Rover cases include markers.

Drawing surface: We recommend butcher paper held in place with painters tape on a hard surface.

MAKE IT MOVE!

New Program:



```
EDITOR: DRIVE
PROGRAM LINE 0002
import ti_rover as rv
rv.forward(_
```

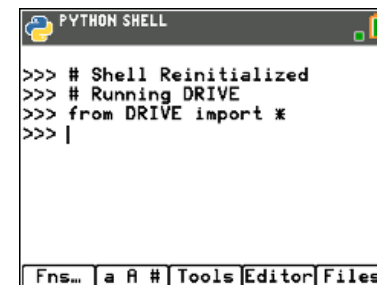
Fns... | a A # | Tools | Run | Files

Press [Fns...], left arrow, then 7:ti_rover... for the Rover menus.

Press [Run] to run the program in the Python shell.

Task: Discover how far Rover drives per unit.

Use differing values (1-20) to determine what 1 Rover unit is.



```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running DRIVE
>>> from DRIVE import *
>>> |
```

Fns... | a A # | Tools | Editor | Files

From the Python shell, press [Editor] to move from the shell to the Python editor.

Set the color

New Program:



```
EDITOR: MYCOLOR
PROGRAM LINE 0002
import ti_rover as rv
rv.color_rgb(.,.)
```

Fns... | a A # | Tools | Run | Files

Press [Fns...], left arrow, then 7:ti_rover... for the Rover menus.

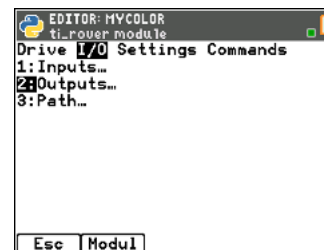
Press [Run] to run the program in the Python shell.

Task: Set the color output of the RGB LED.

Each color takes a value (0-255).

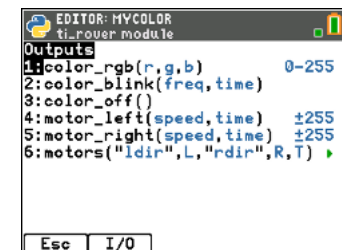
Challenge Task: Try to make Yellow

Find the color_rgb() function on the Rover Outputs menu. Enter values for the red, green and blue components of the color to display.



```
EDITOR: MYCOLOR
ti_rover module
Drive I/O Settings Commands
1: Inputs...
2: Outputs...
3: Path...
```

Esc | Modul



```
EDITOR: MYCOLOR
ti_rover module
Outputs
1: color_rgb(r,g,b) 0-255
2: color_blink(freq,time)
3: color_off()
4: motor_left(speed,time) ±255
5: motor_right(speed,time) ±255
6: motors("Idir",L,"rdir",R,T) ▶
```

Esc | I/O

Explore angles

New Program:

```
EDITOR: DRIVESQ
PROGRAM LINE 0001
import ti_rover as rv

rv.forward()
rv.left()
rv.forward()
rv.left()
rv.forward()
rv.left()
rv.forward()
rv.left()
```

Fns... | a A # | Tools | Run | Files

The program above is a framework for driving a square.
Enter values for distance and turn angle.

Press [Fns...], left arrow, then 7:ti_rover... for the Rover menus.

Press [Run] to run the program in the Python shell.

Task: Drive a square.

Challenge Task: Try to drive an equilateral triangle.

