Human Interface Guide

Designed and made by New Zealanders (also known as Kiwis)

for their ❤ of teaching children

computer science, coding & robotics.

Thank you for purchasing KaiBot. To get the most benefit from KaiBot please be sure to read all instructions thoroughly and share this document with others who will also use KaiBot.

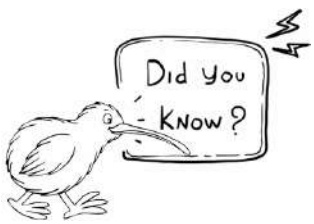
Disclaimer

Autonomous Works Limited has made every effort to provide clear and accurate information in this Manual, which is provided solely for the user's information. While thought to be accurate, the information in this document is provided strictly "as is" and Autonomous Works Limited will not be held responsible for issues arising from typographical errors or user's interpretation of the language used herein that is different from that intended by Autonomous Works Limited. All safety and general information are subject to change due to changes in applicable laws. Autonomous Works Limited reserves the right to revise this Manual and make changes from time to time in the content hereof without obligation to notify any person of such revisions or changes. In no event shall Autonomous Works Limited, its employees, or authorized agents be liable to you for any damages or losses, direct or indirect, arising from the misuse of any technical or operational information contained in this document

Symbols and Conventions

📖 This icon marks reference to other pages in this manual.
Menu items, options, and messages are displayed in bold.

Notices



The kiwi is not only a fruit.

There are three different meanings of Kiwi. A person who lives in New Zealand, a Kiwi bird is a flightless bird that is also used as the national symbol, and, lastly, there is the delicious fruit called Kiwi fruit.

Index

Let’s do this!

Everything you need is inside the box. Well, nearly everything...

If you are going to pair KaiBot with your Chromebook, PC or MAC then that device is not inside the box, lol that would be funny if your computer was also in the box!

The great thing about KaiBot is that it can work without your computer (screen-free) or it can work alongside it using Kainundrum.com.

1. Help KaiBot feel at home.

KaiBot works best on clean, flat surfaces with or without his optional Kai-Tiles. He likes open spaces either on the floor or on a table. But he prefers his magnetic blue Kai-Tiles because he always knows where he is when he's placed on them. He’ll work fine without the blue magnetic tiles though.

2. Power up.

Plugin the USB charge cable to the back of KaiBot to charge him up.
He feels more secure if you have the optional Charge Dock as then he’ll be able to charge himself when he gets tired and be refreshed and ready for the day ahead.
Do not use the robot during charging. KaiBot gets around 1-2 hours of play from a 30-minute charge when connected to a powered USB outlet.

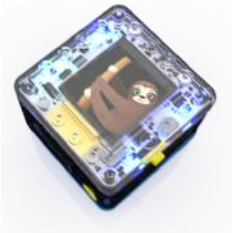
3. Layout the Coding Cards

Once you’ve got KaiBot charged, it’s time to teach him how to code and be ready for World Domination. (don’t forget to pack lunch).

KaiBot acts as a computer interpreter. KaiBot interprets and translates the coding cards into machine code, instruction by instruction, and then KaiBots CPU executes each instruction before the internal interpreter moves on to translate/scan the next instruction. If KaiBot scans a coding card that is invalid the interpreted code will show an error Bug message on the screen as soon as it encounters a problem, so this makes it easier to debug your code.

What’s included

What’s in the box	

KaiBot rechargeable robot	
USB to Micro USB charging cable	 *Optional autonomous charging dock available
Deck of 33 Standard Coding Cards that KaiBot can scan & read	 *Optional Advanced Coding Cards available
Various KaiBot character sleeves, to customize your robot	
Activities & lesson plans	https://kaiseducation.com/lesson-plans/

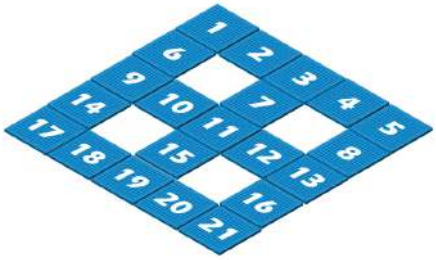
Optional add-ons

KaiBots Autonomous Charging Dock, KaiBot becomes a mini-Roomba like a robot and can self-charge!

When KaiBot is placed on the purple tile, KaiBot will automatically navigate to the charge dock.

If you also have KaiTiles along with Kainundrum, then KaiBot will autonomously drive to the nearest available charge dock and fill up.

Magnetic Kai-Tiles, to create physical maze layouts. Each tile is a unique number from 1 to 60. Kai-Tile packs are available from 1-60 and available in packs of 10.
KaiBot can be used without tiles, however with tiles, KaiBot can automatically correct the direction he is travelling in if he starts going off course and also read the x,y position and the tile number he is on.



Advanced Coding Cards, is a deck of over 100 advanced coding cards that KaiBot can read.

With the Advanced deck you'll get:
Repeat until condition
Color mixing with Red, Green & Blue
Function 1 & 2 sets to create nested programs
Comparison Operators like greater, less and equal to.
Conditional Statements like, IF, THEN, ELSE etc.
Variables
Plus extra copies of the standard cards.

KaiBot in the Classroom

Picture: Using KaiBot with KaiTile and coding cards in the classroom.

Where do I start as a Teacher?

It’s movie time!

https://youtu.be/YF-6fguar0M

Introduction Activity

Have pairs of students act out the instructions based on the cards laid out in front of them. This activity will help students understand the sequencing required for KaiBot to undertake his tasks.

Engineering Connection

The engineering team at Kais Education who created KaiBot had many problems to solve in creating him. In this activity, students act as if they are engineers designing programs to solve problems. Through this lesson, students will build a basic understanding of robotics by thinking creatively about how to solve problems and see how the Kais Education team faced them.

Learning Objectives

After this activity, students will be able to:

- Define what a robot is.
- Describe the main components of a robot.
- Explain how KaiBot can be programmed to move.
- Explain how troubleshooting and debugging are an important part of engineering.

Introduction

In this activity students learn how to think like a robot in order to understand how to code and debug them.

Have one student act as the computer - giving out step-by-step instructions and the other student.

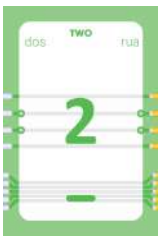
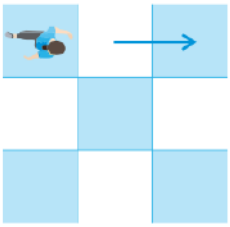
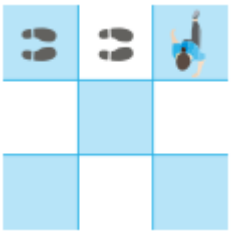
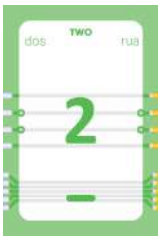
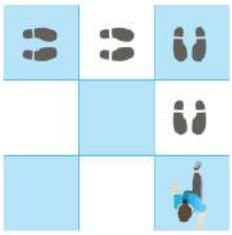
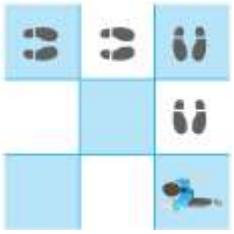
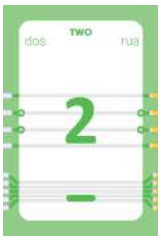
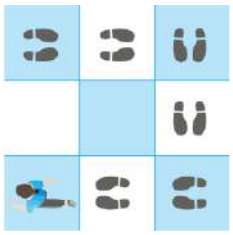
The other student acts as the robot, records the entire sequence in their memory and then walks out of the sequences.

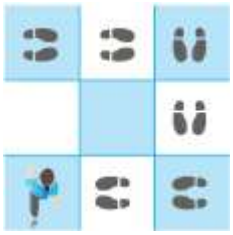
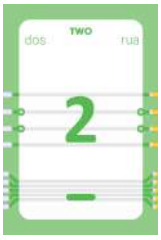
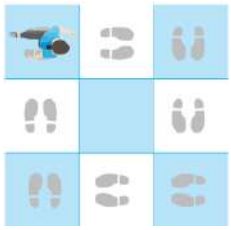
Level 1: Concrete to Abstract

1. Depending on the age, I suggest we act out what a human robot will do.
2. Create a physical pathway in the classroom. You have to walk to make a square. Example: Take 2 steps forward, turn right. Take 2 steps forward and turn right. Again repeat this twice to make up a square.
3. Choose the same cards from the deck of coding cards. Remember to always start with a RECORD PROGRAM - START and end with RECORD PROGRAM - END

It is recommended to teach students to layout the cards in a vertical approach, as this becomes their foundations for learning to code.

			Record Program Start. Open brackets

			Walk 2 strides forward 
			Turn 90 degrees right 
			Walk 2 strides forward 
			Turn 90 degrees right 
			Walk 2 strides forward 

			Turn 90 degrees right 
			Walk 2 strides forward 
			Turn 90 degrees right 
			Record Program End. Close brackets
	Once the robot student is presented this card, they should act out the sequence they were programmed to do. If they make a wrong turn, indicate to the student, there is a bug and they should try and correct it.		

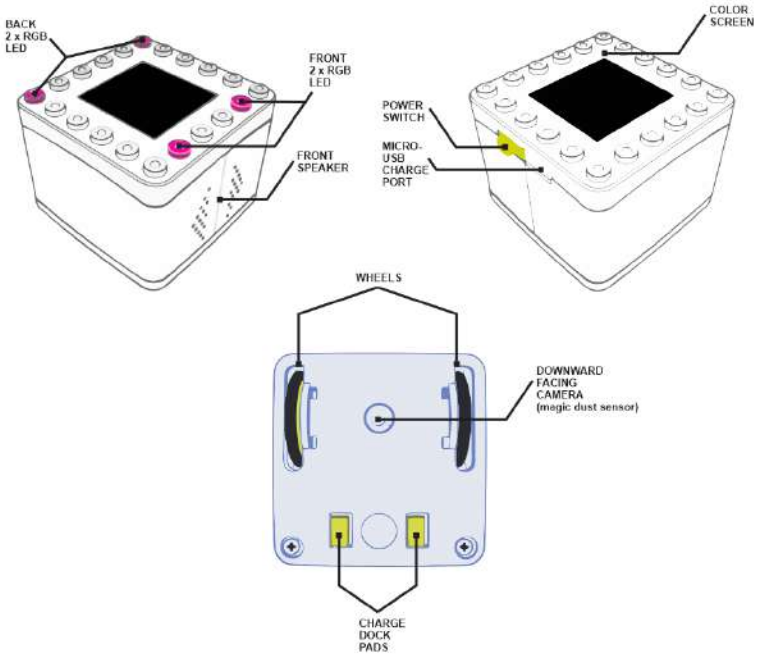
4. Students can hold these cards in their hands and make the movements to complete a square.

