

Getting Started with InO-Bot and Logo

Terrapin Logo and InO-Bot are the perfect combination for exploring coding and robotics. This guide will get you started. Then you can continue to discover all that InO-Bot can do on your own.



Contents

1. First Steps
2. Meet InO-Bot
3. Launching the Logo Program
4. Pairing InO-Bot to Logo via Bluetooth
5. Entering Logo Commands
6. List of Logo Commands to Try
7. Logo Challenges
8. Checking the Battery Status
9. Playing Sounds
10. Programming the Lights
11. Programming the Sensors
12. Calibrating InO-Bot
13. How to Learn to Program in Logo: A List of Resources
14. InO-Bot Specifications, Charging Information, Care & Maintenance

1. First Steps

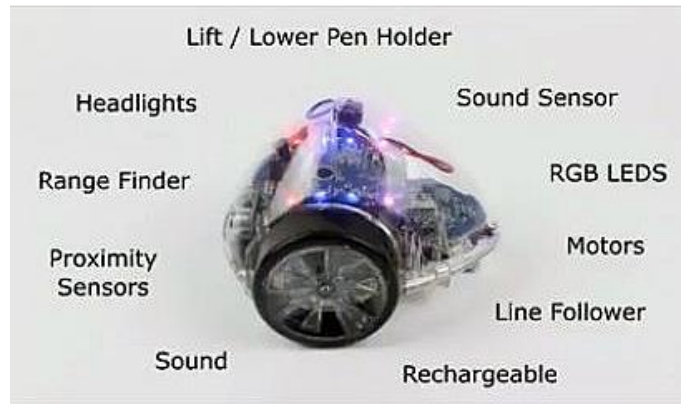
- a. Take InO-Bot out of its box. Store InO-Bot in the box when not in use.
- b. Make sure that InO-Bot's power and sound switches are turned off when charging. The switches are on the underside of InO-Bot.
- c. Plug the square end of the charging cable into InO-Bot. The charging port is on its underside.
- d. Plug the USB end of the charging cable into a USB port on any device. InO-Bot will shine a red light.
- e. The InO-Bot will charge for about four hours. When it is fully charged, the red light will go out.

For more information about charging, see Section 14.

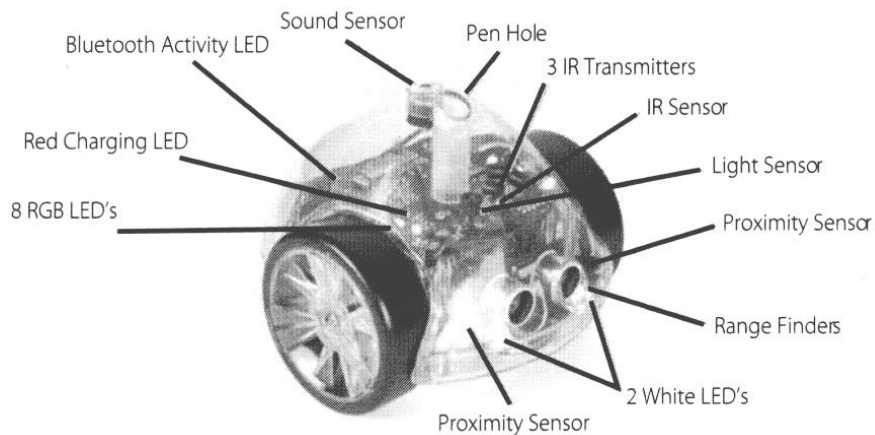
2. Meet InO-Bot

While you wait for InO-Bot to finish charging, let's look at some of its features.

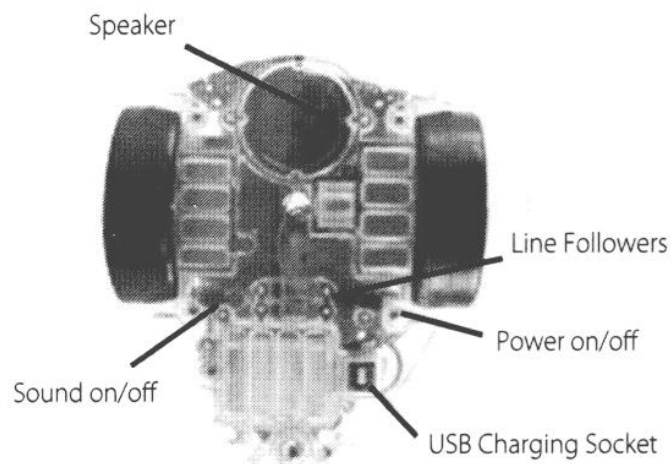
Why is InO-Bot named InO-Bot? The **I** is for Input and the **O** is for Output. InO-Bot can receive information (Input) and also report conditions (Output). In these activities to try, you will see many examples of both.