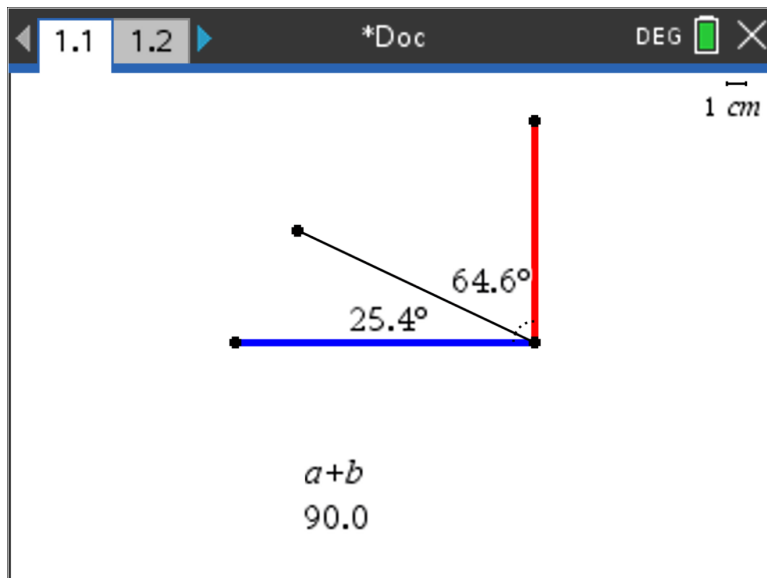
See the inputs for the most common drive functions below.

```
EDITOR: DRIVE
ti_rover module
Drive I/O Settings Commands
1:import ti_rover as rv
2:forward(distance) unit
3:backward(distance) unit
4:left(angle) degrees
5:right(angle) degrees
6:stop()
7:resume()
8:stay(time) seconds
9:to_xy(x,y)
0↓to_polar(r,theta) ∠ degrees
Esc | Modul
```

Quick Math Reminders

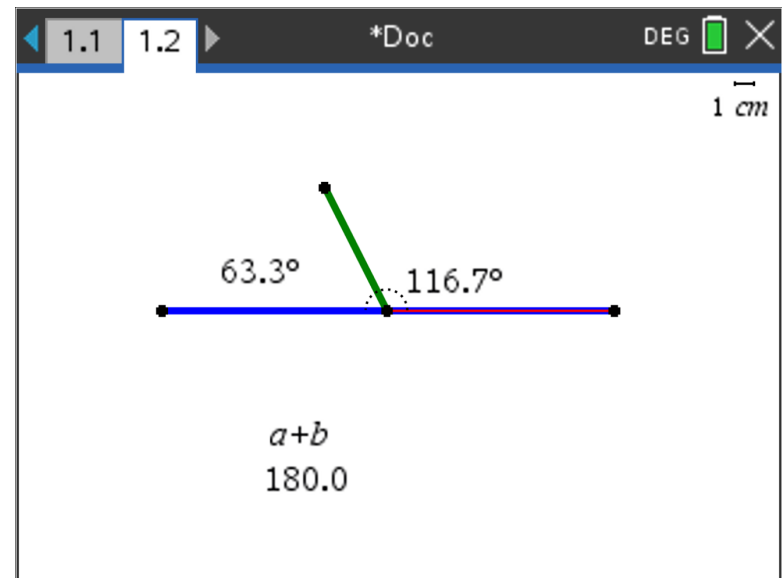
» Complementary Angles:

» Sum to 90 degrees



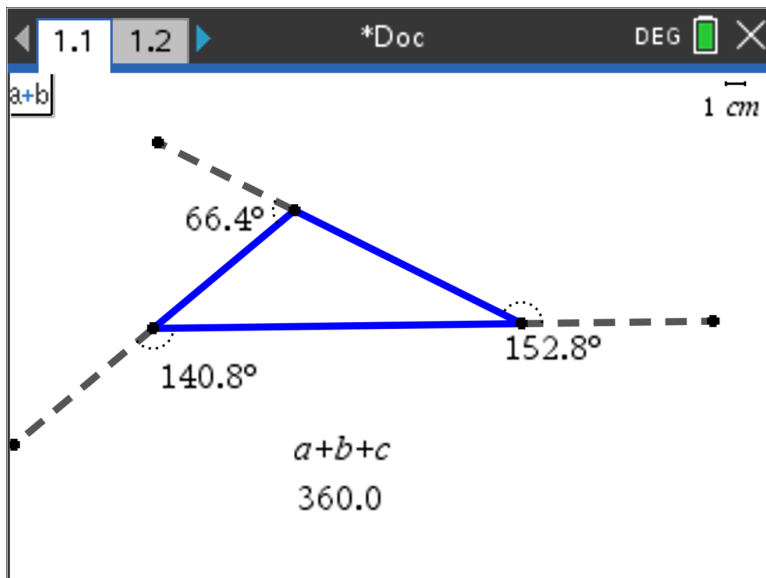
» Supplementary Angles:

