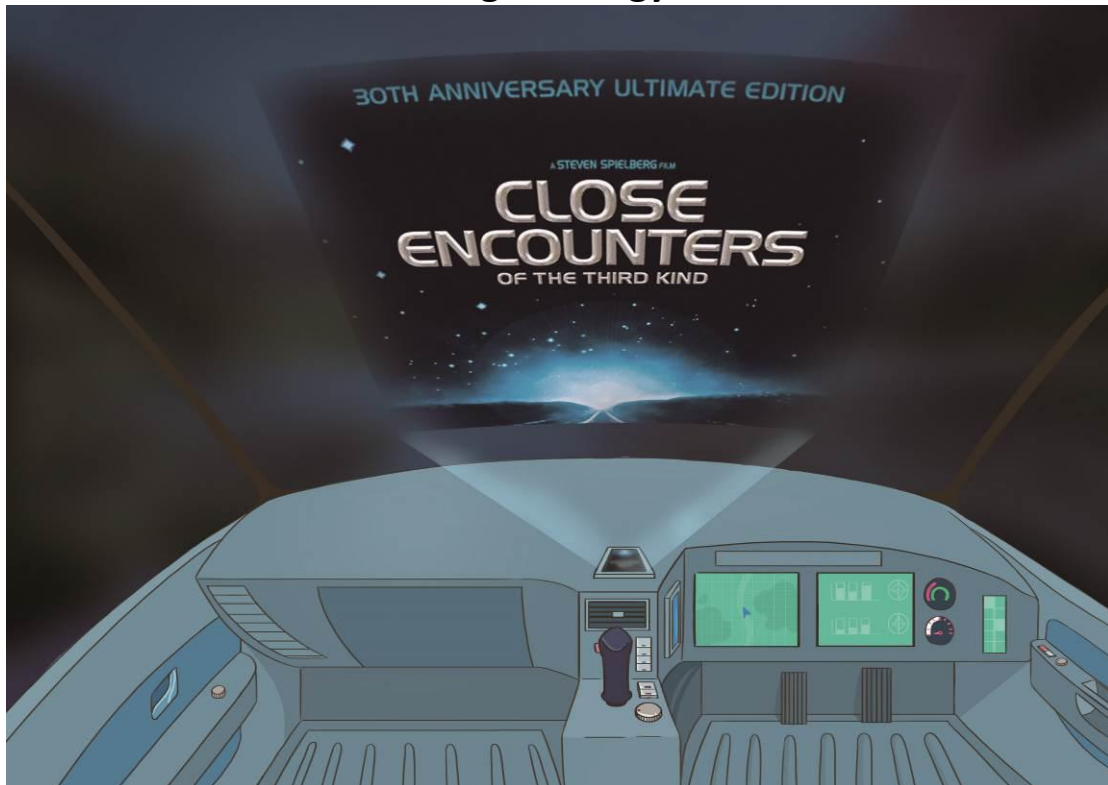


## Light Energy



Two weeks ago, Mark's Velociraptor broke down due to overloading of the water cooling system of the energy converter, resulting in full evaporation of Rn119 solid catalyst which is used to convert water into hydrogen. The photochemical molecular converter which Mark is proud of is capable of producing a large amount of Rn119 catalyst with only a little sunshine, so Velociraptor's catalyst storage chamber is designed as dainty and small considering Mark's demanding requirement for car appearance. Because of energy loss and gradual loss of speed, the car pulled to the side of the road and the sound of the overload alarm woke Mark up (because Mark rarely drives himself). He stretched his body and looked at the time. "What! It's already half past four in the morning!" He opened and checked the power compartment and restarted the cool down process on the overloaded water cooling system.

"Anyway, it'll brighten up soon. Let's wait here until the morning light." Mark said to Rex, who is sitting next to him. "Raptor uses a hydrogen fuel power system. Why do we still have to wait for the sunlight, like with a solar powered car?" Rex quizzically asked. Mark laughed without responding. He picked up his phone after getting back in the car and projected a 3D image onto the windshield. "Help me pick a Sci-Fi film." Mark activated voice command.

