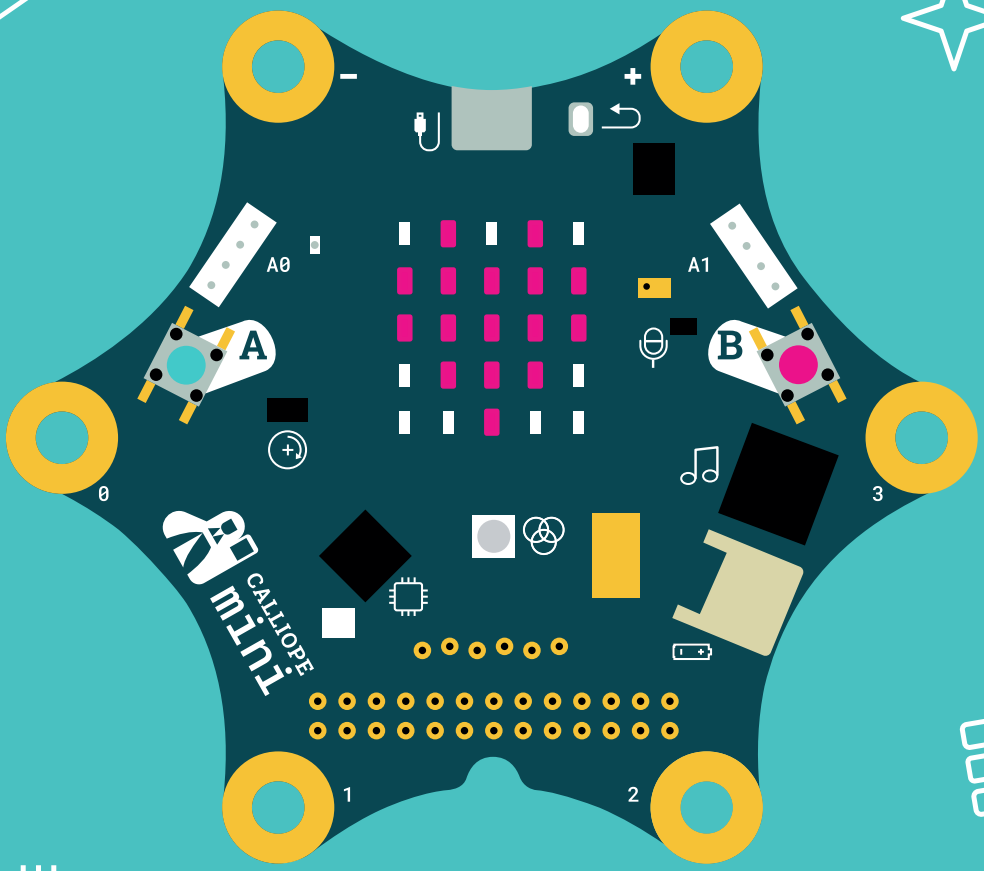


Coding with the Calliope mini

Student material

CALLIOPE

Coding with the Calliope mini

Programing in primary school

Student material from 3rd grade on

Authors: Michael Abend (Morse with the Calliope mini, create image impulses and stimulus words with the Calliope mini and glossary p. 39)

Kirstin Gramowski (The Calliope mini as a mini piano and the Calliope mini as automatic scooter light)

Lars Pelz (The Calliope mini as a random generator and the Calliope mini as multiplication tables trainer)

Bernd Poloczek (Introduction and glossary p. 40)

Consultants: Michael Abend, Kirstin Gramowski, Lars Pelz, Bernd Poloczek

Editors: Kirsten Pauli, Patrizia Schwarzer

Illustration: Benedikt Beck, Nürnberg
Calliope gGmbH, Berlin: pages 7, 10, 15, 40 and U1, U4
Open Roberta Lab: pp. 4, 5
zweiband.media GmbH, Berlin: all vector graphics

Cover design: original cover design made by COSAKitchen, Corinna Babylon, Berlin
English cover design made by Calliope gGmbH, Berlin

Layout, graphics and technical implementation: zweiband.media GmbH, Berlin

www.cornelsen.de/calliope
www.calliope.cc/en/shops

1st edition 2019

All prints of this edition are unchanged in content and can be used side by side in class.

Translated by Calliope gGmbH based on the original title “Coden mit dem Calliope mini”

Index

 Task in booklet

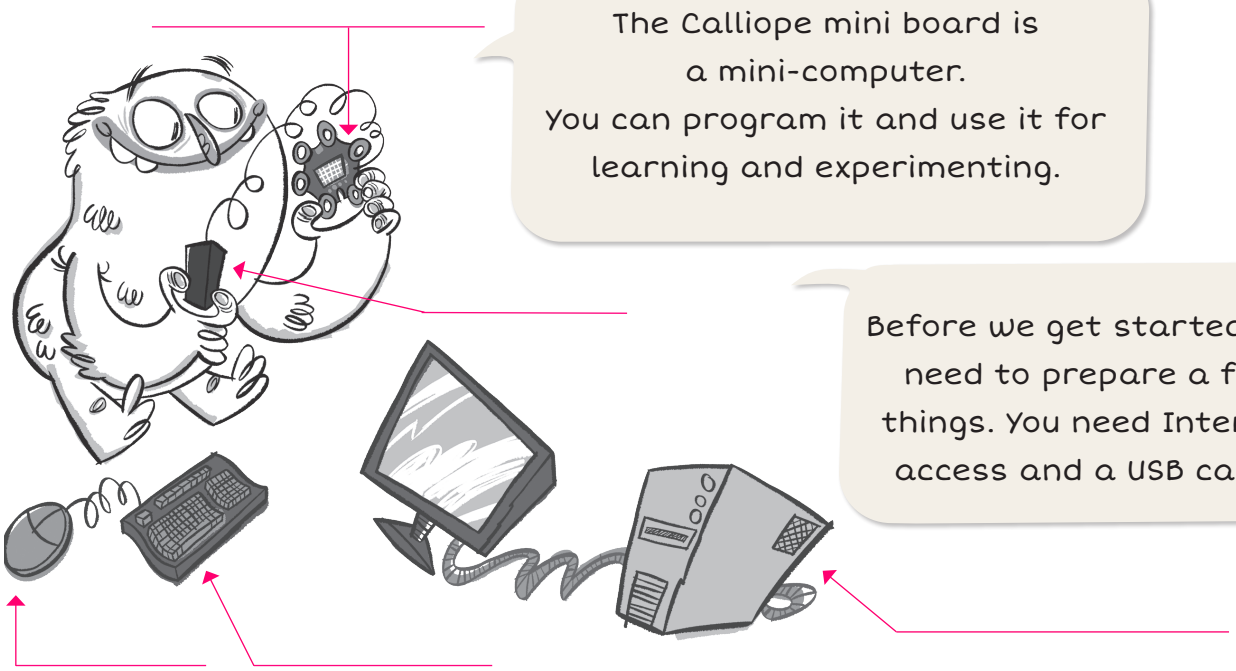
 Task on the computer

 Task with the Calliope mini

Introduction to coding with the Calliope mini

I am Lio.
Welcome to the world of coding.
Do you want to learn
how to program a computer?
I would be happy to help you.

The world is full of computers. Some are big, some are really tiny.
Computers are part of an incredible amount of devices.



The Calliope mini board is
a mini-computer.
You can program it and use it for
learning and experimenting.

Before we get started, you
need to prepare a few
things. You need Internet
access and a USB cable.

1. Label the pictures with the following words:
computer, mouse, Calliope mini, keyboard, battery

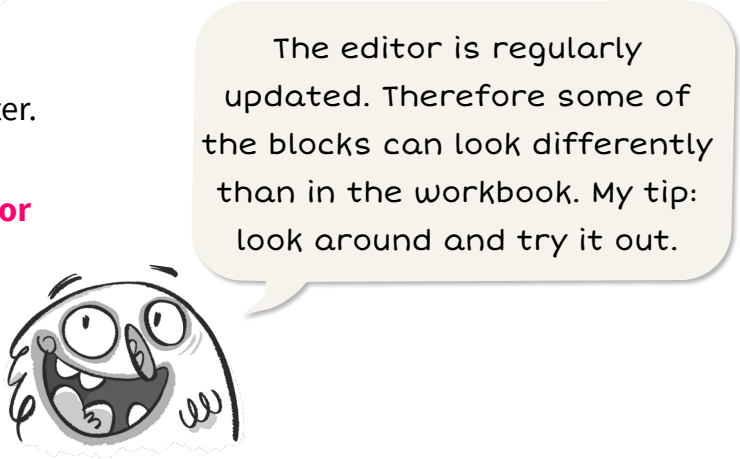
2. The instructions of how to eat a banana got mixed up.
Put them in the correct order.

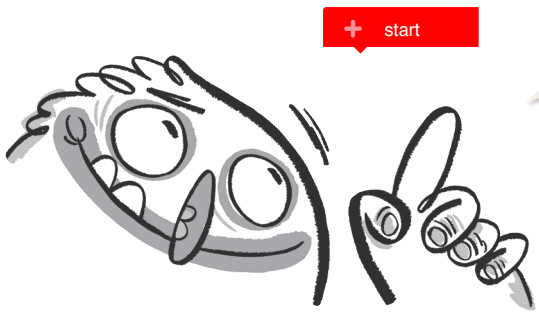
- ☐ Take the banana.
- ☐ Chew.
- ☐ Bite a piece off the banana.
- ☐ Repeat the following procedure until the banana is eaten.
- ☐ Swallow.
- ☐ When the banana is eaten, throw the peel into the garbage.
- ☐ Remove the peel.

Writing such commands is called programming or coding.

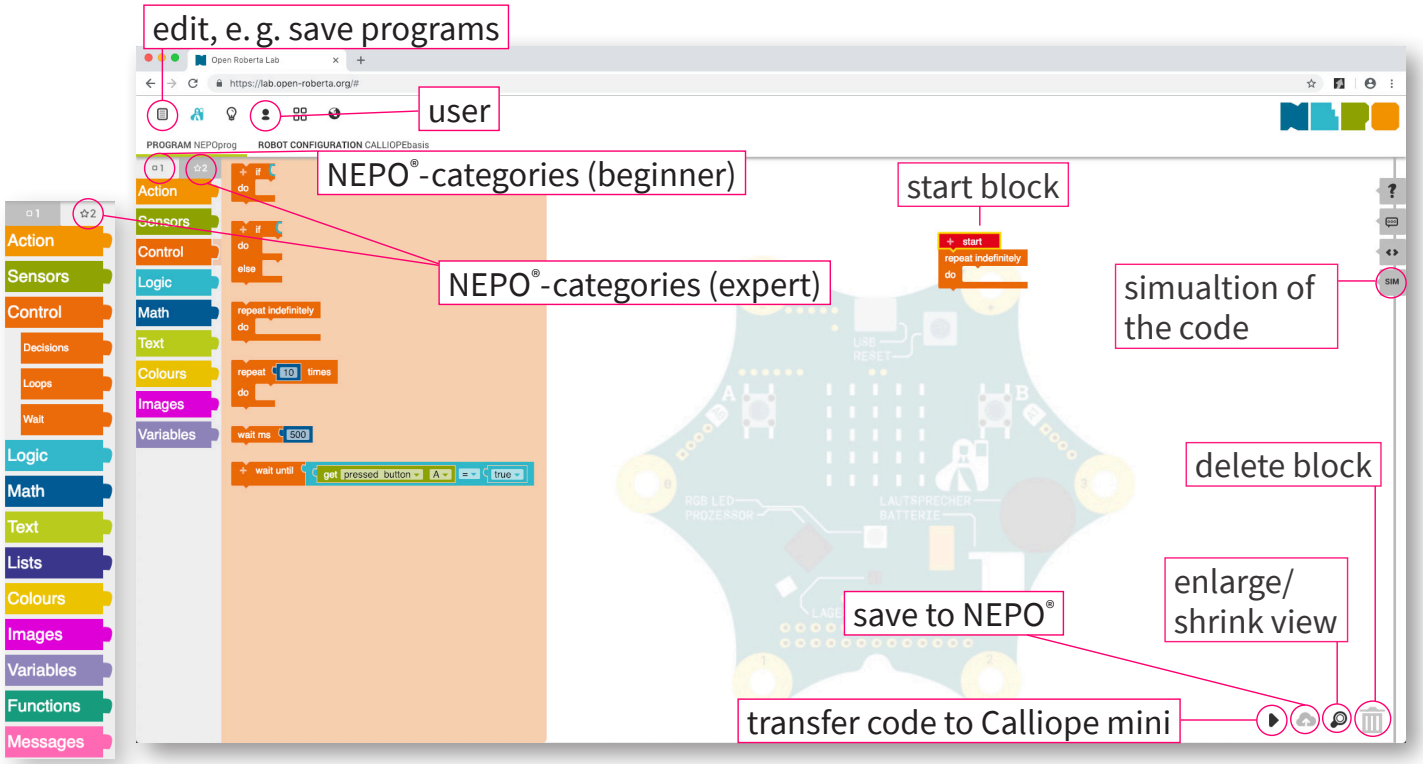
3. Open the NEPO® editor on your computer.

