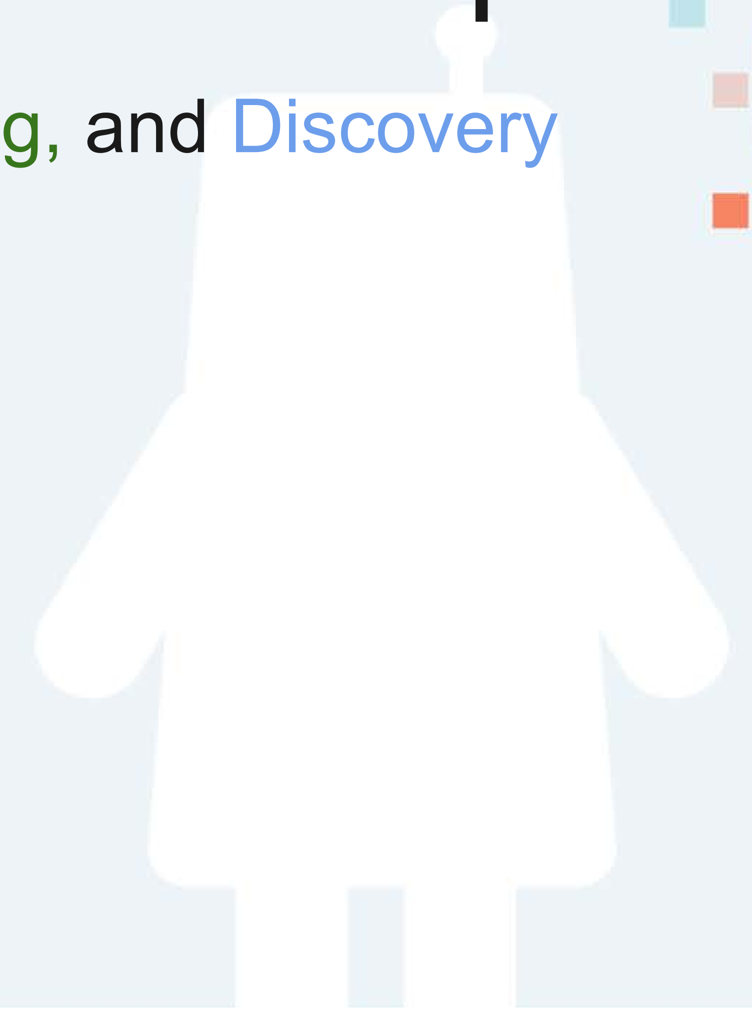
KOOV

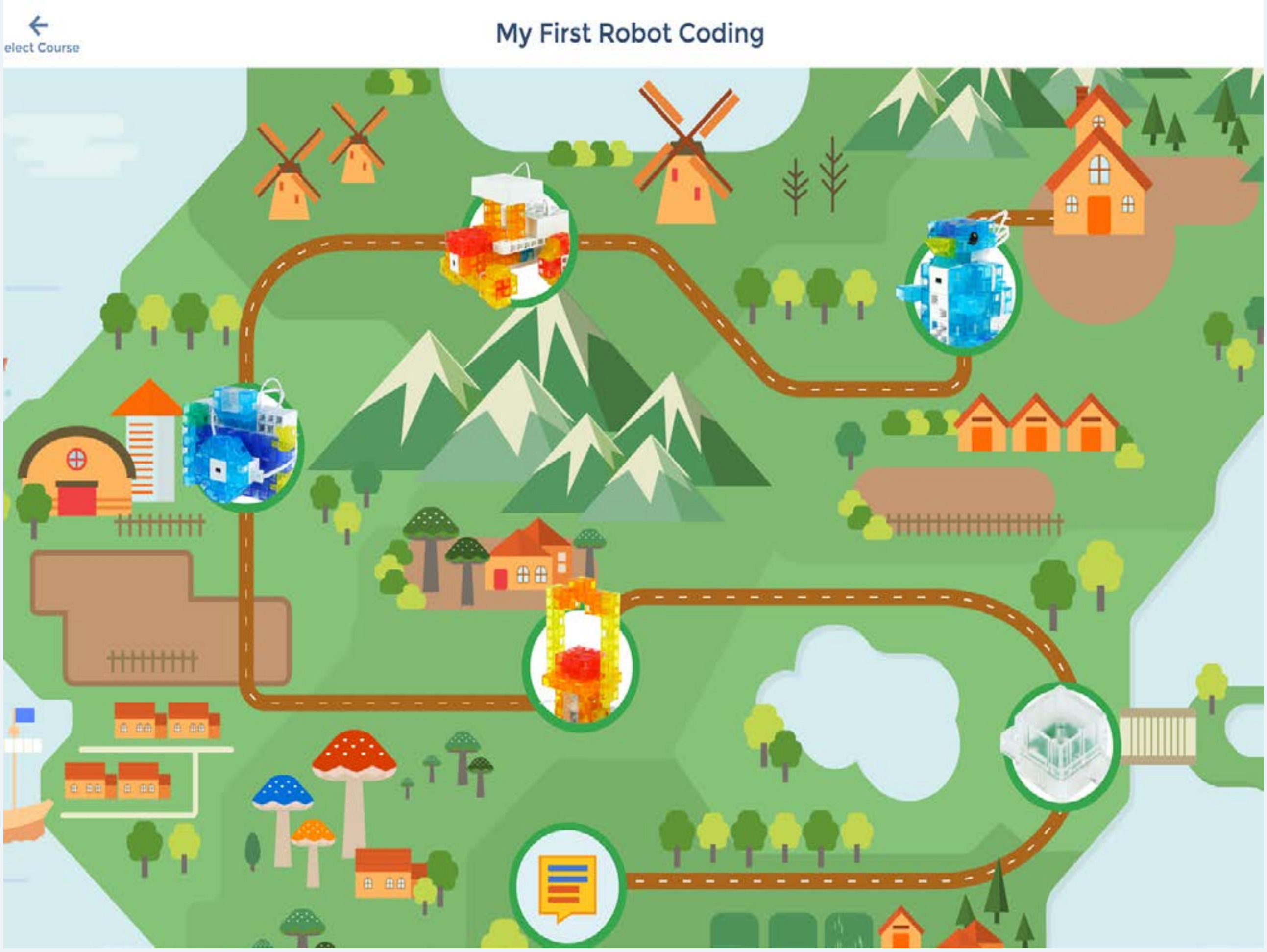
Robotics Camp

Fun, Learning, and Discovery



Fun Road Ahead

Welcome to the KOOV Robotics Camp. At this camp, you are going to put on the hat of a student robotic engineer and learn by building. You will build and program several “robots” each day,following instructions in this curriculum and from My First Robot Coding Map. Then you will be given a Design Challenge to brainstorm for better design. Each day will be filled with learning, fun, and discovery.



Week In A Glance

Day 1: KOOV Education Suite

Learn about Robotics and KOOV

Day 2: Blink and Beep!

Coding with LEDs and Buzzer

Day 3: Light and Camera

Detect Light and Distance

Day 4: Turn that Wheels

DC Motor and Robot Car

Day 5: Dancing Feet

Servo Motor and Penguin

DAY 1

KOOV Educational Suite

Day 1

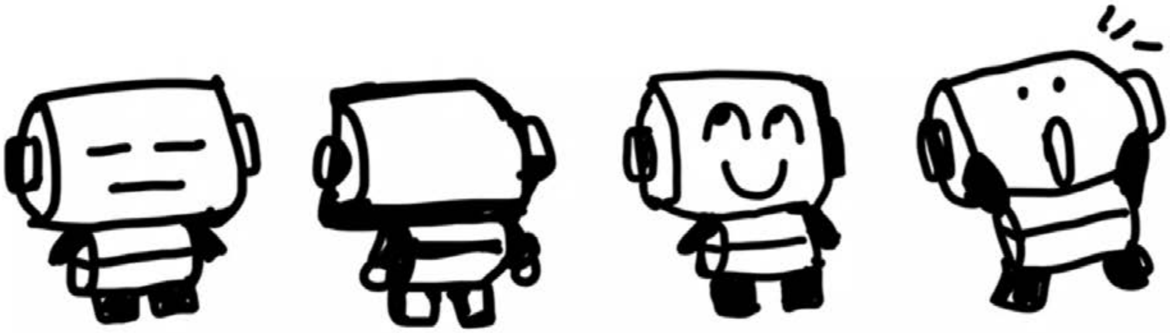
I Introduction	Robots
E Electronics	KOOV, LEDs
C Coding Concepts	Intro to Coding
M Map	KOOV Code Editor & Coding & LEDs

Intro

Robots

E

Robots



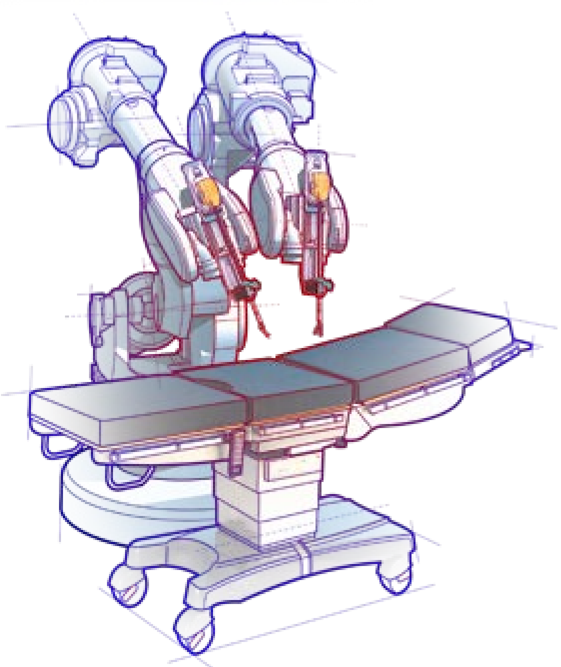
Have you ever seen a robot? Can you point out the robots around you? Have you ever wondered what makes a robot move?

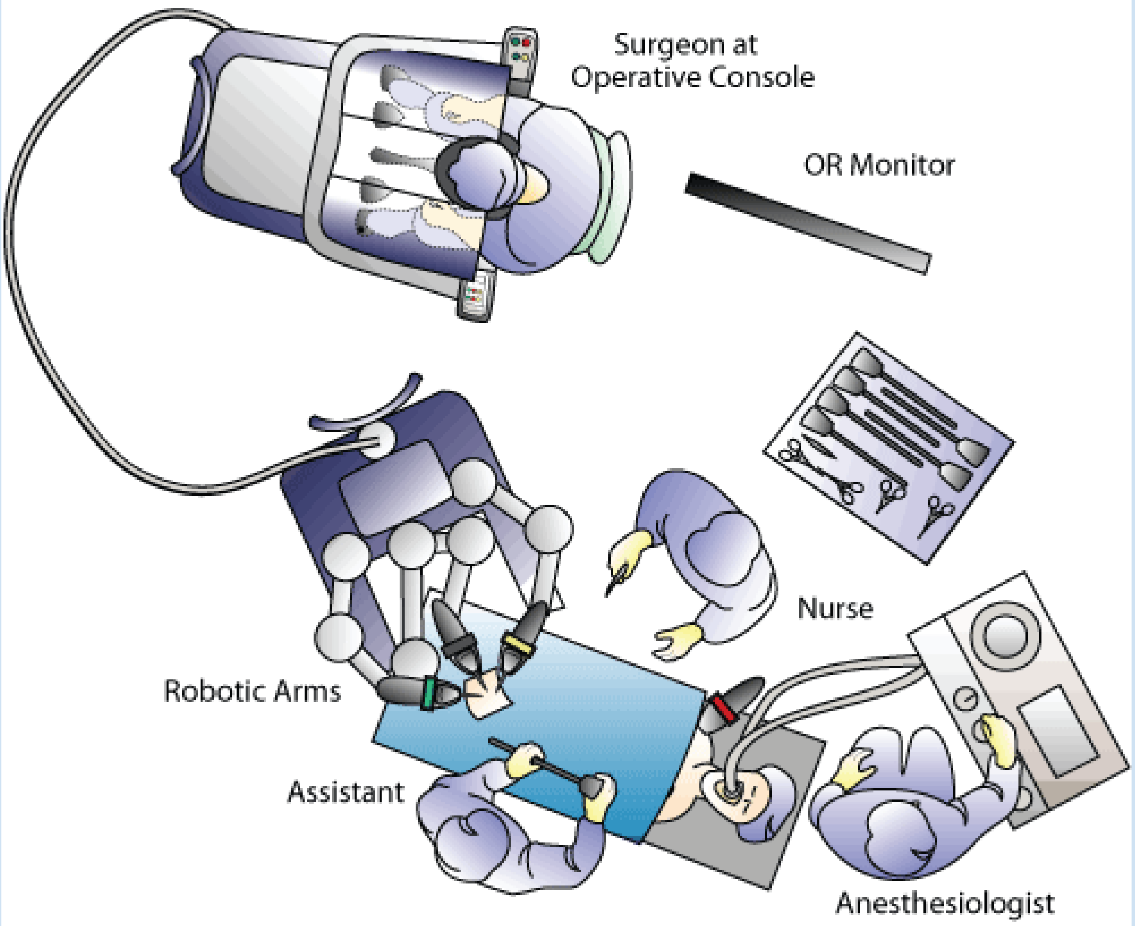
Researchers and companies created many robots to improve our lives. Can you name some?

Let’s take a look at some examples of these robots.

Robots that Help Us

Surgical Robots



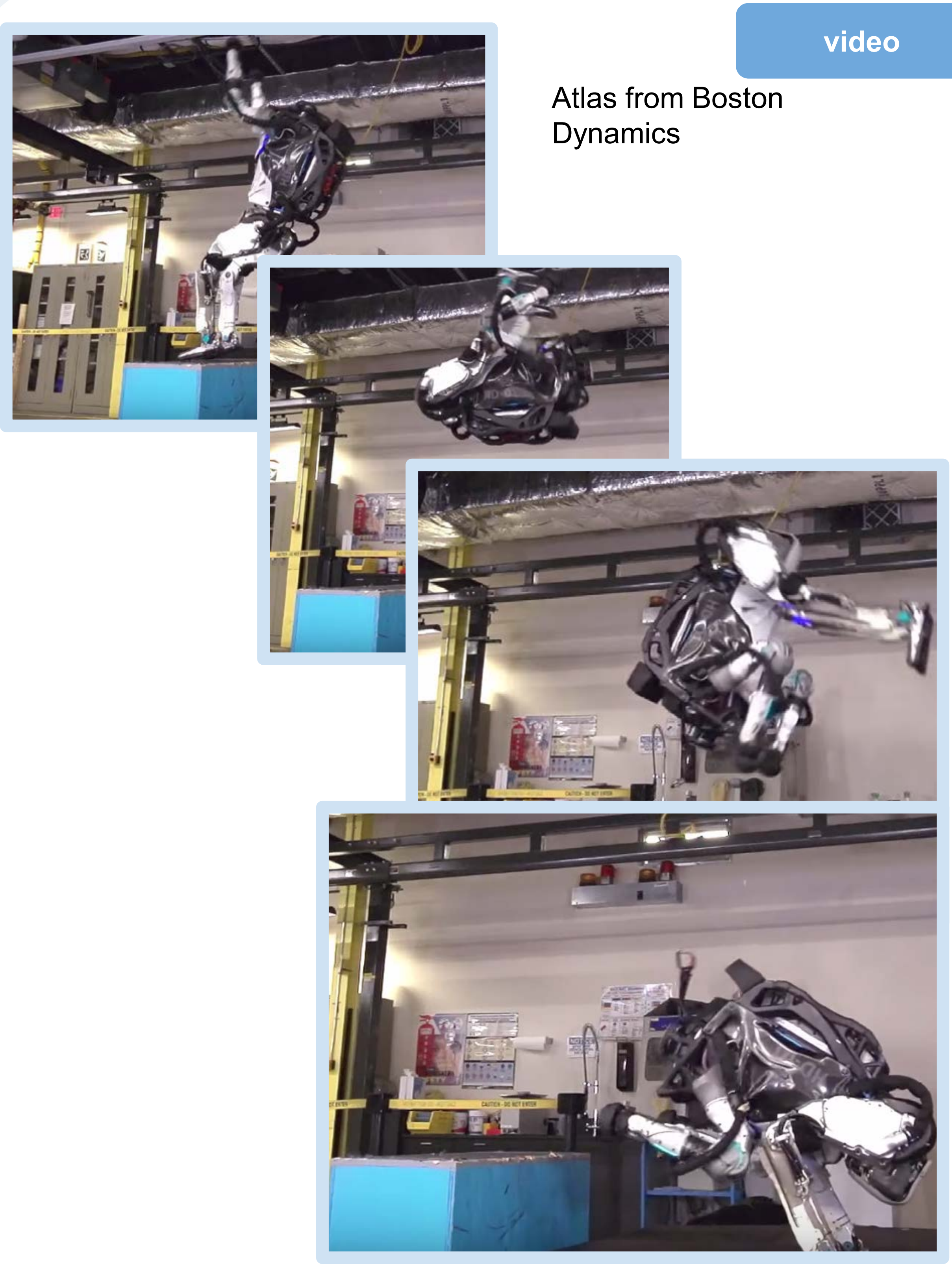


E

(Super) Human Like Robots

video

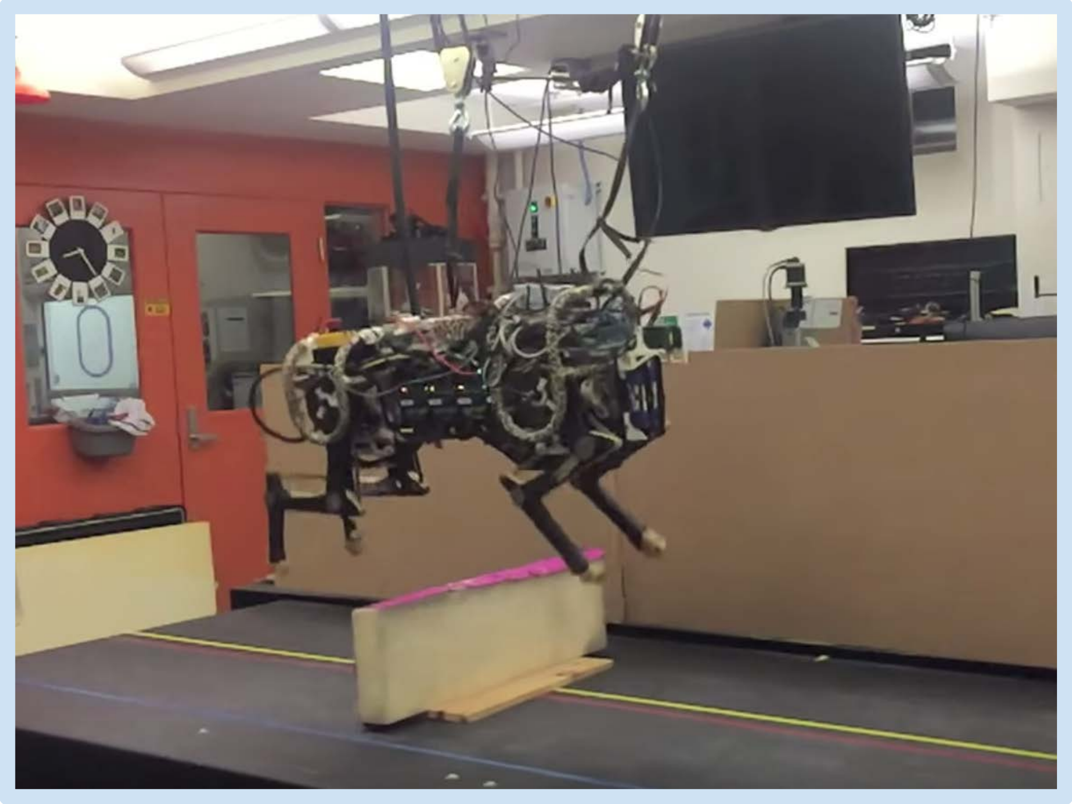
Atlas from Boston Dynamics



E

(Fast) Animals Robots

video



Robotic Cheetah from MIT

video

BionicKangaroo from Festo

KOOV

Electronics

Intro to KOOV Kit

KOOV is for You

If you are curious about how these robots work, and want to start making your own robots, then KOOV Robotics and Coding Kit is for you.

KOOV Robotic and Coding Kit includes a small computer, blocks, electronic devices, and a KOOV coding editor to let you write code to control your robot. Moreover, KOOV comes with rich learning material to teach you, step-by-step, how to build and control a robot.

KOOV Robot

A **KOOV robot** is a robot built with KOOV Core (computer), blocks, electronic devices, and cables.

The **KOOV Core** is a mini computer that can follow step-by-step instructions. You can program the KOOV core by uploading code to it.

The **blocks** are non-electronic blocks for building the body of a robot.

The **electrical devices** help a robot interact with its surrounding.

The **cables** are wires that connect KOOV electrical devices to the computer.

KOOV Core

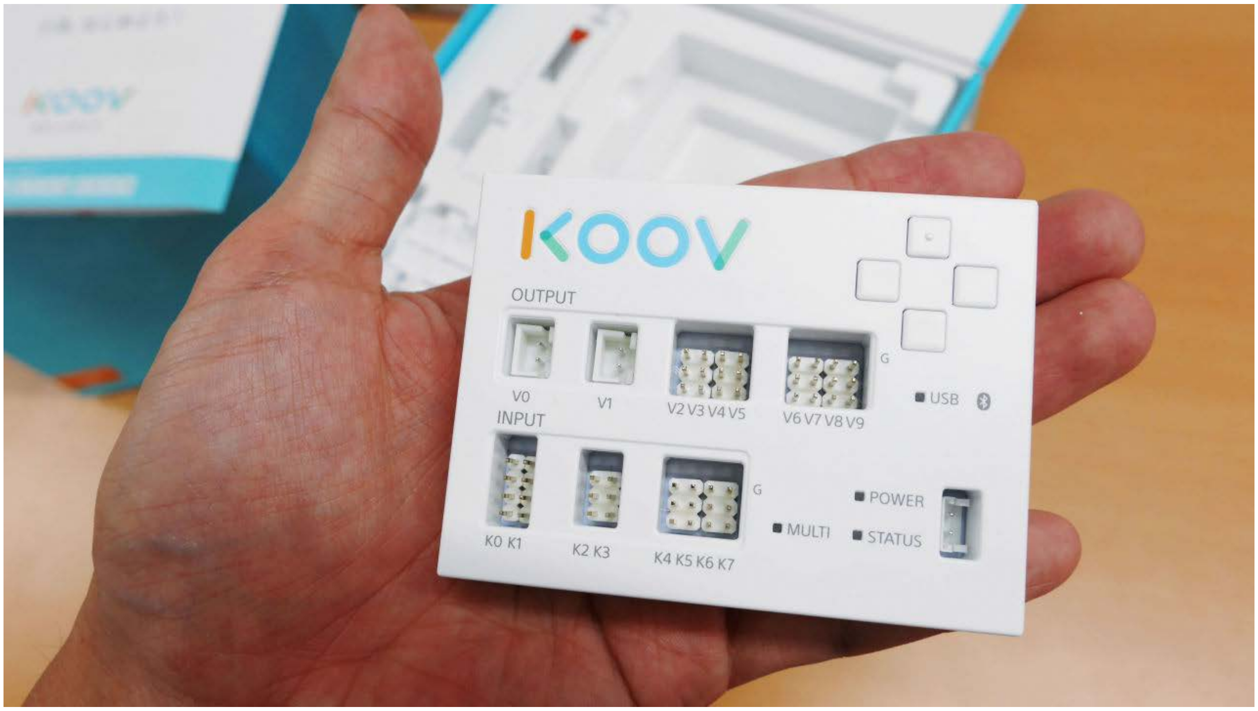
animation

KOOV Core is a small computer and it is the “brain” of a KOOV Robot.

It has input ports and output ports for the KOOV electronic devices, such as a LED.

It also has built-in Core Buttons and status indicators.

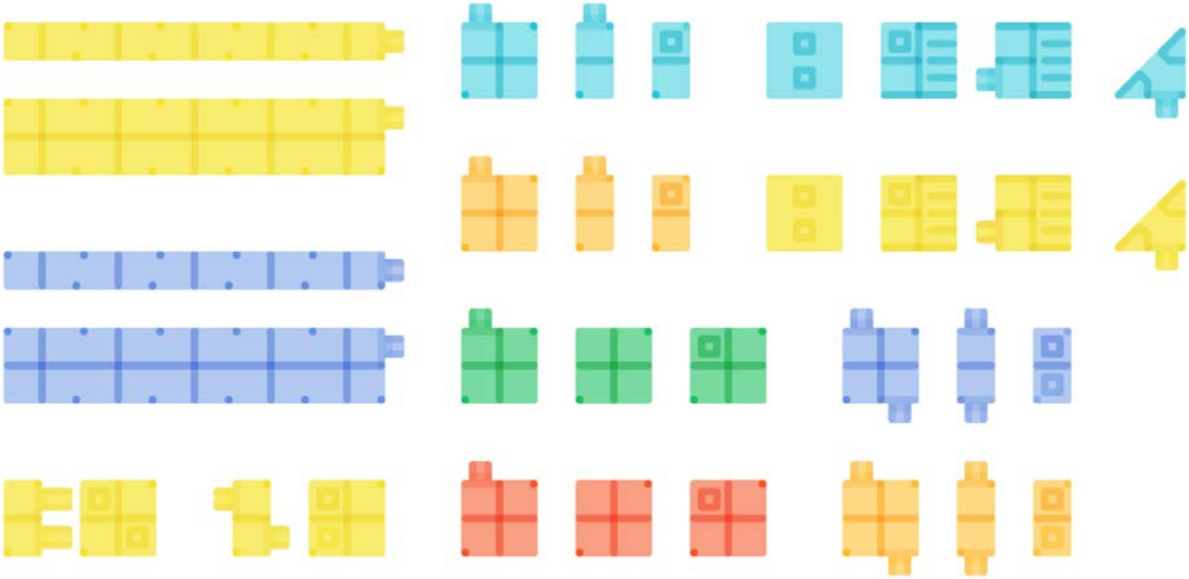
KOOV Core controls the KOOV electronic devices and you can program the KOOV Core using code. Such as turning on and off a LED.



E

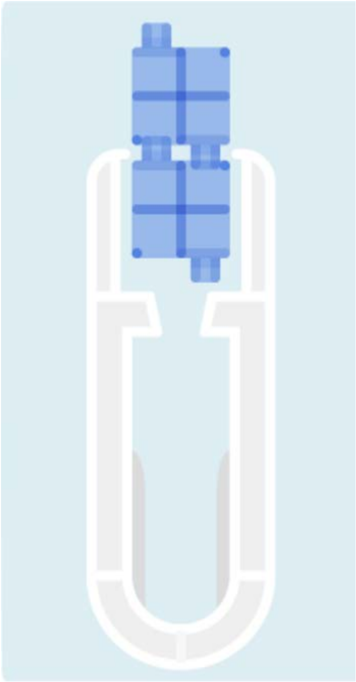
KOOV Blocks

KOOV blocks are transparent blocks that come in different shapes and colors.



KOOV blocks are like puzzle pieces and more fun and challenging to build. They can be connected from multiple sides.

Since the blocks interlock better and tighter, sometimes you would need to remove them using a Block Remover.



KOOV Buttons

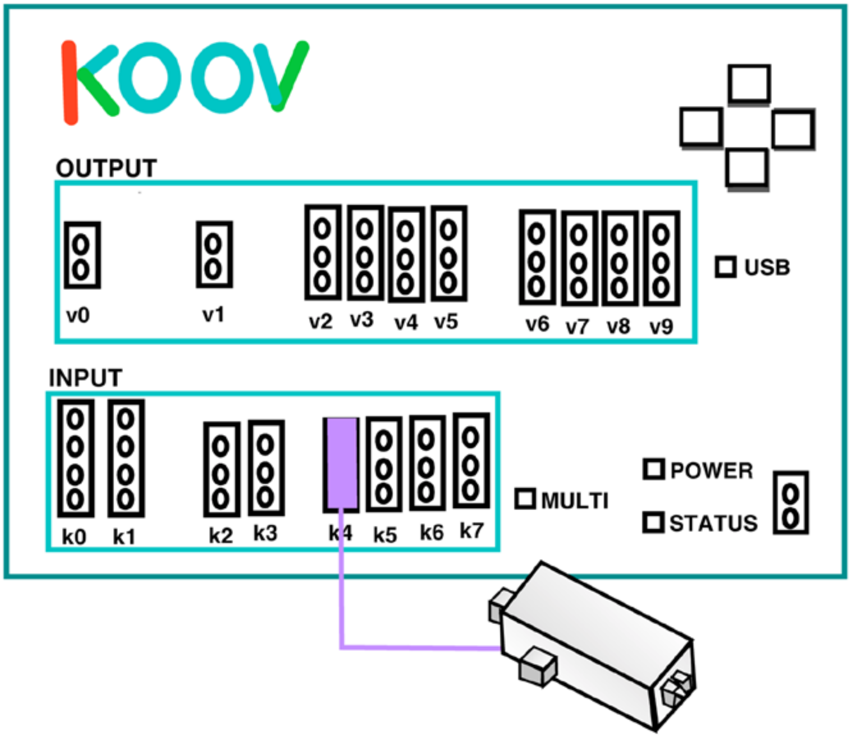
animation

KOOV Buttons are input devices that let you control your robot.A button can only be in ON or OFF state.

There are two different types of KOOV Buttons: the Core Buttons and the KOOV Push Switch.

The KOOV Core buttons are on the KOOV Core and include top, bottom, right, and left core button. The KOOV Push Switch is an electronic part that can be connected to KOOV Core by one of the cables.

Buttons can be used to control other devices. For example, you can check if a button is pressed, and if so, turn on LED, or sound a buzzer



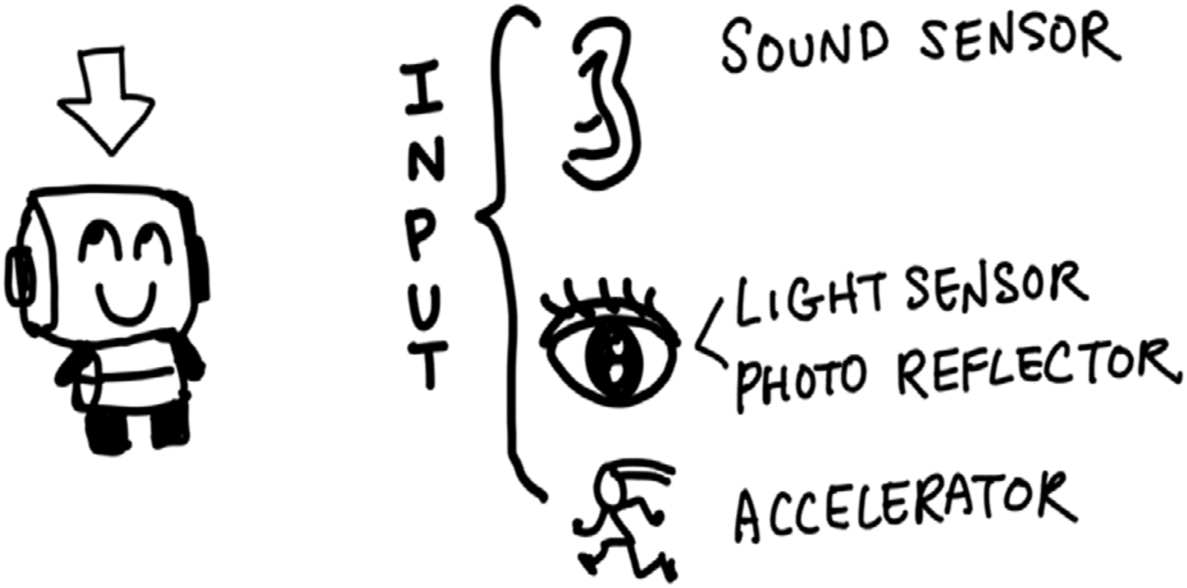
E

KOOV Sensors

animation

Just as we learn about our environment through our eyes, ears, and nose, KOOV robots learn about its environment through SENSORs. Sensors are input devices.

KOOV kit comes with various sensors to help a robot detect how bright it is, if there is anything in front of it, and how fast is it moving.

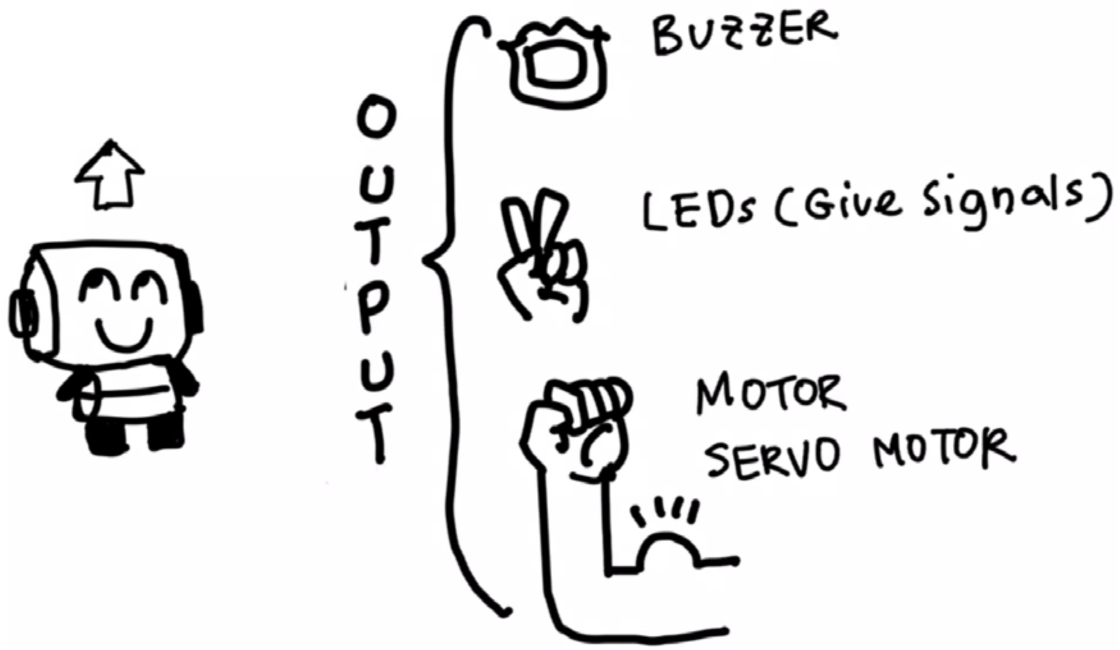


KOOV LEDs, Buzzer and Motors

animation

Besides sensors and buttons to interact with KOOV robot,
KOOV also comes with OUTPUT electrical devices for KOOV robot to interact with the world.

The OUTPUT devices include LEDs, buzzer, and motors. A robot can use LEDs to make signals, buzzer to create alarms, and motors to move or rotate.



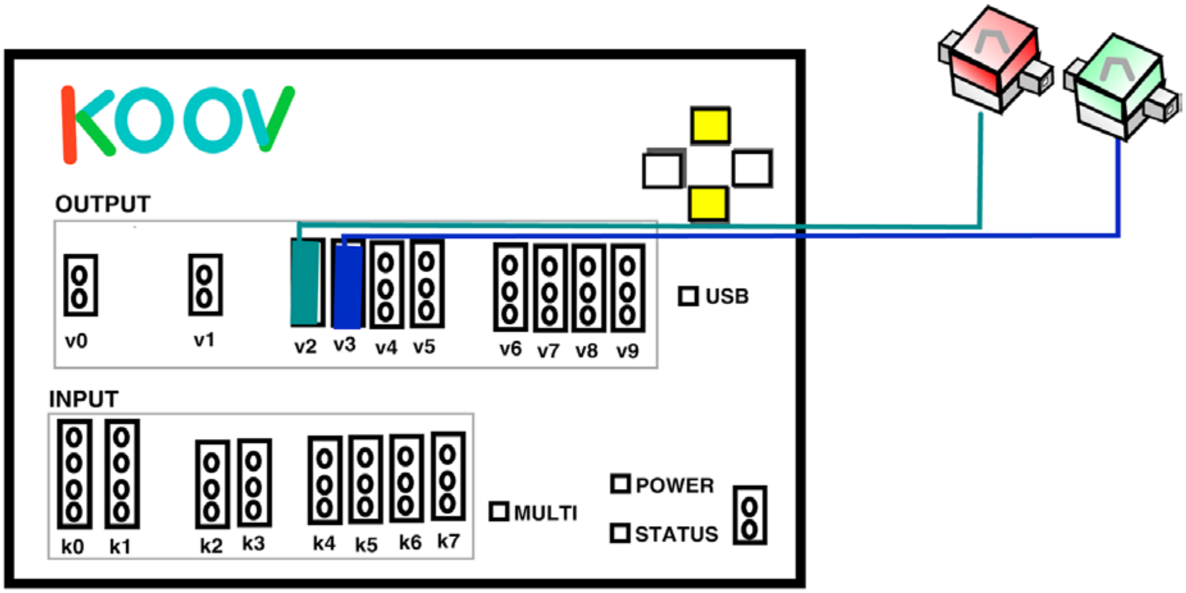
Electronics

LEDs

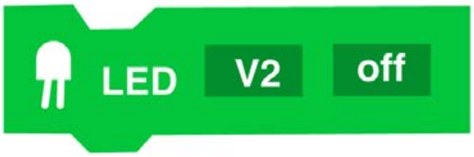
KOOV LEDs

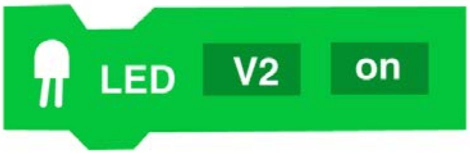
animation

KOOV LED is connected via cable and to KOOV Core. It comes with several colors.



When using LED with KOOV Core, you can turn on and off the LED by sending “LED On/Off” command to the KOOV Core, which then send electric signal supplies power to the LED and turn it on.

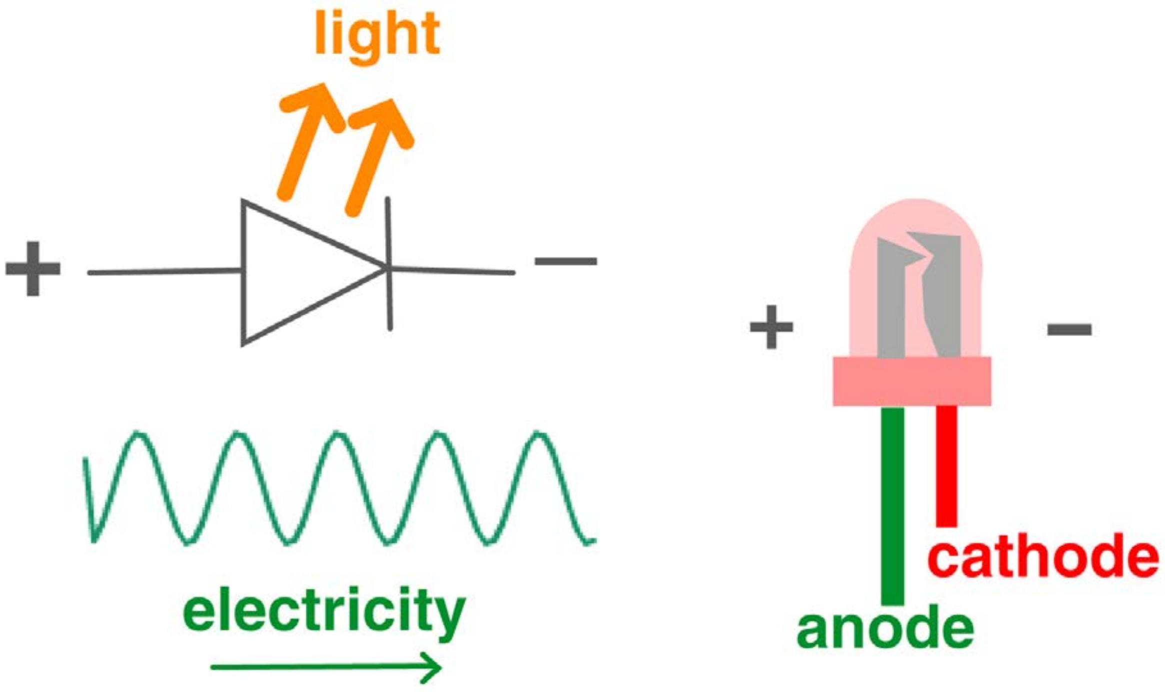




How a LED Works

animation

LED stands for Light-Emitting Diode (Di-ode). Light-Emitting means giving out light. Di means two and it means the two pins in LED.



The LED has two pins called anode and cathode.

Anode is the longer pin and connects to the positive voltage. Cathode the shorter pin and connects to the negative voltage. The electricity flows from anode to cathode.

As electricity flows through the LED, the electricity is changed to light energy, emitting light.

c
o
d
i
n
g

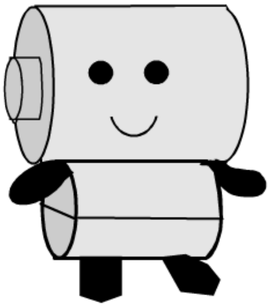
Intro to Coding

C

Coding = Give Instructions

animation

Coding is **to give step-by-step instruction** for computers to complete a task. Once instructions or codes are written, they are turned into a program that you can run on a computer.



move right 1 step

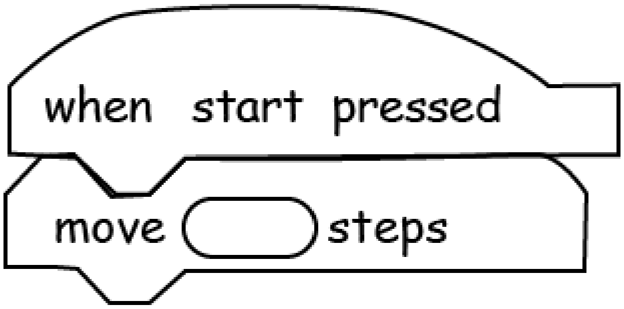
move right 2 steps

move left 3 steps

Moreover, just like humans speak in different languages, computer speak in Programming Languages. There are many different kinds of Programming Languages. We will learn to speak to a computer in a language called **block-based programming language**.

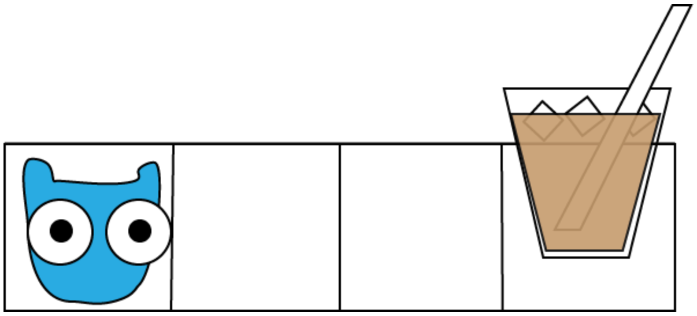
A Simple Program

To write codes in block-based programming language, you snap code blocks together. Let’s take a look at a very simple example to move a character. This code tells a robot to move when the program starts.

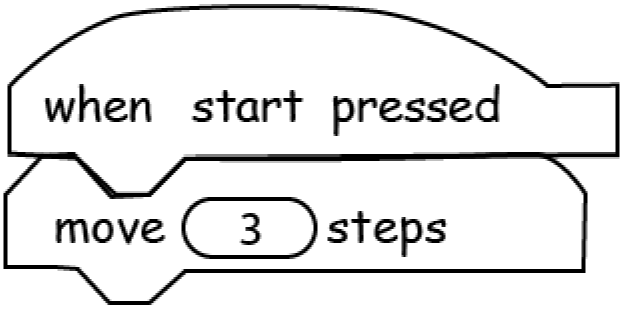


```
when start pressed
move (   ) steps
```

when KOOVy, an adorable cyber pet, is really thirsty, how would you direct it to get to the cold, refreshing iced tea?



Did you guess it right? The code would look like this:



```
when start pressed
move ( 3 ) steps
```

M

a

p

First Robotic Coding Map

□ KOOV Code Editor

□ Map: What is Coding

□ Robotic Coding: LEDs

KOOV Code Editor

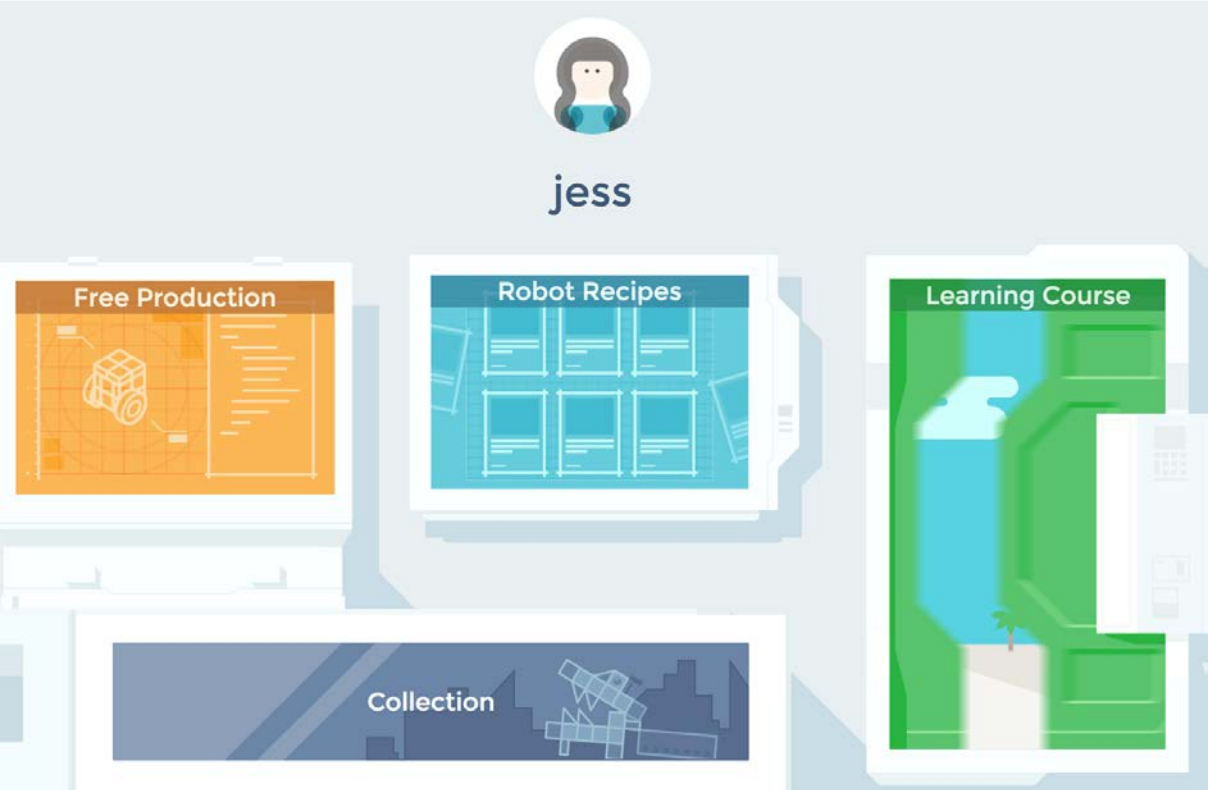
L

Start the KOOV Code Editor on your computer.

Use your provided login, follow the instruction on KOOV editor to update your account.

Then:

- Go to Home
- Select Free Production.
- At the Free Production Panel, click “Start”



Map: What is Coding

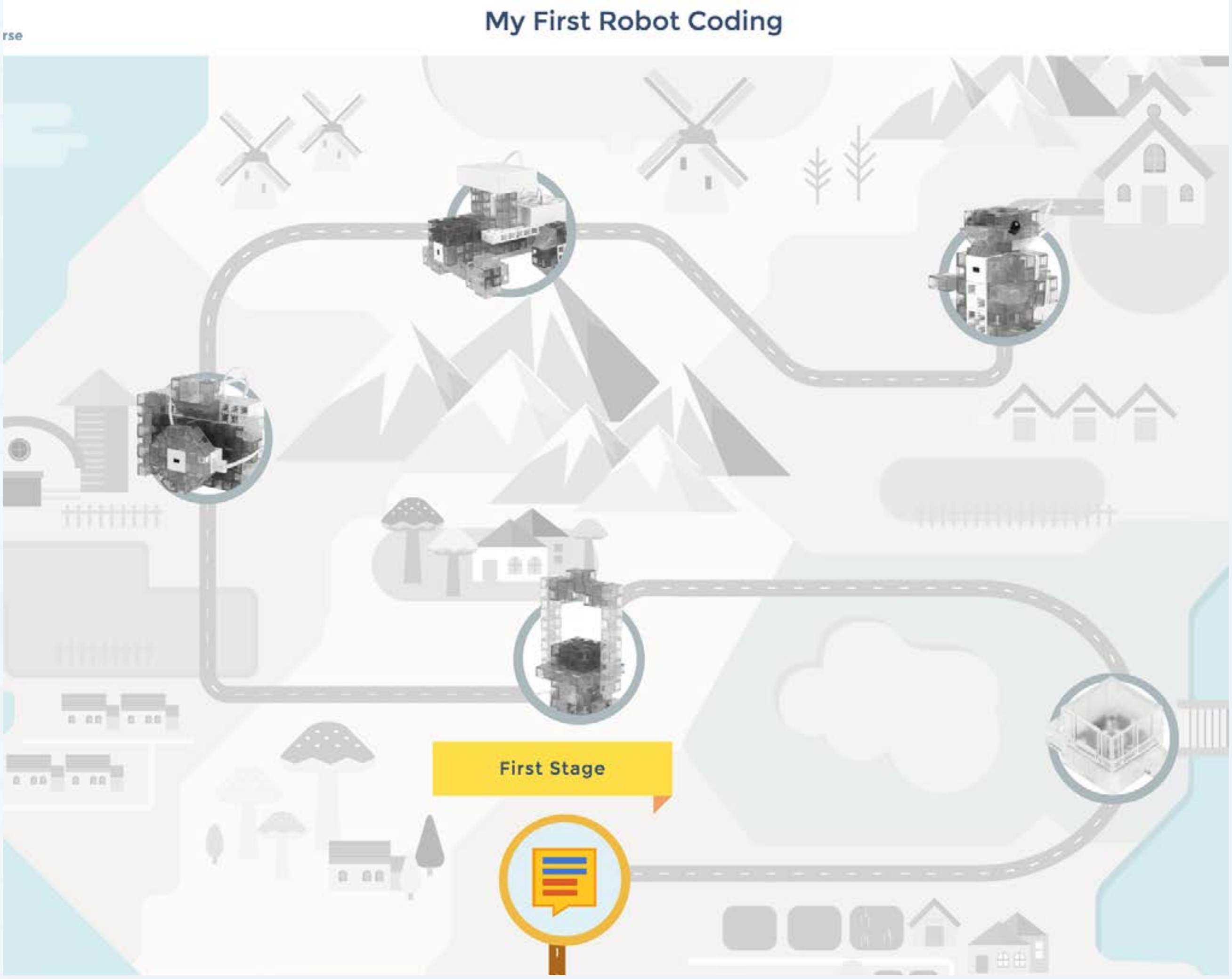
Map: My First Robot Coding

Stage: Let's Start Coding with KOOV

Mission: What is Coding?

To go to Map:

1. Login
2. Select "Learning Course"
3. Select "My First Robot Coding"



L

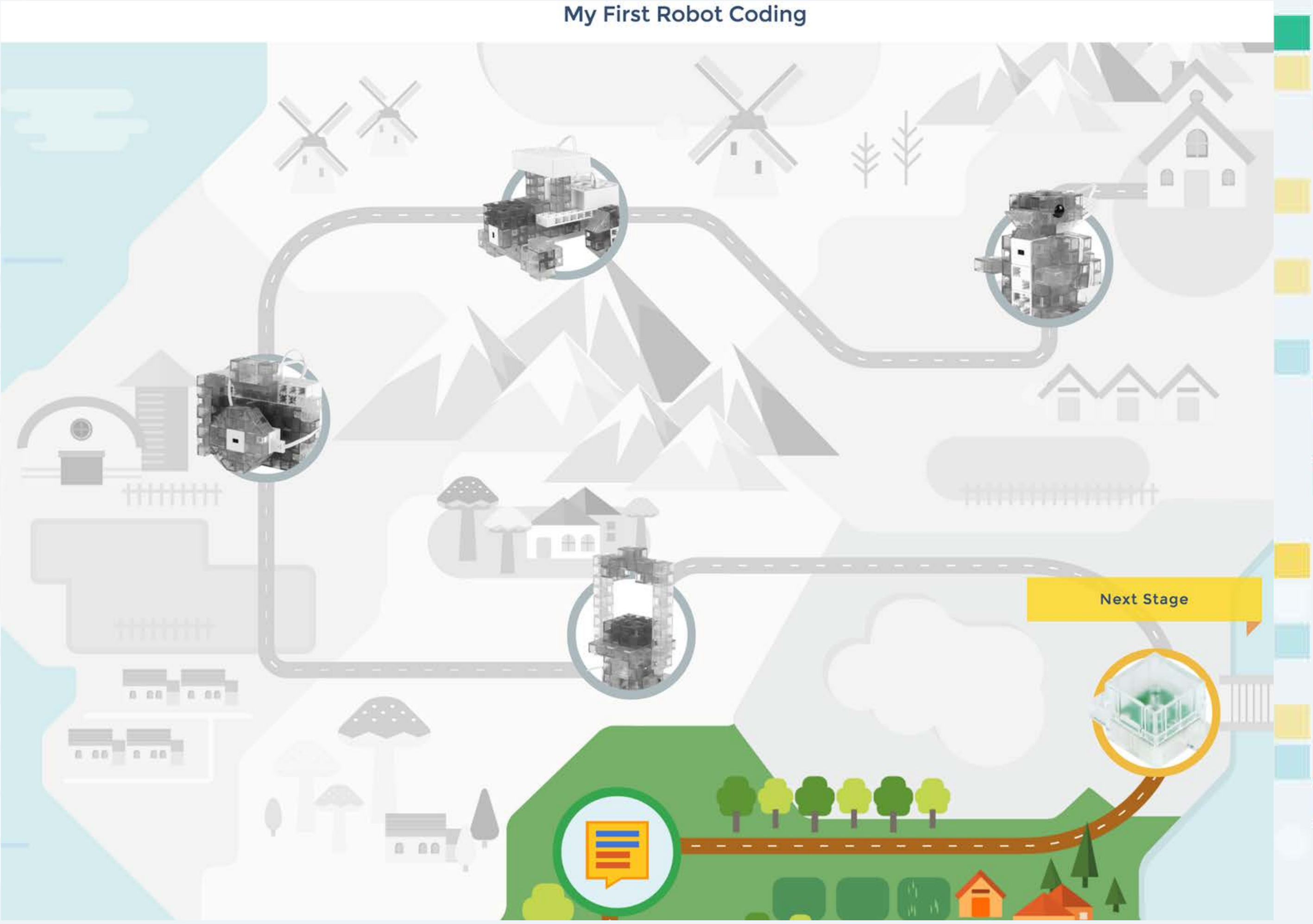
My First Robot Coding

Stage 2: Getting into Electronics

Mission 1: Make an LED light up
Mission 2: Make an LED blink
Mission 3: Use two LEDs
Mission 4: Learn how to use the app

To go to Map:

1. Login
2. Select "Learning Course"
3. Select "My First Robot Coding"



My First Robot Coding

DAY 2

Blink N Beep @ Night

Day 2 Objectives

I Introduction	Sound
E Electronics	Light Sensor
C Coding Concepts	Loops and Conditionals
M Map	Lantern

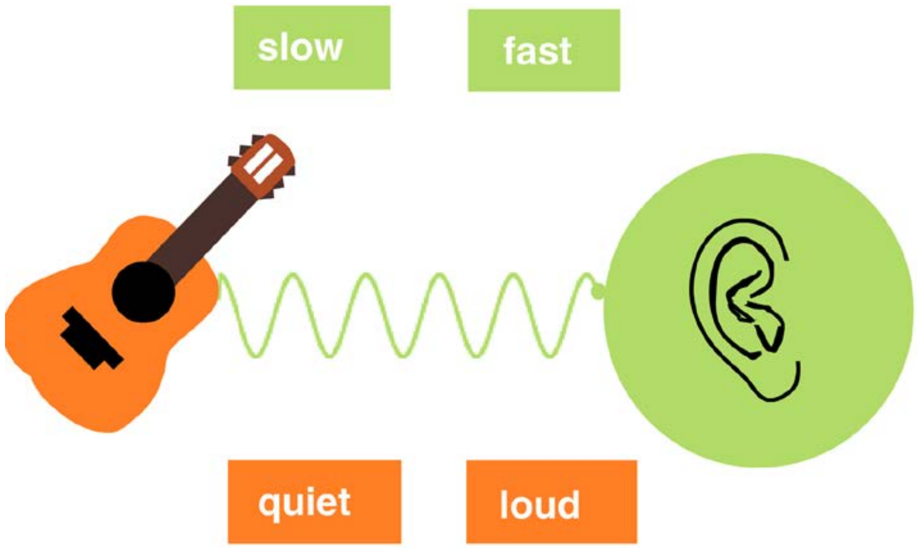
Sound

Sound

animation

Sound is the vibration of air particles, which travels to our ears through air. The vibrations of sound in the air are called sound waves.

When a guitar string is plucked, the string vibrates the soundboard, which sends sound waves through the air to your ear.



A buzzer works by **vibrating** the air at different rates, sending the sound wave to your ear. The faster the air vibrates, the faster the sound wave move. The speed of vibration is called **frequency**.Moreover, the higher the sound wave is, the louder the sound is. The height of the sound wave is called **amplitude**.

Electronics

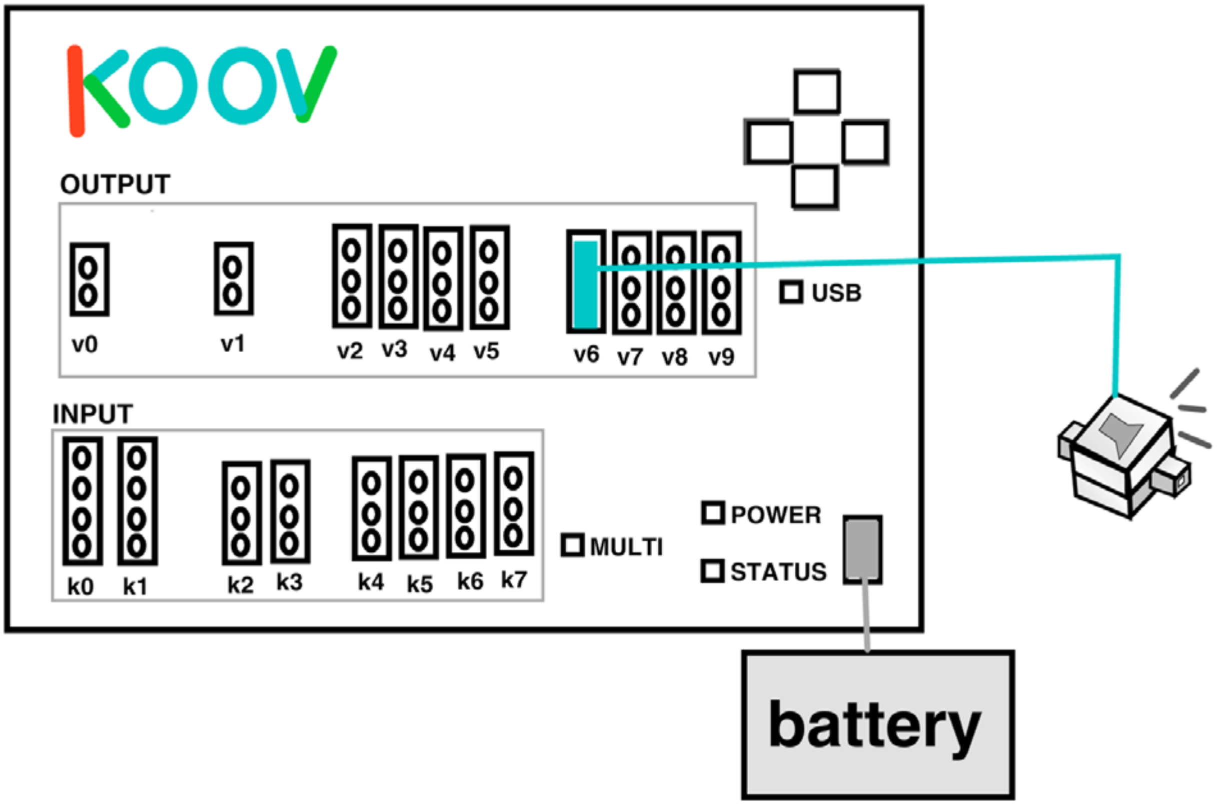
KOOV Buzzer

KOOV Buzzer

KOOV Buzzers makes vibration at different **frequencies** to make different sounds.

Buzzers draw more current and requires the battery. When using the KOOV Buzzer, make sure to connect the KOOV battery pack.

To connect a buzzer to the KOOV Core, connect it to any output port from V2 to V9.