» Sum to 180 degrees

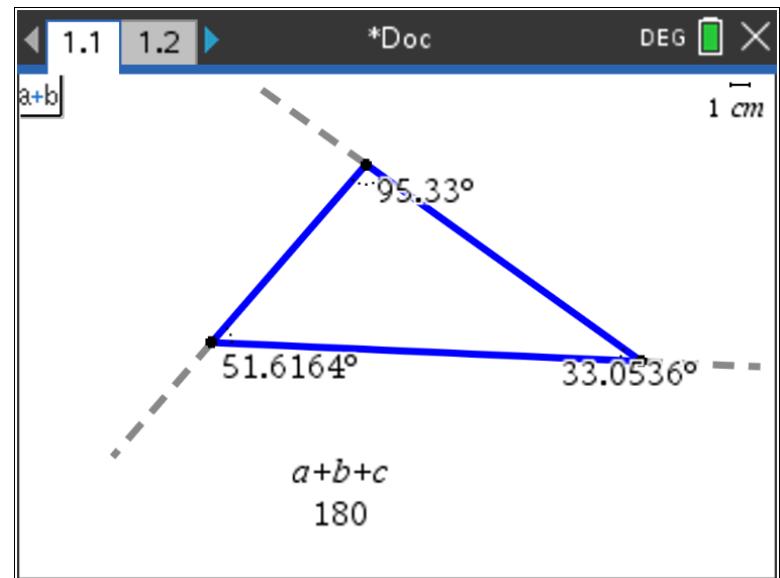


Quick Math Reminders

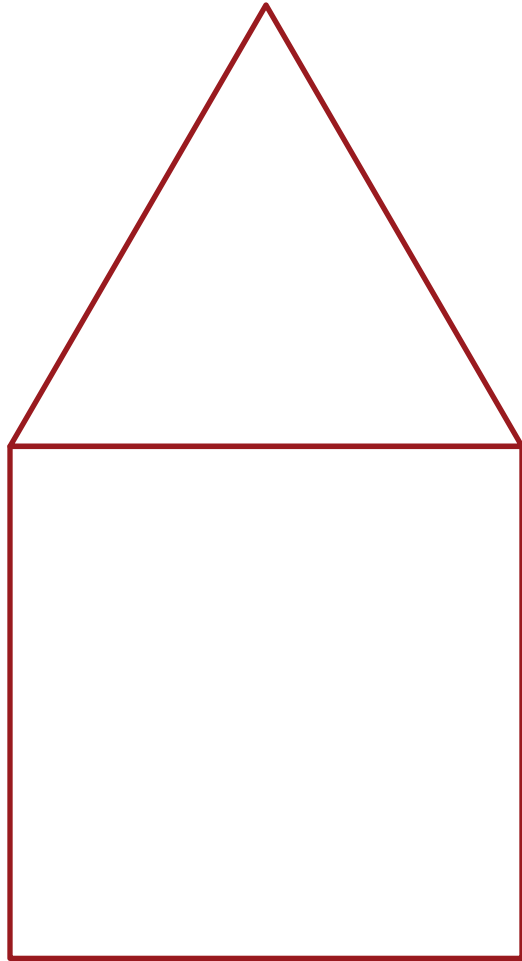
» Exterior angles:



» Interior Angles:



Logic Challenge

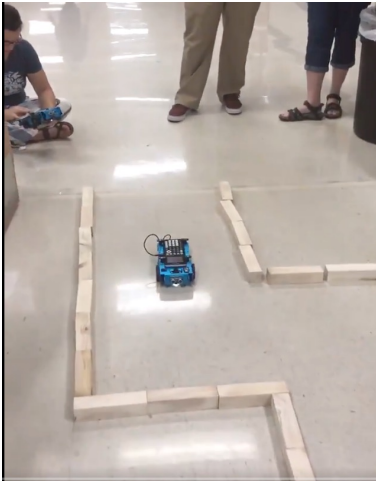


Task: Drive the figure shown without crossing any lines or going back over a line and without picking up the pen.