- Open our browser and enter
  https://calliope.cc/en/lets-start/editor
- Scroll down and click
  lab.open-roberta.org

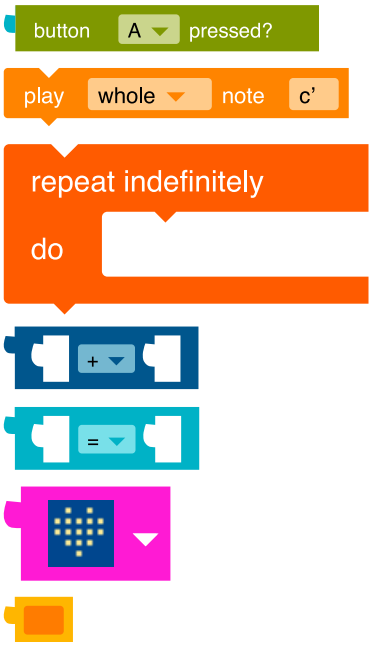


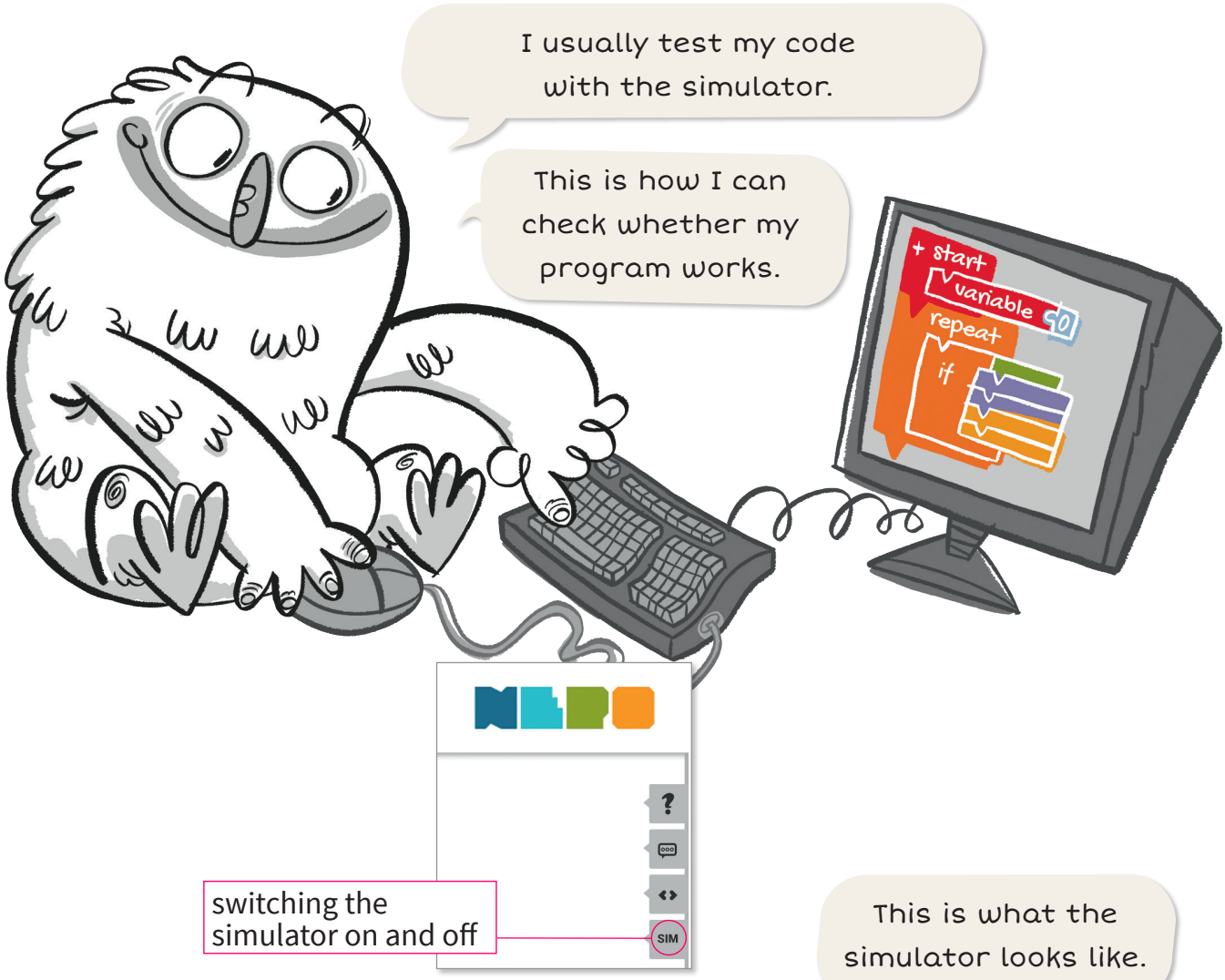


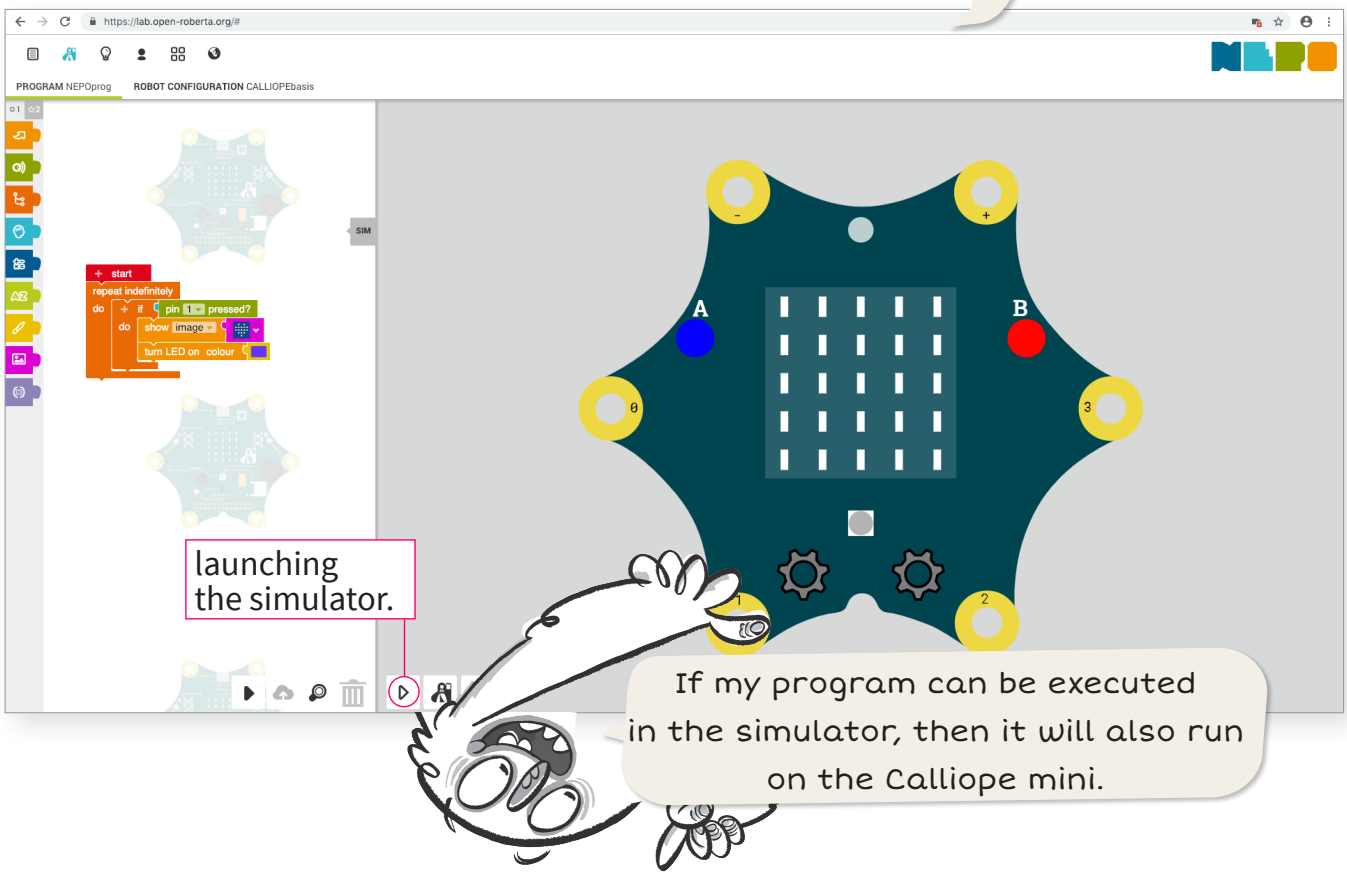
This is what the editor looks like! In the nine categories on the left you'll find the blocks you can code with. The starting block "start" is already there.

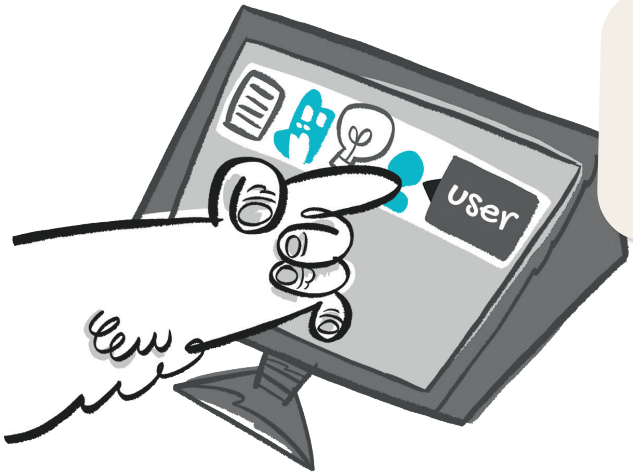


4. Click on the categories and discover which blocks are hidden behind them. In which categories do you find the following blocks? Write them down.




I usually test my code with the simulator.
This is how I can check whether my program works.
start
variable
repeat
if
switching the simulator on and off
SIM


This is what the simulator looks like.
https://lab.open-roberta.org/#
PROGRAM NEPOprog
ROBOT CONFIGURATION CALLIOPEbasis
start
repeat indefinitely
pin 1 pressed?
show image
turn LED on colour
SIM
A
B
launching the simulator.
If my program can be executed in the simulator, then it will also run on the Calliope mini.



I always save my code in my user account in the NEPO® editor.

5. Creating a user account.

- click “user”
- click “login …”
- Click on “new” in the lower right corner on the initial registration. Note: You do not need to enter your email address.

6. Enter your username and password here.

7. Save your programs in the user account.

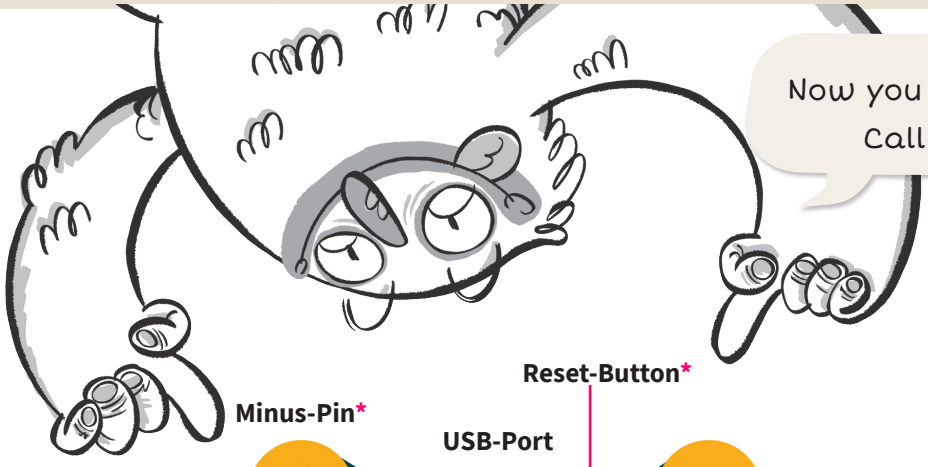
- click “edit”
- click “save as”
- assign a name for the program
- click “ok”

This allows me to keep working on my program at any time.