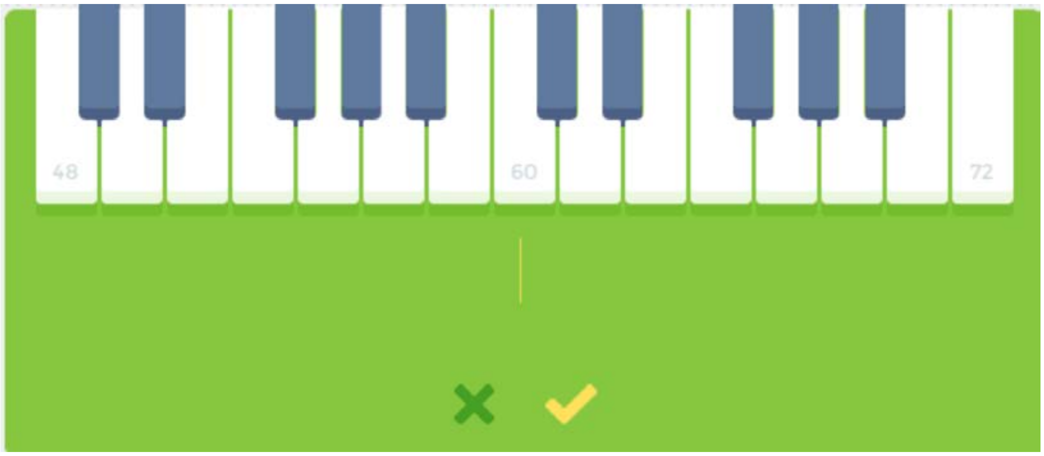


E

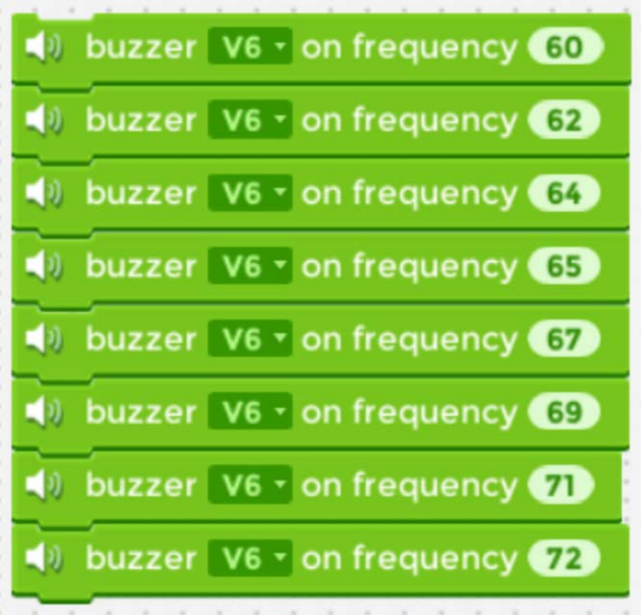
Play a Tune with Buzzer

simulation

A KOOV Buzzer can buzz at different frequencies, using buzzer ( ) on frequency ( ) code block. the frequency from a piano board.



You can make your own tune if you use the right frequencies. Here is an example of a simple tune:



```
buzzer V6 on frequency 60
buzzer V6 on frequency 62
buzzer V6 on frequency 64
buzzer V6 on frequency 65
buzzer V6 on frequency 67
buzzer V6 on frequency 69
buzzer V6 on frequency 71
buzzer V6 on frequency 72
```

Electronics

Light Sensor

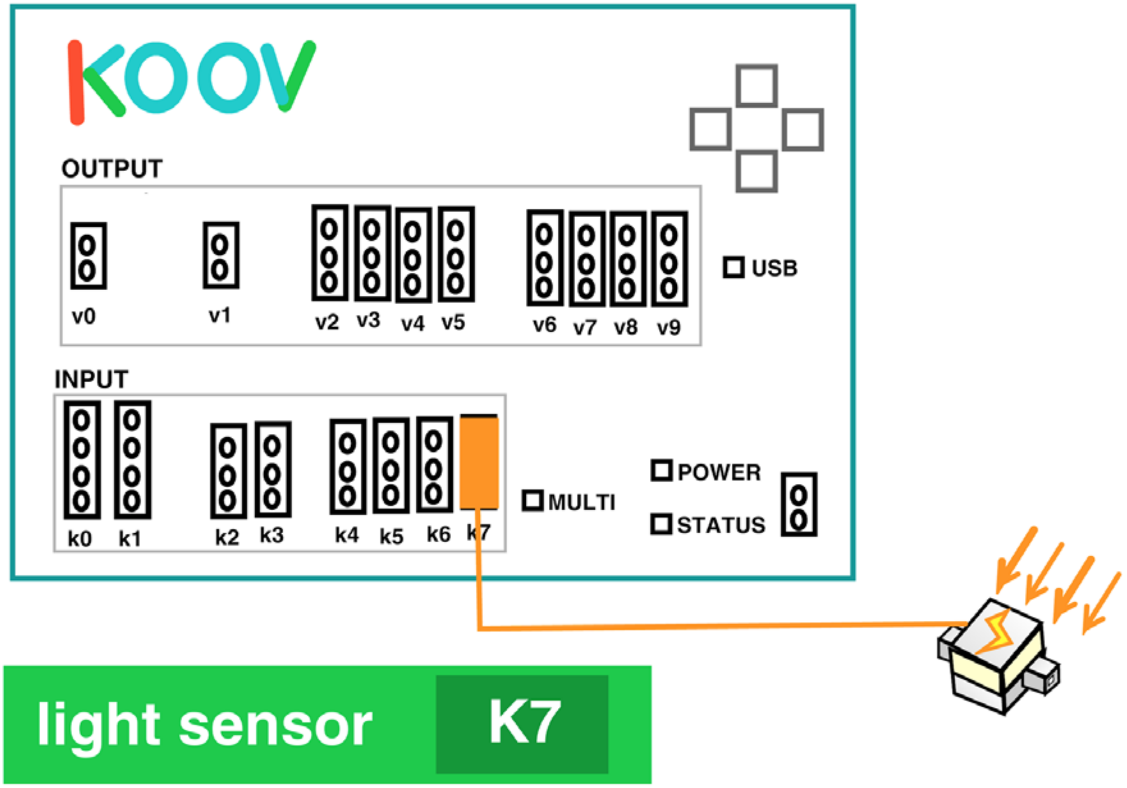
KOOV Light Sensor

animation

KOOV Light Sensor detects the brightness of its surrounding. If it’s bright in the room, the Light Sensor would detect higher energy; if it’s dimmed, it would detect lower energy.

KOOV Light Sensor can be connected to the Output ports K2 to K7 on the Core. The KOOV Light Sensor shown below is connected to input port K7.

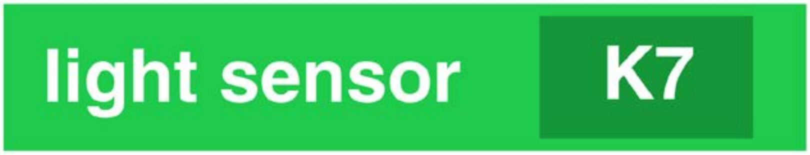
The **light sensor ()** code block returns the energy level from the Light Sensor.



light sensor K7

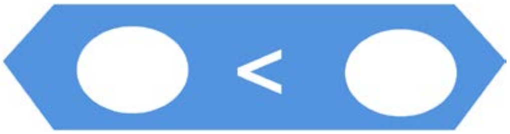
Light Sensor Code

The **light sensor ()** code block gives the amount of electric current detected by the Light Sensor device..



light sensor K7

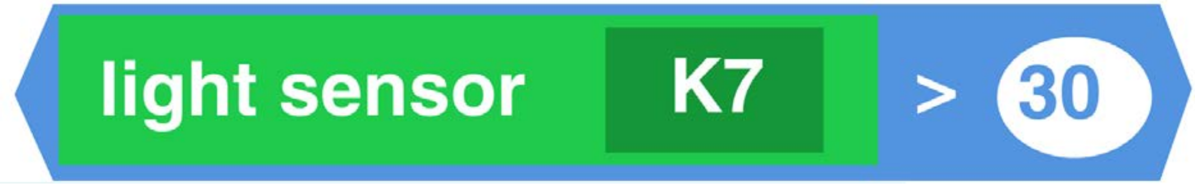
To check if a room is dark, use the Light Sensor device and **light sensor ()** code block, as well as the **( ) < ( )** operator.



<

Typically, it’s dark when the energy level is less than 30. Therefore, to check if a room is dark, you can check if the current is less than 30.

The **sensor (K7) < 30** code segment returns True or False. If it’s dark, then this code segment would return True. If it is bright, then it would return False.



light sensor K7 > 30

Loops and Conditionals

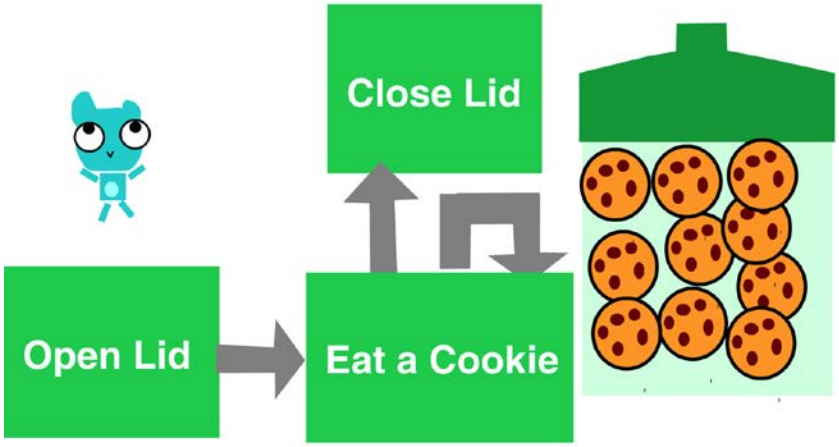
C

Loops and Iterations

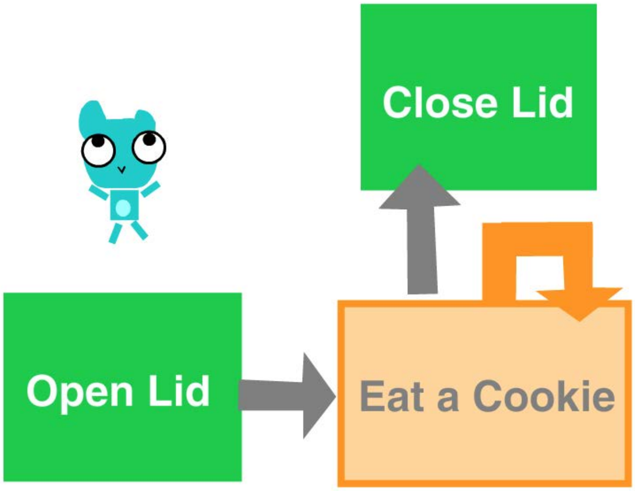
animation

A loop is a set of steps that can be repeated. Therefore, that step is in a loop. An iteration is a repetition through a loop. Since coding is to provide step-by-step instruction to complete a task,

if we want a robot to eat three cookies, and each green box below is a step, which step would it repeat?



It's the "Eat a Cookie" step. Each time a robot goes through the loop to take and eat a cookie, it's one iteration.



KOOV

Conditional

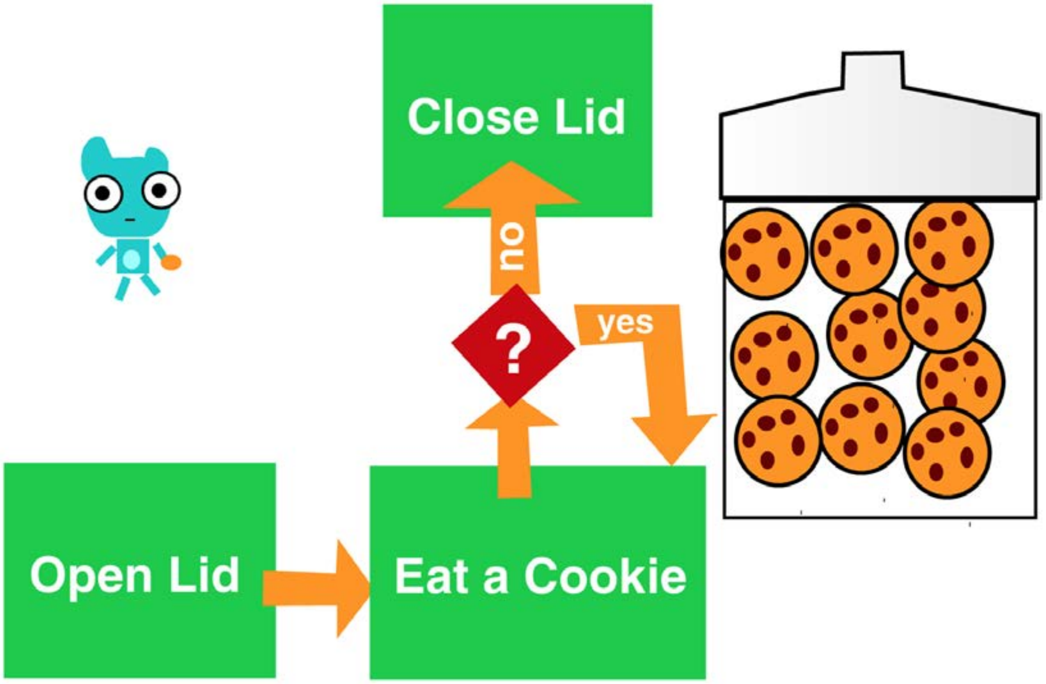
animation

We make decisions daily, such as:

> “IF Vinny gives me a french fries, THEN I will give him a candy.”
> “IF Matt brings his tennis racquet tomorrow, THEN I will bring mine too. ELSE we can just run around the park.”

Computers and robots make decisions as well. **Conditionals** are the way computers make decisions. Robots make decisions based on conditions or the state of something, such as the brightness or the loudness of the room.

A **conditional loop** is a loop that is only repeated if a condition is true. An example would be a conditional cookie loop. Before repeating the “Eat a Cookie” step, check if there are more cookies. Eat another cookie if there are still cookies, if not, close the lid.



M

a

p

Lantern

KOOV

L

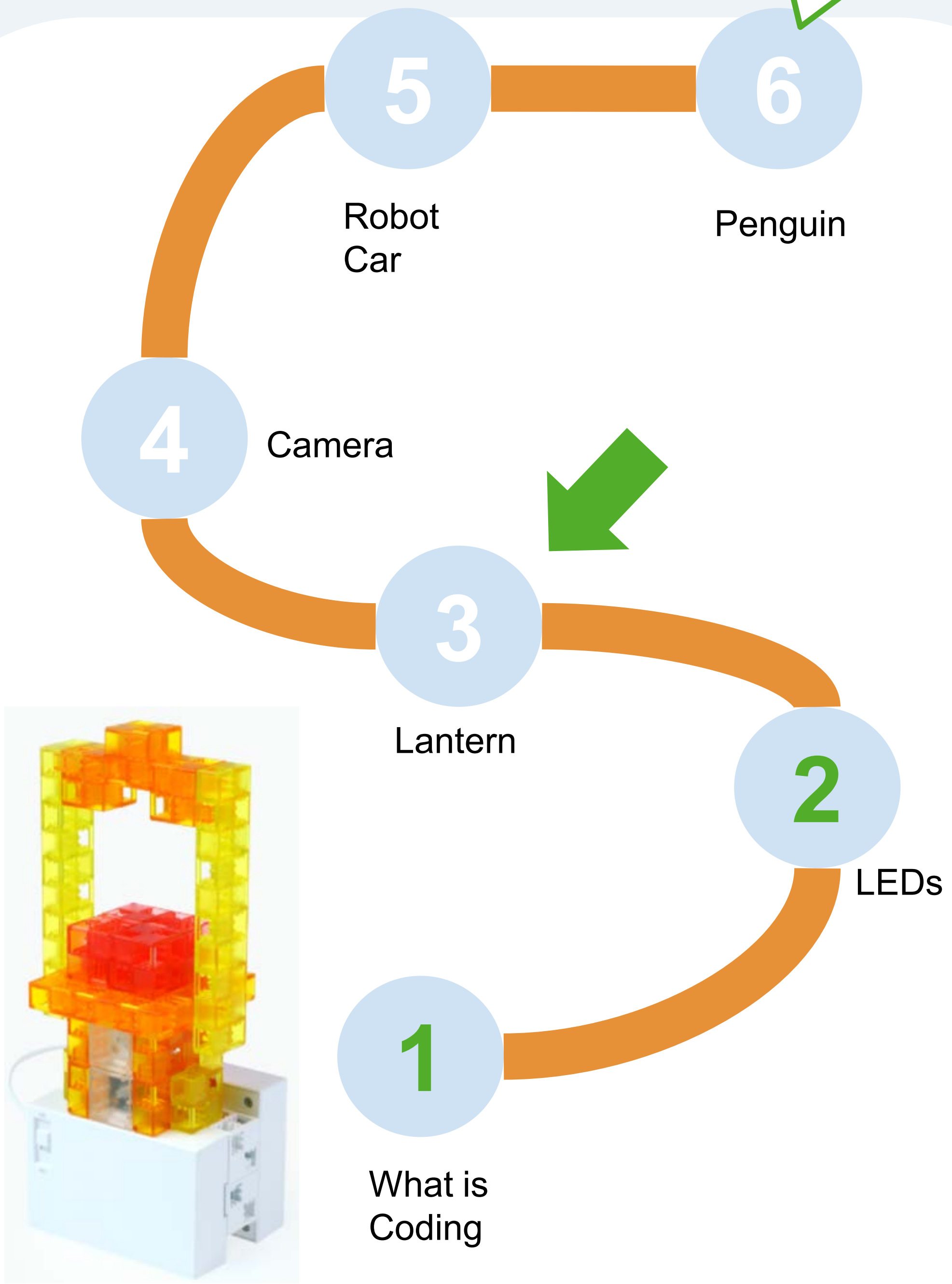
Robotic Coding Map: Lantern

Map: My First Robotic Coding
Stage: Stage Three

Mission 1: Make a lantern
Mission 2: Use the Core Buttons
Mission 3: Learn how to use a Light Sensor

To go to Map:

1. Login
2. Select "Learning Course"
3. Select "My First Robot Coding"



KOOV

KOOV

Robotics Camp

Fun, Learning, and Discovery

Student Manual

Day 1

I
Introduction

Robots

E
Electronics

Intro to KOOV Kit

C
Coding Concepts

Intro to Coding

L
Lab

Blink that LED

Day 1 Electronics: KOOV Robotic and Coding Kit

What does the KOOV Kit include?

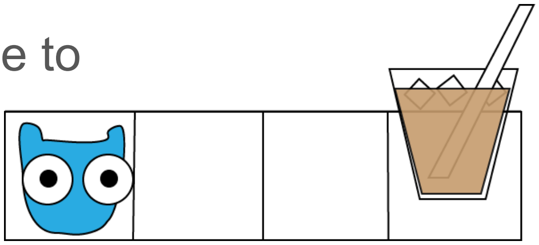
Match KOOV Parts with Human Body Parts

Sound Sensor	Muscle
Light Sensor	Mouth
Blocks	KOOV Core
KOOV Core	Brain
Buzzer	Eye
LED	Ear
Motor	Hand (to make signals)

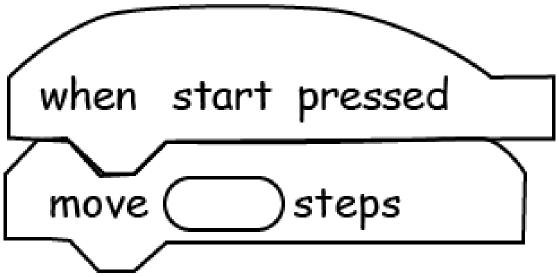
Day 1 Coding Concept: Intro to Coding

Coding is to give step-by-step _______________ for computers

How many steps does KOOVie have to take to reach the iced tea?



Fill in the blank in the code segment at the right.



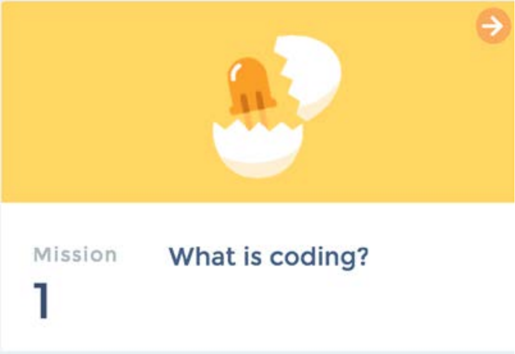
```
when start pressed
move (    ) steps
```

To talk to a human, we speak in different languages; to talk to a computer, we would write in ______________________ language.

Day 1 Map: Stage 1, Mission 1 - What is coding?

Login.
Click on “Learning Course”.
Click on “My First Robot Coding”.
Go to “My First Robotic Map”.
Select Stage 1, “Let’s Start Coding with KOOV”.

Complete this stage, then answer the questions.



Mission 1 What is coding?

Coding is a way to give what to robots and computers?

a) Big Reward
b) Strong Electricity
c) Step-by-step Instruction
d) Fancy Wrapper

The instructions for robots must be written in?

a) Paper
b) Whiteboard
c) Code
d) Novel

Day 1 Map: Stage 2, Mission 1 - Make an LED light up

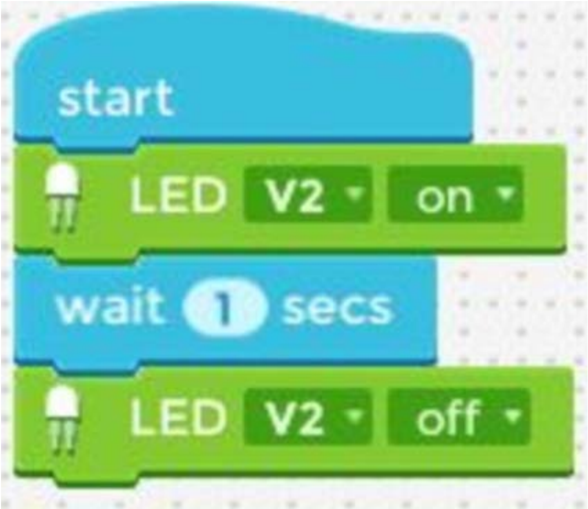
Login
Click on "Learning Course"
Click on "My First Robot Coding"
Go to "My First Robotic Map"

Complete the "Starting with Electronics" Stage in the Map.

Then answer the questions below.

Mission 1

Make an LED light up



start
LED V2 on
wait 1 secs
LED V2 off

What does **LED [V2] [on]** code block do?

____

____

____

What does **wait [1] secs** code block do?

____

____

____

What does **LED [V2] [off]** code block do?

____

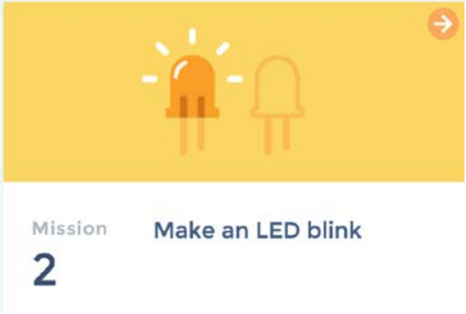
____

____

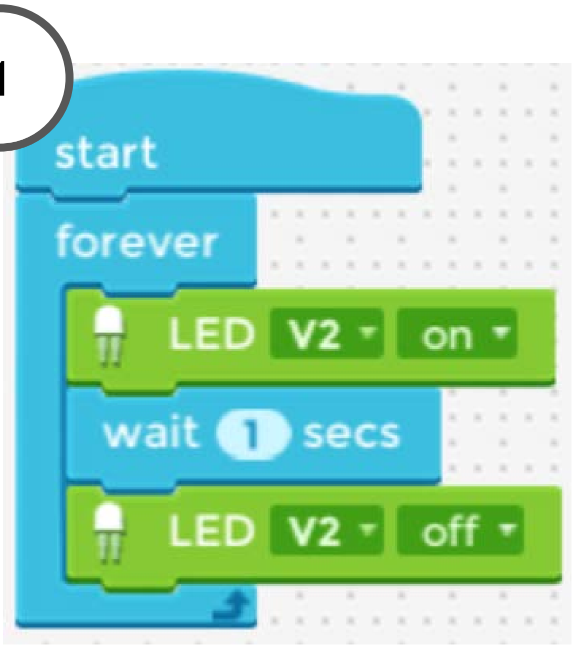
Day 1 Map: Stage 2, Mission 2 - Make an LED Blink

Go to “My First Robotic Map”
Select “Second Stage”.
Select “Mission 2”.

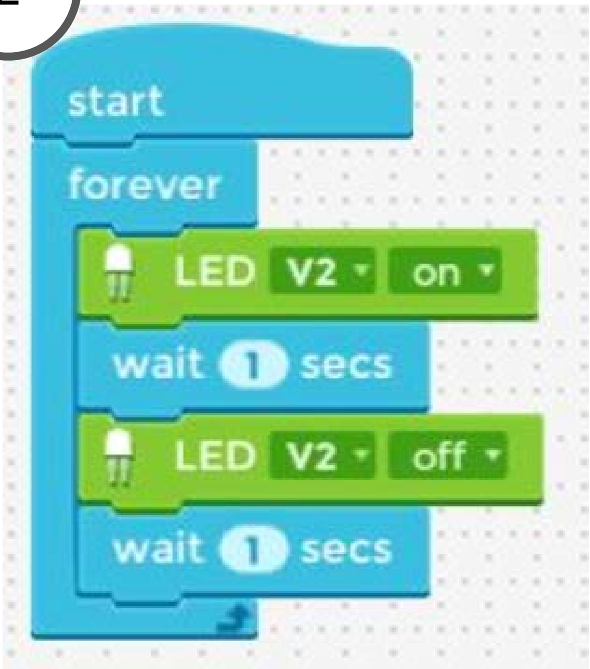
Complete this mission, then answer the questions.


Mission
2
Make an LED blink

1


```
start
forever
  LED V2 on
  wait 1 secs
  LED V2 off
```

2


```
start
forever
  LED V2 on
  wait 1 secs
  LED V2 off
  wait 1 secs
```

Would program would make a LED blink? One or Two?

____

____

Why does not both programs work?

____

____

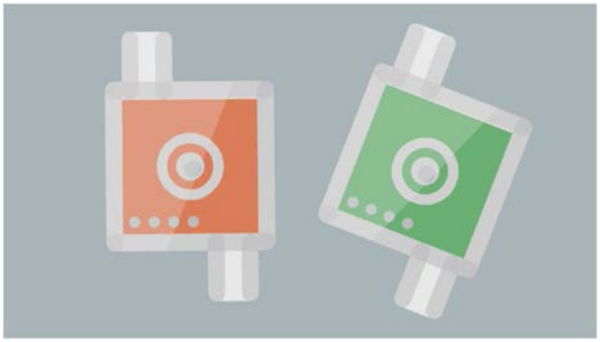
What does the **forever {}** code block do?

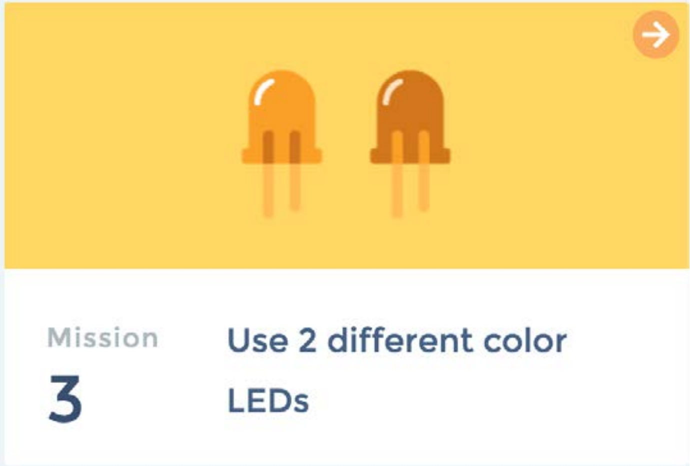
____

____

Day 1 Map Stage 2 Mission 3: Use Two LEDs

This mission's goal





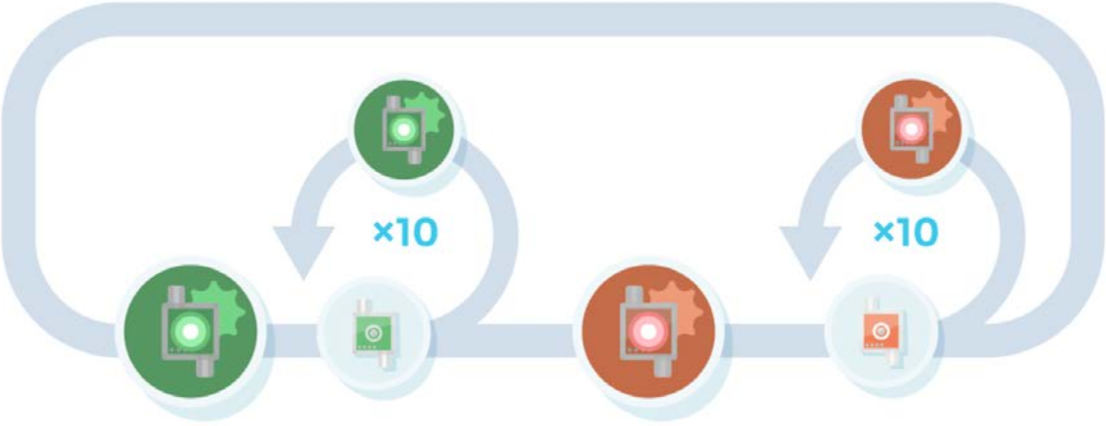
Mission

3

Use 2 different color LEDs

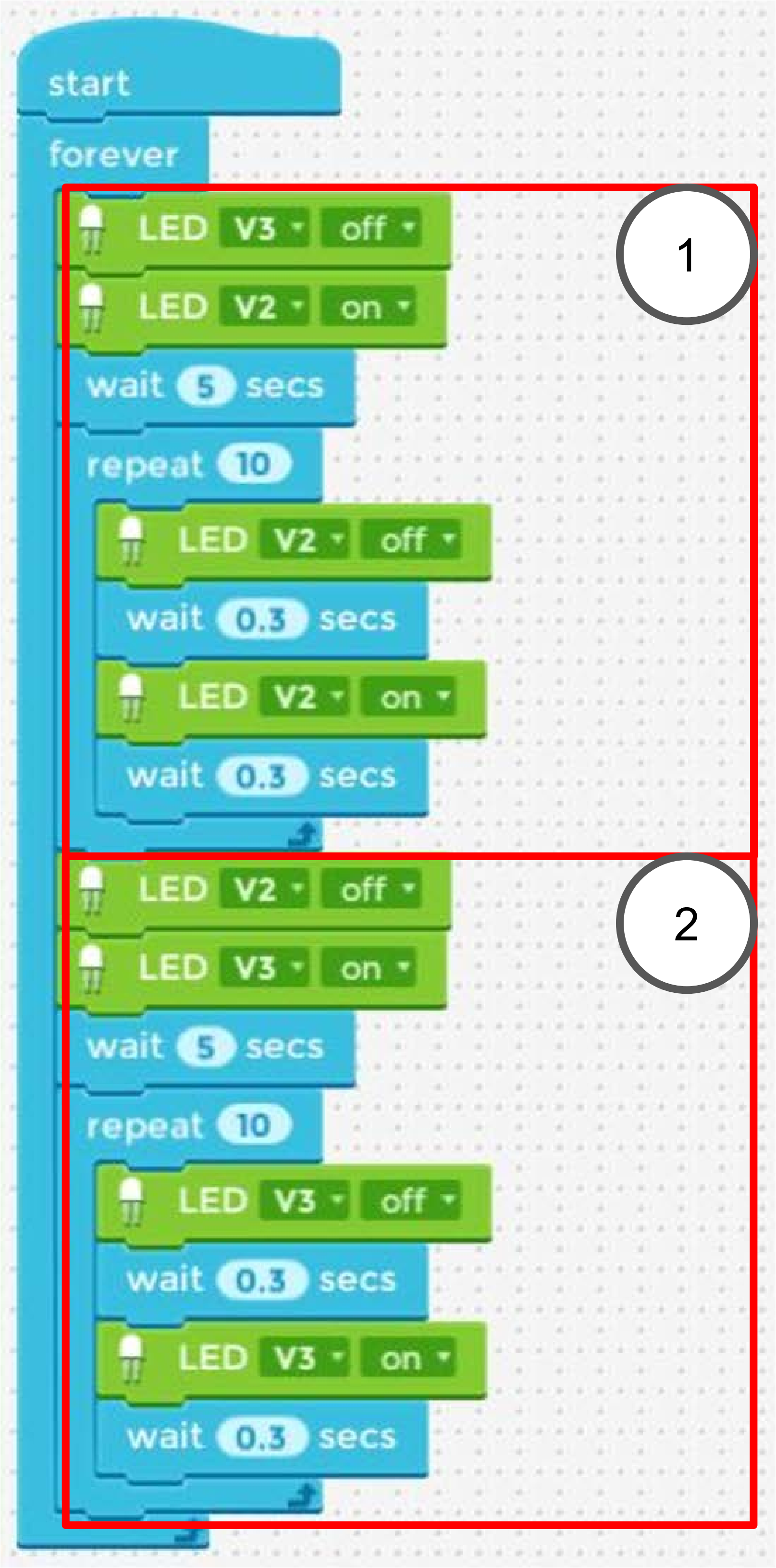
Let's code so that the 2 different color LEDs "light up for 5 seconds and blink 10 times", and repeat this pattern.

Coding plan



Make two instructions so that each LED "lights up for 5 seconds and blink 10 times". Then, surround the entire code with the "forever" block. Let's make the LEDs wait 0.3 seconds between each blink.

Day 1 Map: Stage 2, Mission 3 - Use Two LEDs
(Understand the Code)



```
start
forever
  LED V3 off
  LED V2 on
  wait 5 secs
  repeat 10
    LED V2 off
    wait 0.3 secs
    LED V2 on
    wait 0.3 secs
  LED V2 off
  LED V3 on
  wait 5 secs
  repeat 10
    LED V3 off
    wait 0.3 secs
    LED V3 on
    wait 0.3 secs
```

If the Green LED is connected to V2 port, and the Red LED is connected to V3 port, which LED would blink first?

____

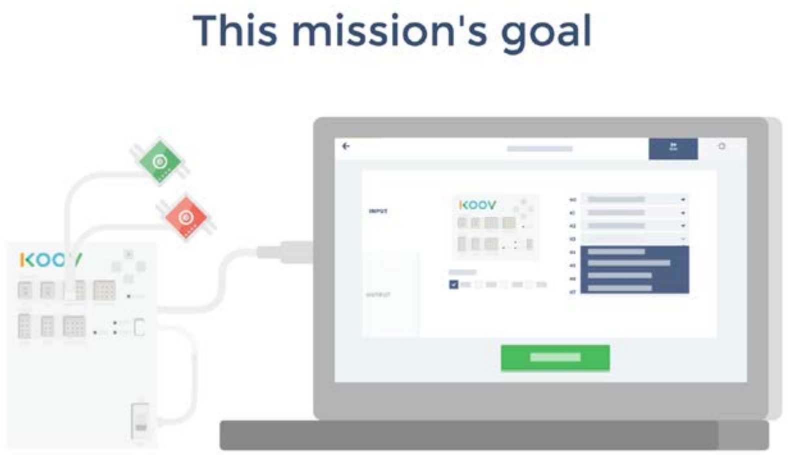
Would both LEDs blink at the same time? Why or Why not?

____

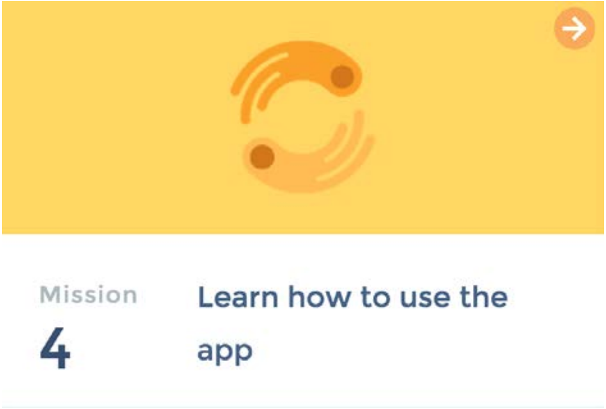
Does code segment 1 or 2 make the Green LED blink? (Green LED is connected to V2 port)

____

Day 1 Map: Stage 2, Mission 4 - Learn how to use the app



This mission's goal

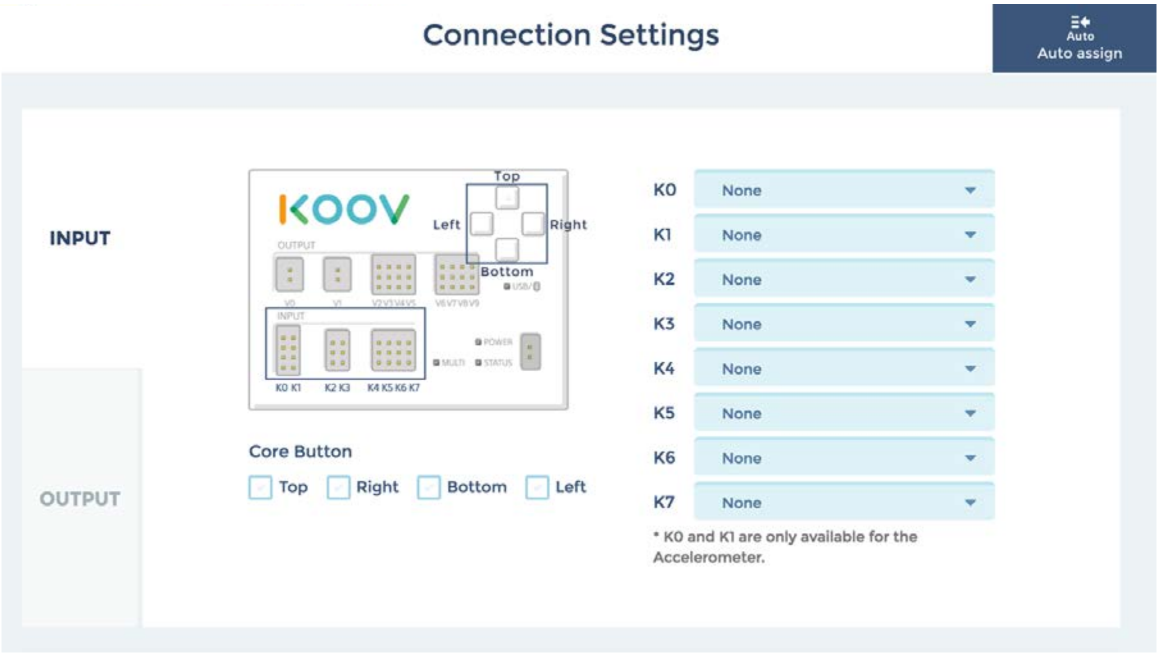


Mission 4

Learn how to use the app

Goal 1: Learn how to use Connection Settings
Goal 2: Learn how to run a program in Test Mode

Goal 1



Connection Settings

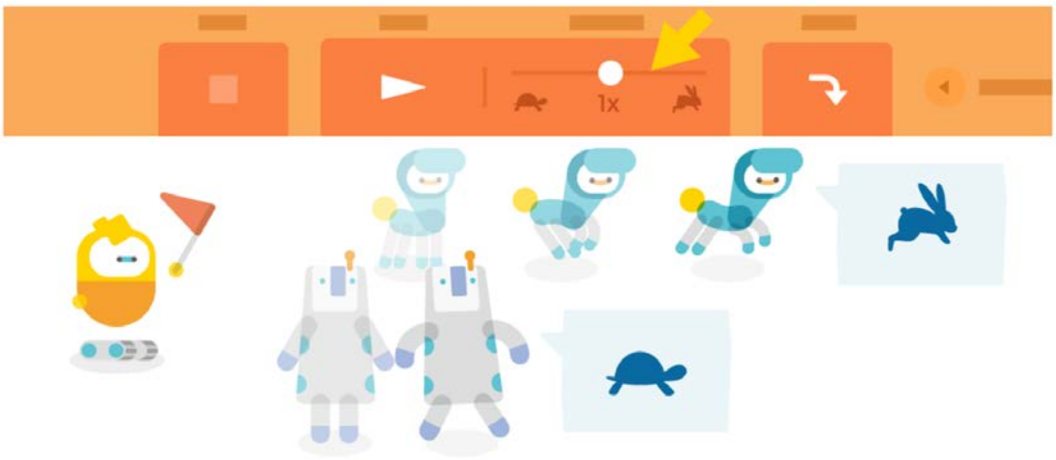
Core Button

□ Top □ Right □ Bottom □ Left

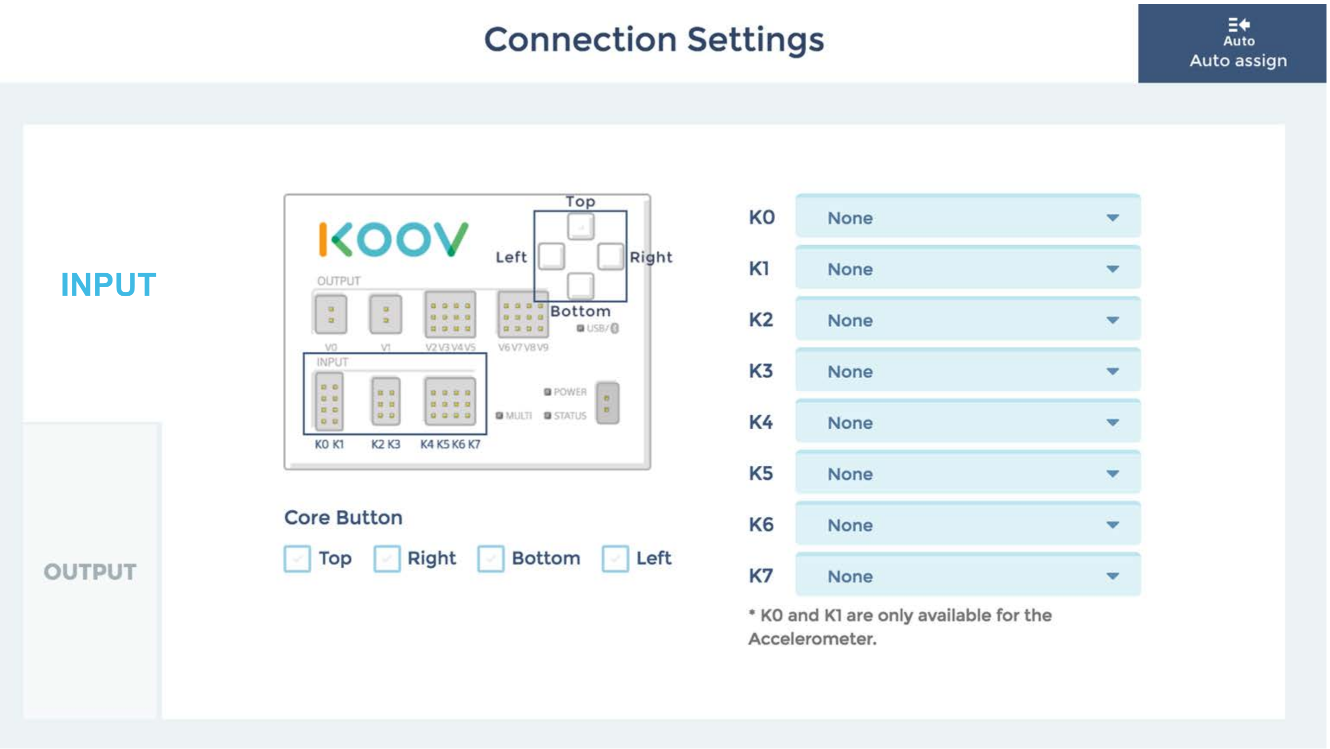
K0	None
K1	None
K2	None
K3	None
K4	None
K5	None
K6	None
K7	None

* K0 and K1 are only available for the Accelerometer.

Goal 2



Day 1 Map: Learn how to use the app (INPUT Tab)



Connection Settings

Core Button

□ Top □ Right □ Bottom □ Left

K0	None
K1	None
K2	None
K3	None
K4	None
K5	None
K6	None
K7	None

* K0 and K1 are only available for the Accelerometer.

1. On the INPUT Tab, there are four Core Buttons:

_______________, _______________, _______________, and _______________.

1. List all the input ports:

____

____

1. The _______________________ can only be plugged into input ports K0 and K1.

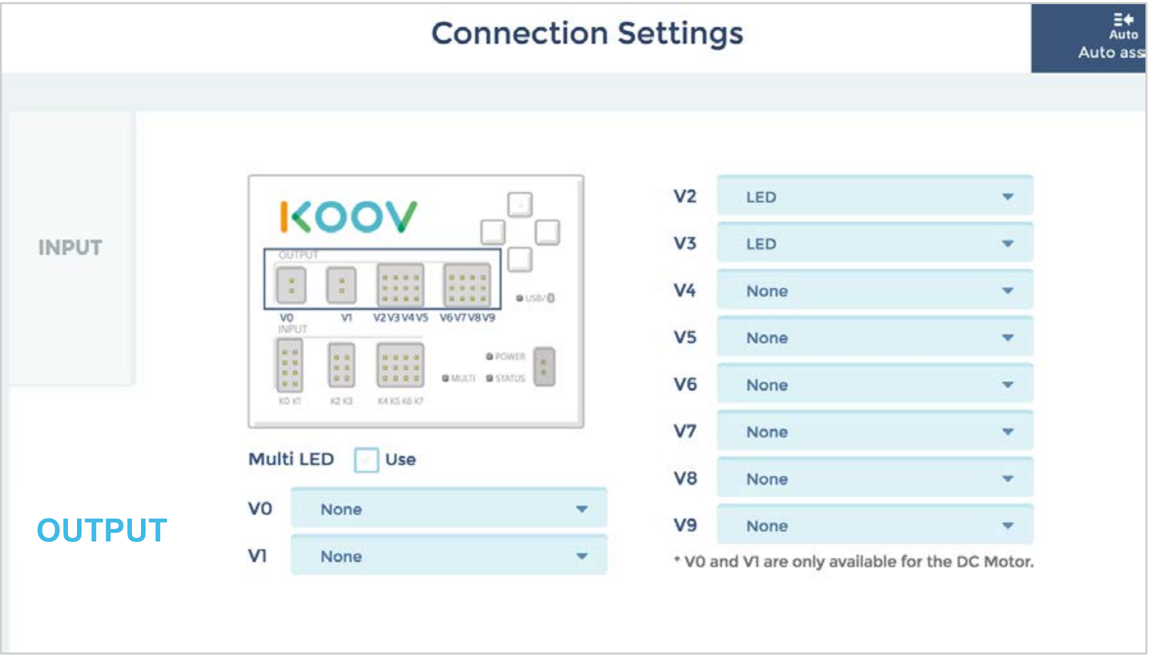
2. You can connect the following electronic devices to port K2 to K7:

____

____

____

Day 1 Lab: Learn how to use the app (OUTPUT Tab)



Connection Settings

Multi LED □ Use

V0 None

V1 None

V2 LED

V3 LED

V4 None

V5 None

V6 None

V7 None

V8 None

V9 None

* V0 and V1 are only available for the DC Motor.

1. At the OUTPUT Tab, you can check a ______________________ box to blink the "MULTI" light on KOOV Core.
2. List all output ports:

   ____

   ____

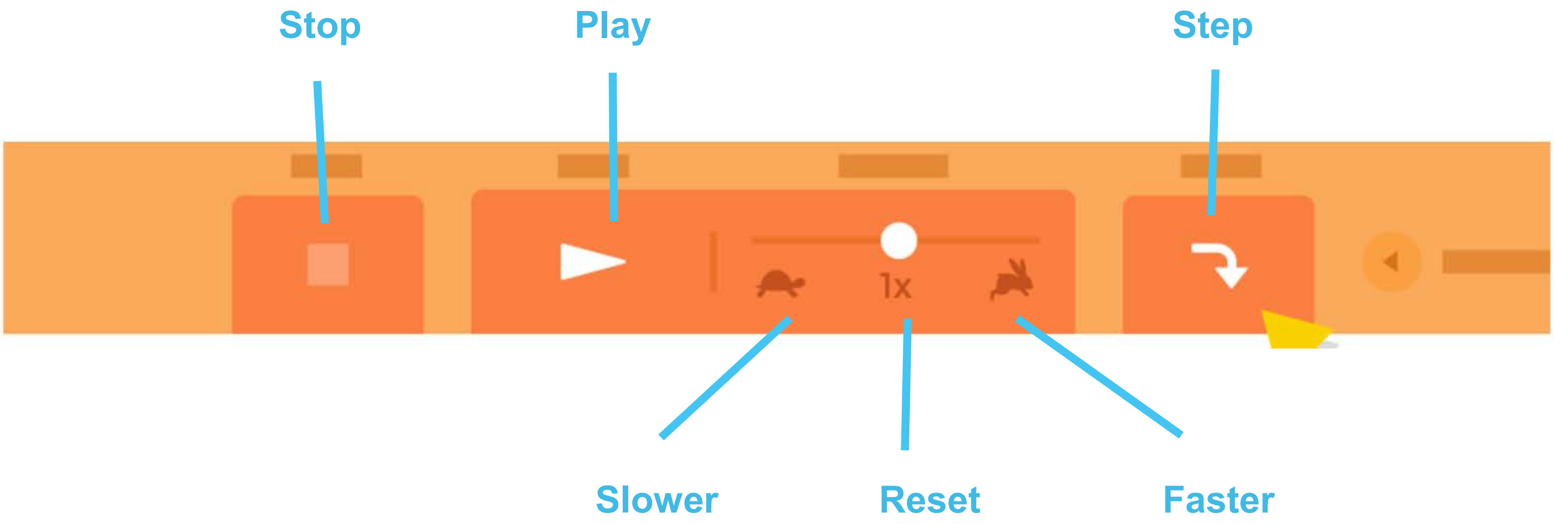
1. You can only connect this electric device to output port V0 and V1:

   ____

1. You can connect the following output electronic devices to port V2 - V9:

   ____

Day 1 Lab: Robotic Coding Map - KOOV Code Editor



You can change the running speed of your code in the Test Mode.

1. How to start and stop the code? Which button?

   ____

1. How do you go slower? Which button?

   ____

1. How do you go faster? Which button?

   ____

1. What does the “Step” button do?

   ____

1. Can you guess why it is useful to slow down the running of the code?

   ____

Day 2

E LEDs and Buzzers

C Loops and Conditionals

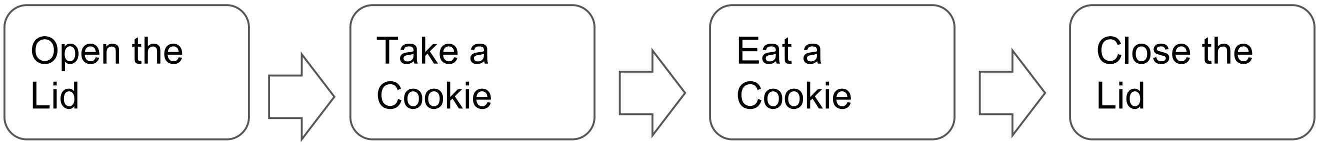
L Lantern

Day 2 Concept: Loops

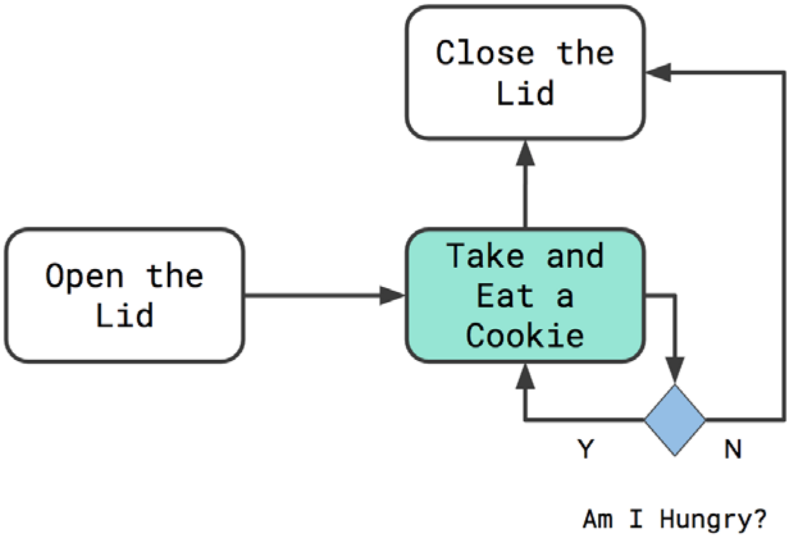
A loop is a set of _______ that can be repeated. Each repetition through the loop is called a ____________________

The diagram below shows steps to eat cookies. To eat three cookies, which steps below can be in a loop?

____

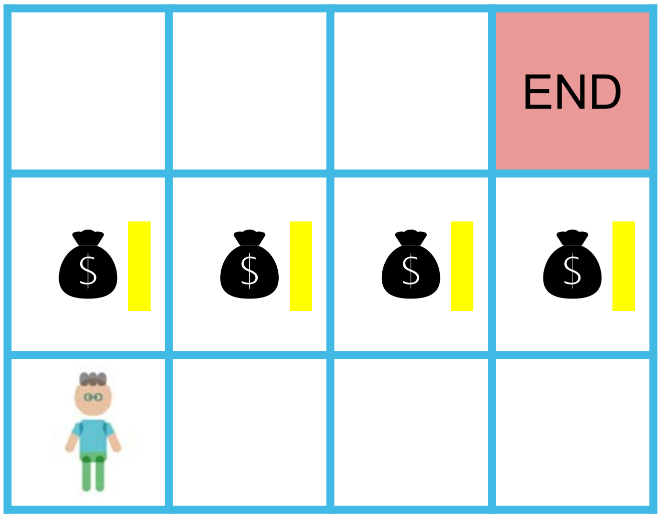


A loop can be ___________________ if a decision is made before repeating that loop.



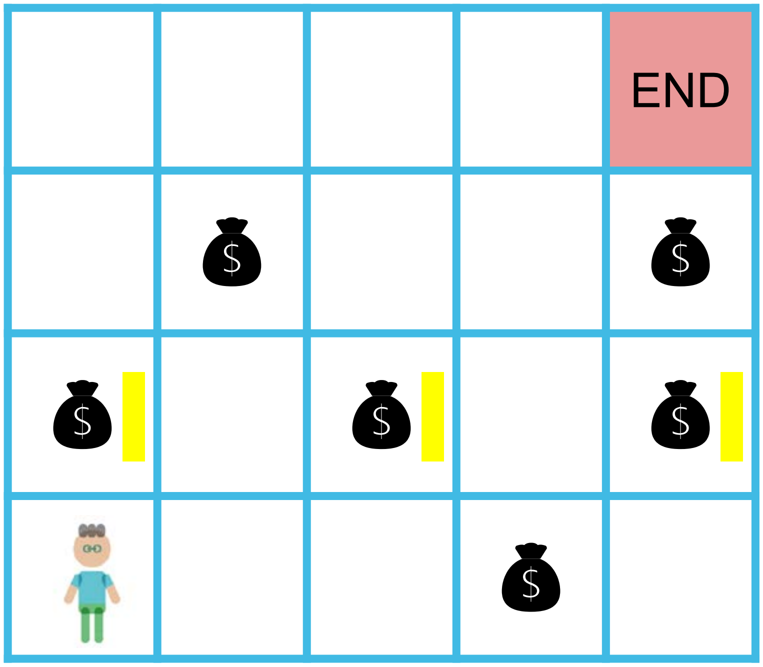
Day 2 Concept: Repeat Loop in a Maze

Below is a maze. Collect 💰 as you go through the maze to the end.



```
start
move up
move right
move right
move right
move up
```

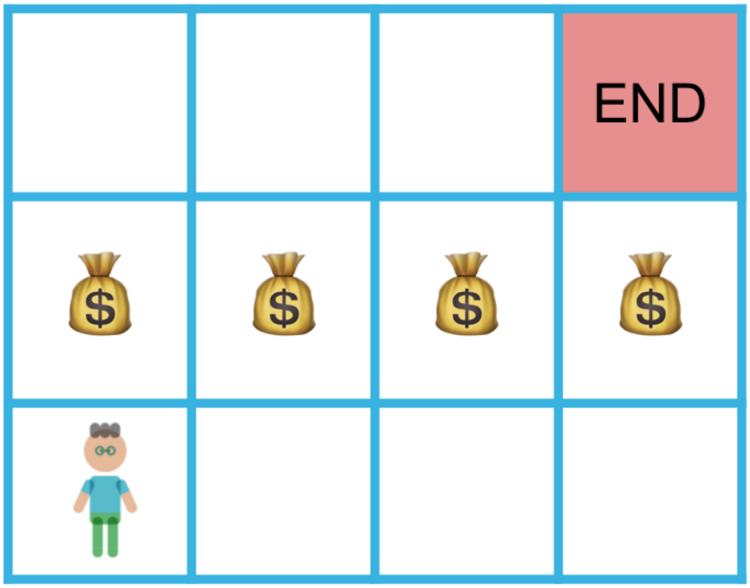
Can you write the code to move through this maze?



```
start
```

Day 2 Concept: Repeated Steps in a Maze

Did you notice that the following code has some repeated lines?



```
start
move up
move right
move right
move right
move up
```

You can rewrite the code with the **repeat []** code block.

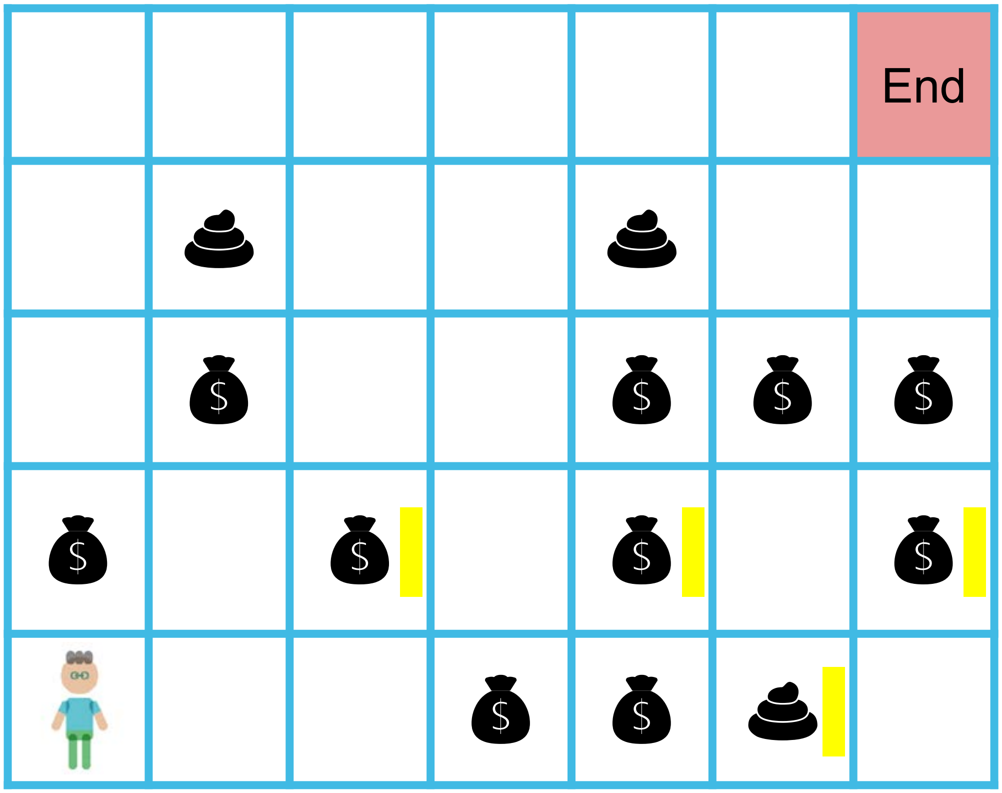
```
start
move up
move right
move right
move right
move up
```

=

```
start
move up
repeat [3]
    move right
move up
```

Day 2 Concept: Repeated Steps in a Maze

Write the code to go from **Start** to **End**. Collect all 💰 and avoid 💩



```
start
```

Code Blocks:

move up
move down
move right
move left
repeat

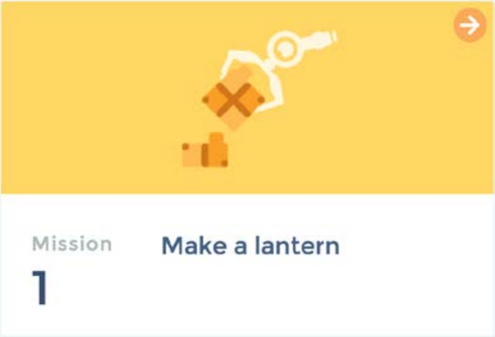
Use at least one **repeat** code block in your program.

Day 2 Map Stage 3, Mission 1: Make a lantern

Login.

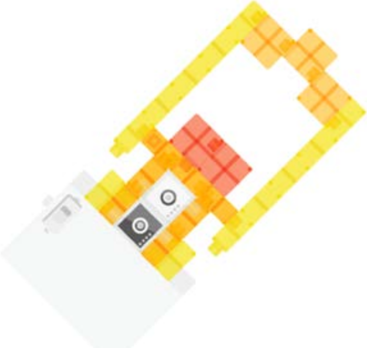
Click on “Learning Course”.
Click on “My First Robot Coding”.

Select Stage 3.
Complete all missions in this stage.



Mission 1 Make a lantern

This mission's goal

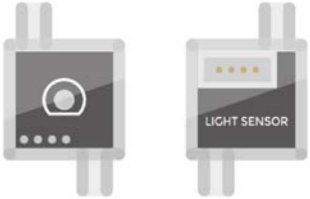


Make a lantern with Core Buttons and Light Sensor

Core Buttons

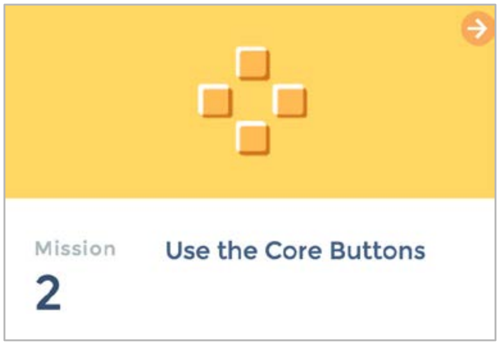
On the top right of the Core, there are 4 buttons. These buttons can be programmed to whatever function you like.

Light Sensor



This sensor detects and measures brightness.
The sensor has numeric values from 0 to 100. The brighter the light, the larger the value.

