LEGO
MINDSTORMS®
EV3
10+
31313
Software for PC/Mac
USB/Bluetooth
Batteries not included
Infrared Sensor
Makes your robot see
Colour Sensor
Makes your robot recognize colours
Programmable Brick
The brain and voice of your robot
Touch Sensor
Makes your robot feel
EV3RSTORM
TRACK3R
GRIPP3R
R3PTAR
SPIK3R
Remote Control with two modes
Command your robot via infrared light
3 Interactive Servo Motors
Make your robot move
Command your robot with your smart devices
CREATE & COMMAND YOUR OWN ROBOT

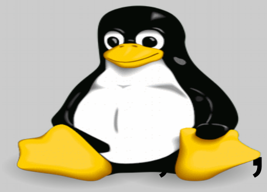
Content:

- What is LEGO® MINDSTORMS®?
- Getting Started
- Some programming basics + activity
- EV3 + Python

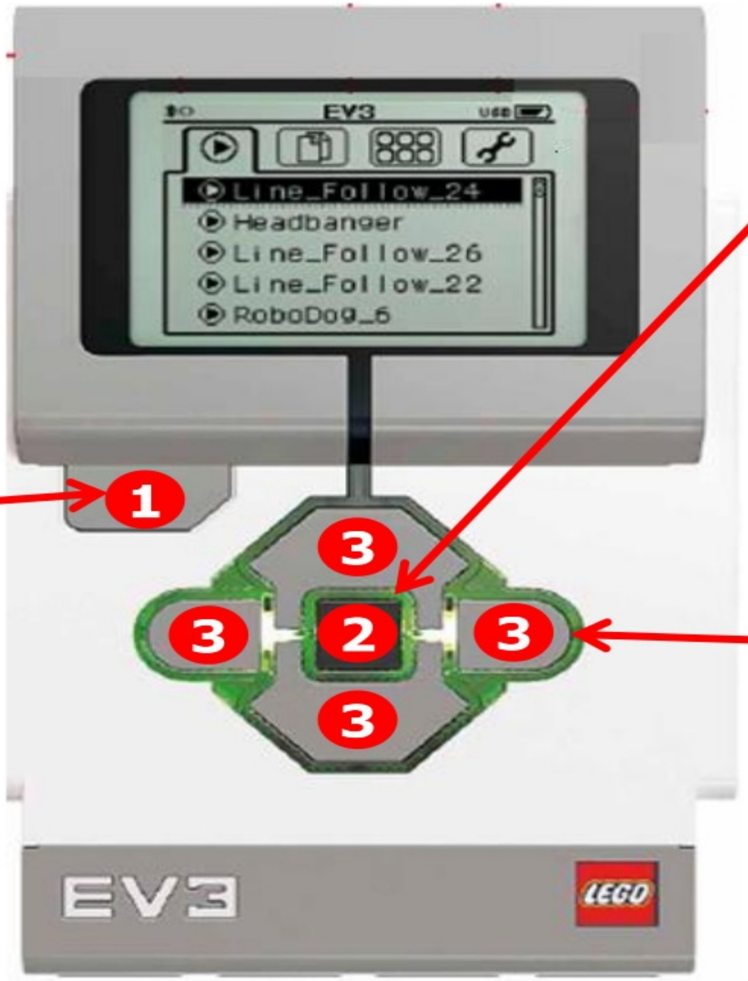
What is LEGO® MINDSTORMS®?

- LEGO MINDSTORMS is a programmable robotics construction series that gives you the power to build and program your own LEGO robots. The LEGO MINDSTORMS EV3 set includes everything you need – motors, sensors, programmable brick (P-brick), cables, remote control and TECHNIC elements – to create robots that walk, talk, move and do whatever you want them to.
- Build your own robot and program it through the intuitive software program, the P-Brick or your smart device.

How to getting started?

- Installing the software (Sorry Linux , software available only on Mac and Windows OS)
- Unboxing the core set and sorting the bricks
- Powering up your EV3 Brick
- Labeling key components

EV3 Brick



<u>1 = Back</u>, this button is used to reverse actions, to stop a running program, and to shut down the EV3.

<u>2 = Center</u>, press the center button to select and accept options, or run a program.

<u>3 = left</u>, right, Up, Down These four buttons are used to navigate through the various menus.

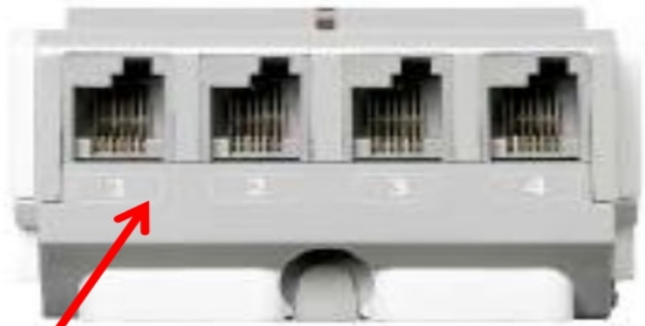
Programmable EV3 Brick

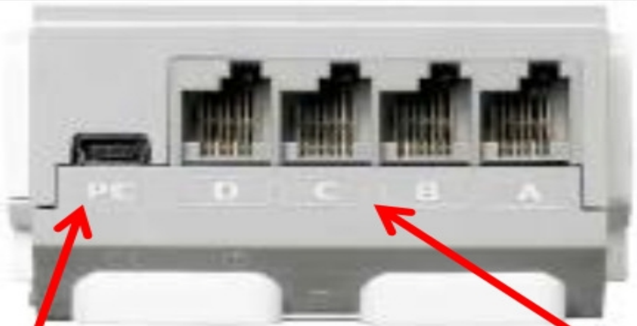
- Four outputs (motors)
- Four inputs (sensors)
- USB, Bluetooth, or Wi-Fi connection
- Improved LCD screen
- 16 MB flash memory
- 64 MB RAM
- SD Card Port: 32 GB
- Multiple onboard utilities
- 1,000 samples per second
- EV3 Brick Button lights
- Sound

Ports, Sensors and Motors

1, 2, 3, 4 = Input ports used for sensors.

A, B, C, D = Output ports used for motors.



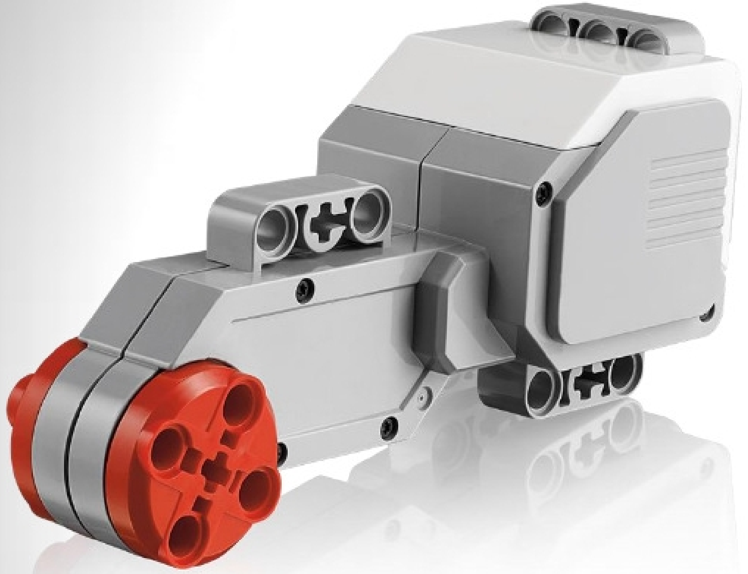


Large Motor

Medium Motor

The PC USB port is used to connect to The PC so you can download the Programs into EV3 Controller

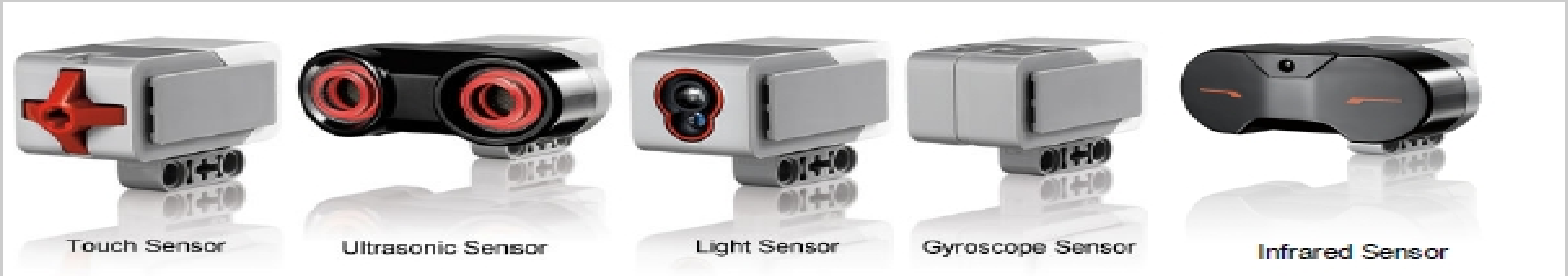
EV3 Motors



- Two types of motors
- Redesigned to allow easy construction
- The Large Motor is a strong and powerful full motor.
- The Medium Motor is a less powerful motor but runs at a higher revolution rate.
- Both motors have tacho feedback enabling 1 degree resolution.
- Both motors are Auto ID–supported.
- The Medium Motor is smaller and lighter to allow more construction options.

EV3 Sensors:

- Color – measures color and darkness
- Gyro – measures rotation of robot
- Ultrasonic – measures distance to nearby surfaces
- Touch – measures contact with surface
- Infrared – measures IR remote's signals



EV3 Ultrasonic Sensor

- Detects distance
- Accurate to 1 cm or 0.3 inches
- Can listen for other ultrasonic sensors
- Improved design for easier build solutions
- Eyes light up to identify which mode the sensor is operating in
- Auto ID

EV3 Color Sensor

- Detects eight different colors
- Detects ambient light, from dark to sunlight
- Detects reflective red light
- Built-in cancelling of backlight makes sensor more reliable
- Improved design for easier build solutions
- Auto ID

Touch Sensor

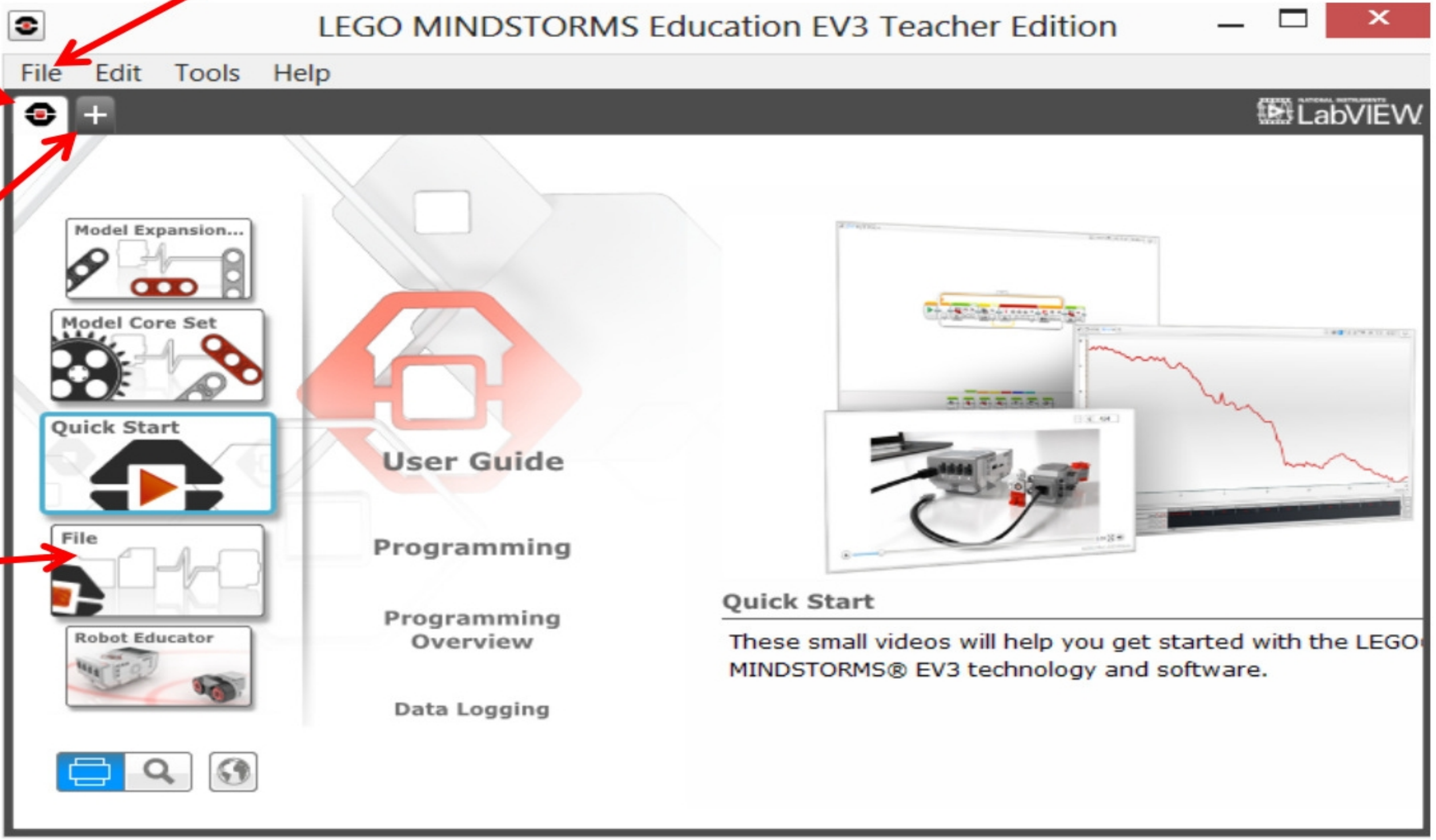
- Detects pressed
- Detects released
- Detects bumped
- Improved design for easier build solutions
- Auto ID

Gyro Sensor

- Angle mode
- Gyro Sensor mode
- Angle and Gyro Sensor modes
- Can reset accumulated angle value
- Improved design for easier build solutions
- Auto ID

EV3 software(MAC / WINDOWS OS)

EV3 Navigation



These small videos will help you get started with the LEGO MINDSTORMS® EV3 technology and software.