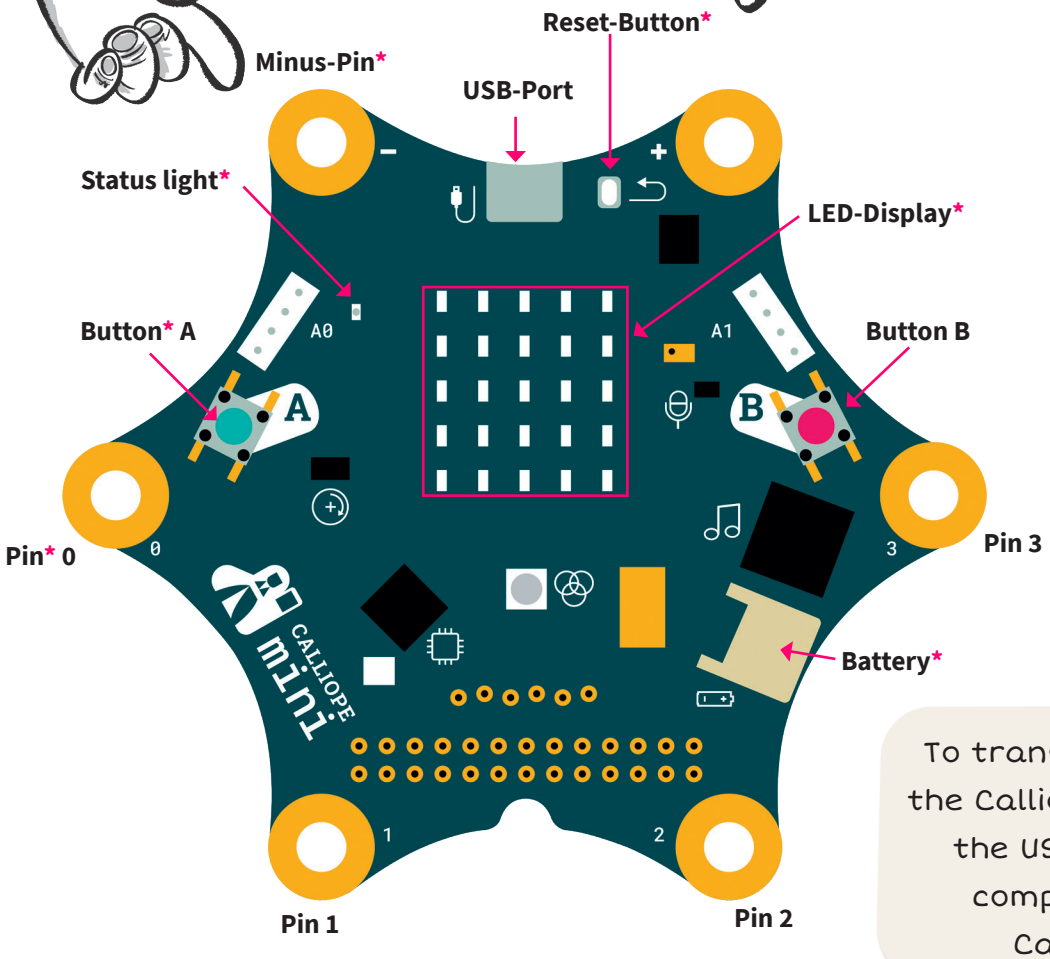
Tip: Save changes over and over again !

8. Open your saved programs.

- click “user” and login with username and password
- click “edit”
- open “my programs”
- Select the program and double-click to open it.



Now you will get to know the Calliope mini board.



To transfer the code to the Calliope mini, connect the USB cable to the computer and the Calliope mini.

9. Draw a line between Lios USB cable and the USB port* on the Calliope mini.
Note: All terms with a * are explained on page 39/40.

Now I'll transfer the code to the Calliope mini. I click on ▶ and follow the instructions in the NEPO®. editor. At the end I click OK.

As long as the Calliope mini is connected to a power source, it works everywhere.

Calliope mini as a mini piano

Lio and the mini piano

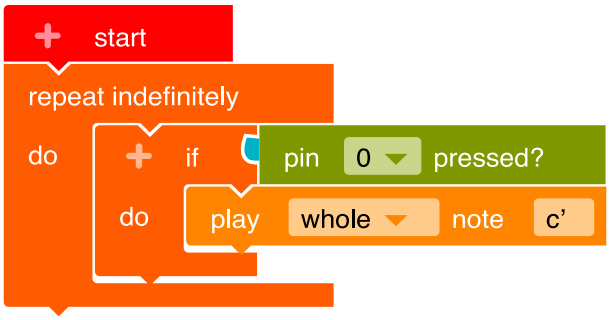
Lio wants to play music with friends during the parade.
They play guitar, trumpet and piano.
But carry a piano through the city? That‘s too difficult!
A mini piano for your pocket would be a solution.

The mini piano

Your Calliope mini must turn into a mini piano, that can produce different sounds. This mini piano works much like an electronic piano.
A sound is played when a pin and the ⊖-pin are touched and therefore the circuit is closed.

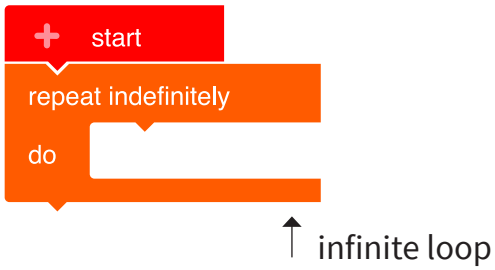
The code

As soon as Pin* 0 and the ⊖-Pin of the Calliope mini is touched, a tone is to be played. This is how the code looks like for a mini piano that can play a single note.

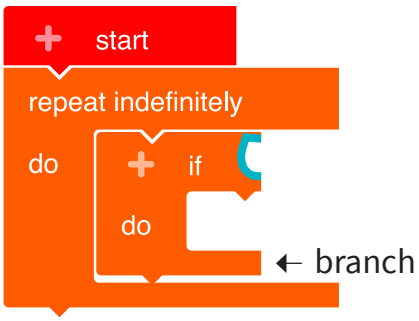


1. Program this code in the NEPO® editor (beginner) ▢
Proceed step by step.

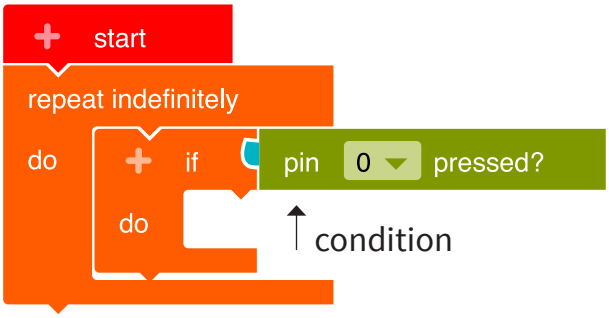
- In order for the notes to be played not just once, but over and over again, you need an infinite loop.*.
  Control → "Repeat indefinitely / do"



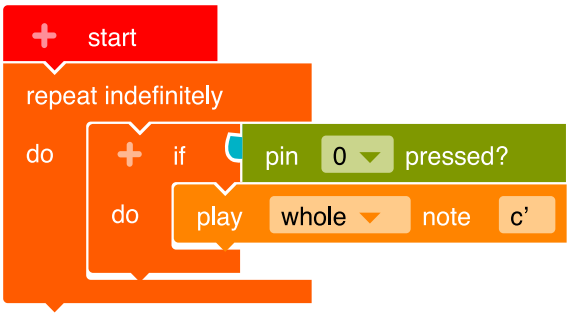
- If a pin is touched (if), a sound shall be played (do). For this you need a branch*.
  Control → "if / do"
  Insert the block into the infinite loop.



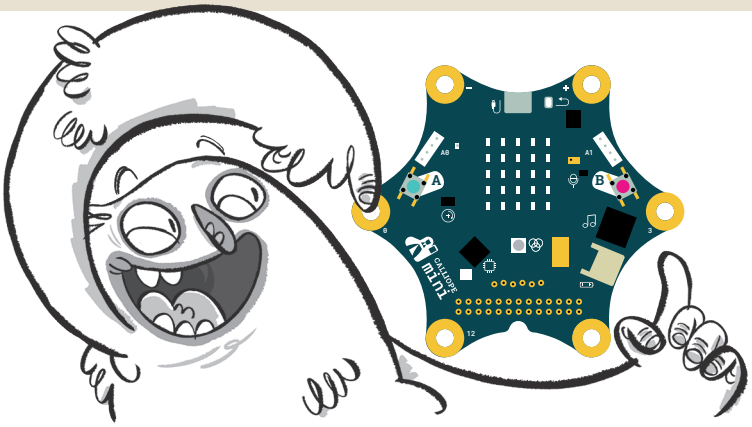
- If pin 0 is touched, a tone shall be played.
  → "pin 0 pressed?" Add the block as a condition (blue area) to the branch.



- A sound shall be played when pin 0 is touched and the condition is true.
  Action → "play whole note c'"
  Add the block as a condition to the branch.



2. ▶ Transfer the code to the Calliope mini and run the program. As long as the status light* is flashing, the transfer to the Calliope mini is not yet complete.



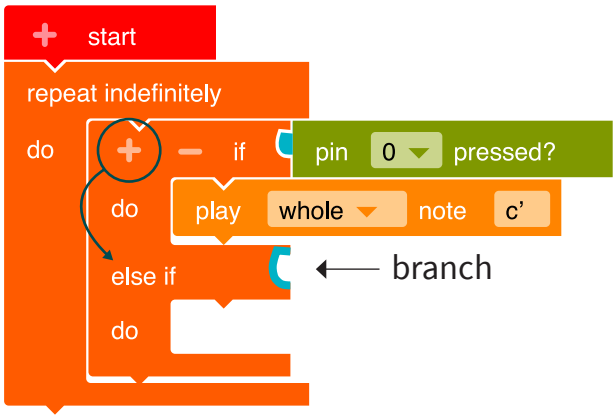
3. a) Draw where Lio must hold the other finger on the Calliope mini to produce a sound.

b) Why do you need two fingers on the Calliope mini to play a sound?

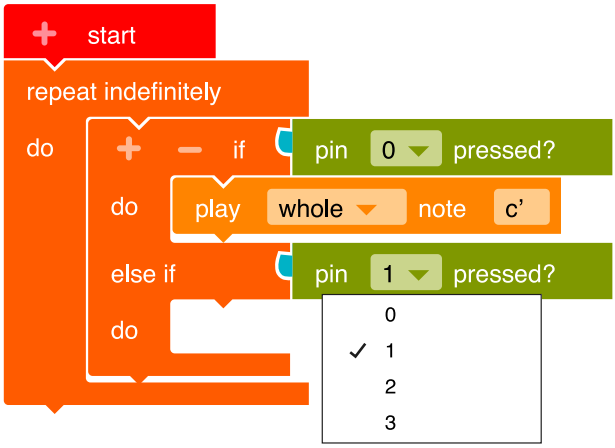
---

4. Expand the program so that the other three pins (1, 2, 3) can also play sounds. Proceed step by step.

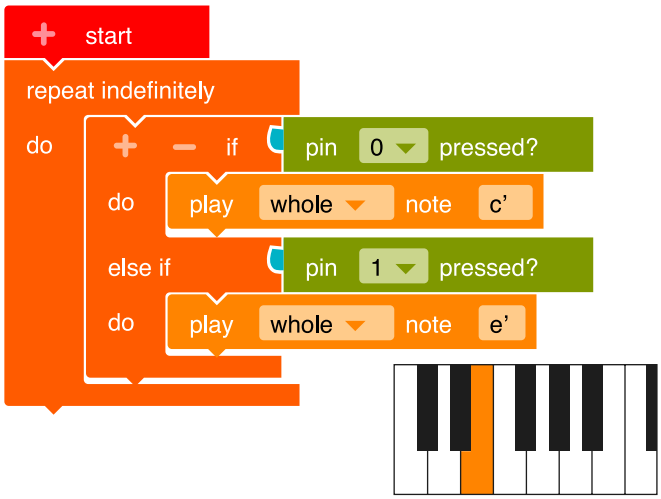
- When touching more pins (1, 2, 3), a different sound shall be played in each case.
  Each new note requires a further branch.
  To create another branch, click on the "+" next to the "if".



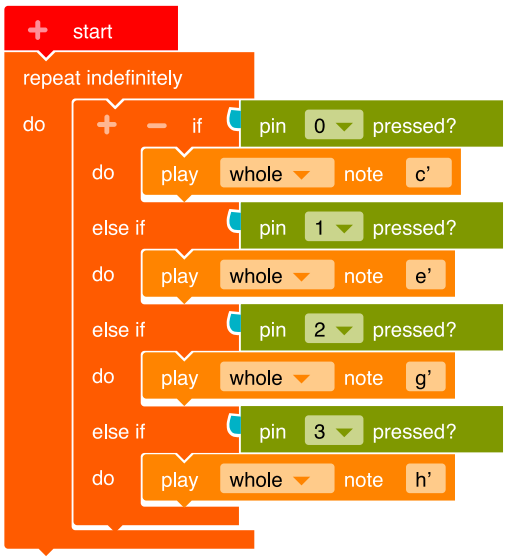
- If pin 1 is touched, another sound shall be played.
  Sensors → "pin 0 pressed?"
  Append this block to the new branch.
  Click with the mouse on the pin "0". A drop-down menu opens. Choose here pin "1".



