



Python Syntax Reference:

Statement	Example	Behavior
<code>import module_name as name_space</code>	<code>import ti_rover as rv</code>	Required for all TI Rover Python programs. Imports the ti_rover module into the Python program. The module provides the methods for controlling the Rover. Sets the current position of the RV as the origin and the heading as 0 degrees measured from the x-axis. The <code>import ti_rover as rv</code> statement is available from the Fns>Modul>ti_rover>Drive menu.
<code>from module_name import *</code>	<code>from ti_system import *</code>	Imports all the functions in the ti_system module for use in the program. It is necessary to import the ti_system module to use <code>disp_at()</code> , <code>sleep()</code> and while not escape() :. The <code>from ti_system import *</code> statement is available from the Fns>Modul>ti_rover>Commands menu.
<code>rv.forward(distance)</code>	<code>rv.forward(10)</code>	Rover drives 10 units forward. The default unit is 10 cm.
<code>rv.backward(distance)</code>	<code>rv.backward(10)</code>	Rover drives 10 units backward. The default unit is 10 cm.
<code>rv.stop()</code>	<code>rv.stop()</code>	Rover stops. This function is executed as soon as Rover receives it.
<code>rv.color_rgb(red,green,blue)</code>	<code>rv.color_rgb(255,0,0)</code>	Turns the color LED on with the color red. Values for the red, green and blue LED components are between 0 (off) and 255 (full power). The <code>rv.color_rgb()</code> function is available from the Fns>Modul>ti_rover>I/O>Outputs menu.
<code>sleep(seconds)</code>	<code>sleep(3)</code>	The calculator will wait 3 seconds before moving to the next line in the program. <code>sleep()</code> is available from the Fns>Modul>ti_rover>Commands menu.
<code>var=rv.waypoint_x()</code>	<code>x=rv.waypoint_x()</code>	Stores the current x-coordinate position in units of the TI-Rover into the variable x. The default unit is 10 cm. <code>rv.waypoint_x()</code> is available from the Fns>Modul>ti_rover>I/O>Path menu.

Statement	Example	Behavior
<code>disp_at(row,"text",align)</code>	<code>disp_at(3,"x position = "+str(x),"left")</code>	<p>The disp_at() function displays a text string on a specified row with an alignment of left, center or right. When variable x has a value of 7.6, the following is displayed on row 3, aligned to the left: x position = 7.6</p> <p><code>disp_at()</code> is available from the Fns>Modul>ti_rover>Commands menu.</p> <p>Note: The str() function converts a numeric value to a string. The + operator is used to join two strings. str() is available from the Fns>Type menu.</p>
while not escape(): block	while not escape(): <code>x=rv.waypoint_x()</code> <code>disp_at(3,"x position ="+str(x),"left")</code>	<p>Defines a while loop that will continue until the [clear] key is pressed.</p> <p>While loops repeat the statements in the block if the condition at the top of the loop is true. In the example, looping continues until the [clear] key is pressed to escape the loop. Not pressing a key or pressing any key but escape means that "not escape()" will return "true". True for the loop condition means that looping continues. If [clear] is pressed "not escape()" will return "false". False for the loop condition means that looping stops. Program execution skips to the statement just after the loop. Note: The block starts with a colon and includes the indented lines that follow. while not escape(): is available from the Fns>Modul>ti_rover>Commands menu.</p>
<Boolean expression> value 1 operator value 2	<code>2+3==6</code> (result is false) <code>x+4>=y</code> (if x=1 and y=3, the result is true) <code>"enter"!="esc"</code> (result is true)	<p>Boolean expressions evaluate to either true or false.</p> <p>Note: <code>==</code> is the Python operator to check equality. <code>>=</code> is the Python operator to check whether the value to the left is greater than or equal to the value on the right. <code>!=</code> is the Python operator to check inequality. Boolean operators are available from the Fns>Ops menu or from the [2nd] [test] menu on the keyboard.</p>
if <Boolean expression>: block	if <code>0<x<2:</code> <code>rv.color.rgb(255,0,0)</code>	<p>Checks to determine if the value of variable x is between 0 and 2. If the statement is "true" then the statements in the if block are executed. Otherwise, the block is skipped. In the example, when the value for the variable x is between 0 and 2, the calculator will send a command to the TI-Innovator to set the color rgb LED to be red.</p>
if <Boolean expression> and <Boolean expression>: block	If <code>x>=2 and x<4:</code> <code>rv.color.rgb(0,255,0)</code>	<p>If both expressions are true the and function is "true", then the block is executed. Otherwise, the and function returns false, and the block is skipped. In the example, when the value for x is greater than or equal to 2 and less than 4, the calculator will send a command to the TI-Innovator to set the color rgb LED to be green. and, or and not are available from the Fns>Ops menu or from the [2nd] [test] menu on the keyboard.</p>

See the Rover module section beginning on page 26 of the [Python Programming for the TI-84 Plus CE Python Graphing Calculator Guidebook](#) for more programming information.