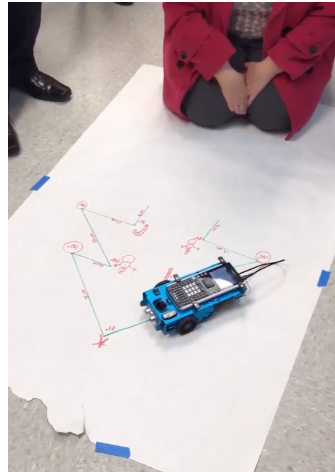
When you are ready put the pen in and trace your path



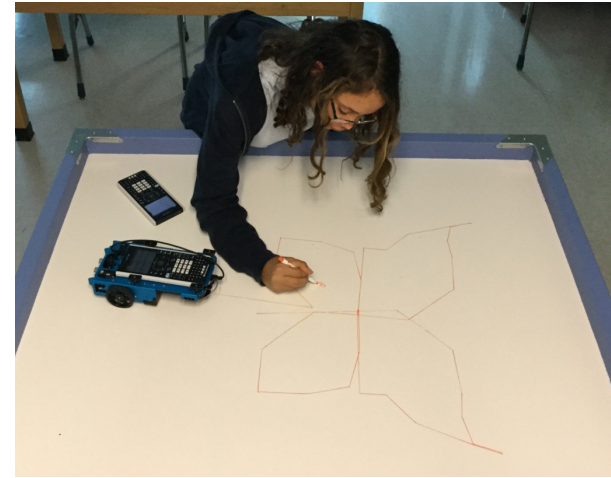
Where can you go next with TI-Rover?