- **Action** → "play whole note c'". Append this block to the branch. Click with the mouse on the "c'". A drop-down menu with a piano keyboard opens. For example, select the tone "e'" here.



- Continue this way and assign further tones to pin 2 and pin 3 as well.

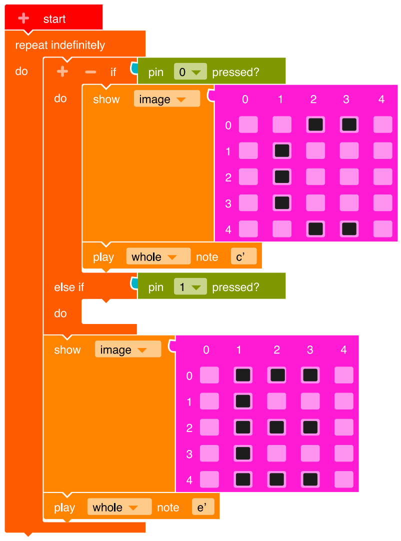


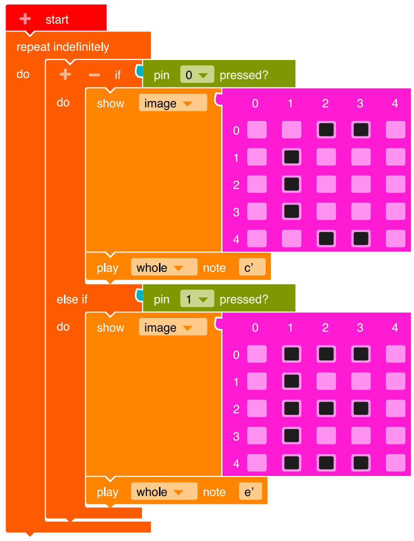
5. a) ▶ Transfer the code to the Calliope mini and run the program. Remember that you have to touch the ⊖-pin.

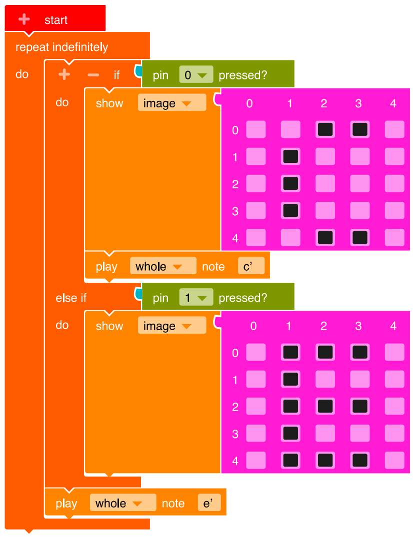
b) Try to close the circuit with two or more people.
Through how many people can the current flow?

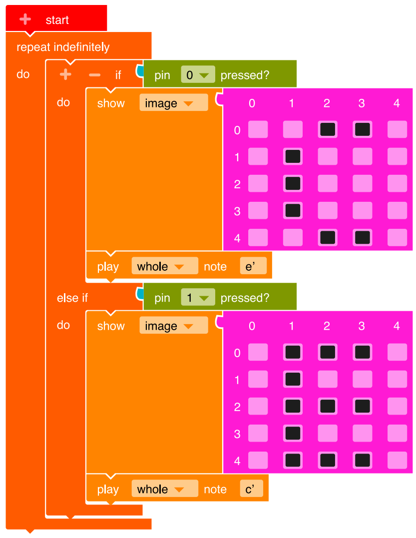
6. In order to see which tone is currently being played, the names of the sounds should be displayed on the LED screen*.

a) Mark the correct code for the display of the letters.
Only one of them is correct.
In the other codes there is an error hidden. Circle the errors.

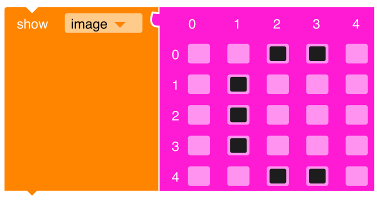
☐ 

☐ 

☐ 

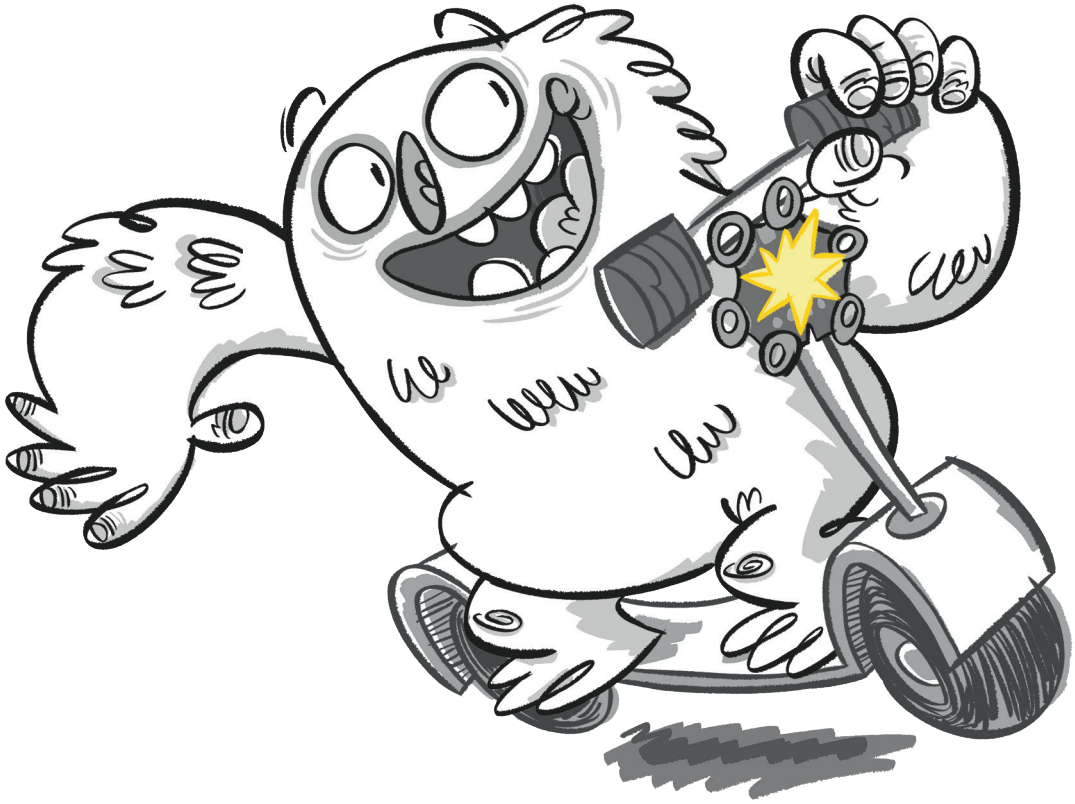
☐ 

b) Now add the letters for the according tones to your program.
Action → "show image"
In the pink block, click on the boxes you want to light up later.



7. ▶

The Calliope mini as an automatic scooter light



Lios scooter light

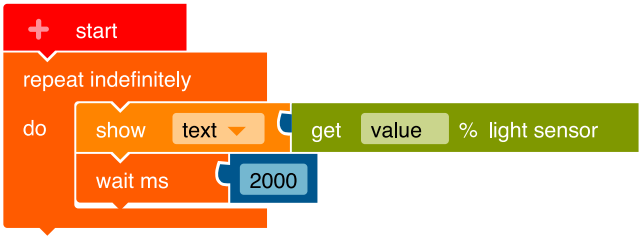
Lio is fascinated: There are cars whose lights turn on automatically in the dark.
Can Lios new scooter also get an automatic light?
That would be simply brilliant.

An automatic scooter light

The lighting of many cars today is programmed to switch on as soon as the surrounding light becomes weaker.
If the surrounding light becomes stronger again, the lighting switches off.
Similar to the lighting system in a car, the Calliope mini can also be programmed, for example as a front light for Lios scooters.

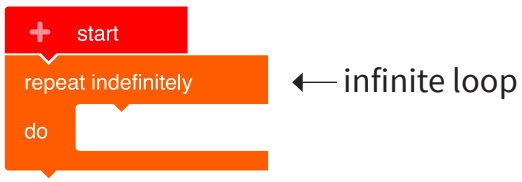
The code to measure brightness

First program a code with which the Calliope mini measures the surrounding light and displays the value on the LED screen*.

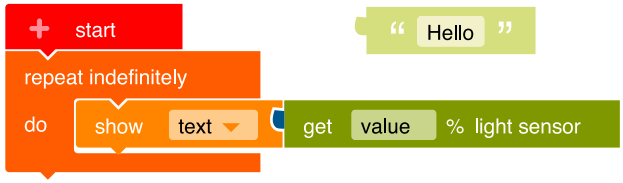


1. Program this code in the NEPO® editor ☐
Proceed step by step.

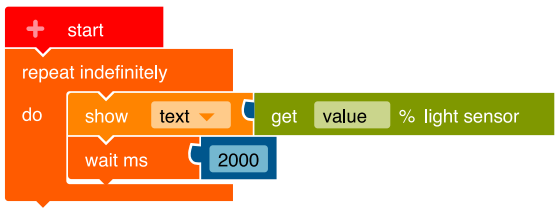
- In order for the Calliope mini to continuously measure the light intensity of its surroundings, you need an infinite loop.*.
  Control → "Repeat indefinitely / do"



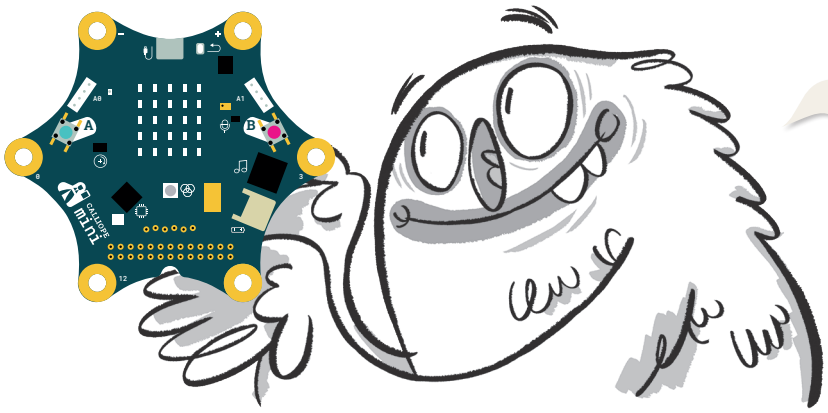
- The Calliope mini shall display the strength of the surrounding light as a numerical value.
  Action → "show text"
  Insert the block into the infinite loop.
  Remove the "Hello" block and throw it in the trash.
  Insert at this position:
  Sensors → "get value light sensor"



- The measured value should be displayed for 2000 milliseconds (ms) – which is 2 seconds.
  Control → "wait ms"
  Attach the block.
  Now click into the blue block and enter the number 2000.



2. ▶ Transfer the code to the Calliope mini and run the program.



If you touch one of the gold-coloured contacts on the Calliope mini, measurement errors may occur. That's not a problem. Then simply touch the Calliope mini in another place.

3. Try out the Calliope mini at the window, under a lamp, under a jacket, under a table …
Write down the values the Calliope mini measures.

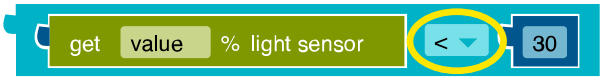
Location	measured value
at the window	
under a lamp	
under a jacket	
under a table	

The code of the automatic scooter light

The light (the RGB LED*) of the Calliope mini should:

- switch on as soon as the surrounding light becomes weaker,
- switch off as soon as the surrounding light becomes stronger again.

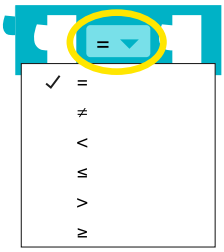
4. The Calliope mini can decide, whether its surroundings are "bright" or "dark". To do so, you have to define a threshold value with the next block:



The surrounding light is measured and translated into a number.

Threshold value. At first enter the threshold value 30 here.

It is possible to compare something with this block from the **Logic** category.



Tip: Click on the charater =. A box opens with further characters. Click on <.

Think and complete the sentence.