"Steven Spielberg's 1977 film: Close Encounters of the Third Kind. Shall I play it for you?" the system responded. "Ok!" Mark answered.

## Learning Objectives

It is important to know how to make and use variables properly during programming. With the help of variables, it is possible to reduce unnecessary readings of sensor and make advanced control over the system using the content of the variables.

In this chapter, we will use variables to store the data of light sensor and interact with our cute panda and the magical mBot Ranger robot via the data.

## Scientific knowledge

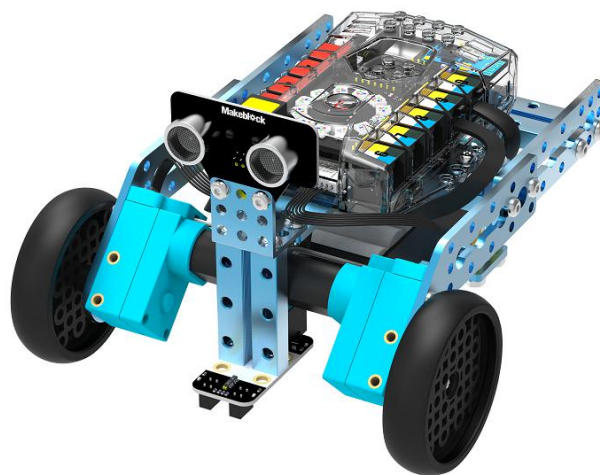
Semiconductor is a kind of conducting material whose conductivity is in between that of insulator and that of a conductor and is controllable. Common semiconductor materials include silicon, germanium, gallium arsenide, etc.

Photoresistor is a kind of semiconductor whose conductive characteristics can change with light intensity. When the material is stimulated by the energy from light, it will produce more conductive free electrons to reduce the resistance and allow more current to flow through. Therefore, it is used as an electronic component to convert optical signal to electrical signal.



## Preparation for Assembly

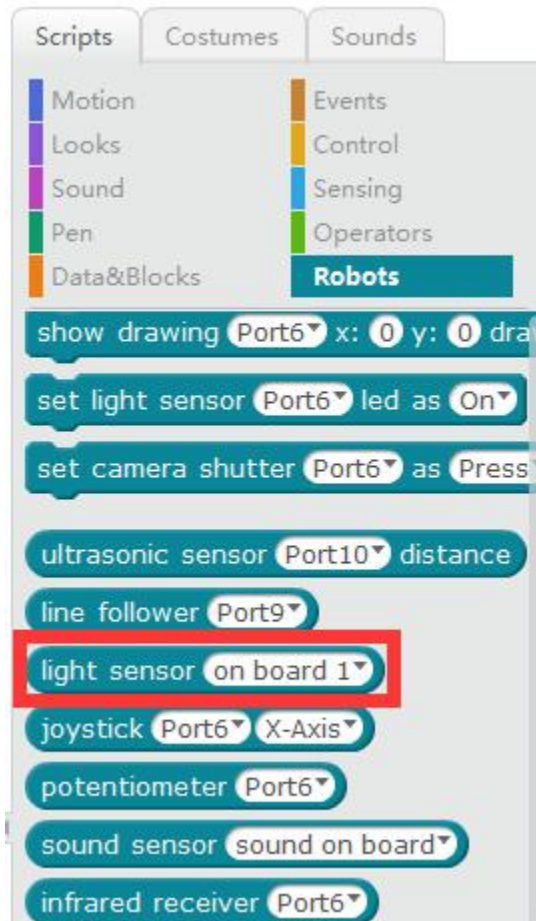
We will use the three-wheeled racing robot in this chapter.