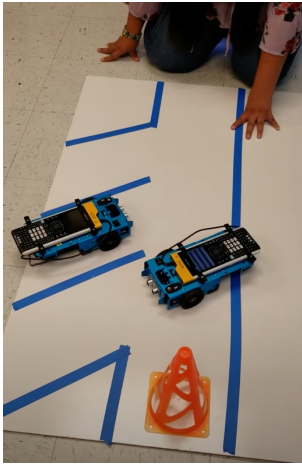
Drive an obstacle course



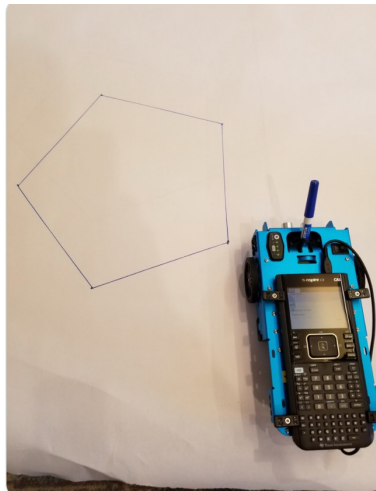
Drive a design



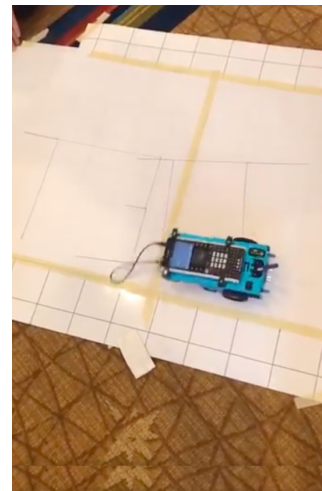
Draw artwork



Park your Rover



Use a For loop
to draw polygons



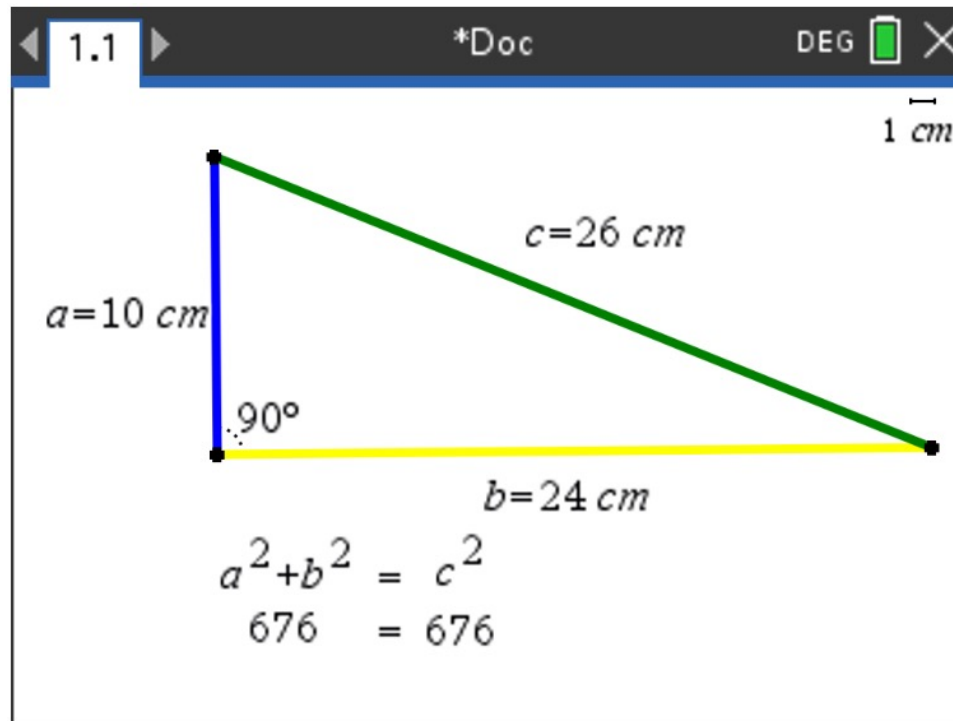
Write your name



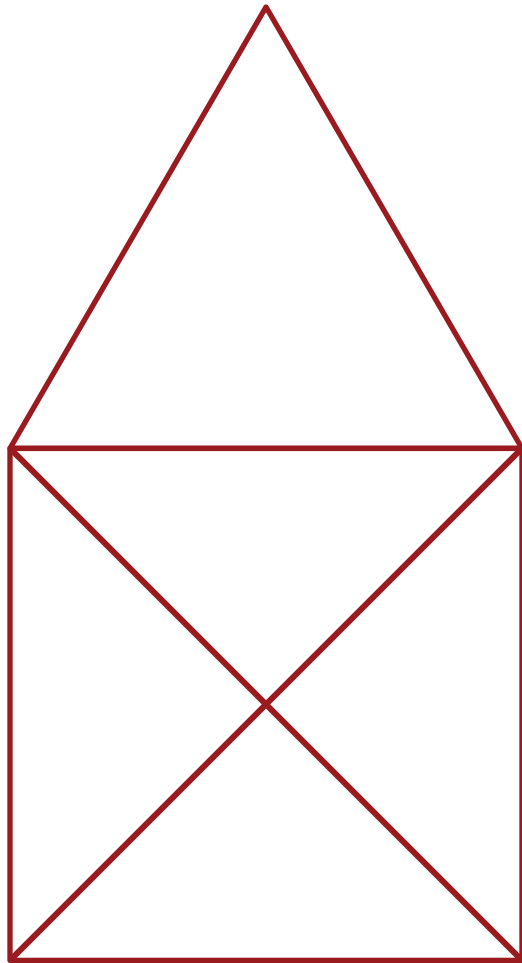
Navigate a map

Quick Math Reminders

» Pythagorean Theorem



Logic Challenge 2



Task: Drive the figure shown without crossing any lines or going back over a line and without picking up the pen.