Below are detailed views and locations of its features:



Top side of InO-Bot



Underneath InO-Bot

What can InO-Bot do? It can:

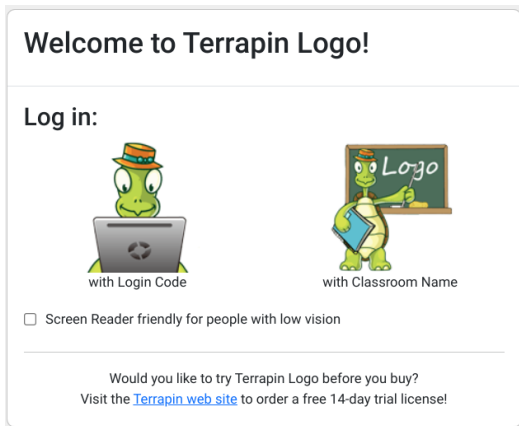
- follow commands you enter in Logo
- raise and lower a pen positioned in the hole in its center so it can draw
- detect an obstacle
- play sounds and music
- turn its headlights on and off independently
- light 8 colorful lights
- detect sound
- carry a sports camera and take videos of its travels
- offer endless fun and challenging programming opportunities



Let's get started!

3. Launching the Logo Program

You may have installed the Logo program on your computer, or you may be using the Web version of Logo in a browser tab. (Be sure you are using the Chrome browser!) In this guide, we'll use WebLogo.

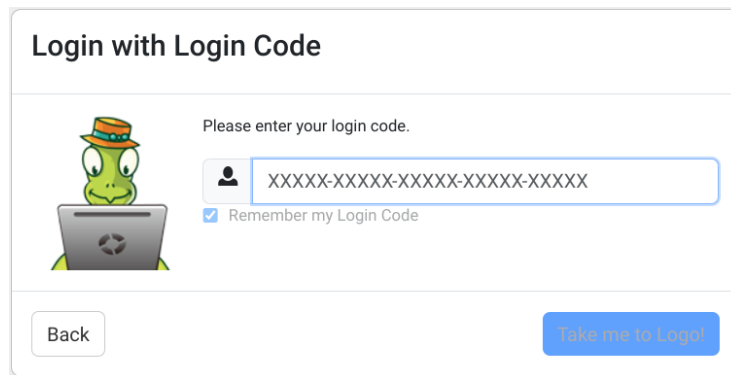


To start up Logo, go to weblogo.terrapinlogo.com or launch the desktop version.

You will see the opening screen shown at the left.

The icon on the left is for a Household license. The one on the right is for a Multi-User License.

Click one of the Login options, depending on your version.