Day 2 Map Stage 3 Mission 2: Use the Core Buttons

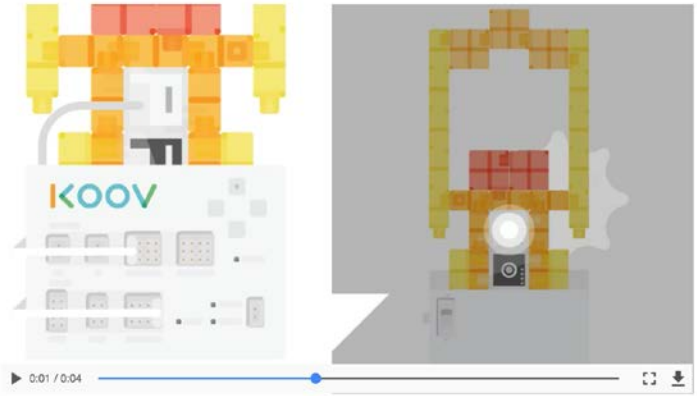


Mission

2

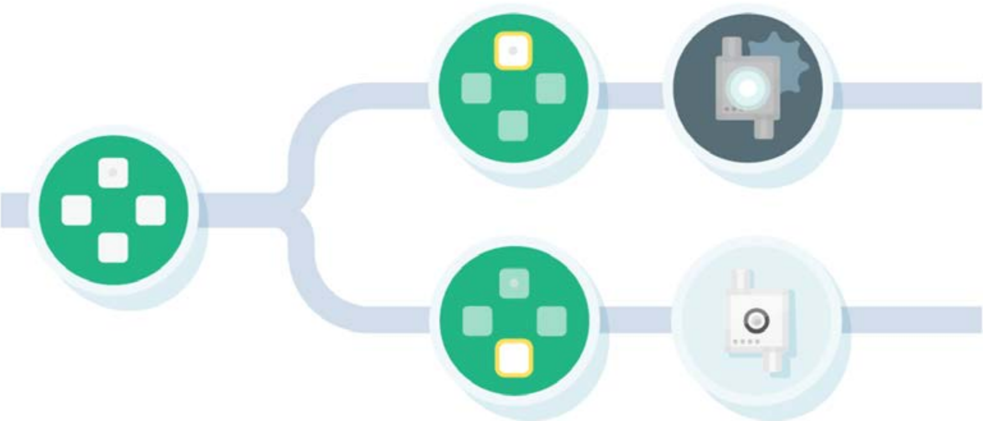
Use the Core Buttons

This mission's goal



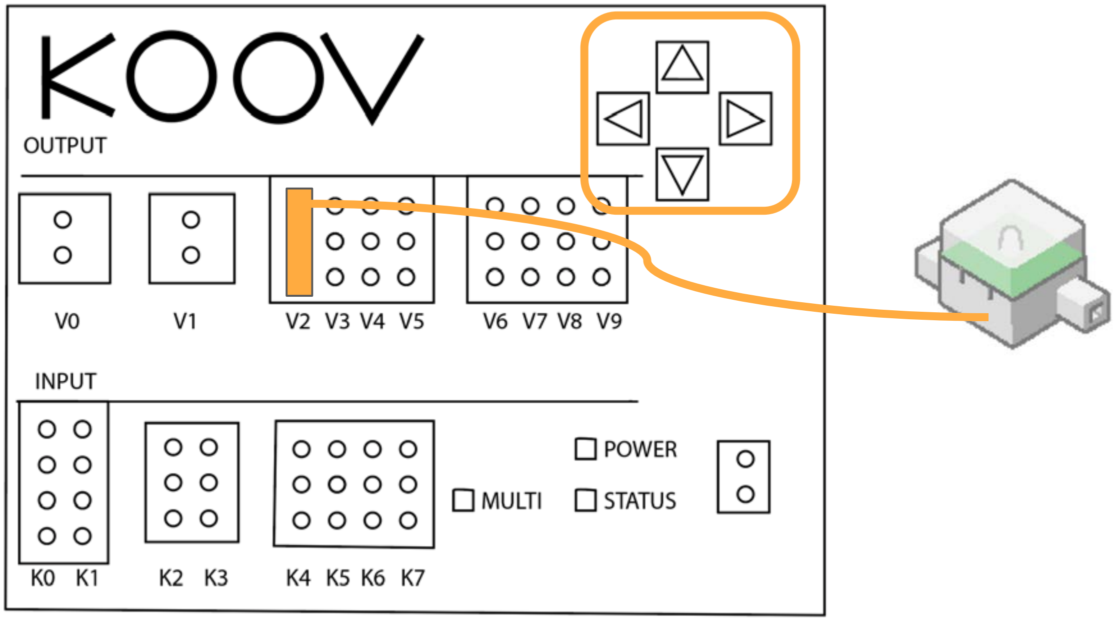
Let's code to turn the LED on and off by using the top and bottom Core Buttons. Let's use the lantern you made in Mission 1.

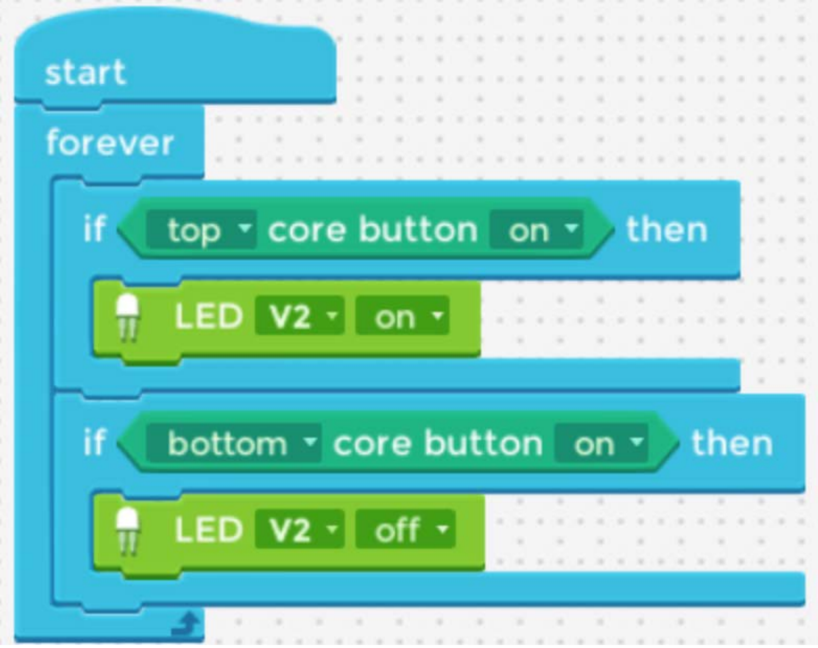
Coding plan



See what happens to the LED "when you press the top button" and "when you press the bottom button." Try thinking about them as two separate instructions.

Day 2 Map Stage 3 Mission 2: Use the Core Buttons





```
start
forever
  if top core button on then
    LED V2 on
  if bottom core button on then
    LED V2 off
```

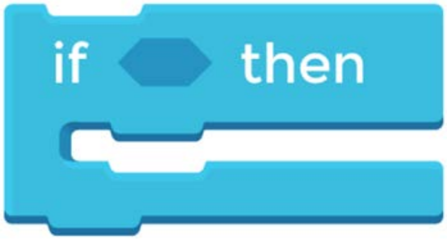
What would happen if the Top Button is pressed?

____

What would happen if the Bottom Button is pressed?

____

Day 2 Map Stage 3 Mission 2: Use the Core Buttons

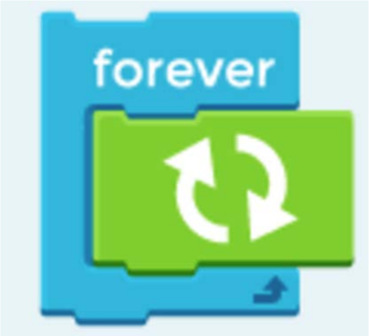


```
if < > then
```

What does the **if [ ] then** code block do?

____

____

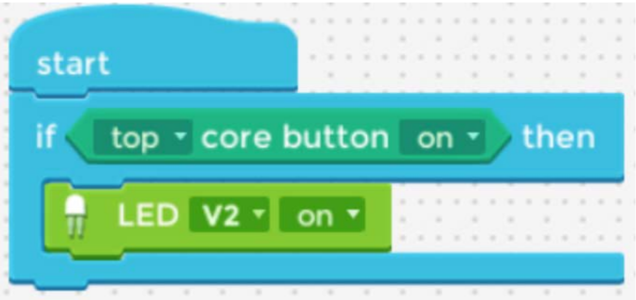


```
forever
```

What does the forever code block do?

____

____



```
start
if < top ▾ core button on ▾ > then
    LED V2 ▾ on ▾
```

This program does not work. What is it missing?

____

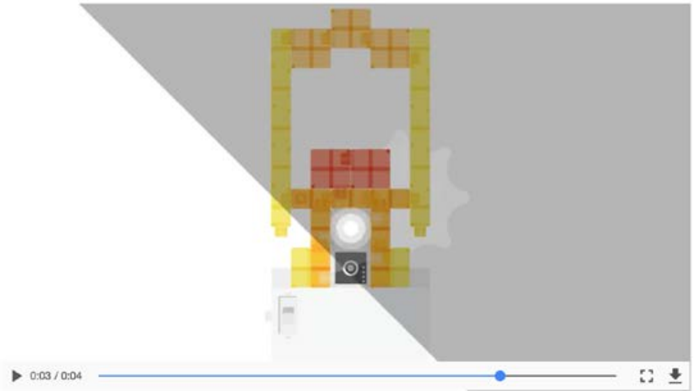
____

Day 2 Map Stage 3 Mission 3: Learn how to use a Light Sensor

Mission 3

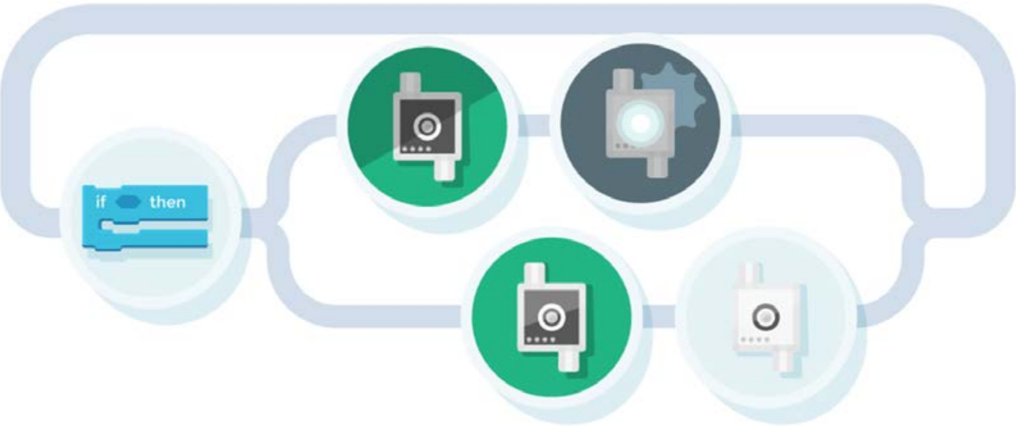
Learn how to use a light sensor

This mission's goal



Let's code so that the LED turns on automatically when the light sensor detects the brightness around it.

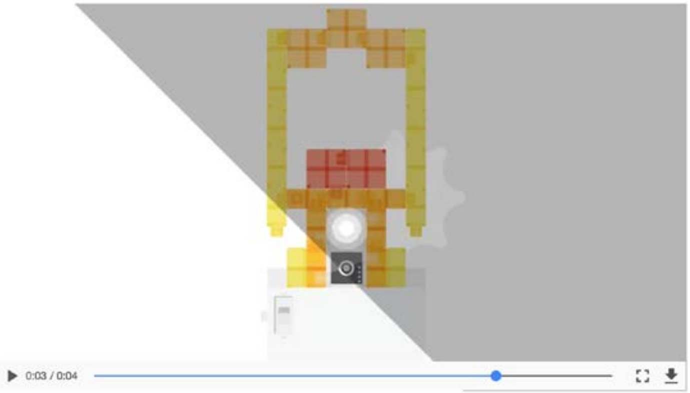
Coding plan



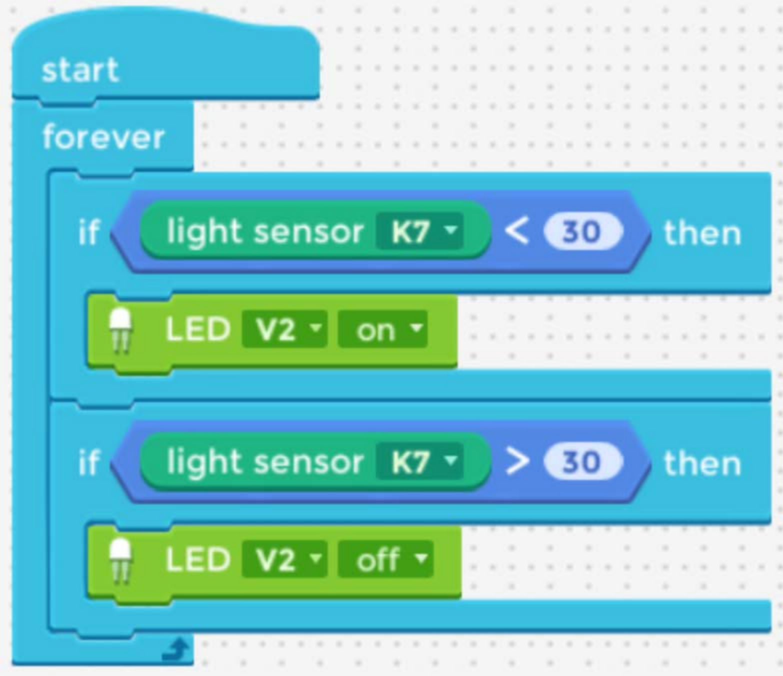
Use "if ⬭ then" block and make two "if-statements": "Turn LED on if the surroundings are dark" and "Turn LED off if light is detected."

Day 2 Map: Stage 3 Mission 3: Learn how to use a Light Sensor

This mission's goal



Let's code so that the LED turns on automatically when the light sensor detects the brightness around it.



```
start
forever
  if light sensor K7 < 30 then
    LED V2 on
  if light sensor K7 > 30 then
    LED V2 off
```

What does this program do?

____

____

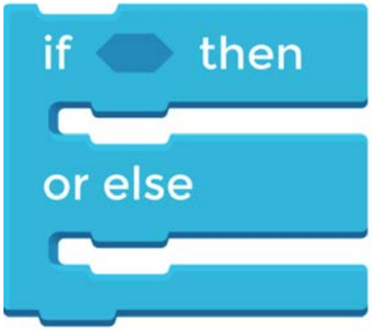
____

____

____

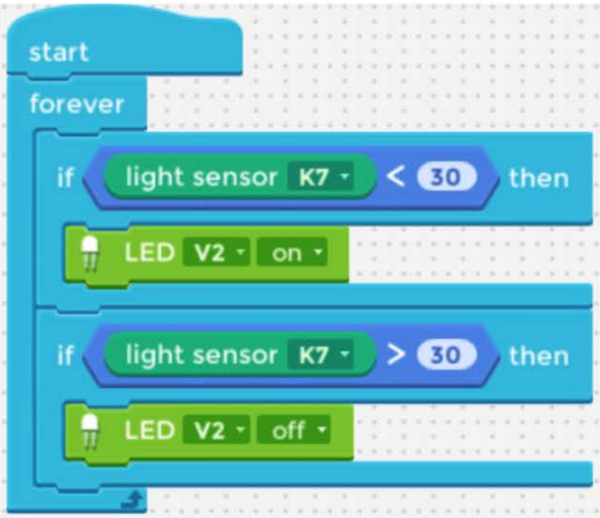
Day 2 Map: Stage 3 Mission 4: Rewrite code

This Mission's Goal: To rewrite the code with **if then or else** code block.

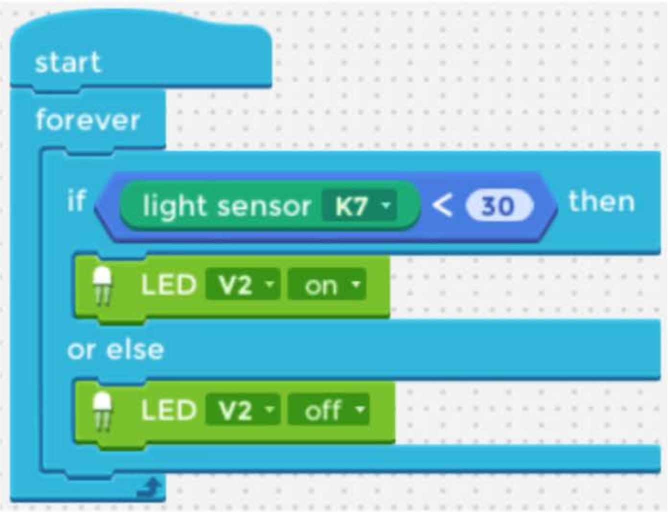


Use this block to show 2 different conditions: "If specified" and "Other than specified"

The program at the left is from Mission 3 and it uses two if then code block. It can be rewritten with if then or else code block. The updated program is at the right. Write the code in the box.



```
start
forever
  if light sensor K7 < 30 then
    LED V2 on
  if light sensor K7 > 30 then
    LED V2 off
```



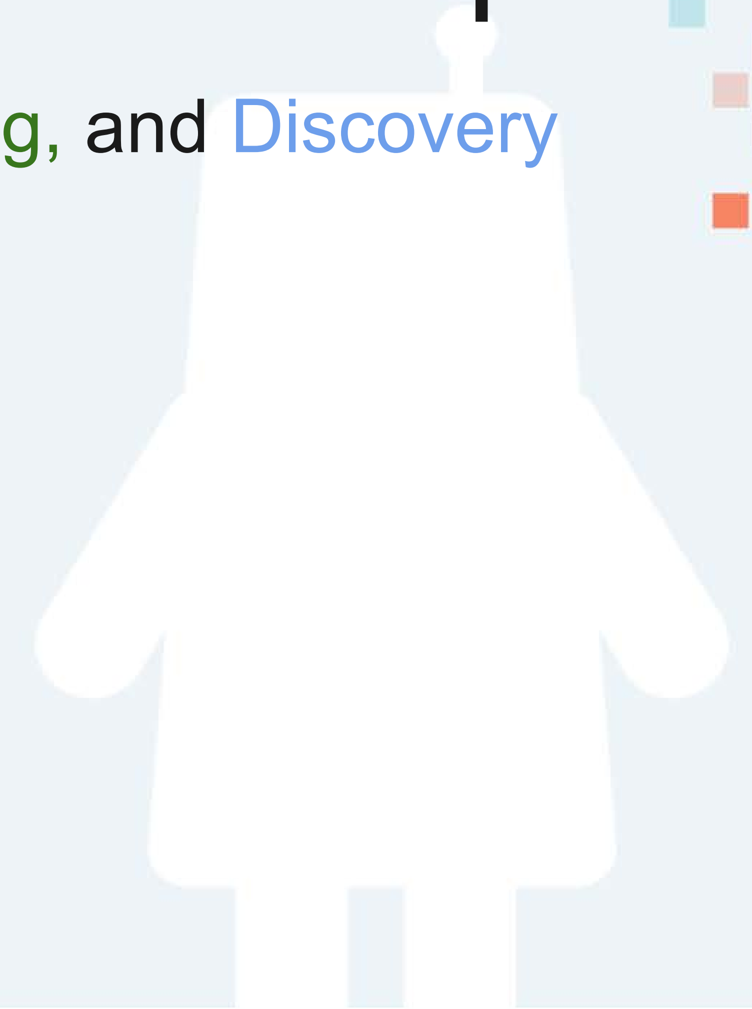
```
start
forever
  if light sensor K7 < 30 then
    LED V2 on
  or else
    LED V2 off
```

A program to turn on LED if it's dark, off if it's bright	Same program but rewrite with if...then...or else
start if light sensor K7 < 30 then LED V2 on if light sensor K7 > 30 then LED V2 off	start

KOOV

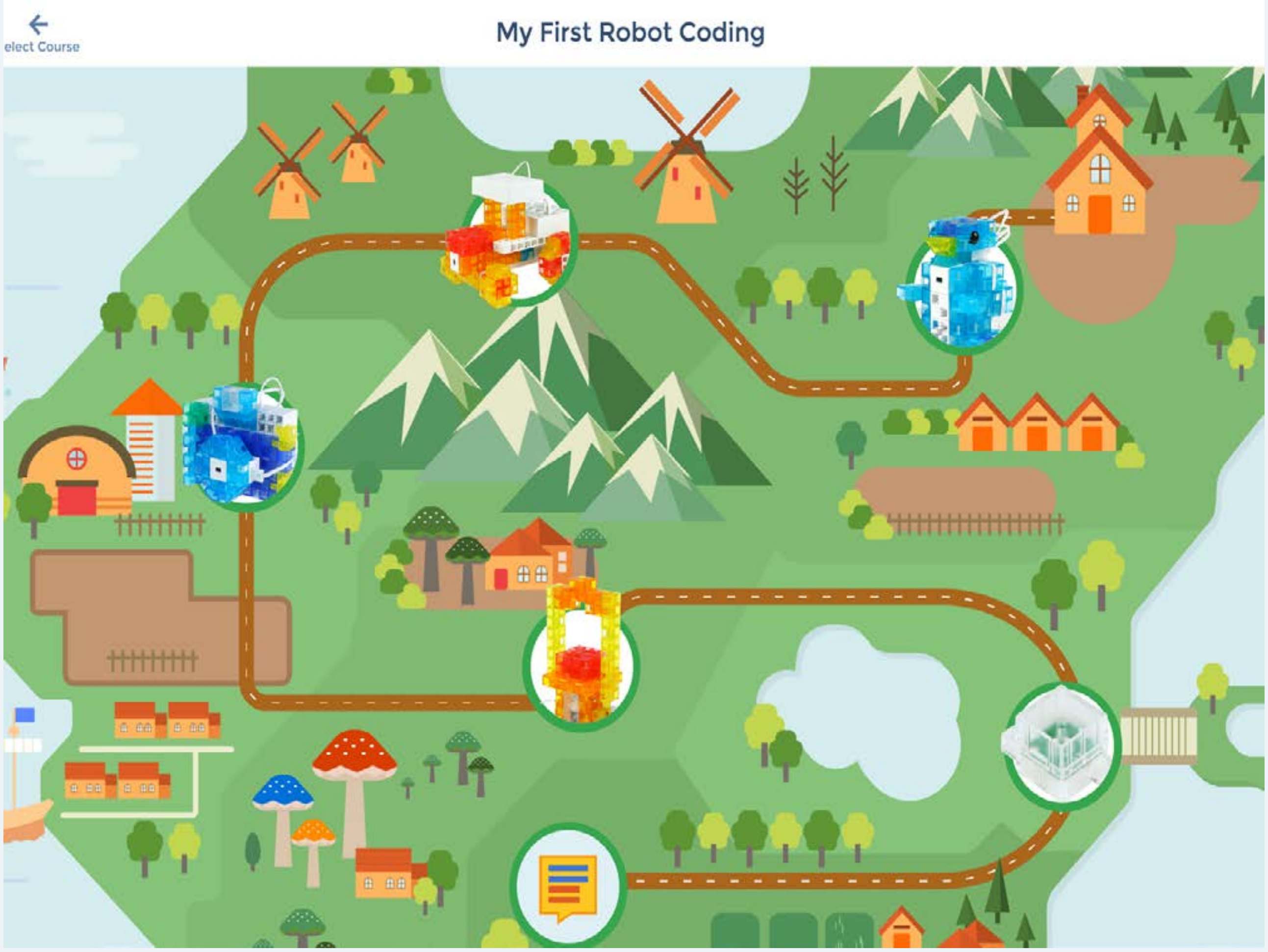
Robotics Camp

Fun, Learning, and Discovery



Fun Road Ahead

Welcome to the KOOV Robotics Camp. At this camp, you are going to put on the hat of a student robotic engineer and learn by building. You will build and program several “robots” each day,following instructions in this curriculum and from My First Robot Coding Map. Then you will be given a Design Challenge to brainstorm for better design. Each day will be filled with learning, fun, and discovery.



Week In A Glance

Day 1: KOOV Education Suite

Learn about Robotics and KOOV

Day 2: Blink and Beep!

Coding with LEDs and Buzzer

Day 3: Light and Camera

Detect Light and Distance

Day 4: Turn that Wheels

DC Motor and Robot Car

Day 5: Dancing Feet

Servo Motor and Penguin

DAY 3

Light, Action, Camera!

Day 3 Objectives

I
Introduction

Robot Vision

E
Electronics

Push Switch

IR Photoreflector

C
Coding Concepts

Function

M
Map

Camera

Intro

Robot Vision

“See” with Reflection

animation

If you throw a ball against an wall, the ball would bounce back harder if the wall is closer.

The ball would bounce back lighter or even not bounce back if the wall is further.

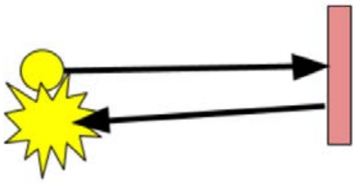
If you cannot see, then this is a good way to check if there is a wall or something in front of you.

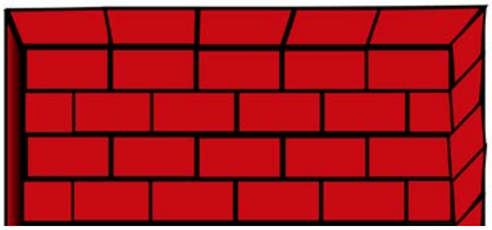
Those with vision impairment use white canes to make sounds and to scan their surrounding by comparing the difference in sounds reflected back. The louder the sounds, the closer something is.

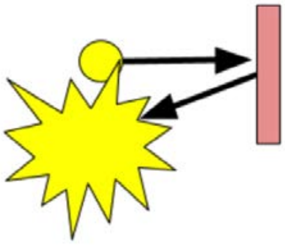
Instead of sound, robots use light to scan its surrounding. It reads the reflected light energy. The higher the energy, the closer something is.

far

close







Electronics

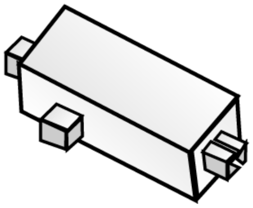
Push Switch & Core Buttons

E

KOOV Push Switch

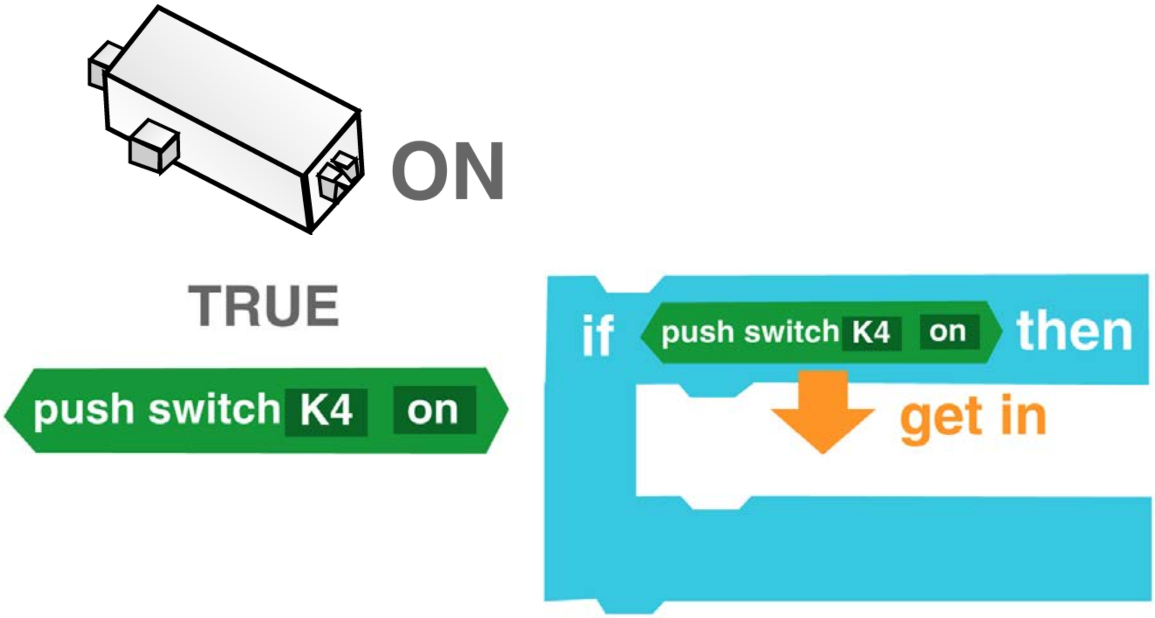
simulation

This is a KOOV Push Switch. When pressed, it sends electrical signals to KOOV Core. When not pressed, there is no electric signal.



We can read the button value with the **push switch on** sensor code. If it is pressed, the condition **push switch on** would be true; if it's not pressed, the condition would be false.

We can also process different code if the condition is true. In this code, if the button is pressed, the code inside of the if then block would be processed. If not, the program would skip the code and continue.



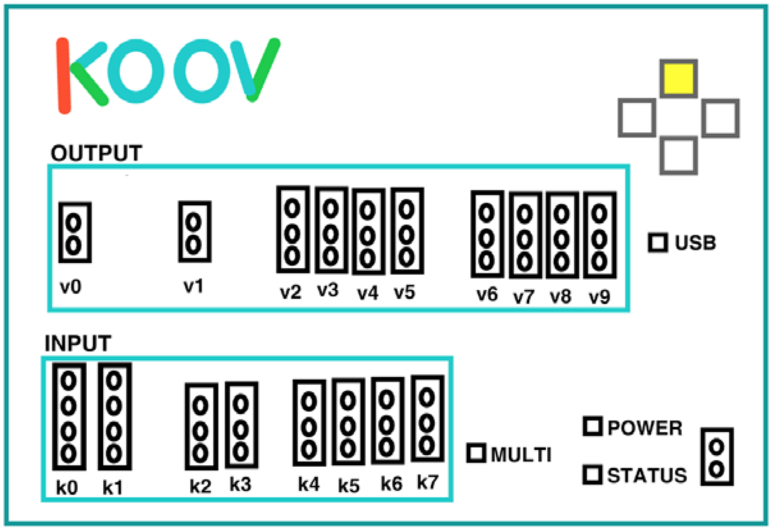
KOOV Core Buttons

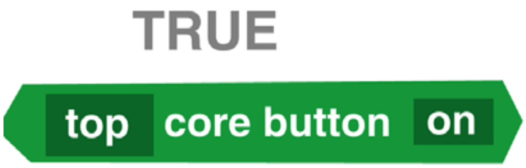
simulation

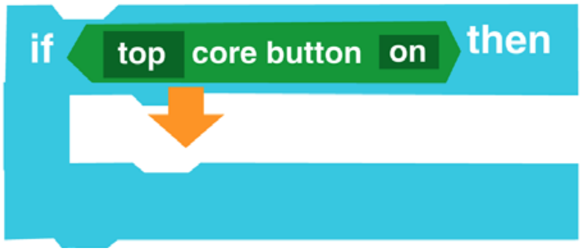
Besides Switch Button, KOOV has four Core Buttons that are on the KOOV Core.

We can read the button value with the [] core switch [] code block. If it is pressed, the [] core switch [] code block would return True; if it's not pressed, it would return False.

In the code below, if the top core button is pressed, the code inside of the if then code block would be processed. If not, the program would skip the code and continue.







Electronics

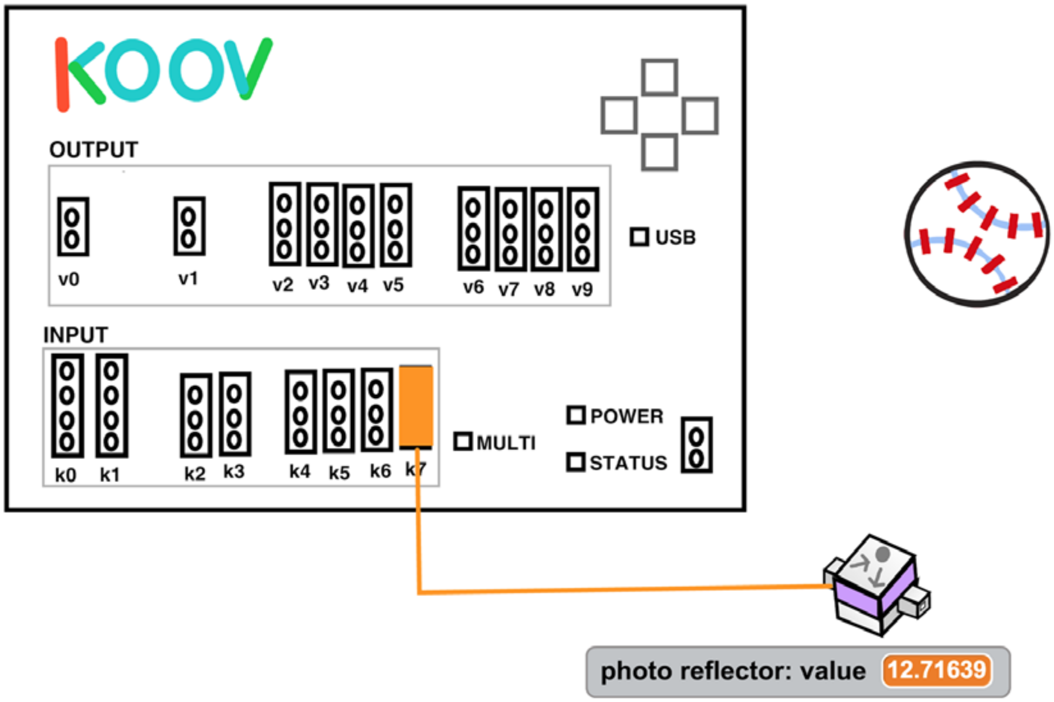
IR Photoreflector

E

KOOV Photo Reflector

simulation

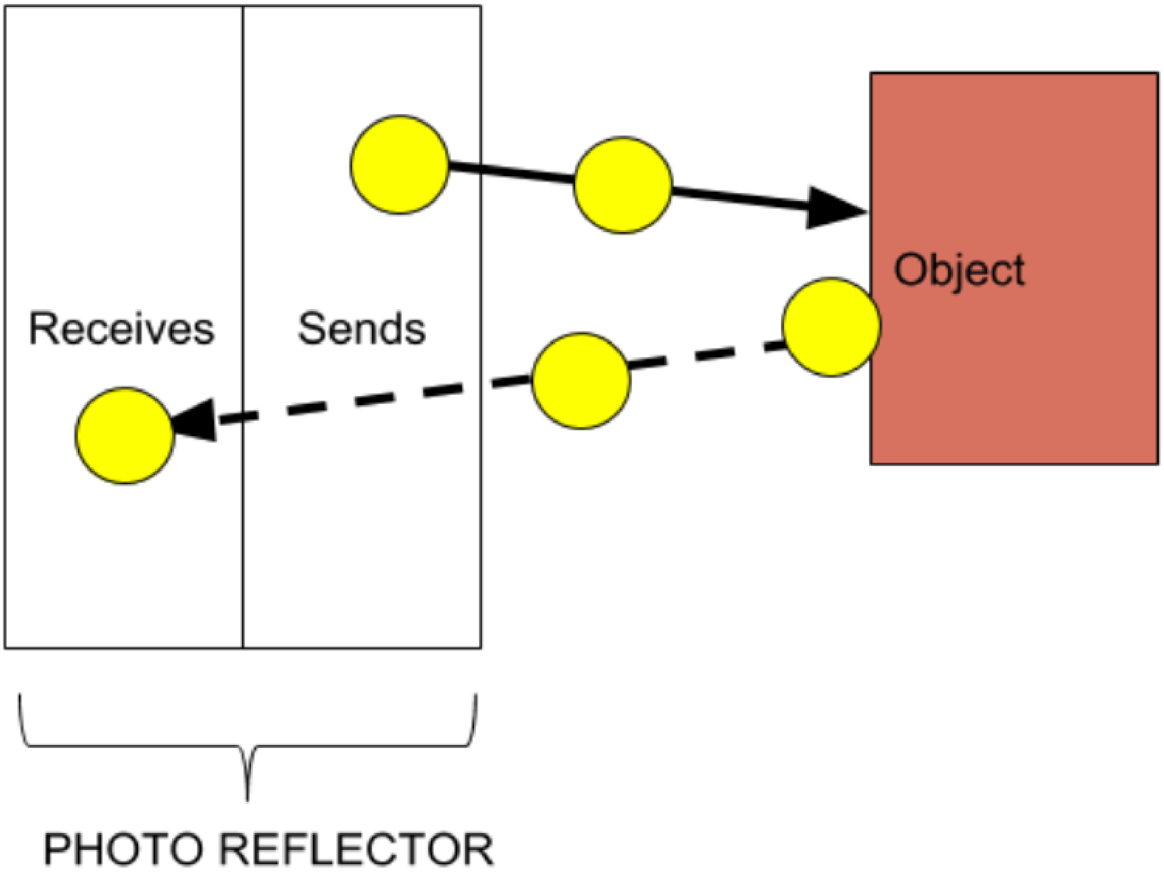
The KOOV IR Photoreflector detects if something is close. It detects energy from light that bounces off of the closest object in front of it.



How Photo Reflector Works

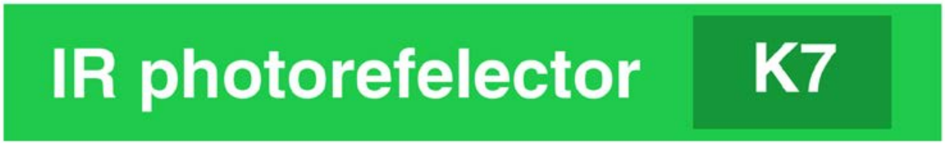
An IR Photo Reflector typically has two parts. One part shoots out light, and the other part the takes in light and converts it to energy.

The closer an object is to the IR Photoreflector, the higher the energy is; the further that object is, the lower the energy.



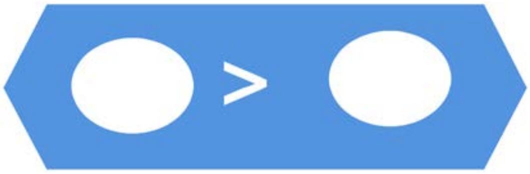
Using an IR Photoreflector

The **IR photoreflector ()** code block gives the amount of electric current detected by the KOOV IR Photoreflector device.



IR photorefelector K7

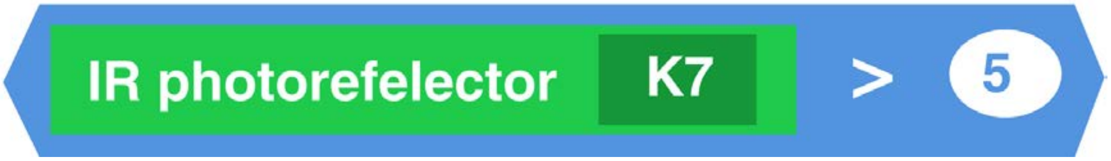
To check if something is close, use the **IR photoreflector ()** code block and the **( ) < ( )** code block.



>

Typically, when something is close, the KOOV IR photoreflector would detect an energy value larger than 5. Therefore, to check if something is close, you can check if the energy is larger than 5.

The **IP photoreflector (K7) > 5** code segment returns *True* or *False*. If something is in front of it, then it return *True*. If nothing is in front of it, then it returns *False*.

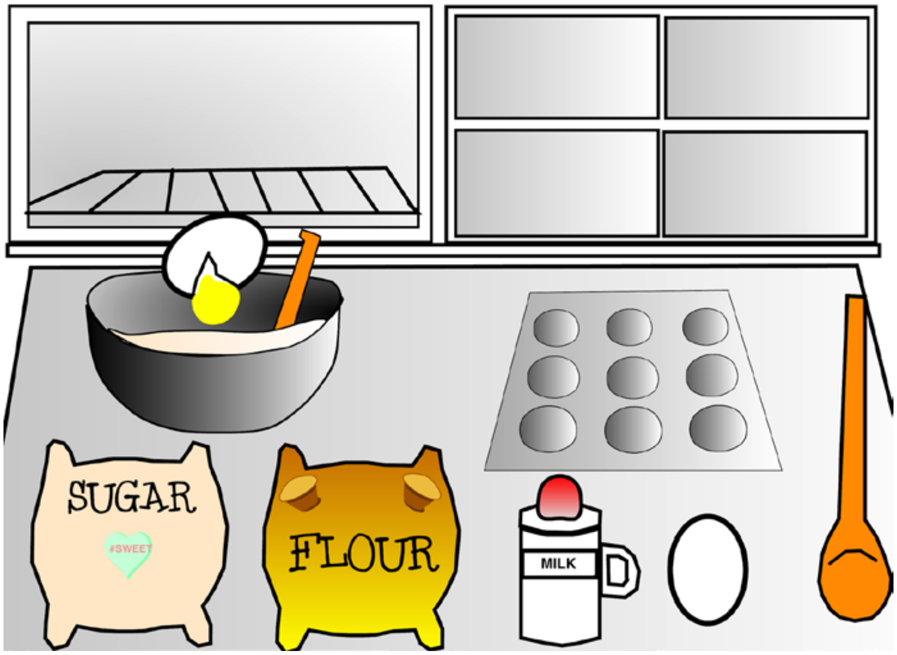


IR photorefelector K7 > 5

c
o
d
i
n
g

Functions

A Function is like a Skill



If we want to tell a robot to bake a cupcake, we can give the robot step-by-step instructions to crack an egg (hit egg on the edge of a bowl, open egg shell on top of the bowl, let egg come out into the bowl),

The instruction to crack one egg:

1. Take an egg out of the box
2. Crack egg shell on the edge of a bowl
3. Open egg shell on top of the bowl
4. Let egg come out into the bowl

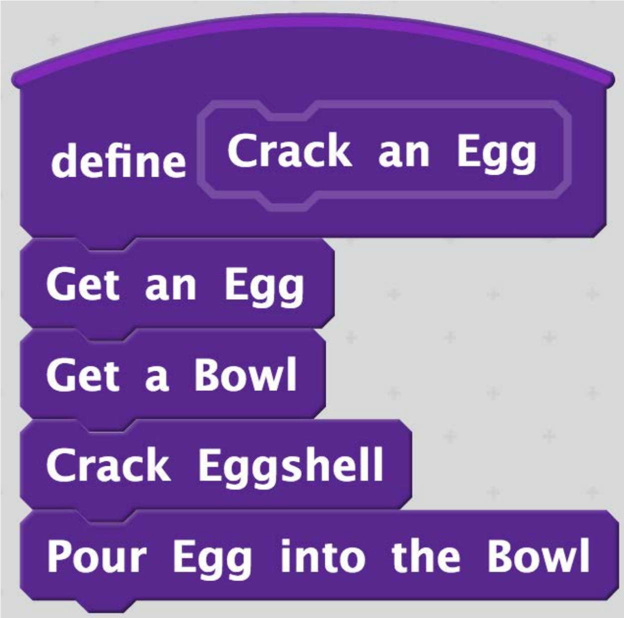
What if you need 100 eggs? 400 lines!!! And it would be hard to read what you are trying to do.

What if we put these steps in a skill, called “crack an egg”.

Such **skills** in the code are called **functions**. And to define a function is to add **step-by-step instructions** inside of this function.

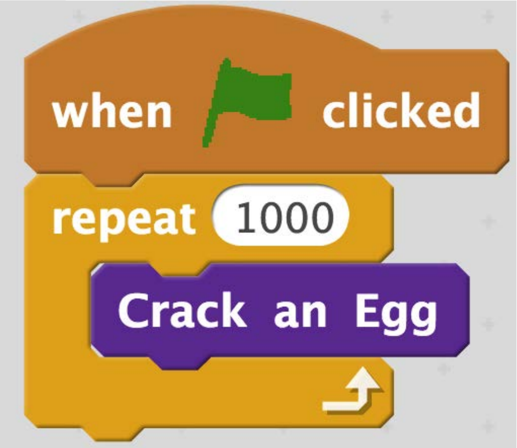
Use Functions in Code

To create a function called “Crack an Egg”, first we define the function:



```
define Crack an Egg
Get an Egg
Get a Bowl
Crack Eggshell
Pour Egg into the Bowl
```

Then to repeat it 10000 times, we can just write:



```
when [flag] clicked
repeat 1000
    Crack an Egg
```

X 1000

M
a
p

Camera

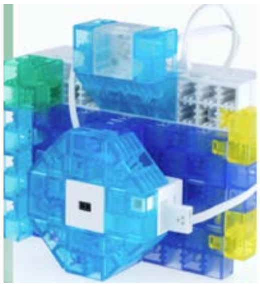
Robotic Coding Map: Camera

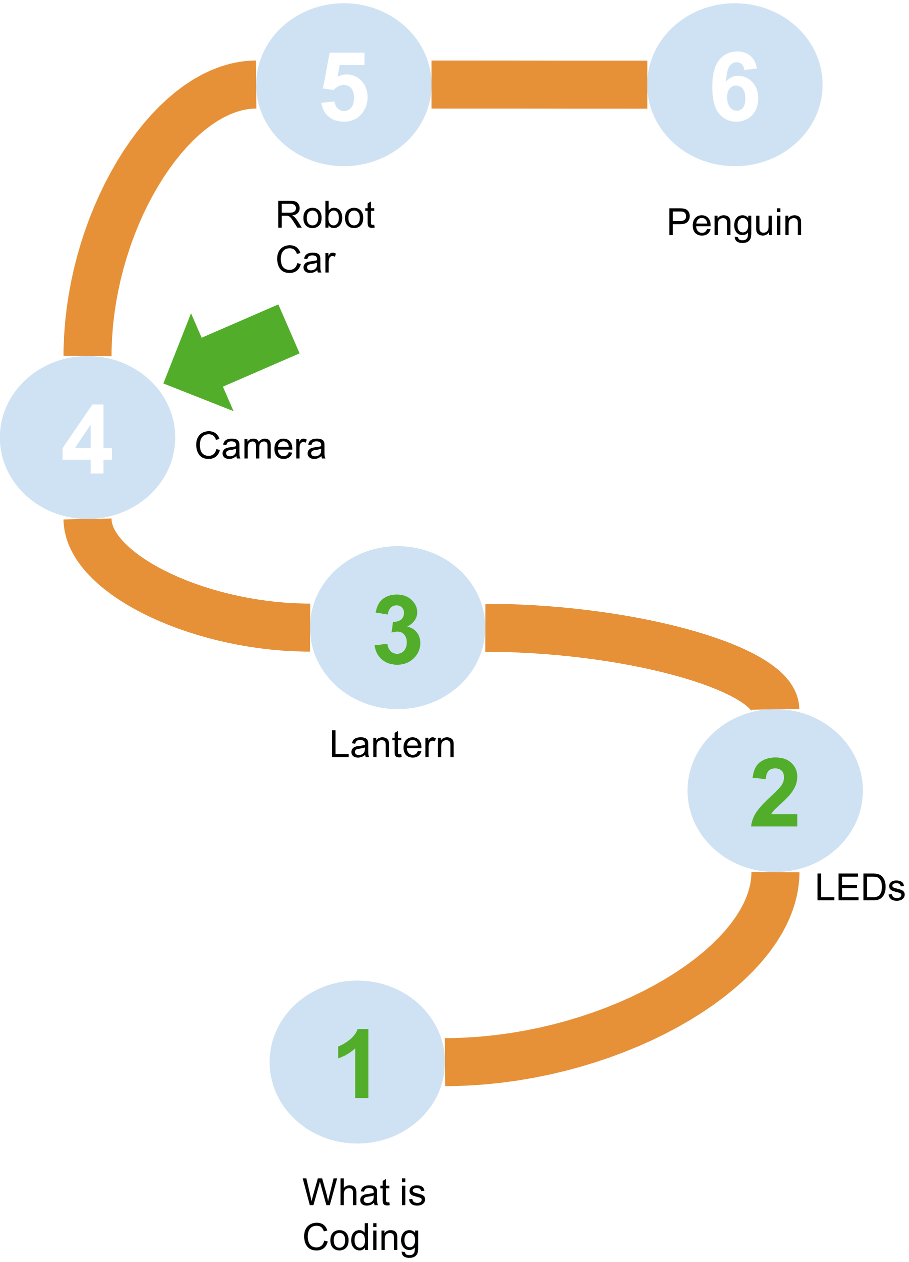
Map: My First Robotic Coding
Stage: Stage Four

Mission 1: Build a Camera
Mission 2: Use the IR photoreflector
Mission 3: Ring the buzzer
Mission 4: Code a robot camera

To go to Map:

1. Login
2. Select "Learning Course"
3. Select "My First Robot Coding"





5 Robot Car

6 Penguin

4 Camera

3 Lantern

2 LEDs

1 What is Coding

DAY 4

Robot Car

Day 4 Objectives

I	Introduction	Rotation Speed & Direction
E	Electronics	DC Motor
C	Concepts	Variables
M	Map	Tractor

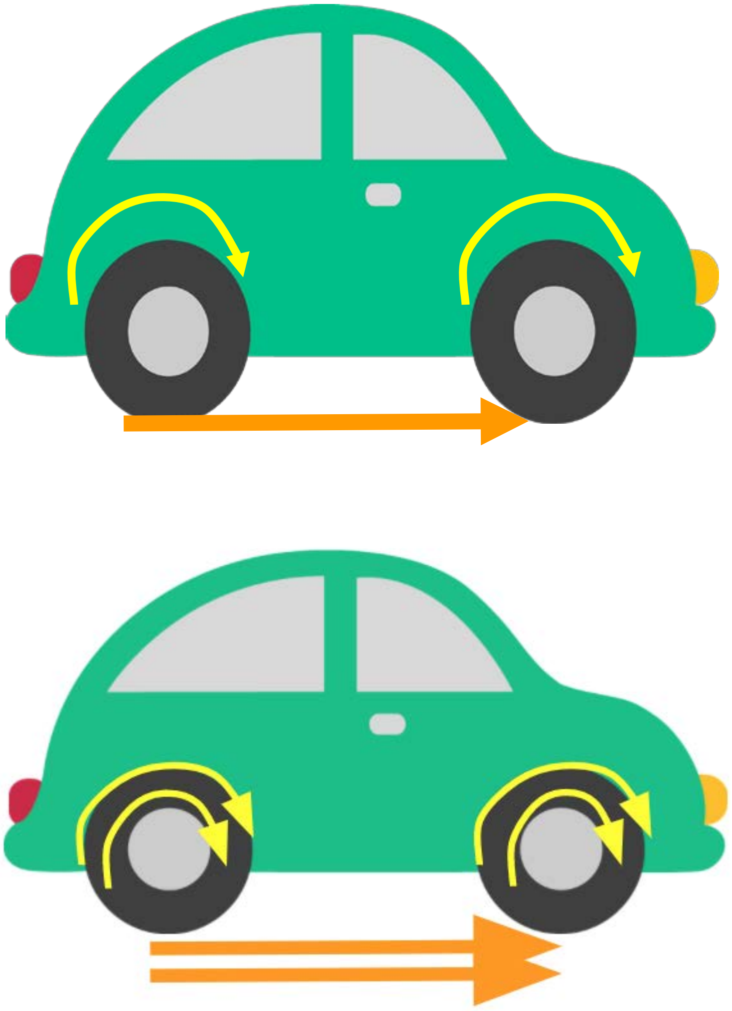
Intro

Rotation Speed & Direction

Rotation Speed

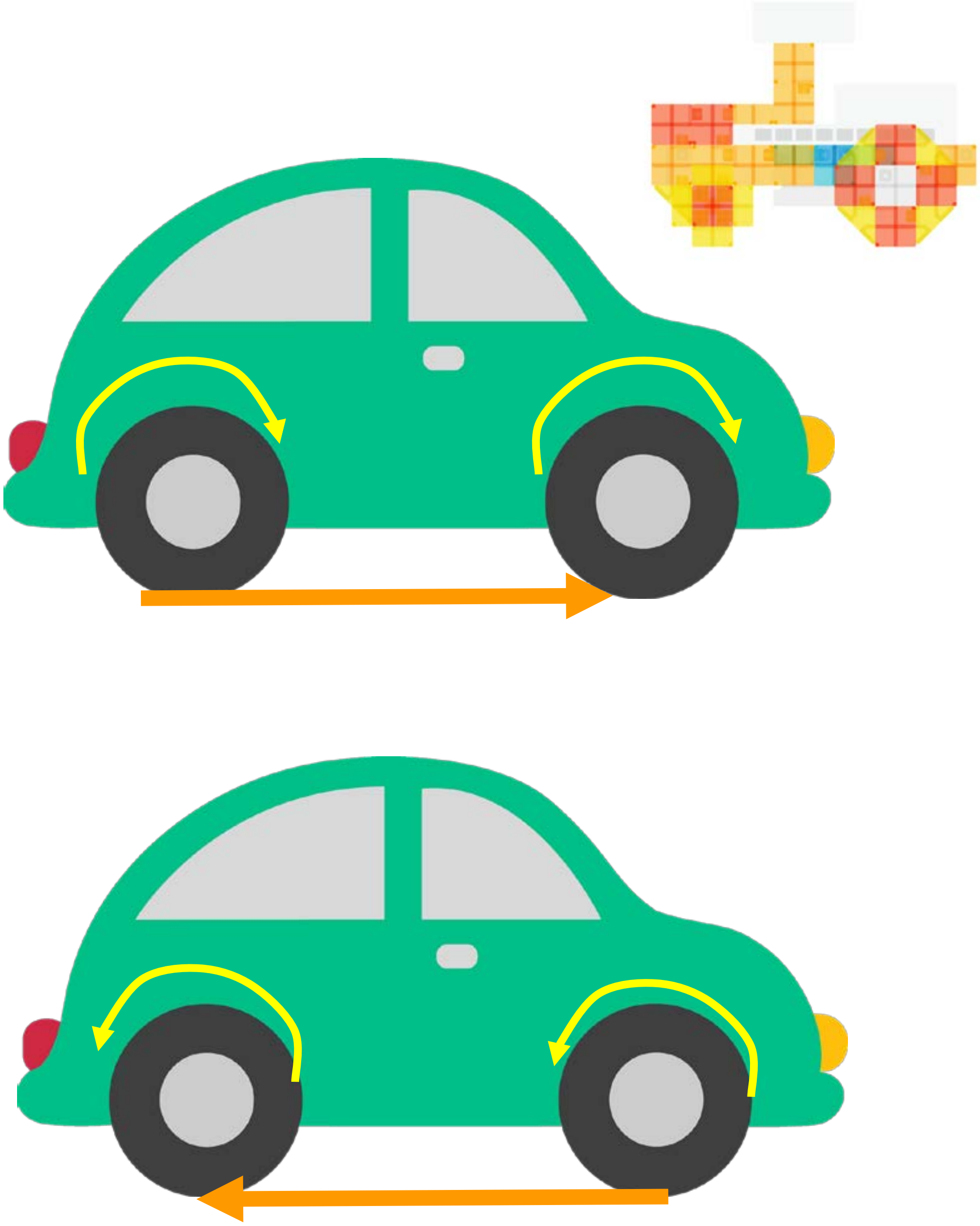
We will use the DC Motor to make a tractor in a lab later.But by adding wheels to the DC Motor and building a body on top of the wheels, rotates the motor and the tractor will move.

When the DC Motor rotates faster, the tractor moves faster.



Rotation Direction

Depending on how a DC Motor is connected, the tractor would move either forward or backward, when DC Rotor rotates clockwise. In the tractor that we will build later, the tractor moves forward when the DC Motor rotates clockwise. When it rotates counterwise, the tractor moves backward.



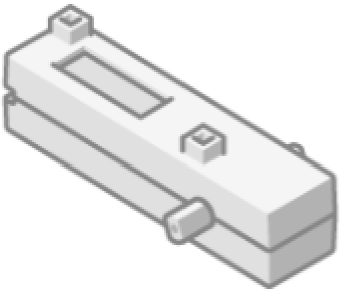
Electronics

KOOV DC Motor

E

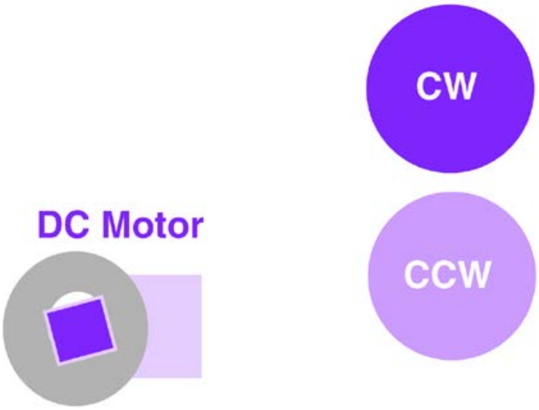
KOOV DC Motor

A KOOV DC Motor can turn clockwise and counterclockwise at a given speed. When connected with the right KOOV blocks, we can make a robot move forward and backward. But it depends on on which the DC Motor is attached.



Video: DC Motor

Video: DC Motor with Gears



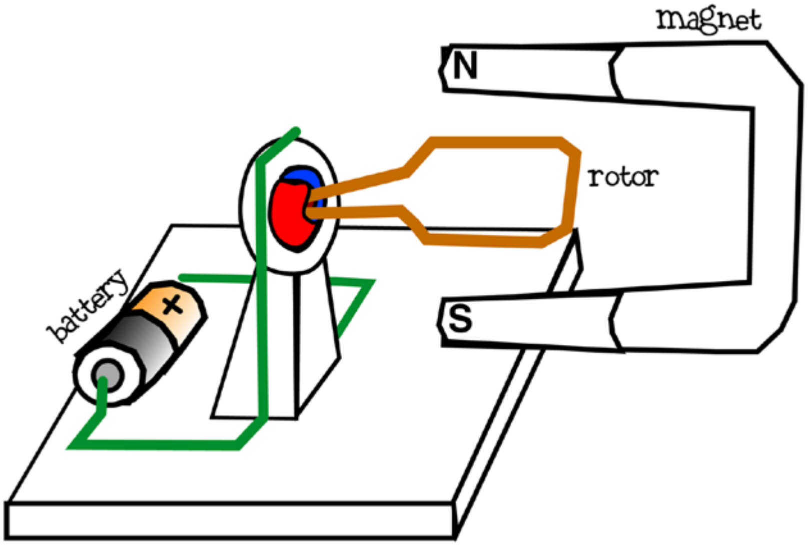
simulation

How DC Motor Work?

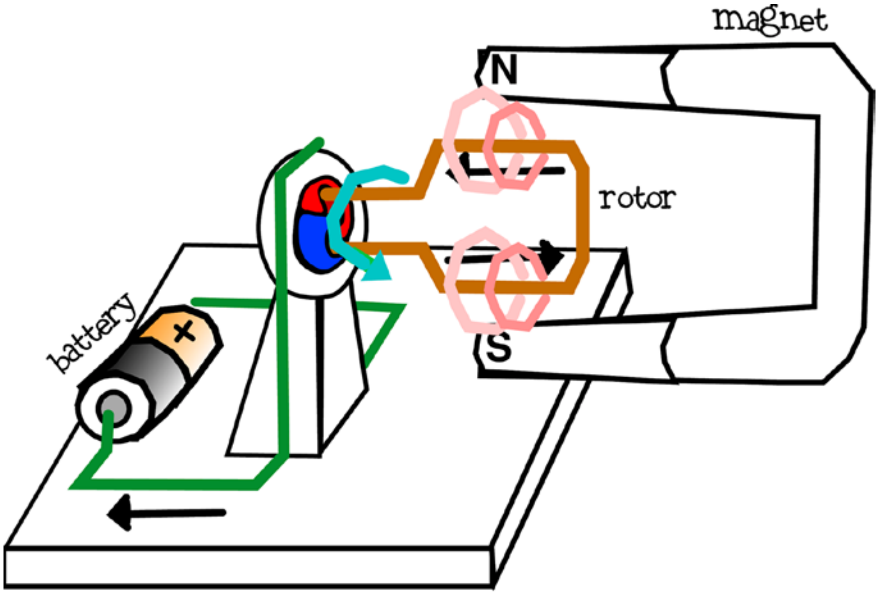
simulation

A typical DC Motor has a rotor and a magnet.

Below are two illustrations. The first is when there is no electricity through the wire rotor.



The second illustration shows electricity runs through the wire, creating a magnetic field around the wire of the rotor causing the rotor to interact with the magnet and making it spin.

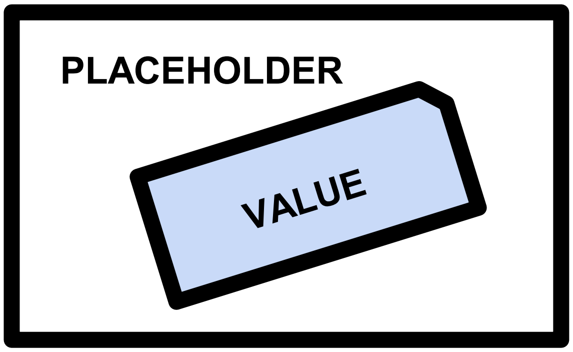


c
o
d
i
n
g

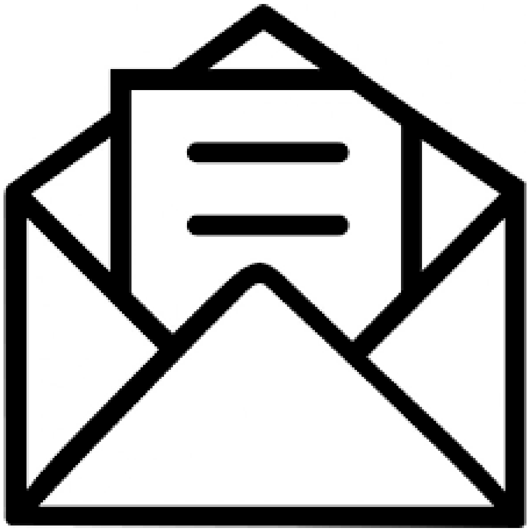
Variables

variables

What is a variable? A variable is a placeholder for a value. And that value can change.



A variable is like an envelope, and a value in a variable is like a letter in that envelope. The letter (value) in that envelope (variable) can change and can also be read.



C

Using Variables in Coding

How can we use variables in coding? In a computer game, game programmers use it to track score, the number of lives, and other information that would change.

How about using it in KOOV? We can use a variable to track the current DC Motor speed and accelerate a motor slowly until it reaches a maximum speed.

Also, if you are to design a robot cat that moves through a maze on its own, you need to give the robot instruction how long to go forward, when to turn, and turn which direction. This information can be stored in variables.

Let's take a look at using a variable called "speed".

In KOOV, you need to create that variable first. Then, set the starting value of the variable *speed*

```
set (speed) to 0
```

The DC Motor increases its rotation speed by 5 until it hits 60 per hour.

```
if (speed < 60) then
    change speed by 5
```

You will try using such variable in the Lab.