## Learning Tasks

### Learning Task 1 - The Value of Light Sensor

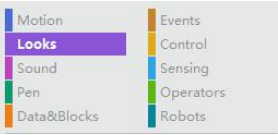

The task is to display the values of light sensor. This task can be done with the light sensor block in mBlock. Drag [light sensor...] block from [Robots], as shown in the picture below:



		Light sensor block
<p>This block is to read the values of light sensor, and it ranges from 0 to 970.</p> <p>(Onboard): Options to select corresponding light sensors. The default value [Onboard 1] corresponds to the light sensor 1 on Me Auriga, while [Onboard 2] corresponds to light sensor 2.</p>		

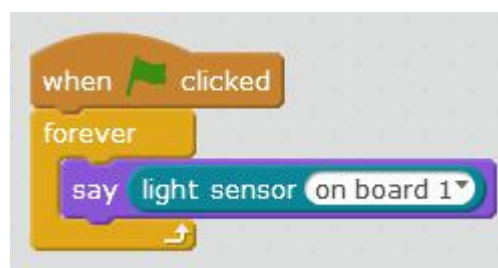
Click the inverted triangle on the block, then select the light sensor that you want to read in the drop-down list. Please write a program to make the panda speak out the

value of the light sensor. Panda is the sprite of mBlock. Here we use [Say...] block under [Looks], as shown below:

		Say [Hello]
<p>Function: blocks for the sprite in mBlock. A dialog box will pop up next to the sprite to display the text in the white box.</p> <p>The default text is [Hello]. We can input text into the white box directly.</p>		



In order to let the panda “say” the current value of light sensor, we need to write a program as below:



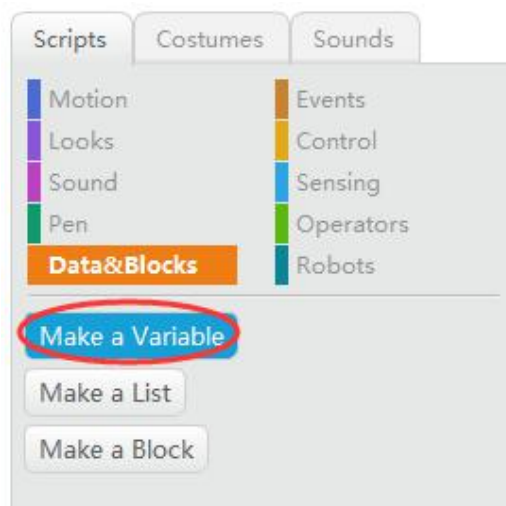
Connect the Ranger with mBlock, then click the green flag. Observe the value of light sensor said by the panda. Try to cover light sensor 1 on Ranger with your hands, then observe whether the value becomes smaller. The lower the light intensity, the smaller the value of light sensor we get.



## Learning Task 2 - Description & Use of Variables

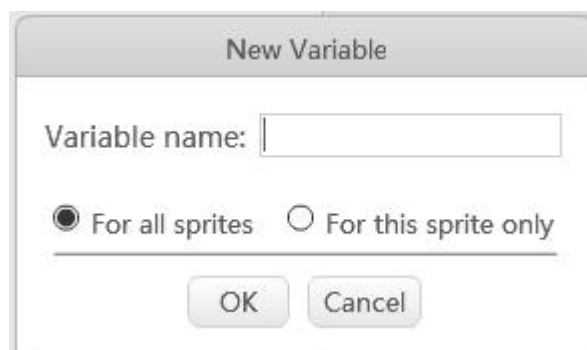
Variable is used to store numbers or texts, just like a drawer or a cabinet. We can set the contents of variables or read the contents that variables store.

Click [Data & Blocks] --> [Make a Variable], as shown below:

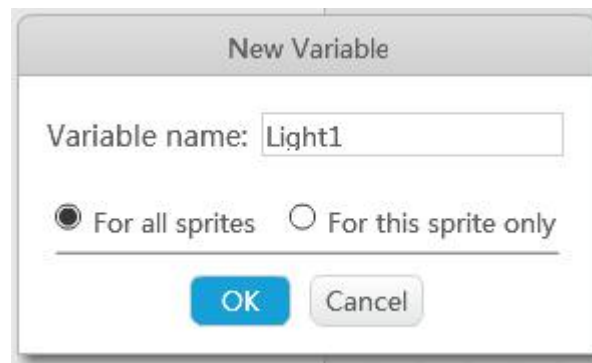


When a small window called [New Variable] pops up (as below), input a new name for the variable.