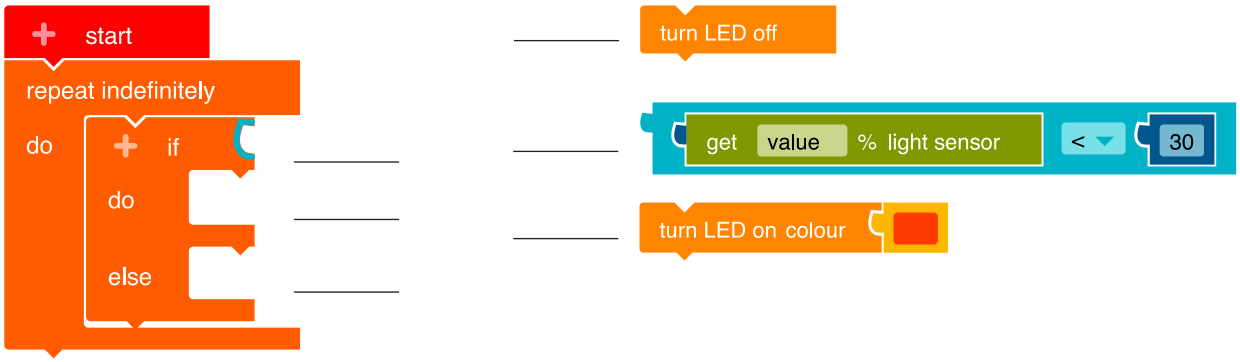
bigger, smaller

This sign **<** means that the light is always turned on

when the measured value of surrounding light is _____ than

the set threshold value.

5. a) The code of the automatic scooter light requires the following blocks. Connect the blocks on the right to the correct place in the left block.



b) Program this code in the NEPO® editor ▢.

6. ▶ Transfer the code to the Calliope mini and run the program.

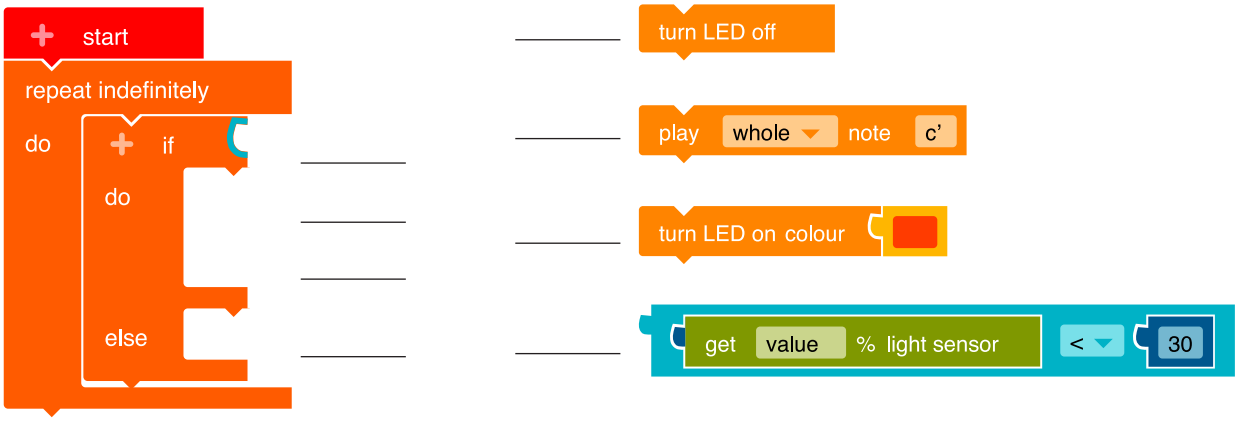
7. a) Test your program by taking the Calliope mini to places of varying brightness. You've set the threshold to 30. What happens now? Check the boxes.

b) Change the threshold value and try the Calliope mini again.

	Threshold value: 30		Threshold value: ☐	
The calliope mini …	light on	light off	light on	light off
shade with the hand				
under the table				
under a jacket				
at the window				
under a lamp				

8. Program the Calliope mini as an alarm system. It alerts you when your Calliope mini comes out of the dark schoolbag.

a) For your program you need the following blocks.
Assign the blocks to the started code. Connect.



b) Program this code in the NEPO® editor ☐.

c) ▶ Transfer the code to the Calliope mini and run the program.

The Calliope mini as a random generator

Lio doesn‘t trust chance

Lio is playing a dice game with Jack.
Jacks dice always lands on ⚅ or on the ⚂.
He wins every time.
This is making Lio angry.
Lio now wants to make a dice himself.
Therefore Lio writes a dice program.

The dice - a random generator

If you roll a dice, you get a number between ☐ and ☐.
No one can know in advance what number will appear.
It‘s called "chance". The number shown is called a "random number".
When you play a game with a dice, the dice shows random numbers.
The dice is therefore called a "random number generator".

The code

The Calliope mini can also be a random number generator:
When Button A* on the Calliope mini is pressed
a new random number shall be displayed.
The random number should be one of the dice numbers (1, 2, 3, 4, 5 or 6).

1. a) For your program you need the following blocks.
Put them together in the correct order in the NEPO® editor ☆ (expert).

Solution: see page 20

Make sure that NEPO® is in expert mode.

b) SIM Try the code in the simulator.
Compare the sequence with the program description under "**the code**".
Remember to always start the simulator with ▷

c) ▶ Transfer the code to the Calliope mini and run the program.

2. a) Check if your Calliope mini works as a random number generator:

- Press button A to display a new random number.
- Draw a dash in the table behind the random number displayed in the "Frequency" column.
- Repeat this procedure 30 times.

random number	frequency	in total
1		
2		
3		
4		
5		
6		

b) Count how often a number was rolled.
Enter your results in the "in total" column.

This is the most frequent number: _____

That‘s the least frequent number: _____

Compare your results with each other. What did you notice?
Discuss it with your classmates.

3. Use your Calliope mini as a dice. Play the ludo game or any other dice game with the other kids in your group.

A child plays with a normal dice. Is the game still fair?
Explain your answer.

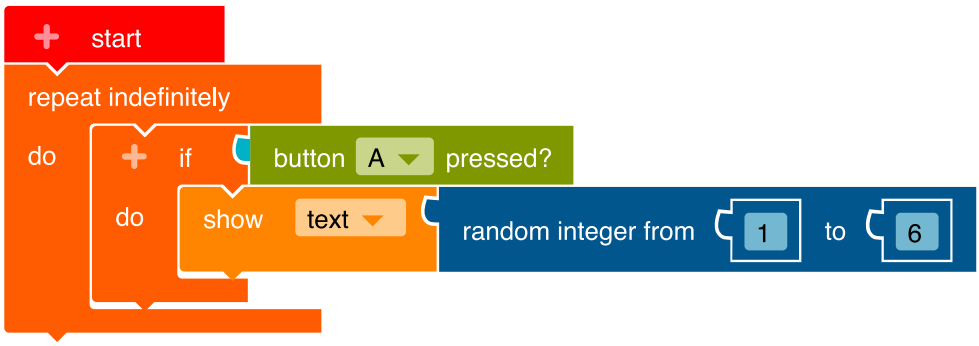
_____

_____

_____

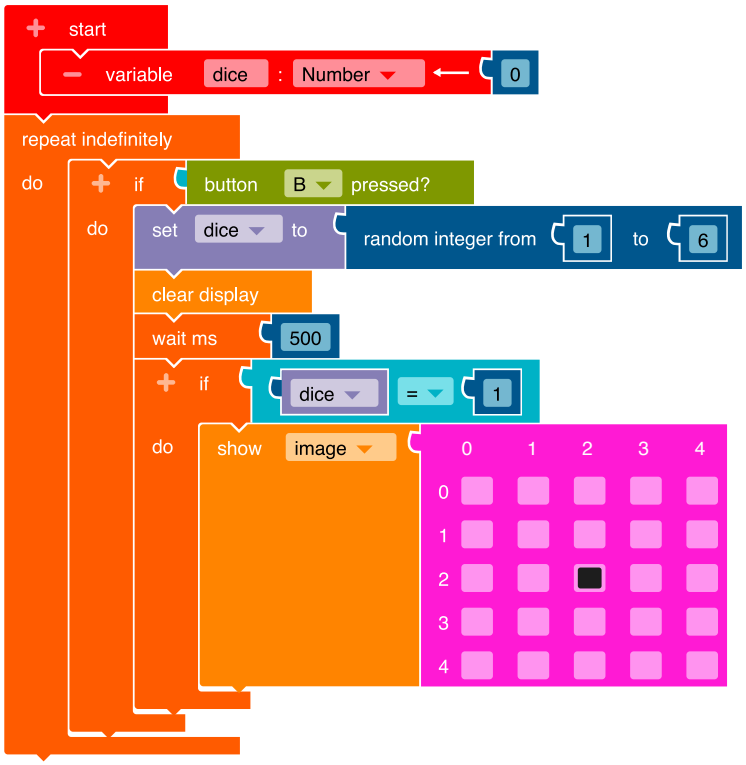
4. You want to win all the time?
Program your own "cheat dice".

a) What do you have to change? Mark the spot.
Program the "cheat dice".



b) SIM Open and ▷ start the simulator. Try the program.

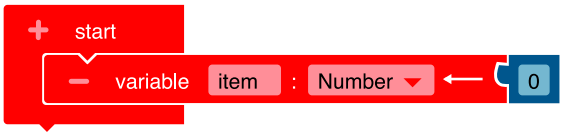
5. This code commands the Calliope mini:
If a 1 is rolled, display one dice point [•].



Program this code in the NEPO® editor [☆] (expert).
Proceed step by step.

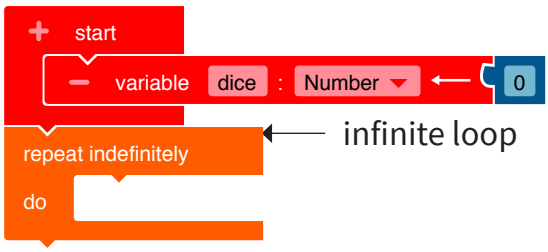
When naming variables, certain words must not be used. In case you happen to use one of the forbidden words, the editor adds a "2" to the name.

- In order to generate new numbers again and again, a variable* must be created. Click on the "+" next to "start". Click on the word "item" and type in the new variable name "dice".



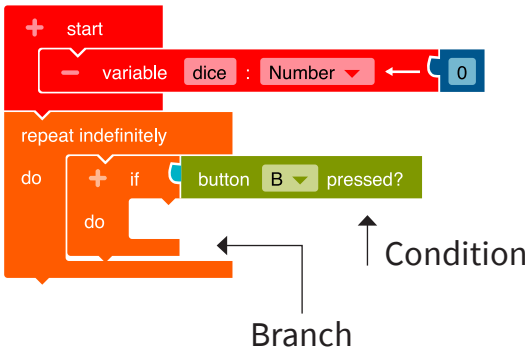
- In order for the random numbers to be generated infinitely, you need an infinite loop*.

Control ▶ Loops → "repeat indefinitly/do"



- If button B is pressed (if), a random number should be generated (do). You need a branch* .

Control ▶ Decisions → "if/do"
Insert the block into the infinite loop.



Sensors → "button B is pressed?""
Add the block as a condition (blue area) to the branch. Click on button "A" and select "B" instead.

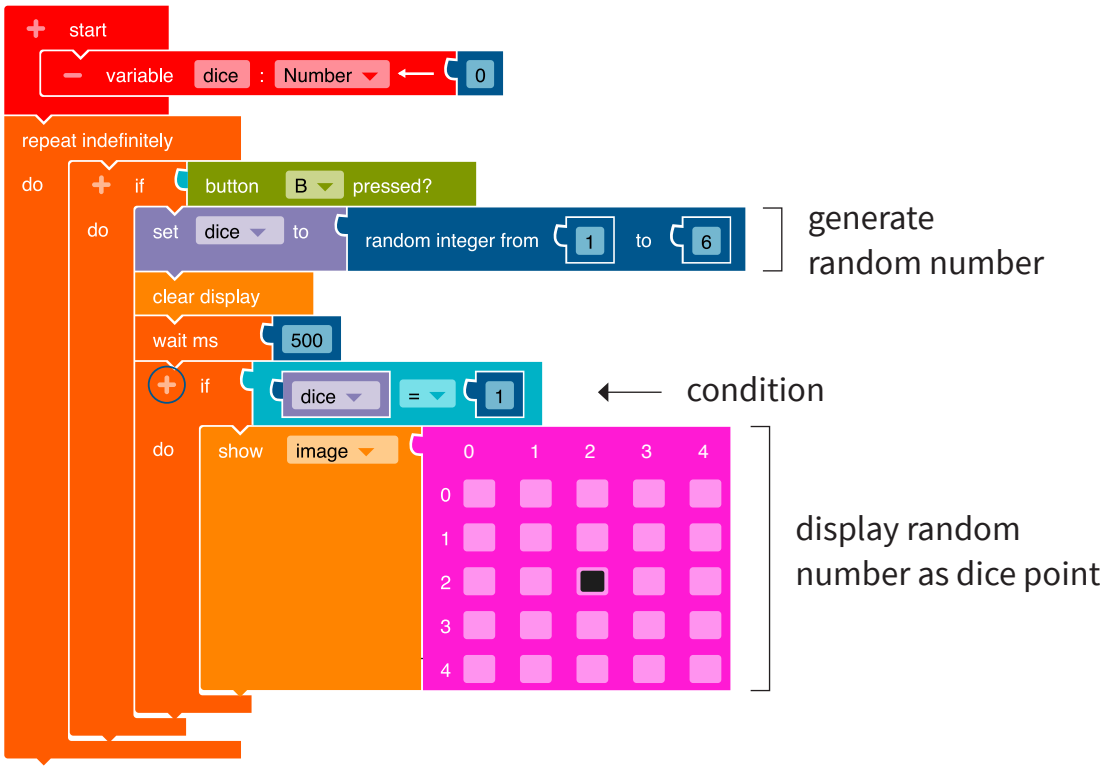
- The Calliope mini should display the random number as dice points. First the random number has to be determined.