Map

Tractor

M

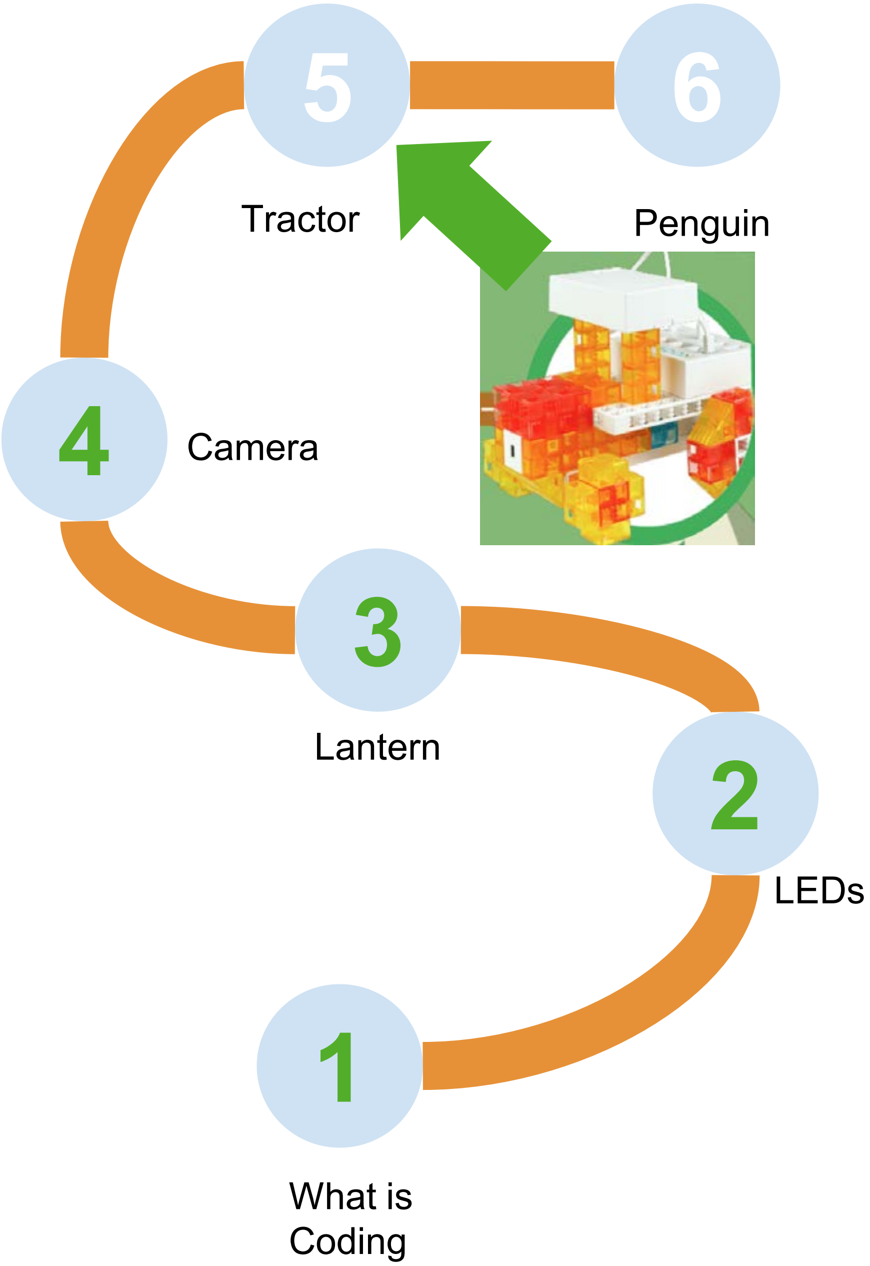
Robotic Coding Map: Tractor

Map: My First Robotic Coding
Stage: Stage Four

Mission 1: Build a tractor
Mission 2: Use the DC Motor
Mission 3: Link the Motion of DC motor and sensors
Mission 4: Use variables

To go to Map:

1. Login
2. Select "Learning Course"
3. Select "My First Robot Coding"



5 Tractor

6 Penguin

4 Camera

3 Lantern

2 LEDs

1 What is Coding

KOOV

Robotics Camp

Fun, Learning, and Discovery

Student Manual

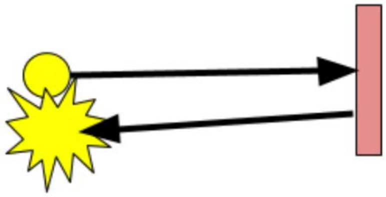
Day 3 Objectives

E Light Sensor & Photo Reflector

C Function

L Camera

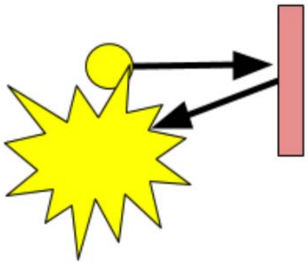
Day 3 Electronics: What is a Photo Reflector?



If you throw a ball against an wall, the ball would bounce back _______ if the wall is closer.

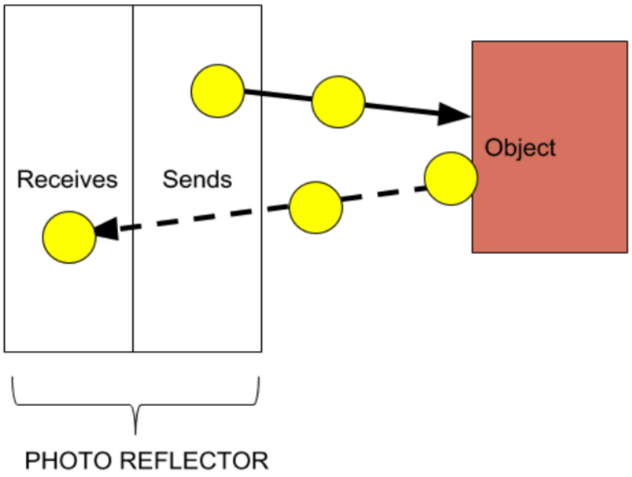
A. Harder

B. Lighter



Robots do this to check if something is close by using ________________________________

The Photo Reflector has two parts. It has a part that ___________________, and another part the takes in ________ and converts it to ________. The closer that object is from the Photo Reflector, the higher the energy would be detected; the further that object is, the lower the energy.

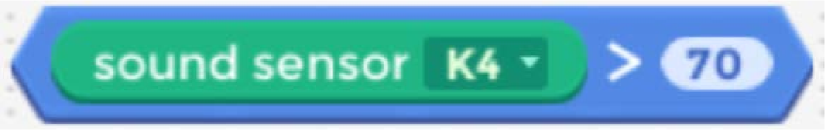


Day 3 Electronic: Use IR Photoreflector

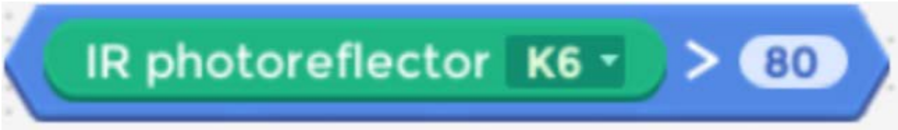
The __________________ code block returns the energy level detected at the IR Photoreflector.

The ___________________ code block returns true if one value is larger than another value.

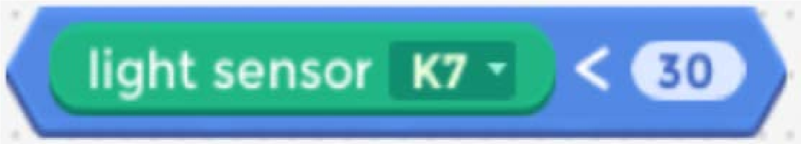
Circle the code blocks that return true if something is close to the Photoreflector



```
sound sensor K4 > 70
```



```
IR photoreflector K6 > 80
```



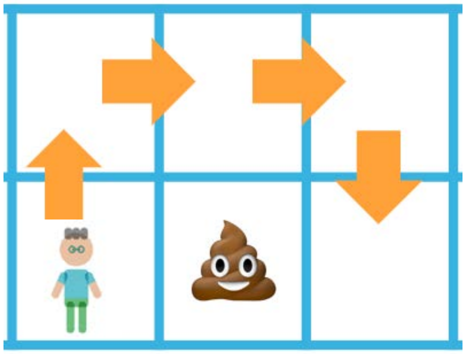
```
light sensor K7 < 30
```

Day 3 Concept: Function is like a skill

A **function** is a custom code block that has built-in code blocks. To call a function is to run the code blocks inside.

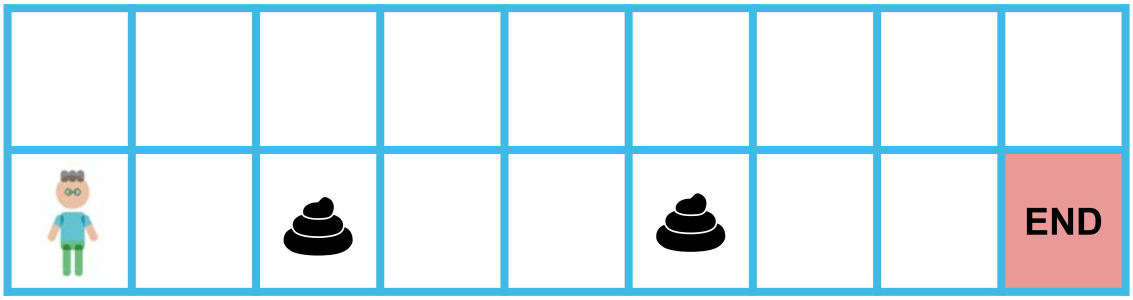
Let's create a function call "jump" to avoid 💩 in a maze.

The arrow shows the steps that one would take to jump.



A function to jump

```
function jump
      move up
      move right
      move right
      move down
```



A program to avoid 💩

```
start
move right
call function jump
move right
call function jump
move right
```

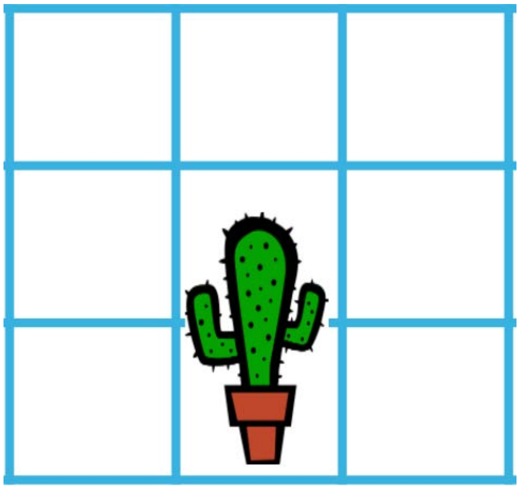
Code Blocks:

```
move up
move down
move right
move left
function jump
```

Day 3 Concept: Practice using function

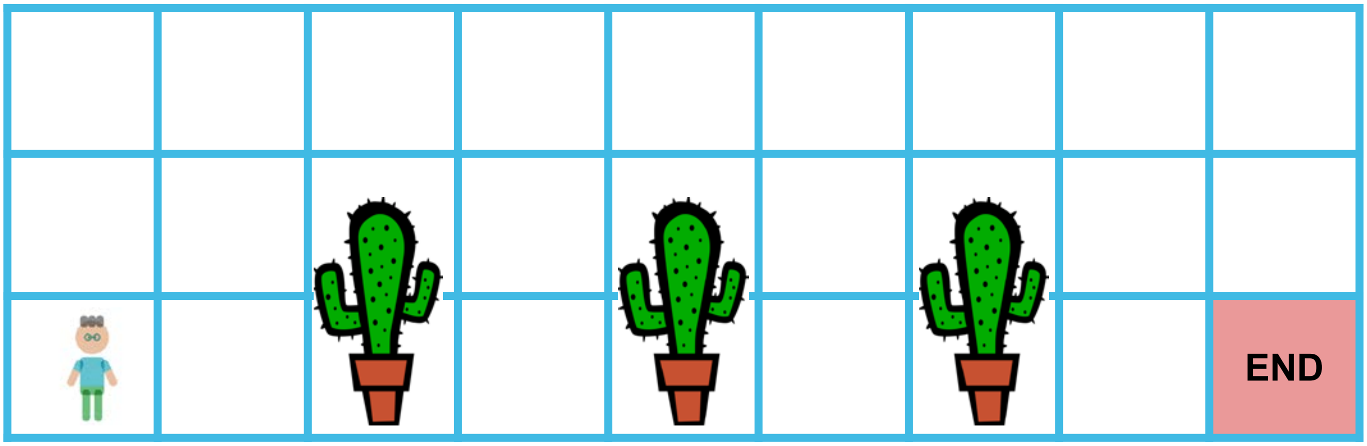
Let’s practice creating another function and call it to move in a maze.

Let’s create a function call “high_jump” to avoid 🌵



A function to do a high jump

```
function high_jump
```



A program to avoid 🌵

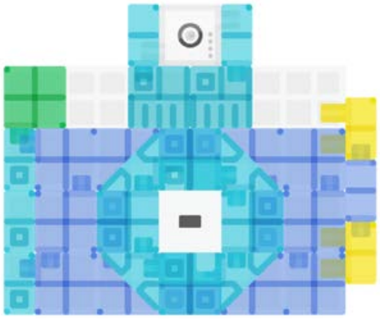
```
start
```

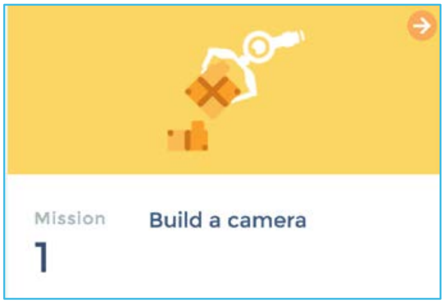
Code Blocks:

```
move up
move down
move right
move left
function high_jump
```

Day 3 Map Stage 4 Mission 1: Build a camera

This mission's goal

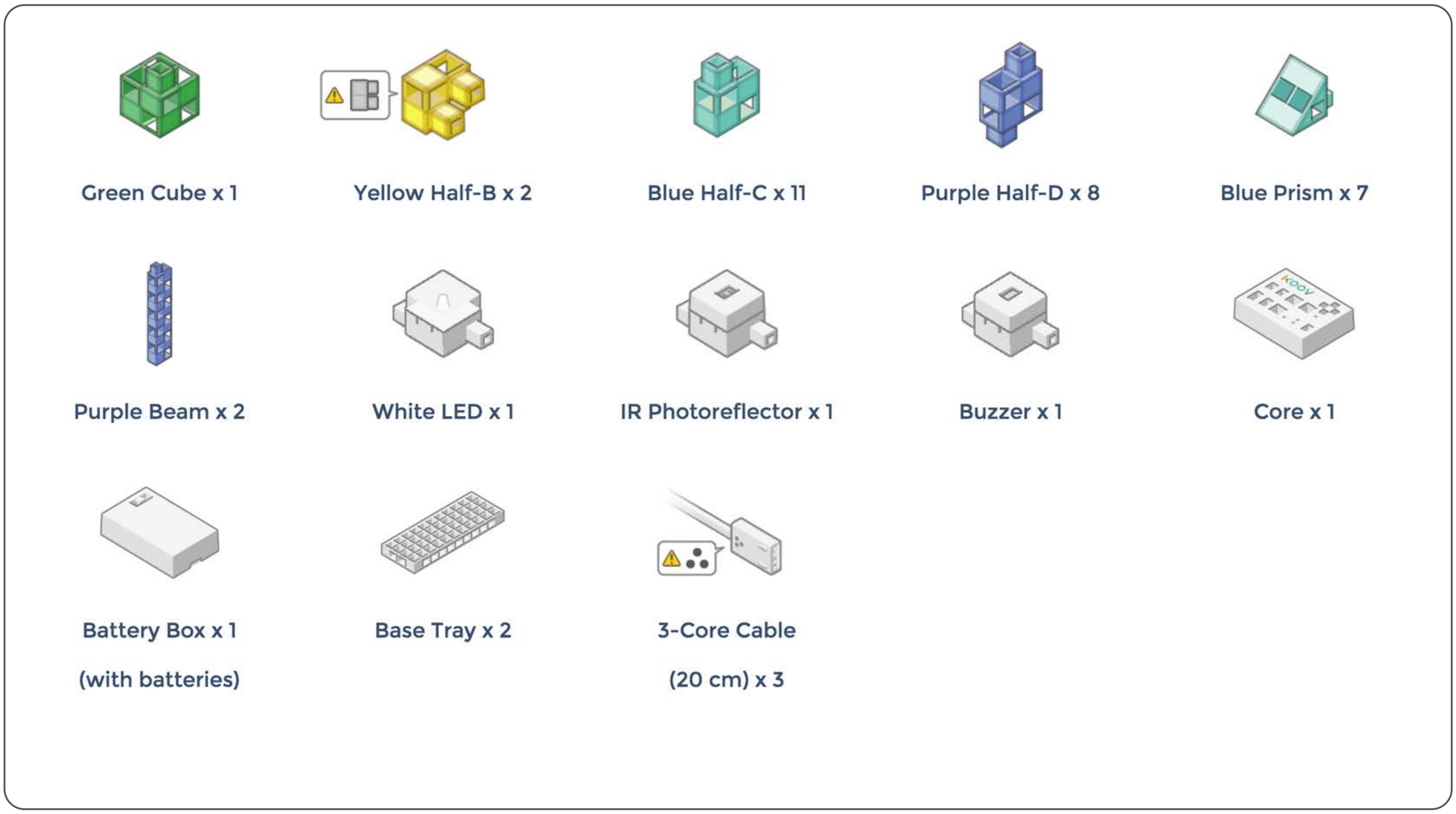




Mission 1 Build a camera

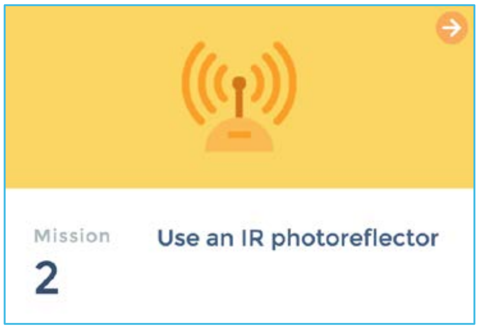
Make a camera with a IR Photoreflector, a LED, and a Buzzer, and KOOV Blocks.

Parts you need

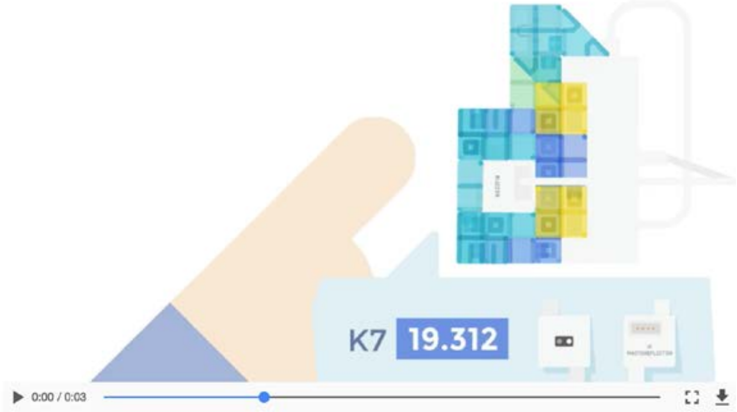


- Green Cube x 1
- Yellow Half-B x 2
- Blue Half-C x 11
- Purple Half-D x 8
- Blue Prism x 7
- Purple Beam x 2
- White LED x 1
- IR Photoreflector x 1
- Buzzer x 1
- Core x 1
- Battery Box x 1 (with batteries)
- Base Tray x 2
- 3-Core Cable (20 cm) x 3

Day 3 Map Stage 4 Mission 2: Use an IR photoreflector


Mission
2
Use an IR photoreflector

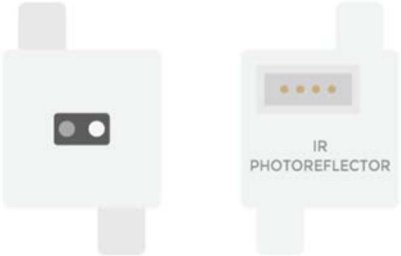
This mission's goal



Use the IR photoreflector and code so that LED flashes when something gets close enough to the front of the camera.

Let's use your camera from Mission 1.

IR Photoreflector



This sensor projects infrared rays and measures the distance to an object by detecting its reflection. The sensor value is shown by numbers 0 to 100. The closer the distance, the greater the value.

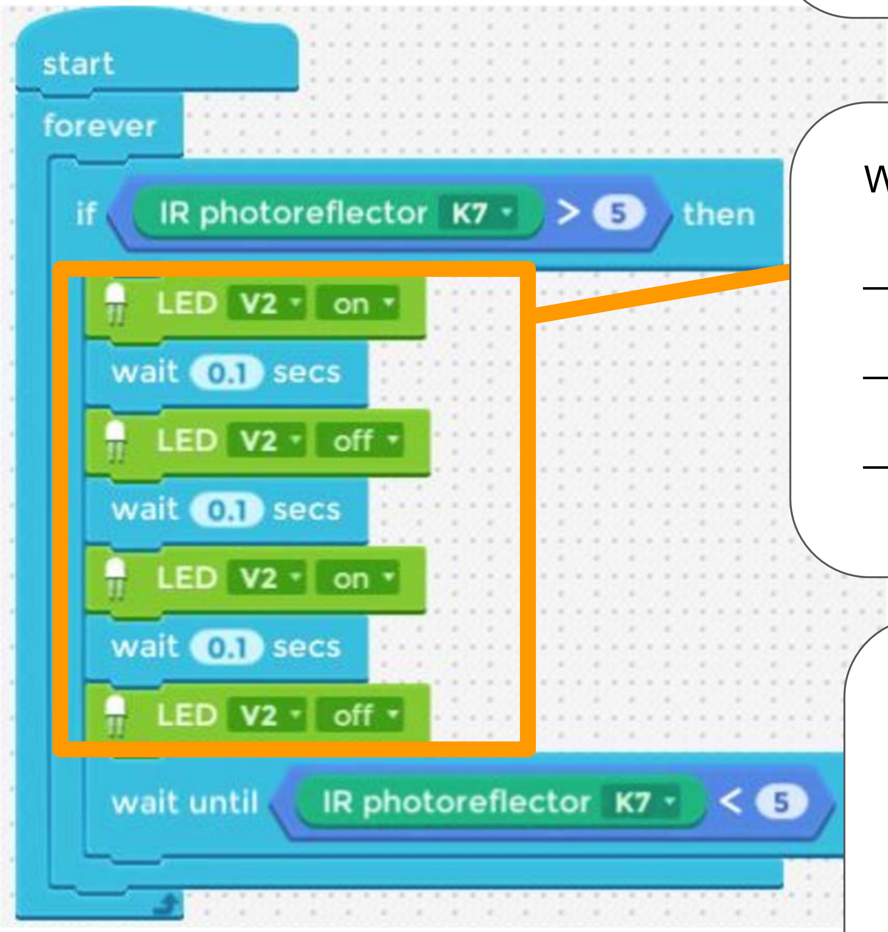
Day 3 Map Stage 4 Mission 2: Use an IR photoreflector

What does **if IR photoreflector K7 > 5 then** code statement do?

____

____

____



```
start
forever
  if IR photoreflector K7 > 5 then
    LED V2 on
    wait 0.1 secs
    LED V2 off
    wait 0.1 secs
    LED V2 on
    wait 0.1 secs
    LED V2 off
    wait until IR photoreflector K7 < 5
```

What does the code in this box do?

____

____

____

What does the **wait until IR photoreflector K7 < 5** code statement do?

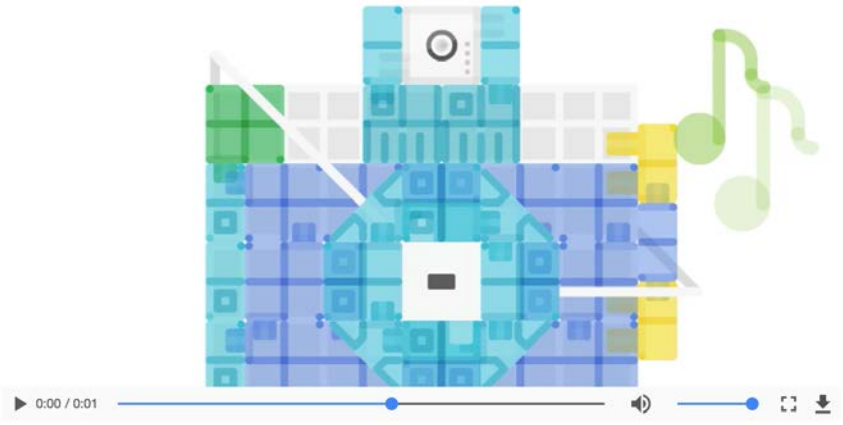
____

____

____

Day 3 Map Stage 4 Mission 3: Ring the buzzer

This mission's goal

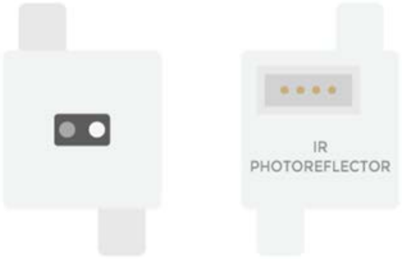


Mission
3
Ring the buzzer

Use the buzzer and make the camera make a "Beep Beep" sound.

Use your camera from Mission 1.

IR Photoreflector



Can you think of how IR Photoreflector can be used in our daily life?

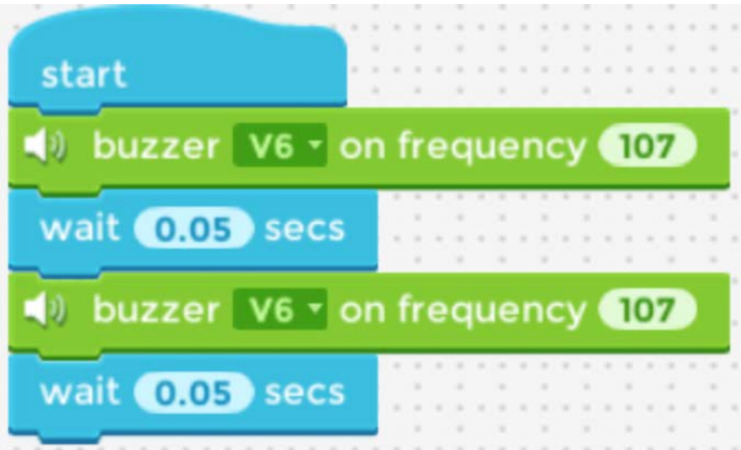
____

____

This sensor projects infrared rays and measures the distance to an object by detecting its reflection. The sensor value is shown by numbers 0 to 100. The closer the distance, the greater the value.

Day 3 Map Stage 4 Mission 3: Ring the buzzer

Mission 3 Ring the buzzer

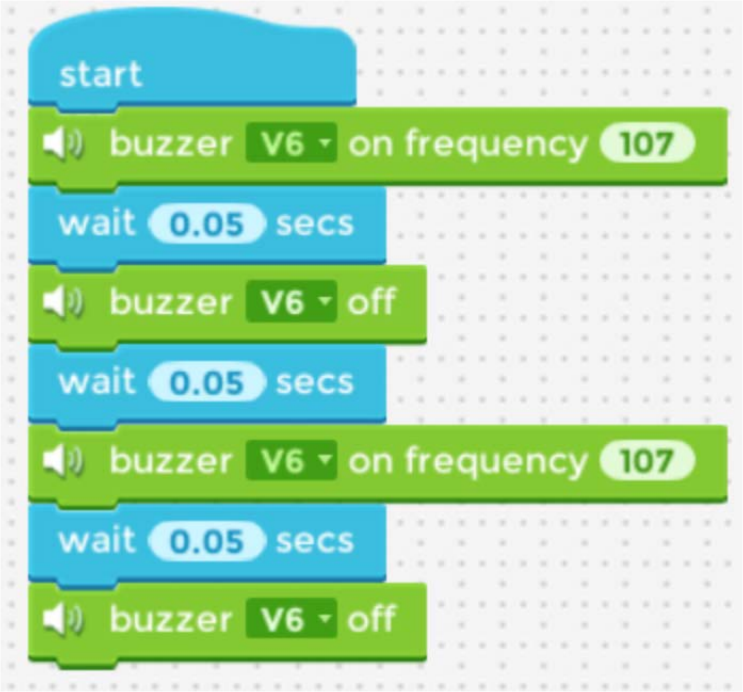

```
start
buzzer V6 on frequency 107
wait 0.05 secs
buzzer V6 on frequency 107
wait 0.05 secs
```

Does this program make the Buzzer go “beep beep”? How can you fix it?

____

____

____


```
start
buzzer V6 on frequency 107
wait 0.05 secs
buzzer V6 off
wait 0.05 secs
buzzer V6 on frequency 107
wait 0.05 secs
buzzer V6 off
```

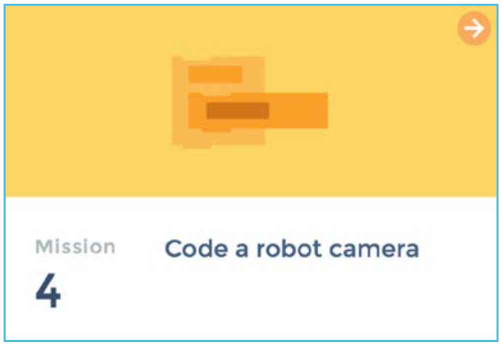
Does this program make the Buzzer go “beep beep”? How can you fix it?

____

____

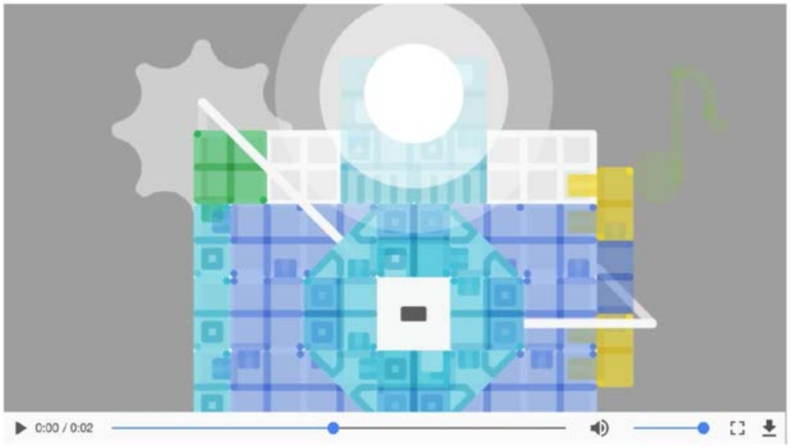
____

Day 3 Map Stage 4 Mission 4: Code a robot camera

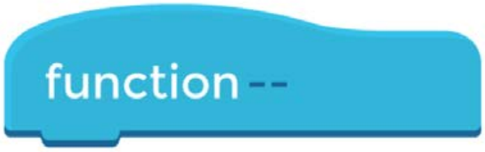


Mission 4 Code a robot camera

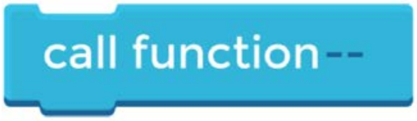
This mission's goal



Add another sound and combine with the "flash sound" and the "Beep Beep" sound to complete your robot camera.

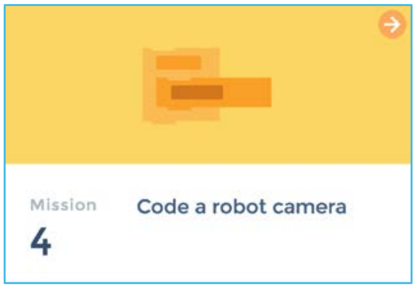


It's a block that can make your code short and simple. You can snap detailed instructions under this block.

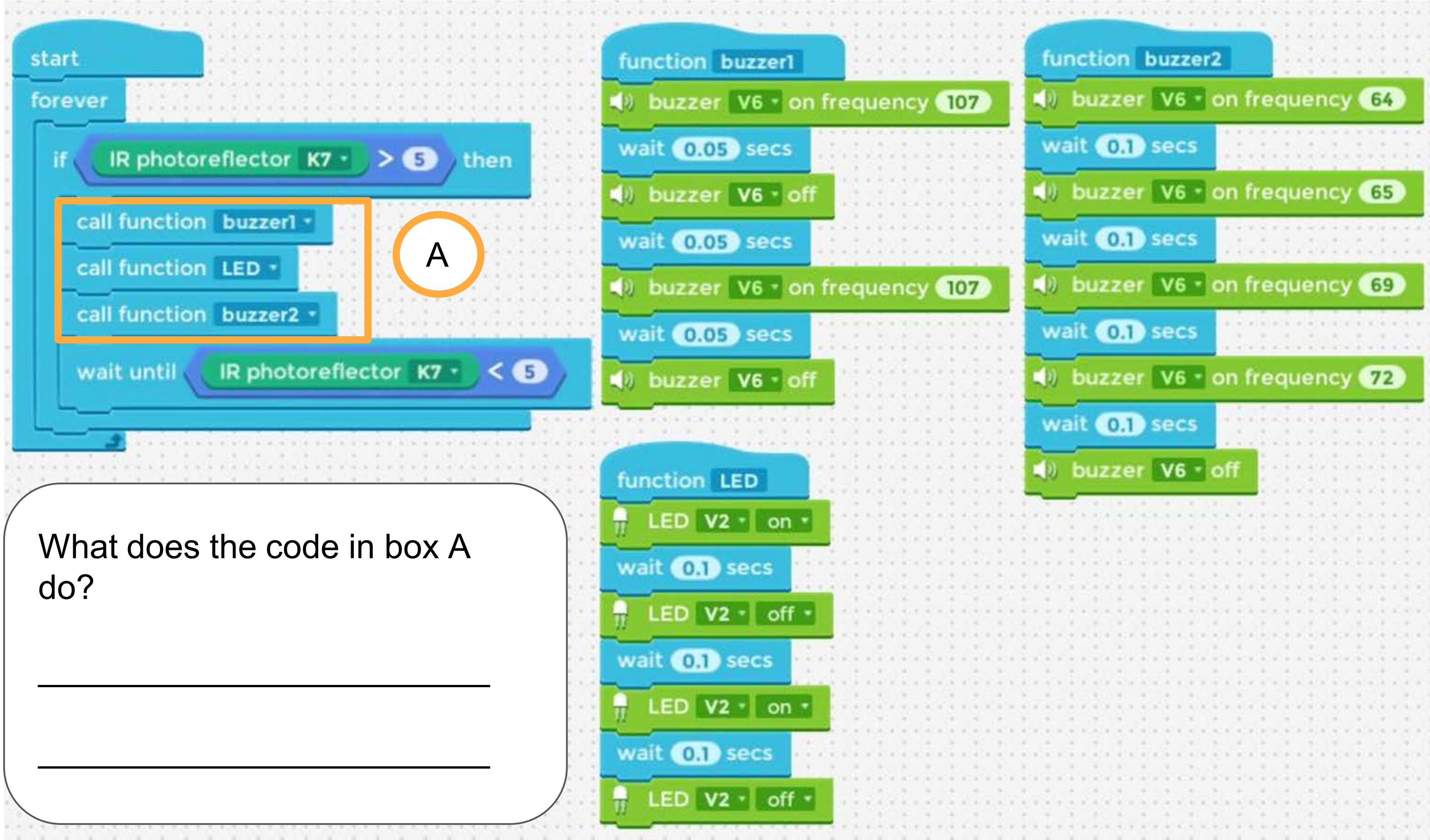


You can "call" a function from other place and run that code. Use this block for the code snapped under start .

Day 3 Map Stage 4 Mission 4: Ring the buzzer



Mission
4
Code a robot camera



```
start
forever
    if IR photoreflector K7 > 5 then
        call function buzzer1
        call function LED
        call function buzzer2
        wait until IR photoreflector K7 < 5

function buzzer1
buzzer V6 on frequency 107
wait 0.05 secs
buzzer V6 off
wait 0.05 secs
buzzer V6 on frequency 107
wait 0.05 secs
buzzer V6 off

function LED
LED V2 on
wait 0.1 secs
LED V2 off
wait 0.1 secs
LED V2 on
wait 0.1 secs
LED V2 off

function buzzer2
buzzer V6 on frequency 64
wait 0.1 secs
buzzer V6 on frequency 65
wait 0.1 secs
buzzer V6 on frequency 69
wait 0.1 secs
buzzer V6 on frequency 72
wait 0.1 secs
buzzer V6 off
```

What does the code in box A do?

____

____

List all three function names shown above:

____

Functions are used to _________ (Select One)

A. Make code easier to read when some instructions are repeated

B. To stop the code

C. To make the code run faster

Day 4 Objectives

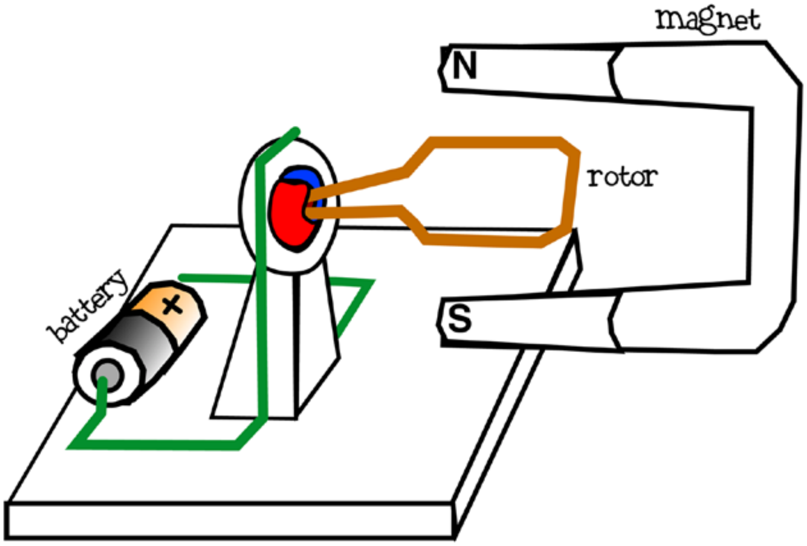
E DC Motor

C Variables

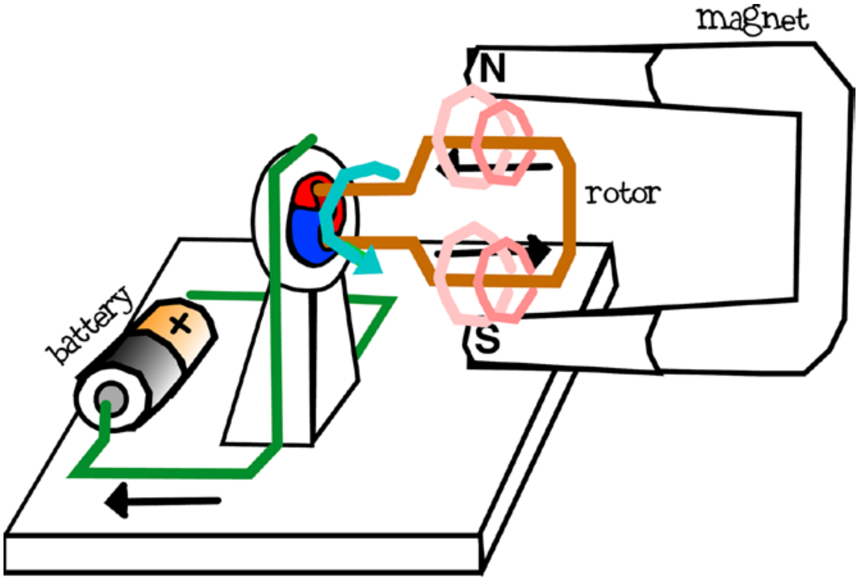
L Robot Car

Day 4 Electronics: How DC Motor Work?

A DC Motor has a **rotor** and a **magnet**. Below are two illustrations. The first is when there is no electricity through the wire rotor.



The second illustration shows electricity runs through the wire, creating **magnetic field** around the wire of the rotor, and causing the rotor to interact with the magnet, making it spin.



Day 4 Concept: variables

A **variable** is a placeholder for a **value** that can change. A _____________ is like an envelope, and a _________ in it is like a letter. The letter in that envelope can change and can also be read.

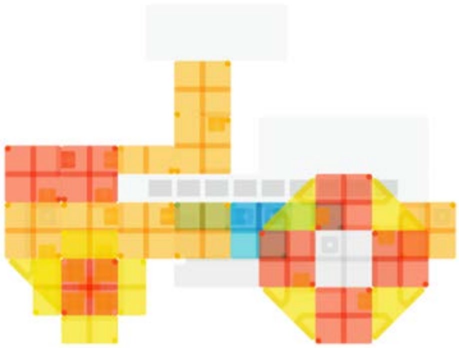
In KOOV codes, we can use variables to track changing values in our program. For example, we can track the rotation speed of a DC Motor. We can update a variable and also read from that variable.

Day 4 Map Stage 5 Mission 1: Build a tractor

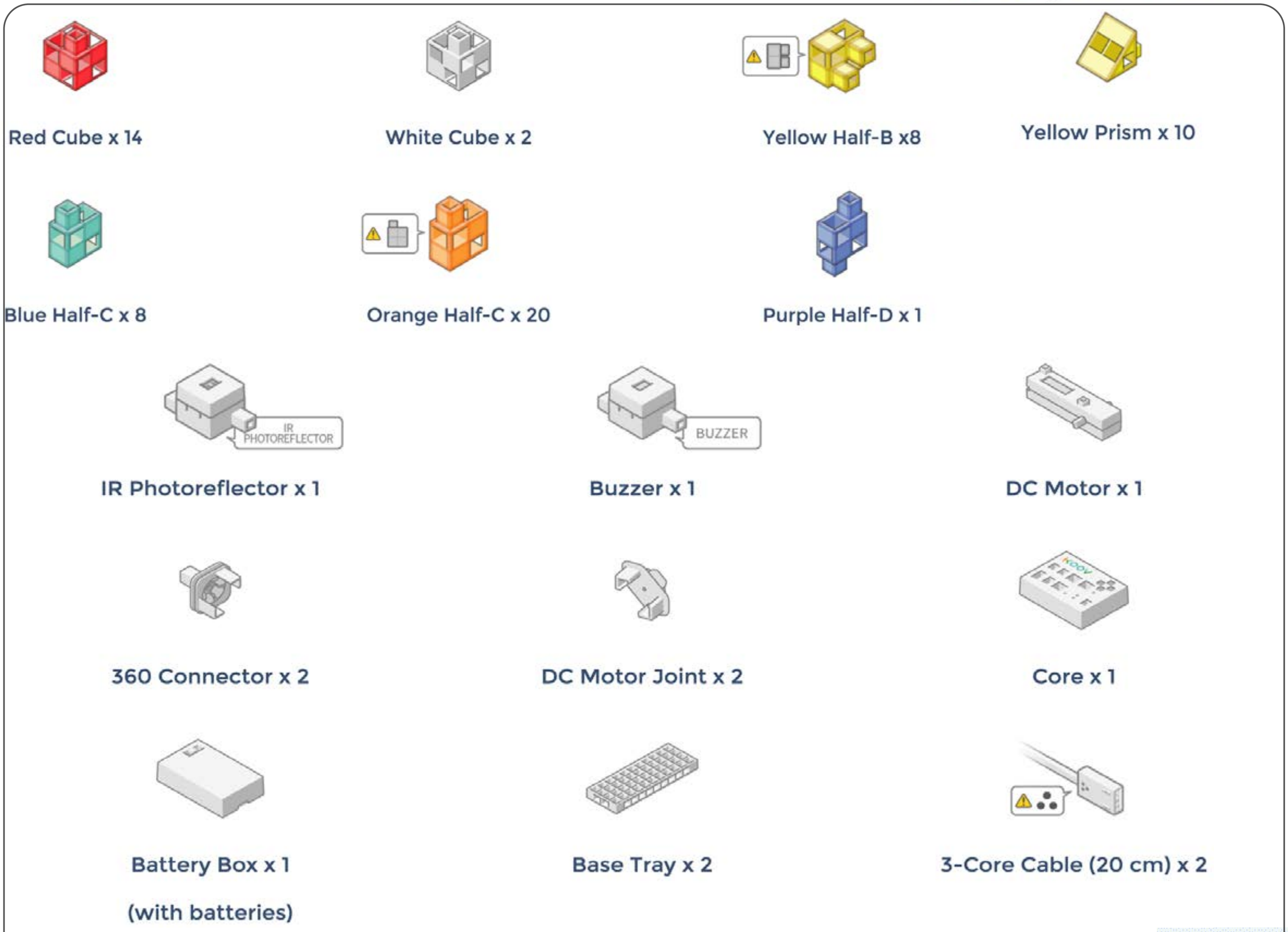
This mission's goal

Build a tractor with DC Motor, 360 Connector, DC Motor, IR Photoreflector, Buzzer, and KOOV Blocks.

Mission
1
Build a tractor



Parts you need



Red Cube x 14

White Cube x 2

Yellow Half-B x8

Yellow Prism x 10

Blue Half-C x 8

Orange Half-C x 20

Purple Half-D x 1

IR Photoreflector x 1

Buzzer x 1

DC Motor x 1

360 Connector x 2

DC Motor Joint x 2

Core x 1

Battery Box x 1

(with batteries)

Base Tray x 2

3-Core Cable (20 cm) x 2

Day 4 Map Stage 5 Mission 2: Use the DC Motor

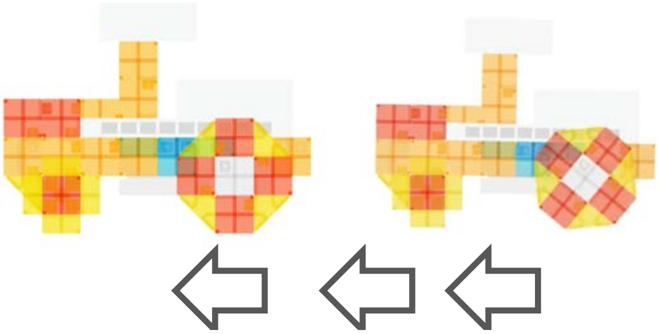
This mission's goal

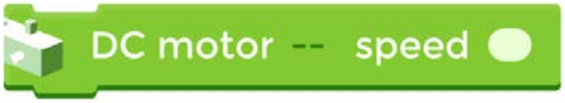
Use the DC motor to move the tractor back and forth.

Mission

2

Use the DC motor

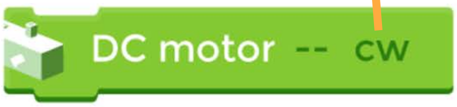




DC motor -- speed

This block shows the rotating speed of DC motor, using numbers from 0 to 100. The greater the number, the faster the rotating speed. Use a number greater than 10 for clockwise rotation.

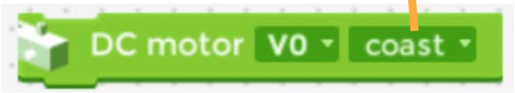
cw: clockwise
ccw: counterclockwise



DC motor -- cw

This block shows the direction of rotation. Keep in mind that the direction may change depending on the way you attach the DC motor.

coast: stop slowly
brake: stop immediately



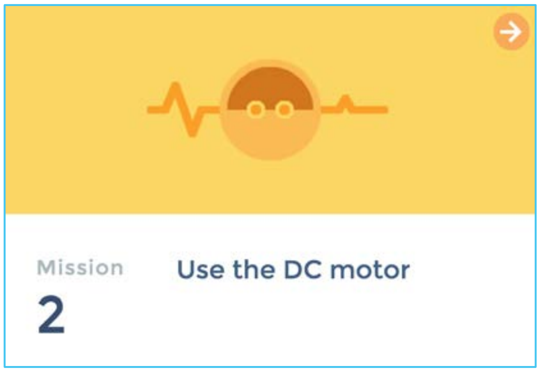
DC motor V0 ▾ coast ▾

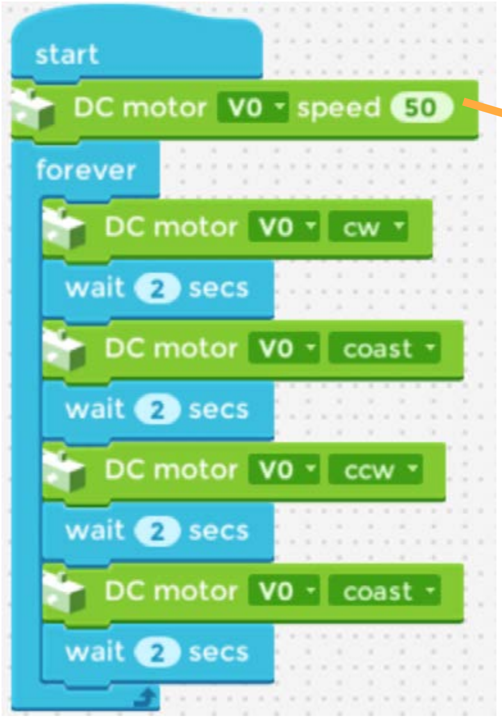
Coast mode is to stop adding force to the DC motor rotation, and brake is to stop the rotation completely.

Day 4 Map Stage 5 Mission 2: Use the DC Motor

This mission's goal

Use the DC motor to move the tractor back and forth.



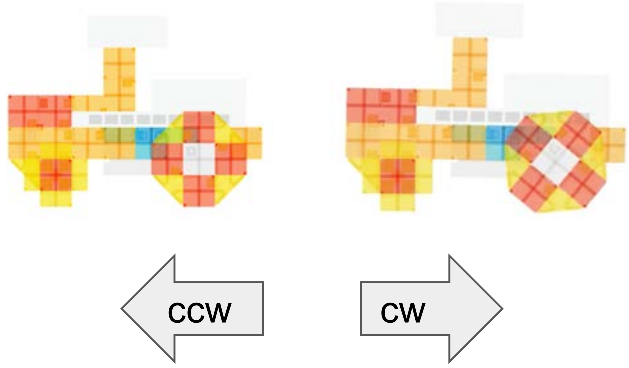


```
start
DC motor V0 speed 50
forever
    DC motor V0 cw
    wait 2 secs
    DC motor V0 coast
    wait 2 secs
    DC motor V0 ccw
    wait 2 secs
    DC motor V0 coast
    wait 2 secs
```

What would happen if this line is inside the **forever** code block?

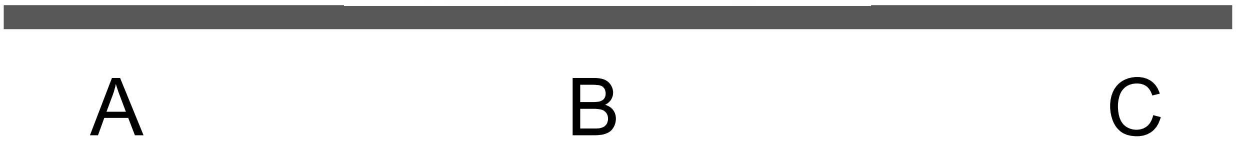
____

____



Which point do you think the tracker would end up? A, B, or C?

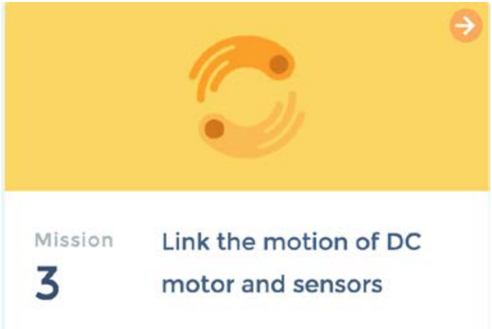
____



Day 4 Lab Stage 5 Mission 3: Link the motion of DC motors and sensors

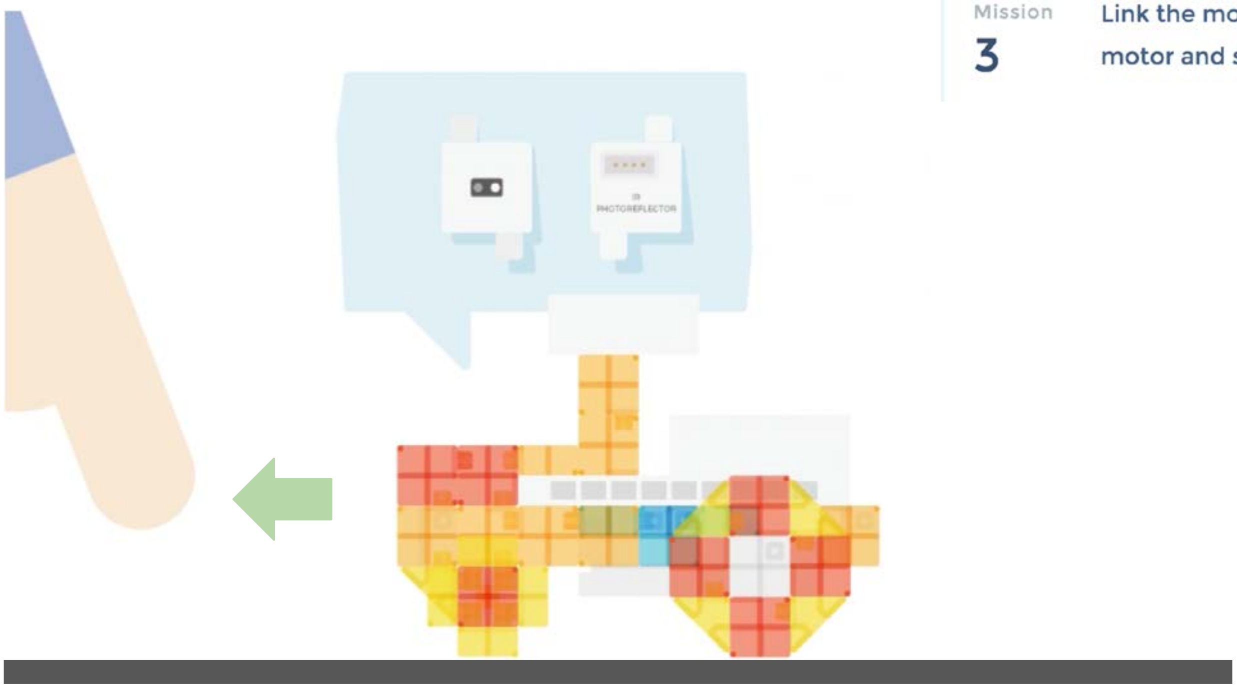
This mission's goal

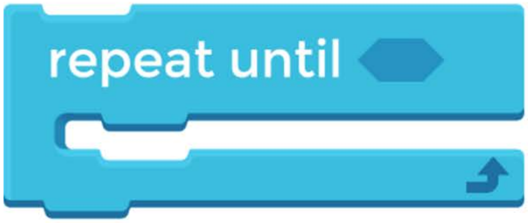
Control the tracker's movement with an IR photoreflector. Add a back-up sound.

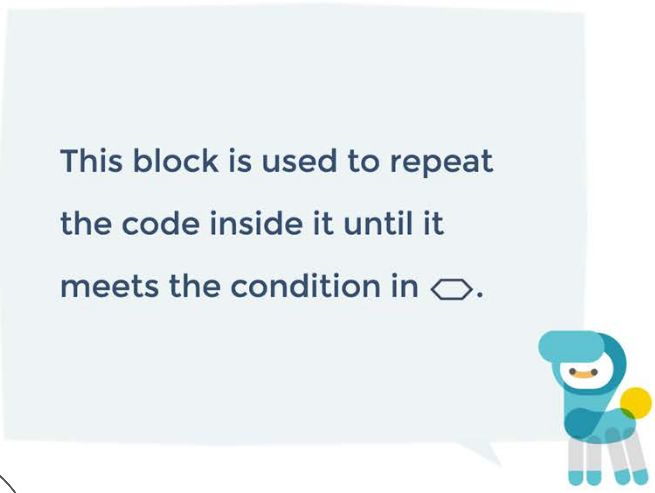


Mission 3

Link the motion of DC motor and sensors







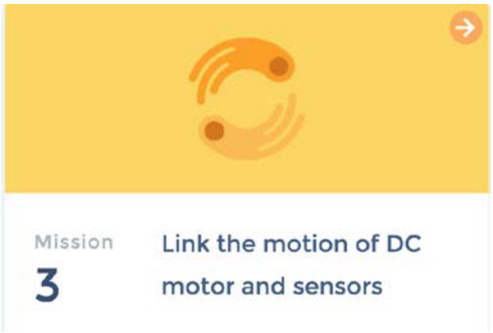
This block is used to repeat the code inside it until it meets the condition in ⬡.

The repeat until code block make the movement smoother

Day 4 Lab Stage 5 Mission 3: Link the motion of DC motors and sensors

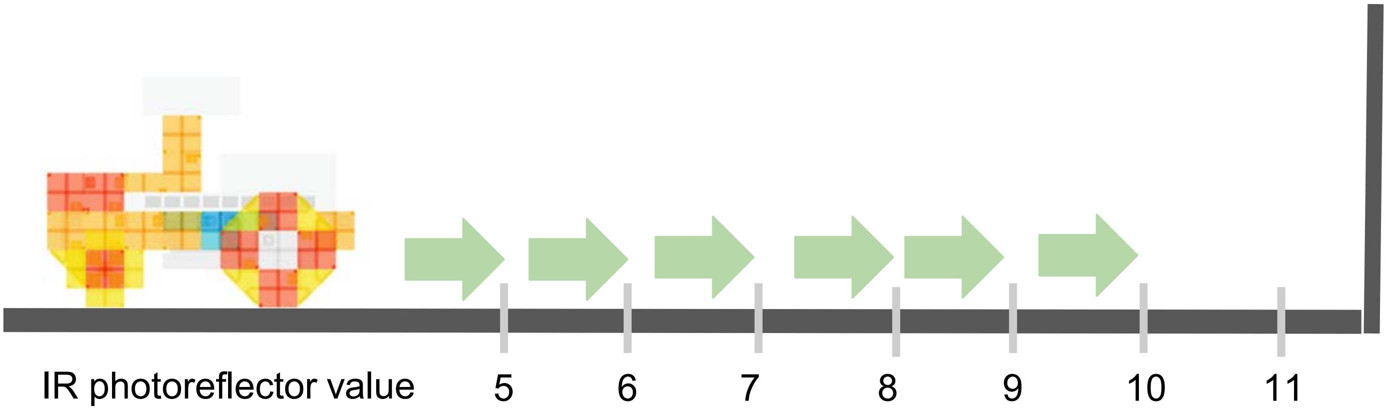
This mission's goal

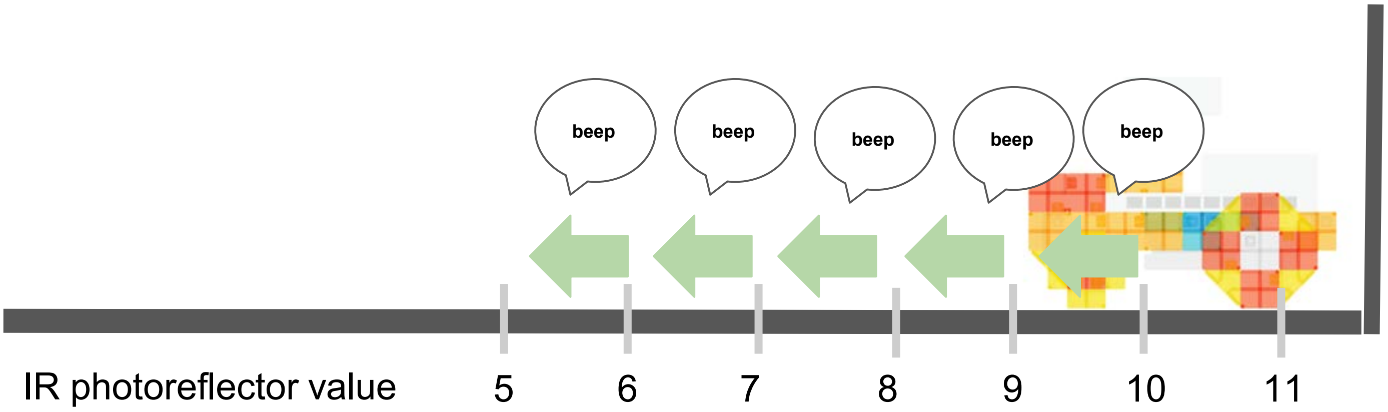
Control the tracker's movement with an IR photoreflector. Add a back-up sound.