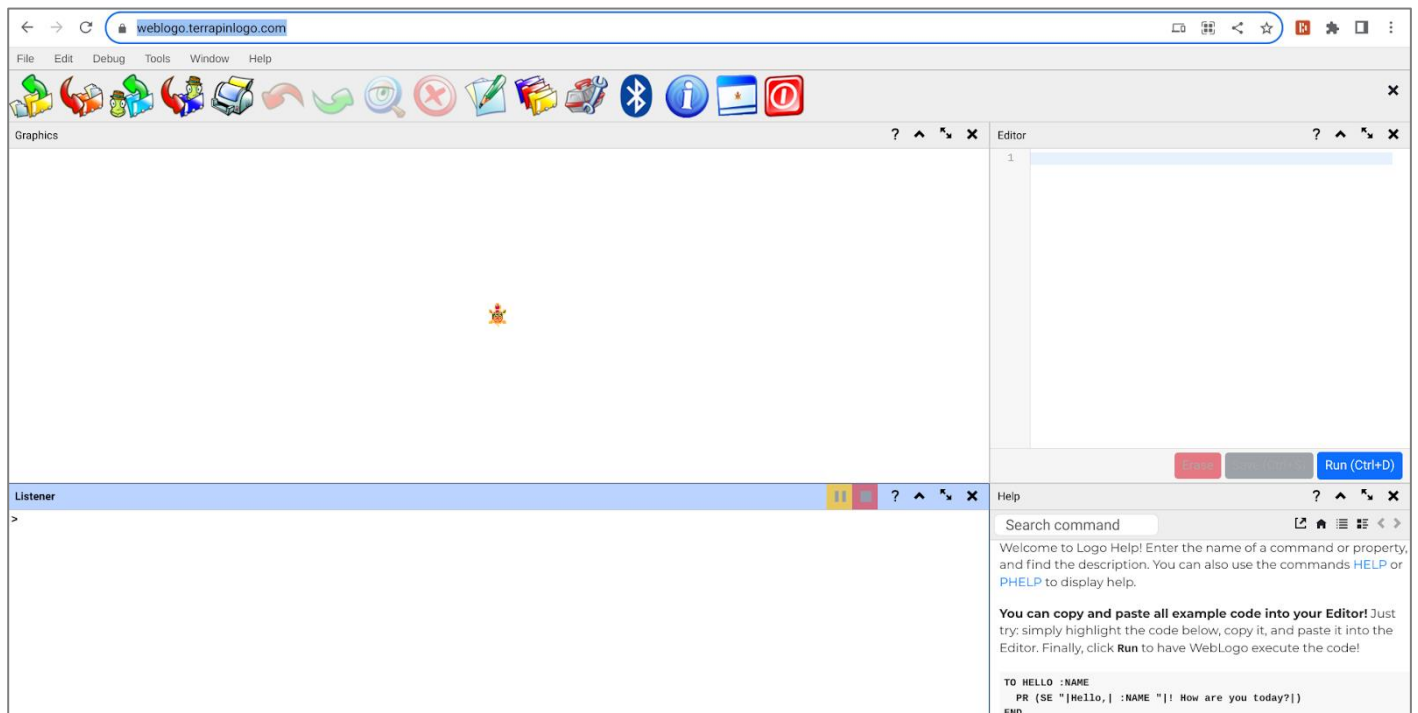


Enter your login code and click this button:

Take me to Logo!

Tip: It's easier to log on the next time if you check the 'Remember my Login Code' box.

You will see a screen that looks like this with a colorful turtle in the center of the Graphics window.



Note that you can set up your Logo windows any way you like.

4. Pairing InO-Bot to Logo

Before InO-Bot can run your Logo commands, it needs to connect to Logo using Bluetooth.

>> **Note: Only the Chrome web browser supports Bluetooth.**

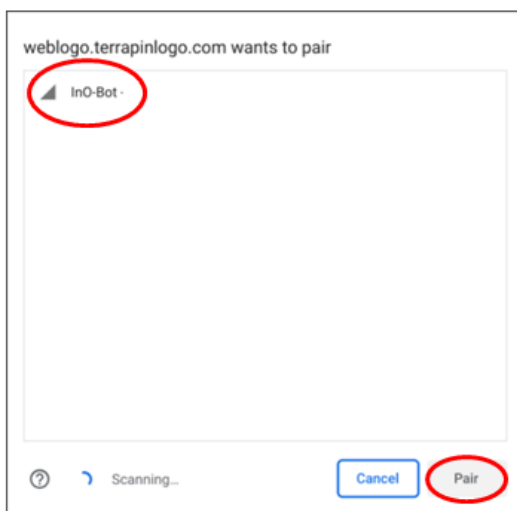
You will need to run Weblogo in a Chrome browser to control InO-Bot.

First, turn on InO-Bot. The On/Off switch is located on the underside of InO-Bot. Slide the switch to the On position. A small blue light will blink on InO-Bot.

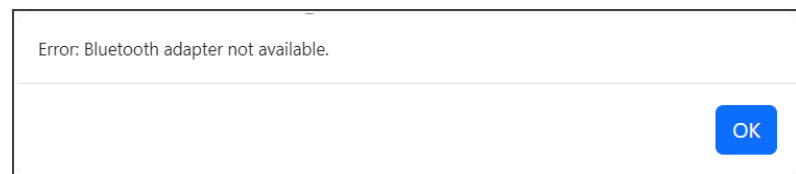
Click the Bluetooth icon in the graphical menu at the top of the Logo screen. It looks like this:



After you click the Bluetooth icon, you will see the screen shown below. Bluetooth will scan for available devices and find InO-Bot. Click the word InO-Bot and then click Pair. Doing this connects InO-Bot to Logo.