When you are ready put the pen in and trace your path

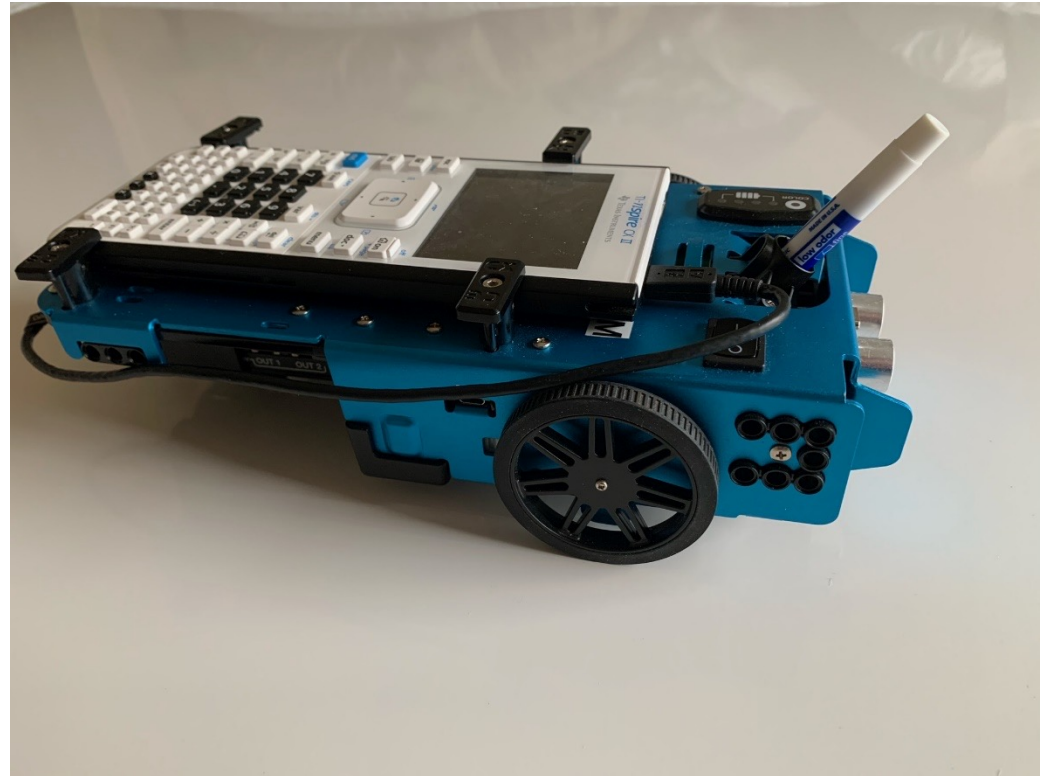
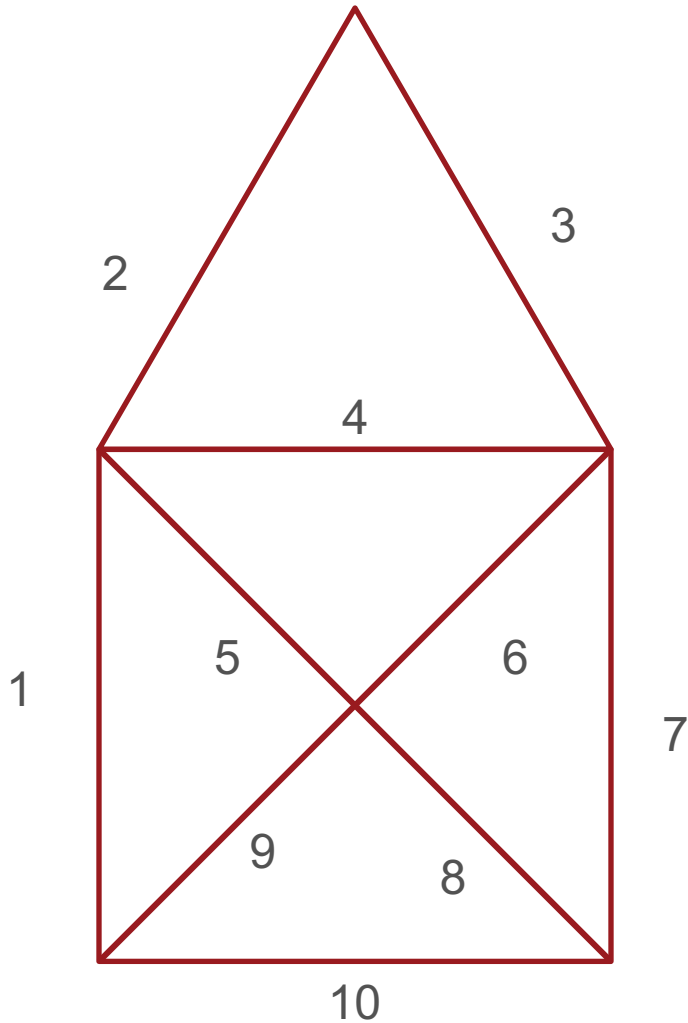
Import the Python Math module in addition to the Rover module for this challenge.

```
EDITOR: RVLOGIC2
Func Ctl Ops List Type I/O Modul
1: math...
2: random...
3: time...
4: ti_system...
5: ti_plotlib...
6: ti_hub...
7: ti_rover...
```