The name must begin with a letter and can only contains letters and numbers.



When naming variables, please use meaningful letters and numbers, which is convenient to read in the future and understand the definition of the variables. Here we set the name as [Light1], which stands for the value of light sensor.



The default value of the variable is [For all sprites], hence every sprite in mBlock can set the content of Light1.

Next, click [Ok] button, and the blocks related to this variable will appear in block area, as below:

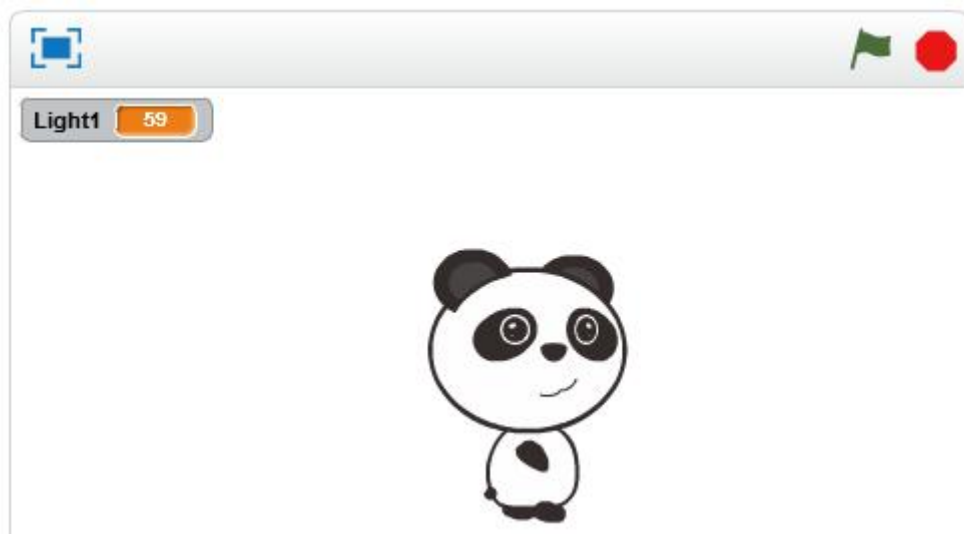


In these blocks, [set light1 to 0] is the block that you can set the content of this variable.

Please write the program as follows:



Connect Ranger with mBlock, then click the green flag. The values of light sensor 1 will show in this variable, as below:



If you cover light sensor 1 with your hands, you will find that the value of Light1 at the upper left corner of stage is changing constantly. The closer your hand is to the light sensor, the less light the sensor detects, hence the value will become smaller.

## Task

### Task 1 - Change the Size of the Panda

The light sensor is used to detect the light intensity around Ranger. With this feature, we can make the panda on the stage change with light brightness.

When we need to check conditions, it can be done with the [If...then...else] block. This block is capable of realizing the selection of [Yes or No] ([Right or Wrong]) according to conditions.

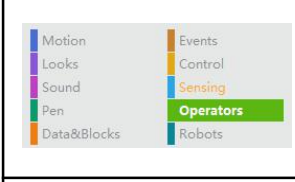

For example, if [Hungry], then [Dine], otherwise [Sleep].

It means: 1. the panda will have a meal when it is hungry. 2. the panda will sleep when it is not hungry.

		<p>If...then...else</p>
<p>Function: This block will check the condition in the hexagonal space. If the condition is true, the code after [then] will activate; if the condition is false, the code after [else] will activate.</p>		