Level 2: It’s time to introduce KaiBot

Try and always have KaiBot charged up and ready for action. Nobody likes a flat KaiBot 
You’ll get around 1-2 hours play time with him fully charged.

Getting to know KaiBot

The first thing to note about KaiBot is the center point of the robot axis, this is where its downward-facing camera (a.k.a. the magic dust sensor) is located between its two wheels. So when KaiBot turns, it turns around the center axis of its wheels.



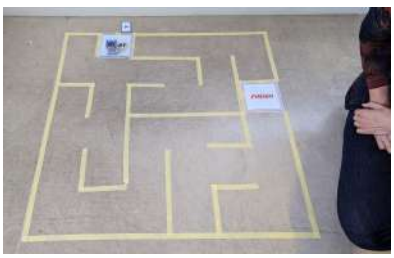
How to Scan Coding Cards

Watch the video

1. Switch on KaiBot and wait for the boot up process to finish and display the unique ID. 💡 **Tip:** We don't recommend you pair KaiBot just yet, we'll get to that later.	
2. Next tap KaiBot on the "Record Program Start" card. This card or a "Record Function Start" card must always be scanned first.	
3. After a successful scan, KaiBot beeps and displays the scanned card on the screen and the LED lights display white	
4. Next lift up KaiBot, about ½" / 1cm away until the display changes red and the LED lights change to red. KaiBot is now ready to scan the next card.	 
5. Tap KaiBot on the next card and repeat the process of lifting KaiBot up and down to scan each consecutive card.	

6. Bug Alert, read the screen and then refer to the Bug section. A bug means an unexpected condition has happened.	
7. Once you have finished your program, you must scan the end program card.	
8. If you have got this far, then you don't have any bugs in your program. Now scan the "Run Program Card", place KaiBot down, then KaiBot displays a 3,2,1 countdown and will then run your program.	
9. You can use KaiBot on the floor or a table and KaiBot will perform your program. BUT why stop here!	You might get wheel slippage on the floor or table so try adding KaiTiles.

If you don't have KaiTiles yet, you can just use masking tape on the floor, HOWEVER, you will get some slippage from the wheels. For accurate driving of KaiBot you can layout your maze with KaiTiles and then add some fun paper craft or MagTiles tiles as walls. MagTiles can be purchased from most leading retailers.

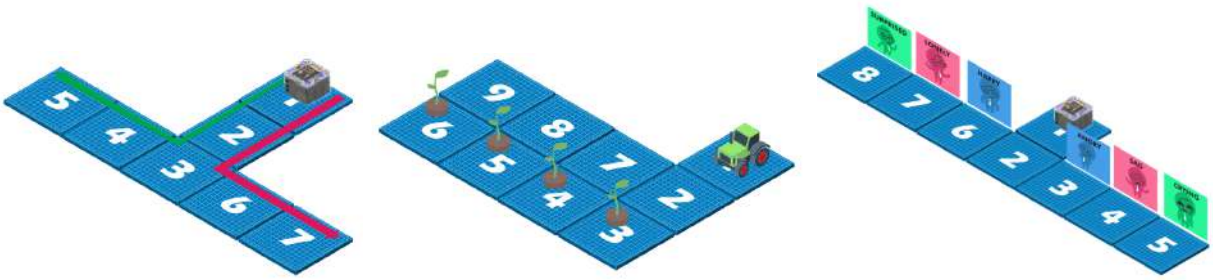


Level 3: Add some Magnetic KaiTiles

	1. Lay-out a square with No1-8 KaiTiles. 2. After you have scanned the “Run Program Card” place him on KaiTile no 1. 3. KaiBot will complete the sequence you have scanned with the coding cards. 4. KaiBot will accurately move and correct any wheel slippage on these tiles. 5. KaiBot can read his x,y and tile number position on the tiles.

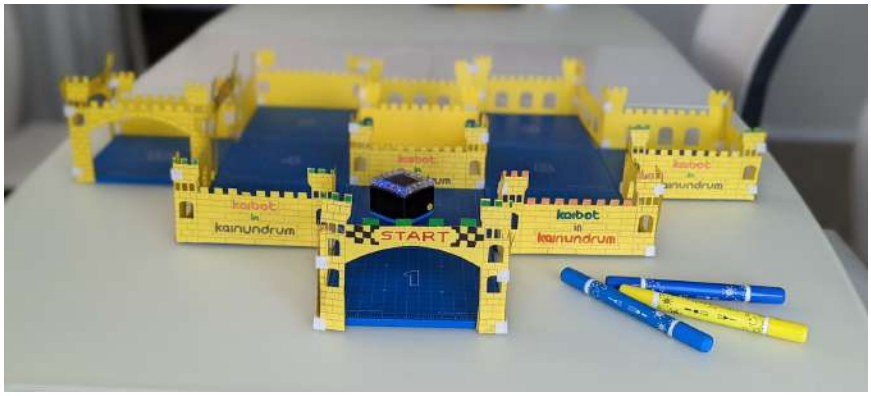
Design different layouts with KaiTiles

💡 **Tip:** When using the Forward or Backward Coding Cards, KaiBot moves an entire tile length. So keep this in mind when designing out challenges for students.



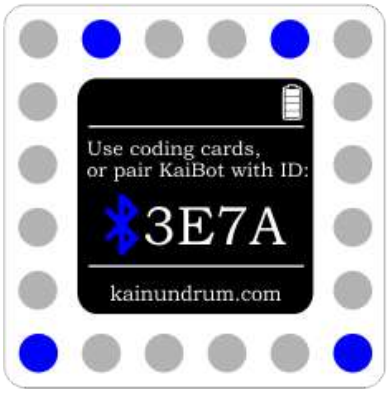
Snap together the KaiTiles and design different challenge layouts and program KaiBot to navigate his way through your maze. As the tiles are magnetic, place a piece of paper in between them to create walls or destination places of interest to reach.

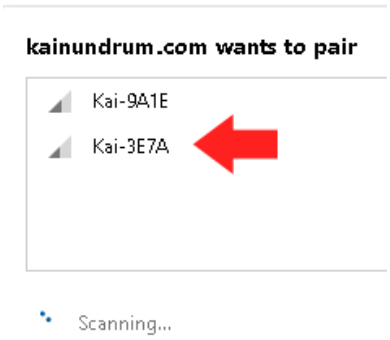
Check out the KaiBot lessons & activities here https://kaiseducation.com/lesson-plans/



Level 4: Pairing your KaiBot with Kainundrum.com

If you are using KaiBot with Kainundrum.com for Hybrid coding you will need Bluetooth to pair it. You can do this by going to Kainundrum.com and looking for the Bluetooth symbol at the bottom of the UI bar.





1. Make sure your KaiBot is powered on and the blue LED lights are flashing and it displays the unique 4-digit code on the screen.

2. Click on the Bluetooth button on your Kainundrum dashboard to begin pairing. A pop-up will appear, displaying the available KaiBots to pair.

   If you have one or more KaiBots appearing on the list, select the one with the same ID as it appears on your KaiBot's screen. The strongest Bluetooth signal KaiBot is always at the top.

3. When you first pair KaiBot to Kainundrum, it may take a minute to load the player's Emoji. The KaiBot's screen will display a percentage bar to show the progress and a similar window will appear on Kainundrum too. The emoji loading can be skipped, but then your KaiBot will not display an emoji.

4. Pairing is successful when KaiBot plays a jingle, the lights flash yellow and your emoji is displayed on the screen. KaiBot remembers the last emoji assigned to it, so if you have set a different emoji on Kainundrum, the robot will have to reload to match.

Once pairing is complete you can start using KaiBot! KaiBot's screen will take on the emoji you have chosen to represent you in Kainundrum. You can always change this by going to Settings > Profile and changing the emoji or color. Remember, just like changing clothes, it takes a while for KaiBot to swap its emoji!

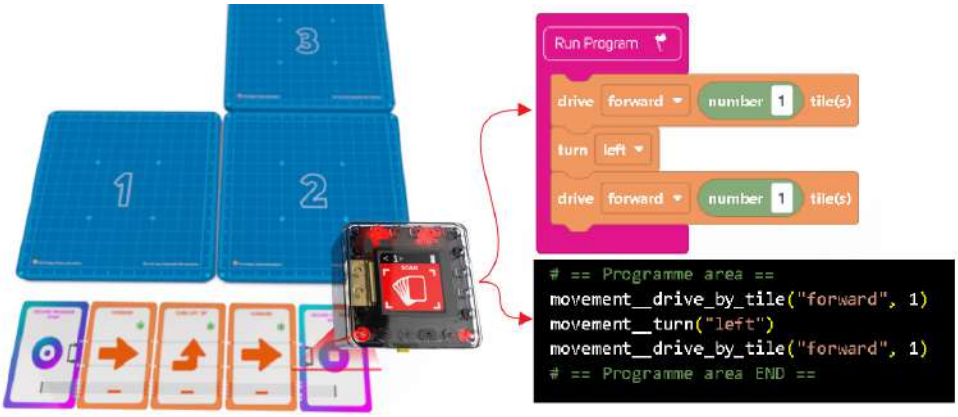
📘 **Note:** Currently Kainundrum.com only works with computer browsers like PC, MAC or Chromebook. Our engineers are working on a tablet edition for Android & iPad.

Progress students from screen free coding to block and text programming.

KaiBot can convert coding card programs into Blockly as well as into text based coding Python. Blockly(similar to Scratch) is a **visual basic coding language** designed to teach beginners how to code.Python is an interpreted high-level general-purpose programming language displayed as **text based coding**.