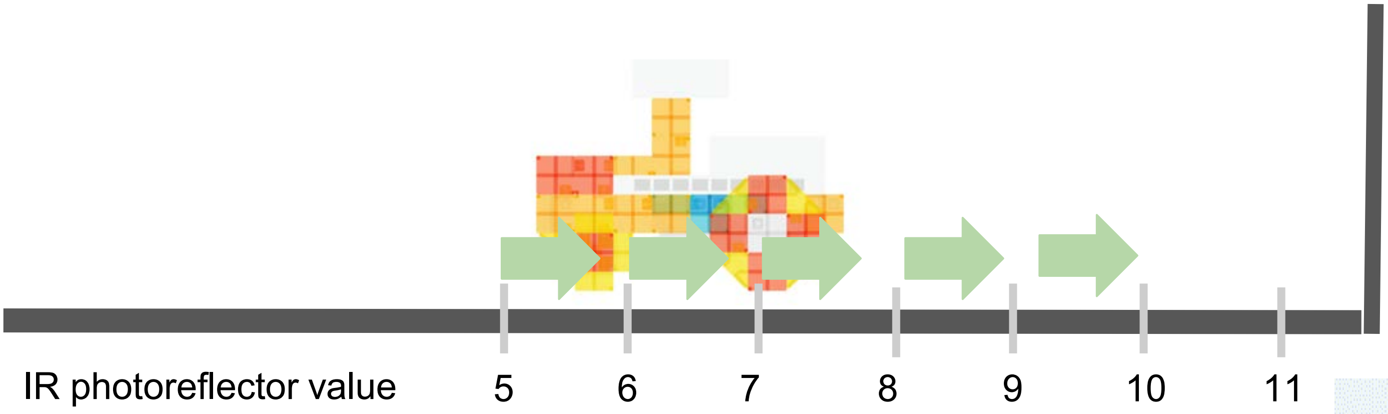


Mission 3

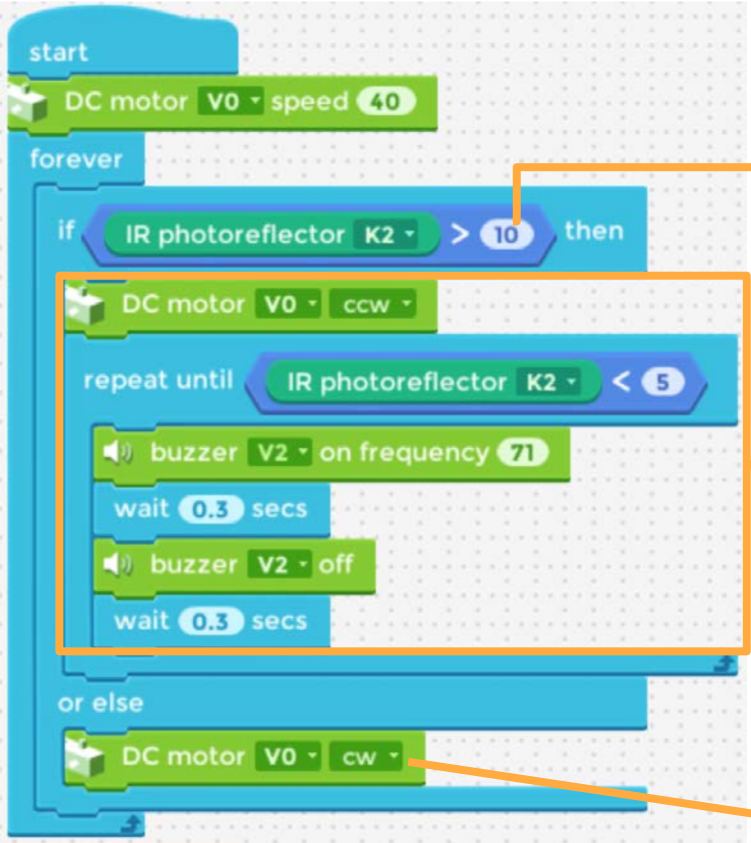
Link the motion of DC motor and sensors







Day 4 Lab Stage 5 Mission 3: Link the motion of DC motors and sensors



```
start
DC motor V0 speed 40
forever
  if IR photoreflector K2 > 10 then
    DC motor V0 ccw
    repeat until IR photoreflector K2 < 5
      buzzer V2 on frequency 71
      wait 0.3 secs
      buzzer V2 off
      wait 0.3 secs
  or else
    DC motor V0 cw
```

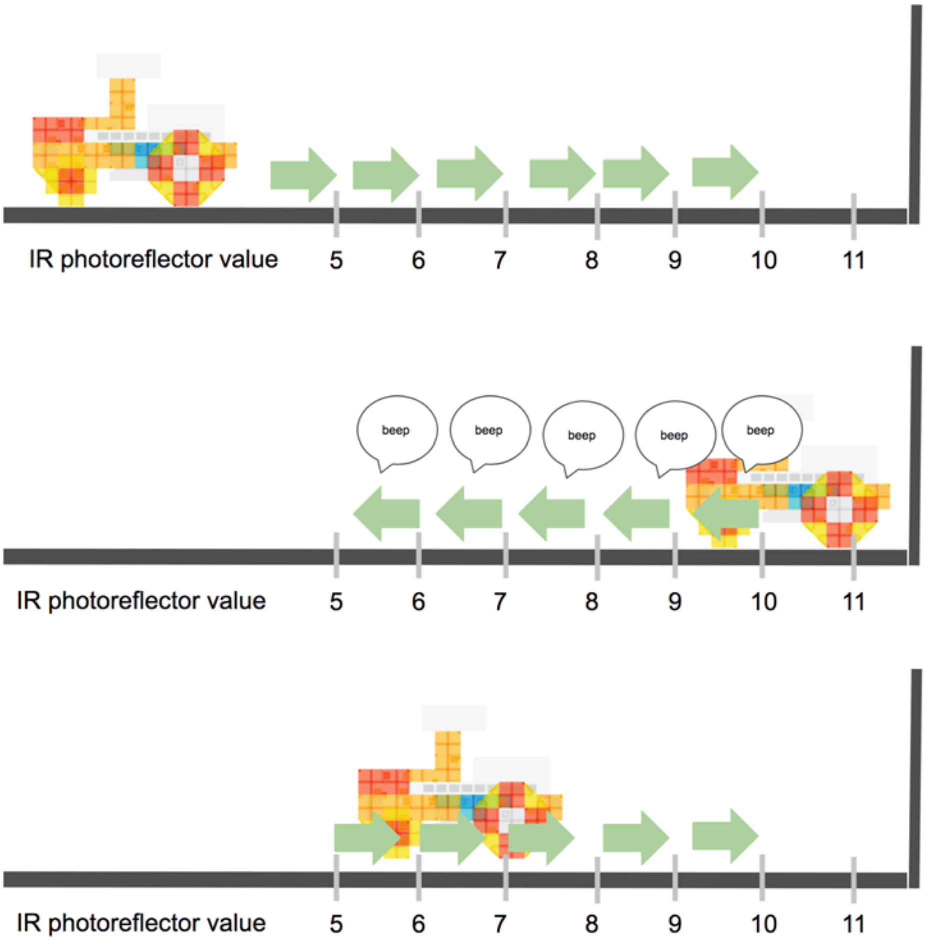
What would change if you make this value from 10 to 20?

____

____

The tractor moves backward until it is far enough from something in front of it.

The tractor moves forward.



forward (cw)

backward (ccw)

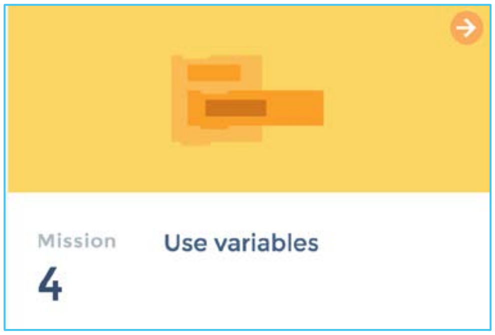
forward (cw)

Day 4 Lab Stage 5 Mission 4: Use variables

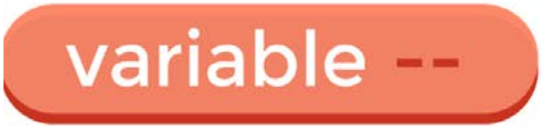
This mission's goal

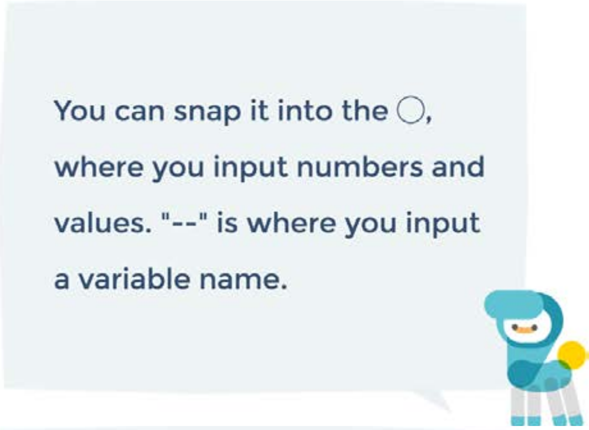
Use **variables** to slow down and stop the tractor.

- The speed is stored in a variable or a placeholder.
- The tractor will slow down by 10 in a loop, until it stops. Each loop takes 3 seconds.

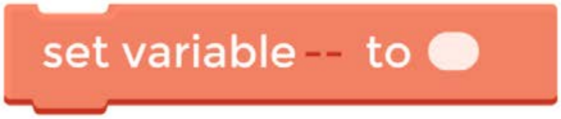


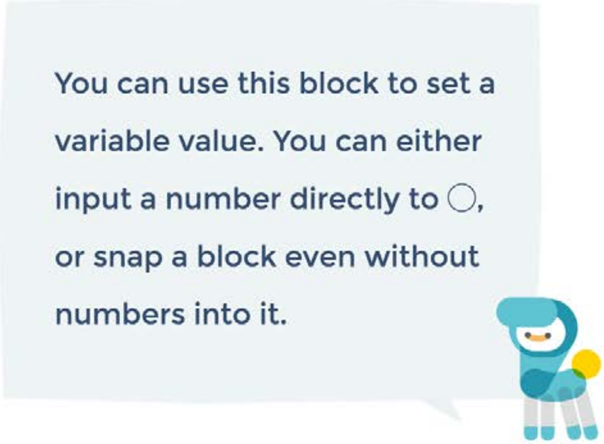
Mission 4 Use variables



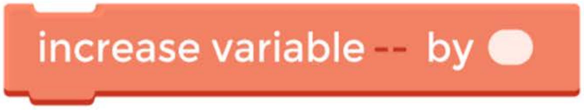


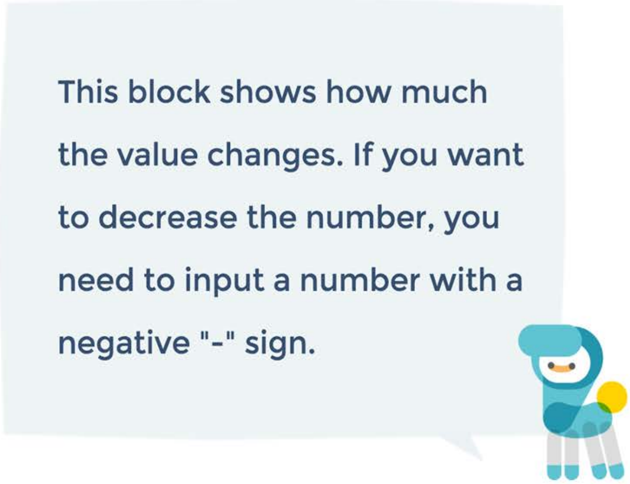
You can snap it into the ○, where you input numbers and values. "--" is where you input a variable name.





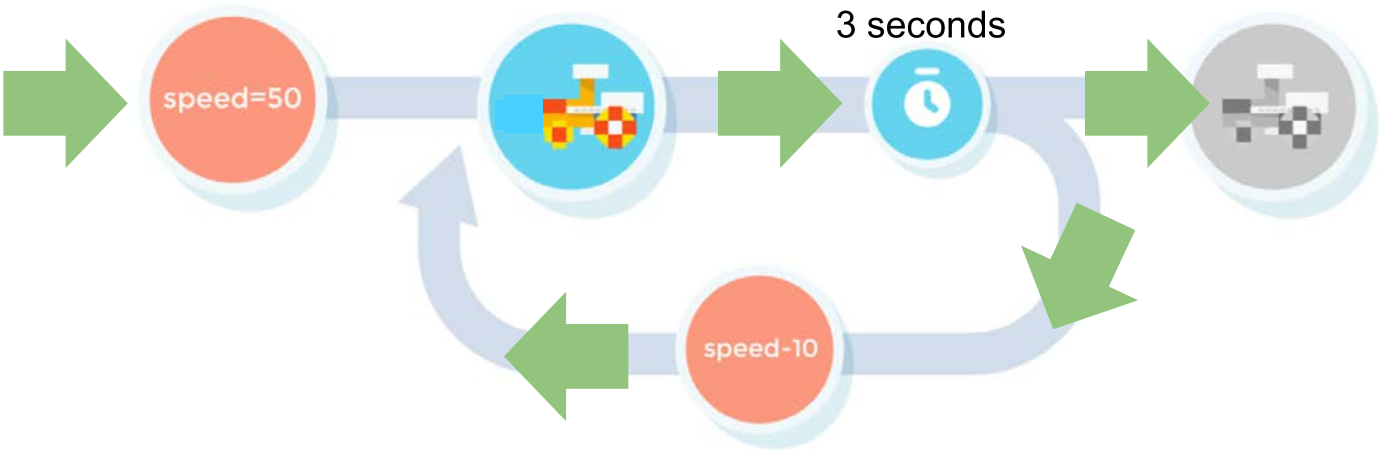
You can use this block to set a variable value. You can either input a number directly to ○, or snap a block even without numbers into it.

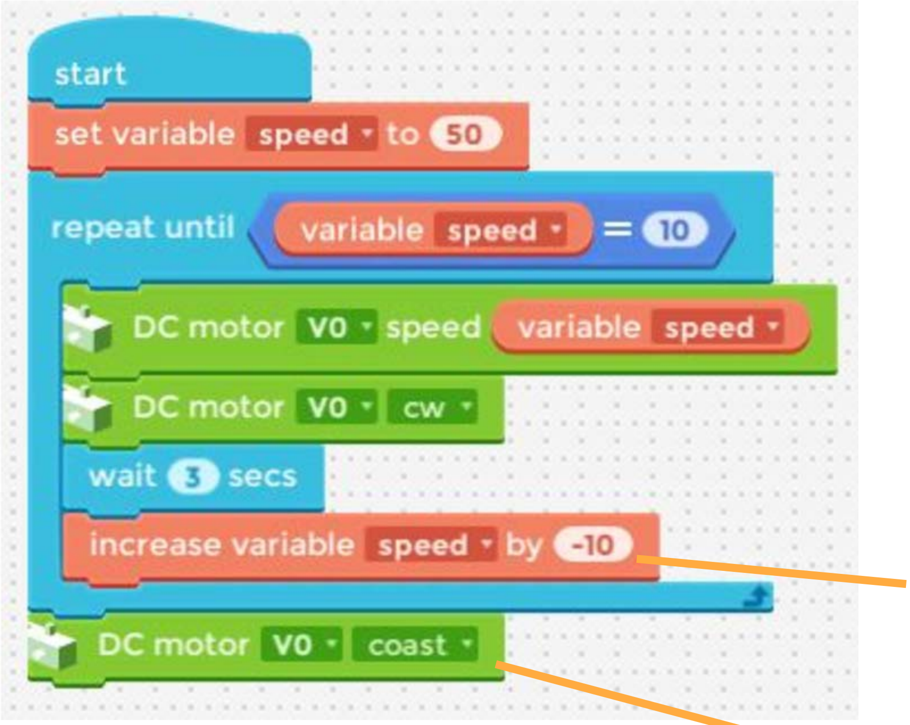




This block shows how much the value changes. If you want to decrease the number, you need to input a number with a negative "-" sign.

Day 4 Lab Stage 5 Mission 4: Use variables





```
start
set variable speed to 50
repeat until variable speed = 10
    DC motor V0 speed variable speed
    DC motor V0 cw
    wait 3 secs
    increase variable speed by -10
DC motor V0 coast
```

Time (seconds)	Value of the **variable “speed”**
0	50
3	40
6	?
9	?
12	10

Increase by a negative number (-10) is to decrease.

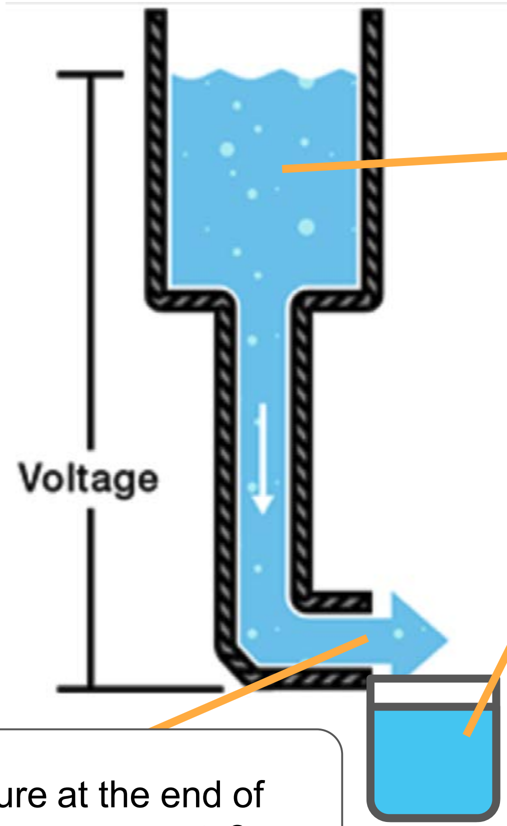
To coast is to stop slowly

Day 4 Physics: DC, Current, and Voltage

In **direct current (DC)**, the electric charge (**current**) only flows in ONE DIRECTION. A battery generates direct current (DC).

We can think of this tank as a battery, a place where we store a certain amount of energy and then release it. The pressure at the end of the hose represents **voltage**. The water represents an **electric charge**. The amount of water that flows through in a given time period represents the **current**.

The more water in the tank, the higher the charge, the more pressure is measured at the end of the hose, and, as a result, the higher the current (water flows faster).



Water represents

____

____

The amount of water that flow throughs a time period represents

____

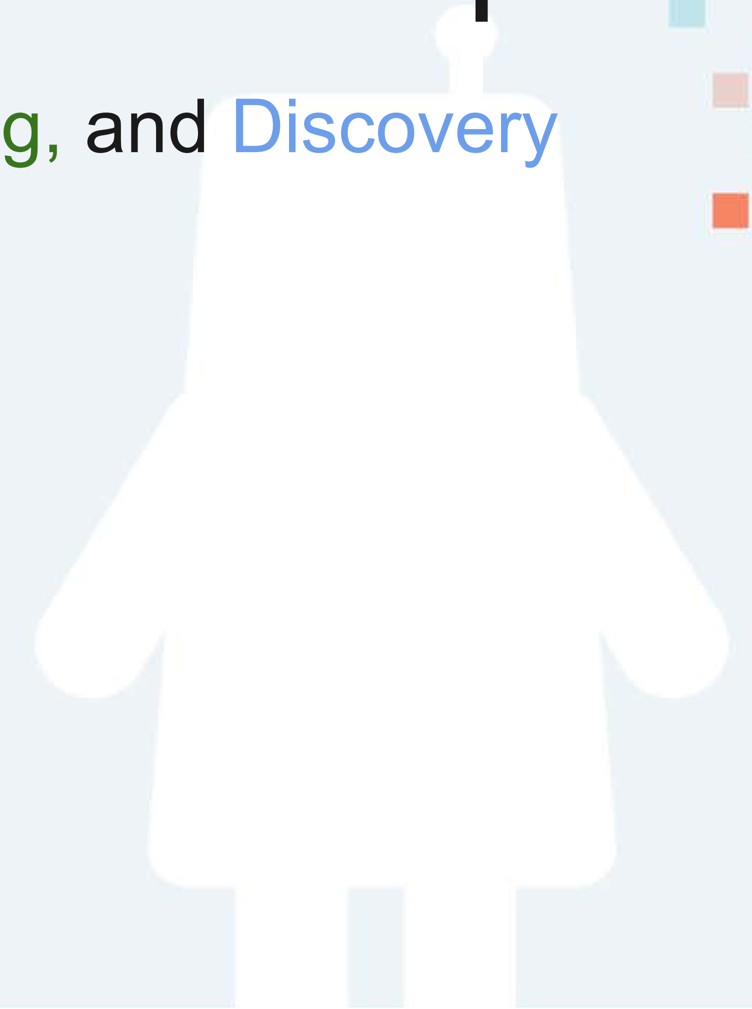
Pressure at the end of the hose represents?

____

KOOV

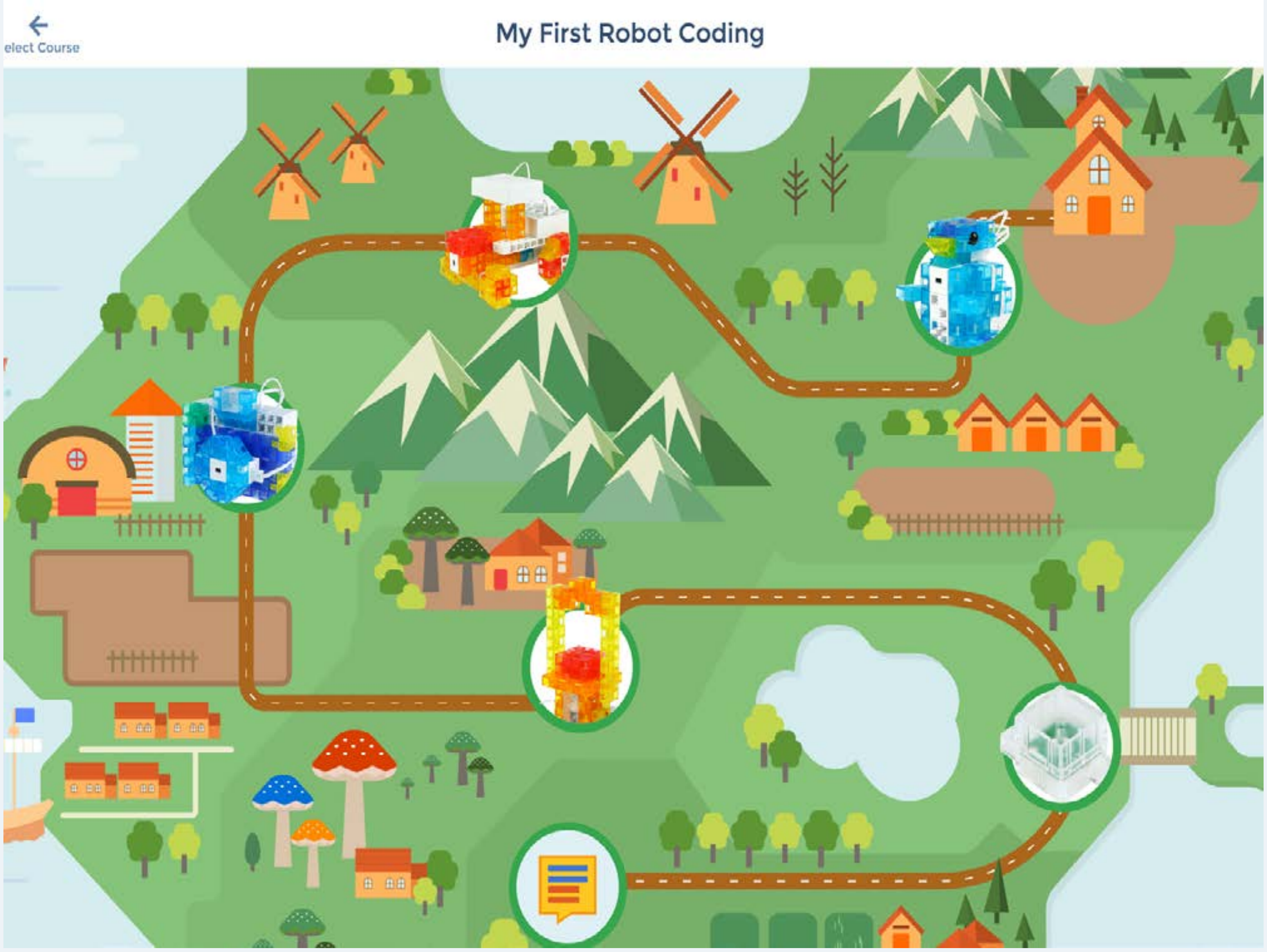
Robotics Camp

Fun, Learning, and Discovery



Fun Road Ahead

Welcome to the KOOV Robotics Camp. At this camp, you are going to put on the hat of a student robotic engineer and learn by building. You will build and program several "robots" each day,following instructions in this curriculum and from My First Robot Coding Map. Then you will be given a Design Challenge to brainstorm for better design. Each day will be filled with learning, fun, and discovery.



Week In A Glance

Day 1: KOOV Education Suite

Learn about Robotics and KOOV

Day 2: Blink and Beep!

Coding with LEDs and Buzzer

Day 3: Light and Camera

Detect Light and Distance

Day 4: Turn that Wheels

DC Motor and Robot Car

Day 5: Dancing Feet

Servo Motor and Penguin

DAY 5

Dancing Feet

Day 5 Objectives

E

Servo Motor

C

List
Wait Until
Repeat Until

M

Penguin

Electronics

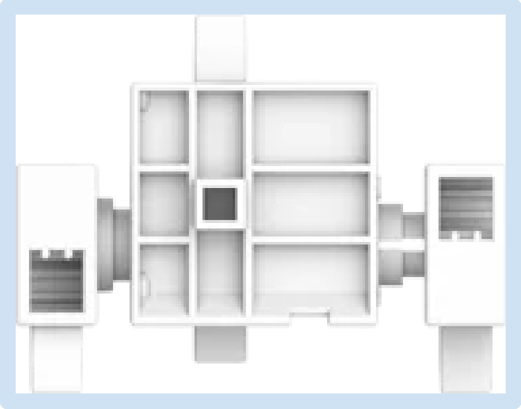
Servo Motor

E

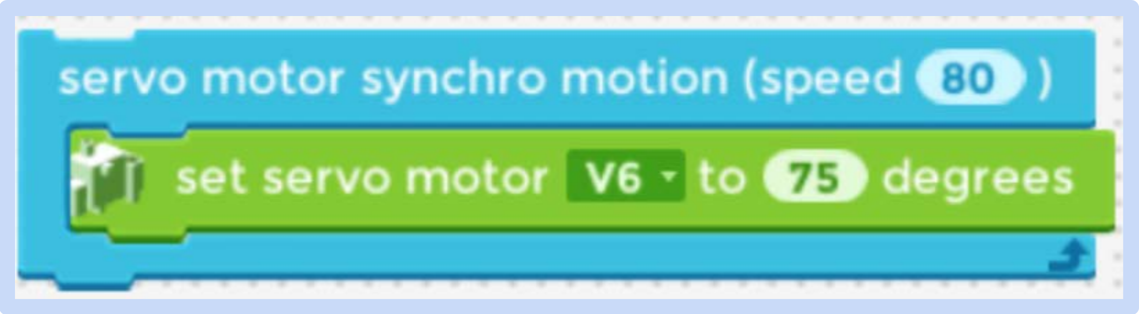
KOOV Servo Motor

Video: Alligator

There are two type of KOOV motors, one is a rotating DC Motor, the other is the Servo Motor. A KOOV DC Motor can turn clockwise and counterclockwise at a given speed, and the Servo Motor provides finer control and let you turn to a specific angle at a specific speed. Servo Motor usually has a servo arm or shaft that can turn 0-180 degrees.



The KOOV Servo Motor can be controlled via KOOV Code Blocks. It can be programmed to rotate at a speed to a specific angular(angular stands for angle) location. The codes below shows how to turn the shaft at speed 80 to 75 degrees.



```
servo motor synchro motion (speed 80 )
    set servo motor V6 to 75 degrees
```

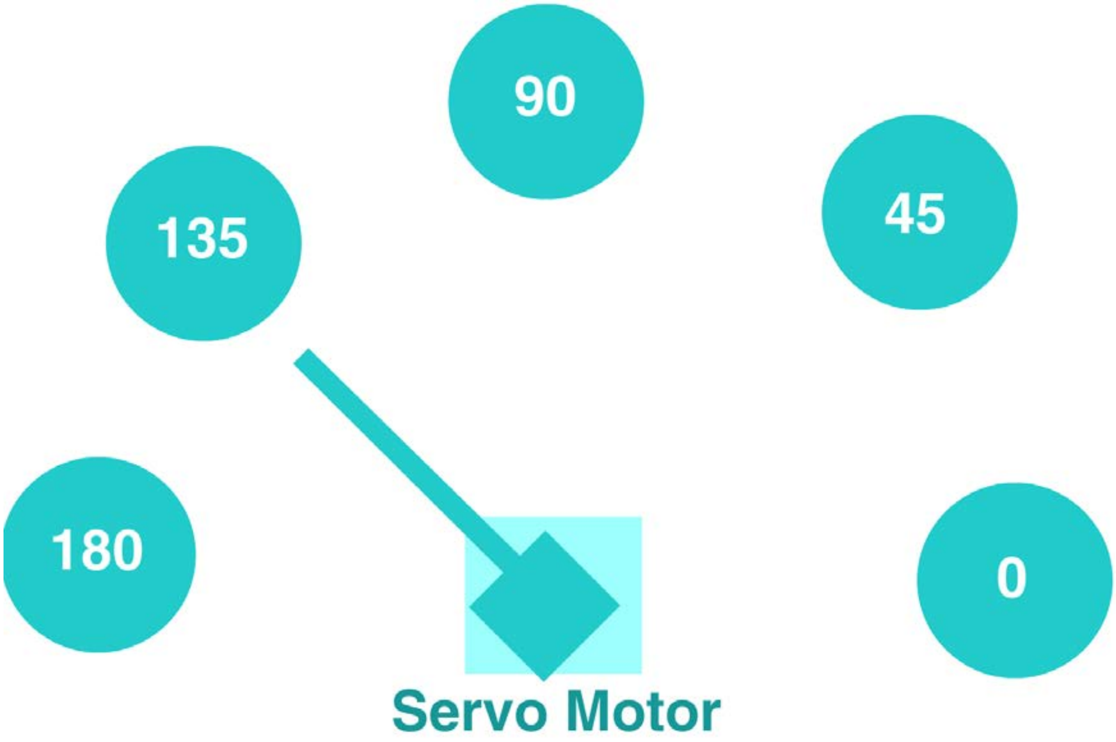
KOOV Servo Motor

simulation

Video: Rotate at different Speed

Video: Change Angles

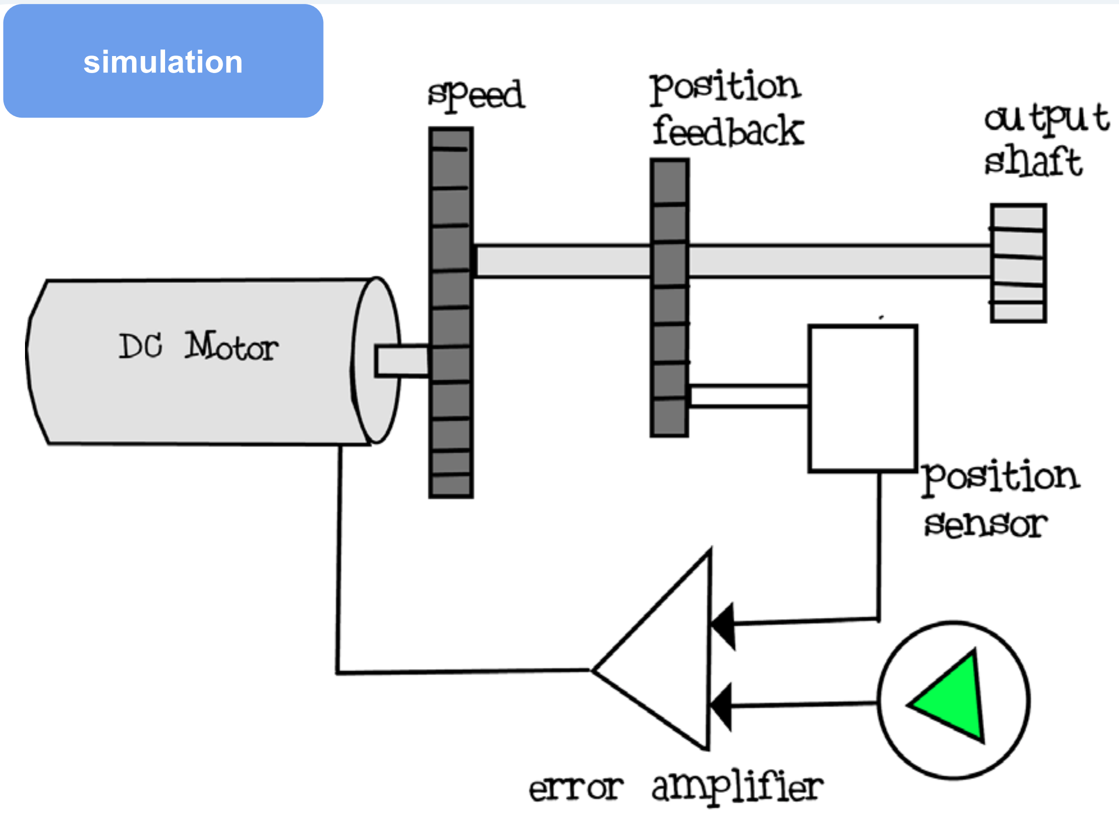
A KOOV Servo Motor can rotate between 0 to 180 degrees. The illustration below shows the location of the shaft when the code rotates a servo motor to different positions.



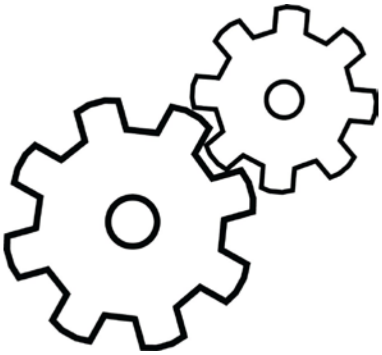
E

Look Inside a Servo Motor

simulation



A Servo Motor contains a DC Motor, gears, a position sensor, and an error amplifier. DC Motor rotates the shaft; gears help control rotation; position sensor provides current location; error amplifier tracks the difference between current and desired location. Together it forms a control loop that keeps going until the desired location is reached.



c
o
d
i
n
g

Wait Until ...

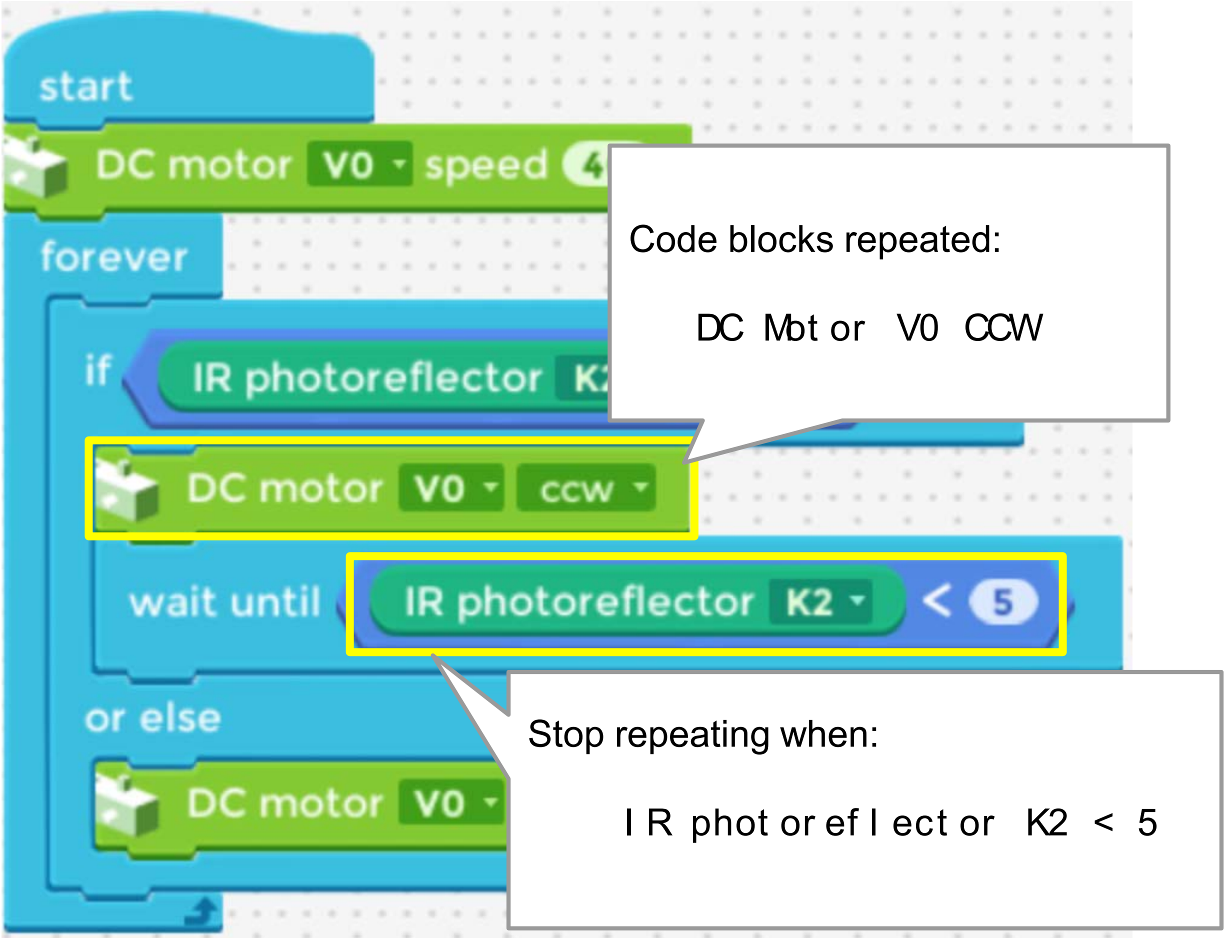
Repeat Until ...

KOOV

Wait until

The wait until block is to hold the current state until a condition is true. For example, a chef could be boiling the water to cook pasta, and she will turn on the fire, and wait until the pasta is done.

The code below is the codes for a tractor. It shows that the DC Motor would keep rotating counterclockwise (DC Motor V0 ccw) until the car's proximity detector (IR photoreflector) is not close to anything.



```
start
DC motor V0 speed 4
forever
  if IR photoreflector K
    DC motor V0 ccw
    wait until IR photoreflector K2 < 5
  or else
    DC motor V0
```

Code blocks repeated:

DC Motor V0 CCW

Stop repeating when:

IR photoreflector K2 < 5

Repeat until

The repeat until ... code block is used to repeat a set of steps that could be inside a function, until a condition is true.

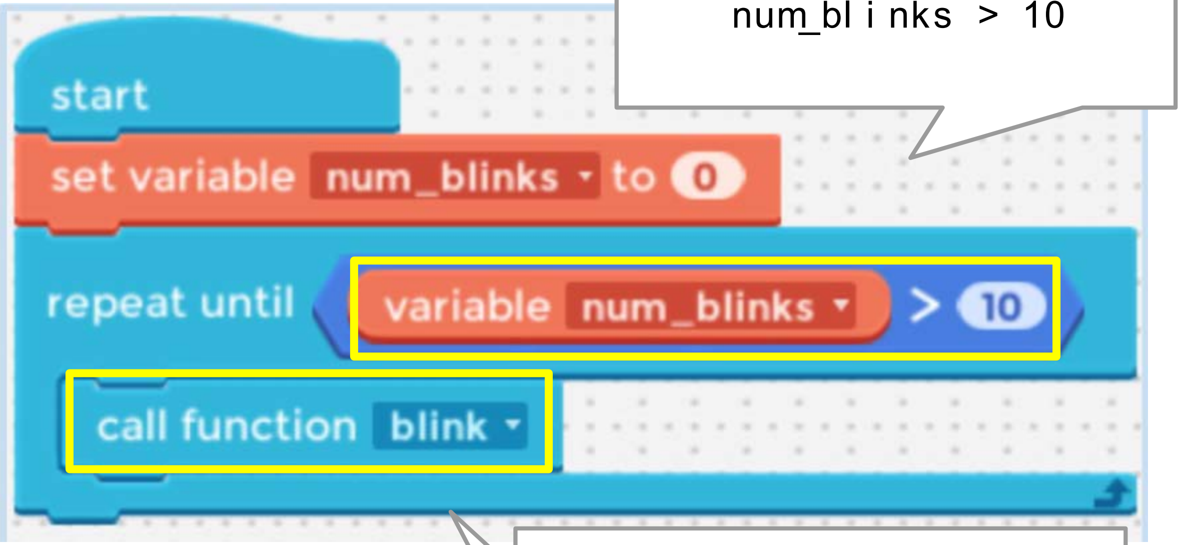
An example in our daily life could be:

- Repeat washing until it's clean
- Repeat playing until I am tired

The codes below make an LED keep blinking (via calling function "blink") 10 times.

Condition to check:

num_blinks > 10



```
start
set variable num_blinks to 0
repeat until variable num_blinks > 10
    call function blink
```

Code blocks repeated:

call function blink

M

a

p

Penguin

M

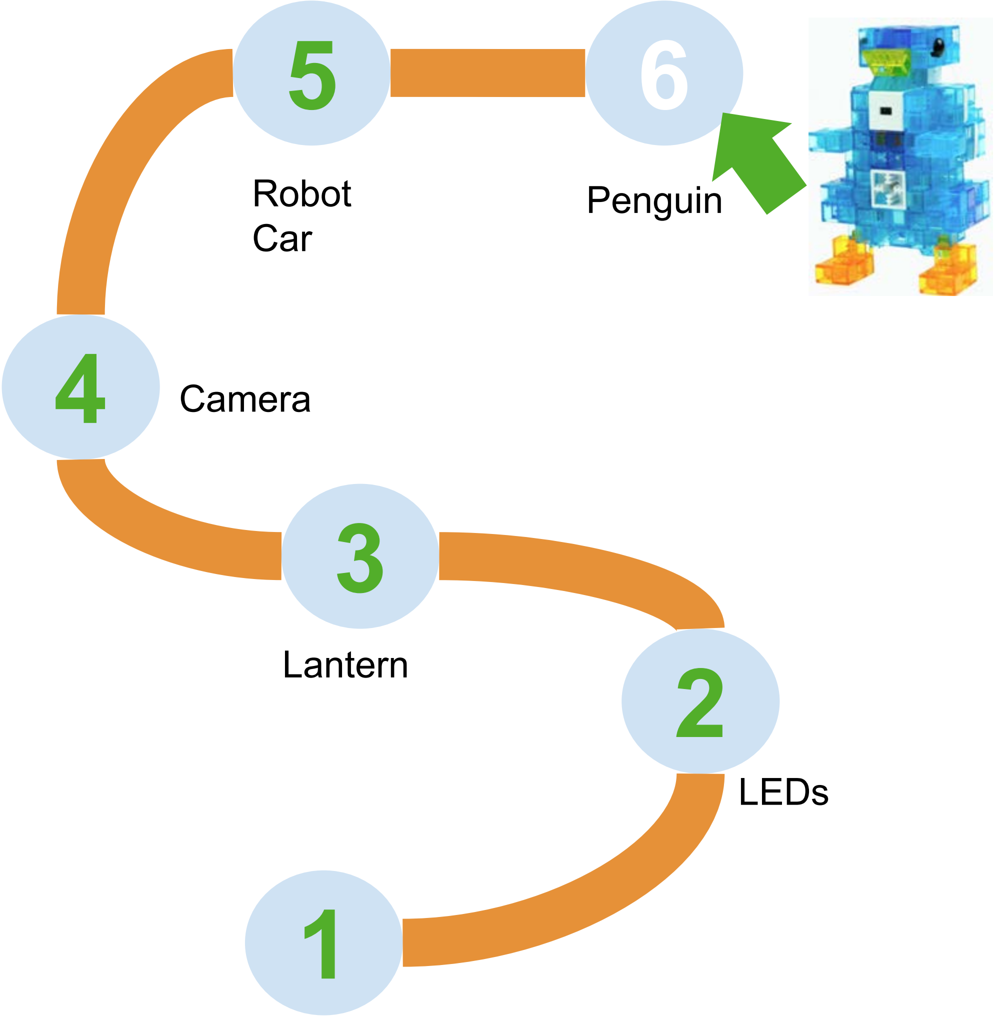
Robotic Coding Map: Penguin

Map: My First Robotic Coding
Stage: Stage Five

Mission 1: Let’s build a penguin
Mission 2: Use the servo motor
Mission 3: Use the Multi LED
Mission 4: Use variables

To go to Map:

1. Login
2. Select “Learning Course”
3. Select “My First Robot Coding”



1 What is Coding

2 LEDs

3 Lantern

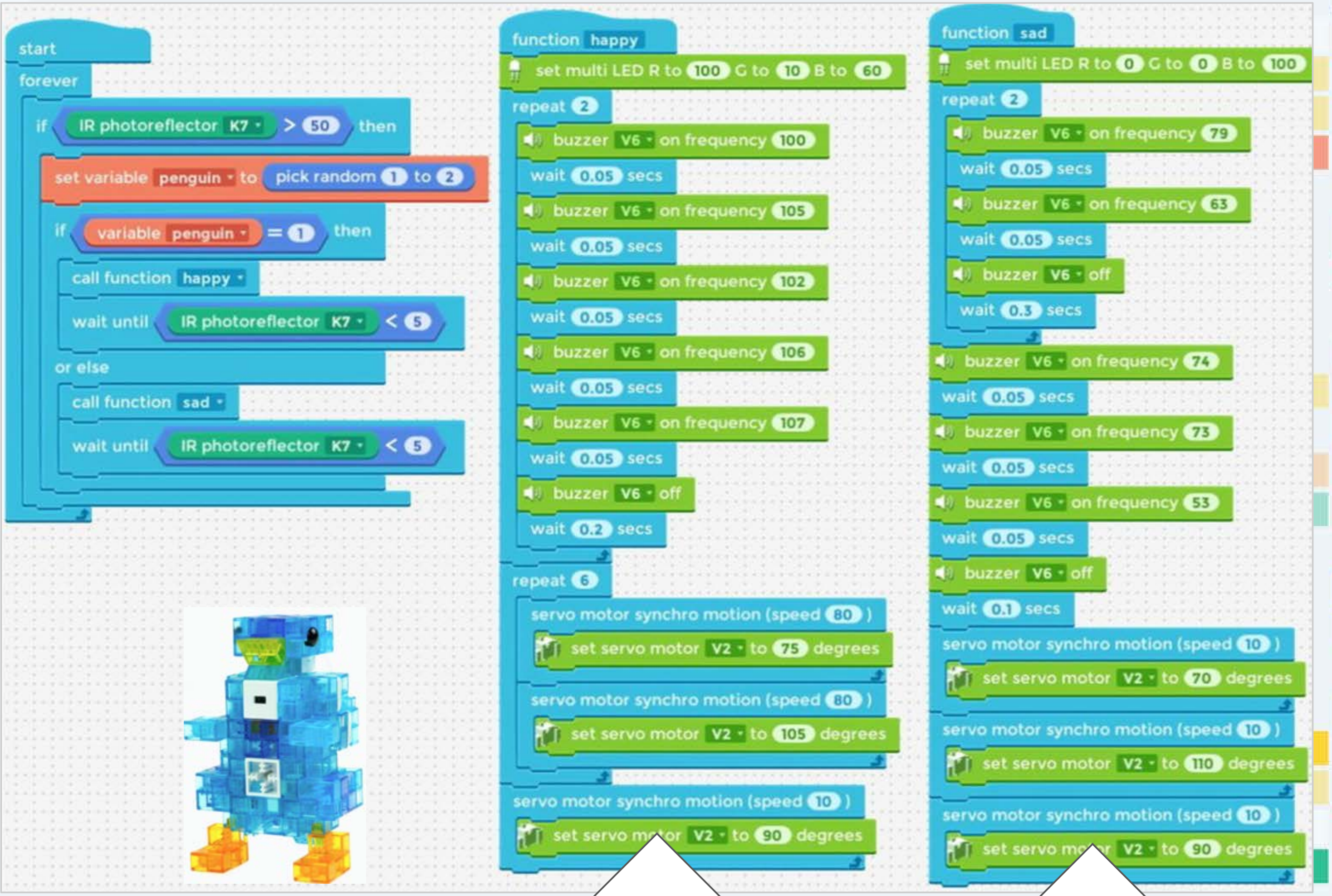
4 Camera

5 Robot Car

6 Penguin

M

Robotic Coding Map: Penguin



```
start
forever
    if IR photoreflector K7 > 50 then
        set variable penguin to pick random 1 to 2
        if variable penguin = 1 then
            call function happy
            wait until IR photoreflector K7 < 5
        or else
            call function sad
            wait until IR photoreflector K7 < 5
```

```
function happy
set multi LED R to 100 G to 10 B to 60
repeat 2
    buzzer V6 on frequency 100
    wait 0.05 secs
    buzzer V6 on frequency 105
    wait 0.05 secs
    buzzer V6 on frequency 102
    wait 0.05 secs
    buzzer V6 on frequency 106
    wait 0.05 secs
    buzzer V6 on frequency 107
    wait 0.05 secs
    buzzer V6 off
    wait 0.2 secs
repeat 6
    servo motor synchro motion (speed 80 )
        set servo motor V2 to 75 degrees
    servo motor synchro motion (speed 80 )
        set servo motor V2 to 105 degrees
servo motor synchro motion (speed 10 )
    set servo motor V2 to 90 degrees
```

```
function sad
set multi LED R to 0 G to 0 B to 100
repeat 2
    buzzer V6 on frequency 79
    wait 0.05 secs
    buzzer V6 on frequency 63
    wait 0.05 secs
    buzzer V6 off
    wait 0.3 secs
buzzer V6 on frequency 74
wait 0.05 secs
buzzer V6 on frequency 73
wait 0.05 secs
buzzer V6 on frequency 53
wait 0.05 secs
buzzer V6 off
wait 0.1 secs
servo motor synchro motion (speed 10 )
    set servo motor V2 to 70 degrees
servo motor synchro motion (speed 10 )
    set servo motor V2 to 110 degrees
servo motor synchro motion (speed 10 )
    set servo motor V2 to 90 degrees
```

Function: happy

Function: sad

Competition and Games

Competition

□ Giant Robots

□ Build the Cubes

□ S.O.S

□ Speed Building

□ Train Race

Design a Giant Robot

Design Challenge - Giant Robot

Imagine if you had an infinite number of KOOV blocks (of any sizes and color) and KOOV electric parts, can you build these?

Design a large eye that can turn around and look

- What electric parts might you need?
- What problems may you run into?
- What can you use this robot for?

Design a robot foot that can wiggle all its toes

- What electric parts might you need?
- What problems may you run into?
- What can you use this robot for?

Design a robot hand that can hold a cup

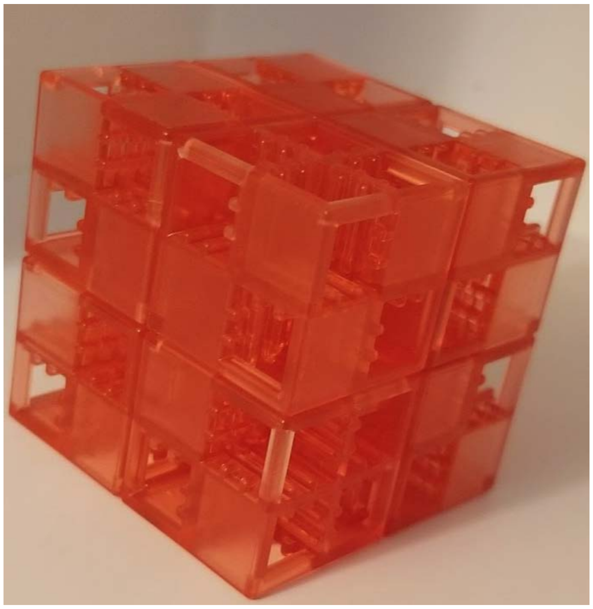
- What electric parts might you need?
- What problems may you run into?
- What can you use this robot for?

Build the Cubes

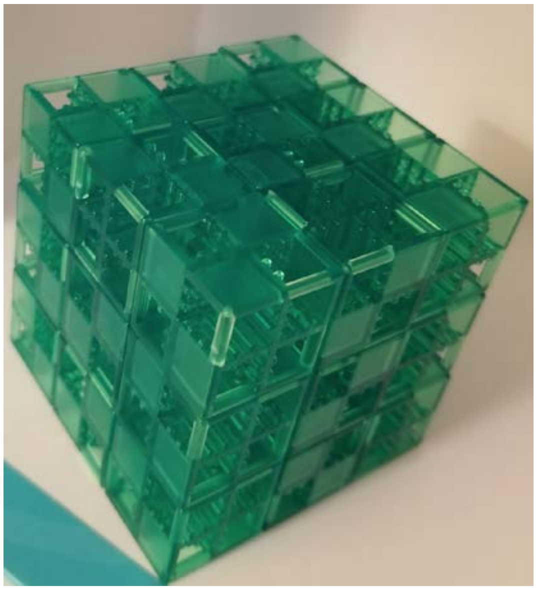
Build cubes:
 2 blocks by 2 blocks and
 3 blocks by 3 blocks

Challenge:
 How about 4 blocks by 4 blocks?

2 x 2



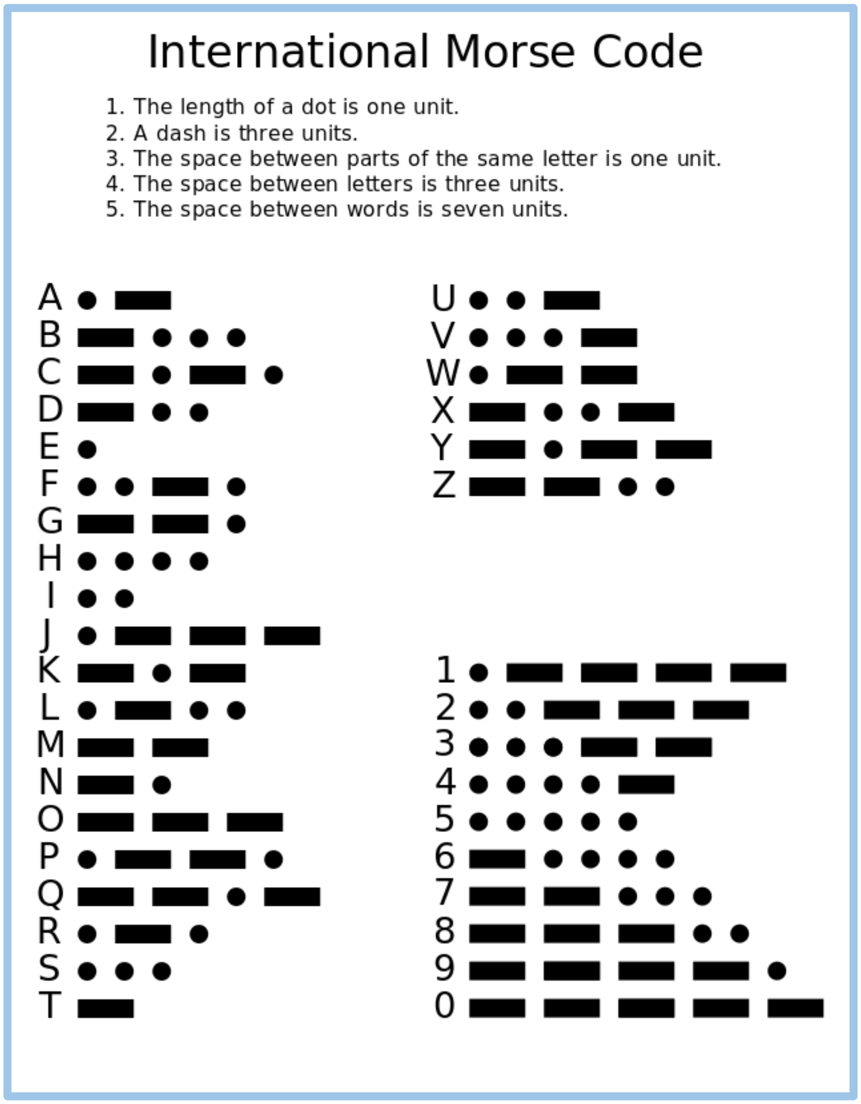
3 x 3



S.O.S

Morse code is a method of transmitting text information as a series of on-off tones, lights, or clicks. You can send S.O.S Morse code distress signals by using the KOOV LED and Buzzer. Build a program to blink and beep S.O.S in Morse Code: short, short, short, long, long, long, short, short, short:

S ● ● ●   O ▬ ▬ ▬   S ● ● ●



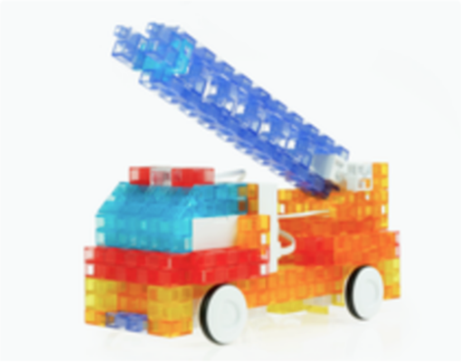
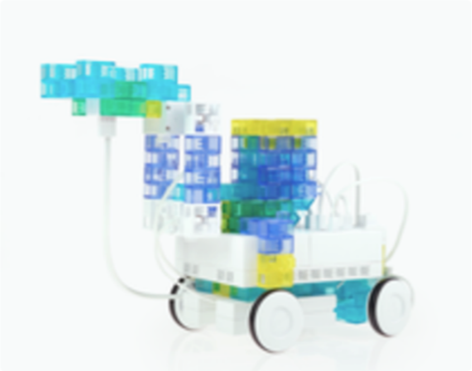
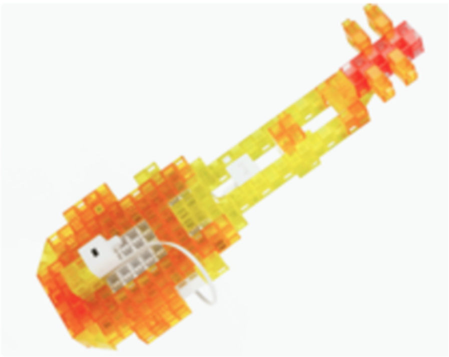
International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

A	● ▬
B	▬ ● ● ●
C	▬ ● ▬ ●
D	▬ ● ●
E	●
F	● ● ▬ ●
G	▬ ▬ ●
H	● ● ● ●
I	● ●
J	● ▬ ▬ ▬
K	▬ ● ▬
L	● ▬ ● ●
M	▬ ▬
N	▬ ●
O	▬ ▬ ▬
P	● ▬ ▬ ●
Q	▬ ▬ ● ▬
R	● ▬ ●
S	● ● ●
T	▬
U	● ● ▬
V	● ● ● ▬
W	● ▬ ▬
X	▬ ● ● ▬
Y	▬ ● ▬ ▬
Z	▬ ▬ ● ●
1	● ▬ ▬ ▬ ▬
2	● ● ▬ ▬ ▬
3	● ● ● ▬ ▬
4	● ● ● ● ▬
5	● ● ● ● ●
6	▬ ● ● ● ●
7	▬ ▬ ● ● ●
8	▬ ▬ ▬ ● ●
9	▬ ▬ ▬ ▬ ●
0	▬ ▬ ▬ ▬ ▬

Speed Building

See how many robots a team can build in an hour. Each team selects the robots they want to build. Once they finish a robot, they raise their hand, and the mentor will check it and give points for accuracy, up to the max points listed below.

Recipe	Picture
Firetruck (Difficult) Where: From KOOV Robot Recipe Point: 20	
ForkLift (Medium) Where: From KOOV Robot Recipe Point: 10	
Guitar (Basic) Where: From KOOV Robot Recipe Point: 5	

Train Race

Build:

Locomotives (KOOV robot recipe)

Follow a dark line (use IR Photo reflector to detect the distance between the train and the dark line on the paper

Maze

Build a train that can get out of the maze the quickest

G
a
m
e
s

Games

□ Human Robotic Relay

□ Robotics Charade

Games - Robotics Relay

Material:

A list of questions with answers,

Strips of these questions (with answers)

Small ziplock bags

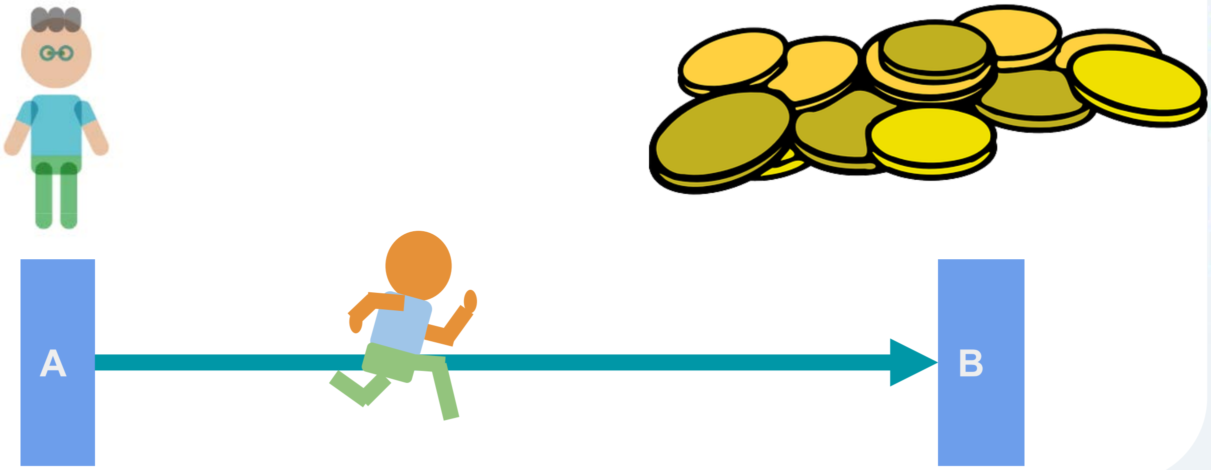
Plastic gold coin or paper printout

How to Play:

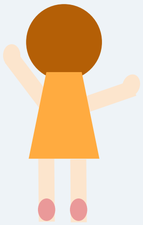
- Have kids in groups and line up at line A. From each group, send one runner at a time to run to B.
- When a runner arrives at line B, the mentor asks a random selected question. If this runner gets it right, they get a gold coin to put in their bag.
- Each runner runs back and passes the bag to the next runner.
- The game ends when the time is up.

How to Win:

The team with the most gold coins wins.



Games - Robotics Charade



Material:

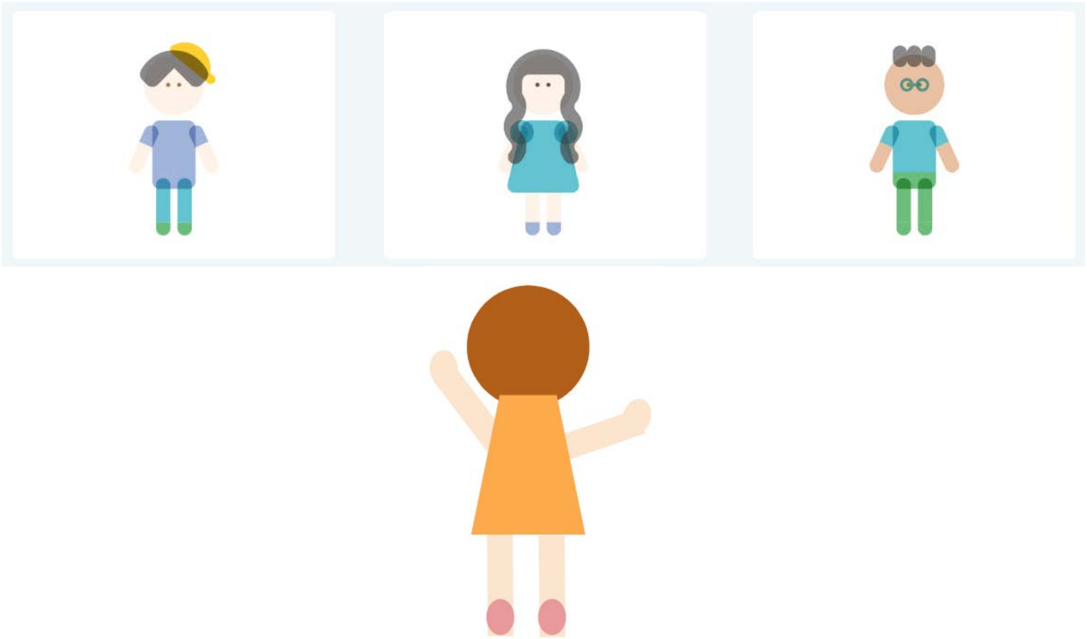
Timer, Word Cards

How to Play

- Game controller (the teacher or mentor) holds the word cards.
- Each team takes turns sending a player to act out the word, but all of the teams can guess. Teams take turns guessing, and each team has three guesses. The team who guesses right scores one point.
- Each word should take only 1 min, and if no teams guess the word when the minute is up, no one scores.
- The game is over when all word cards have been tried.