Open a previously saved project

Lobby Button

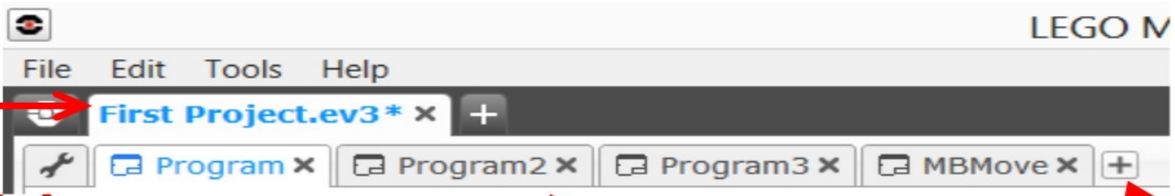
Open New Project

Open New Project or previously saved ones

Projects and Programs

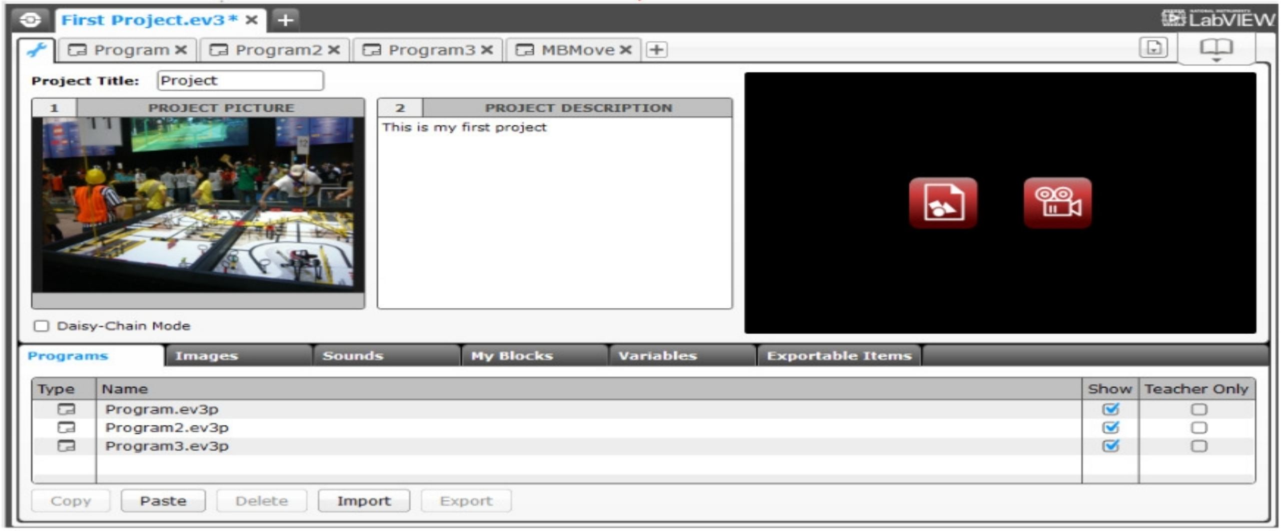
Opened Project

Project Properties



Currently Opened Programs belonging to opened project

Click to create a new program within the current project



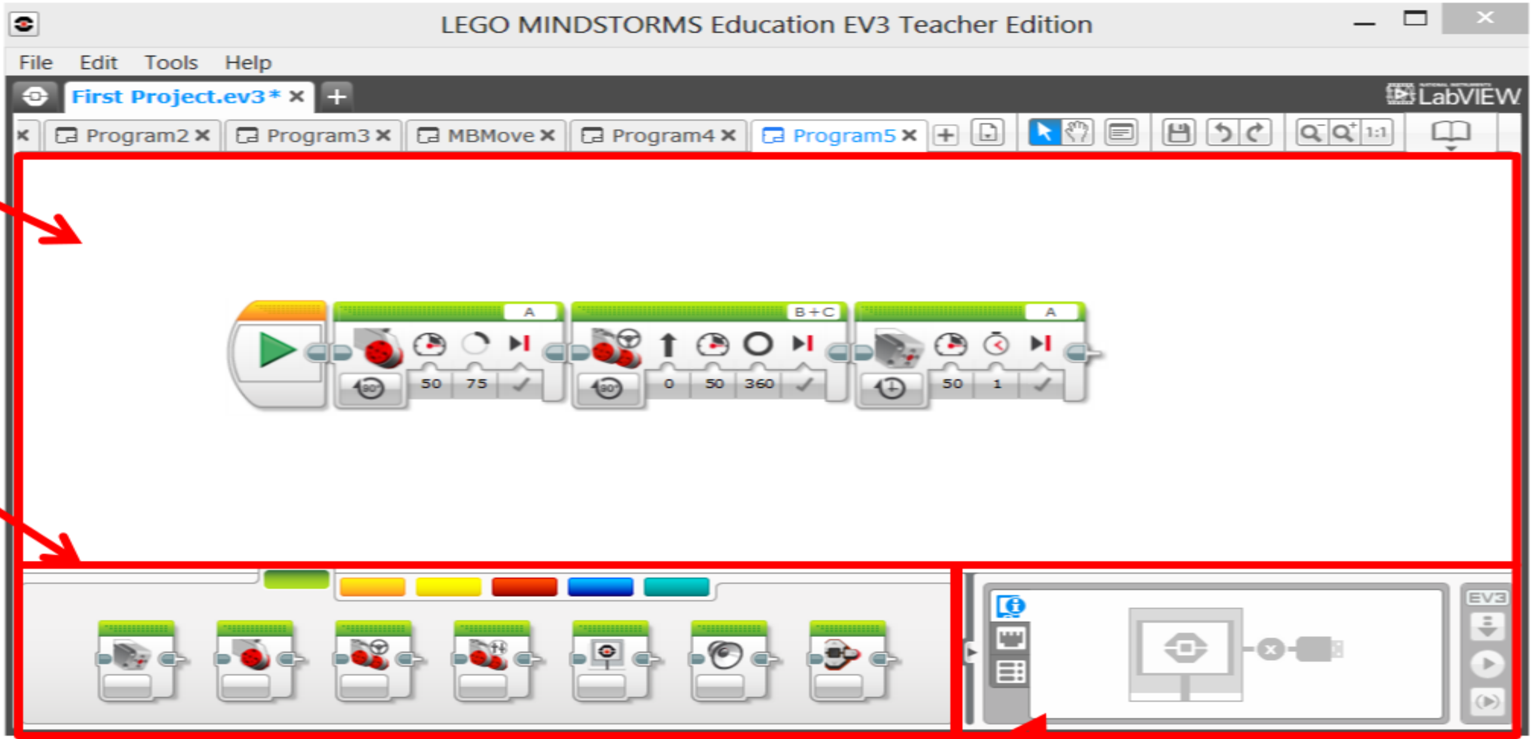
Project Title: Project

1 PROJECT PICTURE	2 PROJECT DESCRIPTION
	This is my first project

☐ Daisy-Chain Mode

Type	Name	Show	Teacher Only
	Program.ev3p	☑	☐
	Program2.ev3p	☑	☐
	Program3.ev3p	☑	☐

Programming Environment Workspace



Programming canvas where you can lay out the program's blocks / instructions

Programming palettes where you can find the various building blocks

Hardware page establishes communication with the EV3 brick and where you download programs into the EV3, view memory usages, battery level, and to find out motors or sensors and where they are connected.

EV3 software(MAC / WINDOWS OS)

Green – action blocks

Orange – flow blocks

Yellow – sensor blocks

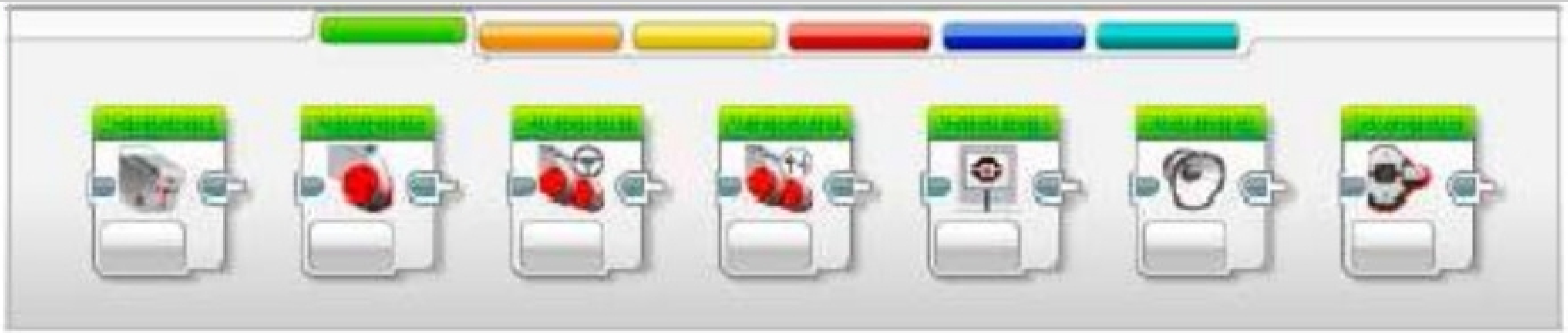
Red – logic blocks

Dark blue – advanced blocks

Light blue – My blocks

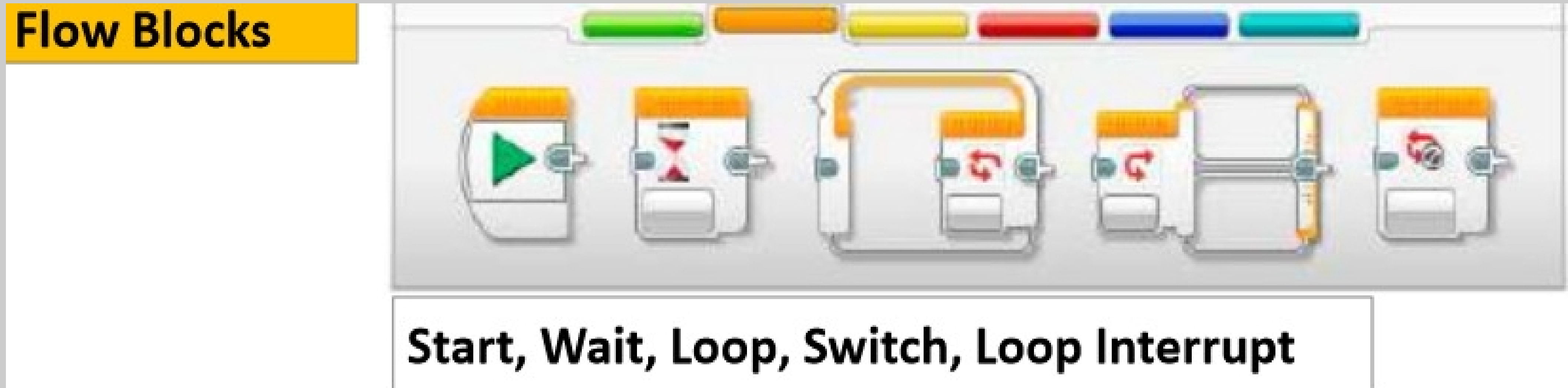
EV3 software(MAC / WINDOWS OS)

Action Blocks



Medium Motor, Large Motor, Move Steering, Move Tank, Display, Sound, Brick Status Light.

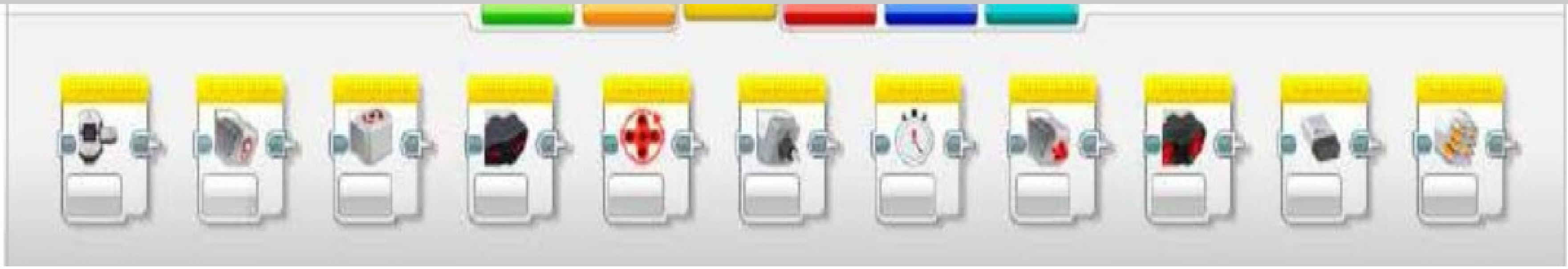
EV3 software(MAC / WINDOWS OS)


Flow Blocks

Start, Wait, Loop, Switch, Loop Interrupt

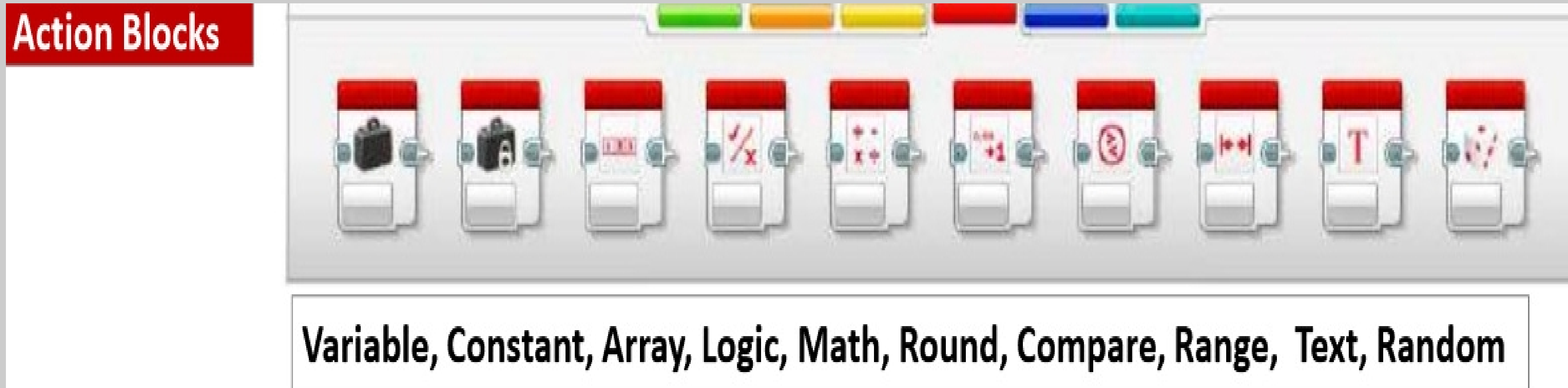
EV3 software(MAC / WINDOWS OS)

Sensor Blocks



Brick Buttons, Color, Gyro, Infrared, Motor Rotation, Temperature, Timer, Touch, Ultrasonic, Energy Meter, Sound

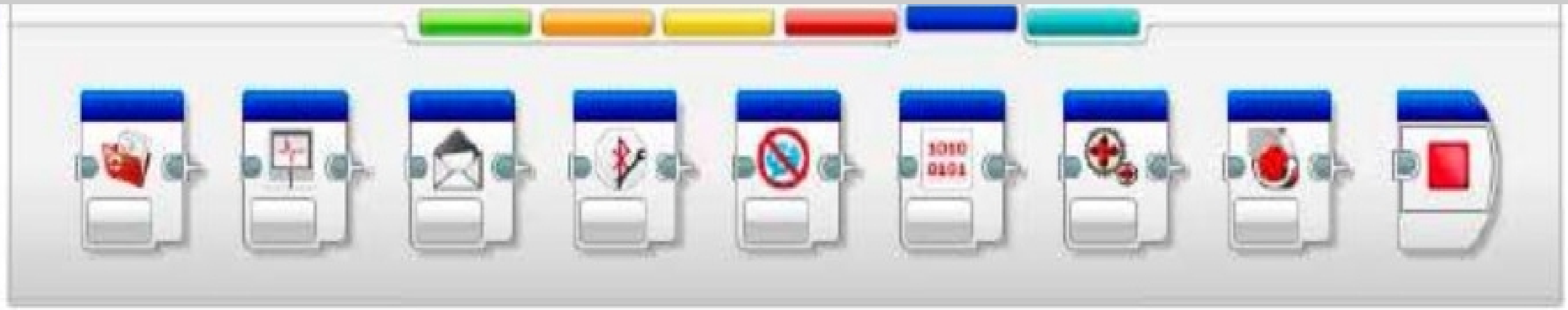
EV3 software(MAC / WINDOWS OS)