How to pair KaiBot to Kainundrum.com

1. On your computer browser open [Kainundrum.com](), it is a very large application so wait for everything to load.
2. Switch on KaiBot
3. Next click the Bluetooth pair icon on the the bottom Kainundrum toolbar.
4. The Bluetooth pairing window will pop up, listing all Kaibots nearby. Select the Bluetooth pairing ID that matches your KaiBot. It's good manners not to pair to other peoples KaiBots!
5. In the waiting room you cannot control your KaiBot,
6. Load a project and press start START to start your level.
7. Once you press Start the level will load and you'll be asked to confirm the virtual Kainundrum "Layout matches your physical KaiTiles". If you don't have those tiles then choose another level or create your own level to match the tiles you have.
8. Once you confirm, you'r virtual + physical KaiBot is placed on the tiles.
9. There are several ways to get KaiBot to move:
    a. Use the coding cards. Note: Once you scan KaiBot on the "Record Program End" card, KaiBot will send his program through to Kainundrum and the program will be converted to Blockly and Python text code.
    b. Use Blockly inside Kainundrum
    c. Use Python inside Kainundrm
    d. Or use the arrow keys or WASD to control KaiBot



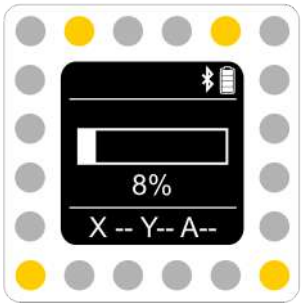
10. If you have a TV or interactive display project the Virtual 3D display in full screen so students can see the interaction between the virtual and physical.

Trouble Shooting the Pairing process

No Bluetooth? You'll need to have Bluetooth on your computer if you want to use KaiBot with Kainundrum. You could purchase a USB Bluetooth adapter. However, Both KaiBot and Kainundrum can be used independently.	**Device ID not in the pairing list?** If you are having trouble with pairing or the ID does not appear in the list, you may need to re-scan for nearby KaiBots. If your KaiBot is displaying an emoji on its screen and you cannot see the ID on the pairing list, that means it's already connected to another device somewhere else.
KaiBot unpairs from my computer while in use.	**KaiBot does power on when switched on.**

Make sure no one else is pairing your KaiBot while you are using it.	KaiBot's battery is flat, you'll need to plug KaiBot into a powered USB socket.
The loading of the emoji is taking too long. Only the last loaded emoji remains in KaiBot memory, so to speed up the process, change your Kainundrum emoji to match your KaiBot, this will dramatically speed up the pairing process. You can also skip the emoji loading, but your KaiBot will not display the image.	

Tip: Have you noticed that all KaiBots are identical siblings? So how do we tell them apart? Every KaiBot includes number stickers you can use to easily identify them. Stick them on KaiBot's body so students can avoid taking another person's little companion by mistake.



Tip: Keep KaiBot close to your computer with the browser open to speed up the emoji loading process

KaiBot is a versatile little guy

This is a comparison chart to see what you can do with KaiBot and the optional accessories which we will break down:

	Scan and program KaiBot with coding cards	Coding cards converted to Blockly code *	Coding cards converted to Python text code *	Move around accurately on KaiTiles	Virtual KaiBot in Kainundrum mirrored with physical Kaibot robot.
KaiBot	✓	-	-	-	-
KaiBot with KaiTiles	✓	✓	✓	✓	✓

Human Robot Interface

KaiBots Home Screen

Once KaiBot is switched on and initialized, you'll be presented with his home screen.
In this mode, you can either scan a "Record Program Start", or "Record Function Start" card or pair your KaiBot with your PC, MAC or Chromebook using www.Kainundrum.com

The top right battery icon shows roughly how much battery is left.

Coding Card Screen

Every valid card that KaiBot scans is counted in the top left corner.
Scanning the same valid card twice can also increment this value or cause a bug notification depending if it is a valid programming statement.

Scan Coding Card Mode

To put KaiBot into scanning mode, place him on a "Record Program Start" or "Record Function Start"

1. Once a card has been read, lift up KaiBot away from the coding card
2. Wait for the lights to flash red and "scan" appears on the screen.
3. Place KaiBot down on the next card.

Repeat steps 1 to 3 until you read the "Record Program End" card.

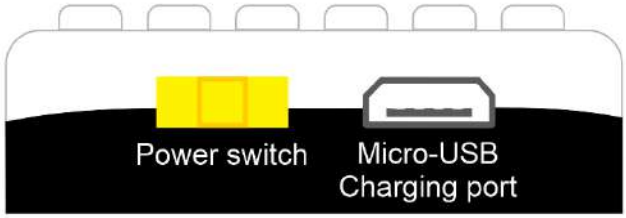
💡 **Tip:** At anytime you can put KaiBot into scanning mode, place him on a "Record Program Start" or "Record Function Start". It's like using a barcode scanner at the shopping mall.

Paired with Kainundrum mode

When KaiBot has been paired with Bluetooth with Kainundrum.
His display will show:
Top right next to battery: Bluetooth paired icon
Top left: Current tile number (if KaiBot sitting on a KaiTile.
Bottom bar: The current KaiTile x, y, and angle that KaiBot is sitting at.

Charging your KaiBot

When you first unbox your KaiBot, it will have some fuel in its tank. However, we recommend to fully charge KaiBot first before you start using it. There are two different ways of charging the robot:
a.) via a USB cable that comes with KaiBot or
b.) using the optional add-on KaiBots Autonomous Charging Dock.

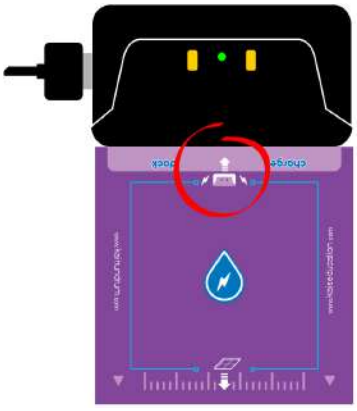


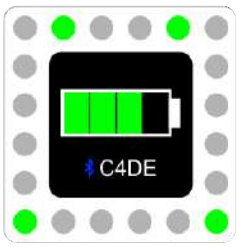
Charging with a USB cable *(included)*

1. Switch off KaiBot
2. Plug in the Micro-USB cable into the micro-usb charging port on the back of KaiBot, next to the yellow power switch.
3. Once it begins charging the lights will flash green and when it's fully charged it will show 100%.

Unplug the cable once fully charged. Overcharging KaiBot can damage the battery.

Charging with KaiBots Autonomous Charging Dock *(optional)*





1. Snap the charging dock tile to the dock and connect the magnetic USB cable to the **left-hand side** of the dock.
2. Switch on KaiBot & simply place him onto the purple charge tile any way you like. KaiBot is super smart and can drive itself onto the charging dock without any help, no matter how you place it.
3. Once charging begins, the lights will flash green and when it's fully charged, it will drive itself off and flash purple.

   If KaiBot is completely flat, switch him off and plug him into the included Micro USB cable.

   KaiBot can also be charged when switched off. Again push him onto the dock and he'll switch on and start charging or use the included charging cable.

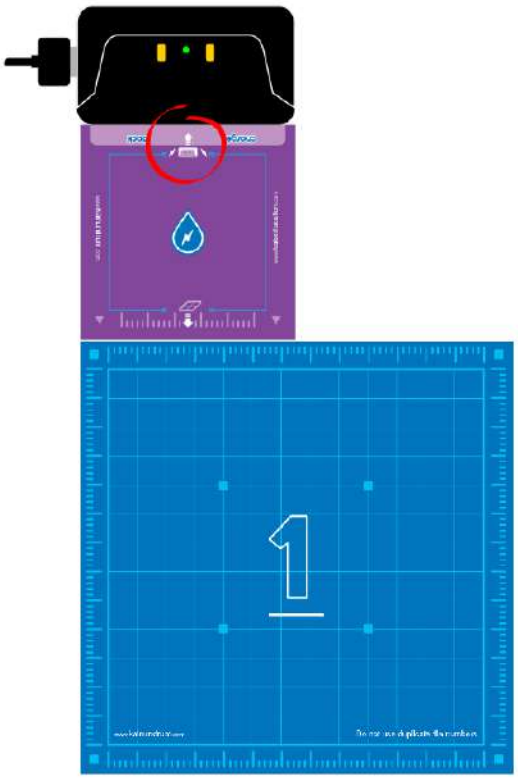
When KaiBot is full, he will move away from the dock and stop charging. If KaiBot is full, he will not automatically navigate to the charging dock when placed on the purple charge tile.

Nap time, If you don't keep KaiBot entertained, he'll start getting sleepy. After 10 minutes of inactivity, he'll turn off the screen and go into a light sleep to conserve battery. To wake him up, just move him around a card or tile. After 30 mins of inactivity, KaiBot goes into a deep sleep and can only be woken by switching him off and back on again. In the deep sleep mode, the battery from a full charge can last around 2 days.

Charging KaiBot from empty takes around 2 hours, sometimes longer if you have nested multiple charging docks together.

Nesting KaiBot Autonomous Charging Docks

With the Charging docks you can connect these via the magnetic connections and nest a bunch of charging docks together. You then only need to use one charging cable and you will be able to charge multiple KaiBots. Using a standard USB port (.5-1 amp) with one Charging dock (.5 amps) will take about 2 hours to charge and you will have 1 hour playtime When nesting 4 Charging docks together you can use a higher USB amp (2 amp) and it will still take 2 hours to charge all 4 KaiBots.	

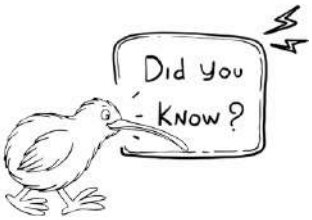


1.  Place the magnetic charge dock on the charge dock icon side of the purple tile and line up the dock and tile.
2.  Place the optional Kai-Tiles on this side of the purple dock tile. The purple dock tile only connects on either the left or right-hand side of a blue Kai-Tiles.
3. Ensure the Charge Dock is powered and the green LED light on the charging dock is on.
4. Drive or place KaiBot onto the purple charge tile and if KaiBots battery is low, it will kick into autonomous charge mode and will automatically drive onto the charging dock.