How to Win:

The team who has the most points wins.



Robotics Charade

Light Sensor

IR Photoreflector

DC Motor

Servo Motor

Buzzer

Red LED

Green LED

wait

Robot

Light

Sound

frequency

Push Switch

Blue LED

Top Core Button

Bottom Core Button

Right Core Button

Left Core Button

KOOV Core

Battery Pack

KOOV

appendix

Advanced

P

Physics

- Light
- Current & Voltage
- Closed Feedback Loop System

E

Electronics

Accelerometer

C

Coding Concepts

List

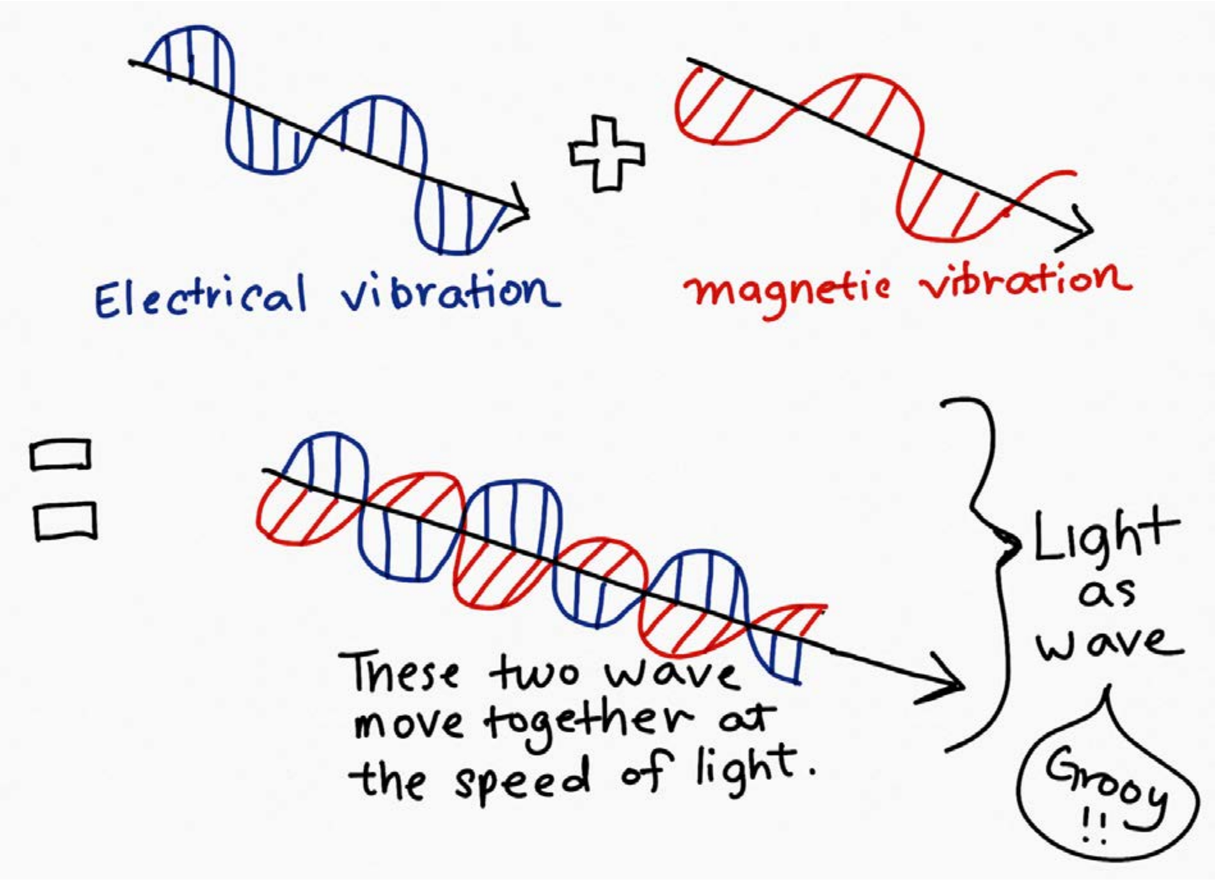
Physics

advanced

Light

Light as a wave

Light is an **Electromagnetic Wave** or EM WAVE that is visible to human eyes. Below is an illustration of an electromagnetic wave, which includes magnetic vibration and electrical vibration, moving together at speed of light.



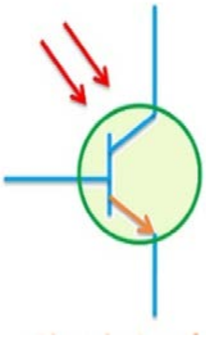
Light is just one type of EM wave, and there are other EM waves that are invisible to us. A chart called Electromagnetic Spectrum contains all EM waves, from the smallest frequency to the highest. Frequency is how many waves are there in a second.

P

How Does Light Sensor Work?

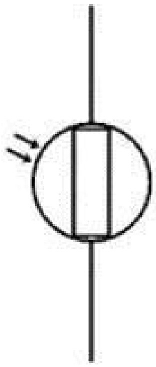
A **Light Sensor** is a device that detect brightness by converting light energy into electrical signals. Light sensors are also known as **Photo Sensors** because the convert light energy (**photons**) into electricity (**electrons**). Light sensors can be grouped into two main categories:

- those which generate electricity when illuminated
- those which change their electrical properties when illuminated



A **phototransistor** is a light sensor that generates electricity when illuminated. The “photo” in phototransistor is light, and the “transistor” means an amplifier, which amplifies or increases something.

The brighter the light, the more electricity is generated.



Photoresistor is a light sensor that changes its electrical properties when illuminated. A photoresistor does not generate electricity when there is light, but it reduces the resistance of electricity flow. Resistance is how hard it is for electricity to flow through an object. When resistance is reduced, electricity flows faster.

The brighter the light, the more electricity is allowed to flow through.

P
h
y
s
i
c
s

advanced

Current and Voltage

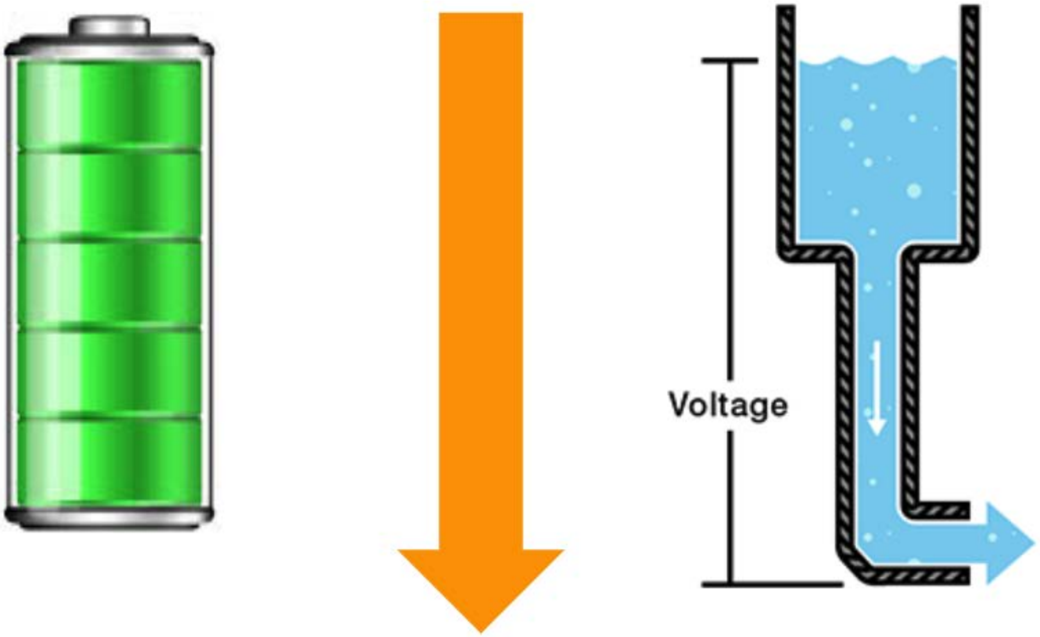
KOOV

DC, Current, and Voltage

In **direct current (DC)**, the electric charge (**current**) only flows in one direction. A battery generates direct current.

The pressure at the end of the hose can represent **voltage**. The water in the tank represents charge. The more water in the tank, the higher the charge, the more pressure is measured at the end of the hose.

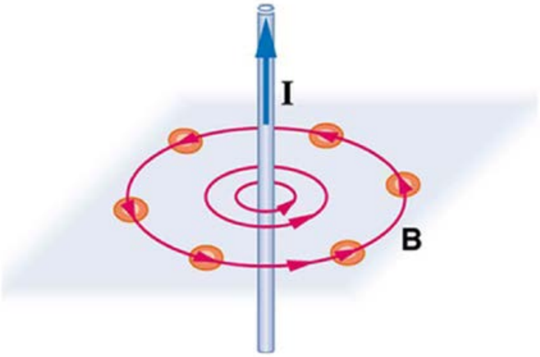
We can think of this tank as a battery, a place where we store a certain amount of energy and then release it. If we drain our tank a certain amount, the pressure created at the end of the hose goes down. We can think of this as decreasing voltage, like when a flashlight gets dimmer as the batteries run down. There is also a decrease in the amount of water that will flow through the hose. Less pressure (voltage) means less water is flowing, which brings us to current.



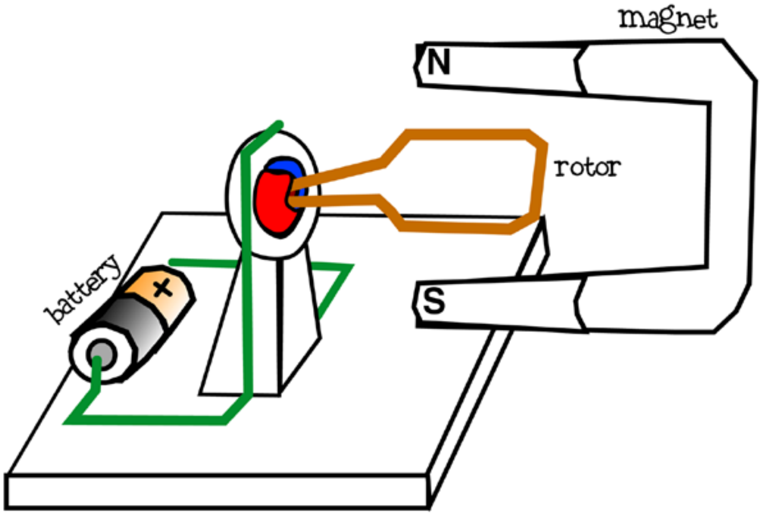
DC Motor rotates via

Magnetic Force

Electric current produces a **magnetic field.** This magnetic field can be visualized as a pattern of circular field lines surrounding a wire.



DC Motor works by running current through the rotor, generating magnetic field, which interacts with another magnet to keep the rotor turning. The higher the electric current, the stronger the magnetic field, and the stronger the interaction with that magnet inside the DC Motor, causing the rotor to spin faster.



Physics

advanced

Closed Feedback Loop System

Servo

The Servo in Servo Motor means that it is a close control loop using feedback.

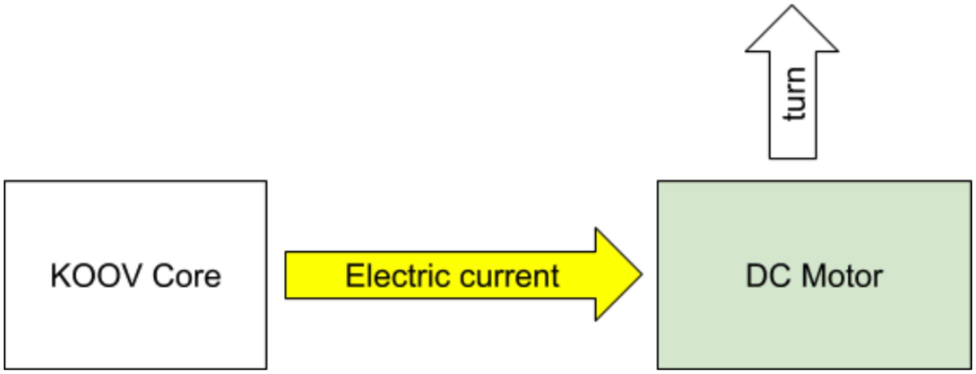
Servo is a general term for a closed loop control system using negative feedback.

The cruise control in a car is one example of a servo system. It measures your speed and feeds that back into a control circuit which adjusts the car acceleration to maintain a stable speed.

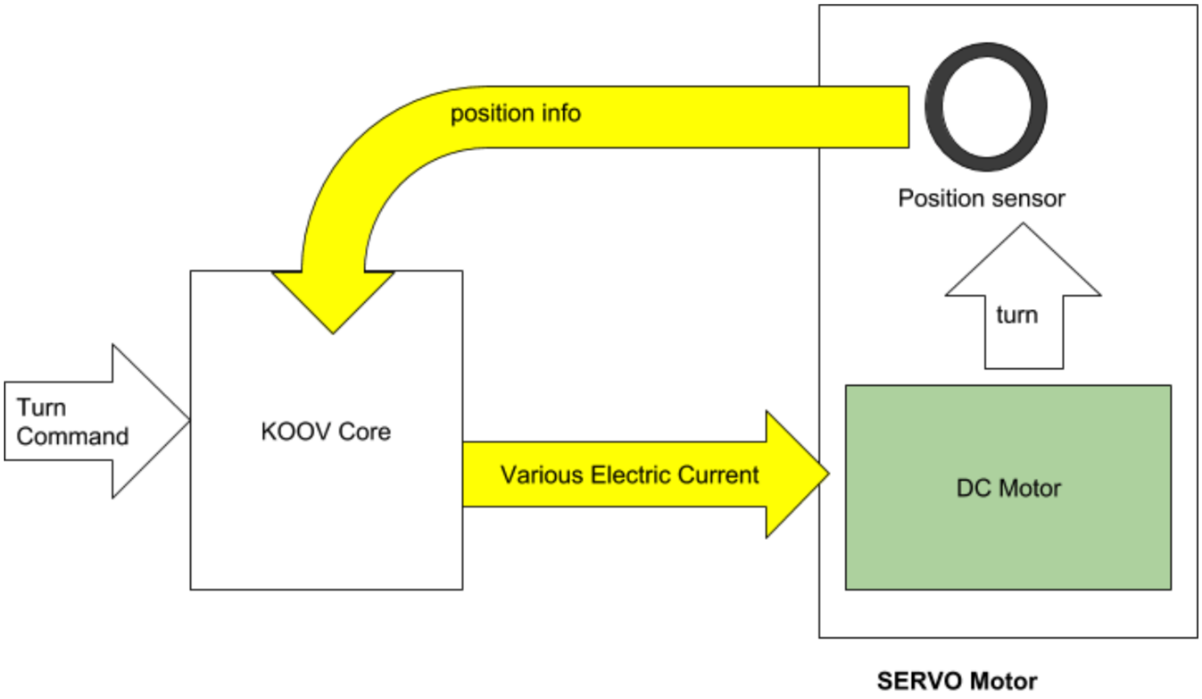
P

Closed and Open Loop System

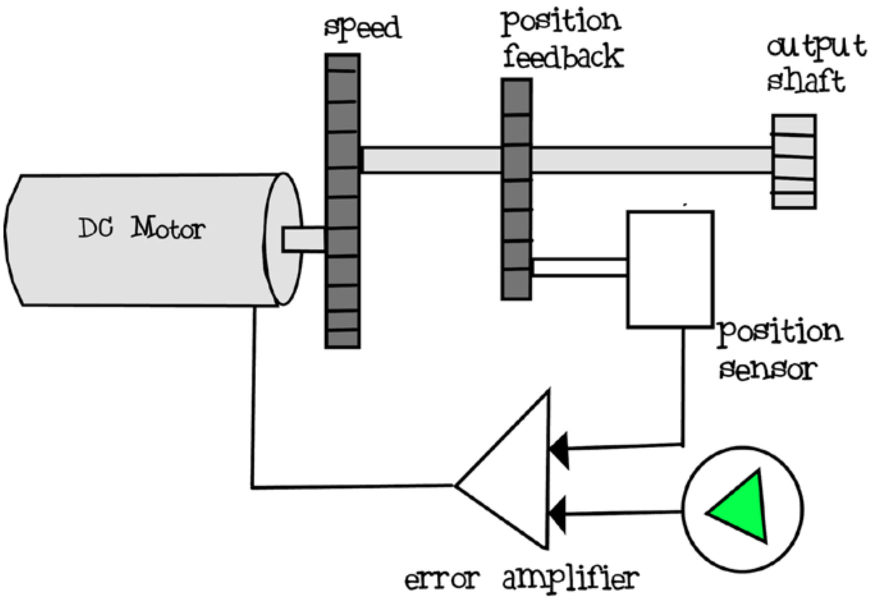
An "**Open** Loop" system has no feedback, so there is no way to verify that it is performing as expected. The **DC** Motor is in a Open Loop system, meaning that the current position is not fed back to the loop.



A "**Closed** Loop" system uses a feedback signal to adjust the speed and direction of the motor to achieve the desired result. The **Servo** Motor is a closed loop system that uses the current position to adjust the rotation speed



Look Inside a Servo Motor



A Servo Motor contains a DC Motor, gears, a position sensor, and an error amplifier. Together it forms a control loop that keeps going until the desired location is reached.

The DC Motor rotates the shaft. The gears decreases the rotation speed of the shaft for accurate positioning. The output shaft also drives another gear which rotates the position sensor. When rotated, the position sensor provides a voltage indicating the shaft output position. That voltage is fed back into an error amplifier.The error amplifier produces an error signal which is the difference between the desired position and the current position.
The error signal is then used to drive the motor. When the current position equals the desired position, then the error signal would be zero, and the DC Motor would stop.

Electronics

advanced

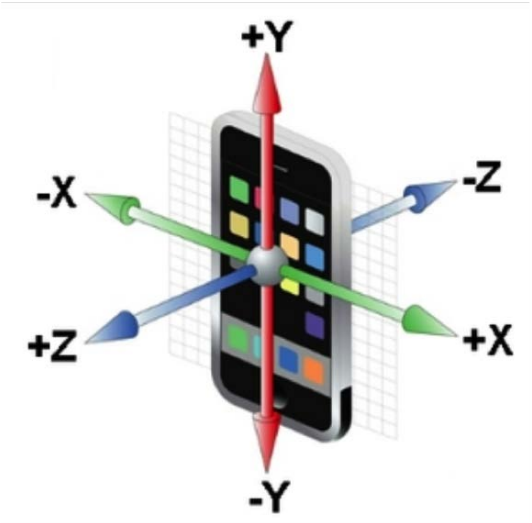
Accelerometer

How Does Tilting Works?

Have you ever play a game on a phone or a tablet that controls the game object by tilting? For example, you may have played a car racing game and “drives” the car by tilting your phone. How does your phone know how much you tilt or which directing you tilt? It knows by using a electronic device called an **accelerometer**.

Accelerometer

The Accelerometers are commonly used in game controllers and smartphones to use natural intuitive motion, such as motion similar to driving a car with steering wheel.



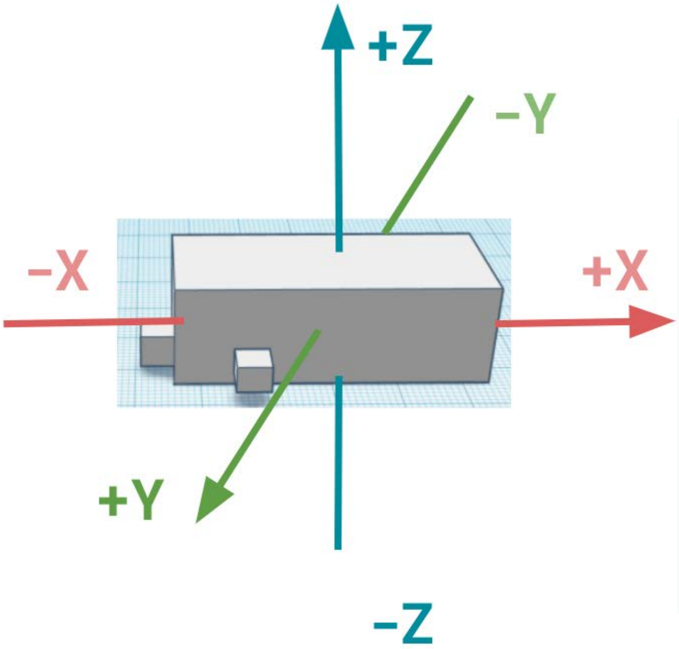
The Accelerometer tells the two-dimensional or three-dimensional positions by measuring the acceleration force along the direction of the axis. The acceleration force could be caused by gravity or movement.

E

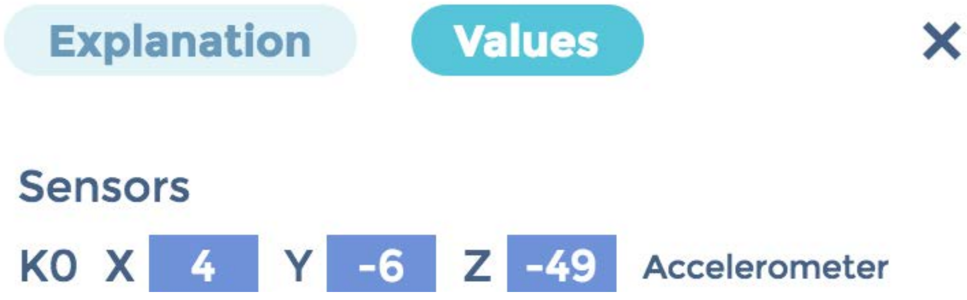
KOOV Accelerometer

video

The KOOV Accelerometer three-dimensional positions by measuring the acceleration force along the direction of the axis. You can learn how to use KOOV Accelerometer in the Firetruck project (found in the KOOV Robot Recipes) to control the ladder.



You can tilt a KOOV Accelerometer and watch its values change in KOOV.

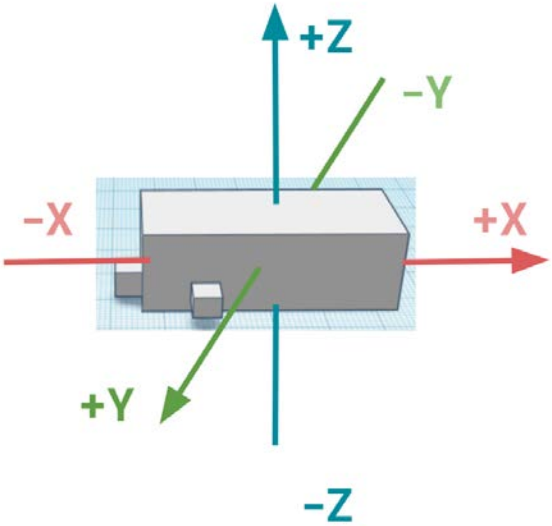


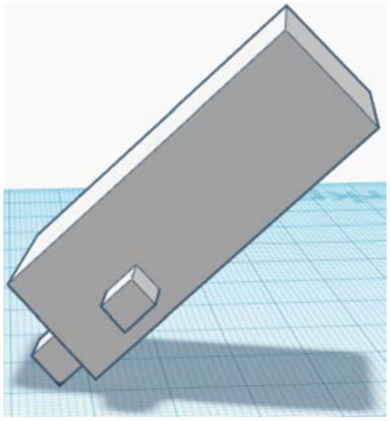
Sensors

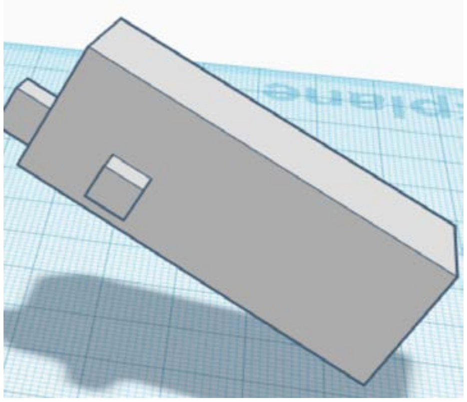
K0 X 4 Y -6 Z -49 Accelerometer

Rotate around X-axis

start





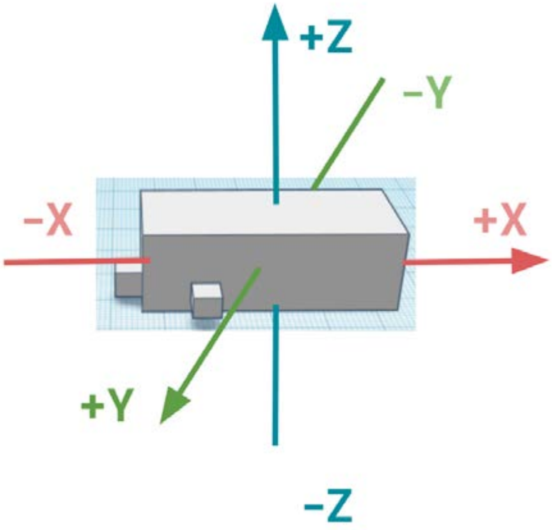


To decrease in X, tilt toward left

To increase in X, tilt toward right

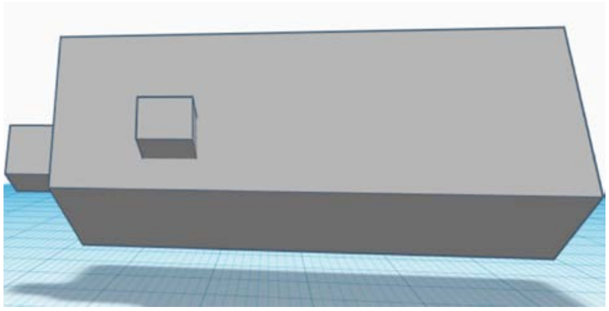
Rotate around Y-axis

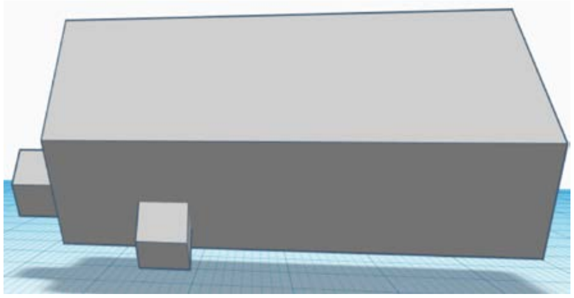
start



Tilt away from you to move ladder forward (+Y)

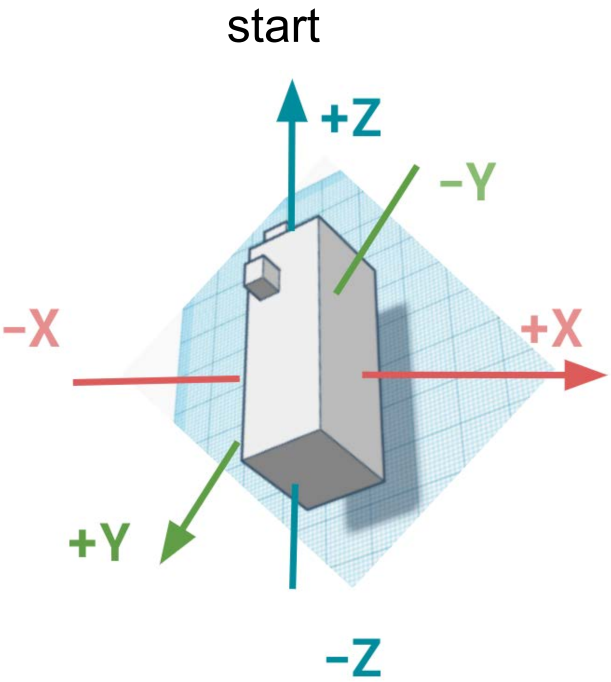
Tilt toward you to move increase in Y:

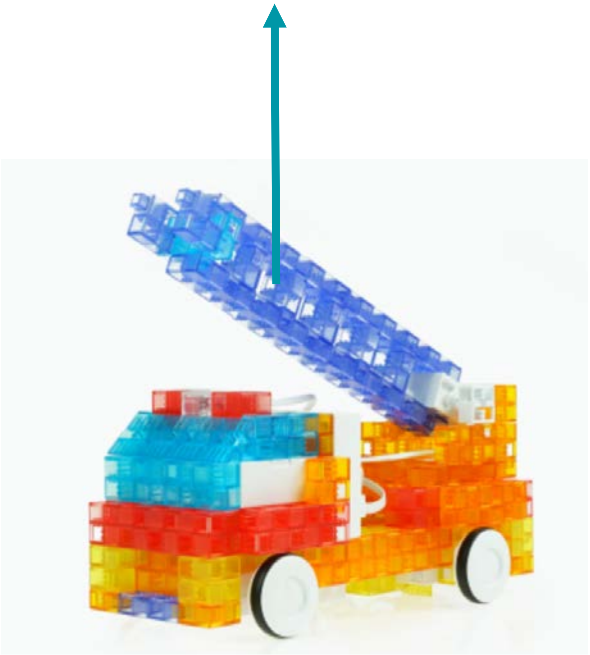




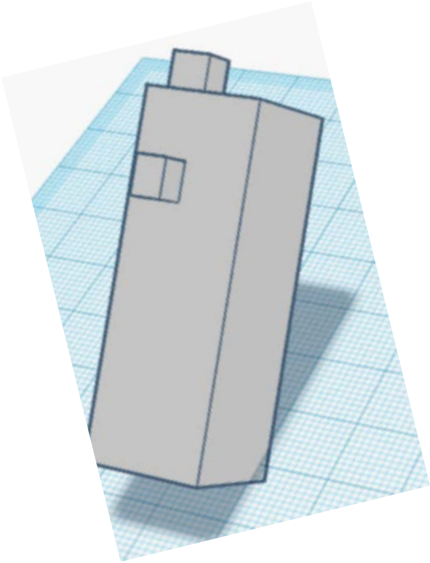
E

Rotate around Z-axis

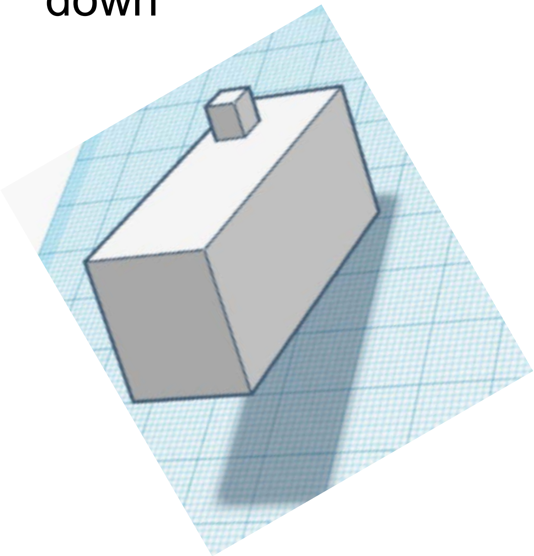




Lift up the controller (+Z) to lift up the ladder



Press down the controller (-Z) to move the ladder down



c
o
d
i
n
g

advanced

List

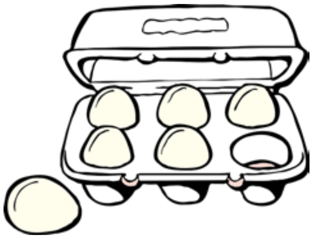
Lists

We use lists in our everyday life. If you are baking cupcakes, and you need to go shopping. You would need to put together a grocery shopping list. An example grocery shopping list could include sugar, flour, eggs, milk.

sugar

flour

eggs



milk

List name:
My Shopping List

List size: 4

List items:

1. sugar
2. flour
3. eggs
4. milk

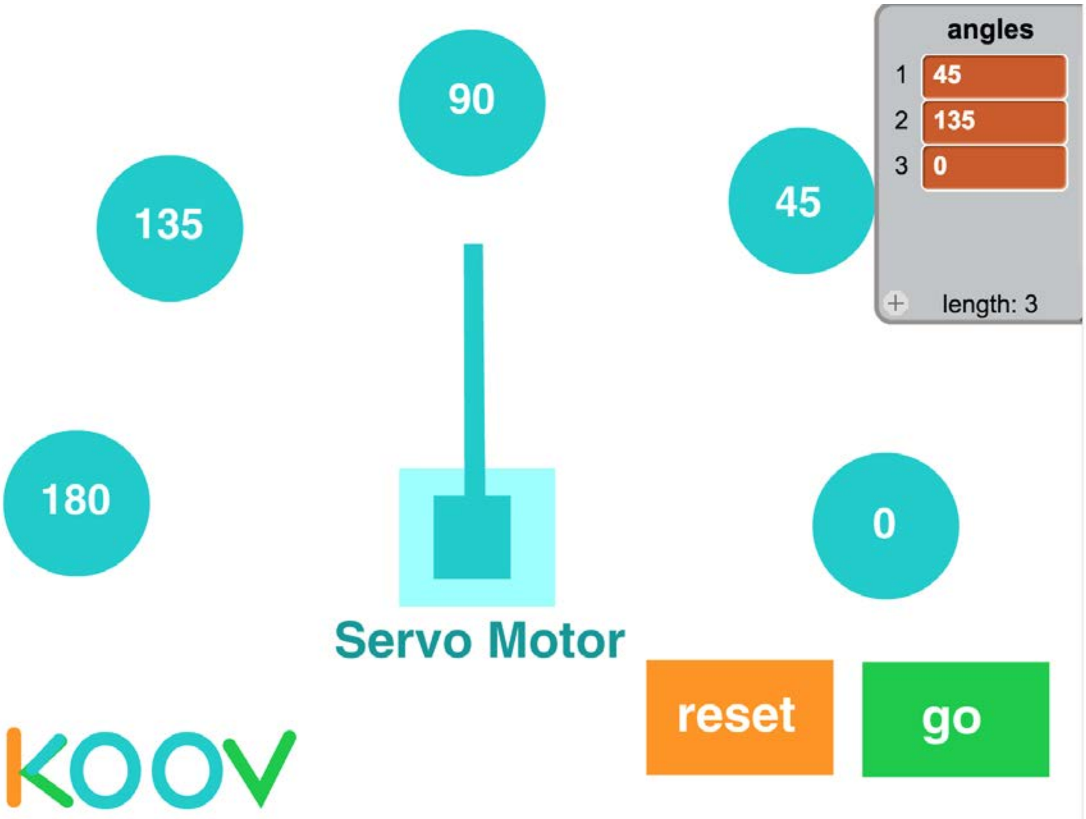
In coding, programs often store data in lists. A program can **add** something to a list, **read** something from a list, or **remove** something from a list.

Using Lists in code

demonstration

A **list** is a set of elements. In KOOV, an element can be either a numeric value or a variable.

In code, we can store data in a list and use it to perform task. For example, below is a simulator that lets you click on desired stops in a list, and when you click “go”, the Servo Motor would turn to each position in the *angles* list.



angles	
1	45
2	135
3	0
+	length: 3

Troubleshooting

Troubleshooting

E

Problem: Motor not moving

Reason 1: Battery too low
To confirm, use the Digital Multimeter or test with buzzer. If buzzer still works with the battery, then it's not the battery.

Video

Reason 2: The motor is jammed
To confirm, try a different motor
If another motor works with the same code, the teacher can open the DC motor with a screwdriver and adjust the gears.

Problem: Can move to the next step in Map

Reason 1: Code does not match
To confirm, check the tutorial.
To fix, update your code to be exactly like that in tutorial.

Reason 2: Extra variables not mentioned in tutorial
To confirm, check the data tab for unused variables.
To fix, remove extra variables.

Reason 3: Extra functions not mentioned in tutorial
To confirm, check the control tab for unused functions.
To fix, remove extra functions.

Reason 4: Extra lists not mentioned in tutorial
To confirm, check the data tab for unused lists.
To fix, remove extra lists.

Problem: Cannot see the codes

Reason 1: You accidentally delete it
To confirm, use the undo button.
To fix, undo.

Reason 2: It's hidden
To confirm, zoom out.
To fix, zoom out.

Troubleshooting

E

Problem: LED does not turn off

Reason 1: Did not set LED to off
To fix, make sure there is a LED () off line

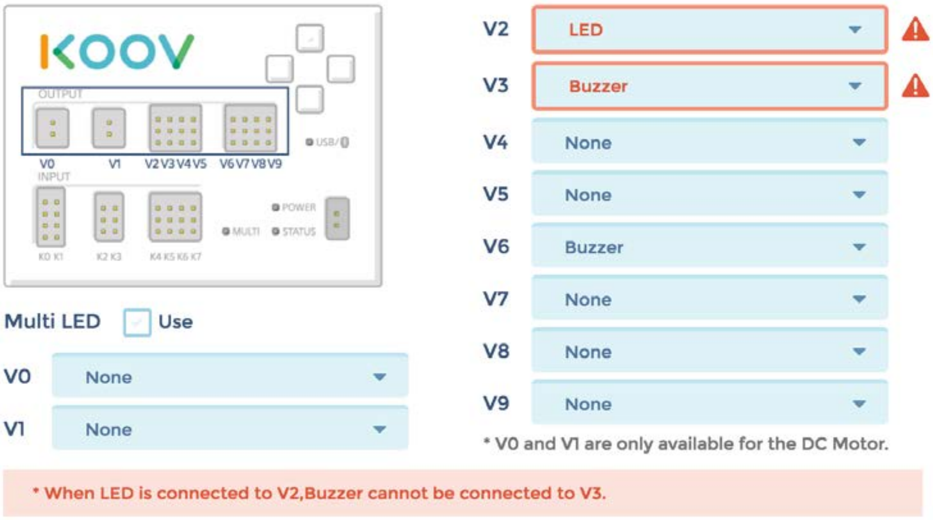
Reason 2: Wrong port
Check Connection Settings and your KOOV Core to make sure the LED ports match

Reason 3: Current too high
To fix, move to a different port.

Buzzer Connection Error

video*

Moreover, if there are other devices already connected to a section (say V1-V5), then connect the buzzer to a different section. For instance, if an LED is already connected to V2, you cannot connect a buzzer to V3, because it's in the same section, and there is not enough current to support both devices.



Multi LED □ Use

V0 None

V1 None

V2 LED ⚠

V3 Buzzer ⚠

V4 None

V5 None

V6 Buzzer

V7 None

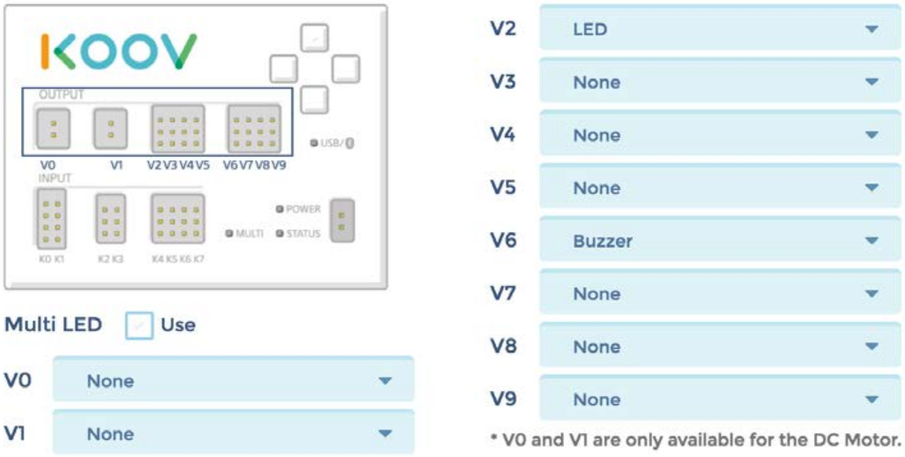
V8 None

V9 None

* V0 and V1 are only available for the DC Motor.

* When LED is connected to V2,Buzzer cannot be connected to V3.

Instead, connect the Buzzer to a port in the other section, such as V6.



Multi LED □ Use

V0 None

V1 None

V2 LED

V3 None

V4 None

V5 None

V6 Buzzer

V7 None

V8 None

V9 None

* V0 and V1 are only available for the DC Motor.

KOOV

Robotics Camp

Fun, Learning, and Discovery

Student Manual

Day 5 Objectives

E Servo Motor

C list

L Penguin

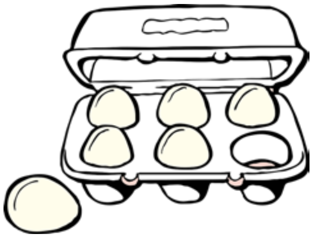
Day 4 Concept: list

We use lists in our everyday life. If you are baking cupcakes, and you need to go shopping. You would need to put together a grocery shopping list. An example grocery shopping list could include sugar, flour, eggs, milk.

sugar

flour

eggs



milk

List name: My Shopping List

List size: 4

List items:
1. sugar
2. flour
3. eggs
4. milk

What is the first item in the list?

____

What is the third item in the list?

____

In coding, programs often store data in lists. A program can **add** something to a list, **read** something from a list, or **remove** something from a list.

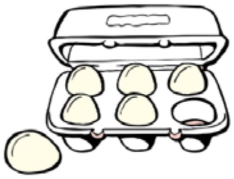
Day 4 Concept: working with list

Let's try practice working with list. Answer the question about this list after a program changes it. Note that when adding, an item is added to the end of a list.

sugar

flour

eggs



milk

```
start
remove item [3] from list
add [chocolate] to list
```

List name: My Shopping List

List size of the new list: ____________________

List Items in the new list:

1. ___________________________

1. ___________________________

1. ___________________________

1. ___________________________

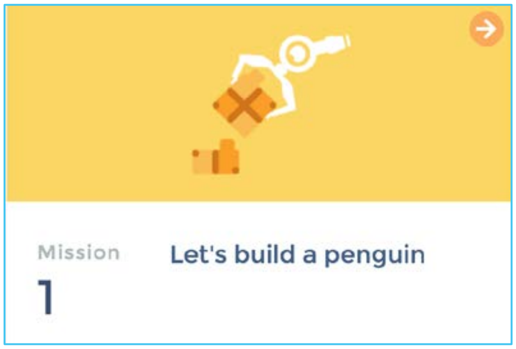
1. ___________________________

The 3rd item in the list: ___________________

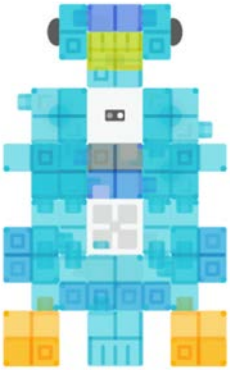
Day 5 Map Stage 6 Mission 1: Let's build a penguin

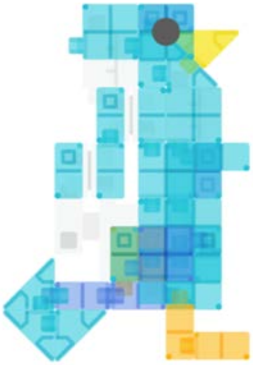
This mission's goal

Build a penguin with Servo Motor, IR Photoreflector, and KOOV Blocks.



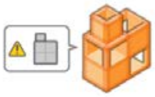
Mission 1 Let's build a penguin





Parts you need

Blue Half-C x 35



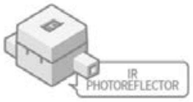
Orange Half-C x 7

Purple Half-D x 8

Yellow Prism x 1

Blue Prism x 4

Buzzer x 1



IR Photoreflector x 1

Servo Motor x 1

360 Connector x 4

Eye x 2

Core x 1

Battery Box x 1

(with batteries)

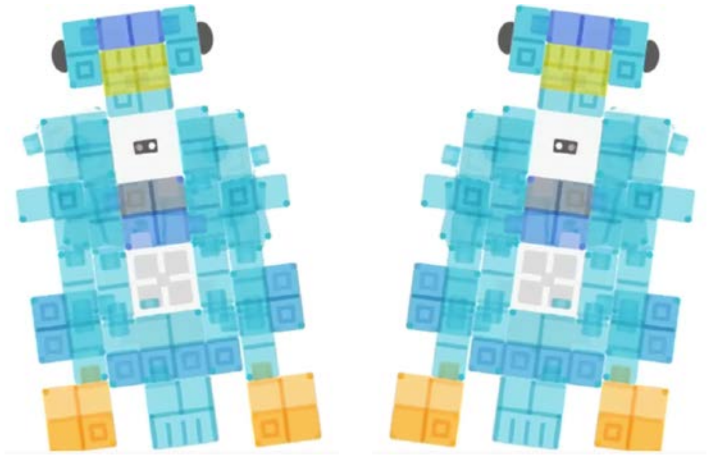
Base Tray x 2

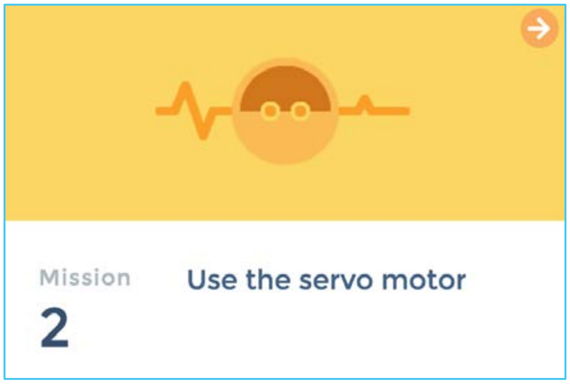
3-Core Cable (20cm) x 2

Day 5 Map Stage 6 Mission 2: Use the servo motor

This mission's goal

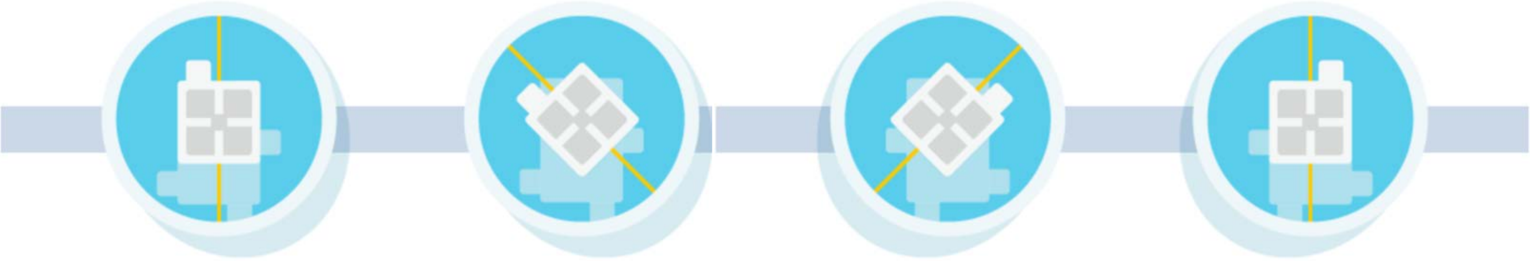
Use Servo Motor to make the penguin dance side to side.



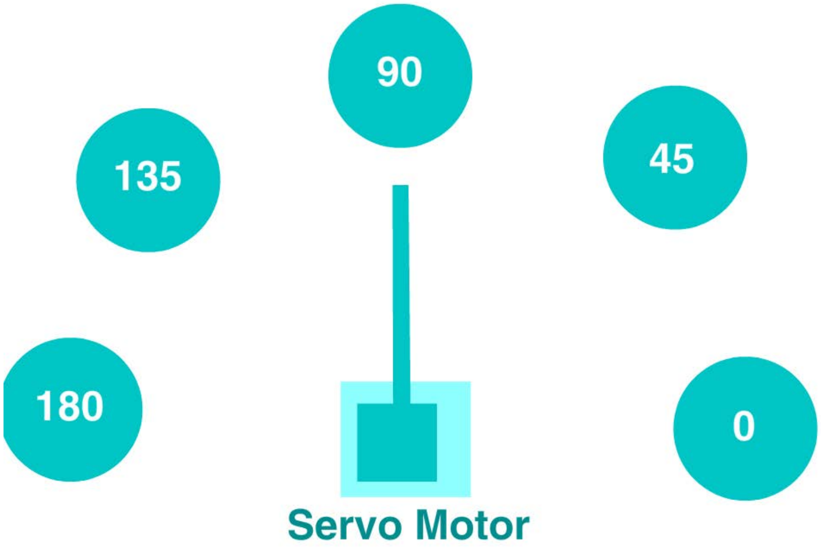


Mission
2

Use the servo motor



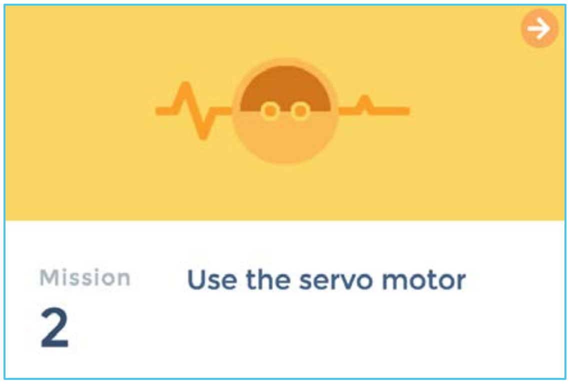
The Servo Motor points to 90 degree by default.



Day 5 Map Stage 6 Mission 2: Use the servo motor

This mission's goal

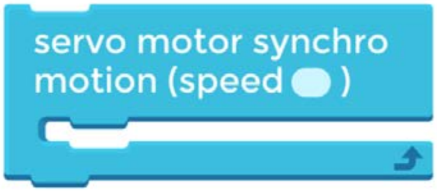
Use Servo Motor to make the penguin dance side to side.



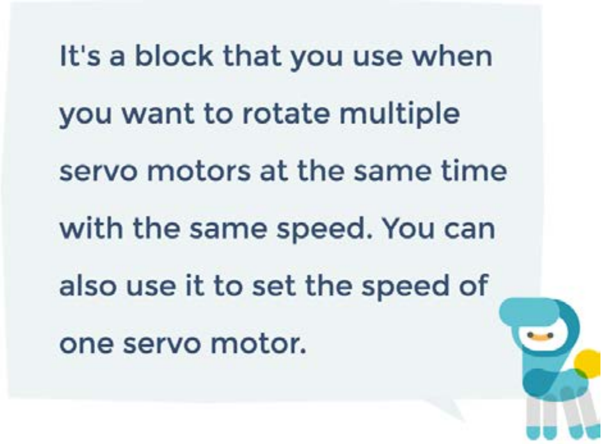
Mission

2

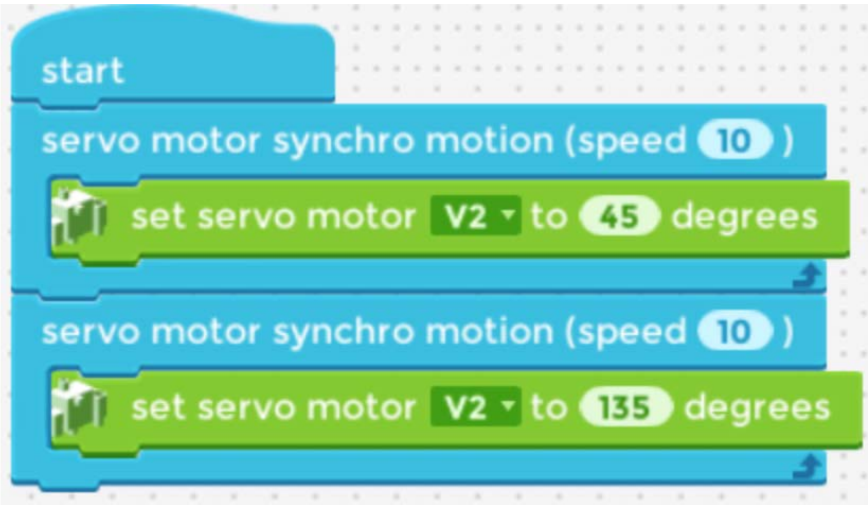
Use the servo motor



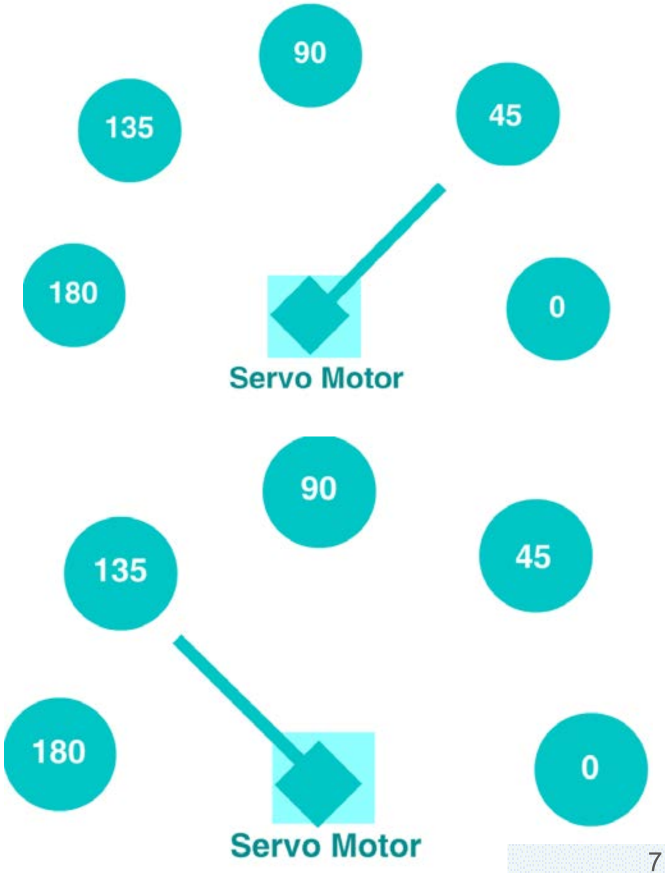
servo motor synchro motion (speed)



It's a block that you use when you want to rotate multiple servo motors at the same time with the same speed. You can also use it to set the speed of one servo motor.



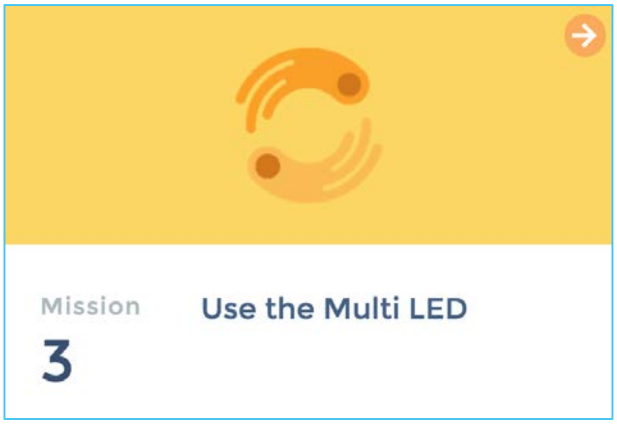
```
start
servo motor synchro motion (speed 10 )
    set servo motor V2 to 45 degrees
servo motor synchro motion (speed 10 )
    set servo motor V2 to 135 degrees
```



Day 5 Map Stage 6 Mission 3: Use the Multi LED

This mission's goal

Learn to use the Multi LED, and how to make a color with light.

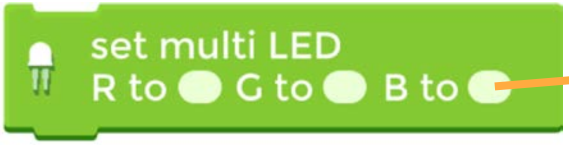


Mission

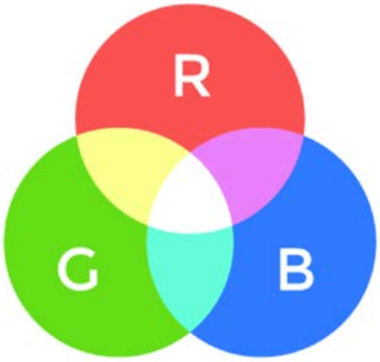
3

Use the Multi LED

Red, **G**reen and **B**lue are the **primary colors** of light. Mixing the same amount of these colors make another color:

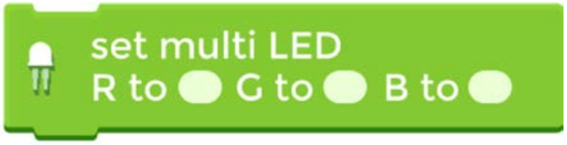


value from 0 to 100

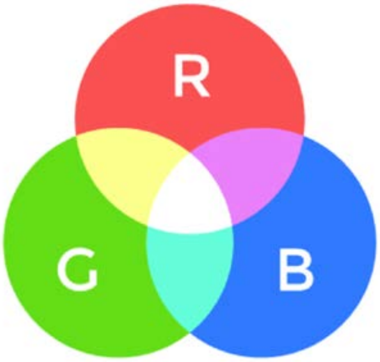


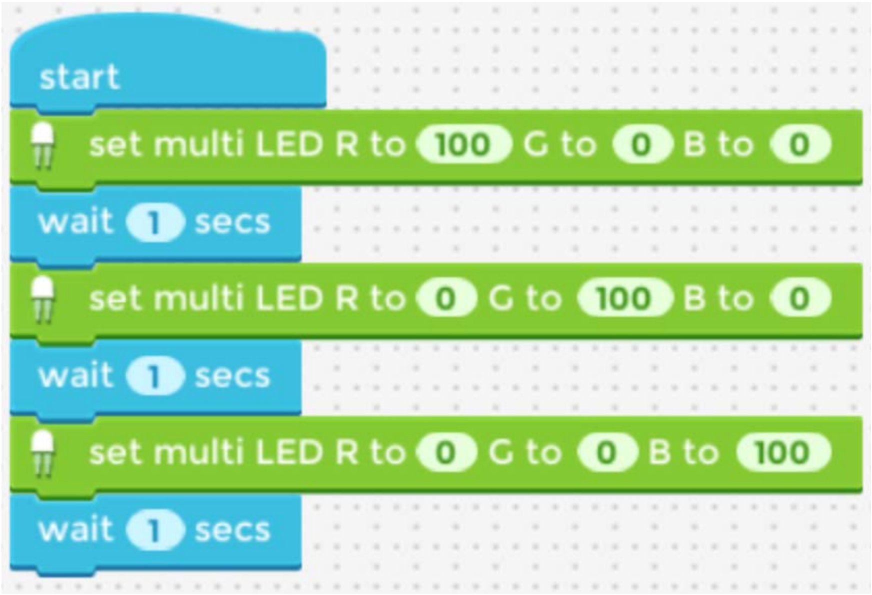
Day 5 Map Stage 6 Mission 3: Use the Multi LED

Red	Green	Blue	Result
100	0	0	Red
0	100	0	Green
0	0	100	Blue
100	100	0	Yello
0	100	100	Cyan
100	0	100	Magenta
100	100	100	White



```
set multi LED
R to ⬭ G to ⬭ B to ⬭
```





```
start
set multi LED R to 100 G to 0 B to 0
wait 1 secs
set multi LED R to 0 G to 100 B to 0
wait 1 secs
set multi LED R to 0 G to 0 B to 100
wait 1 secs
```

What color pattern does the Multi LED light up with the program above?

Day 5 Lab: Stage 6 Mission 4 Use random numbers

1) What does this code do? What possible numbers would this code block return?

```
pick a random 1 to 10
```

Answer: ____

1) What would the variable penguin be?

```
set variable penguin to ( 10 ) - ( 2 )
```

Answer: ____

1) What is the value variable big_penguin after running the following code?

```
set variable penguin to ( 10 ) - ( 2 )
set variable big_penguin to ( penguin ) - ( 2 )
```

Answer: ____

1) What does this code do?

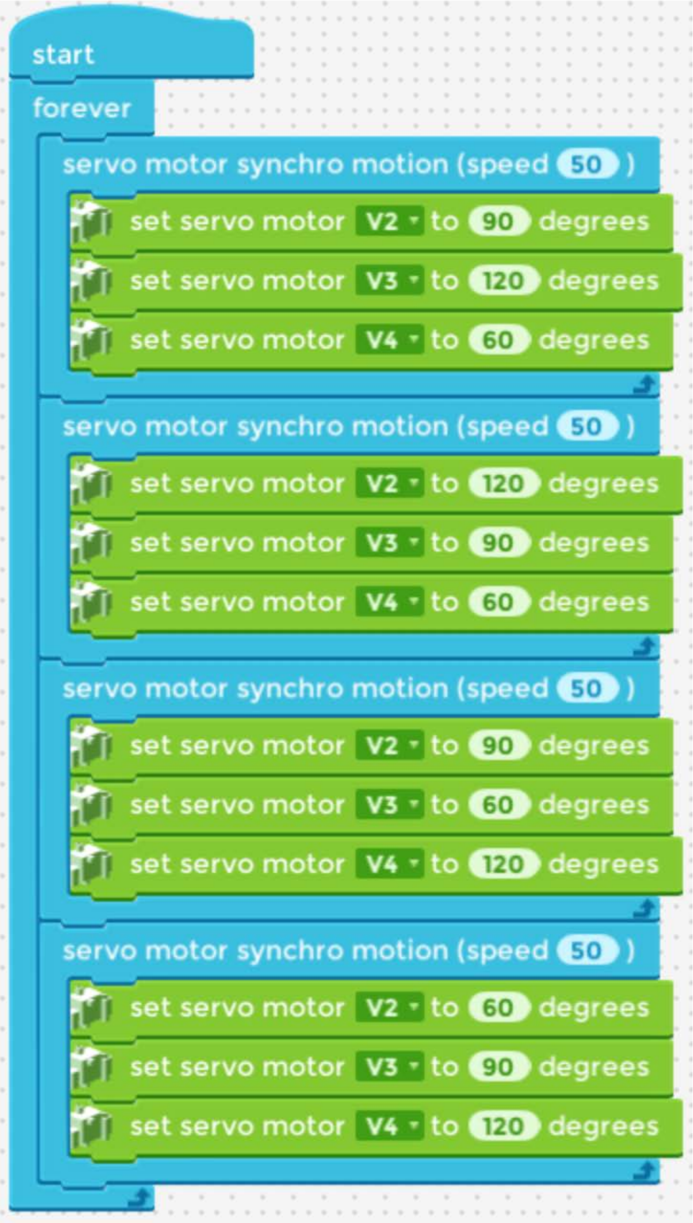
```
servo motor synchro motion (speed (50))
set servo motor (V2) to (90) degrees
set servo motor (V3) to (120) degrees
set servo motor (V4) to (60) degrees
```

a) Jump
b) Dance
c) Sing

Answer: ____

Day 5 Lab: Buddy from Robot Recipe

Build Buddy following the Robot Recipe. You will use three Servo Motors to make its legs turn. Build and load the code to Buddy and test.



```
start
forever
    servo motor synchro motion (speed 50 )
        set servo motor V2 to 90 degrees
        set servo motor V3 to 120 degrees
        set servo motor V4 to 60 degrees
    servo motor synchro motion (speed 50 )
        set servo motor V2 to 120 degrees
        set servo motor V3 to 90 degrees
        set servo motor V4 to 60 degrees
    servo motor synchro motion (speed 50 )
        set servo motor V2 to 90 degrees
        set servo motor V3 to 60 degrees
        set servo motor V4 to 120 degrees
    servo motor synchro motion (speed 50 )
        set servo motor V2 to 60 degrees
        set servo motor V3 to 90 degrees
        set servo motor V4 to 120 degrees
```

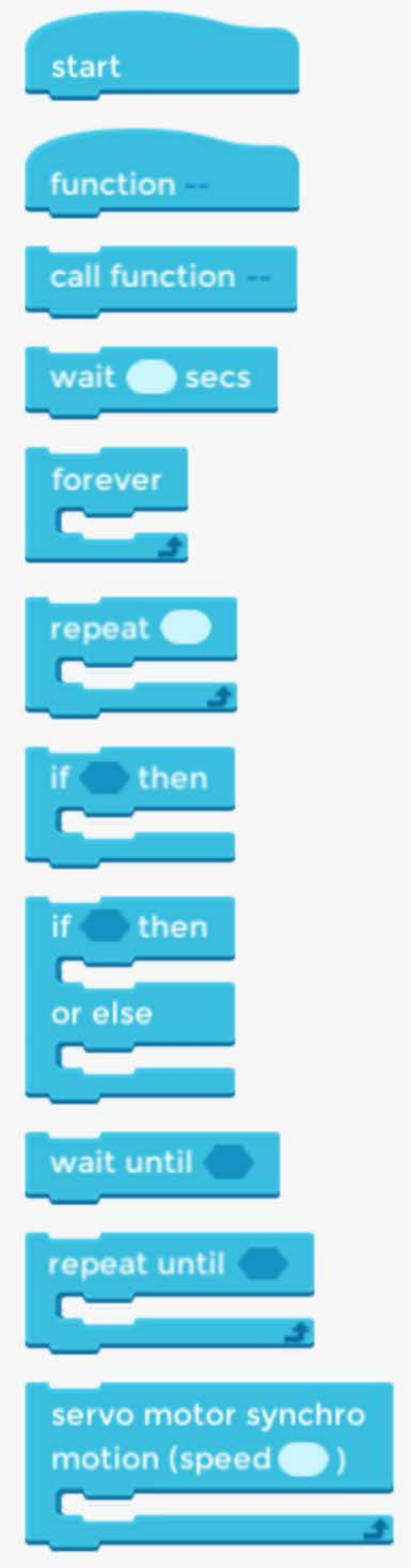
What does this code do?

```
Servo motor synchro motion (speed (50))
Set servo motor (V2) to (90) degrees
Set servo motor (V3) to (120) degrees
Set servo motor (V4) to (60) degrees
```

Can you make it dance faster?