Variable → "set dice to"
Insert the block into the branch.

Math → "random integer from 1 to 100"
Add this block and change the value 100 to the number 6.

Action ▶ Display → "clear display"

Control ▶ Wait → "wait ms"



- You are now instructed* to display the random number (if) as a pip on the LED screen*.
  You need to the following:

Control ▶ Decisions → "if/do“"
Insert the block into the branch.

Logic → 

Variable → "dice"

Math →  Change the number to a 1.
Add these three blocks as a condition to the branch.

Action ▶ Display → "show image"
In the pink block, click on a square that should flash as the dice point 1.

- If you want to display the numbers 2 to 6 as pips on the LED screen, complete your program: Click on the "+" next to the circled "if" above and repeat the previous steps.

6. ▶ Transfer the code to the Calliope mini and run the program.

Morse with the Calliope mini

Lio and the signals

Lio remembers grandpa‘s morse phone, with which he used to transmit messages. Grandpa could only talk to other people via short and long signals.
Could Lio also build a morse phone with the Calliope mini?
Hint: Please be aware that if you have successfully completed the following exercise you can no longer connect the Calliope mini via Bluetooth. In order to do so again the demo software is needed. You’ll find it here: calliope.cc/en/ble.

The morse code

About 180 years ago the American inventor Samuel Morse built an instrument, which enabled people to transmit encrypted letters electrically and later using radio technology over long distances.
Samuel Morse had the idea of translating all letters into combinations of two characters: short or/and long sounds. These sounds can also be represented as a dot for the short sound and a dash for the long sound.

The code

In our morse program, each letter is converted into a sound and a light signal. Up to four light signals stand for one letter.
On the Calliope mini, pressing the "A" button produces a dot and a short sound, and pressing the "B" button produces a dash and a long sound. Sent dots and dashes are thin, received dots and dashes are displayed boldly on the LED screen*.

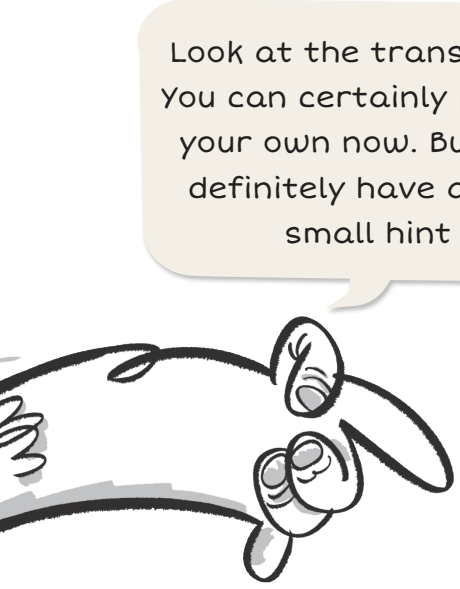
1. In the table, the letters are translated into the morse alphabet and into button sequences on the Calliope mini.
Complete the missing button sequences in the table.

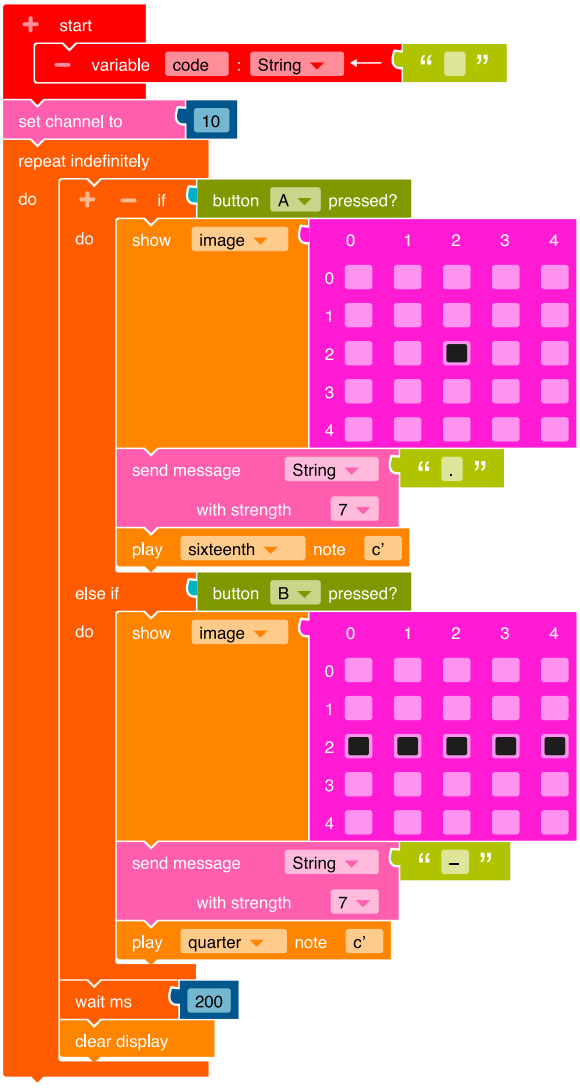
letter	morse code	button sequence on the Calliope mini
A	• —	A B
B	— • • •	B A A A
C	— • — •	
D	— • •	B A A
E	•	A
F	• • — •	
G	— — •	B B A
H	• • • •	
I	• •	A A
J	• — — —	A B B B
K	— • —	B A B
L	• — • •	
M	— —	B B
N	— •	
O	— — —	
P	• — — •	
Q	— — • —	B B A B
R	• — •	A B A
S	• • •	
T	—	B
U	• • —	
V	• • • —	A A A B
W	• — —	
X	— • • —	B A A B
Y	— • — —	
Z	— — • •	

2. Enter the morse code for the word "L U N C H" in the empty field below. Use a vertical line between the letters to indicate a pause.

	L	U	N	C	H
morse code	— — |				
button sequence	B B |				

3. Using the morse program on page 26, two or more Calliope mini can send and receive messages between them, as long as they have programed the same radio channel. Each Calliope mini requires a program with a transmitter and a receiver module. Program the code for the transmitting and receiving section in the NEPO® editor ☆ . Proceed step by step.





Hint:

Transmitting part of the code

The block "send message" with the text block “ ” executes the following:

If button* A is pressed this “ . ” will be sent and

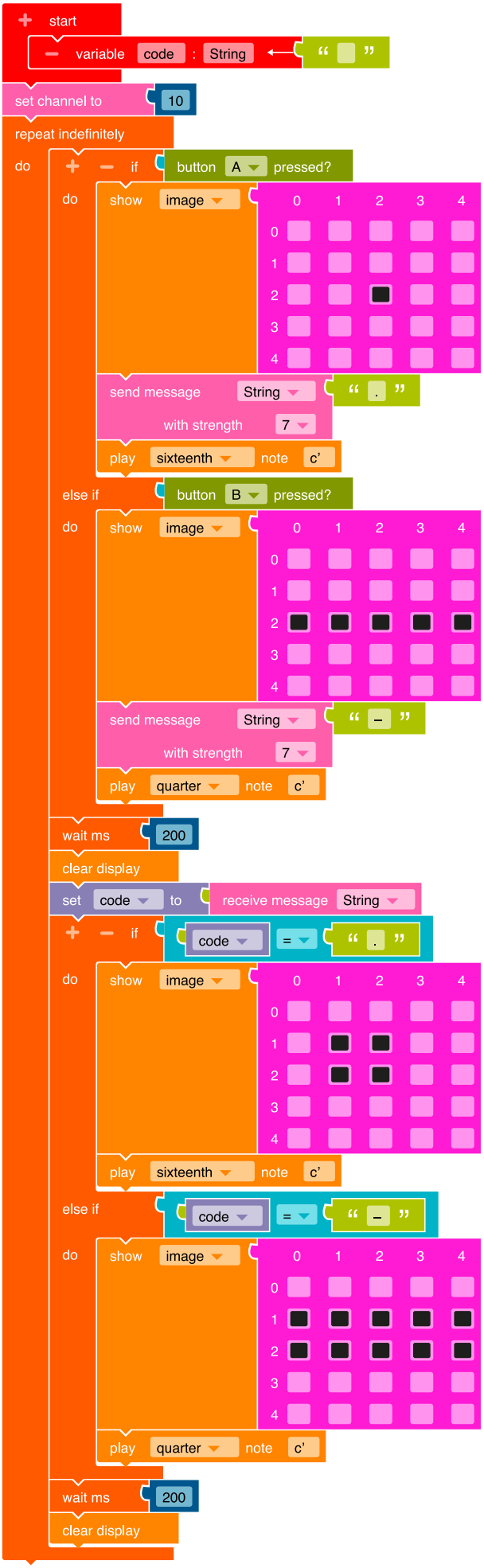
if button B is pressed, this “ – ” will be sent.

Receiving part of the program

This part of the program displays the messages sent by another Calliope mini. If the Calliope mini receives the message “ . ” it displays this on the LED screen and plays a short sound. If the Calliope mini receives the message “ – ” it displays this on the LED screen and plays a long sound.

4. ▶ Transfer the code to the Calliope mini and run the program.

5. Send single letters with your Calliope mini.
Your other classmates must use their Calliope mini and the morse alphabet to identify the letters.
In order to identify them, you have to take a short break between the sent letters.



6. Think of a word and write down the morse code for the word.
Then send the code.

Notice when sending:
- It should be a short word with no more than five letters.
- Pay attention to the breakes between the letters.
- Agree with the other students who is allowed to send.
  Only one student can send at a time.
  The other students can receive the morse characters.

Transmitted word:

letters					
morse code					
button sequence on the Calliope mini					

Notice when receiving:
- Work in groups of two: One student announces the morse code received, another student writes it down.
  You then translate the signs with the morse code table on page 24.
- Remember to pay attention to the breaks between the letters.

Received word:

morse code					
letters					

Received word::

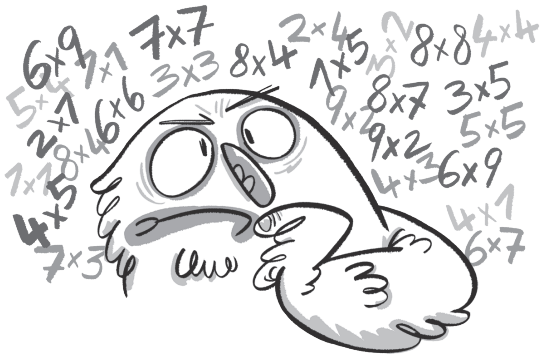
morse code					
letters					

If transmitting and receiving with the short words worked, you can also transmit and receive longer words or even whole sentences.

The Calliope mini as multiplication tables trainer

Lio is practicing for the upcoming math exam

Lios math teacher has announced a class test for next week. It is going to be about multiplication. Lio has already solved all tasks in the exercise book. But that is not enough. Lio wants to write a program that keeps creating new exercises.



The calculator

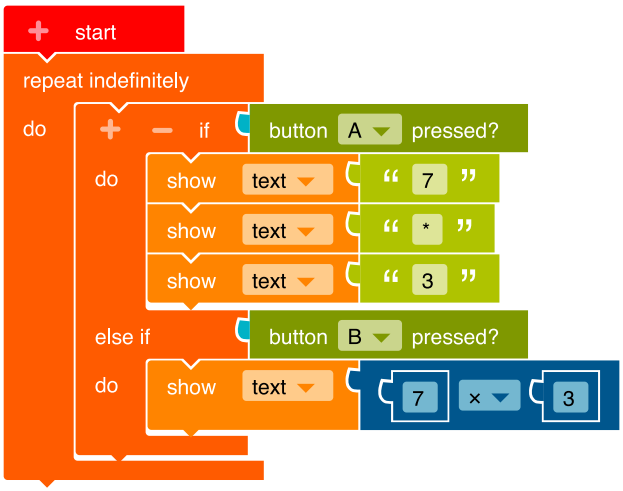
Computers calculate the most difficult math tasks within the blink of an eye. What a computer shall calculate must first be programmed. If the computer is given the correct program, it can even create math tasks for people.