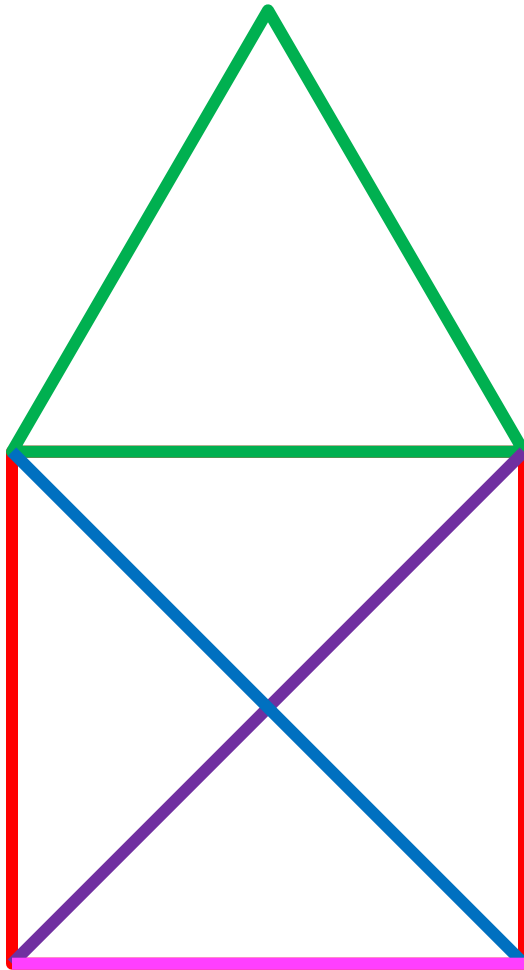
```
EDITOR: RVLOGIC2
math module
Math Const Trig
1: from math import *
2: fabs()
3: sqrt()
4: exp()
5: pow(x,y)
6: log(x,base)
7: fmod(x,y)
8: ceil()
9: floor()
0: truncate()
```

```
EDITOR: RVLOGIC2
PROGRAM LINE 0003
import ti_rover as rv
from math import *
-
```

Logic Challenge 2



Logic Challenge 3



Task: Drive the figure shown without crossing any lines or going back over a line and without picking up the pen.

Now match the colors using the RGB LED. Don't worry about using the pen.

Import the Python Math module in addition to the Rover module for this challenge.

Use `wait_until_done()` from the Rover Commands menu to synchronize Rover drive functions with the RGB LED.

```
EDITOR: RVLOGIC2
ti_rover module
Drive I/O Settings Commands
1:from ti_system import *
2:sleep(seconds)
3:disp_at(row,"text","align") >
4:disp_clr() clear text screen
5:disp_wait() [clear]
6:disp_cursor() 0=off 1=on
7:while not escape(): [clear]
8:wait_until_done()
9:while not path_done():
0:position(x,y)
```

```
1.1 1.2 *Doc RAD 11/16
*drive.py
import ti_rover as rv
from math import *
rv.color_rgb(0,255,0)
rv.forward(3)
rv.wait_until_done()
# wait_until_done holds the program
# at that location until the drive function
# before is completed
rv.color_rgb(255,0,0)
rv.forward(5)
```

Thank you!

See www.TIstemProjects.com for more TI STEM and coding activities and projects.