📘 **Note:** If the charging dock does not have power (green LED) then KaiBot will fail to dock.

📘 **Note:** If the charging dock is not aligned correctly to the purple tile, then KaiBot might fail to dock correctly.

📘 **Note:** Magnets located inside the dock, purple charge tile and Kai-Tiles should help align up, but you might have to correct it slightly for better docking.

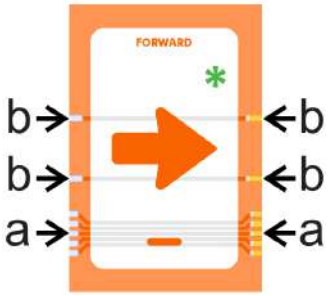


New Zealand launches rockets into space?

New Zealand firm Rocket Labs launched satellites into orbit from its own private launchpad on New Zealand's Mahia Peninsula.

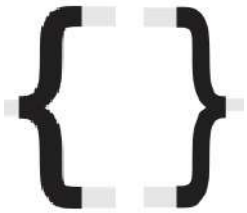
Coding Cards

Elements of Coding Cards



Connectors
These cards also have a coating of magic dust from New Zealand that lets KaiBot know what card he is reading. The cards are specially printed to have a coordinate system and card ID, so KaiBot can read each card.

a. All the coding cards use a generic connecting system, much like a circuit board. They connect to each other from left(silver) to right(gold).
b. The middle connectors help to show what card can be connected next.



What do Curley Braces do?
Curly braces play a big role in code structure within popular programming languages such as Python, C++ and more. In languages like C curly braces {} are used to create program blocks used in flow control. In Python, curly braces are used to define a data structure called a dictionary.

What does the green asterisk * mean?
Any card that displays the green asterisk, lets you know you can add a number card directly afterwards or a “read variable” or a “read tile number” card.
Tip: Placing two numbers cards together makes an integer. E.g. (Forward *) (2) (2) would move forward 22 spaces.

Types of Coding Cards

Object-Oriented Programming (OOP) is a style of programming that involves structuring code into logical, self-contained objects. In object-oriented programming data structures or objects are defined, each with its own properties or attributes. Each object can also contain its own procedures or methods.

The Structure of a Coding Card Program

Data Types and Variables
Programming is all about manipulating data, but what is data? Data is information that we store in our computer programs. For example, your name is a piece of data, and so is your age. The color of your hair, how many siblings you have, where you live, all of these things are data. In the coding cards, we only have one type of data and that is **Numbers**. In coding languages like Python or Javascript, you can also have other data types like Strings and Booleans.

Strings are used to represent text like your name. A string can look like this:
"Hello KaiBot"

Booleans are a value that can either be True or False. E.g. Are you wearing a hat. Boolean looks like this:
true;

Numbers within the coding cards are treated as integers. Fractions and decimals are not integers. A **Number** can be a positive or negative **Number**. E.g. -7 is an integer. A **Number** in Python looks like this:
99;

Variables
In computer programming, a variable has a name and contains a value. Think of a box containing a number of marbles. The box is the **name** and the number of marbles in the box is a **Number**. A variable can be of different data types, **Numbers**, **strings** and **Booleans**. A variable might look like this in Python:
Box = 75;
First_Name = "KaiBot";
Hat = true;

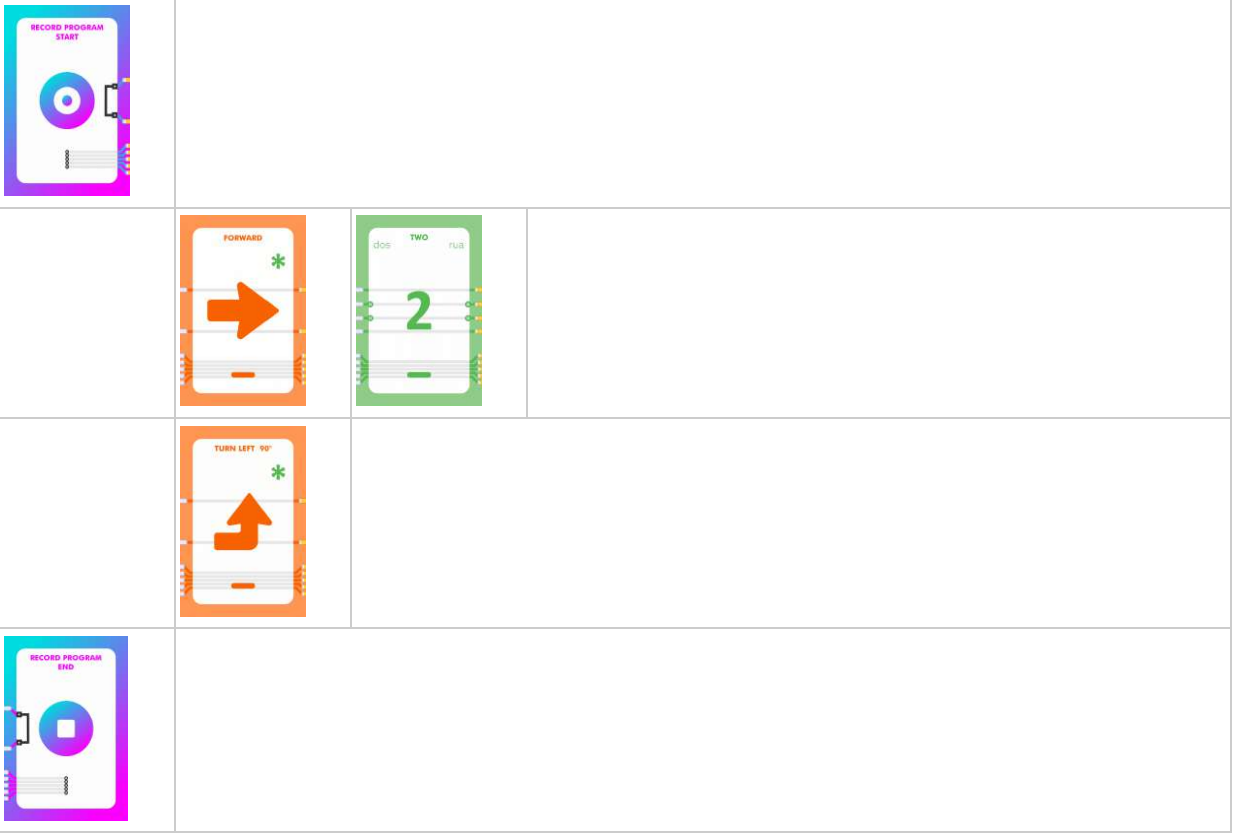
How to correctly structure your coding cards

There are two ways that you can layout the coding cards. The easiest way is to lay them out one after the other left to right, horizontally. The benefit of this method is, that it doesn't use much space up but longer programs become difficult to understand. This might be easy for you as you get started. A programmer would normally code in a vertical indented method as shown below.

Horizontal Layout

The best way to layout coding cards is in a vertical fashion with grouped elements indented. This is much easier for anyone else to be able to read and understand the code and once the student progresses into text-based coding this then becomes natural for them. It does take up a lot of desk space and helps with the read ability of the code.

Vertical Layout - recommended



Check out some of the examples to see how you can improve your code layout.

Basic Coding Card Explainer

The following describes what each card does and how they interact with each other. The below describes the Standard Coding Card Pack.

	CONTROL CARDS
	This is your goto card to start your program! KaiBot must read this card first.
	And like all goods stories, they all must come to an end! Make sure you end your program with this card. If you have KaiBot paired with Kainundrum.com, then your coding card program is converted to both Blockly and text based Python code!
	KaiBot loves this card! Once you tap KaiBot on the **Run Program** card, you have a 3-second countdown before KaiBot runs your assembled program. **Tip:** Make sure KaiBot is facing the correct way. The yellow switch is his rear. This card cannot be used inside your program.
	If KaiBot needs to be stopped in his tracks, pick him up and place him on this card to stop your program from running. You can run the same stored program again, by placing KaiBot on the **Run Program** card again. This card cannot be used inside your program.

	MOVEMENT CARDS
	This card will get KaiBot to move forwards by 15cm or one tile length. You can also add a * **number** card, a “Read variable” or a “Read Tile” card afterwards, to move forward a set * **number** of times. E.g. Forward, 12; moves KaiBot forward 12 spaces.
	Use this card to turn right and once again you’ll see the green * to tell KaiBot how many turns. 1 turn = 90 degrees, 2 turns = 180 degrees, 3 turns = 270 degrees. Try also using “read variable” and see what happens if the variable is a negative number?
	The writer got bored explaining right and left and just used a search & replace in the document. Use this card to turn left and once again you’ll see the green * to tell KaiBot how many turns. 1 turn = 90 degrees, 2 turns = 180 degrees, 3 turns = 270 degrees. Try also using “read variable” and see what happens if the variable is a negative number? Right?
	If you want to reach for the Moon then you going the wrong way if you use this card. Add a * **number** card next to this card and you’re sure to make the distance between the Moon & KaiBot even greater.
	This card makes KaiBot pause for a set * **number** of seconds defined by adding a * **number** card or a “read variable” card.

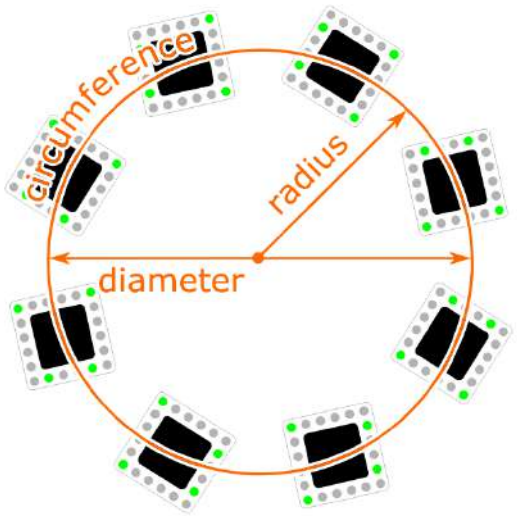
MOVEMENT CARDS

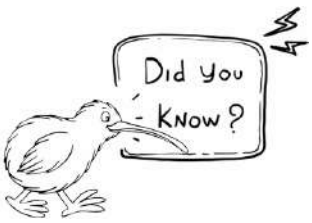
Radius - The radius is the distance from the center point to the edge of the circle.