If you are using an older computer that does not support Bluetooth, you will see this message:



If you see the message **No compatible devices found.**, InO-Bot's power switch is probably turned off. Turn the Power switch to On and try again.

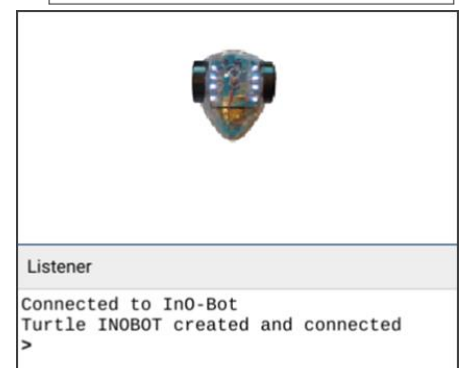
If you stop using InO-Bot for a while, it may become unpaired. Click the Bluetooth icon again. Even though it says that InO-Bot is paired, click its name and then Pair to reconnect it.

Is InO-Bot connected? To find out, type this: `INOBOT?`
The result will be either `TRUE` or `FALSE`.

When InO-Bot connects to Logo via Bluetooth, several things happen:

- InO-Bot plays a brief sound to confirm it is connected,
- The turtle in the Graphics window changes to an image of InO-Bot.
- A message in the Listener window tells you that InO-Bot is connected.
- If you type `WHO`, you see that the turtle's name is `INOBOT`.
- The browser tab that Logo is in shows the Bluetooth icon.

You're now ready to start giving InO-Bot commands from Logo.



5. Entering Logo Commands

Enter some Logo commands and watch what InO-Bot does!

Try It!

Try a simple command like this: `FD 25`

InO-Bot should move forward 25 centimeters, or about 10 inches.
As InO-Bot moves on the floor, the InO-Bot icon in Logo's Turtle window moves at the same time.

Did InO-Bot move as you expected it to?

If it didn't move, the reason might be that InO-Bot has been sitting in a closet for a long time, unused. Manually turn the wheels back and forth a few times and try again. Did it work this time?

How does InO-Bot move and turn?

InO-Bot moves in units of centimeters (cm). The command `FORWARD 10` causes InO-Bot to move forward 10 cm. The maximum distance that InO-Bot can move is 127 cm. If you need InO-Bot to move farther, you can send it multiple forward commands.

InO-Bot turns in degrees. Each unit is a single degree, so the command `RIGHT 20` turns InO-Bot right 20 degrees. The maximum turn is 180 degrees in either direction. If you want to turn InO-Bot more than 180 degrees, enter an additional turn command.

How to use InO-Bot's pen to draw

You can add code to your program to raise and lower the InO-Bot's pen.
First, put a pen or thin marker into the hole in the middle of InO-Bot.
You may find that you need to press it down quite firmly so that it extends to the bottom of InO-Bot.

Try It!

Type `PU` to raise the pen. Then move InO-Bot.

Type `PD` to lower the pen. Then move InO-Bot.

Figure it out

How many degrees does InO-Bot turn to complete a circle?
Can you program InO-Bot to draw circles of different sizes?

If InO-Bot doesn't come very close to getting back to where it started, you can try calibrating it. See Section 12 for information on how to do this.

If you are not familiar with Logo or would like a handy list of Logo commands to use with InO-Bot, see the next section.



Terrapin's [Dry Erase Mat](#) comes with markers that work well with InO-Bot. It offers a good surface for drawing designs.

6. Logo Commands to Try

The following is a brief list of Logo drawing commands. It is not a complete list of commands, but includes those that you are likely to use most often with InO-Bot. The input, when listed, is what type of information needs to follow the command for Logo and InO-Bot to understand and execute it.

Try these commands!