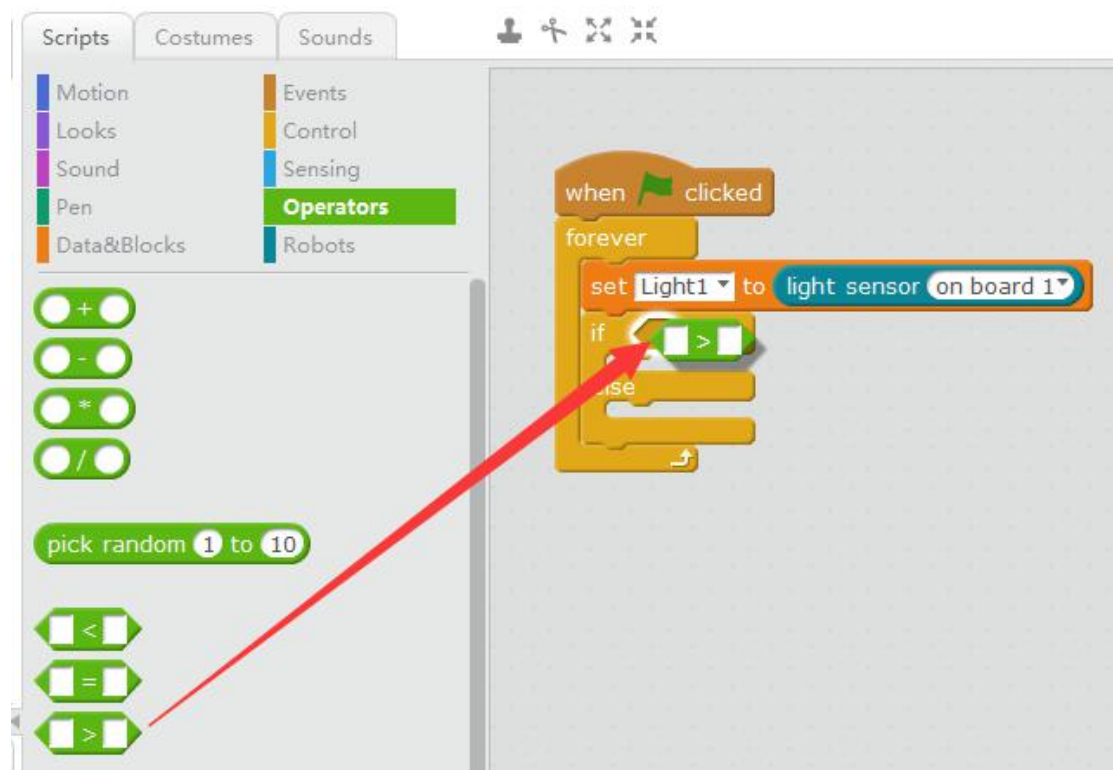
Block [If...then...else] can be used to judge based on the conditions. Conditions can be put in the hexagonal space. This hexagonal block is also called Boolean Block.

After reading the values of light sensors, we can compare these values to achieve certain effects. There are some hexagonal logical blocks under [Operators].

		[greater than] comparative block
Function: Hexagonal condition block. If the condition is true, output [Yes] (True); if the condition is false, output [No] (False).		

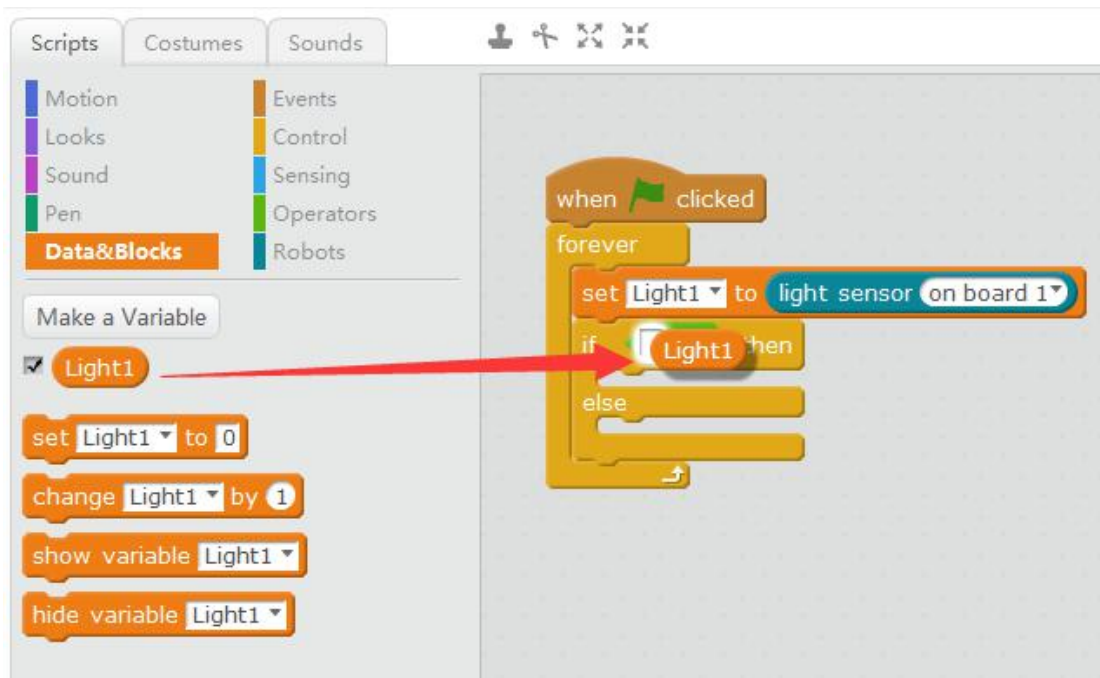
We will modify the program in Learning Task 2 with some comparative blocks. Add a [[If...then...else] block into the program, then drag a [greater than] block into the hexagonal space of the [If...then...else] block.