Diameter - The diameter is a straight line that goes across the circle and through the center. It's twice the length of the radius.

Using this card makes KaiBot drive in a complete circle based on the radius defined by the * **number** card.

E.g. Adding ***number** 5, will make KaiBot drive in a complete large circle, based on the radius of 5 cm away from the centre point of the circle. Radius 1 would make KaiBot drive in a small circle.





More sheep than people.

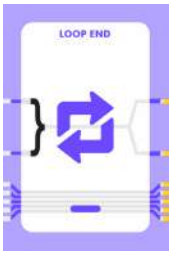
The livestock industry is one of the main industries in the country, particularly sheep. Did you know that for every person living in New Zealand there are at least 10 sheep? The population of New Zealand is 5 million, so can you calculate how many sheep we have?

LOOP CARDS

Loop repeats the code that is between the “Loop Start” and “Loop End”, not including the * **number** cards. Adding * **number** card after this card, defines how many times the loop runs. .

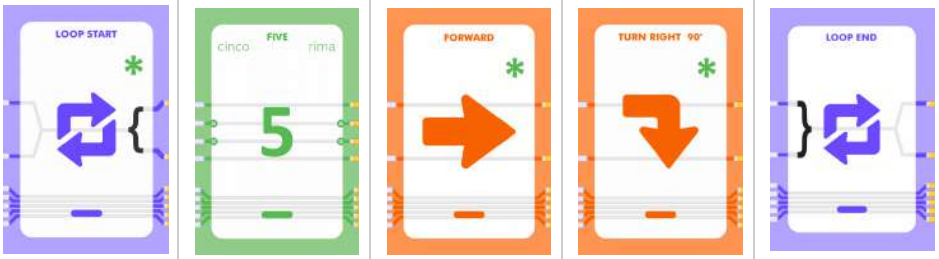
You can replace the * **number** card for a “Read variable” or a “Read Tile” card
E.g. Loop Start 12 {Move Forward} Loop End; this moves KaiBot forward 12 spaces.

Loop is different to the Conditional Repeat.



Loop End is used to close the braces {} of the loop.

LOOP EXAMPLE



This example shows (move forward, turn right), 5 times in the loop.

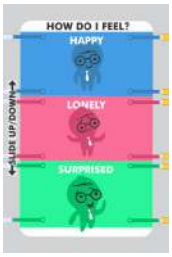
NUMBER CARDS

Number cards can be added directly after any card that displays the green * They can be used along with movement, loops, conditionals and variables.

NUMBER EXAMPLE

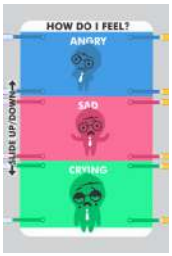
In the above example, adding 2 and then 5 directly after the forward card, makes KaiBot move forward 25 spaces.

SOCIAL EMOTIONAL LEARNING (SEL)



How Do You Feel Today? Some children struggle to communicate their needs and emotions, particularly younger children. Let your kids express their feelings using KaiBot to show their feelings.

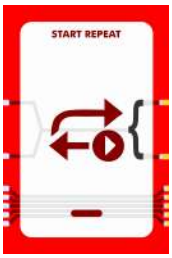
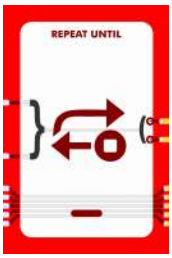
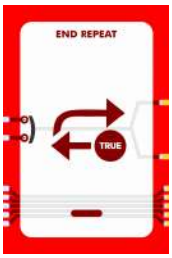
Slide KaiBot up or down the card to program KaiBot to perform that emotion. Only the last emotion selected will be recorded.

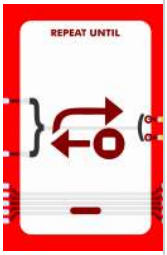
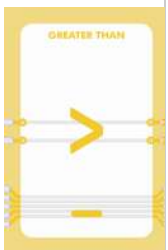
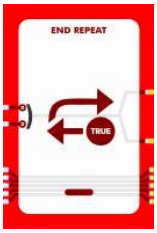


Slide KaiBot up or down the card to program KaiBot to perform that emotion. Only the last emotion selected will be recorded. We rather you dont feel sad, and speak to a teacher or adult if you need help.

Advanced Coding Card Pack (optional)

A star below refers to the "⭐Advanced Coding Card Pack".

⭐Advanced Coding Card Pack	REPEAT CARDS
	Repeat is different to a Loop. A loop loops the code a certain number of times and you can't exit the loop. A repeat will continue to repeat forever, or until a true condition is met. The code that you want to be repeated goes in between the braces {}
	This card is used to end the repeat, it represents the end brace } After this card you need to add the condition that needs to be met. Note the open bracket (
	This card is used to close the bracket) and close the repeat sequence.

REPEAT EXAMPLE						
						

REPEAT EXAMPLE				
In the above example, because 2 will never be greater than 5, KaiBot keeps moving forward. You can replace one of the numbers with a variable card.				

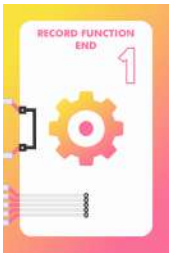
⭐Advanced Coding Card Pack	COLOR MIXING CARDS
	Computer coding involves many different knowledge and skills. Have you ever tried color mixing with coding? On computers, a very common color-coding system is the RGB color model. The RGB color model is an additive color model in which red, green and blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue. When you run your code, the KaiBots screen will display the resulting color, as well as the RGB, LED lights will also display the color. If you have KaiBot paired with Kainundrum.com and run your code, the virtual KaiBot can paint with your newly mixed color in Picasso mode.
	Place KaiBot on this card to select the percentage of red you want to display or mix with the other colors. If you only want to display a percentage of red, then don't add the other color mixing cards.
	Place KaiBot on this card to select the percentage of green you want to display or mix with the other colors. If you only want to display a percentage of green, then don't add the other color mixing cards.
	Place KaiBot on this card to select the percentage of blue you want to display or mix with the other colors. If you only want to display a percentage of blue, then don't add the other color mixing cards.

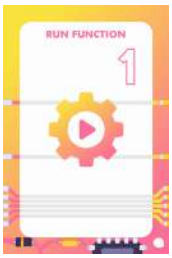
⭐Advanced Coding Card Pack	COLOR MIXING CARDS
	Always close with the Color Mixer End card

COLOR MIXING EXAMPLE

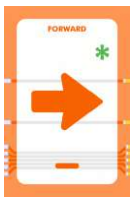



In the above example, adding 100% of Red and 100% of Blue will result in yellow

⭐Advanced Coding Card Pack	FUNCTION CARDS
	There are two sets of function cards. Set 1 and Set 2. Functions are sets of code that contain one or more steps to complete a specific action. Functions can be small, simple, straightforward or long, complicated, and multifaceted. Functions can contain all the regular coding cards within them, excluding the Start, End & Run Program cards. The [bracket opens the function and any cards after this must then be closed with a] Record Function End. **Always record functions, before recording your main code.**
	Place your code between the Record Function Start [and the] Record Function End.

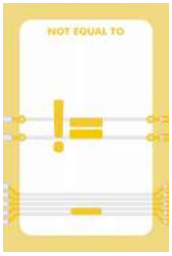
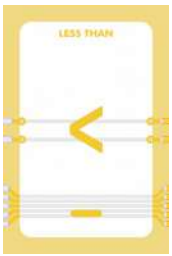
⭐Advanced Coding Card Pack	FUNCTION CARDS
	Place the Run Function card in your main code. Refer to the below example. Placing KaiBot on the Run Function card will not make him immediately perform the function. The function is treated as a subset program of your main code.
	There are two sets of function cards. Set 1 and Set 2. Functions are sets of code that contain one or more steps to complete a specific action. Functions can be small, simple, straightforward or long, complicated, and multifaceted. Functions can contain all the regular coding cards within them, excluding the Start, End & Run Program cards. The [bracket opens the function and any cards after this must then be closed with a] Record Function End. **Always record functions, before recording your main code.**
	Place your code between the Record Function Start [and the] Record Function End.
	Place the Run Function card in your main code. Refer to the below example. Placing KaiBot on the Run Function card will not make him immediately perform the function. The function is treated as a subset program of your main code.

FUNCTION EXAMPLE

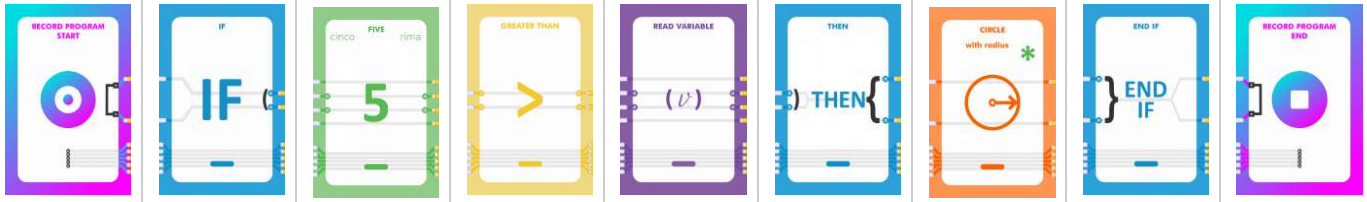


FUNCTION EXAMPLE

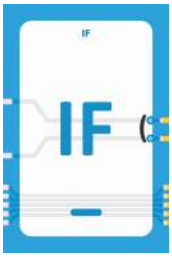
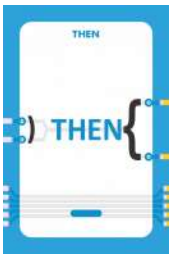
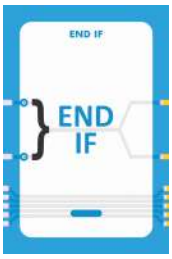
In the above example, the function is first recorded with KaiBot, and then the main program is recorded. When KaiBot runs the main program, the main program will also run the function. The example would have moved the robot forward 10 spaces. Functions can be used to save space in your code.