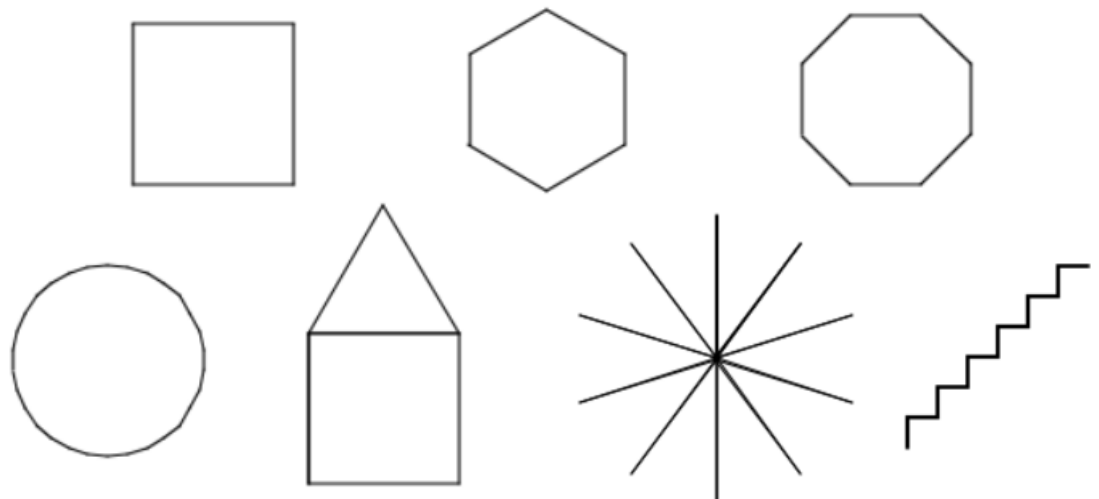
| Command | Input | What InO-Bot Does |
|-----------------------|--------------------------------------|--|
| FORWARD or FD | <i>number</i> (from 1–127) | moves InO-Bot forward that number of centimeters |
| BACK or BK | <i>number</i> (from 1–127) | moves InO-Bot backward that number of centimeters |
| RIGHT or RT | <i>number</i> (from 1–180) | turns InO-Bot that many degrees to the right (clockwise) |
| LEFT or LT | <i>number</i> (from 1–180) | turns InO-Bot that many degrees to the left (counterclockwise) |
| SETHEADING or SETH | <i>number</i> (from 1–360) | turns InO-Bot to point to the specified heading, in degrees |
| HEADING | | reports the heading of InO-Bot in degrees |
| REPEAT | <i>number</i> [<i>commandlist</i>] | runs the list of commands as many times as <i>number</i> |
| DISTANCE | | reports the distance from the InO-Bot to a coordinate location |
| XCOR or GETX | | reports the X-coordinate of InO-Bot |
| YCOR or GETY | | reports the Y-coordinate of InO-Bot |
| POS or GETXY | | reports the X- and Y-coordinates of InO-Bot |
| PENUP or PU | | raises InO-Bot's pen |
| PENDOWN or PD | | lowers InO-Bot's pen |
| SETPOS or SETXY | | allows you to set the X- and Y-coordinates of InO-Bot |
| HOME | | sends InO-Bot to its home position, equivalent to SETXY [0 0] |
| PDIST | | reports the distance from InO-Bot to its home |
| TOWARDS | [<i>xcor ycor</i>] | reports the angle InO-Bot should turn to aim at that coordinate position |
| WAIT | <i>number</i> | inserts a pause before the next instruction runs; 1000 is one second |

Be careful not to enter the DRAW command. This changes the drawing icon back to the onscreen turtle and disconnects you from Bluetooth. You would then need to turn InO-Bot off and on and pair it again.

Figure it out!

Can you use REPEAT to:

- draw a square
- draw a hexagon
- draw an octagon
- draw a circle
- draw a house
- draw a sunburst
- draw a staircase



Try the challenges in the next section. Also, visit the [Turtle Tuesdays webpage](#) for more projects and their solutions!

7. Logo Challenges

Draw Some Letters

Try drawing some letters of the alphabet. Some are easy to draw, and some present more of a challenge. You could even draw the outline of a letter.



Draw Some Stars



Try Designs with Nested Repeats (one REPEAT instruction inside another one)

What is the basic shape used in each of these designs?



Logo and InO-Bot Tips

As you try these challenges or, better yet, think up your own projects to try, keep these tips in mind:

- Create your drawing first in Logo before pairing InO-Bot. That will help you perfect your project.
- When you are ready for InO-Bot to draw it, pair it to Logo and run (and debug) your instructions.
- Keep a record of the Logo commands and procedures you have created. (Save your Logo files!)

8. Checking the Battery Status

Logo uses Property Lists for many InO-Bot functions.

One handy property list command reports InO-Bot's battery status in volts.

Try It!