The code

The program shows a single calculation task.

Describe what the Calliope mini generates if you press button A or B.
Go over the program commands carefully.

If button A is pressed, then ...



1. a) Program this code in the NEPO® editor ▢.

b) Change the numbers to generate a new multiplication task.

c) SIM Open and ▷ start the simulator. Try the program.

d) Write down what the Calliope mini displays:

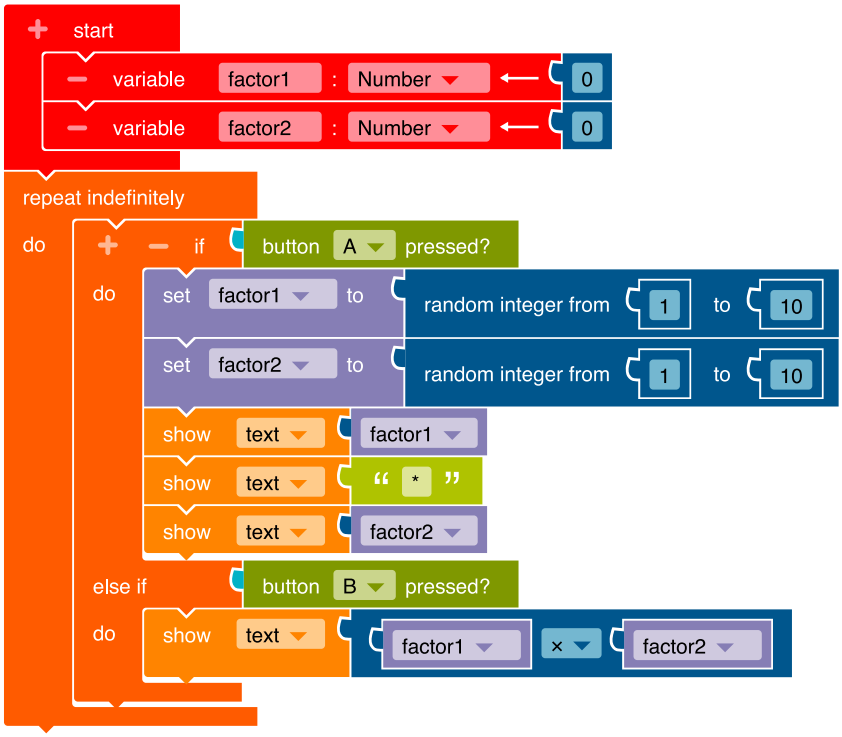
if button A is pressed: _____

if button B is pressed: _____

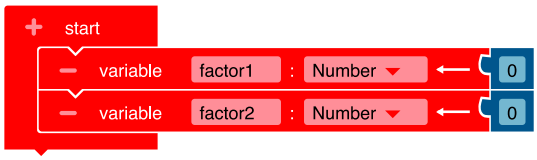
2. The program shown on page 28 always generates the same calculation task.

Now create a program that does the following :
If button A is pressed, generate a multiplication with random numbers. If button B is pressed, display the result of the task.

This is the code of the finished program:



Program this code in the NEPO® editor ☆ .
Proceed step by step.



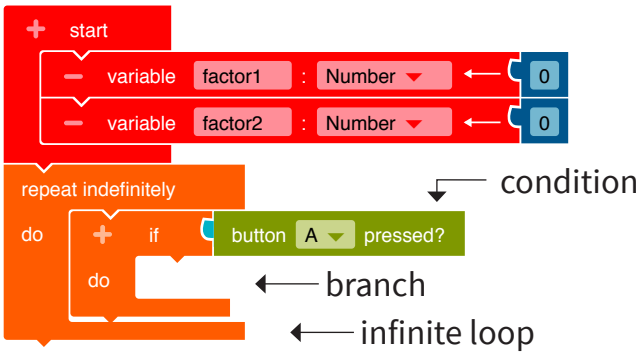
- In order to generate new tasks again and again, two variables* must be created in the beginning. Click on the "+" next to "start". Click on the word "item" and type in the new variable name "factor1".
  Repeat both steps.
  Give the new variable the name "factor2".

- To create multiplication tasks as many times as you want, you need an infinite loop*.

Control ▶ Loops → "Repeat indefinitely/do"

If button A is pressed (if), a multiplaction task shall be displayed (do):
For this you need a branch*.

Control ▶ Decisions → "if/do"
Sensors → "button A pressed?"

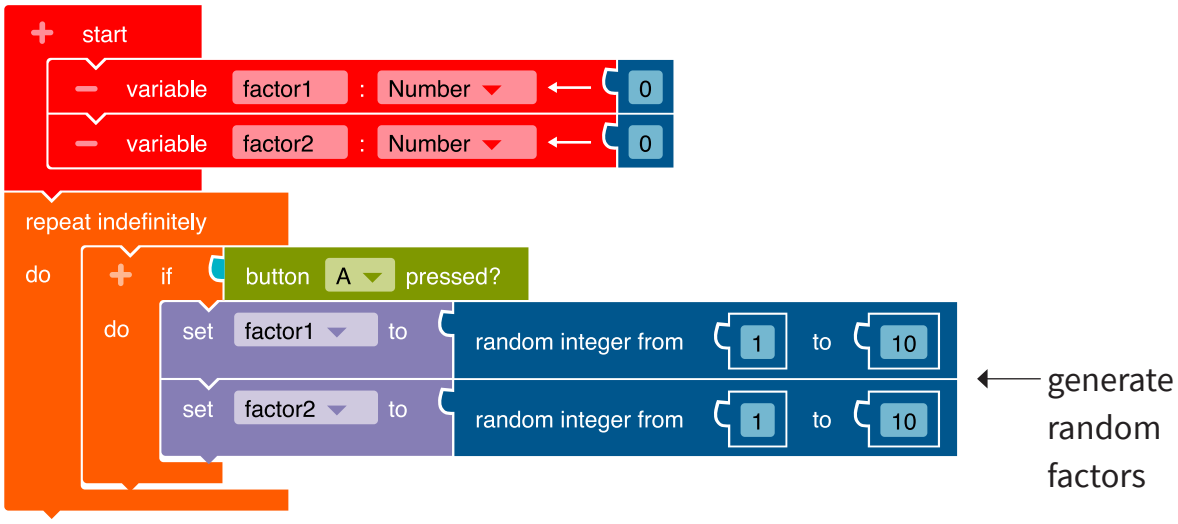


- In order to create the factors for the task, random numbers must be generated. The numbers are stored in the variables "factor1" and "factor2".

Variables → "set factor1 to"
Add the block to the branch.

Math → "random integer from 1 to 100"
Add this block and change the value 100 to the number 10.
By doing so, you determine that the random number cannot be greater than 10.

Repeat the two steps for the variable"factor2".



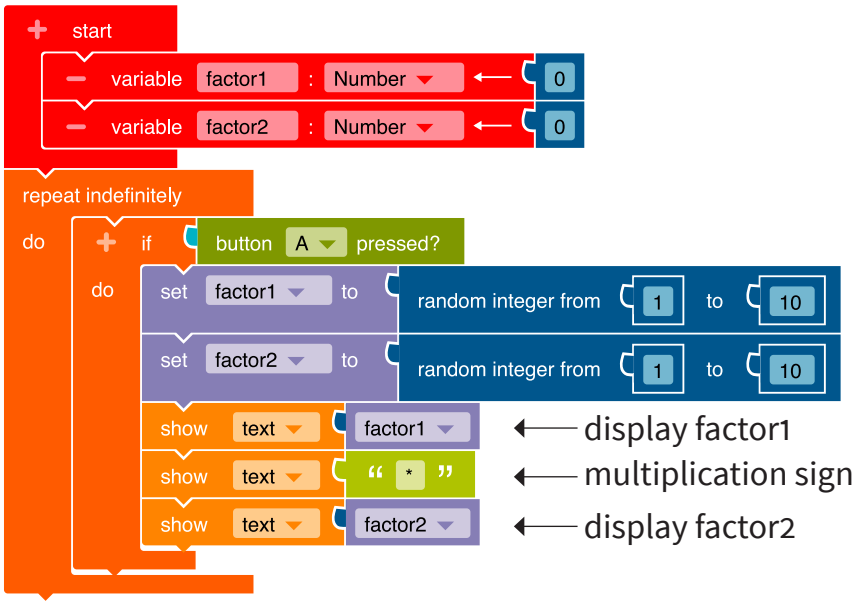
- The Calliope mini should display the task on the LED screen*
  Program the output of the first factor.

  Action ▶ Display → "show text"
  Delete the block "Hello".
  Variables → "factor1"
  Attach the block to the "show text" block.

SIM Open and ▷ start the simulator. Try the programm.

- In the next step, the multiplication sign should be displayed.

  Action ▶ Display → "show text"
  Click on the word "Hello" and type in an asterisk (*) as a multiplication sign.

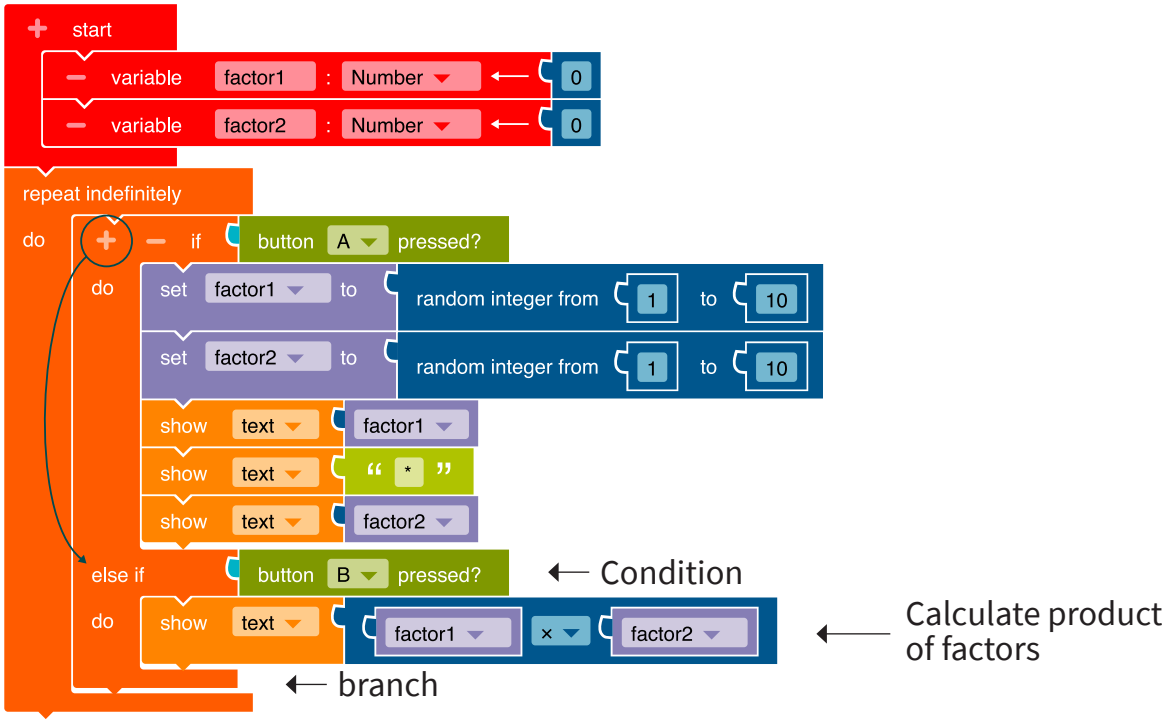


- Now program the output of the second factor.
  Proceed as you did for the first factor. But use the variable "factor2".

SIM Open and ▷ start the simulator. Try the program.

• If button B is pressed (if), the result of the multiplication task should be displayed (do).
Click on the "+" next to "if". Another branch appears.

Sensors → "button A pressed?"
Add the block as a condition to the branch.
Click "A" and select "B".



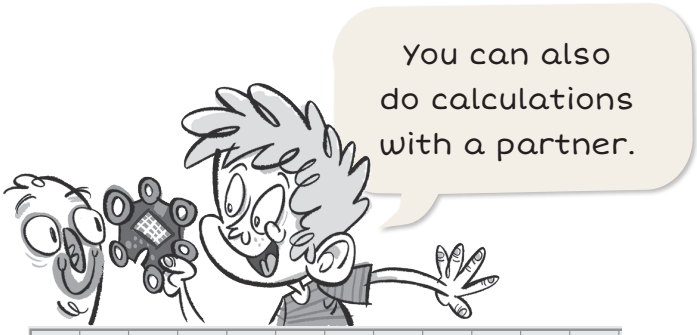
• The Calliope mini calculates the product of the two randomly selected factors and displays it on the LED screen.