⭐Advanced Coding Card Pack	COMPARISON OPERATORS
	Are used to compare two values to each other and can also be used to control the flow of a program based on the value of a number or variable. The 'equal to' comparator is also referred to in coding as two equal signs back-to-back '=='. It is not setting two things equal but is returning 'true' if two things are equal. You will often want to compare two things and do something if they are equal, like checking if the current time is equal to the alarm you have set.
	The not-equal-to operator (!=) returns true if the operands don't have the same value; otherwise, it returns false.
	The less-than and greater-than comparators return 'true' when the comparison is true and false when the condition is not-true.
	The less-than and greater-than comparators return 'true' when the comparison is true and false when the condition is not-true.

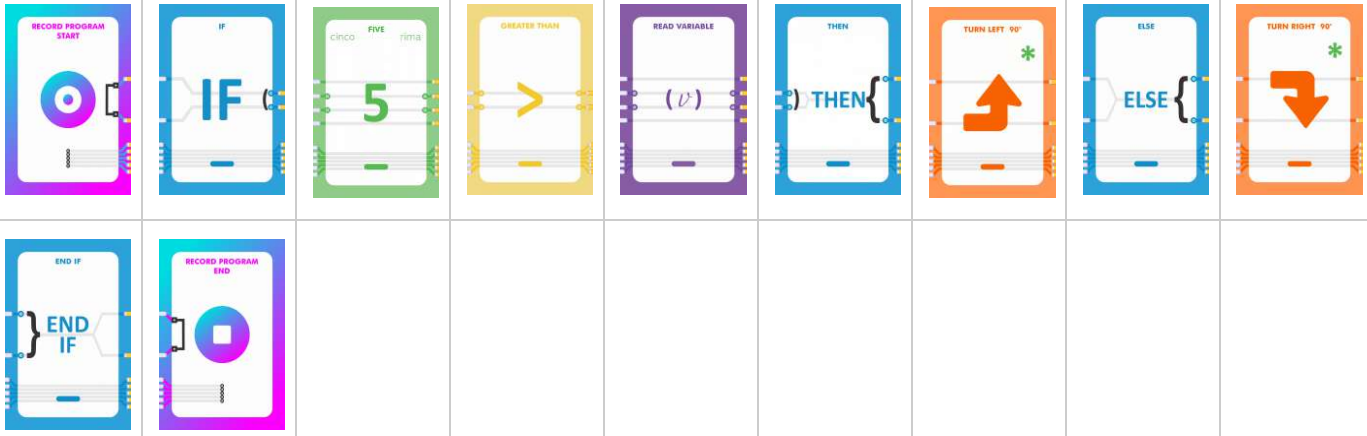
COMPARISON EXAMPLE



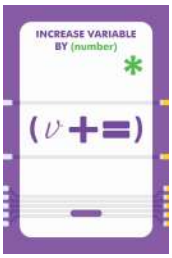
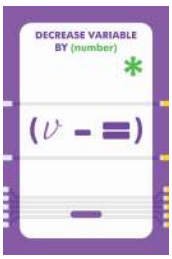
In the above example, IF 5 is greater than the value stored in the variable, then the KaiBot turns in a circle? Remember to always set your variable to a value first.

⭐Advanced Coding Card Pack	CONDITIONAL STATEMENTS
	A statement helps a computer decide what to do next. A condition statement has an If/Then format. For example, If v = 1, then turn left if(condition here is true){ run all of these code statements } else{ run these statements} An **"if"** statement creates a branch within your program so that your code 'makes decisions'. It tests a condition and, if that condition is true, some code statements will run. If the condition is false, the 'if' statement is passed over.
	The "then" statement performs the following code if the condition is true.
	The optional "else" statement performs the following code if the condition is false.
	This ends and closes the entire IF statement.

CONDITIONAL EXAMPLE



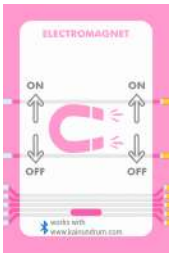
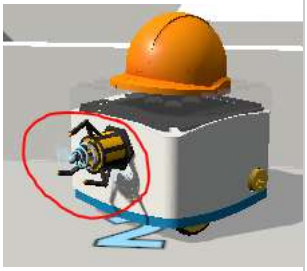
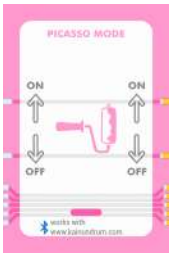
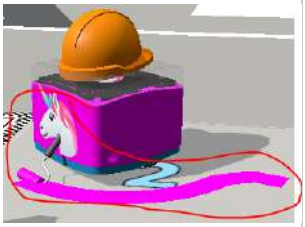
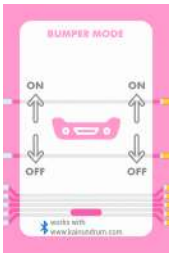
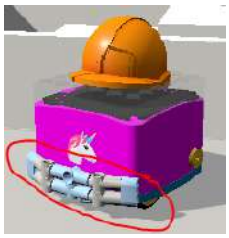
In the above example, IF 5 is greater than the value stored in the variable, then the KaiBot turns left, else KaiBot turns right. Remember the “Else” card is optional.

⭐Advanced Coding Card Pack	VARIABLES
	A variable is a container that holds a value, such as a piece of text or a number. The value can change, which is why it's stored in a variable. The Coding Cards only work with integers, whole numbers that are positive or negative. This card is used to set v to a ✱ number card, placed directly afterwards.
	The Read Variable card is used to load the current value of the variable.
	This card returns the current Kai-Tile number that KaiBot is sitting on. 📘 **Note:** If KaiBot is set to move forward 10 tiles, reading the tile number will not work. Instead, move forward 1 tile, then read the current Kai-Tile number.
	Use this to increase the variable value, by the ✱ number placed after this card.
	Use this to decrease the variable value, by the ✱ number placed after this card.

KaiBot in Kainundrum Cards

The following cards can be used alongside Kainundrum.com when you pair KaiBot, then your virtual KaiBot will mirror the programmed actions from your physical KaiBot.

Requirements:
KaiBot
PC, MAC or Chromebook with Bluetooth
Magnetic Kai-Tiles that are laid out to match the virtual world in Kainundrum.
Tip: You can also create your own layouts in Kainundrum using the Game Builder.

	KAINUNDRUM CARDS	
	Add this card alongside your code to turn on or off your virtual KaiBot Electromagnet. This card acts as a switch, having KaiBot scan the top of the card is on, and the bottom of the card is off. The Electromagnet can be used to pick up cubes in the game.	
	Add this card alongside your code to turn on or off Picasso mode. This card acts as a switch, having KaiBot scan the top of the card is on, and the bottom of the card is off. Picasso mode allows you to paint the path your virtual robot drives. If you want to move the brush to the centre of the robot in order to draw shapes, then use Blockly under tags: 	
	Add this card alongside your code to turn on or off the virtual Bumper. This card acts as a switch, having KaiBot scan the top of the card is on, and the bottom of the card is off. The Bumper allows you to use Blockly or Python to sense if the virtual KaiBot has collided with an object.	
	The engineers who programmed Kainundrum thought it would be fun to hide secret Easter Eggs inside the program. We asked them what they are, but they wouldn't tell us, after all, it is a secret. If you find out what it is, please do let us know, the writer is still looking. *Why did the programmer take a break?* *Because he wanted a byte to eat.*	

Debugging Code

KaiBots built-in interpreter will record each scanned card, alongside this process, is KaiBots bug identifier.
The debugger is in charge of constantly checking the scanned code and ensuring that it's valid.

When it comes to teaching children problem-solving, few methods are as powerful as learning to debug code. The process of finding bugs and implementing debugging strategies is great practice for problem-solving as well as helping children understand how to solve problems they come across in the real world.

When working with computers and writing code, something can always go wrong. It's a major part of computational thinking and problem-solving. When things don't work as you wanted them, it's okay. "Just keep on swimming".

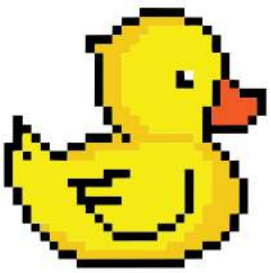
On 9 September 1947 the word "bug" was used by Grace Hopper in her diary, where she referred to a moth as a bug. As this bug was found in the computer, the term caught on and is now used to define errors or faults found in our computers.

Incremental Testing
No programmer wants to deal with a dozen or more bugs all at the same time. Moreover, they don't want to have to search hundreds of lines of code to find the source of a bug. This is why — instead of finishing a large project and then testing — programmers test the code as they go, breaking the code into smaller sections and testing each small section of code.

Even the most complicated computer program, video game, or algorithm can be decomposed into tiny steps. Coders take these individual steps, create them one at a time, and test each as they go. By following this best practice, even beginner coders can manage to debug with ease.

Once KaiBot finds a bug in your code, get the student to try and figure out why it's occurring. The best way to start on this is just to take a look at what the code is doing and get the student to try and perform and walk out the steps and sequences to understand what the code is supposed to do.

Rubber Duck Debugging
In software engineering, rubber duck debugging is a method of debugging code by articulating a problem. This method helps your students find bugs in their code. The term rubber duck refers to an entity that has less than half a brain and understands almost nothing about a given problem.



Rubber Duck Debugging is a programming methodology where you explain the task you are performing with all the details you are trying to achieve, line-by-line. By describing the problem the programmers force themselves to express their ideas in a clear way and step by step sequence. This explanation is called “telling the duck your problem”.

When a coder does so, the bug will be easily spotted as you will be forced to face the logic problems you might have ignored previously. All with the help of rubber duck programming.

Once you understand debugging you are on a path to becoming a better coder and problem solver. 🦆

Computer Programming Jargon

Syntax is the rules of how a programming language works. To put it simply, the syntax of a programming language defines how you should write the statements in a program. It is a set of rules that tell you what combinations of symbols are valid or invalid.

Here is an example of the construct of an English sentence. E.g. - What is your name?
Here we put the pronoun first, then the verb and so on. We don’t say “Name your is what?” or you might sound like Yoda.