Type this command and check the result.

```
GPROP "INOBOT "BATTERY  
Result: 3.7
```

Note: If all 8 lights start to flash red, your InO-Bot needs recharging! See Section 14 for more details.

9. Playing Sounds

InO-Bot can play 43 different sounds. These include sound effects and notes that sound like a piano, xylophone, or harp.

You can use the `PLAY.NOTES` and `PLAY.SOUND` commands.

Try these commands!

```
PLAY.NOTES [C D E F G A B O5 C] ; O5 sets the notes to play in a higher octave  
PLAY.SOUND 1  
FOR "I 1 10 [PLAY.SOUND :I WAIT 300]
```

This last instruction plays all the sounds from 1 to 10 in turn, pausing between them.

Piano sounds go from 11 to 17.

Xylophone sounds go from 18 to 30.

Marimba sounds go from 31 to 43.

You can create commands for these ranges of sounds?

Can you play a scale (the white notes on a piano keyboard starting with C)?

Can you get InO-Bot to play a simple tune?

Try playing "Twinkle, Twinkle, Little Star." All you need are notes 31, 32, 33, 34, 35, and 36.

For more information about making sounds with Logo, [visit this webpage](#).

10. Programming the Lights

InO-Bot has eight LEDs (Light Emitting Diodes). The first input is either a number from 1 to 8 or a list containing the numbers of the LEDs to set. The second input is a color name or a color value. The third optional input is the duration in milliseconds that each LED should stay turned on. Not all LEDs are turned on at the same time, but rather one after another, staying on for the given time. This allows for interesting light effects.

Try It!

```
LED 1 "RED
```

What happened when you entered that command?
Which light turned on? What color is it?

Figure it out!

How are the LEDs numbered? Which one is #1, #2, #3, and so on?
Knowing this will help you write your programs that control the lights.

What color names work? Which ones don't?
Can you think of how you might turn the lights off? (Hint: See the next example.)

Try these!

Set LEDs 2, 3, and 4 to gold for half a second each.

```
LED [2 3 4] "GOLD 500)
```

Turn all LEDs off.

```
LED [1 2 3 4 5 6 7 8] "BLACK
```

Try these!

InO-Bot also has headlights that you can turn on and off.

```
PPROP "INOBOT "LIGHTS [1 1]    turns both headlights on
PPROP "INOBOT "LIGHTS [1 0]    turns one on and the other off
PPROP "INOBOT "LIGHTS [0 1]    and vice versa
PPROP "INOBOT "LIGHTS [0 0]    turns both headlights off
```

Can you figure out a way to make the headlights blink on and off?

You could add a `WAIT` command between them.

Try this just for fun!

```
REPEAT 50 [LED (RANDOM 8) PICK ["BLUE "WHITE "RED "PINK "ORANGE "YELLOW
  "PURPLE] PPROP "INOBOT "LIGHTS LIST ((RANDOM 2) - 1 (RANDOM 2) - 1]
```

11. Programming the Sensors

InO-Bot has a light sensor, a sound sensor, and 4 proximity sensors. It also has line sensors, which you can learn more about in the next section.

Light Sensor

The Light Sensor is located in front of the Pen Holder and measures the amount of ambient light. You can write programs that have InO-Bot react based on a desired light threshold. The Light Sensor value range is 0–1.0, with 1.0 being the brightest.

To check the level of light, use this property:

```
GPROP "INOBOT "LIGHT
```

The result will be a value between 0 and 1.

How could you use this value?

Try this!

Can you write a program in Logo that will turn on the headlights when the room is dark? You would have to continually check the light sensors, and issue a headlight command when the light dims.

Check the light sensor when the room is bright, and when it is dim.

Using Logo's `WHEN` command makes this easy.

```
WHEN [name propertyname compare-to-value] [runlist]
```

Use `INOBOT` for `name`. Use `LIGHT` for `propertyname`.

For `compare-to-value`, figure out the value the sensor needs to report. It is better to use `<` than a specific value.

For `runlist`, enter the instructions to turn on the headlights. See Section 10 for how to do this.

Can you write the opposite instruction, to turn the lights off when the room gets brighter?

Did you figure these out? If not, look up `WHEN` in Logo's help and you'll find the answers.

Sound Sensor

InO-Bot contains a microphone that detects ambient noise levels. The higher the value, the higher the noise level detected.