Action Blocks

Variable, Constant, Array, Logic, Math, Round, Compare, Range, Text, Random

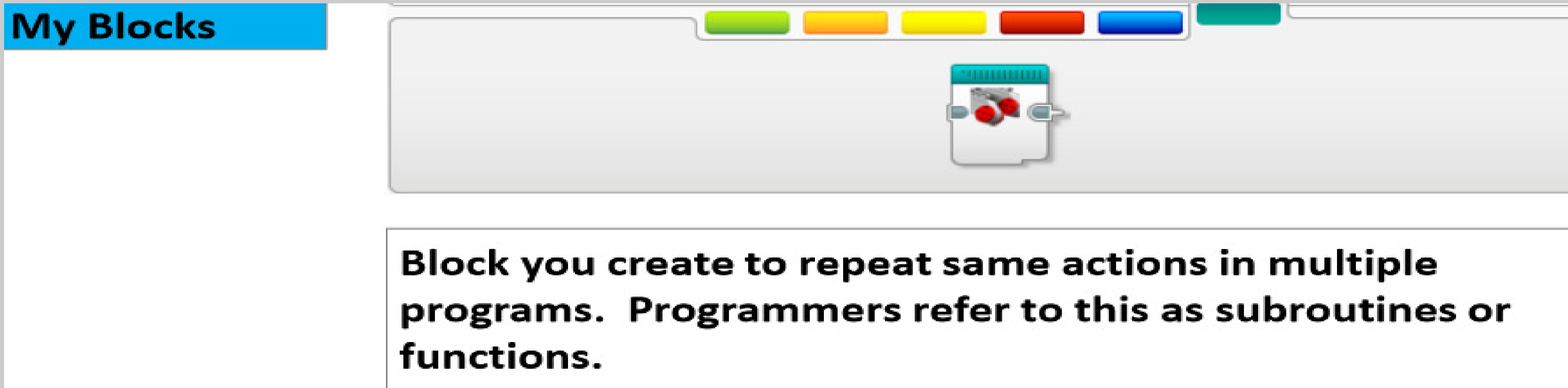
EV3 software(MAC / WINDOWS OS)

Advanced Blocks



File Access, Data Logging, Messaging, BlueTooth, Keep Awake, Raw Sensor Value, Unregulated Motor, Invert Motor, Stop Program

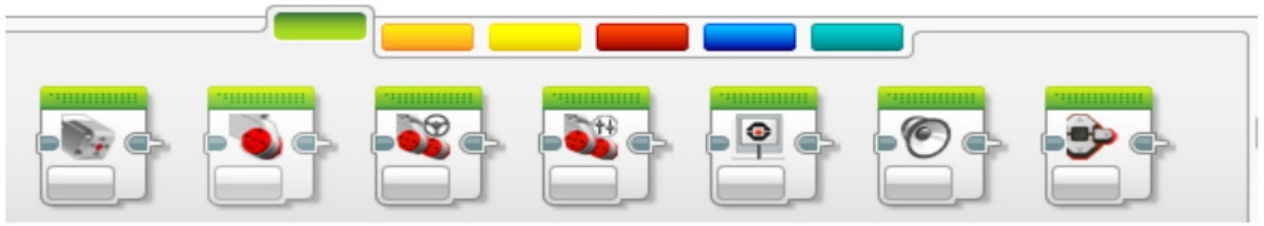
EV3 software(MAC / WINDOWS OS)



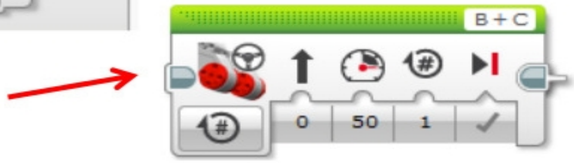
My Blocks

Block you create to repeat same actions in multiple programs. Programmers refer to this as subroutines or functions.

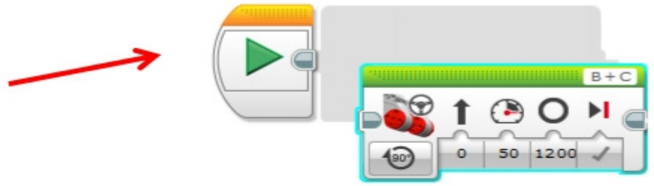
Steps to create a program



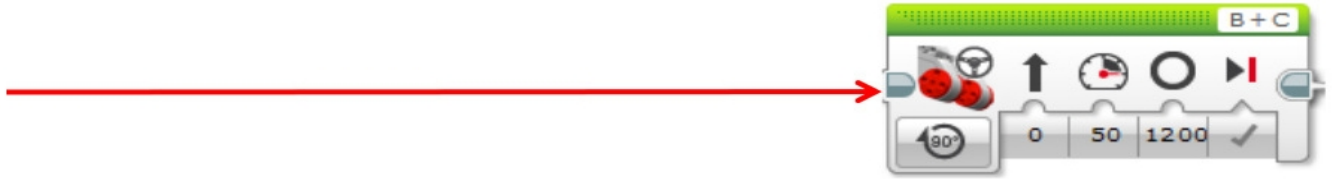
1. Click and hold block with left mouse button to drag it



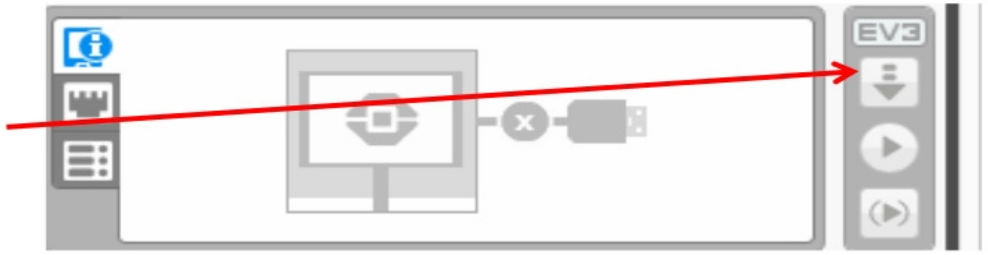
2. Drop the programming block when grey box appears



3. Select / enter options



4. Click download to compile and load the program in the EV3 controller



What’s new

- All files are stored within the Project file, i.e., programs, my blocks. You can move / copy the project file to other computers and it will work. Now you can backup the entire project or even use a memory stick to store your project!

- Turning on/off the EV3 now takes about 30 seconds

- The **MOVE** block is replaced by the **MOVE STEERING** and **MOVE TANK** blocks
  - **MOVE STEERING** has single power control; motors are regulated, i.e., if one motor moves faster than the other, the faster motor will be slowed down to compensate.
  - **MOVE TANK**: has independent power controls for each motor where one can move faster than the other or even in opposite direction. This too is regulated.
    NOTE: with limited testing, it appears that issues using steering in NXT are solved in EV3!

- The **MOTOR** block is replaced by **LARGE MOTOR** and **MEDIUM MOTOR**

- In NXT-G you specified direction, in EV3, you specify either negative or positive power to control the direction of the motors

- The **unlimited** duration option is replaced by **ON**

- The **STOP** option of **MOVE** and **MOTOR** blocks are replaced by **OFF**

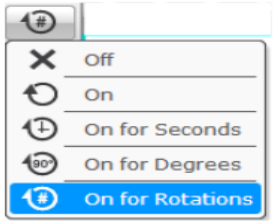


Image 1, Ref - see slide 17-18

What is a Robot?

A look at characteristics of robots using the LEGO EV3 as a specific example

(50 minutes)

PRE/POST-ASSESSMENT SHEET - What is a robot?

1. Describe in one sentence below what you understand by the term 'robot'

2. What are the main parts of a robot?

3. What do people do to make a robot move?

PRE/POST-ASSESSMENT SHEET - What is a robot?

1. Describe in one sentence below what you understand by the term 'robot'

A robot is a machine that gathers information about its environment (senses) and uses that information (thinks) to follow instructions to do work (acts).

2. What are the main parts of a robot?

Computer (to make decisions), Input ports (connected to sensors), and Outputs (connected to motors, for example).

3. What do people do to make a robot move?

They program it using software telling it precisely what to do, step by step.

Robots in the World

- We always think of Robots in movies
  - Human Characteristics
  - Learning
  - Emotions
- While there have been advances is Artificial Intelligence (AI)....

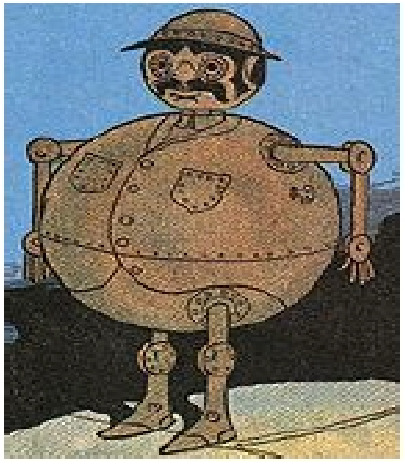


Image 2-5, Ref - see slide 17-18

Robots in the World

- …most robots look and act nothing like what we imagine in movies.
- In fact, many everyday objects are robots, even though you may not have ever noticed.
- This doesn't mean that what they do is any less amazing!

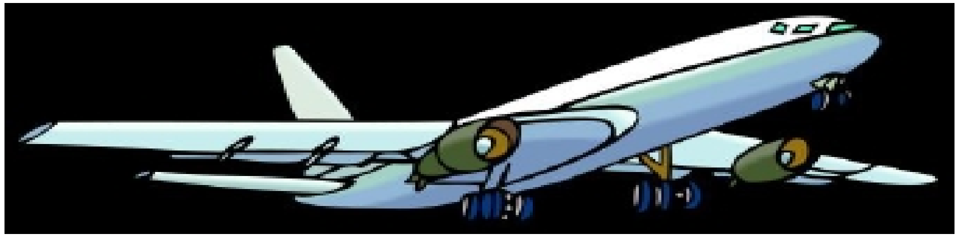


Image 6-8, Ref - see slide 17-18

How do you define a robot?

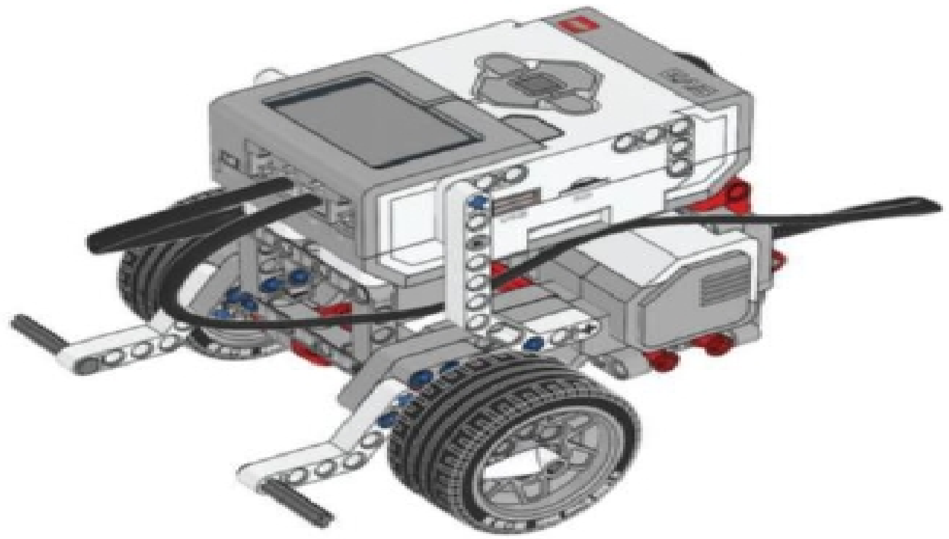
- Definition: “A robot is a machine that gathers information about its environment (senses) and uses that information (thinks) to follow instructions to do work (acts).”[1]
- Senses – using SENSORS; thinks – using COMPUTER; acts – using MOTORS, for example.
- Engineers design robots to perform complex tasks more easily and with greater accuracy. Some everyday examples of robots include: automatic car washes, vending machines, automatic doors, robotic arms used in manufacturing, remote control cars and trucks, automatic teller machines (ATMs)
- Can you name some devices that are robots?
- ...and some that are not?

Some Robot videos

- LEGO Robots
- Engineering for the Red Planet
- Anatomy of a Rover
- Kismet
- RoboSnail
- Robofly
- Design Inspired by Nature

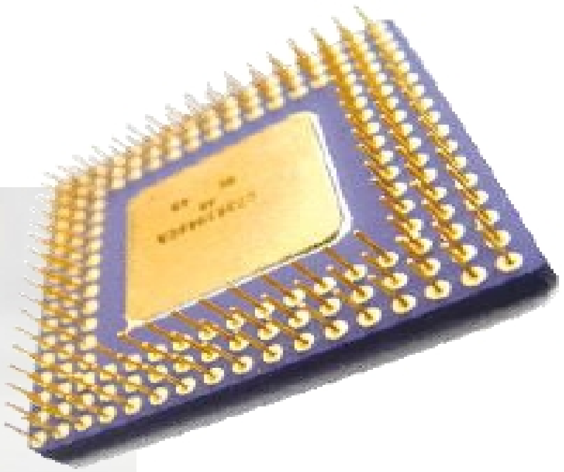
The EV3

- The specific robot that we will be working with is the LEGO EV3.
- The first robot you will make with the EV3 is called the "Robot Square".



What Makes Something a Robot? #1:
A Computer (to help it 'make decisions')

- There is one characteristic that ALL robots have:
  - They are ALL controlled by a computer
    - Not necessarily a computer like the one you have at home!
    - Computers come in all shapes and sizes.



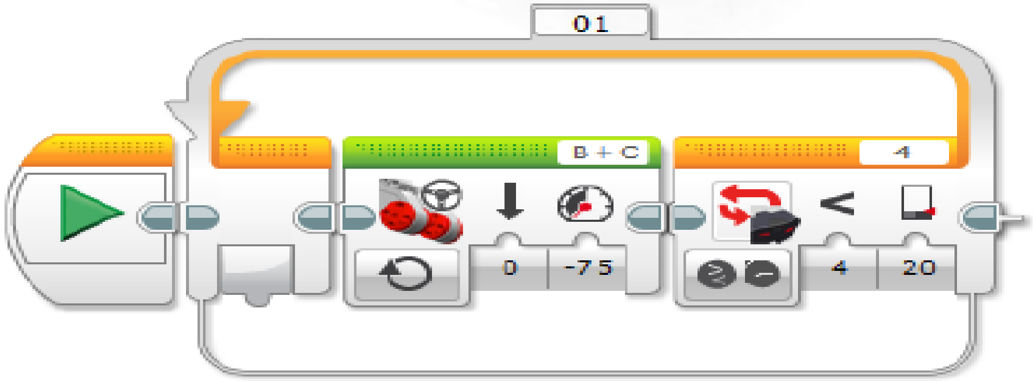
Believe it or not, these all are computers!

What Makes Something a Robot? #1: A Computer (to help it 'make decisions')....Contd.

- **Since robots cannot think <u>on their own</u> like us, they have to follow specific instructions <u>to make decisions</u>.**
- **These instructions are given to their computer in the form of a *program*.**
- **The computer can then read the program and act like the robot's brain, controlling the robot based on just running each instruction in the program in sequence.**

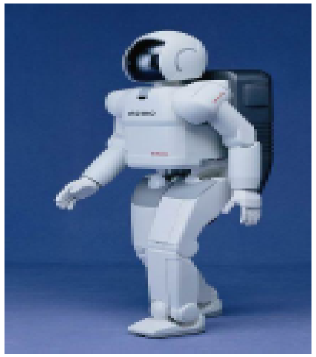
Computer of the EV3

- The computer of the EV3 is called the EV3 computer brick.
- We can program the EV3 using a programming software that LEGO provides with the EV3.