How to drag: drag the [greater than] block near the hexagonal space. When there is a white halo around the hexagonal space, release your mouse, then the [greater than] block will insert into the space correctly.



We have already set “Light1” as the values of light sensor 1 in previous task, hence we can use this variable directly in the following task. Drag [Light1] into the first space of [greater than] block (there will be a white halo around the first space), as shown in the picture below:

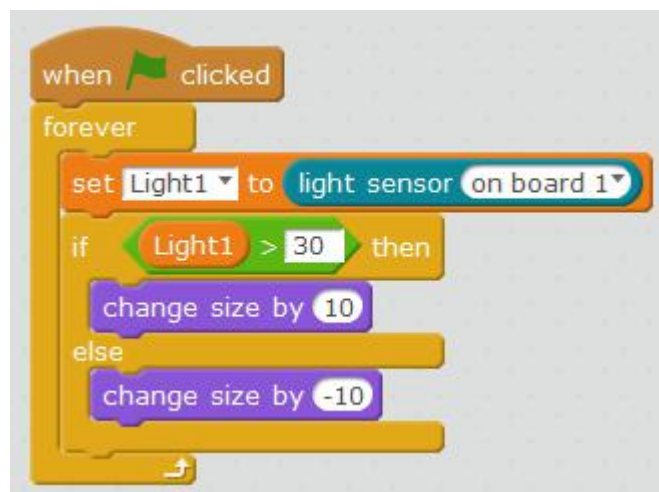




Input "30" in the second space of [greater than] block, then click [Looks] --> [change size by...] block.

		<p>Change the size of the sprite (unit: percentage)</p>
<p>Function: enlarge or shrink the size of the sprite</p> <p>Positive numbers represents "enlarge", while negative number stands for "shrink". The default value of this block is 100, which means 100% (the original size of the sprite)</p>		

The whole program is as follows:

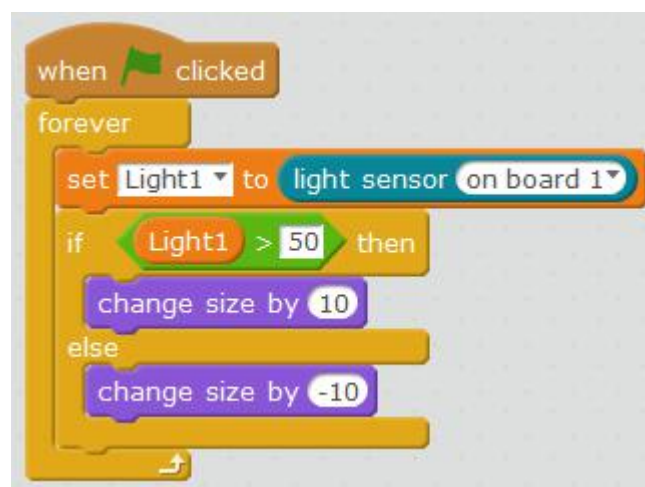


Click the green flag to implement your program. Put your hands over light sensor to cover the light, what will happen to the Panda? Then raise your hands higher, does the size of your panda change?