This Sound Sensor is located behind the Pen Holder and measures the level of sound. You can write programs that have InO-Bot react based on an input of a specified sound level. The Sound Sensor value range is 0–255.

To check the current noise level detected by InO-Bot's microphone, use this property:

```
GPROP "INOBOT" "NOISELEVEL"
```

Try detecting the noise level when InO-Bot is in a quiet environment and then when music is playing to calibrate the differences in the value reported. When InO-Bot is moving, the microphone detects the noise of the motors so the `NOISELEVEL` property reports a higher value than when InO-Bot is stationary.

Try this!

How could you use this information?

If the sound level is high, choose a sound to play. You could also have InO-Bot move around.

InO-Bot could continue to do some type of dance when the sound level is high and then stop when the music stops. (It sounds like a game of Musical Chairs, doesn't it?)

You'll need to do some testing to determine the level of sound that makes sense to use.

Use the same type of instruction as the one listed above for the Light sensor in the example in that section.

Distance Sensor

The Distance Sensor uses ultrasound to measure how far ahead an object is and reports the distance in centimeters. The range is from 2–15cm, with about 5% accuracy.

Try this!

To check the distance, use this property:

```
GPROP "INOBOT" "DISTANCE"
```

How could you use this information?

You might program InO-Bot to stop when it got to within a certain distance of the object.

Front and Rear Sensors

The Proximity sensors are located in InO-Bot's four corners: `FRONTRIGHT`, `FRONTLEFT`, `REARRIGHT`, `REARLEFT`. These sensors report either `TRUE` or `FALSE` if they detect an object close to that part of InO-Bot.

To check the level of light, use this property:

```
GPROP "INOBOT" "FRONTLEFT"  
Result: TRUE
```

How could you use this value?

Try this!

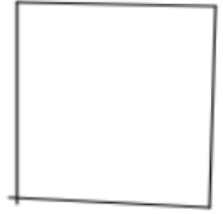
Can you write a program that will stop InO-Bot when it detects an object near it?

Note that in at least one InO-Bot we've seen, the right and left sensors are reversed, so that `FRONTLEFT` reports `TRUE` when the `FRONTRIGHT` sensor is activated.

12. Calibrating InO-Bot

You may find that InO-Bot needs to be calibrated to make its movements and turns more accurate.

If, for example, you program InO-Bot to draw a square and it doesn't end up where it started, you can make adjustments using the CALIBRATION property. Your square might look like one of the images at the right if InO-Bot's calibration needs to be adjusted.



Using GPROP with this property reports a list of two elements.

The first element is a turn adjustment factor.
The second element is a movement adjustment factor.
Initially, both values are set to 1.



Example:

```
GPROP "INOBOT" "CALIBRATION"  
Result: [1 1]
```

To adjust the calibration use PPROP to enter new values.

Now, tell InO-Bot to turn right 90 degrees, and measure the number of degrees that InO-Bot actually turned. Let us say that InO-Bot actually turned 95 degrees. In that case, you would issue the following command:

```
PPROP "INOBOT" "CALIBRATION LIST 90/95 1
```

The 90 represents the amount you want InO-Bot to turn and the 95 is what InO-Bot actually turned. The 1 at the end means that the distance amount will not change.

Type RT 90 again, and see if InO-Bot turns 90 degrees. If necessary, you can repeat the process.

Try to send InO-Bot forward 10 cm, and measure the actual distance. If InO-Bot actually moved 9.5 cm, for example, you enter this command:

```
PPROP "INOBOT" "CALIBRATION LIST 90/95 10/9.5
```

Again, the 10 represents what you InO-Bot to move. The 9.5 is what it did move.

Does InO-Bot advance 10 cm now? If you measure in inches instead of cm, you can use the same method. 10 cm equals 3.937 inches; now, measure the actual distance (9.5 cm would in this example be 3.74 inches), and enter this command:

```
PPROP "INOBOT" "CALIBRATION LIST 90/95 3.937/3.74
```

Did you successfully calibrate InO-Bot so it turns and moves more accurately?

13. How to Learn to Program in Logo

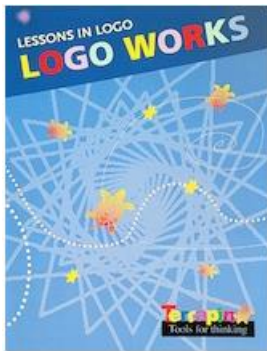
Logo is a fun language to learn. It is very powerful, with and without InO-Bot.