What Makes Something a Robot? #2:
Inputs to 'SENSE' (via sensors)

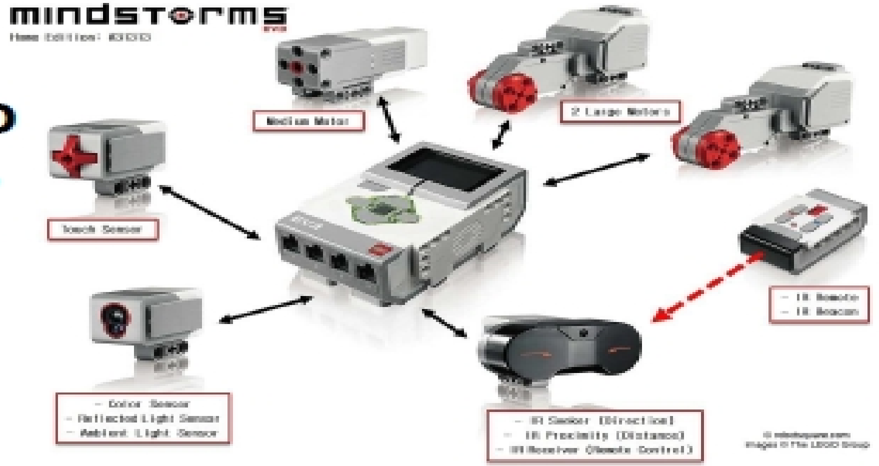
- Robots need inputs, which provide information to the its computer to make decisions.
  - Think of it as a signal going "in" to the computer
- Example
  - The mouse and keyboard on your computer
  - A camera on a robot that can act as its eye
- Note: The important thing about inputs is that they have no effect if the program doesn't tell the computer to look for them!



INPUTS OF THE EV3: (SENSORS)

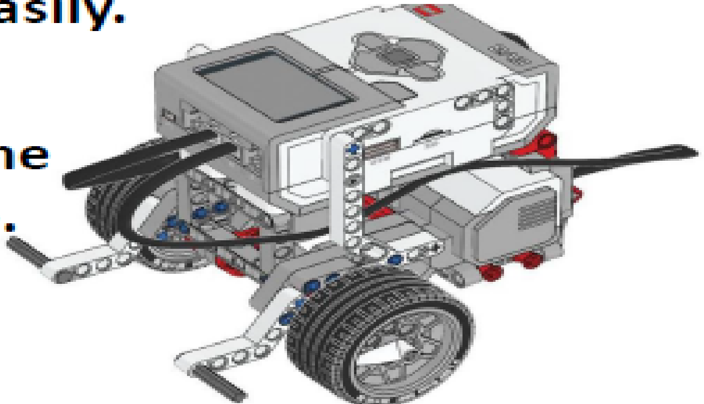
- Four sensors come with the EV3
  - Color sensor, touch sensor,
  - Rotation sensor, ultrasonic sensor

- The sensors are connected to the INPUT ports of the EV3 – 1, 2, 3, 4



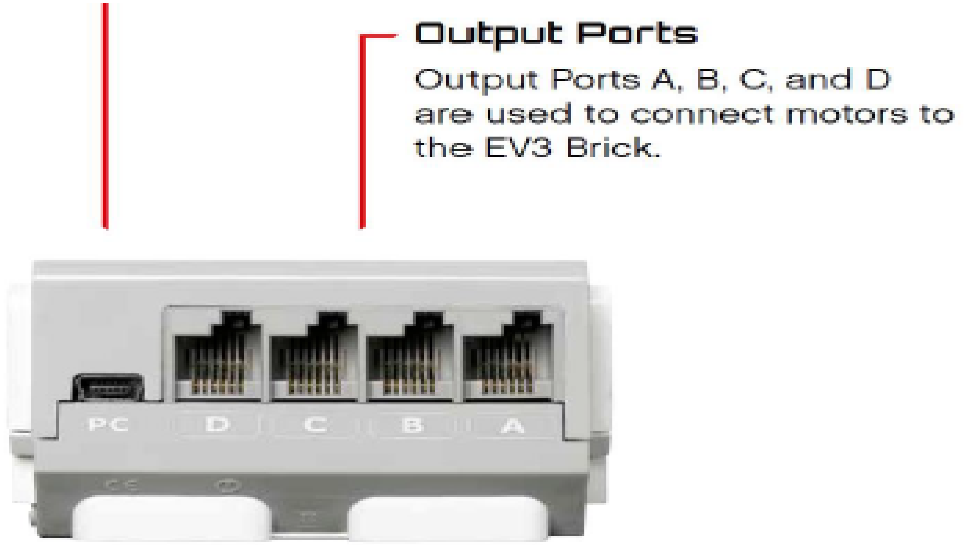
What Makes Something a Robot? #3: Outputs to 'ACT' (via motors, for example)

- An output is a command sent out by the robot computer to some other part of the robot
  - You can think of it as a signal going "out" of the computer.
- Examples:
  - Your computer monitor
    - Your computer sends an image to your computer screen so that you can see what the computer is doing and use it easily.
  - Motors on a robot
    - The computer sends a signal to the motors of a robot to make it move.



Outputs of the EV3

- The output ports are located at the top of the EV3 computer bricks. These ports don't take in any information, they only send it out.
- We can attach motors or lights to the output ports of the EV3 and then program the computer to activate them.

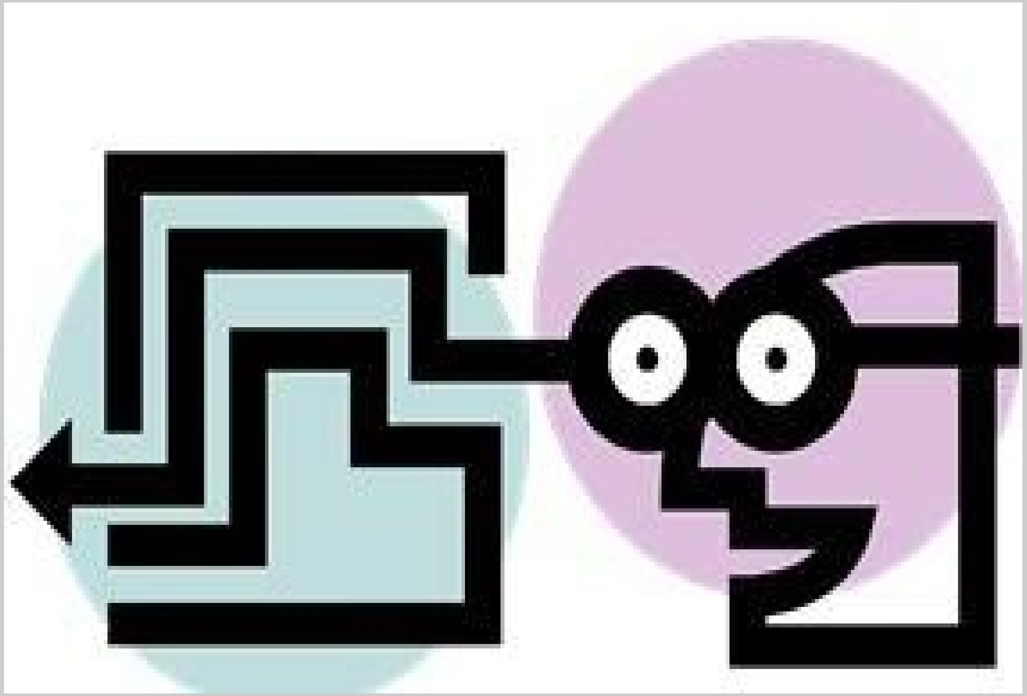

Output Ports
Output Ports A, B, C, and D are used to connect motors to the EV3 Brick.
PC
D
C
B
A

SUMMARY

- The parts of a robot are
  - A computer that needs to be programmed (to make decisions)
  - Inputs (to 'sense' via sensors)
  - Outputs (to 'act', e.g., via motors)

- Our EV3 has all three of these, and in the following activity, we'll see how exactly we can put all these together and make the EV3 robot move based on your commands.

Maze challenge



Pre-Activity Quiz

1. **What is a design challenge?**

2. **How do you program a robot to move 2 feet forward in a straight line?**

Pre-Activity Quiz Answers

1. **What is a design challenge?**

   A design challenge starts with problem you want to solve. You think through the challenge logically and plan a design solution. Using suitable and available materials and following the steps in the engineering design process, you create, test and evaluate your best solution to the challenge. As necessary, you iterate (repeat) this process until a successful design is achieved.

2. **How do you program a robot to move 2 feet forward in a straight line?**

   Calculate the distance that the robot moves forward for one motor rotation. Let's say, it moves X inches with one motor rotation. Then divide 2 feet, which is 24 inches, by X, that is, calculate 24/X and program the motor to rotate 24/X.

What Is a Design Challenge?

Day 1 - 50 minutes

In our modern world, challenges are everywhere!

How can we waste less? How can we harness solar energy and other renewable energy more effectively? How can improve transportation? How can we build roads and bridges? How can we design a house that is not too expensive? How can we build smarter cars? How can we use technology to make our cars safer?

These are big ideas that engineers and scientists work on to help improve the world we live in.

We will investigate how to complete a few design challenges through two different methods:

- Creating different robot designs to help it better complete the challenges
- Creating programs to help the robot complete the design challenge

Your Engineering Challenge:

To build and program a robot to travel through a maze.

- We will look at different robot designs (with and without sensors) in order to determine which is more reliable and which is faster.
- Both the design of the robot and how it's programmed are important for this challenge!

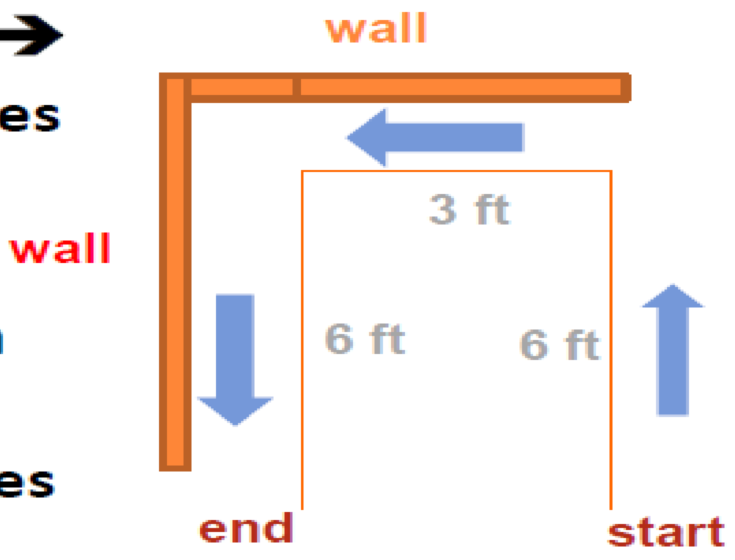
Let’s Look at Some Basics

- Before we start the design challenge, it is important to understand how and why we need to be careful with designing the robot and designing its program.
- Also, it is very important that we understand how a robot follows instructions so we can understand how to program it.
- Let’s do an activity to help us understand two important ideas:
  1. How a robot follows instructions
  2. The importance of sensors

Let's Look at Some Basics (continued)

Maze 1 is a 3-ft wide path in a corner of the classroom.

It looks like this diagram ➔



Use tape (or boxes) to mark the boundaries shown in fine lines.

Blindfold a student "robot" and have him stand at the maze start.

Have a student "programmer" give a series of commands to instruct the "robot" to complete the maze *without touching the maze edges* (keeping hands at his/her sides).

***Command examples*: go forward X steps, turn left, etc.**

Let’s Look at Some Basics (continued)

What did you observe?

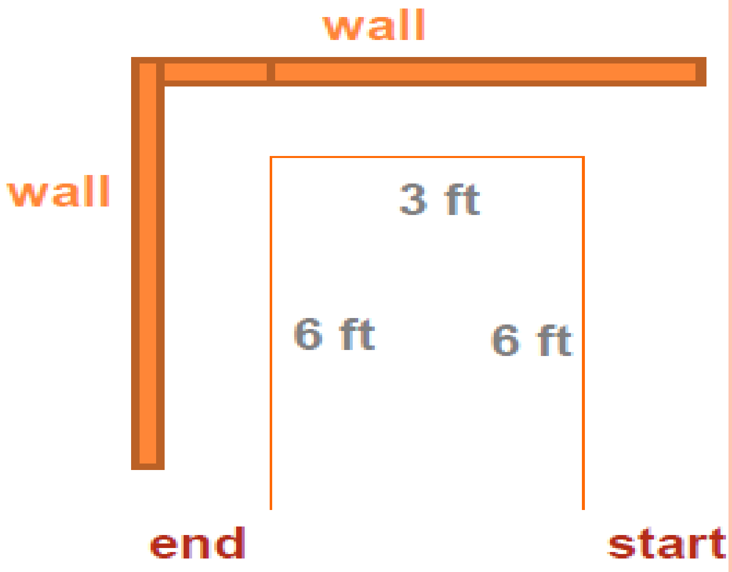
- Did the “programmer” get the blindfolded “robot” through the maze?
- Did the “commander” always tell the “robot” to go the correct number of steps?

Often, it is easier for the “programmer” to give instructions if s/he can tell the “robot” to go forward until s/he senses something

By doing it that way, the “programmer” does not have to worry about telling the “robot” exactly how many steps to move.

Let's Look at Some Basics (continued)

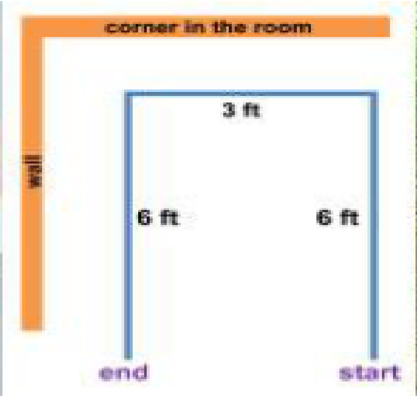
- Choose a different volunteer "robot" to be blindfolded.
- This time, permit the volunteer "robot" to stretch his/her hands out in front to sense when s/he is approaching a wall.
- Have another student give commands to instruct the volunteer to get through the maze
- Now, commands such as "go forward until you sense a wall" are allowed.



Let's Look at Some Basics (continued)

What did you observe?

- Was it easier to give instructions the first time or second time?



- It is generally easier to tell a robot to go forward until it senses something (like a wall) than to tell it exactly how far it should go before turning. This approach also tends to be faster.

Let's Look at Some Basics (continued)

What else did you observe?

- Did you notice that the "commander" had to make sure to tell the "robot" EVERYTHING? …which direction to move, how far or how long to move, when to turn, which way to turn, etc.
- Likewise, the LEGO MINDSTORMS EV3 intelligent brick must be given exactly the same information. It knows nothing and will do exactly what you tell it to do. If you make a programming error, the robot will follow that incorrect command, and the fault is with the programmer, and not the robot!

Before you start…

- Before we start our maze challenges, let’s learn the relationship between motor (or wheel) rotation and the corresponding distance the robot travels.
- *Why is this important?* Because, when we have the robot travel the maze *without* any sensors, we must tell it how much distance to move.
- Use your results from the *Master Driver* activity to inform your programming! (Recall that this is the activity where you determined the distance traveled by the taskbot for every rotation of the EV3 motor.)