Forward 10; is valid
10 Forward; is invalid

A Syntax error is caused by how you wrote your code that breaks the rules of the syntax to make the code invalid.
Logical errors are problems with the logic where the program cannot make sense of what it was asked to do. If a program works differently than you expected it to work, there is a good chance that there’s a logical error somewhere.

Magnetic snap together Kai-Tiles

All robots and even cars suffer from their wheels slipping on the ground, this slipping can cause the direction of the robot to go off course. Even a small amount of error can have a large effect when moving over a greater distance.

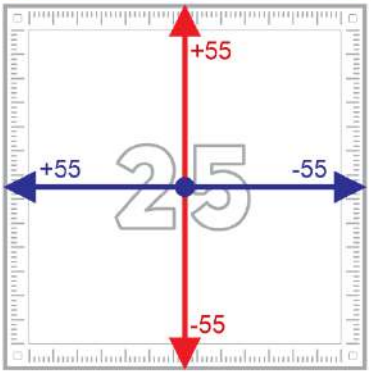
Screen-free without Kai-Tiles
KaiBot can be used without tiles, however, the robot might not always turn to the correct angle or move to the desired position.

Screen-free with Kai-Tiles
To get KaiBot to follow instructions, it's best that you use KaiBot with Kai-Tiles. When using KaiBot with Kai-Tiles, KaiBot always knows where it is as KaiBot can read the tile number as well as the x & y position on the tile.

KaiBot without Kai-Tiles

When KaiBot is moving around just on a flat surface without KaiTiles, his wheels can slip and go off course. When you code him to turn left 90 degrees and he turns less or more than that, his wheels have slipped. Try using KaiBot on different surfaces and you'll see the effects of slipping more or less. Just like a car can lose traction on an icy road. If you have Kai-Tiles then you are in for a treat as KaiBot corrects this slippage.

Tip: Without Kai-Tiles, you won't be able to mirror KaiBot in Kainundrum. However, you will be able to use Kainundrum separately.



KaiBot with Kai-Tiles

Did you know? These magnetic Kai-Tiles are coated with magic dust all the way from New Zealand (only kidding).

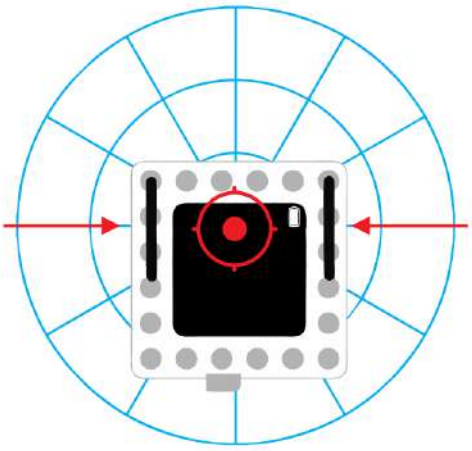
The tiles and coding cards are actually specially printed, so KaiBot can read his x and y positions when placed on them.

KaiBot wheels can still slip on these tiles, however, after every movement, KaiBot compares the destination angle less the angle he is facing and then makes a slight correction.

Each tile is uniquely numbered and uses a mathematical grid. It uses a Cartesian coordinate system with a two-dimensional plane with x & y positions. X determines the horizontal plane and y the vertical plane of each tile.

You can view your robot's current coordinate, both on the robot screen and also in Kainundrum.com in the 3D virtual view, use the top info bar to see the current tile number, angle, x & y coordinate.

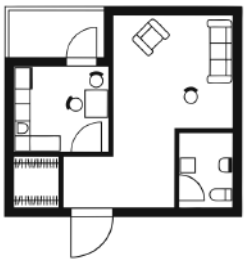
💡 **Tip:** If you place KaiBot in the middle of a Kai-Tile, the *y* position will be out by about 8 points. This is correct as indicated in the diagram in red. Offsetting KaiBot and placing the KaiBot camera over the centre of the tile will result in a 0 *x* & 0 *y* position on a Kai-Tile.



Kai-Tiles in the Classroom

✋ **Stop:** It is highly recommended to always pack away your Kai-Tiles in numerical order as this will dramatically help when it comes to building a tile layout in numerical sequence the next time you use them.

Design & build a Kai-Tile layout



If you are planning to use Kai-Tiles alongside Kainundrum, then just like planning to design & build a house always start with designing your floor plan.
The floor plan design should be created in the Kainundrum game editor tile builder. Once you've built the virtual floor plan, match the same layout and sequence with your physical Kai-Tiles.

💡 **Tip:** Only build the virtual tile numbers that match with the physical Kai-Tiles numbers you have.

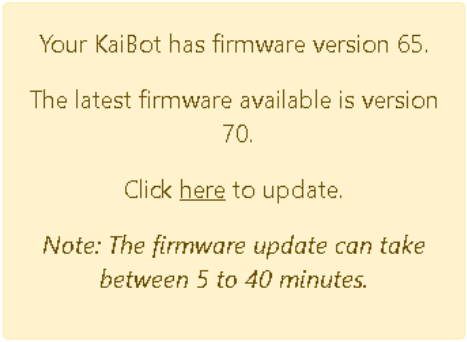
KaiBot firmware update

To check if you have the latest version of KaiBot software on your KaiBot.

1. Pair KaiBot with Kainundrum.com
2. Wait for your emoji from Kainundrum to be loaded onto KaiBot.
3. Click the Bluetooth icon again on Kainundrum.com

You'll see a message saying "This robot is up to date."

Or this message and follow the prompts to update your KaiBot.



KaiBot in Kainundrum with Kai-Tiles

Kainundrum is a high-level block based programming language and website aimed primarily at children as an educational tool for programming and supports two programming languages, Blockly block based programming and Python text based coding.

With the optional Magnetic Kai-Tiles

1. visit Kainundrum.com and pair your KaiBot
2. use the Kainundrum Game Builder to design your tile layout,
3. Create your unique virtual tile layout and ensure that you have the same matching number sequence as your physical tiles.

Once you start you're Kainundrum game, you'll be asked to confirm that your physical tiles match the virtual tiles. If they don't go back to the Game Builder and amend them.

In the waiting room, you only have a virtual robot. But once your KaiBot is in the game it is then paired with the physical KaiBot and both the virtual and physical tiles are mirrored.

Virtual Tiles	**Physical Magnetic Tiles**
15cm x15cm / 6" x 6"	15cm x15cm / 6" x 6"
Coordinates -55 to +55	Coordinates -55 to +55
Each tile is uniquely numbered	Each tile is uniquely numbered

💡 **Tip:** You can renumber your starting tile in Kainundrum to match your physical tile numbers.

Click tile editor , on the top right, in the input box on the top right input your lowest physical tile number.

Do I need KaiTiles?

In order for you to see your physical KaiBot mirrored in Kainundrum, you'll need Kai-Tiles. KaiBot reads his position on the KaiTiles and then his position is mirrored in Kainundrum.

Although you can use KaiBot just on the floor or table without KaiTiles you won't be able to pair your robot with Kainundrum. KaiTiles come in packs of 10 tiles, with each having its own unique numbers.

So if you want to buy 60 KaiTiles you will get numbers 1-60. You can build a simple square to code (remember you can use a loop sequence) or why not build a maze that gets 30 KaiBots ready to race to the finish line. The more tiles you have the more fun and Maze mayhem you'll have with KaiBots.

Get creative and add some walls and build some lasers with red wool and add a few Lego gates. Now try your hand at coding your way around using the screen-free coding cards or jump on Kainundrum and copy your maze creation in the virtual world. You may want to start in the virtual world, Kainundrum and

then copy the virtual maze with your physical KaiTiles. Either way will work. Invite your friends to join your race and you are ready to set and GO!

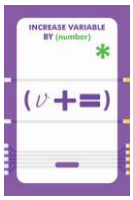
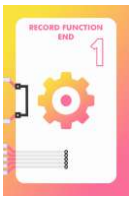
If you need more inspiration then jump on a few lesson plans. These have already got the virtual KaiTiles laid out, so copy those with your own KaiTiles and you are ready to go.

Here are 6 levels you can build that will teach from moving forward to a tricky journey around those squares. Try the code and remember it's always about using the least amount of coding cards to complete each level.
Help a lonely unicorn regain his color - Google Docs

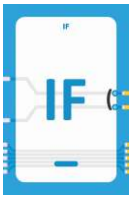
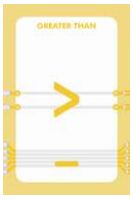
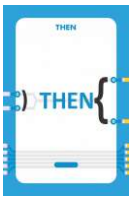
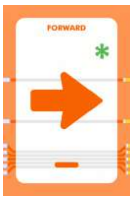
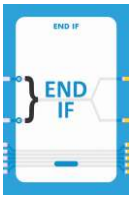
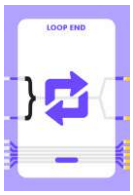
Coding Card Examples

The below examples have been laid out with a text coding vertical layout for ease of understanding and readability. These examples are using the advanced coding card set.

Example 1: Loop + increase variable

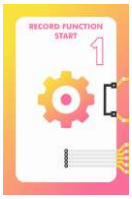
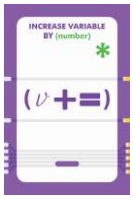
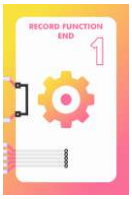
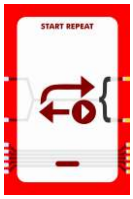
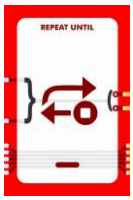
				Always define functions before the record program start card.
				
				Initialise the variable at the start of your program. Do this outside of any loops or conditions.
				Loop 5 times
				Run the Function 1 that increments the value of the variable + 2

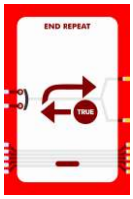
Example 1: Loop + increase variable

						IF v > than 5		
					Move forward 1 tile if v > 5			
					IF v is less than 5, turn left			
			End the IF					
		Run the loop again until it's been run 5 times, then exit the loop						
								

A similar example to above but uses a repeat forever, and only exist when a condition is true

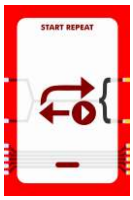
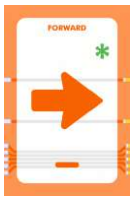
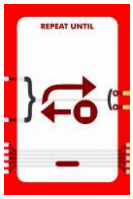
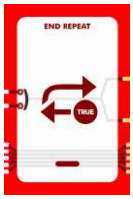
Example 2: Repeat + increase variable

				Always define functions before record program start				
								
		Repeat is a never-ending loop and can only exit when a condition is true.						
			Run the Function 1 that increments the value of the variable + 2					
			Move forward 1 tile / 15 cm / 6”					
					IF v > than 5 then exit the repeat, otherwise repeat forever.			