Code Blocks Used

Control



start

function --

call function --

wait ◯ secs

forever

repeat ◯

if ⬡ then

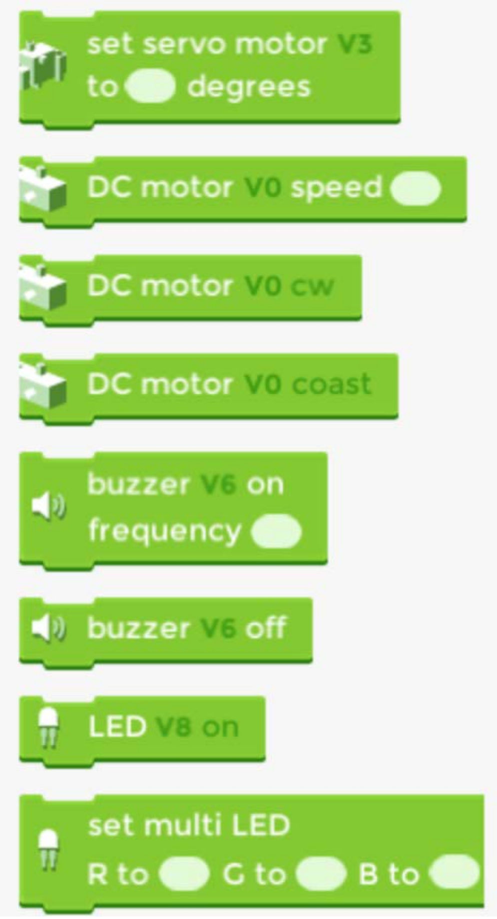
if ⬡ then
or else

wait until ⬡

repeat until ⬡

servo motor synchro motion (speed ◯)

Motion



set servo motor V3 to ◯ degrees

DC motor V0 speed ◯

DC motor V0 cw

DC motor V0 coast

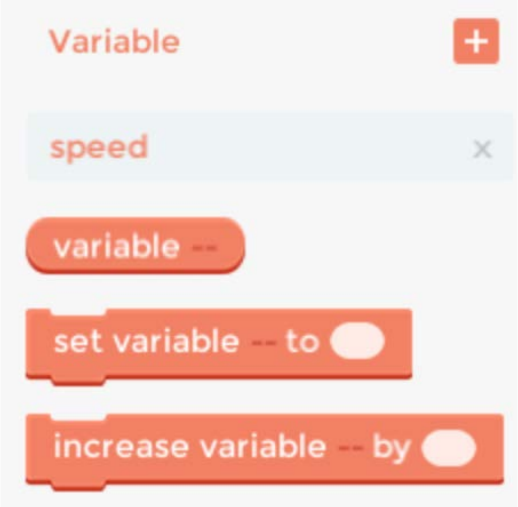
buzzer V6 on frequency ◯

buzzer V6 off

LED V8 on

set multi LED R to ◯ G to ◯ B to ◯

Variable

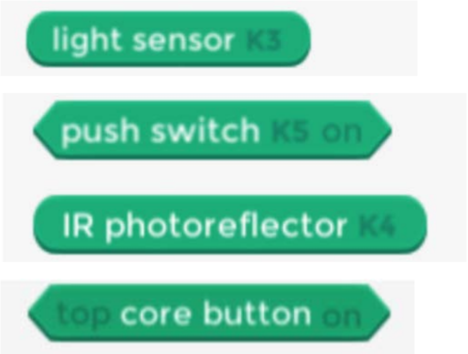


variable --

set variable -- to ◯

increase variable -- by ◯

Sensing



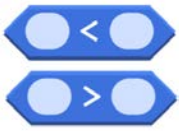
light sensor K3

push switch K5 on

IR photoreflector K4

top core button on

Operator



◯ < ◯

◯ > ◯

Value Range

Sound Sensor	0 to 100
Light Sensor	0 to 100
IR Photoreflector	0 to 100
DC Motor Speed	0 to 100
Servo Motor Speed	0 to 100
Servo Motor Degrees	0 to 180
Buzzer Frequency	48 to 108
Accelerometer	-100 to 100

Troubleshooting page 1

Problem: Motor not moving

Reason 1: Battery too low
Ask the adults to check or replace with new batteries.

Reason 2: The motor is jammed
After replacing the battery, if the motor still does not work, then ask the adults for help.

Problem: Can move to the next step in Map

Reason 1: Code does not match
To confirm, check the tutorial.
To fix, update your code to be exactly like the tutorial.

Reason 2: Extra variables not mentioned in tutorial
To confirm, check the data tab for unused variables.
To fix, remove extra variables.

Reason 3: Extra functions not mentioned in tutorial
To confirm, check the control tab for unused functions.
To fix, remove extra functions.

Reason 4: Extra lists not mentioned in tutorial
To confirm, check the data tab for unused lists.
To fix, remove extra lists.

Problem: Cannot see the codes

Reason 1: You accidentally deleted it
To confirm, use the undo button.
To fix, undo.
Reason 2: It's hidden
To confirm, zoom out.
To fix, zoom out.

Troubleshooting page 2

Problem: LED does not turn off

Reason 1: Did not set LED to off
To fix, make sure there is a LED () off line

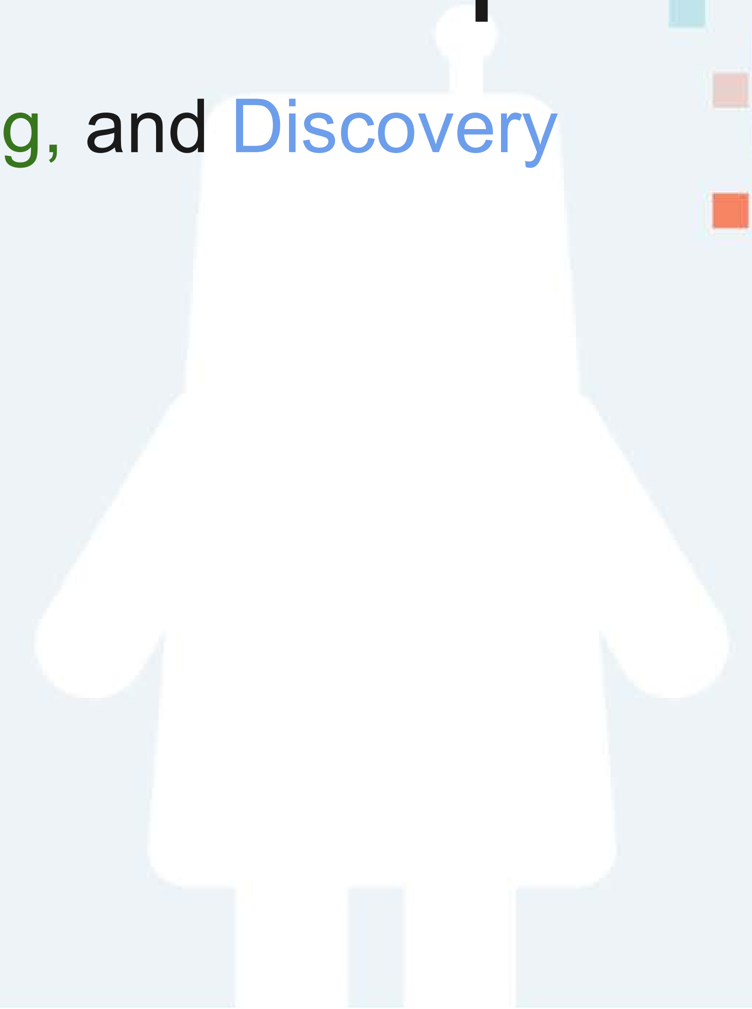
Reason 2: Wrong port
Check Connection Settings and your KOOV Core to make sure the LED ports match

Reason 3: Current too high
To fix, move to a different port.

KOOV

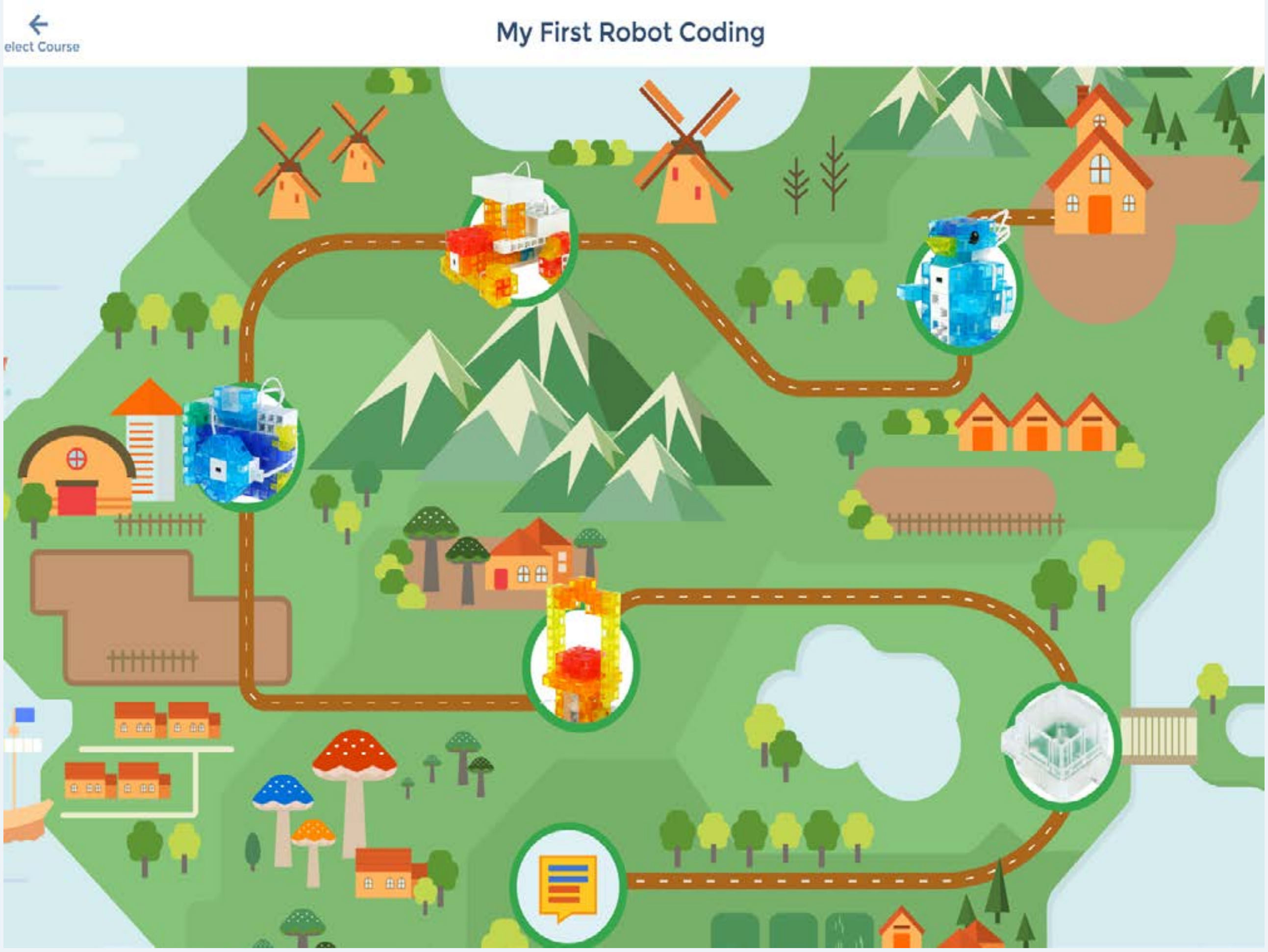
Robotics Camp

Fun, Learning, and Discovery



Fun Road Ahead

Welcome to the KOOV Robotics Camp. At this camp, you are going to put on the hat of a student robotic engineer and learn by building. You will build and program several “robots” each day,following instructions in this curriculum and from My First Robot Coding Map. Then you will be given a Design Challenge to brainstorm for better design. Each day will be filled with learning, fun, and discovery.



Week In A Glance

Day 1: KOOV Education Suite

Learn about Robotics and KOOV

Day 2: Blink and Beep!

Coding with LEDs and Buzzer

Day 3: Light and Camera

Detect Light and Distance

Day 4: Turn that Wheels

DC Motor and Robot Car

Day 5: Dancing Feet

Servo Motor and Penguin

DAY 5

Dancing Feet

Day 5 Objectives

E

Servo Motor

C

List
Wait Until
Repeat Until

M

Penguin

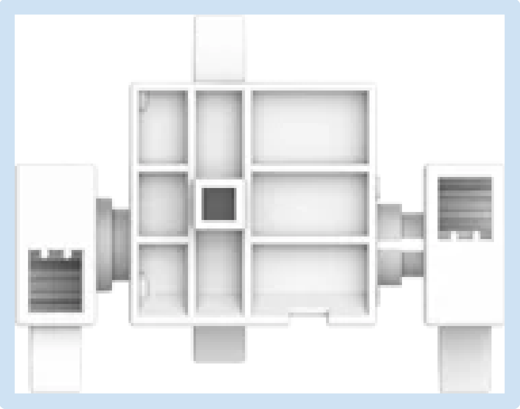
Electronics

Servo Motor

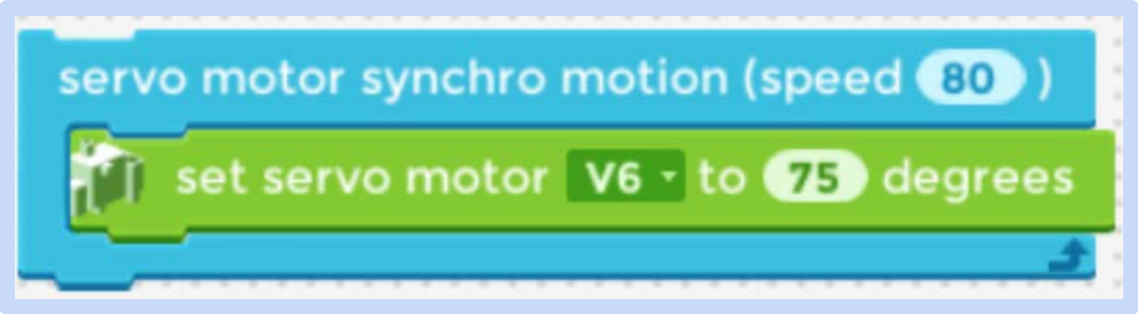
KOOV Servo Motor

Video: Alligator

There are two type of KOOV motors, one is a rotating DC Motor, the other is the Servo Motor. A KOOV DC Motor can turn clockwise and counterclockwise at a given speed, and the Servo Motor provides finer control and let you turn to a specific angle at a specific speed. Servo Motor usually has a servo arm or shaft that can turn 0-180 degrees.



The KOOV Servo Motor can be controlled via KOOV Code Blocks. It can be programmed to rotate at a speed to a specific angular(angular stands for angle) location. The codes below shows how to turn the shaft at speed 80 to 75 degrees.



```
servo motor synchro motion (speed 80 )
    set servo motor V6 to 75 degrees
```

E

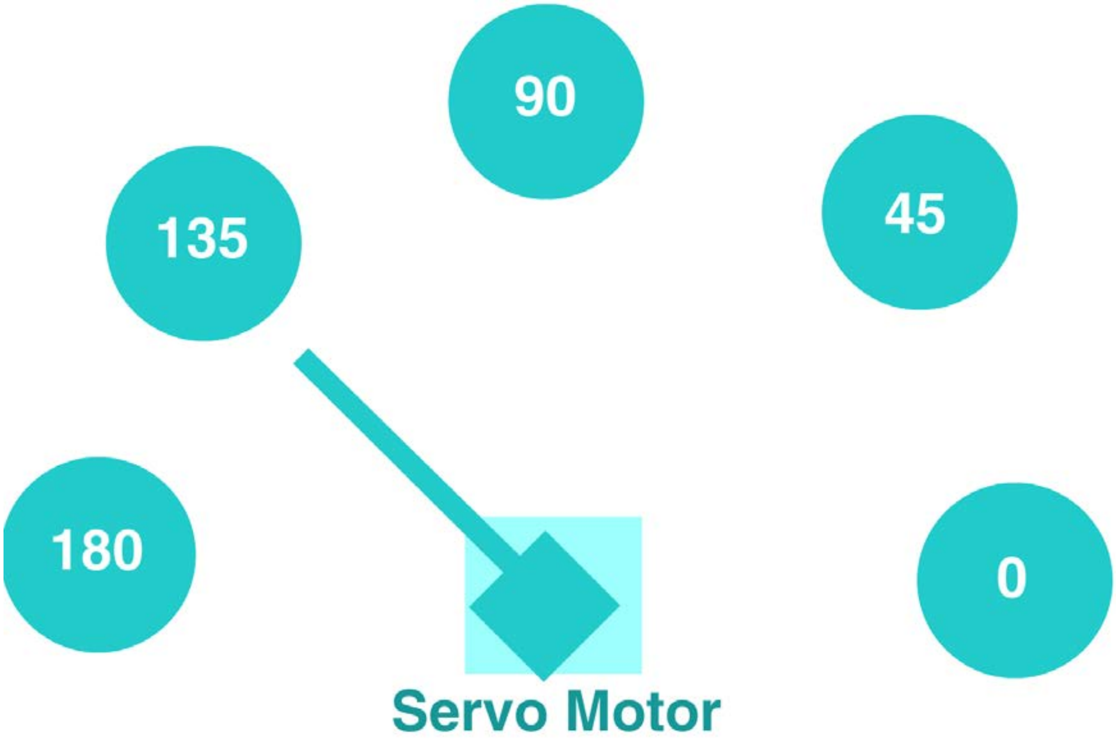
KOOV Servo Motor

simulation

Video: Rotate at different Speed

Video: Change Angles

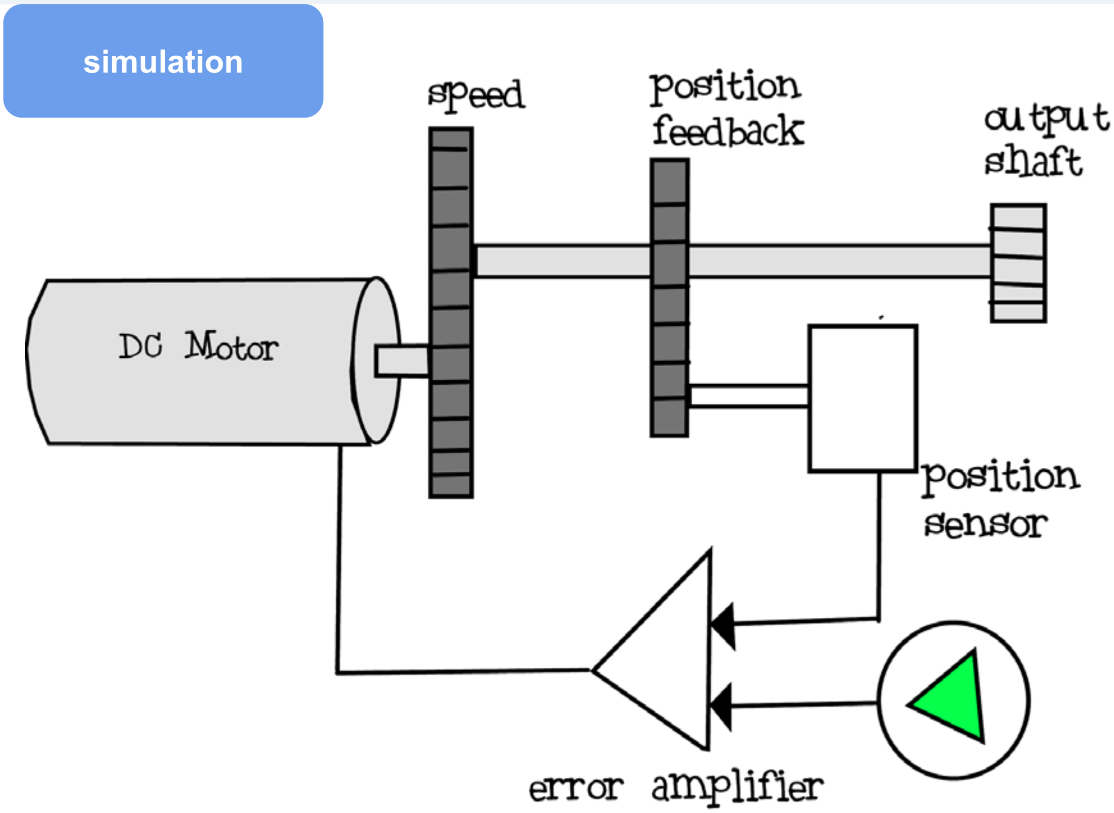
A KOOV Servo Motor can rotate between 0 to 180 degrees. The illustration below shows the location of the shaft when the code rotates a servo motor to different positions.



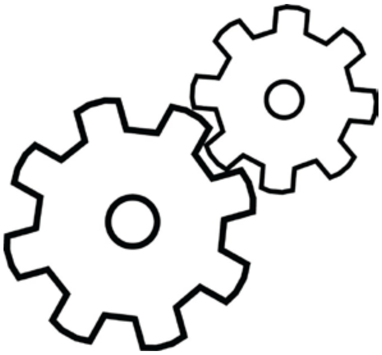
KOOV

Look Inside a Servo Motor

simulation



A Servo Motor contains a DC Motor, gears, a position sensor, and an error amplifier. DC Motor rotates the shaft; gears help control rotation; position sensor provides current location; error amplifier tracks the difference between current and desired location. Together it forms a control loop that keeps going until the desired location is reached.



coding

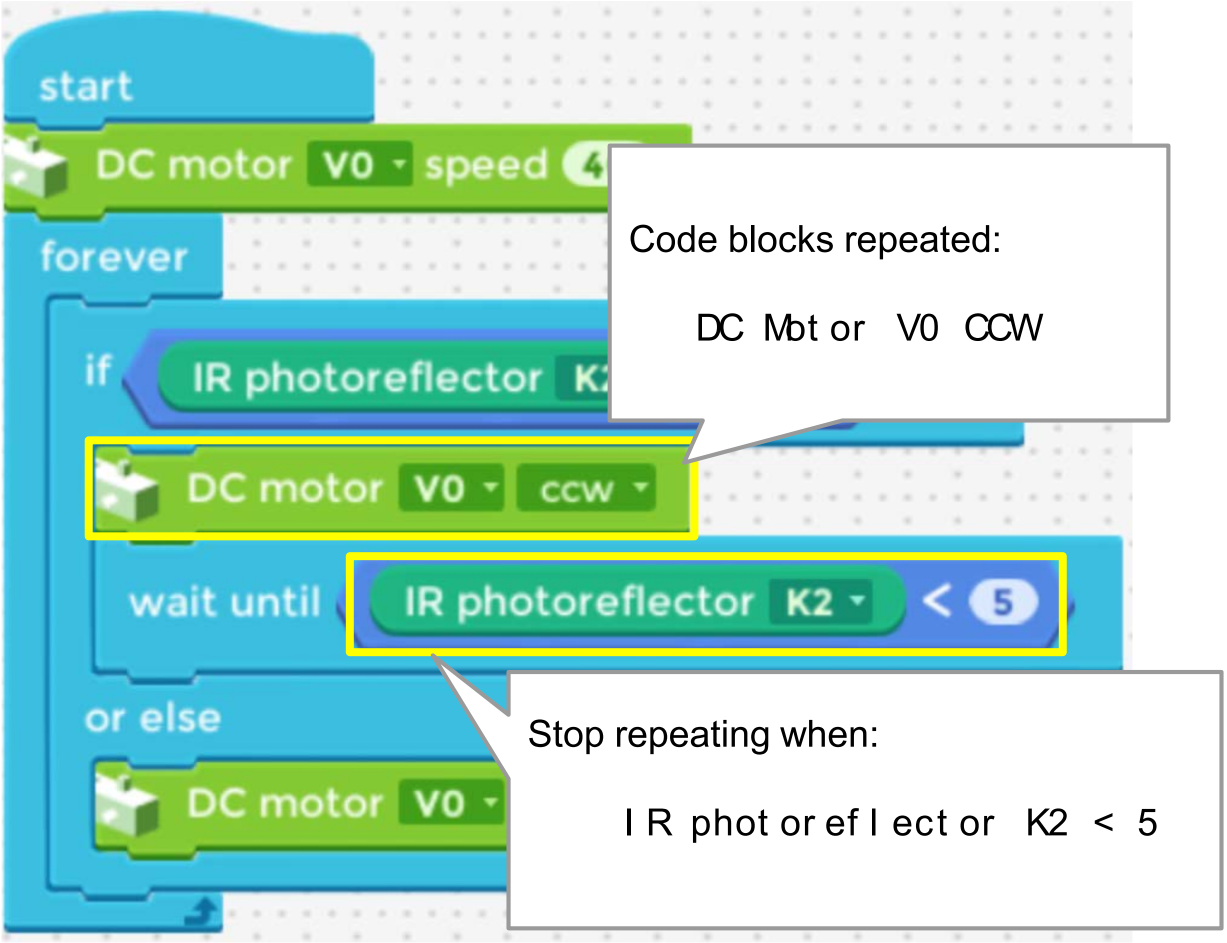
Wait Until ...

Repeat Until ...

Wait until

The wait until block is to hold the current state until a condition is true. For example, a chef could be boiling the water to cook pasta, and she will turn on the fire, and wait until the pasta is done.

The code below is the codes for a tractor. It shows that the DC Motor would keep rotating counterclockwise (DC Motor V0 ccw) until the car's proximity detector (IR photoreflector) is not close to anything.



```
start
DC motor V0 speed 4
forever
    if IR photoreflector K2
        DC motor V0 ccw
        wait until IR photoreflector K2 < 5
    or else
        DC motor V0
```

Code blocks repeated:

DC Motor V0 CCW

Stop repeating when:

IR photoreflector K2 < 5

C

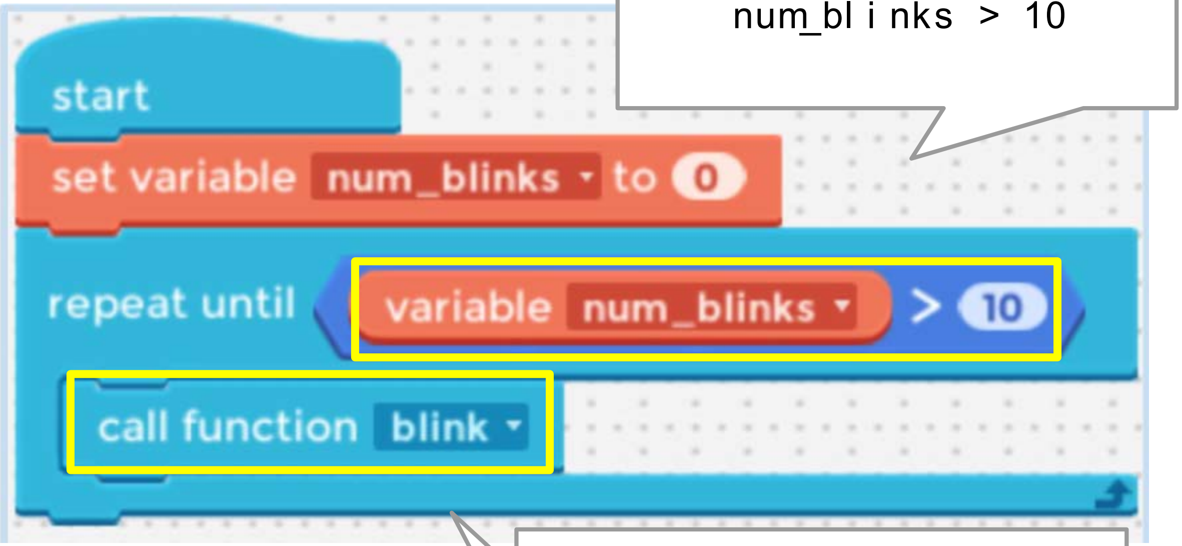
Repeat until

The **repeat until ...** code block is used to repeat a set of steps that could be inside a function, until a condition is true.

An example in our daily life could be:

- Repeat washing until it's clean
- Repeat playing until I am tired

The codes below make an LED keep blinking (via calling function "blink") 10 times.



```
start
set variable num_blinks to 0
repeat until variable num_blinks > 10
    call function blink
```

Condition to check:

`num_blinks > 10`

Code blocks repeated:

`call function blink`

M
a
p

Penguin

M

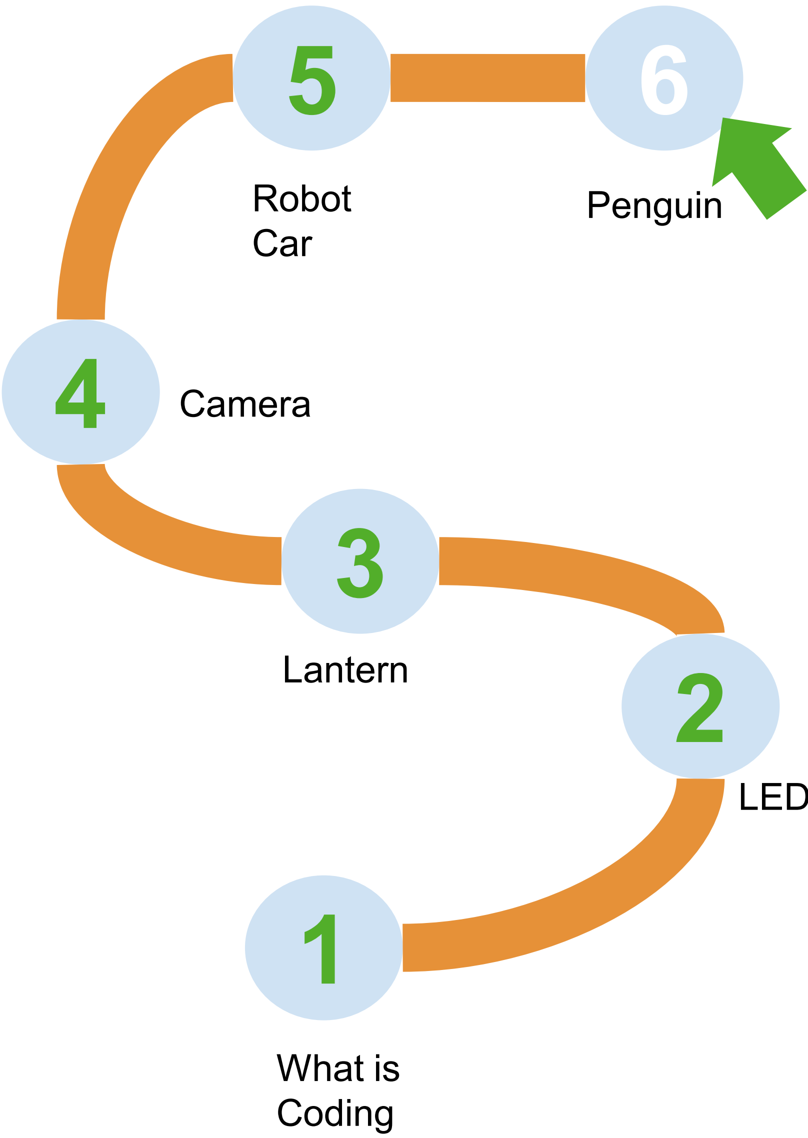
Robotic Coding Map: Penguin

Map: My First Robotic Coding
Stage: Stage Five

Mission 1: Let's build a penguin
Mission 2: Use the servo motor
Mission 3: Use the Multi LED
Mission 4: Use variables

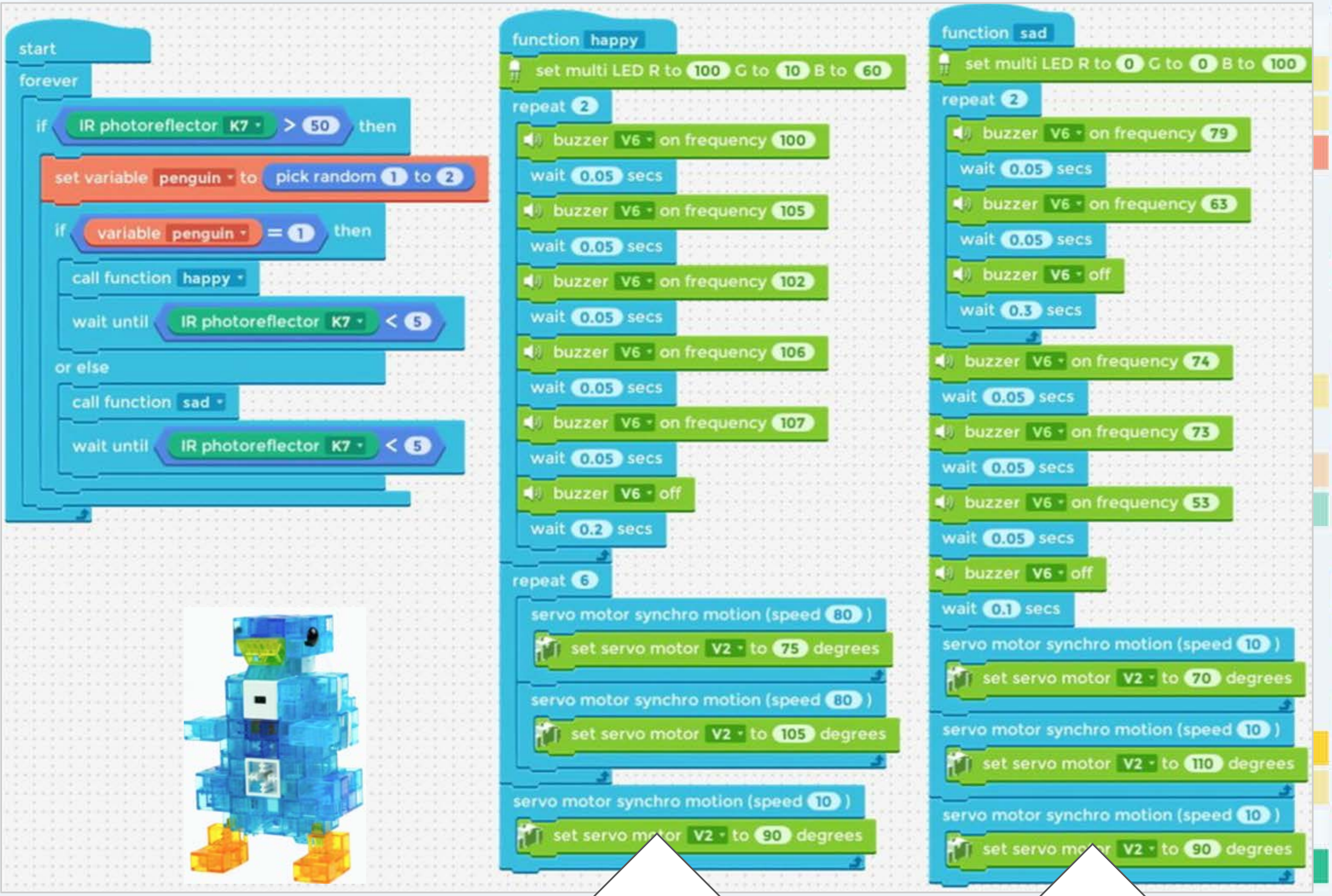
To go to Map:

1. Login
2. Select "Learning Course"
3. Select "My First Robot Coding"



M

Robotic Coding Map: Penguin



```
start
forever
    if IR photoreflector K7 > 50 then
        set variable penguin to pick random 1 to 2
        if variable penguin = 1 then
            call function happy
            wait until IR photoreflector K7 < 5
        or else
            call function sad
            wait until IR photoreflector K7 < 5
```

```
function happy
set multi LED R to 100 G to 10 B to 60
repeat 2
    buzzer V6 on frequency 100
    wait 0.05 secs
    buzzer V6 on frequency 105
    wait 0.05 secs
    buzzer V6 on frequency 102
    wait 0.05 secs
    buzzer V6 on frequency 106
    wait 0.05 secs
    buzzer V6 on frequency 107
    wait 0.05 secs
    buzzer V6 off
    wait 0.2 secs
repeat 6
    servo motor synchro motion (speed 80 )
        set servo motor V2 to 75 degrees
    servo motor synchro motion (speed 80 )
        set servo motor V2 to 105 degrees
servo motor synchro motion (speed 10 )
    set servo motor V2 to 90 degrees
```

```
function sad
set multi LED R to 0 G to 0 B to 100
repeat 2
    buzzer V6 on frequency 79
    wait 0.05 secs
    buzzer V6 on frequency 63
    wait 0.05 secs
    buzzer V6 off
    wait 0.3 secs
buzzer V6 on frequency 74
wait 0.05 secs
buzzer V6 on frequency 73
wait 0.05 secs
buzzer V6 on frequency 53
wait 0.05 secs
buzzer V6 off
wait 0.1 secs
servo motor synchro motion (speed 10 )
    set servo motor V2 to 70 degrees
servo motor synchro motion (speed 10 )
    set servo motor V2 to 110 degrees
servo motor synchro motion (speed 10 )
    set servo motor V2 to 90 degrees
```

Function: happy

Function: sad

Competition and Games

Competition

Competition

- □ Giant Robots
- □ Build the Cubes
- □ S.O.S
- □ Speed Building
- □ Train Race

Design a Giant Robot

Design Challenge - Giant Robot

Imagine if you had an infinite number of KOOV blocks (of any sizes and color) and KOOV electric parts, can you build these?

Design a large eye that can turn around and look

- What electric parts might you need?
- What problems may you run into?
- What can you use this robot for?

Design a robot foot that can wiggle all its toes

- What electric parts might you need?
- What problems may you run into?
- What can you use this robot for?

Design a robot hand that can hold a cup

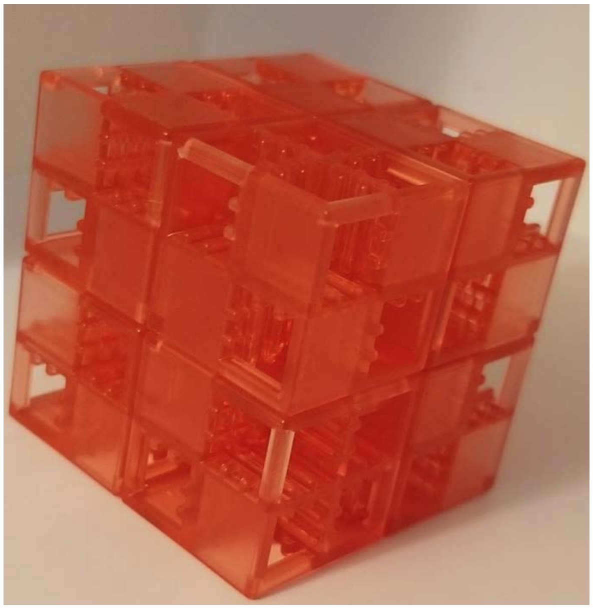
- What electric parts might you need?
- What problems may you run into?
- What can you use this robot for?

Build the Cubes

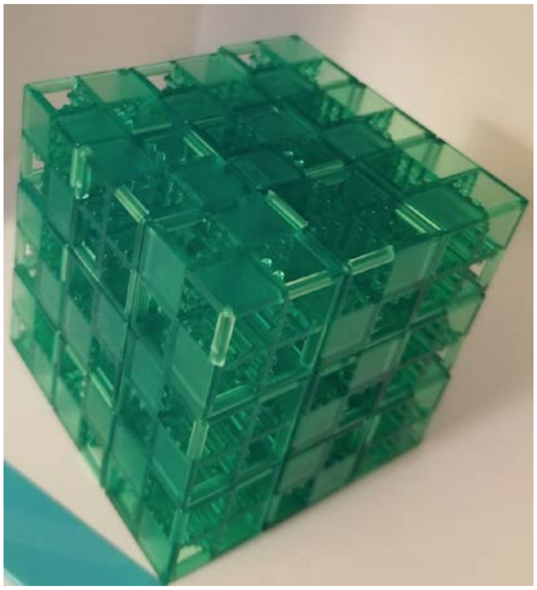
Build cubes:
 2 blocks by 2 blocks and
 3 blocks by 3 blocks

Challenge:
 How about 4 blocks by 4 blocks?

2 x 2

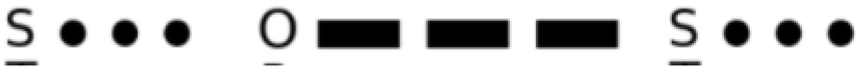


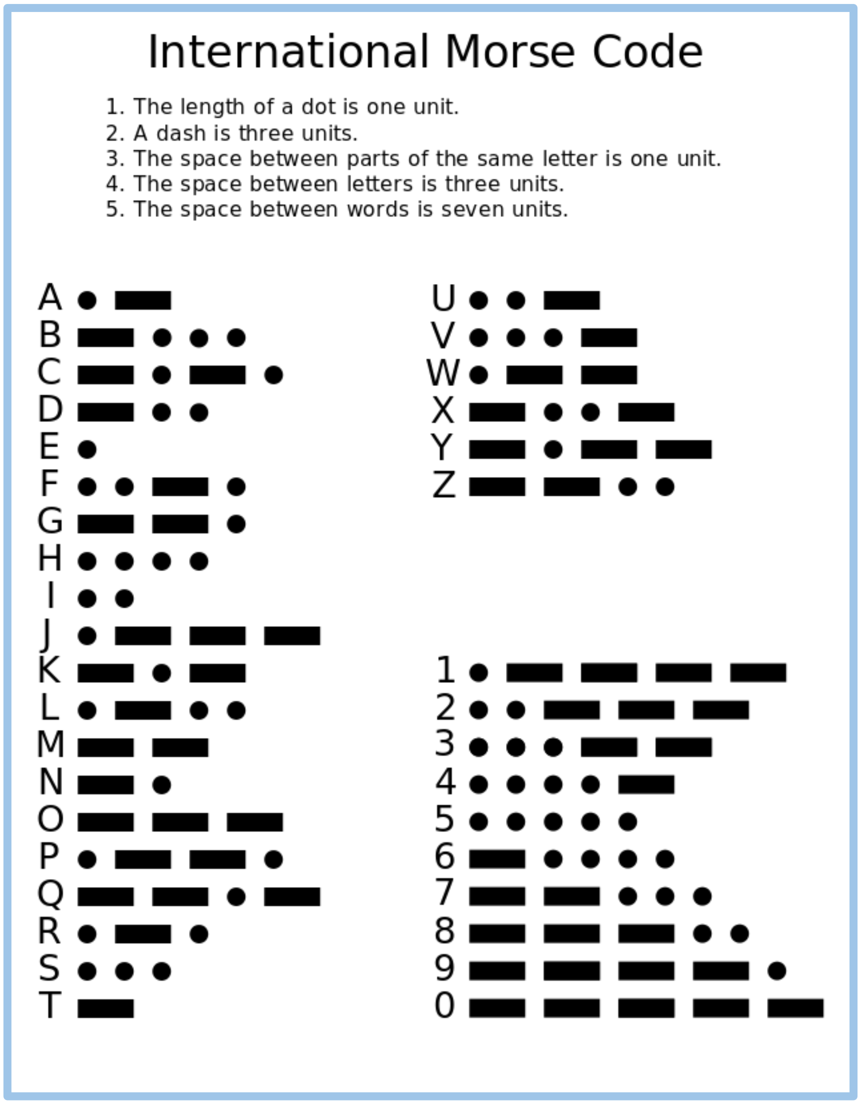
3 x 3



S.O.S

Morse code is a method of transmitting text information as a series of on-off tones, lights, or clicks. You can send S.O.S Morse code distress signals by using the KOOV LED and Buzzer. Build a program to blink and beep S.O.S in Morse Code: short, short, short, long, long, long, short, short, short:





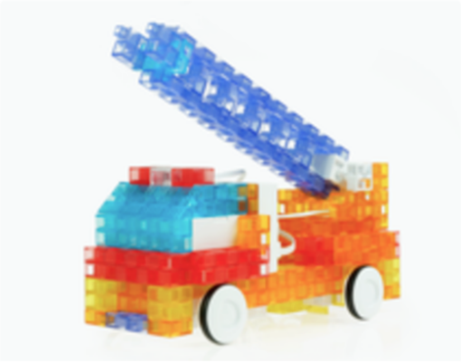
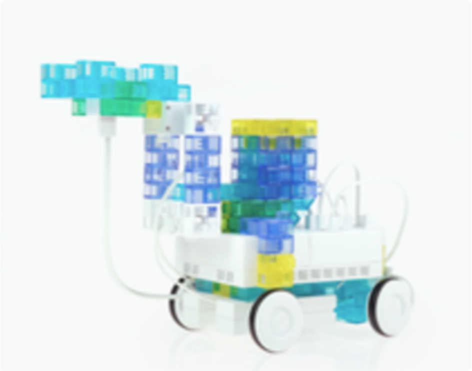
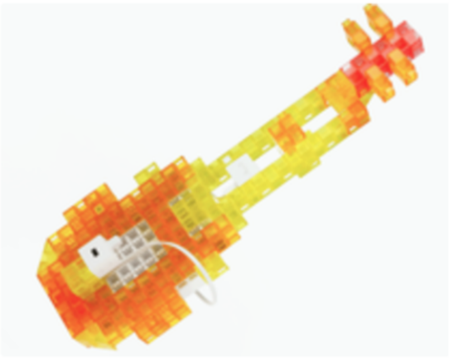
International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

A	● ▬	U	● ● ▬
B	▬ ● ● ●	V	● ● ● ▬
C	▬ ● ▬ ●	W	● ▬ ▬
D	▬ ● ●	X	▬ ● ● ▬
E	●	Y	▬ ● ▬ ▬
F	● ● ▬ ●	Z	▬ ▬ ● ●
G	▬ ▬ ●		
H	● ● ● ●		
I	● ●		
J	● ▬ ▬ ▬		
K	▬ ● ▬	1	● ▬ ▬ ▬ ▬
L	● ▬ ● ●	2	● ● ▬ ▬ ▬
M	▬ ▬	3	● ● ● ▬ ▬
N	▬ ●	4	● ● ● ● ▬
O	▬ ▬ ▬	5	● ● ● ● ●
P	● ▬ ▬ ●	6	▬ ● ● ● ●
Q	▬ ▬ ● ▬	7	▬ ▬ ● ● ●
R	● ▬ ●	8	▬ ▬ ▬ ● ●
S	● ● ●	9	▬ ▬ ▬ ▬ ●
T	▬	0	▬ ▬ ▬ ▬ ▬

Speed Building

See how many robots a team can build in an hour. Each team selects the robots they want to build. Once they finish a robot, they raise their hand, and the mentor will check it and give points for accuracy, up to the max points listed below.

Recipe	Picture
Firetruck (Difficult) Where: From KOOV Robot Recipe Point: 20	
ForkLift (Medium) Where: From KOOV Robot Recipe Point: 10	
Guitar (Basic) Where: From KOOV Robot Recipe Point: 5	

Train Race

Build:

Locomotives (KOOV robot recipe)

Follow a dark line (use IR Photo reflector to detect the distance between the train and the dark line on the paper

Maze

Build a train that can get out of the maze the quickest

G
a
m
e
s

Games

□ Human Robotic Relay

□ Robotics Charade

Games - Robotics Relay

Material:

A list of questions with answers,

Strips of these questions (with answers)

Small ziplock bags

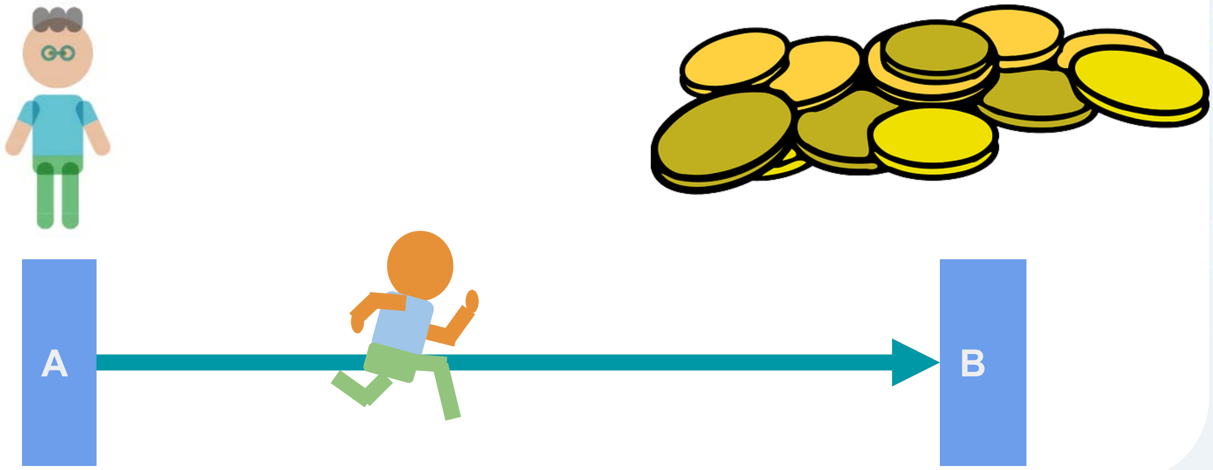
Plastic gold coin or paper printout

How to Play:

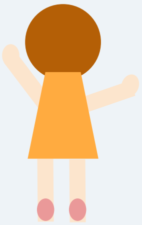
- Have kids in groups and line up at line A. From each group, send one runner at a time to run to B.
- When a runner arrives at line B, the mentor asks a random selected question. If this runner gets it right, they get a gold coin to put in their bag.
- Each runner runs back and passes the bag to the next runner.
- The game ends when the time is up.

How to Win:

The team with the most gold coins wins.



Games - Robotics Charade



Material:

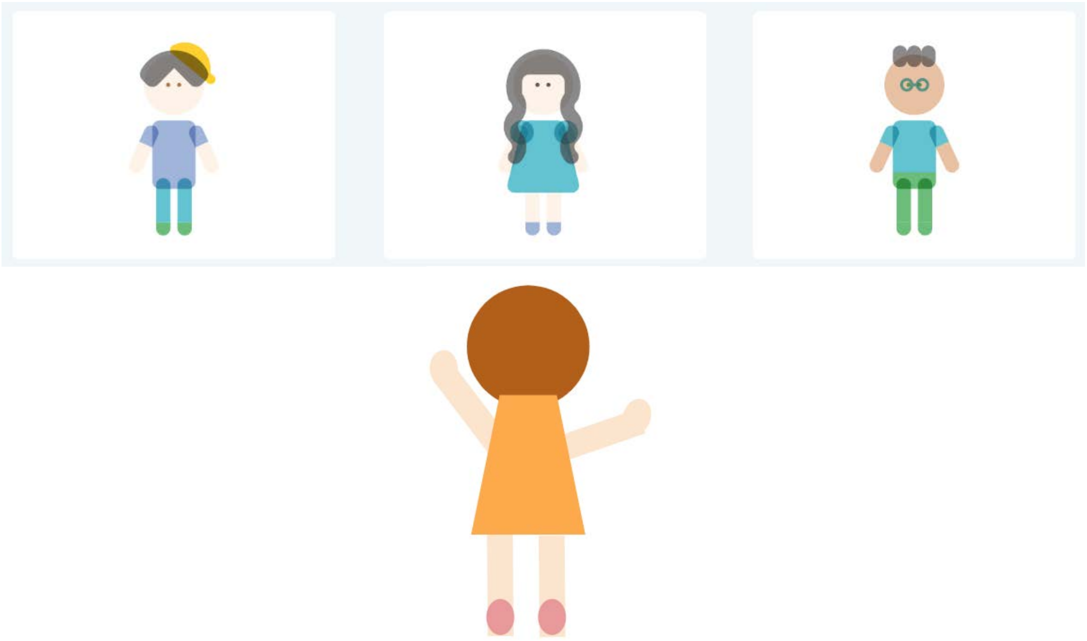
Timer, Word Cards

How to Play

- Game controller (the teacher or mentor) holds the word cards.
- Each team takes turns sending a player to act out the word, but all of the teams can guess. Teams take turns guessing, and each team has three guesses. The team who guesses right scores one point.
- Each word should take only 1 min, and if no teams guess the word when the minute is up, no one scores.
- The game is over when all word cards have been tried.

How to Win:

The team who has the most points wins.



Robotics Charade

Light Sensor

IR Photoreflector

DC Motor

Servo Motor

Buzzer

Red LED

Green LED

wait

Robot

Light

Sound

frequency

Push Switch

Blue LED

Top Core Button

Bottom Core Button

Right Core Button

Left Core Button

KOOV Core

Battery Pack

KOOV

appendix

Advanced

P

Physics

- Light
- Current & Voltage
- Closed Feedback Loop System

E

Electronics

Accelerometer

C

Coding Concepts

List

P h y s i c s

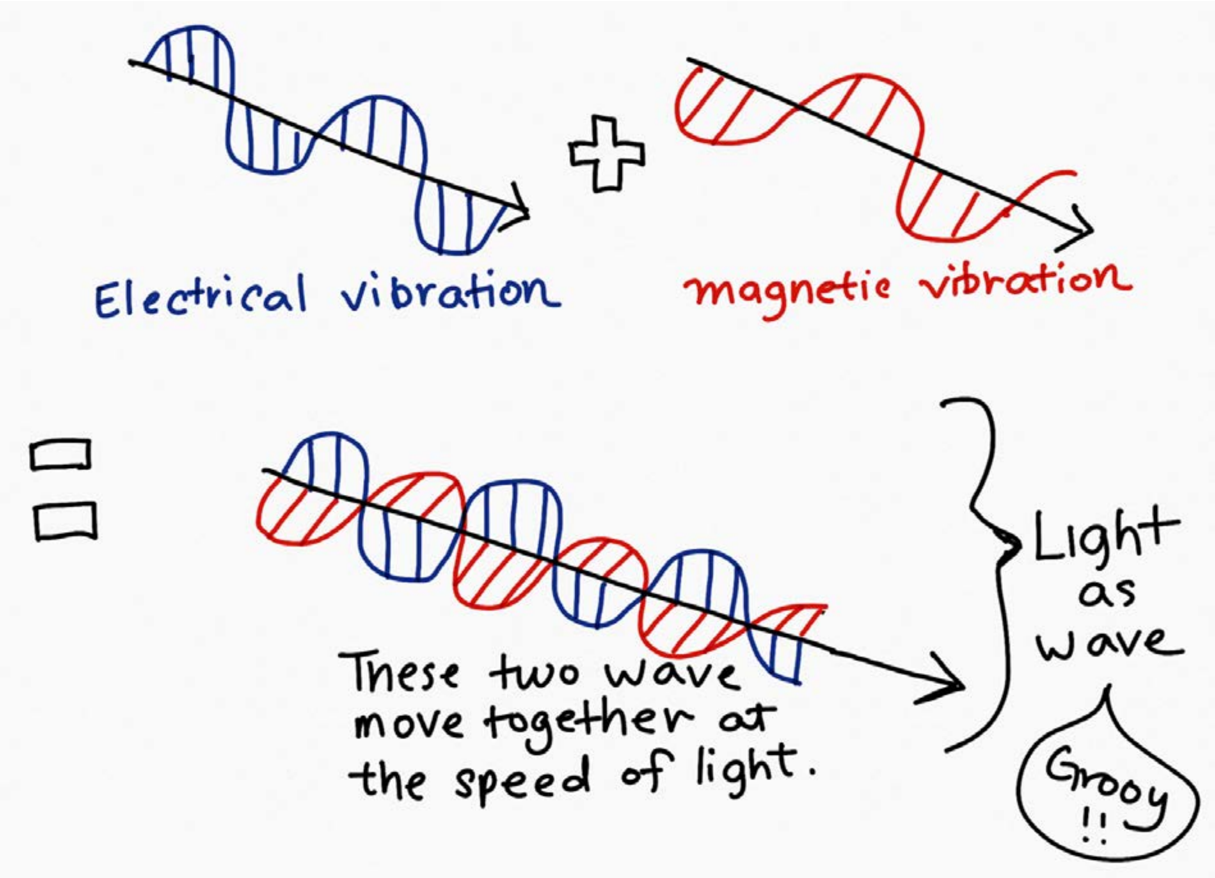
advanced

Light

KOOV

Light as a wave

Light is an **Electromagnetic Wave** or EM WAVE that is visible to human eyes. Below is an illustration of an electromagnetic wave, which includes magnetic vibration and electrical vibration, moving together at speed of light.



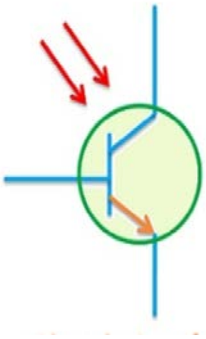
Light is just one type of EM wave, and there are other EM waves that are invisible to us. A chart called Electromagnetic Spectrum contains all EM waves, from the smallest frequency to the highest. Frequency is how many waves are there in a second.

P

How Does Light Sensor Work?

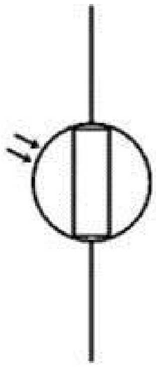
A **Light Sensor** is a device that detect brightness by converting light energy into electrical signals. Light sensors are also known as **Photo Sensors** because the convert light energy (**photons**) into electricity (**electrons**). Light sensors can be grouped into two main categories:

- those which generate electricity when illuminated
- those which change their electrical properties when illuminated



A **phototransistor** is a light sensor that generates electricity when illuminated. The “photo” in phototransistor is light, and the “transistor” means an amplifier, which amplifies or increases something.

The brighter the light, the more electricity is generated.



Photoresistor is a light sensor that changes its electrical properties when illuminated. A photoresistor does not generate electricity when there is light, but it reduces the resistance of electricity flow. Resistance is how hard it is for electricity to flow through an object. When resistance is reduced, electricity flows faster.

The brighter the light, the more electricity is allowed to flow through.

Physics

advanced

Current and Voltage

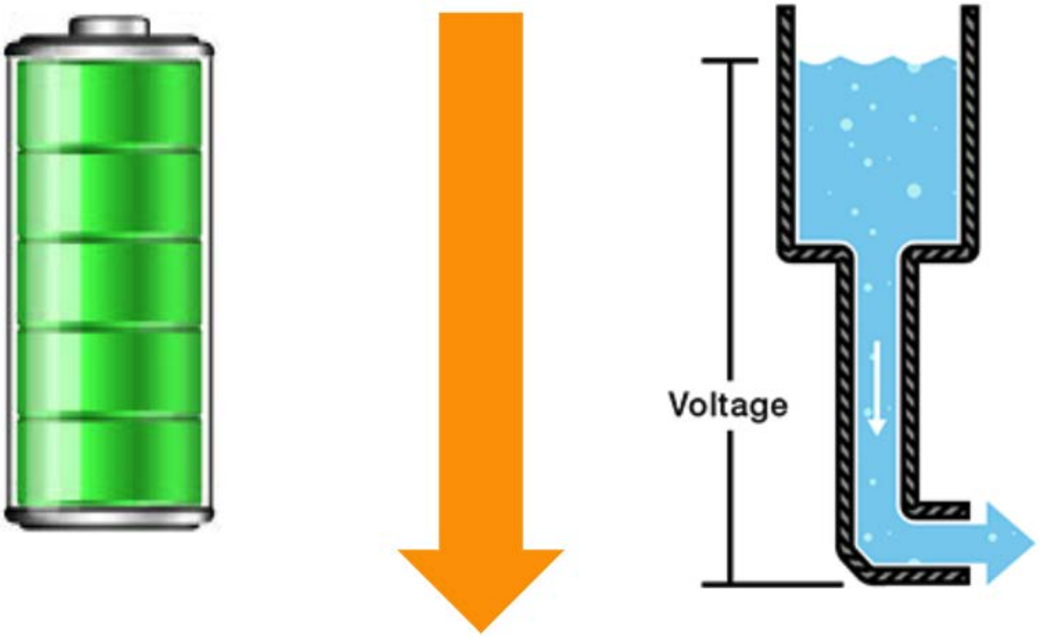
KOOV

DC, Current, and Voltage

In **direct current (DC)**, the electric charge (**current**) only flows in one direction. A battery generates direct current.