A Real Maze

Day 2 - 50 minutes

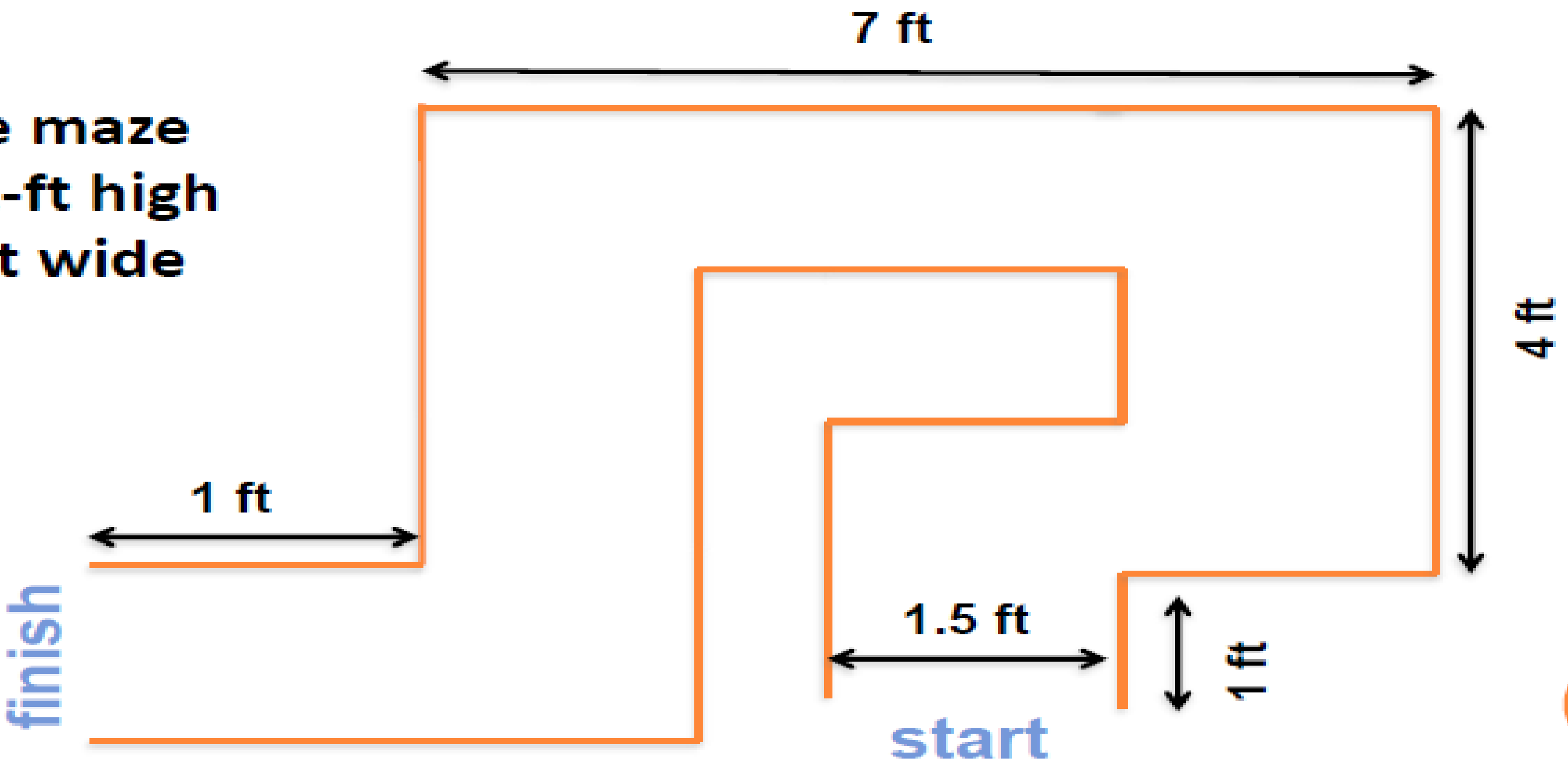
Challenge 1: Make the LEGO robot navigate the maze *without using any sensors*

Challenge 2: Make the LEGO robot navigate the maze using the sound sensor, the touch sensor, and ultrasonic sensor *at least once each*

The Challenges Maze

Suggested dimensions for the maze are shown in the diagram below, with "start" and "finish" locations indicated. ↓

Make the maze at least 1-ft high and 1.5-ft wide



Challenge 1

To program the taskbot to travel the maze *without* using sensors.

- For this first challenge, the robot must go through the maze *without using sensors*!
- How will you program the robot to do this?
- Think back to the *Master Driver* activity and what you learned about the relationship between the number of rotations of the motor and the distance the robot moved.
- You will be provided with a measuring stick or tape measure for this challenge. *Be resourceful!*

Challenge 2

To program the taskbot to travel the maze using the sound sensor, touch sensor, and ultrasonic sensor each *at least* once.

- For this challenge, your robot must use the touch sensor, sound sensor and ultrasonic sensor *at least once each* when navigating the maze.
- Think back to how we did basic programming with the touch sensor, ultrasonic sensor and sound sensor, and combine that information to get the robot to travel through the maze.
- In the robot documentation, find information about attaching sensors to the robot. Or, come up with your own designs for attachment also for this challenge. *Be creative!*

Post-Activity Quiz

1. **What types of problems did you encounter when trying to complete the design challenges?**

2. **How did you change the design of your robot or your programs to help you complete the tasks?**

Post-Activity Quiz Answers

1. What types of problems did you encounter when trying to complete the design challenges?

   Calculating how many rotations to use in order to have the robot move a measured distance in the maze.

   Recalling how to begin programming; how to program each sensor.

2. How did you change the design of your robot or your programs to help you complete the tasks?

   We re-attached the sensors to position them correctly to receive the necessary input information.

   Through troubleshooting we corrected some programming errors.

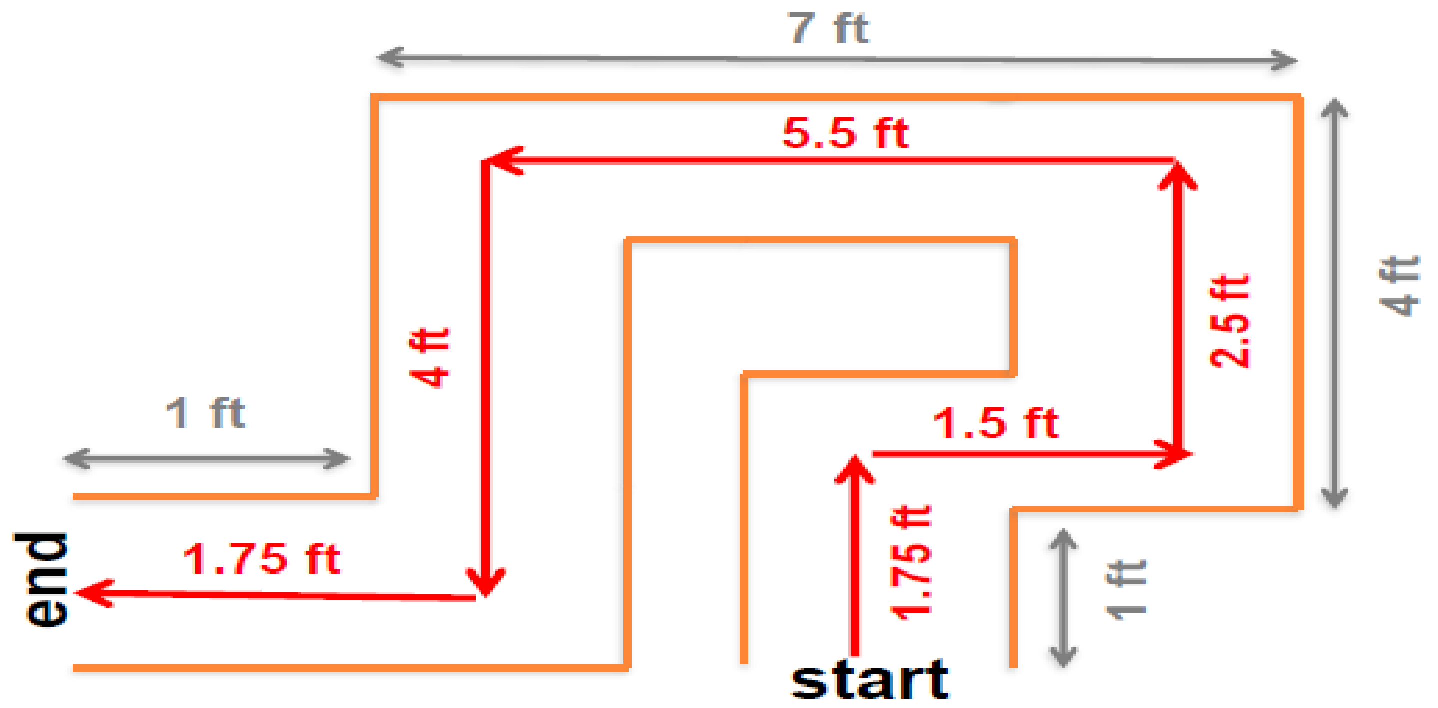
Challenge 1

Maze without Sensors Solution

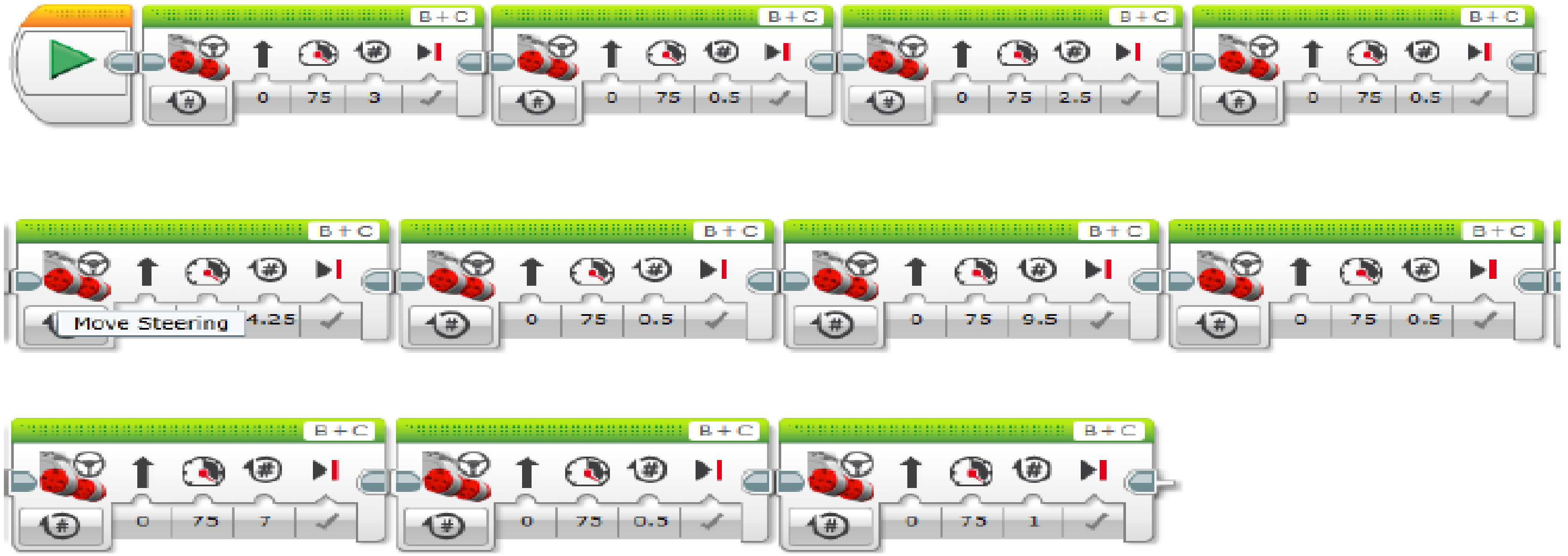
- Recall the relation between rotations and distance: For each rotation, the robot travels a certain distance (inches or cms) and we can use *multiplication* to approximate how far the robot will travel.
- So, we must measure each section of the maze and guess how many rotations we need based off each distance!
- This may not work perfectly the first time, but the number of rotations can be modified so it eventually works.

Challenge 1 Program Solution

In an effort to keep the robot as far from the walls as possible, calculate path distances so that the robot stays in the center of each corridor (.75 ft away from the wall).



Challenge 1 Program Solution



```
B+C  0  75  3
B+C  0  75  0.5
B+C  0  75  2.5
B+C  0  75  0.5

B+C  Move Steering  4.25
B+C  0  75  0.5
B+C  0  75  9.5
B+C  0  75  0.5

B+C  0  75  7
B+C  0  75  0.5
B+C  0  75  1
```

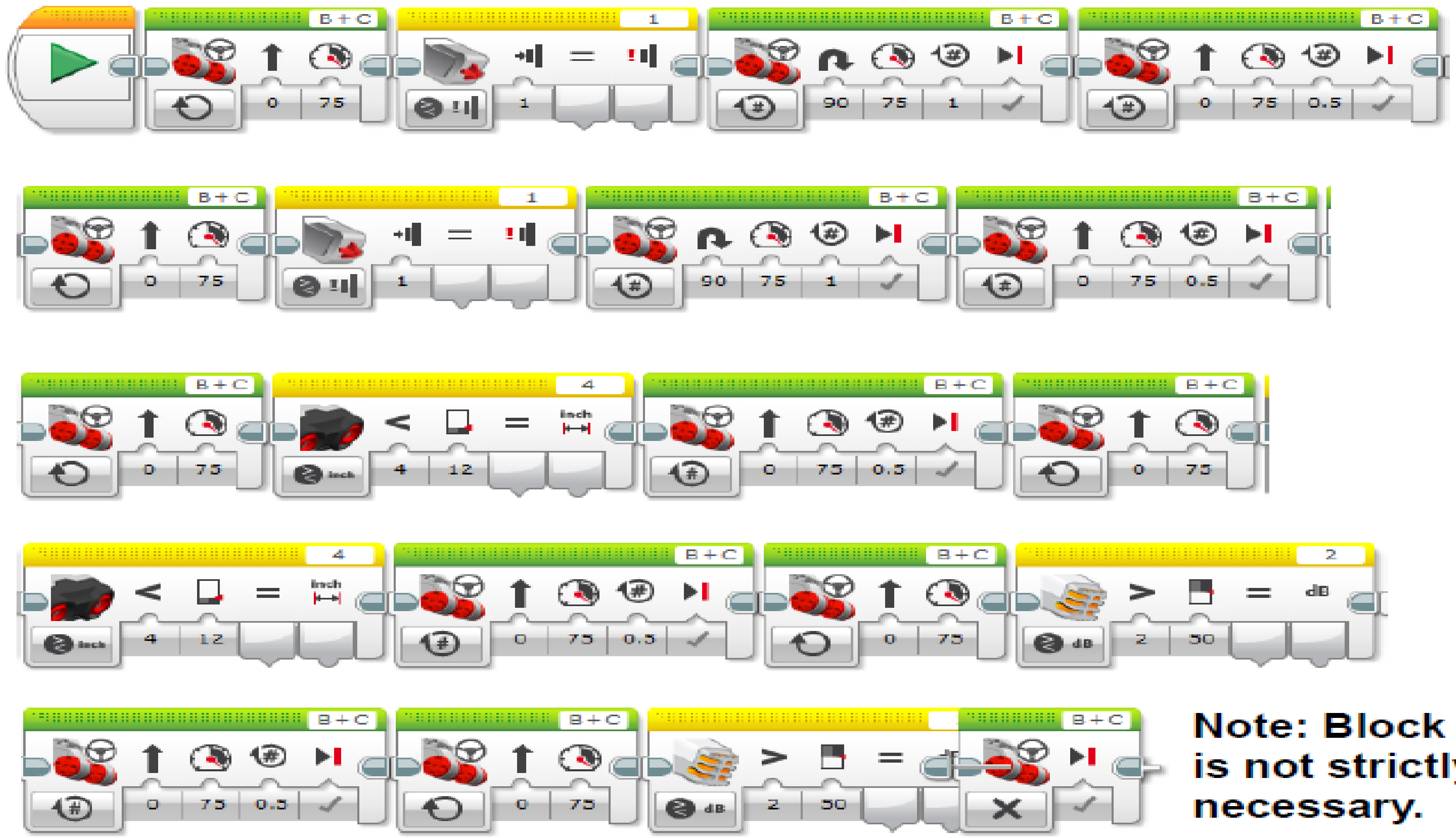
Challenge 2

Maze Using Sensors Solution

How to complete the maze by using the LEGO robot sound, touch and ultrasonic sensors:

- Remember, since we can use the sensors, we do not have to worry about distance and guessing how many rotations we need!
- For each section of the maze, we can program the forward and backward movement blocks using unlimited, and then immediately place a sensor block after the movement block.
- We must use the ultrasonic and sound sensor once in this program, then we can choose to use the sound sensor for the rest of the maze or the ultrasonic sensor for the rest of the maze or both.