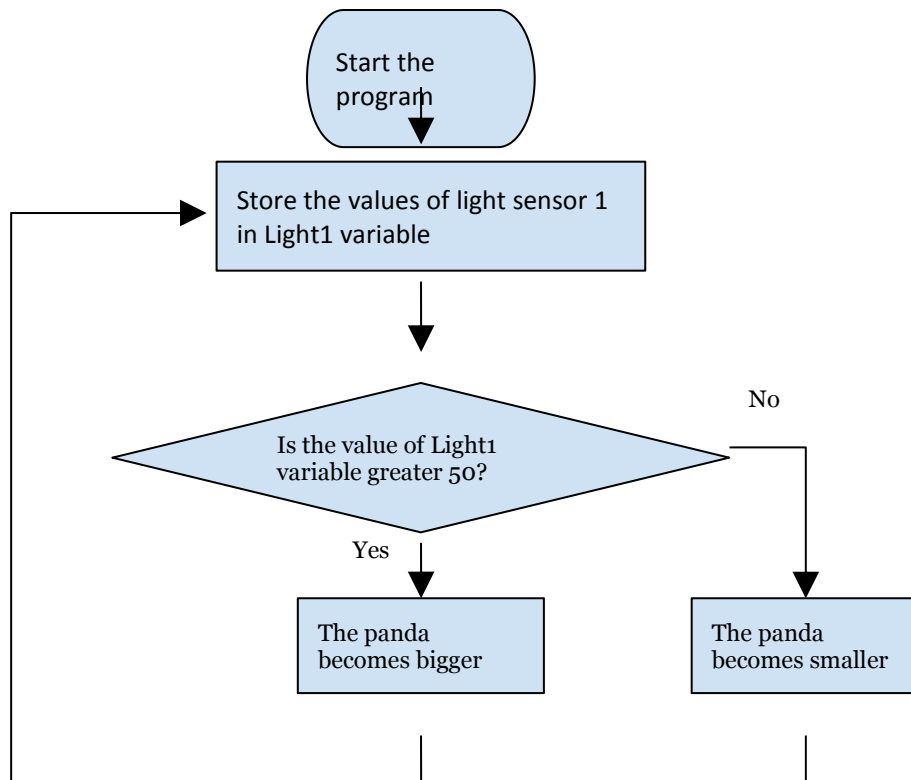
If your panda has no change, it indicates that the comparison of “greater than 30” is not correct, hence we need to adjust the comparison value (30). In general, we should record the value of the light sensor when your hands cover the light sensor (darker) and the value of the light sensor when move your hands away (brighter). Then take the intermediate value between these two values, that is, add these two values and divide the sum by two.

For example:

Value of light sensor (darker)	Value of light sensor (brighter)	[Value of light sensor (darker)+ Value of light sensor (brighter)]÷2
15	85	$(15+85)\div 2=50$



The flowchart of the program is as follows:

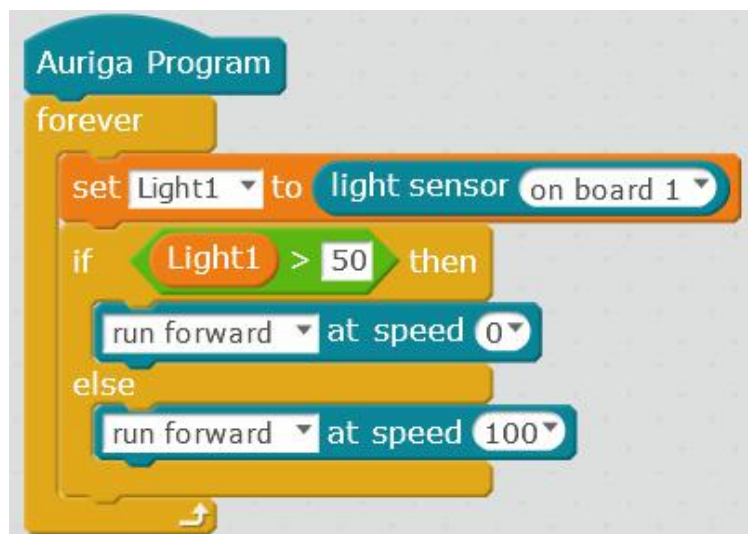


## Task 2 - Ranger Going Forward (stop when bright and move when dark)

In addition to changing the panda on stage, the light sensor also can be used to control Ranger.