The pressure at the end of the hose can represent **voltage**. The water in the tank represents charge. The more water in the tank, the higher the charge, the more pressure is measured at the end of the hose.

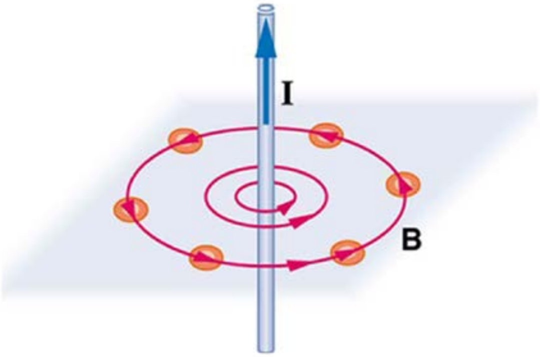
We can think of this tank as a battery, a place where we store a certain amount of energy and then release it. If we drain our tank a certain amount, the pressure created at the end of the hose goes down. We can think of this as decreasing voltage, like when a flashlight gets dimmer as the batteries run down. There is also a decrease in the amount of water that will flow through the hose. Less pressure (voltage) means less water is flowing, which brings us to current.



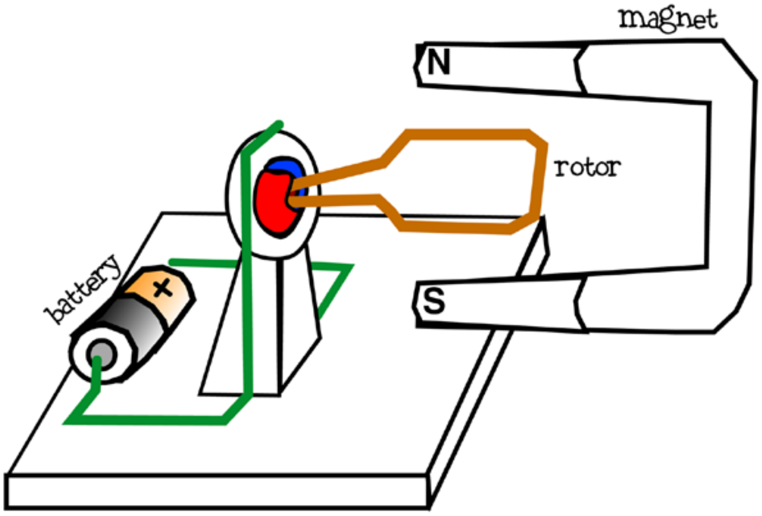
DC Motor rotates via

Magnetic Force

Electric current produces a **magnetic field.** This magnetic field can be visualized as a pattern of circular field lines surrounding a wire.



DC Motor works by running current through the rotor, generating magnetic field, which interacts with another magnet to keep the rotor turning. The higher the electric current, the stronger the magnetic field, and the stronger the interaction with that magnet inside the DC Motor, causing the rotor to spin faster.



P
h
y
s
i
c
s

advanced

Closed Feedback Loop System

P

Servo

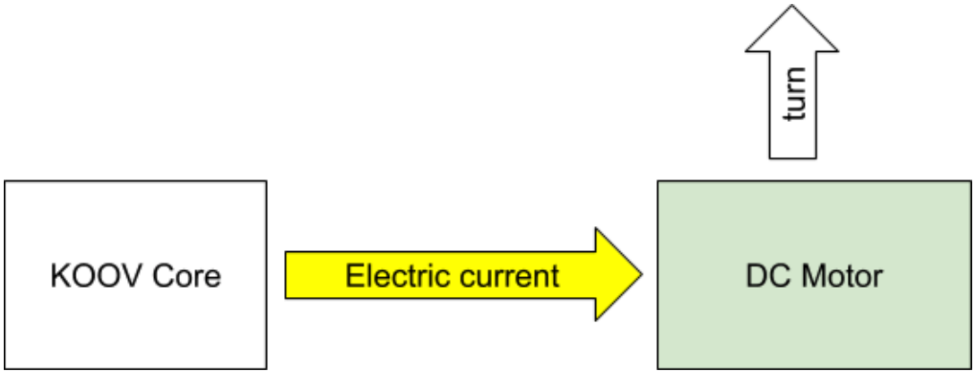
The Servo in Servo Motor means that it is a close control loop using feedback.

Servo is a general term for a closed loop control system using negative feedback.

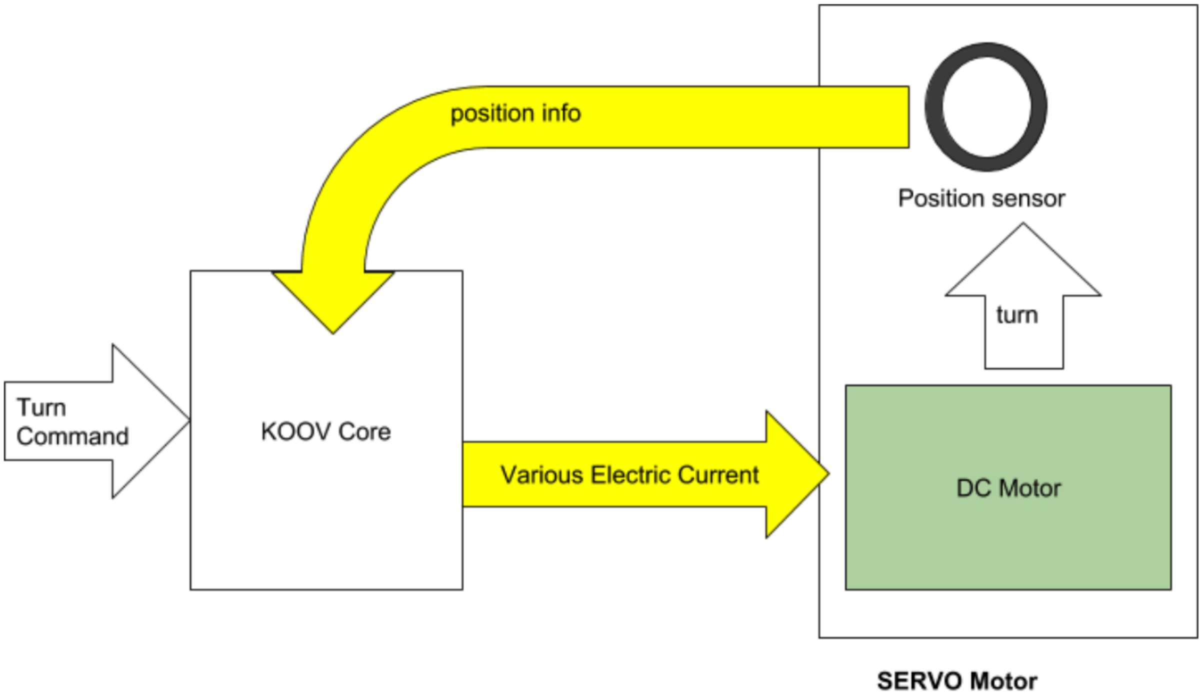
The cruise control in a car is one example of a servo system. It measures your speed and feeds that back into a control circuit which adjusts the car acceleration to maintain a stable speed.

Closed and Open Loop System

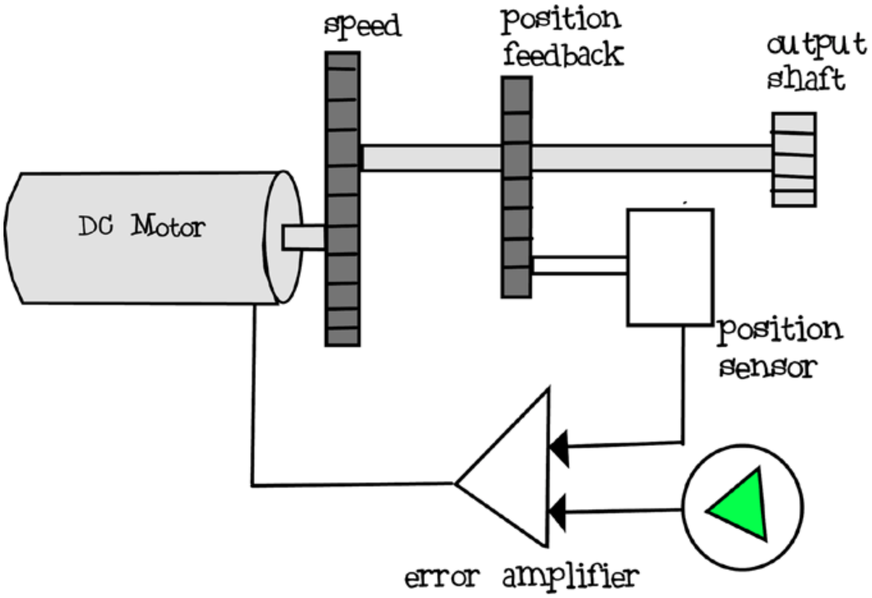
An "**Open** Loop" system has no feedback, so there is no way to verify that it is performing as expected. The **DC** Motor is in a Open Loop system, meaning that the current position is not fed back to the loop.



A "**Closed** Loop" system uses a feedback signal to adjust the speed and direction of the motor to achieve the desired result. The **Servo** Motor is a closed loop system that uses the current position to adjust the rotation speed.



Look Inside a Servo Motor



A Servo Motor contains a DC Motor, gears, a position sensor, and an error amplifier. Together it forms a control loop that keeps going until the desired location is reached.

The DC Motor rotates the shaft. The gears decreases the rotation speed of the shaft for accurate positioning. The output shaft also drives another gear which rotates the position sensor. When rotated, the position sensor provides a voltage indicating the shaft output position. That voltage is fed back into an error amplifier.The error amplifier produces an error signal which is the difference between the desired position and the current position.
The error signal is then used to drive the motor. When the current position equals the desired position, then the error signal would be zero, and the DC Motor would stop.

Electronics

advanced

Accelerometer

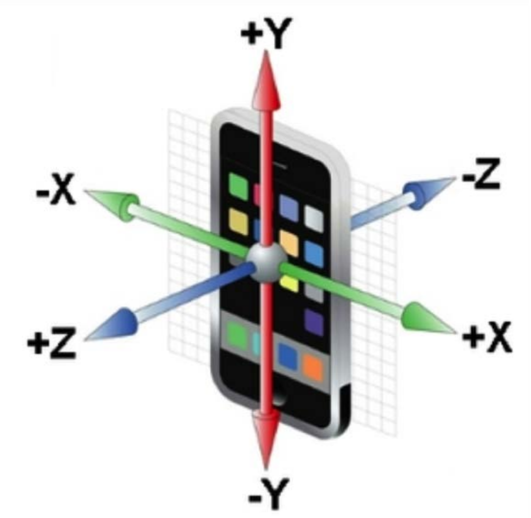
KOOV

How Does Tilting Works?

Have you ever play a game on a phone or a tablet that controls the game object by tilting? For example, you may have played a car racing game and “drives” the car by tilting your phone. How does your phone know how much you tilt or which directing you tilt? It knows by using a electronic device called an **accelerometer**.

Accelerometer

The Accelerometers are commonly used in game controllers and smartphones to use natural intuitive motion, such as motion similar to driving a car with steering wheel.

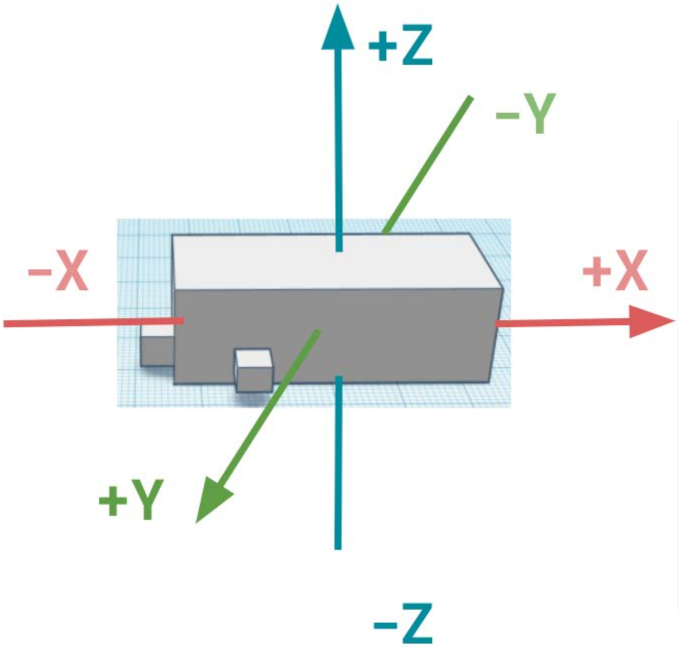


The Accelerometer tells the two-dimensional or three-dimensional positions by measuring the acceleration force along the direction of the axis. The acceleration force could be caused by gravity or movement.

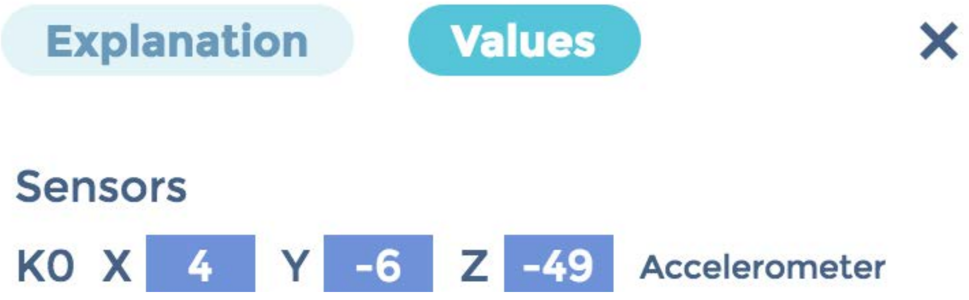
KOOV Accelerometer

video

The KOOV Accelerometer three-dimensional positions by measuring the acceleration force along the direction of the axis. You can learn how to use KOOV Accelerometer in the Firetruck project (found in the KOOV Robot Recipes) to control the ladder.



You can tilt a KOOV Accelerometer and watch its values change in KOOV.



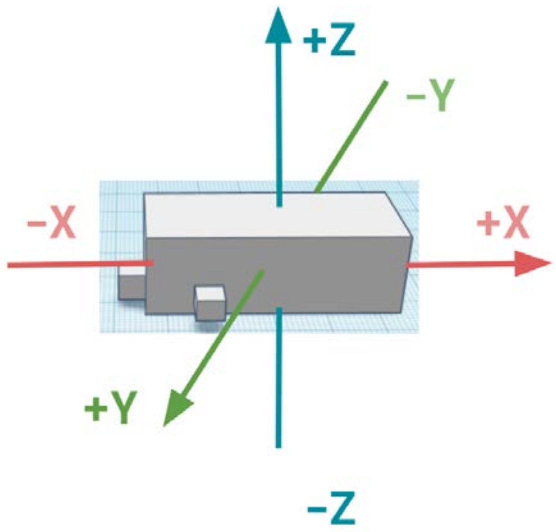
Explanation Values ×

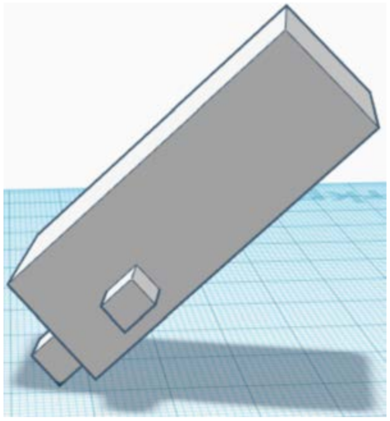
Sensors

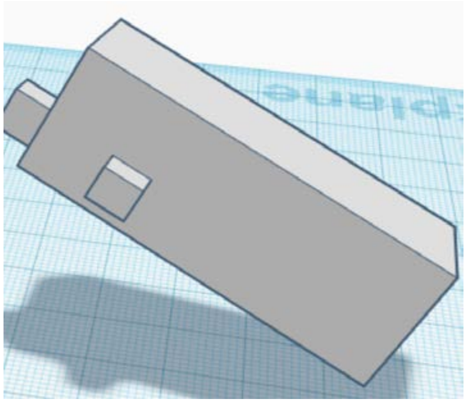
K0 X 4 Y -6 Z -49 Accelerometer

Rotate around X-axis

start







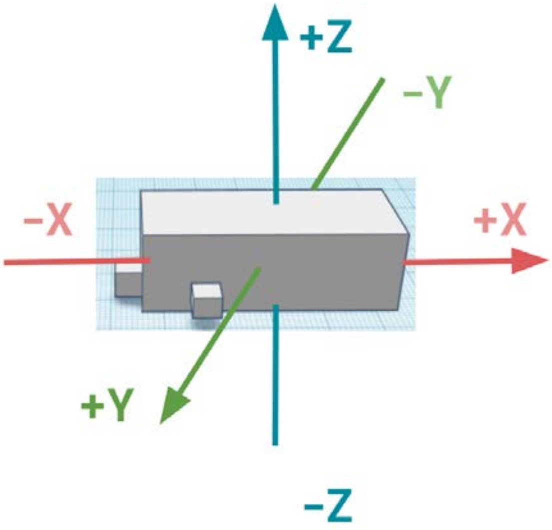
To decrease in X, tilt toward left

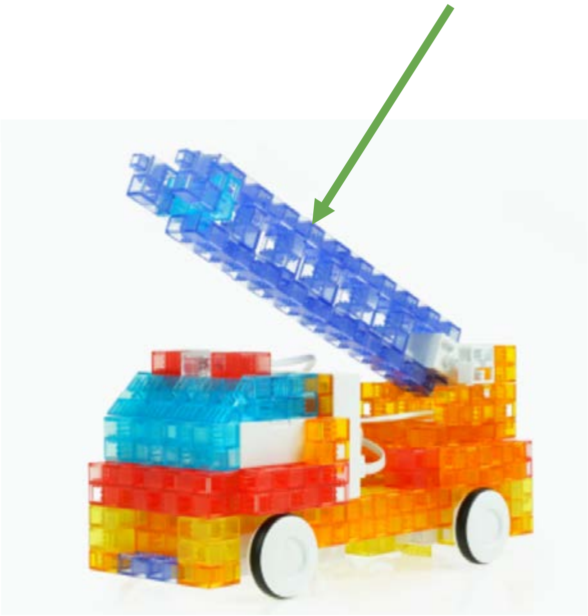
To increase in X, tilt toward right

E

Rotate around Y-axis

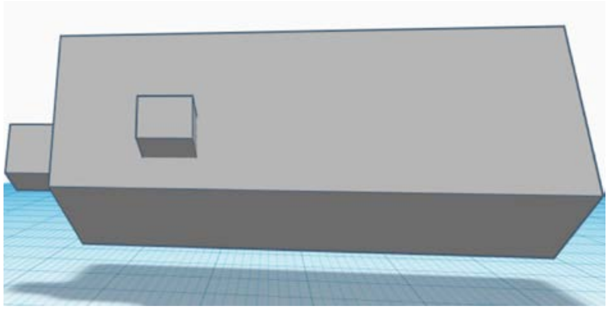
start

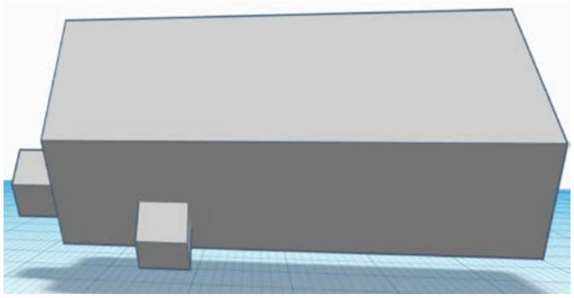




Tilt away from you to move ladder forward (+Y)

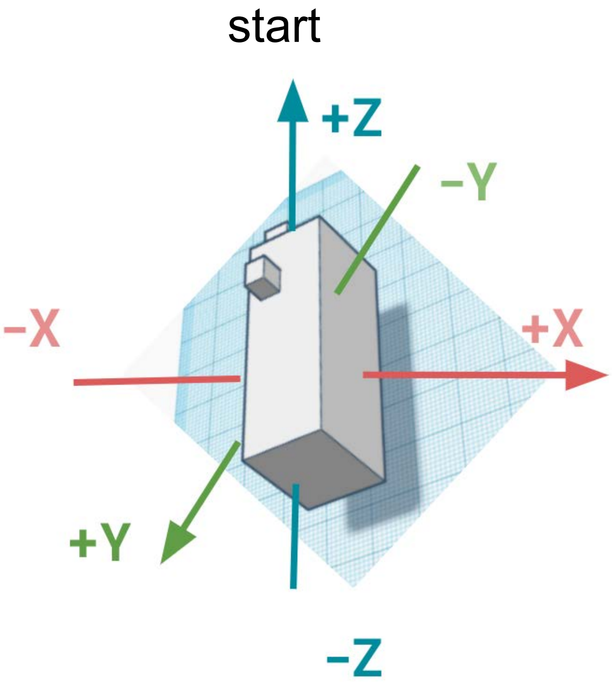
Tilt toward you to move increase in Y:





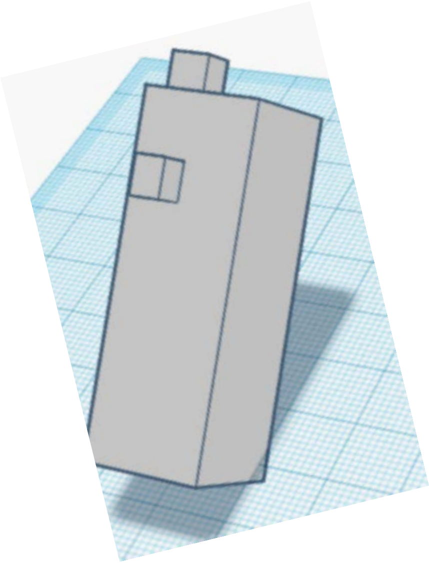
E

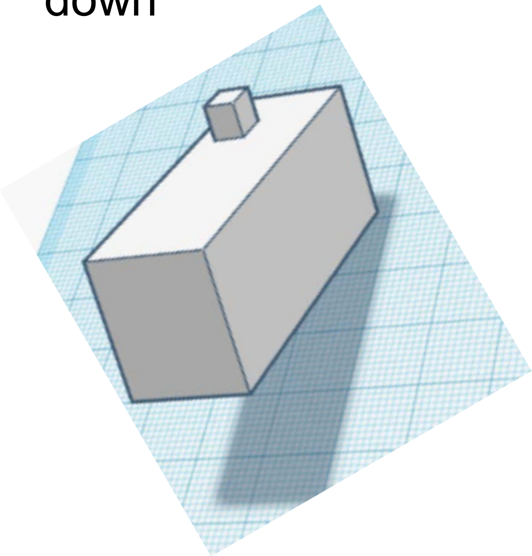
Rotate around Z-axis



Lift up the controller (+Z) to lift up the ladder

Press down the controller (-Z) to move the ladder down





coding

advanced

List

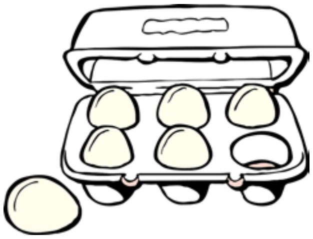
Lists

We use lists in our everyday life. If you are baking cupcakes, and you need to go shopping. You would need to put together a grocery shopping list. An example grocery shopping list could include sugar, flour, eggs, milk.

sugar

flour

eggs



milk

List name:
My Shopping List

List size: 4

List items:

1. sugar
2. flour
3. eggs
4. milk

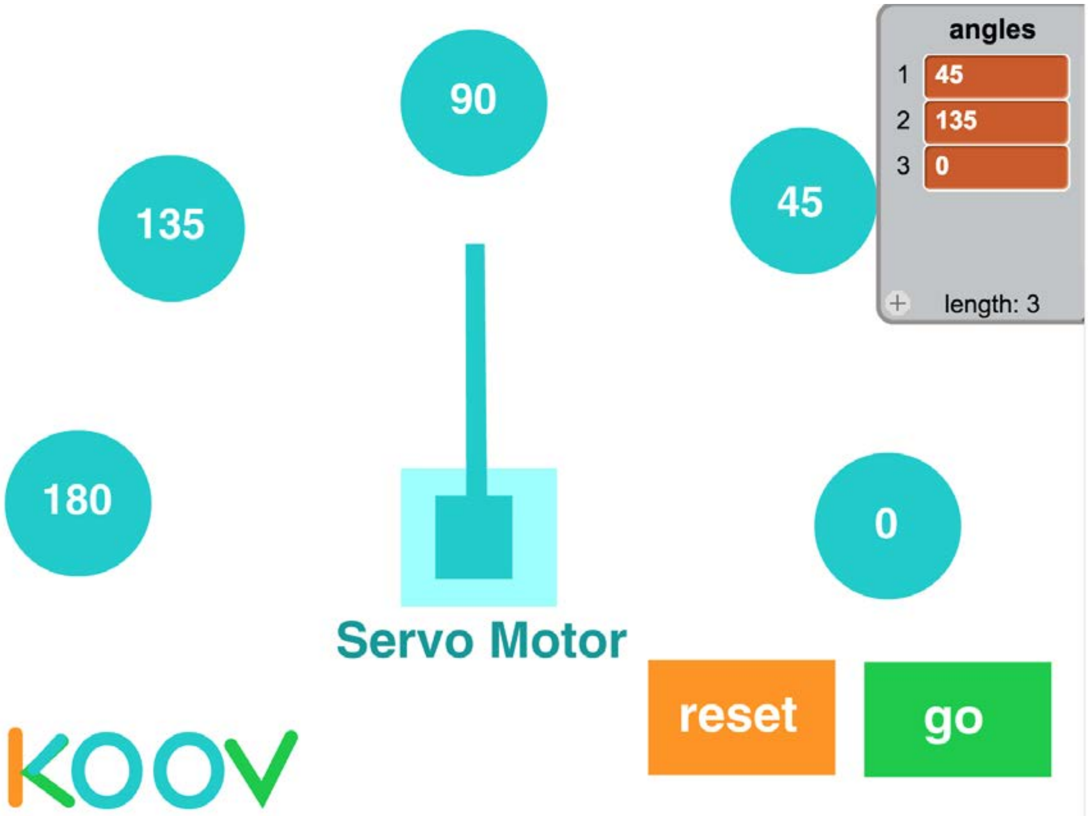
In coding, programs often store data in lists. A program can **add** something to a list, **read** something from a list, or **remove** something from a list.

Using Lists in code

demonstration

A **list** is a set of elements. In KOOV, an element can be either a numeric value or a variable.

In code, we can store data in a list and use it to perform task. For example, below is a simulator that lets you click on desired stops in a list, and when you click "go", the Servo Motor would turn to each position in the *angles* list.



Troubleshooting

We Can Fix It

Troubleshooting

E

Problem: Motor not moving

Reason 1: Battery too low
To confirm, use the Digital Multimeter or test with buzzer. If buzzer still works with the battery, then it's not the battery.

Video

Reason 2: The motor is jammed
To confirm, try a different motor
If another motor works with the same code, the teacher can open the DC motor with a screwdriver and adjust the gears.

Problem: Can move to the next step in Map

Reason 1: Code does not match
To confirm, check the tutorial.
To fix, update your code to be exactly like that in tutorial.

Reason 2: Extra variables not mentioned in tutorial
To confirm, check the data tab for unused variables.
To fix, remove extra variables.

Reason 3: Extra functions not mentioned in tutorial
To confirm, check the control tab for unused functions.
To fix, remove extra functions.

Reason 4: Extra lists not mentioned in tutorial
To confirm, check the data tab for unused lists.
To fix, remove extra lists.

Problem: Cannot see the codes

Reason 1: You accidentally delete it
To confirm, use the undo button.
To fix, undo.
Reason 2: It's hidden
To confirm, zoom out.
To fix, zoom out.

Troubleshooting

E

Problem: LED does not turn off

Reason 1: Did not set LED to off
To fix, make sure there is a LED () off line

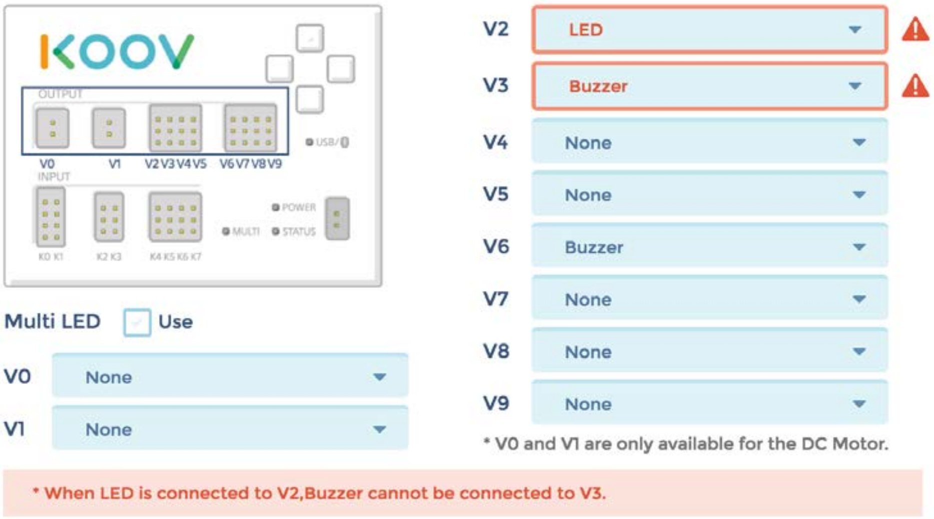
Reason 2: Wrong port
Check Connection Settings and your KOOV Core to make sure the LED ports match

Reason 3: Current too high
To fix, move to a different port.

Buzzer Connection Error

video*

Moreover, if there are other devices already connected to a section (say V1-V5), then connect the buzzer to a different section. For instance, if an LED is already connected to V2, you cannot connect a buzzer to V3, because it's in the same section, and there is not enough current to support both devices.



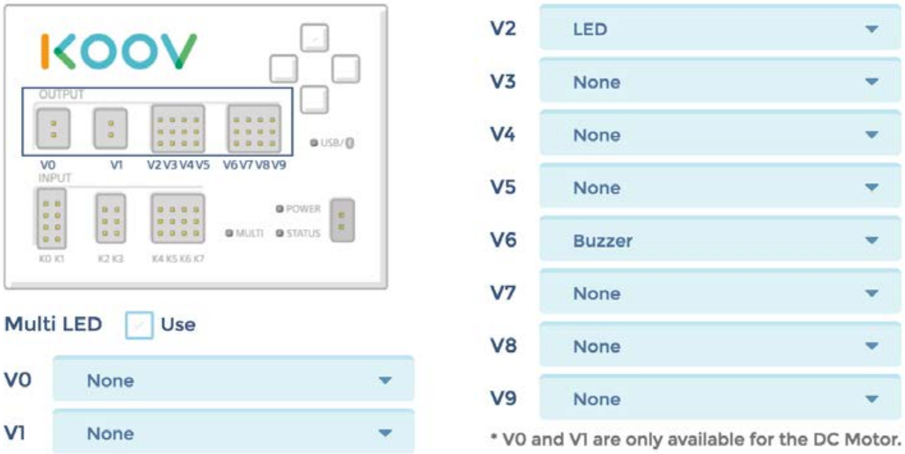
Multi LED □ Use

V0	None
V1	None
V2	LED ⚠
V3	Buzzer ⚠
V4	None
V5	None
V6	Buzzer
V7	None
V8	None
V9	None

* V0 and V1 are only available for the DC Motor.

* When LED is connected to V2,Buzzer cannot be connected to V3.

Instead, connect the Buzzer to a port in the other section, such as V6.



Multi LED □ Use

V0	None
V1	None
V2	LED
V3	None
V4	None
V5	None
V6	Buzzer
V7	None
V8	None
V9	None

* V0 and V1 are only available for the DC Motor.

KOOV

Robotics Camp

Fun, Learning, and Discovery

Student Manual

Day 5 Objectives

E Servo Motor

C list

L Penguin

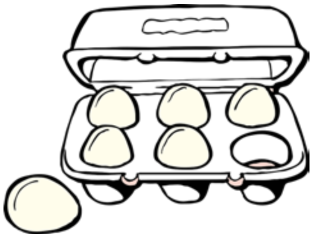
Day 4 Concept: list

We use lists in our everyday life. If you are baking cupcakes, and you need to go shopping. You would need to put together a grocery shopping list. An example grocery shopping list could include sugar, flour, eggs, milk.

sugar

flour

eggs



milk

List name: My Shopping List

List size: 4

List items:

1. sugar
2. flour
3. eggs
4. milk

What is the first item in the list?

____

What is the third item in the list?

____

In coding, programs often store data in lists. A program can **add** something to a list, **read** something from a list, or **remove** something from a list.

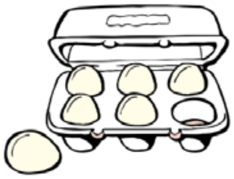
Day 4 Concept: working with list

Let's try practice working with list. Answer the question about this list after a program changes it. Note that when adding, an item is added to the end of a list.

sugar

flour

eggs



milk

```
start
remove item [3] from list
add [chocolate] to list
```

List name: My Shopping List

List size of the new list: ____________________

List Items in the new list:

1. ___________________________

1. ___________________________

1. ___________________________

1. ___________________________

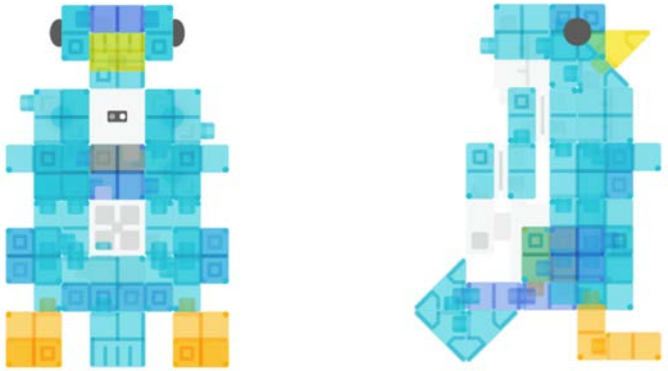
1. ___________________________

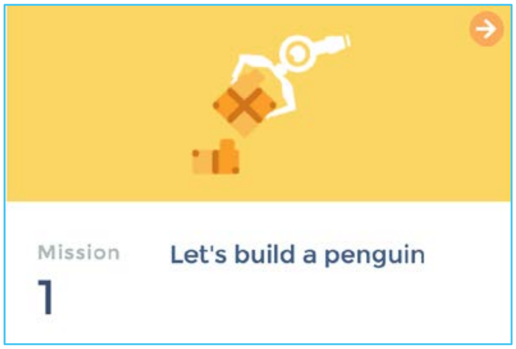
The 3rd item in the list: ___________________

Day 5 Map Stage 6 Mission 1: Let's build a penguin

This mission's goal

Build a penguin with Servo Motor, IR Photoreflector, and KOOV Blocks.



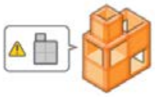


Mission 1

Let's build a penguin

Parts you need

Blue Half-C x 35

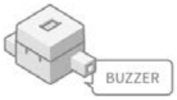


Orange Half-C x 7

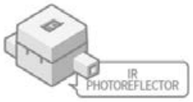
Purple Half-D x 8

Yellow Prism x 1

Blue Prism x 4



Buzzer x 1



IR Photoreflector x 1

Servo Motor x 1

360 Connector x 4

Eye x 2

Core x 1

Battery Box x 1

(with batteries)

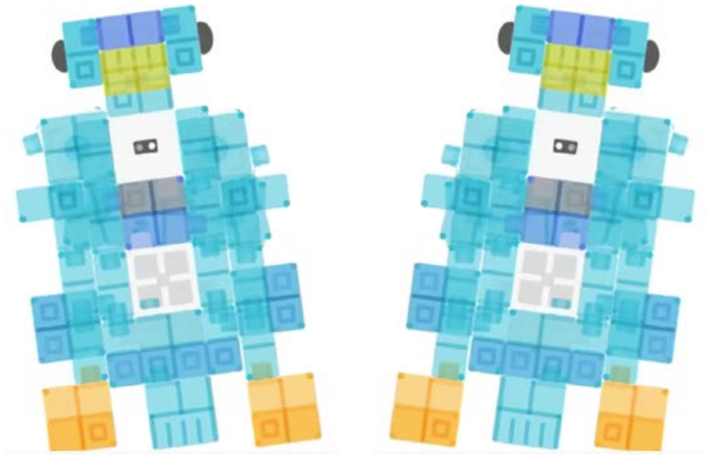
Base Tray x 2

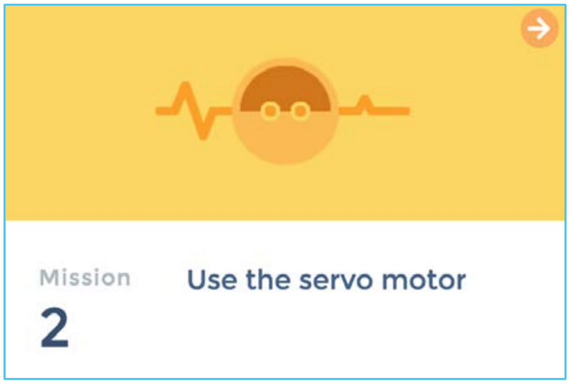
3-Core Cable (20cm) x 2

Day 5 Map Stage 6 Mission 2: Use the servo motor

This mission's goal

Use Servo Motor to make the penguin dance side to side.

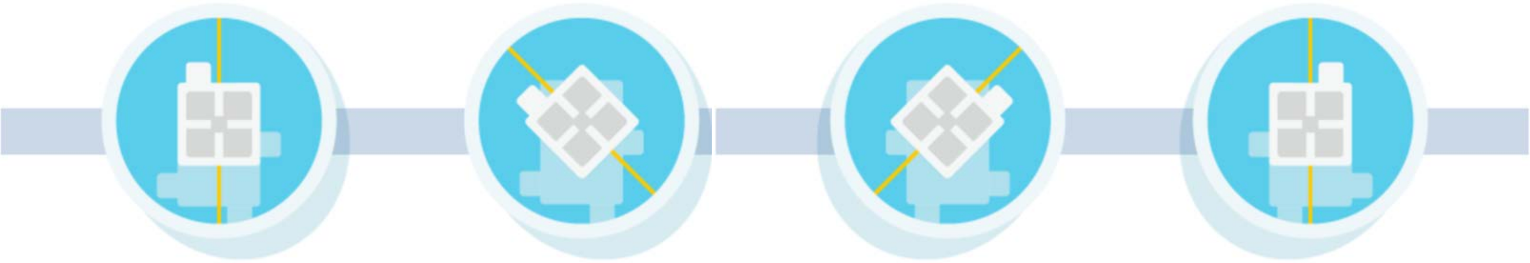




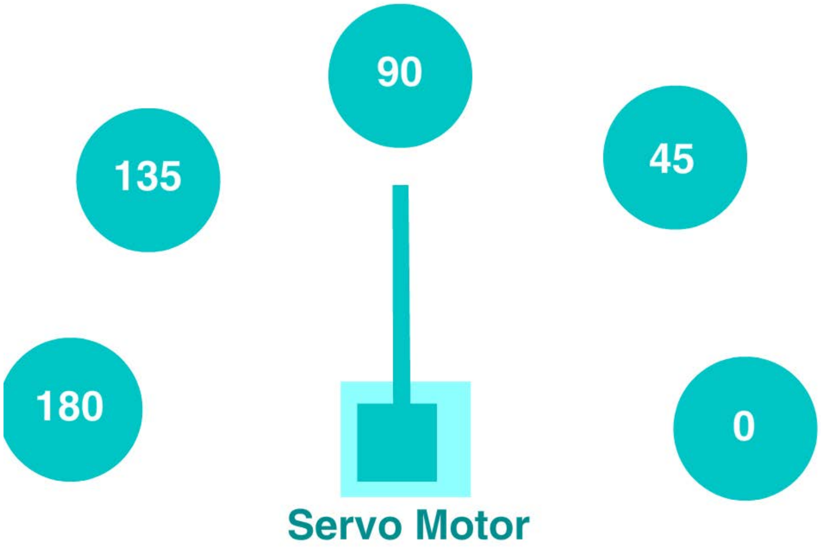
Mission

2

Use the servo motor



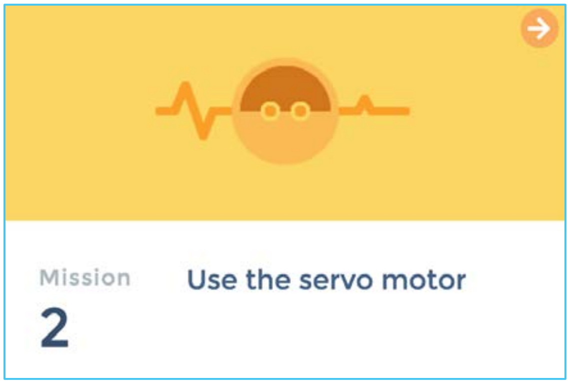
The Servo Motor points to 90 degree by default.



Day 5 Map Stage 6 Mission 2: Use the servo motor

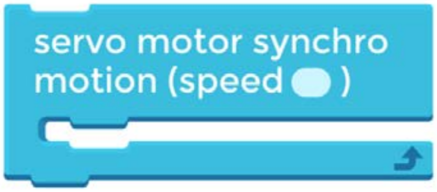
This mission's goal

Use Servo Motor to make the penguin dance side to side.

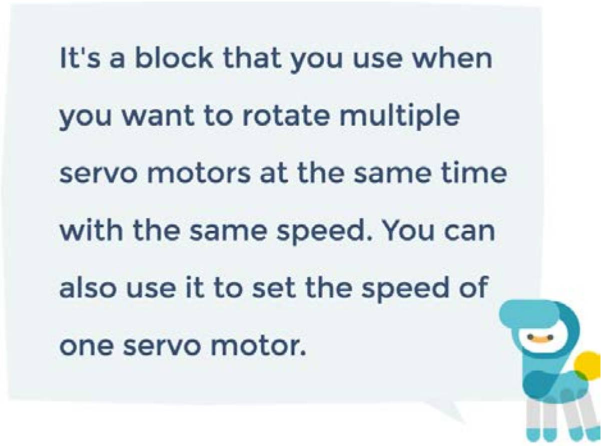


Mission 2

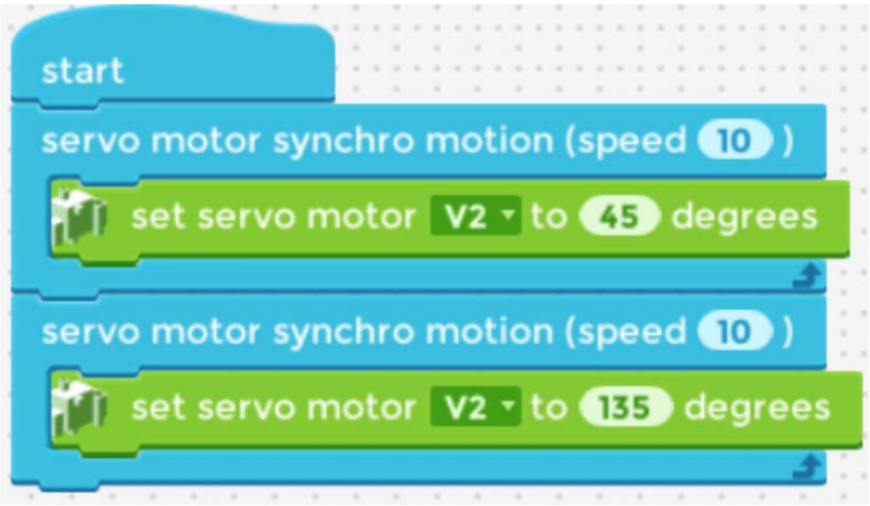
Use the servo motor



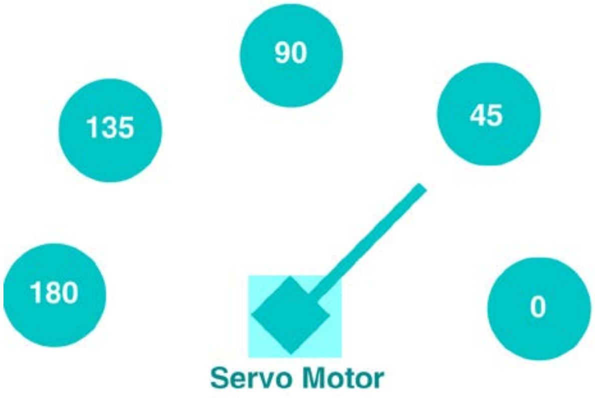
servo motor synchro motion (speed)

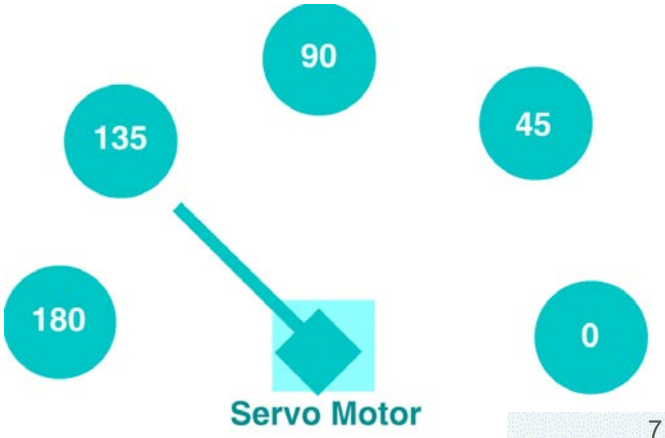


It's a block that you use when you want to rotate multiple servo motors at the same time with the same speed. You can also use it to set the speed of one servo motor.



```
start
servo motor synchro motion (speed 10 )
    set servo motor V2 to 45 degrees
servo motor synchro motion (speed 10 )
    set servo motor V2 to 135 degrees
```

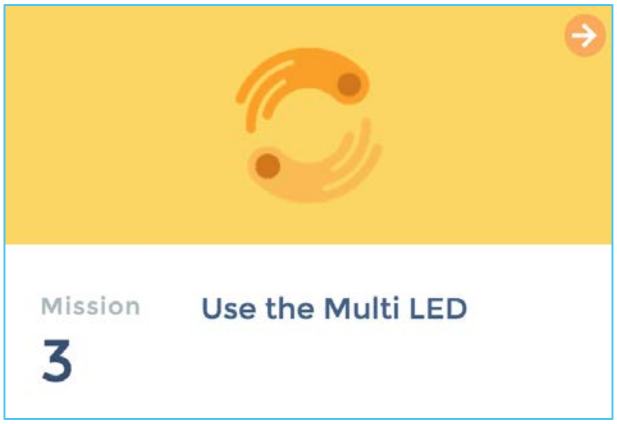




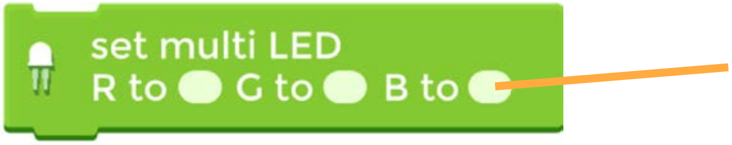
Day 5 Map Stage 6 Mission 3: Use the Multi LED

This mission's goal

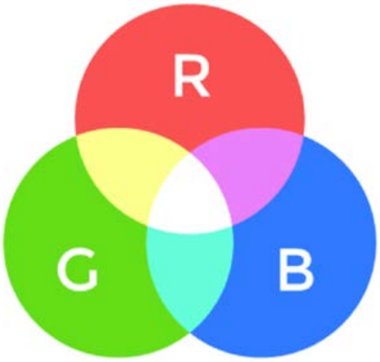
Learn to use the Multi LED, and how to make a color with light.


Mission
3
Use the Multi LED

Red, **G**reen and **B**lue are the **primary colors** of light. Mixing the same amount of these colors make another color:

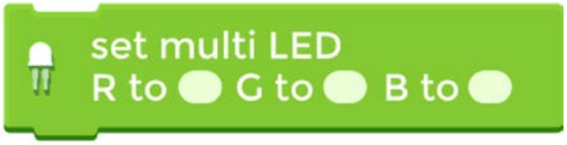

set multi LED
R to G to B to

value from
0 to 100

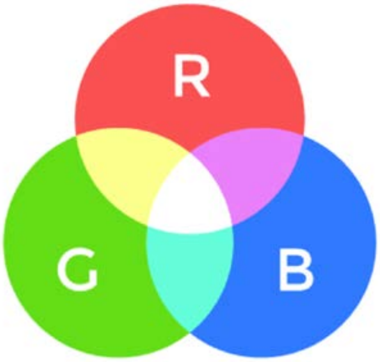


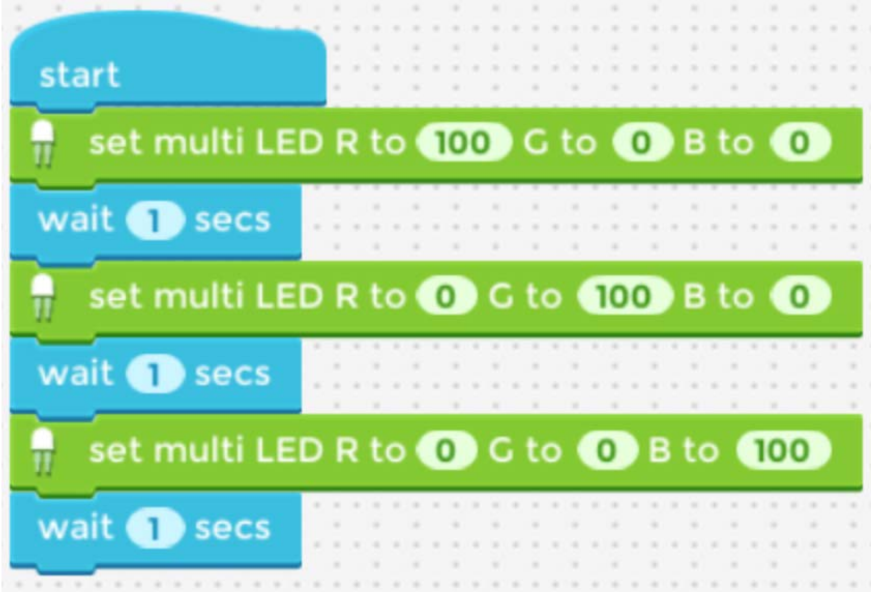
Day 5 Map Stage 6 Mission 3: Use the Multi LED

Red	Green	Blue	Result
100	0	0	Red
0	100	0	Green
0	0	100	Blue
100	100	0	Yello
0	100	100	Cyan
100	0	100	Magenta
100	100	100	White



set multi LED
R to ⬭ G to ⬭ B to ⬭





```
start
set multi LED R to 100 G to 0 B to 0
wait 1 secs
set multi LED R to 0 G to 100 B to 0
wait 1 secs
set multi LED R to 0 G to 0 B to 100
wait 1 secs
```

What color pattern does the Multi LED light up with the program above?

Day 5 Lab: Stage 6 Mission 4 Use random numbers

1) What does this code do? What possible numbers would this code block return?

```
pick a random 1 to 10
```

Answer: ____

1) What would the variable penguin be?

```
set variable penguin to ( 10 ) - ( 2 )
```

Answer: ____

1) What is the value variable big_penguin after running the following code?

```
set variable penguin to ( 10 ) - ( 2 )
set variable big_penguin to ( penguin ) - ( 2 )
```

Answer: ____

1) What does this code do?

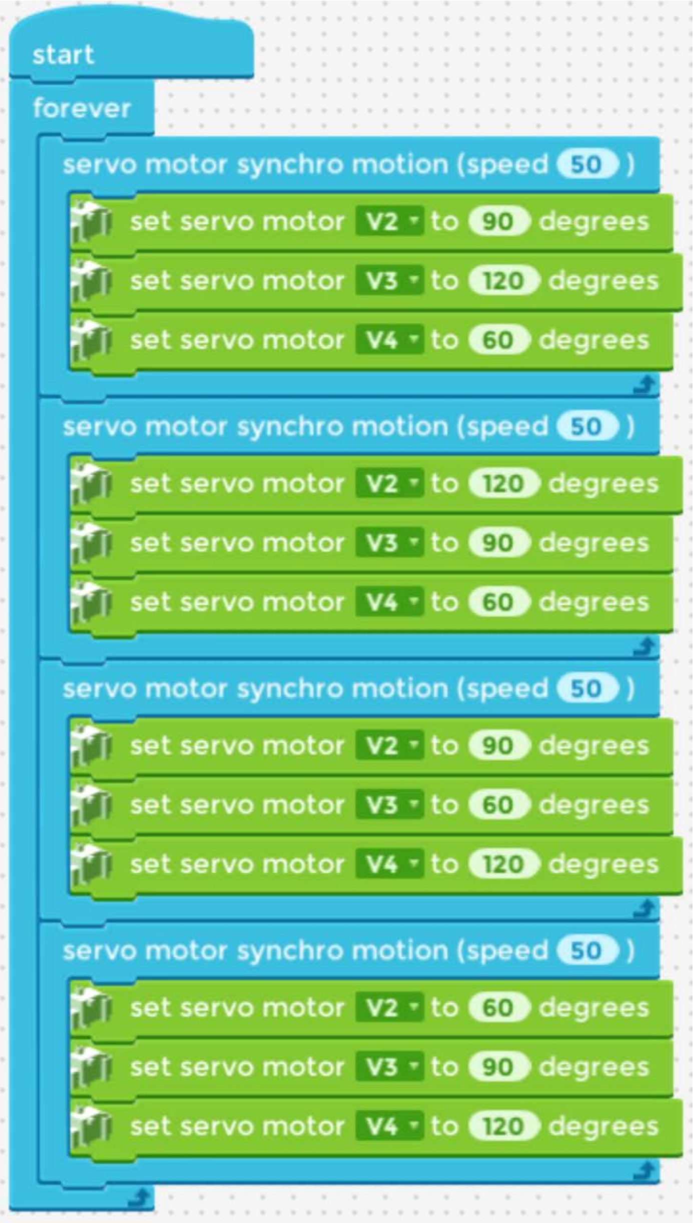
```
servo motor synchro motion (speed (50))
set servo motor (V2) to (90) degrees
set servo motor (V3) to (120) degrees
set servo motor (V4) to (60) degrees
```

a) Jump
b) Dance
c) Sing

Answer: ____

Day 5 Lab: Buddy from Robot Recipe

Build Buddy following the Robot Recipe. You will use three Servo Motors to make its legs turn. Build and load the code to Buddy and test.



```
start
forever
    servo motor synchro motion (speed 50)
        set servo motor V2 to 90 degrees
        set servo motor V3 to 120 degrees
        set servo motor V4 to 60 degrees
    servo motor synchro motion (speed 50)
        set servo motor V2 to 120 degrees
        set servo motor V3 to 90 degrees
        set servo motor V4 to 60 degrees
    servo motor synchro motion (speed 50)
        set servo motor V2 to 90 degrees
        set servo motor V3 to 60 degrees
        set servo motor V4 to 120 degrees
    servo motor synchro motion (speed 50)
        set servo motor V2 to 60 degrees
        set servo motor V3 to 90 degrees
        set servo motor V4 to 120 degrees
```

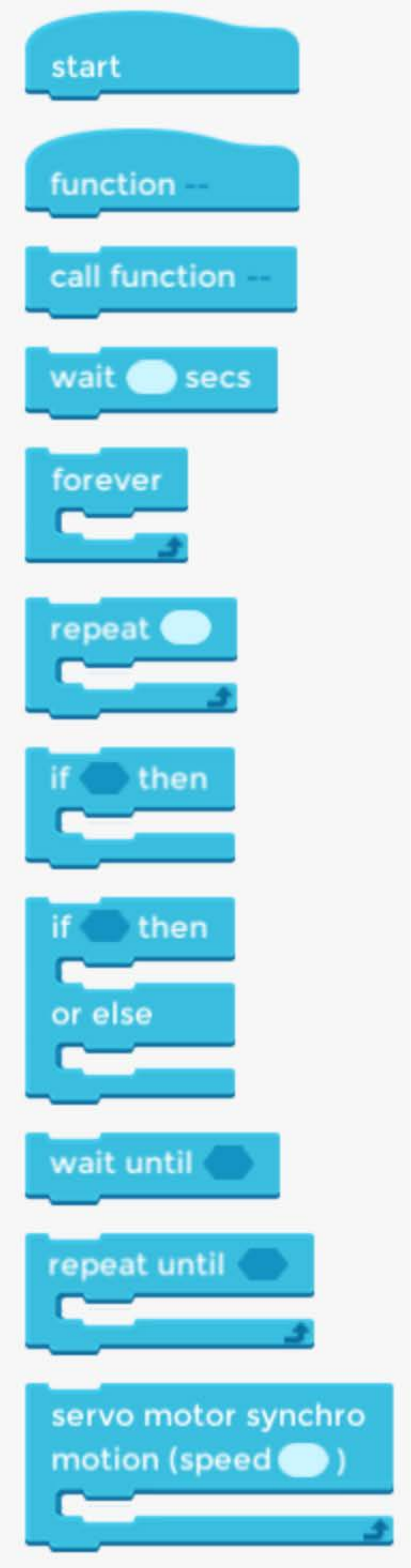
What does this code do?

```
Servo motor synchro motion (speed (50))
Set servo motor (V2) to (90) degrees
Set servo motor (V3) to (120) degrees
Set servo motor (V4) to (60) degrees
```

Can you make it dance faster?

Code Blocks Used

Control



start

function --

call function --

wait secs

forever

repeat

if then

if then

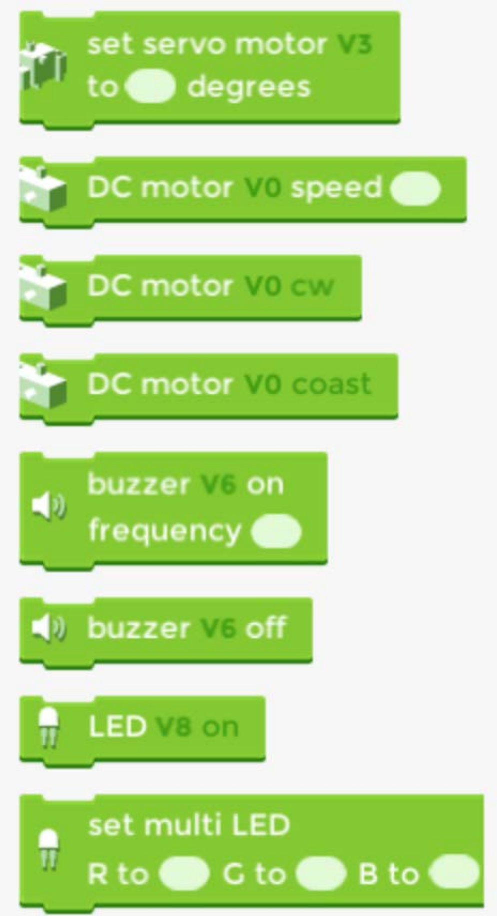
or else

wait until

repeat until

servo motor synchro motion (speed)

Motion



set servo motor V3 to degrees

DC motor V0 speed

DC motor V0 cw

DC motor V0 coast

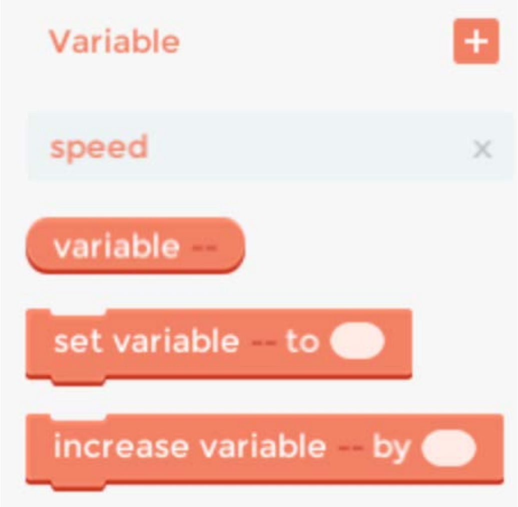
buzzer V6 on frequency

buzzer V6 off

LED V8 on

set multi LED R to G to B to

Variable



Variable +

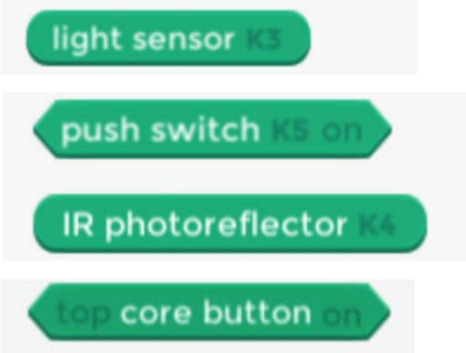
speed ×

variable --

set variable -- to

increase variable -- by

Sensing



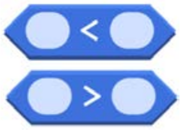
light sensor K3

push switch K5 on

IR photoreflector K4

top core button on

Operator



<

>

Value Range

Sound Sensor	0 to 100
Light Sensor	0 to 100
IR Photoreflector	0 to 100
DC Motor Speed	0 to 100
Servo Motor Speed	0 to 100
Servo Motor Degrees	0 to 180
Buzzer Frequency	48 to 108
Accelerometer	-100 to 100

Troubleshooting page 1

Problem: Motor not moving

Reason 1: Battery too low
Ask the adults to check or replace with new batteries.

Reason 2: The motor is jammed
After replacing the battery, if the motor still does not work, then ask the adults for help.

Problem: Can move to the next step in Map

Reason 1: Code does not match
To confirm, check the tutorial.
To fix, update your code to be exactly like the tutorial.

Reason 2: Extra variables not mentioned in tutorial
To confirm, check the data tab for unused variables.
To fix, remove extra variables.

Reason 3: Extra functions not mentioned in tutorial
To confirm, check the control tab for unused functions.
To fix, remove extra functions.

Reason 4: Extra lists not mentioned in tutorial
To confirm, check the data tab for unused lists.
To fix, remove extra lists.

Problem: Cannot see the codes

Reason 1: You accidentally deleted it
To confirm, use the undo button.
To fix, undo.

Reason 2: It's hidden
To confirm, zoom out.
To fix, zoom out.

Troubleshooting page 2

Problem: LED does not turn off

Reason 1: Did not set LED to off
To fix, make sure there is a LED () off line

Reason 2: Wrong port
Check Connection Settings and your KOOV Core to make sure the LED ports match

Reason 3: Current too high
To fix, move to a different port.