Challenge 2 Program Solution

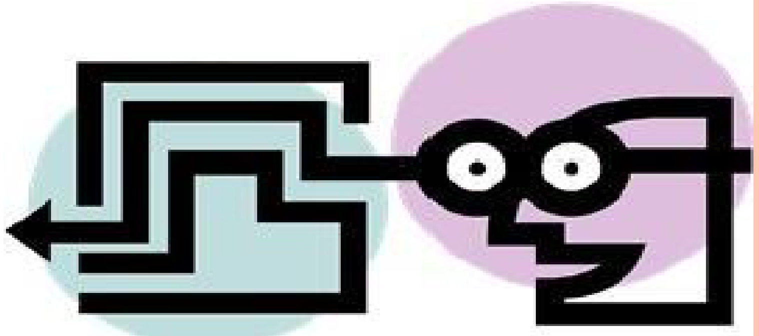


Note: Block 20 is not strictly necessary.

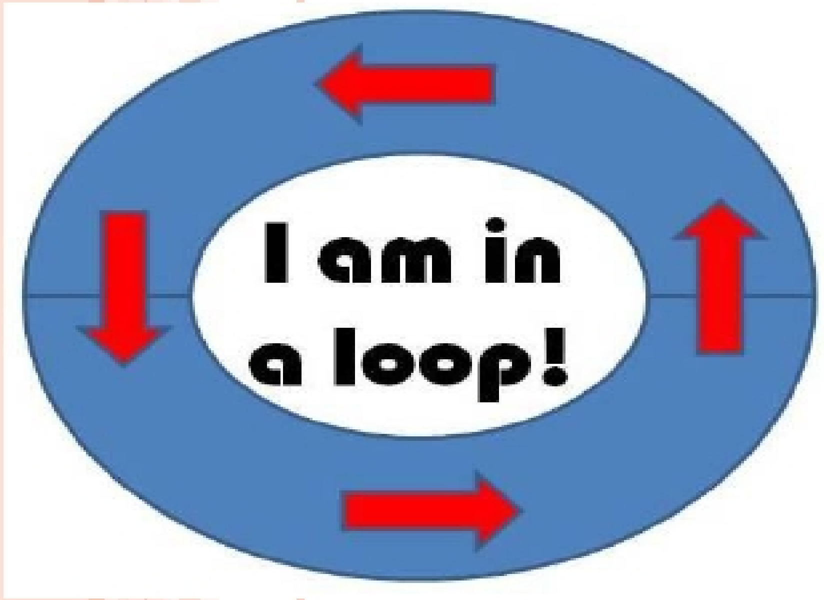
Vocabulary

design: Loosely stated, the art of creating something that does not exist.

engineering: The use of science and mathematics to solve problems to improve the world around us.



Using Waits, Loops and Switches



Waits, Loops and Switches Pre-Quiz

1. In programming, what is a loop? When is a loop useful?

2. How can you control the duration for which a loop repeats?

3. In programming, what is a switch?

Waits, Loops and Switches Pre-Quiz

Answers

1. In programming, what is a loop? When is a loop useful?

   A loop is an operator that allows us to repeat a set of commands indefinitely. Loops are useful when you need to repeat a set of commands multiple times.

2. How can you control the duration for which a loop repeats?

   You can control how many times a loop repeats by clicking the dropdown arrow next to "Control" on the loop block and setting it to forever, time, sensor, count or logic.

3. In programming, what is a switch?

   In programming, a switch is an object that gives different commands depending on the state it is in.

Wait Block Activity

Objective: Combine wait blocks, loops and switches to perform a task

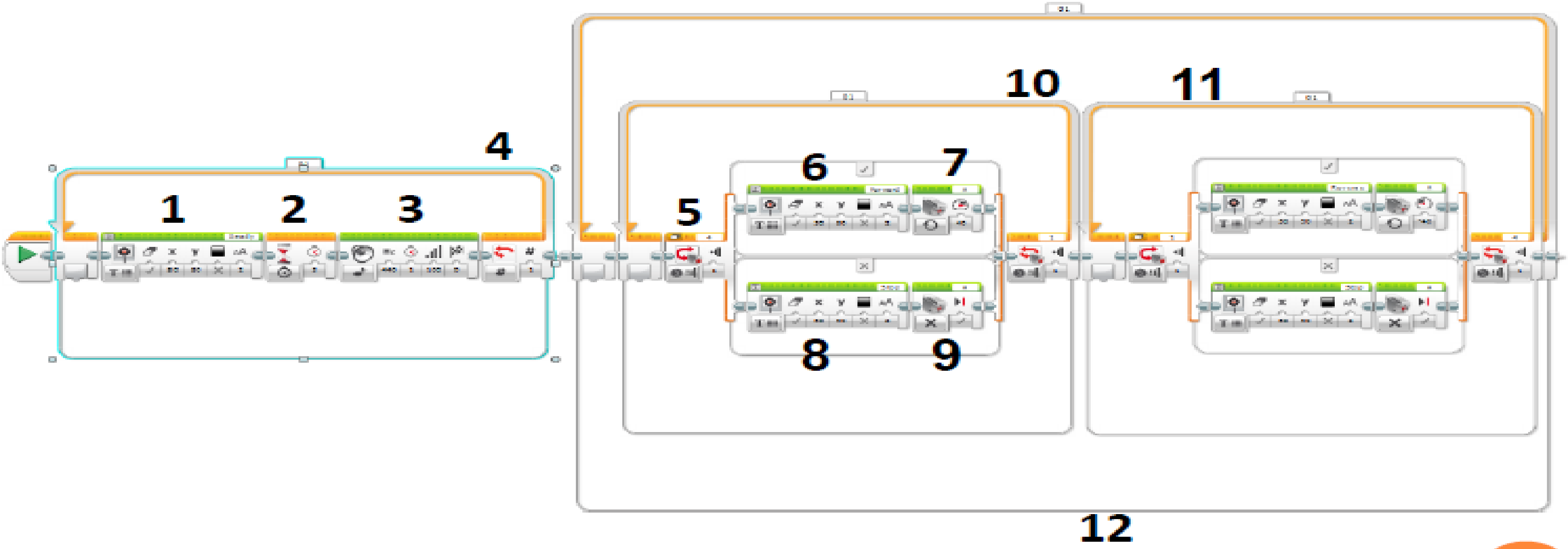
Do This: First, attach 2 touch sensors (name them A and B)

Then program the robot to perform the following task:

- Stay at rest and display "Ready" on the screen.
- Move left and display "Forward" whenever touch sensor 4 is pressed.
- Move right and display "Reverse" whenever touch sensor 1 is pressed
- Stay at rest and display "Stop" when no buttons are pressed.

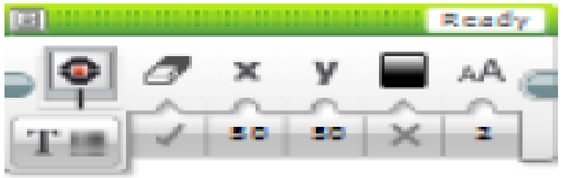
***Hint: Click on the icon, drag the block into the program, and select "Text" from the dropdown menu to display text on screen.

Wait Block Activity Solution



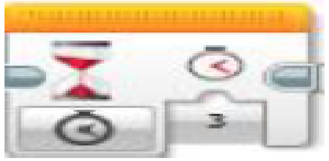
Wait Block Activity Solution

1

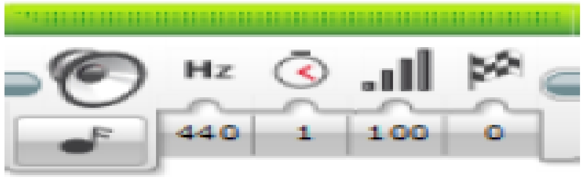


various settings

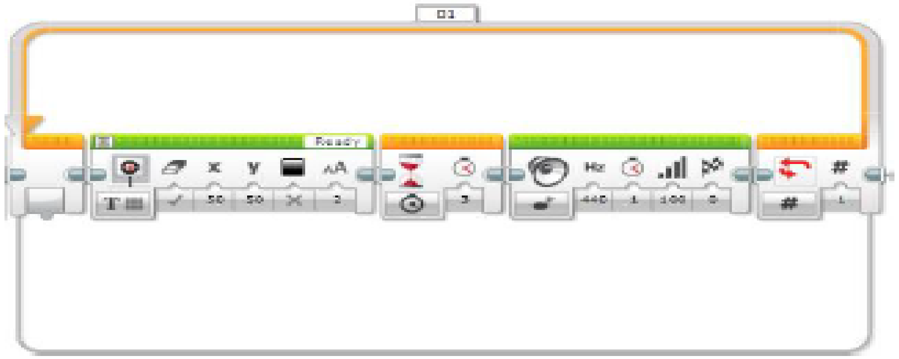
2



3



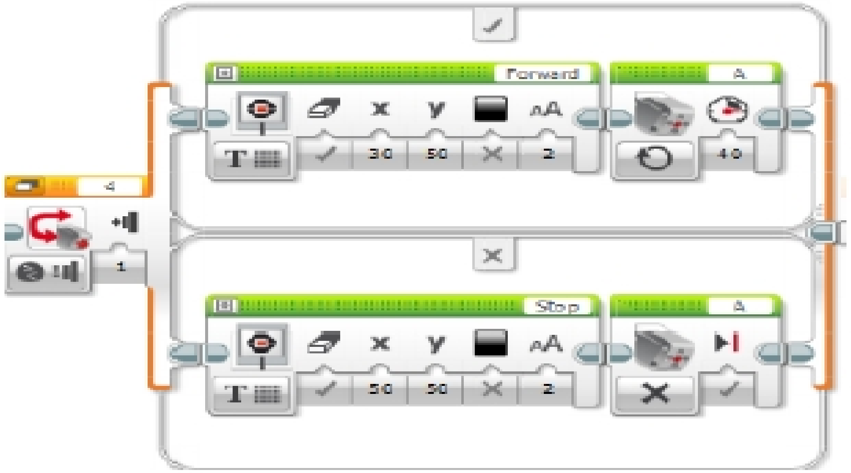
4



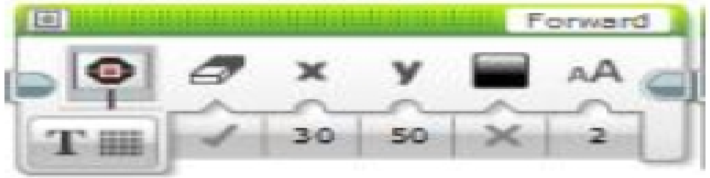
Wait Block Activity Solution

various settings (continued)

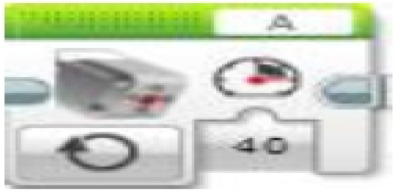
5



6

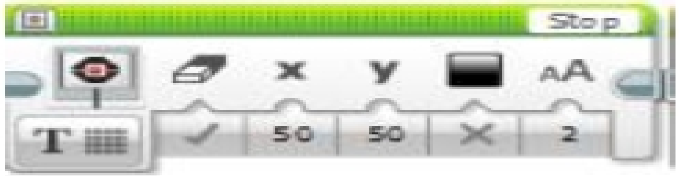


7



Wait Block Activity Solution various settings (continued)

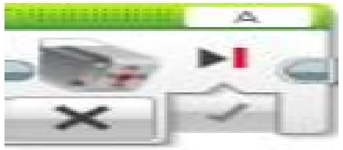
8



Stop

50 50 2

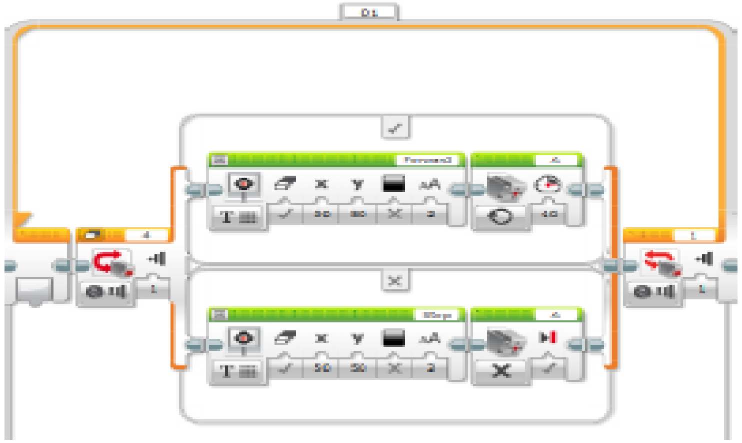
9



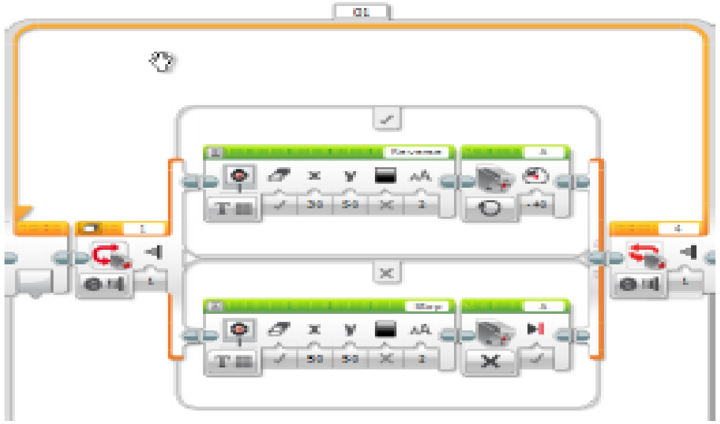
A

Wait Block Activity Solution various settings (continued)