Terrapin Logo comes with many resources that you can use to learn Logo commands and strategies.



The Resources section of the Terrapin website provides a [Logo manual](#), [reference list of all commands and properties](#), a [list of color values](#), and a [Logo Program Library](#) of projects Logo users have created that you can explore. Why not send us your masterpiece!

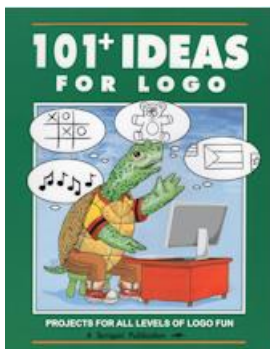
[Click here to get started.](#)



Logo Works: Lessons in Logo gets you started with a step-by-step tutorial.

Click to read the [Table of Contents](#).


To access *Logo Works: Lessons in Logo*, use your Logo login code.




101+ Ideas for Logo is divided into 5 levels of difficulty, just right for all students, from those just starting to learn Logo to the most advanced students. It covers graphics, words and lists, multiple turtles, music, and numbers in over 100 engaging Logo activities.

Click to read the [Table of Contents](#).

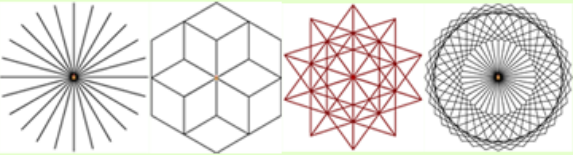
To access *101+ Ideas for Logo*, use your Logo login code.



Turtle Tuesdays



A weekly Logo (and often Pro-Bot) programming challenge!



Join us for a new coding challenge every Tuesday:

- **Facebook** - @BeeBot.us
- **Instagram** - terrapin.tools.for.thinking

Turtle Tuesdays: Every week, Terrapin posts a free challenge for Terrapin Logo users. Most can be done using InO-Bot and many can also be done using Pro-Bot or *Kinderlogo*. Solutions are posted the week after the challenge, so if you haven't figured out how to code the challenge, you can see one way to do it. The challenges are also posted on social media.

14. InO-Bot Specifications, Charging Information, Care & Maintenance

Below are the details about your InO-Bot. The InO-Bot has been designed to be as accurate as possible in its movements. However, the following tolerances do apply. In most cases, you will find that InO-Bot will perform even better on a good clean surface free from bumps.

Product Specifications

| | |
|---|---|
| Forward/backward movement | 100mm \pm 7mm |
| 90° turn | \pm 4 degrees |
| Speed of movement (depends on battery condition) | 100mm/sec |
| Batteries | 3.7V 1500mAh Lithium Polymer |
| Battery life | Approximately 5–6 hours of normal usage, 3 hours when in continuous use |
| Ultrasonic range finders | 2cm to 150 cm, \pm 5% |
| Light sensor range | 0–255 |
| Line sensor range | Will follow a black line 5–10mm thick |
| Sound sensor | 0–255 |
| Proximity sensor | On/Off |
| Infrared | 255–0 |
| Bluetooth range | Approximately 10 meters |

Recharging Your InO-Bot-Bot

- Insert the USB cable into the charging socket on the underside of your InO-Bot.
- Connect the other end to a spare USB port on a PC or laptop. Alternatively, connect to a USB charging plug.
- All RGB LEDs will flash red to indicate that InO-Bot requires charging.
- The red LED indicator will shine solid red when charging.
- The red LED indicator will turn off when InO-Bot is fully charged.
- It may take up to 4 hours to fully charge the InO-Bot if the battery has been fully discharged.
- Once fully charged, InO-Bot will have approximately 5–6 hours of normal usage and approximately 3 hours while in continuous use.
- Always fully charge InO-Bot before storing for long periods. At a minimum, recharge InO-Bot every six months to maintain optimal battery capacity,

Battery Precautions

- InO-Bot is powered by a Lithium Polymer battery pack that is not user-replaceable.
- Rechargeable batteries are only to be recharged under adult supervision.

Care & Maintenance

- If necessary, wipe InO-Bot gently with a clean damp cloth.
- Keep InO-Bot away from direct sunlight and heat.
- Do not allow InO-Bot to come into contact with water or other liquids.
- In the event of a static discharge, InO-Bot may malfunction. If this occurs, switch InO-Bot off and then back on again to reset it.