Example 2: Repeat + increase variable								
								
								

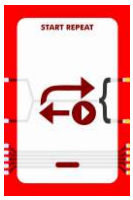
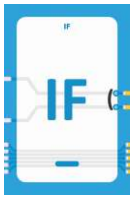
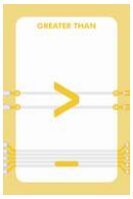
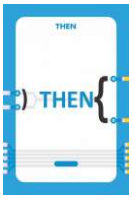
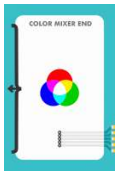
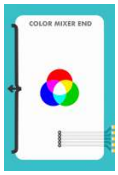
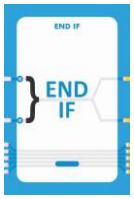
Example 3: Repeat until the Kai-Tile

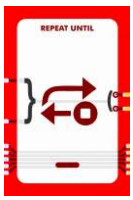
This example gets KaiBot to move forward 1 Kai-Tile or 15cm/ 6” at a time and then the repeat is stopped once KaiBot reads the tile number

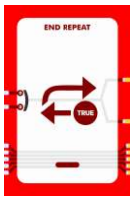
								
			Repeat forever or until KaiBot reads Kai-Tile number 5					
					The repeat will exit if this condition is true.				
		End the repeat						
			Turn right two times					
								

Example 4: Change color based on the value of a tile number

This example gets KaiBot to change his LED lights based on if a Kai-Tile is greater and smaller than a set value.

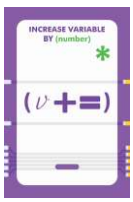
<table>
<tr><td> 
</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td> 
</td><td colspan="7">Repeat forever or until KaiBot reads Kai-Tile number 5</td></tr>
<tr><td></td><td></td><td> 
</td><td> 
</td><td> 
</td><td> 
</td><td colspan="3">When KaiBot reads the Kai-Tile number to be greater than 5, then set it as true else it’s false.</td></tr>
<tr><td></td><td></td><td></td><td> 
</td><td> 
</td><td> 
</td><td> 
</td><td colspan="2">If true display red LED lights</td></tr>
<tr><td></td><td></td><td></td><td> 
</td><td> 
</td><td> 
</td><td> 
</td><td colspan="2">If false display blue LED lights</td></tr>
<tr><td></td><td></td><td> 
</td><td colspan="5"></td></tr>
</table>

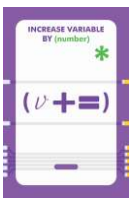




Example 5: Unknown number

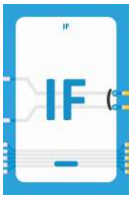
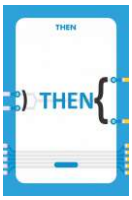
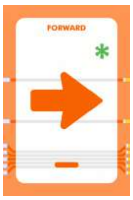
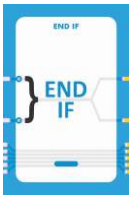
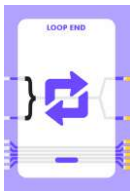
Set the unknown number





Coding cards are based on Pyhton text based coding and use the += operator. The addition assignment operator (+=) adds the value of the right operand to a variable and assigns the result to the variable.

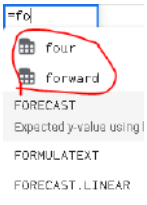
Example 5: Unknown number

							IF v = 18	
					If v = 18 then move forward 1 tile			
					Else left left			
			End the IF					
		Run the loop again until it's been run 5 times, then exit the loop						
								

Design your own code examples

Use this free template in Google Sheets to build and layout your own coding structure or tile layouts.

1. Simply open this link KaiBot Coding Cards Template
2. Browse over the collection of cards
3. Click on the sheet “My code”
4. Click on an empty cell and type *=name of card.* 



or you can refer to the cell of the coding card and simply copy and paste code.

Troubleshooting

KaiBot is not coming on when I switch him on.

What you killed poor KaiBot? Only kidding, his battery is most likely completely drained. Use the included USB to micro USB cable and directly plug in the cable at the back of KaiBot, then plug the USB cable into a powered USB port, like one to charge your phone. He might try and turn it on a few times, just leave him and he’ll soon be back to normal. Once you see the full 4 bars on the screen, go ahead and unplug him.

When I move from one card to the next card, it doesn't scan the card.

You’ll need to lift up KaiBot until he displays the red scan screen, before scanning the next card.
If you printed out your own cards, then these won’t have the magic dust from New Zealand on them and unfortunately, KaiBot will not be able to read them. You must have the official coding cards that come with KaiBot or have the Advanced set of coding cards.

Can I print my own cards out and will KaiBot read them.

KaiBot is one special robot and can do amazing things, but without a coating of Magic New Zealand dust on the cards, he won’t be able to read them. You must have the official coding cards that come with KaiBot or have the Advanced set of coding cards.

Why does KaiBot not drive or turn in a straight line?

Have you ever slipped on a banana skin, no, well I don’t recommend it. KaiBot has to deal with wheel slippage all day long, this is a common thing for cars and robots. If you use KaiBot on the Magnetic Kai-Tiles, he can counteract and correct any wheel slippage by reading the x and y position that KaiBot reads from the tiles.

When I place KaiBot perfectly in the centre of a Kai-Tile, its x & y reading is offset by about 8 points?

Great question, this is because KaiBots downward-facing camera is offset from the centre of the robot. Turn him over and take a look. See how the round hole for the camera is located between the wheel axis. KaiBot is purposely designed this way to actually get its position when turning.

Why did the spider buy a computer?

Because he wanted to browse the web. Lol, okay, enough of the jokes, let's get you coding, KaiBots going to get you to new heights. Don’t forget mistakes are proof that you are trying.

Feature requests & bugs

If there is something you would like added or changed to KaiBot or Kainundrum we would love to hear your feedback after all KaiBot and Kainundrum has been built on user feedback by teachers and parents. Email newideas@kaisclan.ai we would love to hear from you.

Visionary Inspirators

Meet some of the brilliant people that helped to inspire and guide us to make Kainundrum perfect for the classroom.

Brian Host

Brian is a passionate educator that loves working with 21st-century learners and technology. Driven by the current pedagogical research into personalized learning, differentiation, and technology integration he sees students become lifelong learners that engaged with their world.

Martin Levins

Martin is a recovering Director of IT, a tinkerer and a thinkerer, he loves technology because he likes playing & believes it's really important in learning.

Heidi Williams

Heidi Williams is a passionate coding and computational thinking advocate. She has over 30 years of experience in education including 17 years as a 6-8th grade teacher, 6 years as an instructional and technology coach, and 5 years as a K-8 administrator. Williams has shared her passion for integrating coding into the curriculum at local, state, regional, and national conferences, and many have leveraged her expertise for conference presentations, coding coaching, professional development, and K-12 scope and sequence alignment of computer science skills throughout the curriculum. Visit nofearcoding.org for a wealth of CS administrator and teacher resources.

Alicia Verweij

Education consultant with over a decade of experience in corporate America, successful business owner, 12 years of teaching experience in various grade levels, professional development leader, and business consultant. Holds a Master's degree in Educational Leadership, a Bachelor of Science in Business Management, and an endorsement in Gifted Education.

Kerri Wilder

Education consultant with 24 years of elementary and secondary experience in Mississippi public schools as a teacher, professional development coordinator, and administrator. Holds a Master's degree in Educational Leadership, a Bachelor of Science in Education with a specialty in Upper Elementary, and additional licensure endorsements in 7-8 Science, K-12 English, and Psychometry.

Expert Contributors

Although Kai's Education owns all copyrights of Kainundrum and the Kainundrum user manual, we wish to recognize those who have made contributions of one kind or another to Kainundrum. Contributors are listed here, in somewhat random order:

Karen Binns

- IT Teacher/ IT Integrator/IT Support
- School: St George Christian School
- Grade Level: Primary School
- Location: Sydney, Australia

Karen Binns was awarded the International Society for Technology in Education's Making IT Happen award, during a conference in the US.

Her passion lies in improving digital learning opportunities for students. She is well known at her school for trying new things and introducing new technologies. She is adventurous in teaching, always looking for ways to make learning more interesting and engaging. Technology was a natural fit as it makes the impossible possible for students. New technologies can modify and redefine existing pedagogy.

Jodi Mahoney

- Title: Technology Teacher
- School / Organization: Carl Von Linne Elementary
- Grade Level(s): 6-8th
- Location: Chicago, IL

I am a technology teacher at Carl Von Linne Elementary in Chicago, Illinois. One of my passions is helping students develop technology skills in an environment that challenges them to experiment and take risks, and sometimes step outside their comfort zone to discover their amazing potential.
I teach a weekly technology class for K-6 graders. My elective classes for middle school students include Coding and Robotics, Esports, Photography, and Production Studio.

Rudy Neufeld

- Title: Senior Author & CEO
- School / Organization: Neufeld Learning
- Grade Level(s): K - 12
- Location: Ontario, Canada

In the 1980s, Rudy Neufeld, a mathematics leader in schools and in teacher technology educationin Ontario, Canada, met Dr. Seymour Papert at MIT the founder of the programming language Logo which uses code to direct a robot on the floor and on the computer screen. In following years, Neufeld authored several books on exploring math with the use of coding. In 2000, Neufeld Learning Systems, a team of 20 educators and programmers, released “Understanding Math”, for K to 10. 40 years of experience as educator, author and conference speaker in North America and beyond helped Neufeld cultivate a vision in which interactive software empowers learners.

FCC Warning Statement

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment. This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.

- Increase the separation between the equipment and receiver.

- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.

- Consult the dealer or an experienced radio/TV technician for help.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.