10

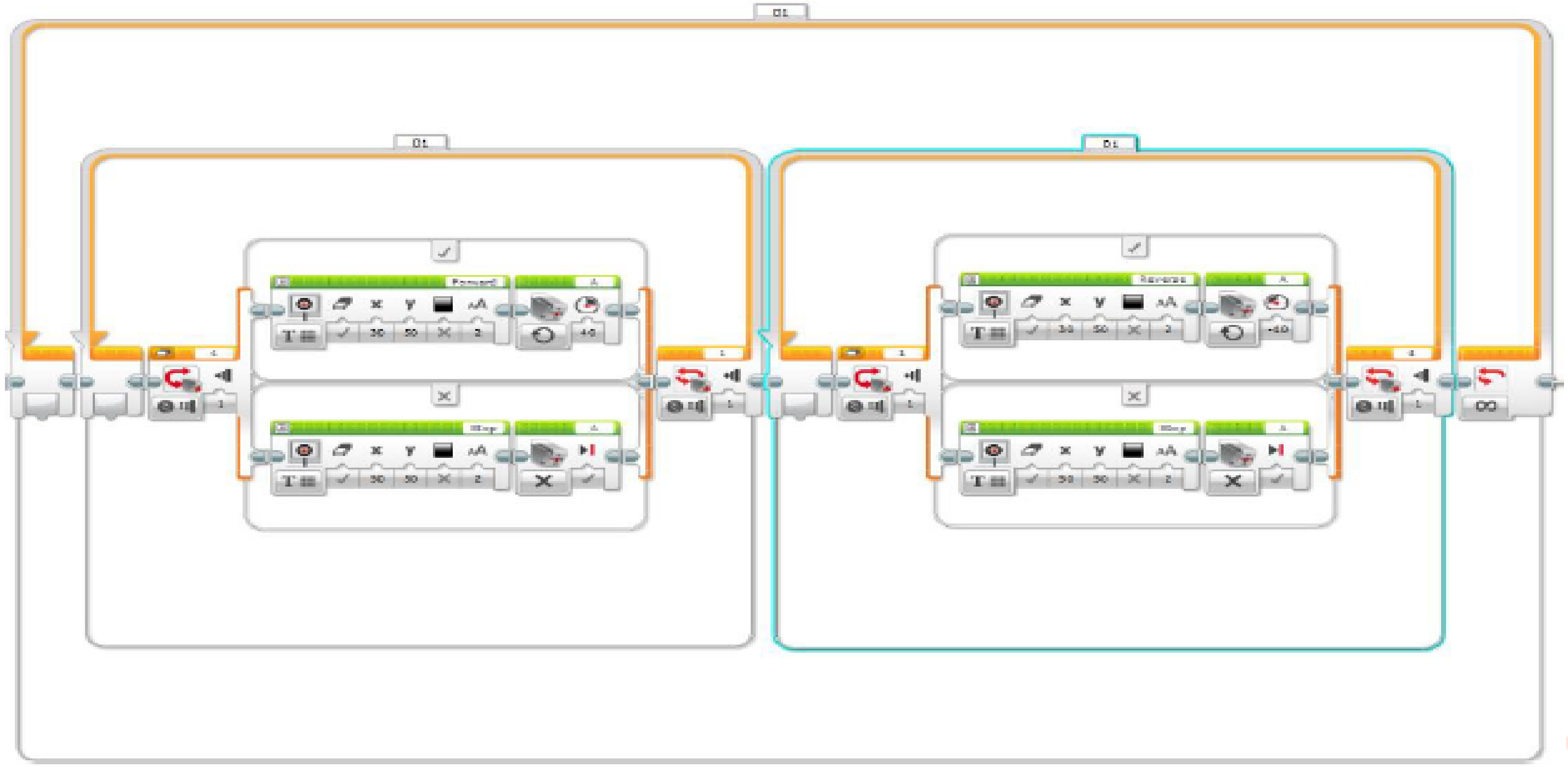


11



Wait Block Activity Solution various settings (continued)

12



Waits, Loops and Switches Post-Quiz

1. In programming, what is a loop? When is a loop useful?

2. How can you control the duration for which a loop repeats?

3. In programming, what is a switch?

Waits, Loops and Switches Post-Quiz

Answers

1. In programming, what is a loop? When is a loop useful?

   A loop is an operator that allows us to repeat a set of commands indefinitely. Loops are useful when you need to repeat a set of commands multiple times.

2. How can you control the duration for which a loop repeats?

   You can control how many times a loop repeats by clicking the dropdown arrow next to "Control" on the loop block and setting it to forever, time, sensor, count or logic.

3. In programming, what is a switch?

   In programming, a switch is an object that gives different commands depending on the state it is in.

Vocabulary

brainstorming: Thinking of ideas as a group.

iteration: Doing something again, especially with the intent to make improvements.

loop: An operator that repeats a set of commands.

switch: In programming, a switch is an object that gives different commands, depending on the state it is in.

Why Use Arrays?

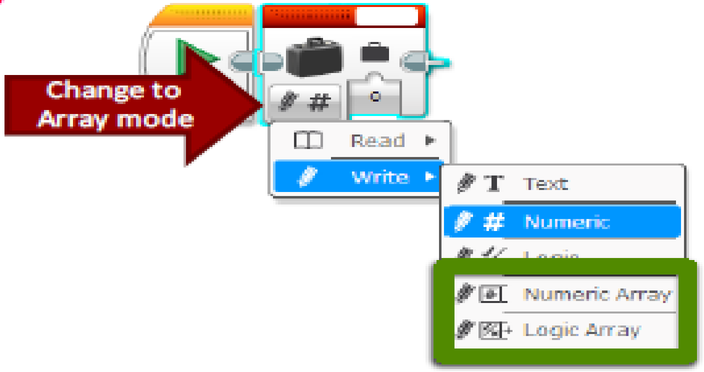
1. Simplify programs by storing multiple related values in a single variable
2. Can be used with loops to make compact and useful programs
3. Are useful for making a custom calibration program (see NXT Light Sensor in EV3 on our contributed lessons tab)

Arrays

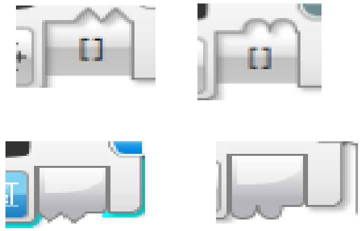
- What is an array?
  - An array is a variable that holds multiple values
- There are two types of arrays:
  - Numeric Array (Holds a set of numbers … 1,2,3,10,55)
  - Logic Array (Holds a set of logic … True, True, False)
- They can be used as either Inputs or Outputs so you can either….
  - Write – put a value(s) into the array
  - Read – get the value(s) from the array out

Array Blocks: Quick Guide

Modes



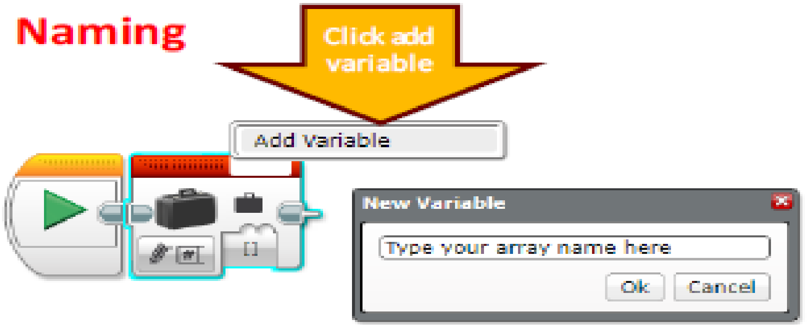
Key



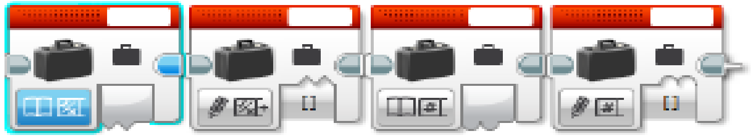
Write (Inputs) have 2 bumps up

Read (Outputs) have 2 bumps down

Naming



Quiz

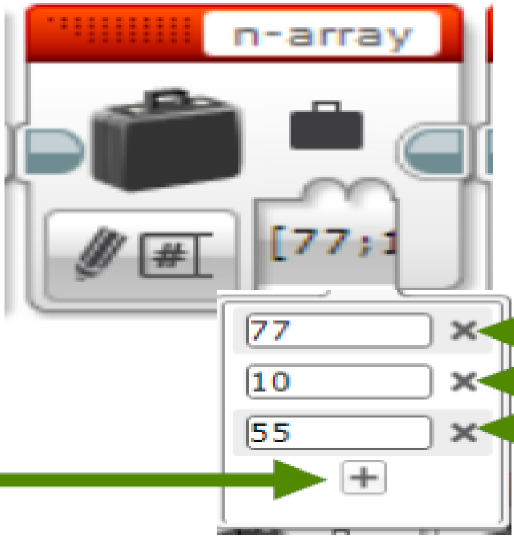


Read logic array | Write logic array | Read numeric array | Write numeric array

Identify if the variables are Inputs/Outputs and if they are Numeric/ Logic

Array Indexes

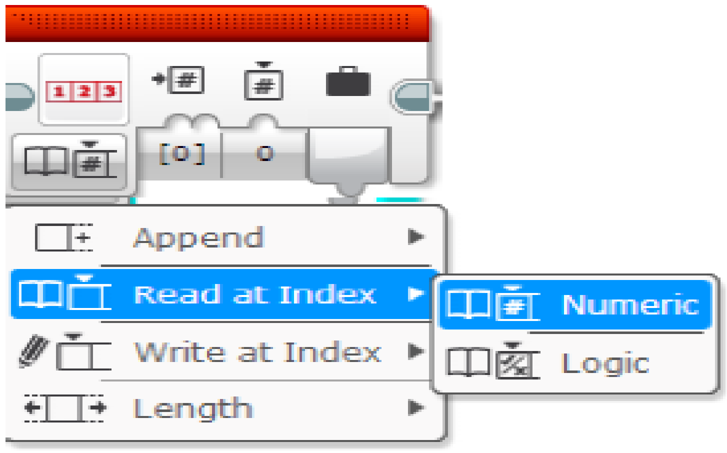
- Each value in an array is assigned an index
- The first value would be at index 0
- Logic arrays would store True/False instead of numbers
- To add a value to an array click the plus +
  - This adds an entry at the next index value (i.e. index 3)



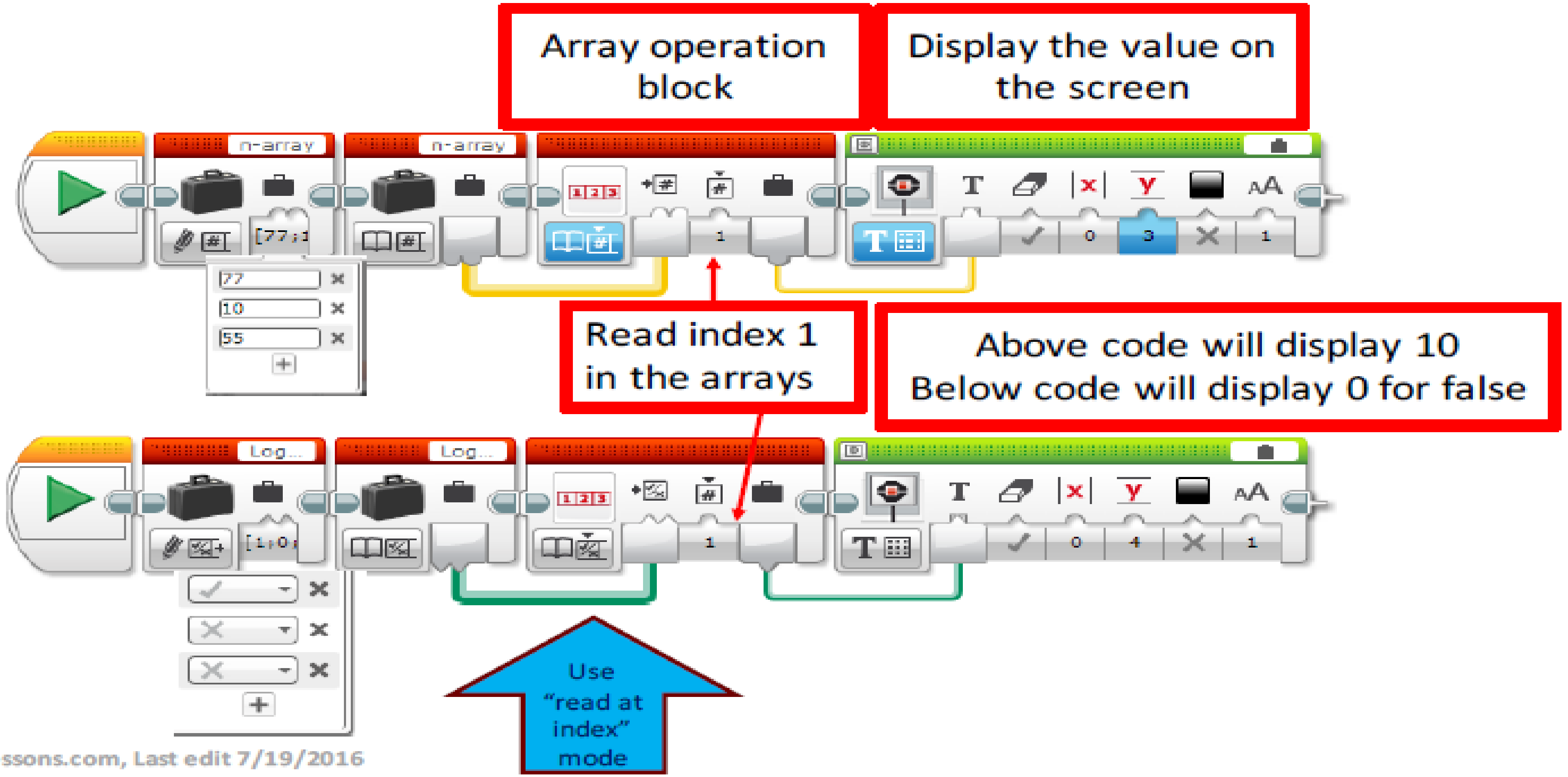
These are values for index 0,1,2

Block: Array Operations

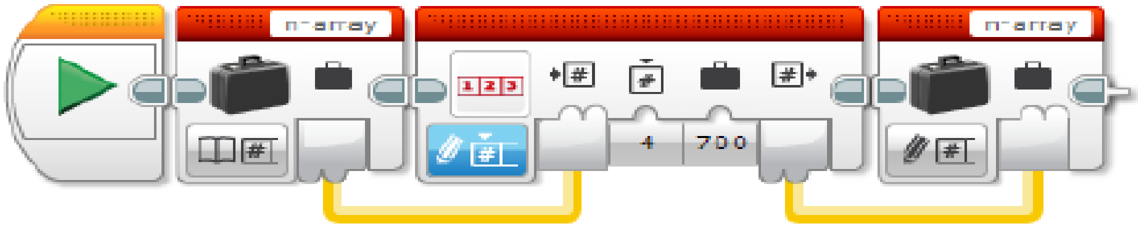
- This block is used to read or write to Logic or Numeric arrays
- Different modes:
  - Append: Add a new entry after the last array index
  - Read at index: Reads the value at a certain index
  - Write at Index: Write a new value to a certain array index
  - Length: How many entries are in the array
- Both write and append output an array → you will need to write this array back to the variable if you wish to update the stored array (see write/append slides)



How do you use Arrays (Reading)?


Array operation block
Display the value on the screen
n-array
n-array
[77;1
77
10
55
1
3
0
1
Read index 1 in the arrays
Above code will display 10
Below code will display 0 for false
Log...
Log...
[1;0;
1
0
4
1
Use "read at index" mode

How do you use Arrays (Writing)?