The principle is the same as task 1, but Ranger have to execute this task on its own.

Off-line control: replace the hat block [when green flag clicked] at the beginning of program with [Auriga Program], then replace [change size by...] block with [run forward at speed...] block, as follows:



Click [Auriga Program] block → switch mBlock to Arduino mode → connect Ranger with mBlock via USB cable → click [Upload to Arduino] → wait for compilation and uploading.

Disconnect Ranger from mBlock, then cover light sensor 1 of Ranger with your hands. Does your Ranger move?

### Challenging Task 1 - The Solar Vehicle

This task is to refit Ranger into a solar vehicle: when illuminating Ranger with a flashlight, Ranger will run forward quickly for one second, then stop. Ranger will repeat the same action continuously in this task.

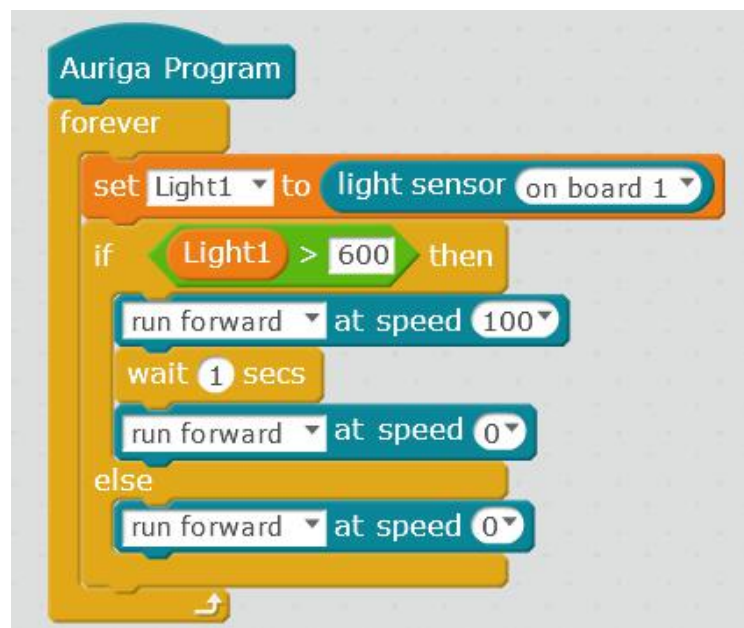
The key point of this task is that Ranger will not move until receiving the light coming out of the flashlight, hence this task can be done [If...then...else] block.

Measure the values of light sensor, calculate the intermediate value, then use it as a comparative value in the [greater than] block.

In this task, Ranger should run forward quickly for one second then stop, so we can use [wait 1 secs] block.

Think it over at first, have a try to write a program by yourself, then check the results with Ranger.

**Answer:**



You can also modify your program to control Ranger to do various interesting movements (e.g. swinging left and right) with the flashlight.

## Conclusion of This Chapter

By changing light intensity, it is possible to change the moving direction of Ranger and interact with the sprite on mBlock interface. Maybe you can also take advantage of light intensity to turn on or turn off LED light to conserve energy. Do you have more ideas?



"Rex, you've done really well. I didn't realize you could record everything I do in a day." Mark jumped out of his chair in excitement, pacing back and forth in the laboratory, thinking it over.

"No wonder I have an impression, it's a movie that's been played thousands of times. Could it be that they are imitating the situation in the movie and sending a friendly a message to Earth?"

"How did they decode the signals in this movie?"

"Don't they have another form or means of expression?"

With such strange ideas coming to him, it seems Mark won't be sleeping tonight!