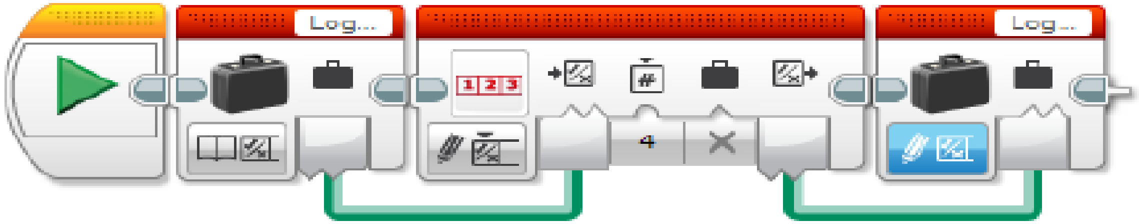


This will write 700 to array at index 4

Read the array you want to write to

Use array operations to write a value to a certain index

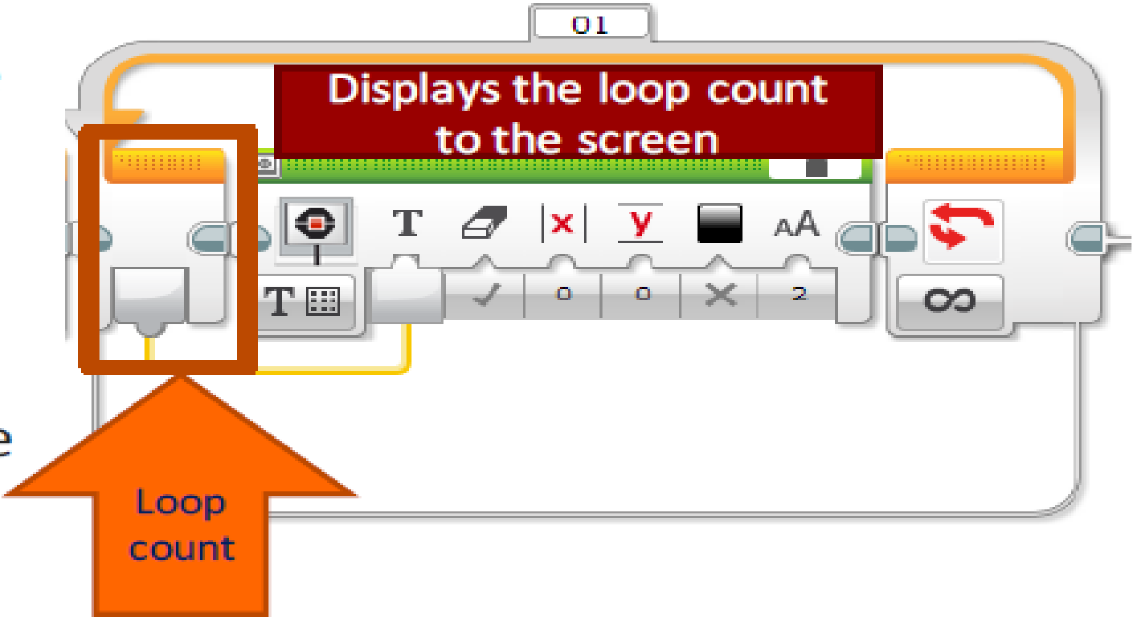
Write the output back to the array



This will write False to array at index 4

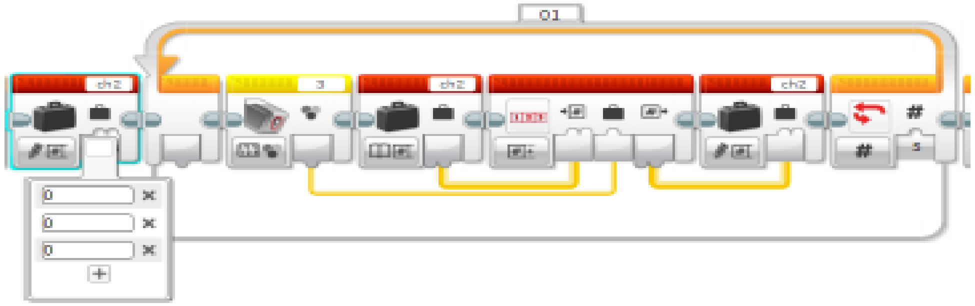
Block Setting: Loop Count

- The loop count outputs the amount of times the blocks inside the loop have played.
- This is useful to create a program that runs different code every time it goes in the loop
- It is also useful for computing on each item of an array



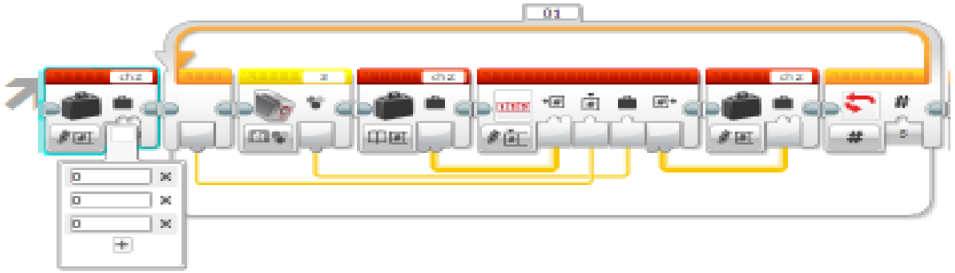
Note: Append vs. Write

- Append adds entries to the end of an array (i.e. creates a new index value)



- This code produces an array with 8 entries (three 0's followed by 5 light readings)

- Write overwrites the entry at the chosen index

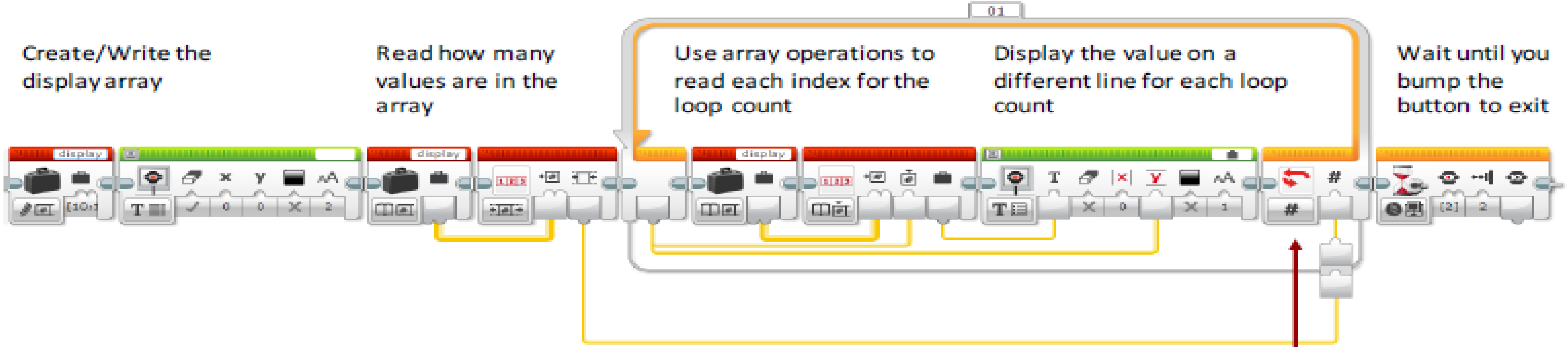


- This code produces an array with 5 entries (just 5 light readings)

Challenge 1

- Make a program that displays all the entries of an array. Display each index on a different line. You can use only one display block.
- Tips: You will need to use loops, loop count, array block, array operations

Challenge 1 Solution



Create/Write the display array

Read how many values are in the array

Use array operations to read each index for the loop count

Display the value on a different line for each loop count

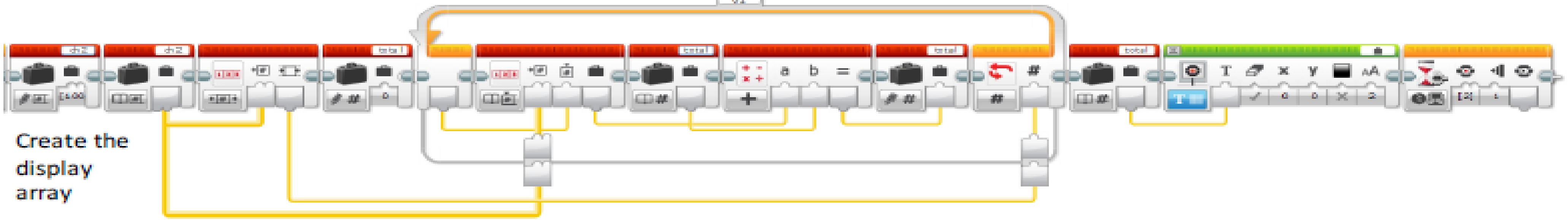
Wait until you bump the button to exit

End after all the indexes have been displayed

Challenge 2

- Make a program that adds up all the entries of an array. Display the sum.
- Tips: You will need to use loops, loop count, array block, array operations

Challenge 2 Solution



Read how many values are in the array

Read the index based on the loop count

Add the array value to the sum of the past values

Display to the screen

Create the display array

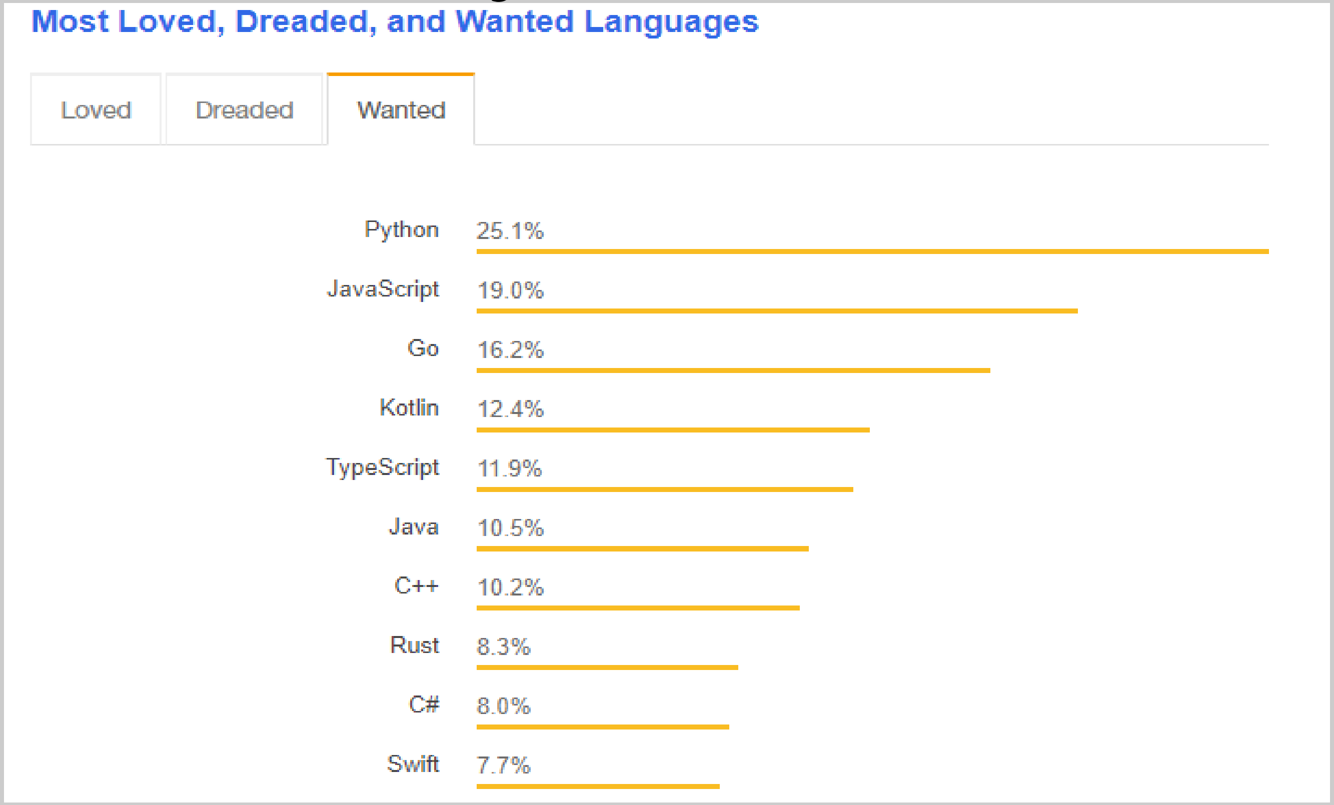
Next Steps

- Here are some fun things to try:
  1. Make a program to compute the average value in an array
  2. Make a program that always saves the last 4 light sensor readings in an array
  3. Create an array that stores calibration values for each sensor port

EV3 Python

The Python programming language is one of the most modern, most popular, most easy-to-learn and most powerful programming languages in the world, so what could be more natural for an owner of the EV3 robot than to want to learn how to program the EV3 robot using the Python programming language? The normal way to that would be to run the EV3dev operating system in the EV3 and run EV3 Python within the EV3dev environment.

Per the Stack Overflow Developer Survey 2018:



Python	25.1%
JavaScript	19.0%
Go	16.2%
Kotlin	12.4%
TypeScript	11.9%
Java	10.5%
C++	10.2%
Rust	8.3%
C#	8.0%
Swift	7.7%

Python is

Interactive

Interpreted

Modular

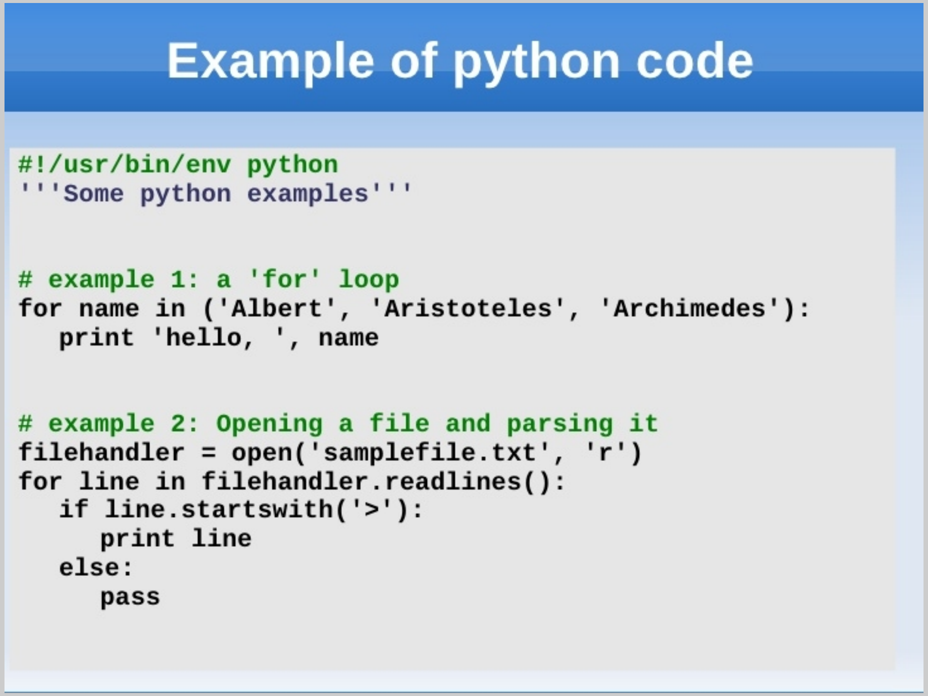
Dynamic

Object-oriented

Portable

High level

Extensible in C++ & C



```
#!/usr/bin/env python
'''Some python examples'''


# example 1: a 'for' loop
for name in ('Albert', 'Aristoteles', 'Archimedes'):
   print 'hello, ', name


# example 2: Opening a file and parsing it
filehandler = open('samplefile.txt', 'r')
for line in filehandler.readlines():
   if line.startswith('>'):
      print line
   else:
      pass
```

How to getting started with Python + EV3

https://www.youtube.com/watch?v=BDr3lDmhkOQ&list=PL7hndBcWv-umf5tdIp0lJfPuU0X9LE97a&index=1

https://www.ev3dev.org/docs/tutorials/

Talk at PyCon(PyCon Australia is the national conference for users of the Python Programming Language)

Programming Lego Mindstorms robots with Pytho
n

Great tutorials in youtube

https://www.youtube.com/watch?v=elNZn-LTLXA&list=PLFd4sjeVCkV8PS841FhnkZzrvacBh5yz0&index=1

https://www.youtube.com/watch?v=-qEJtLvX2Hw&list=PL6leF0cJVwDqO0roBmdFa6OlblSkrQ0Uw

https://www.youtube.com/watch?v=IuHaIE-auLQ&pbjreload=10

Useful other links

ev3lessons

https://www.teachengineering.org/curriculum

https://www.lego.com/en-us/mindstorms

https://www.lego.com/en-SG/service/help-topics/bricks-and-sets/themes/mindstorms