Action ▶ Display → "show text"
Add the block to the second branch. Delete the block "Hello".

Math → " + "
Change the arithmetic character to a multiplication character by clicking the "+" and selecting the "x" in the drop-down menu.

Add the variables:
Variable → "factor1" and "factor2"
Insert the blocks to the left and right of the "x" sign.

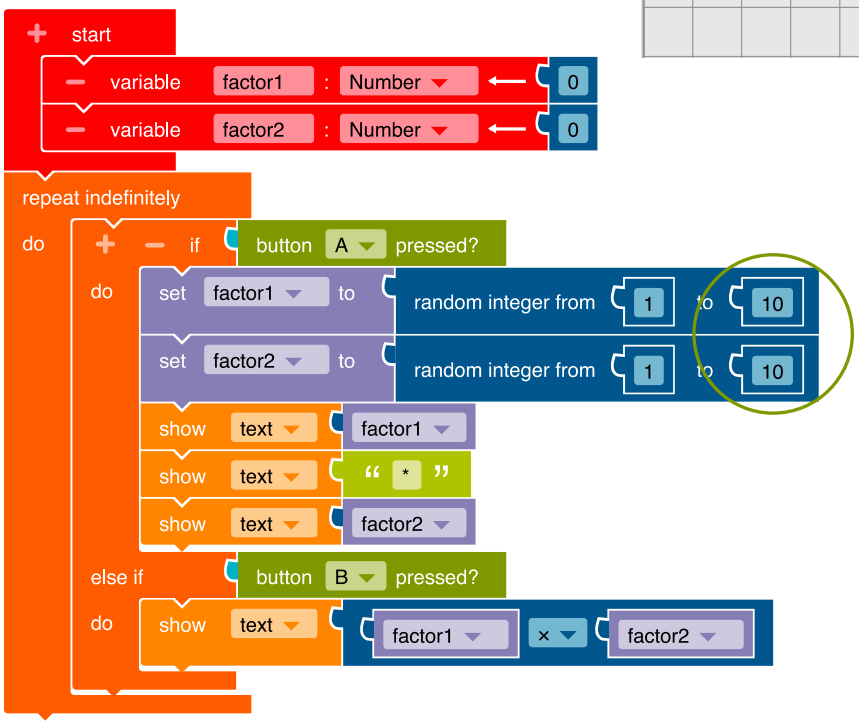
3. a) ▶ Transfer the code to the Calliope mini and run the program.



b) Press button A on the Calliope mini to get a new task. Solve the task. Check your solution by pressing button B.

4. You can change the code by following these steps:

a) Use the Calliope mini to calculate tasks of the great multiplication table. The green circle shows you, where you need to do changes.

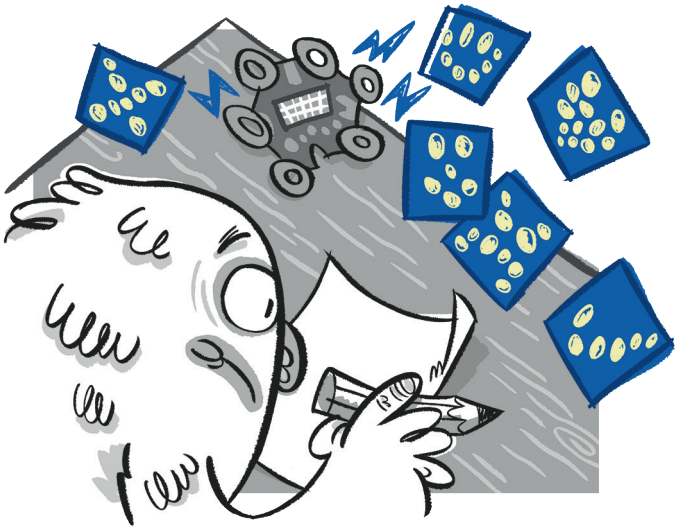


b) Calculate addition problems with the Calliope mini.
Mark the blocks in red where you need to change something.

c) SIM Open and ▷ start the simulator. Try the programm.

5. ▶ Transfer the changed code to the Calliope mini and run the program.

Create image impulses and stimulus words with the Calliope mini



Lio is writing a story

Lio is a big fan of fantasy stories since anything can happen. Lio wants to write a story himself, but has not any ideas yet. That is why Lio is thinking about how the Calliope mini can be programmed as a source of ideas.

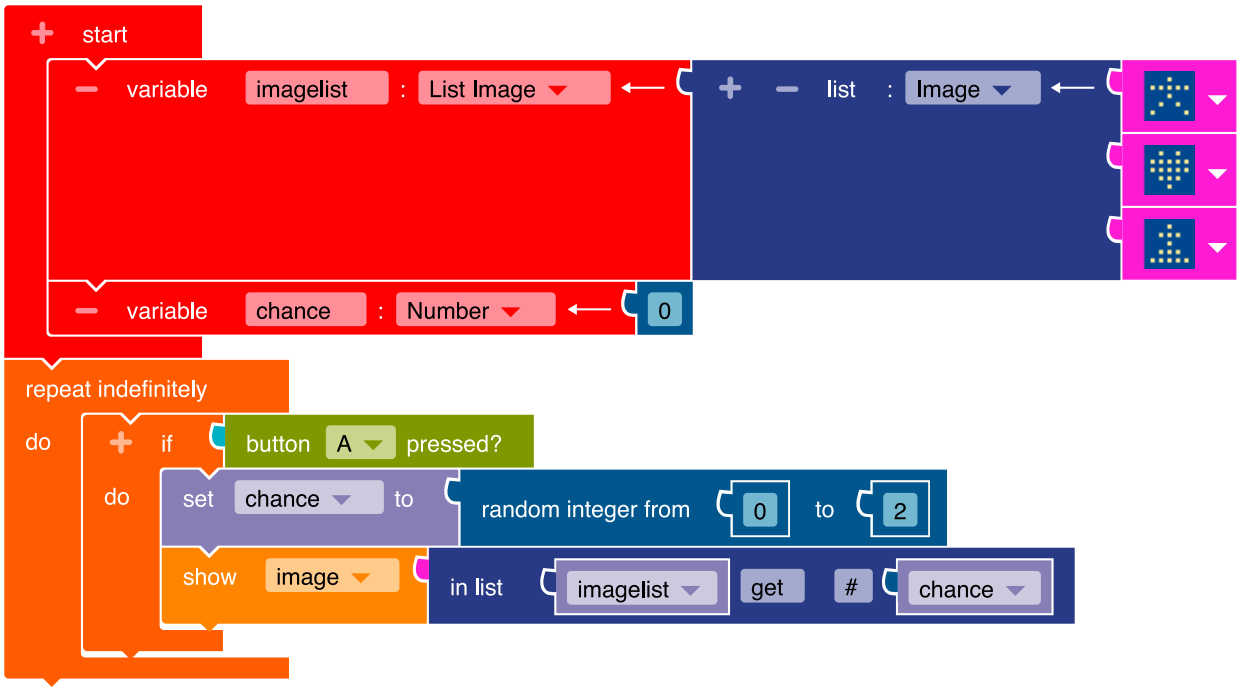
The idea generator

The Calliope mini can be programmed to display random images and display their names.

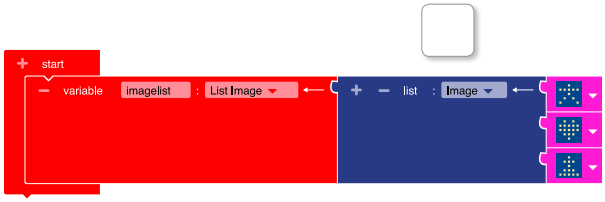
The code

The program displays random images when a button is pressed.

1. a) Take a good look at the program.



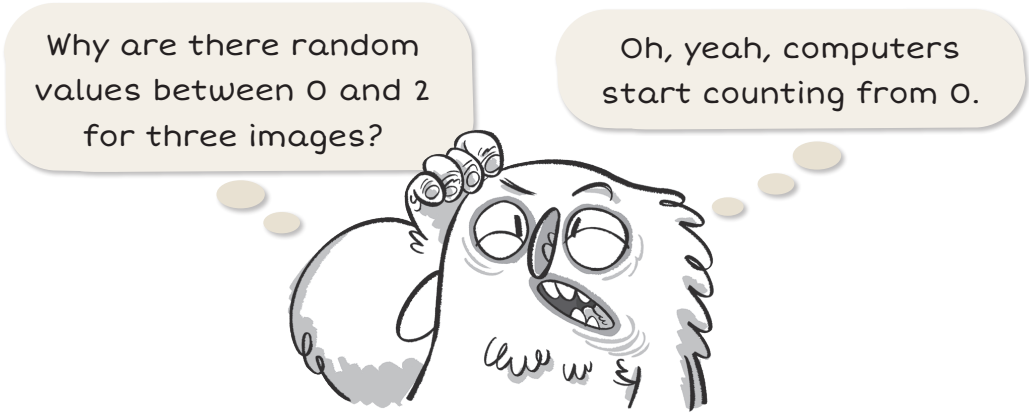
b) For this program you need the following blocks.
Write the numbers of the descriptions to the matching blocks.

block	description
 ☐	1 request pressed button (input)
 ☐	2 show an image (output)
 ☐	3 randomly select an image from a list of images in a fixed order
 ☐	4 repetition (infinite loop*)
 ☐	5 generate random numbers in a specified range
 ☐	6 creating the "chance" variable
 ☐	7 creating an image list with a fixed order

0 

1 

2 



2. How many images are there if the random value is between 0 and 5?

☐

3. Identify the order of the work steps for the program
Decide whether a work step is executed only once or repeatedly.

You can choose from the following steps:

request pressed button (input)

show an image (output)

Randomly select an image from a list of images in a fixed order

repetition (infinite loop)

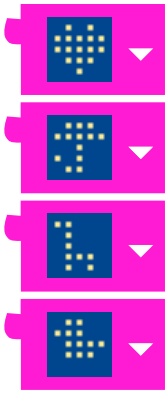
generate random

~~creating an image list with a fixed order~~

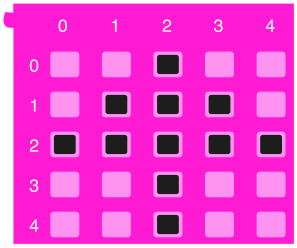
creating the "chance" variable

	work steps	once	repeatedly
1.	creating an image list with a fixed order	X	
2.			
3.			
4.			
5.			
6.			
7.			

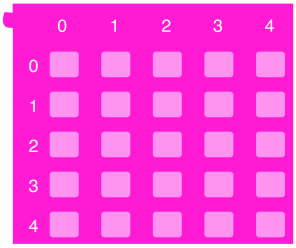
4. The NEPO® editor has some images to choose from.
Write down what the pictures show.

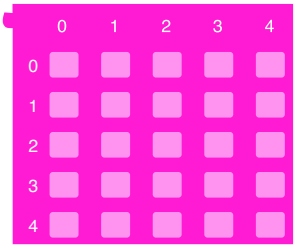


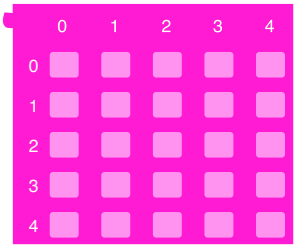
5. Create your own images for your program. Draw them into the LED screens by coloring the corresponding boxes.

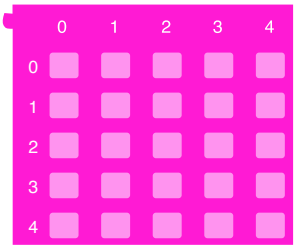


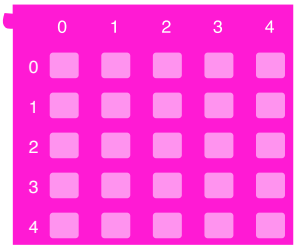
TREE











6. a) Program the code from page 34 in the NEPO® editor ☆ .
Proceed step by step. If you want, you can exchange your own images for the ones already provided.

b) SIM Open and ▷ start the simulator. Try your program.

7. Now make up a random story using the displayed images.
Tell it to one of your classmates.

8. Add the name to each image from the image list.
Three program blocks must be added for this.

- Click on the "+" next to "start".
  A new block appears.
  Click on the word "item" and give the variable* the name "textlist".
  Click on the "Number" and select "List String" aus.
  Now enter the name for each image in the green text fields
  in the order of the "imagelist".

• We need a break to separate the image from the name.

Control ▶ Wait → "wait ms"

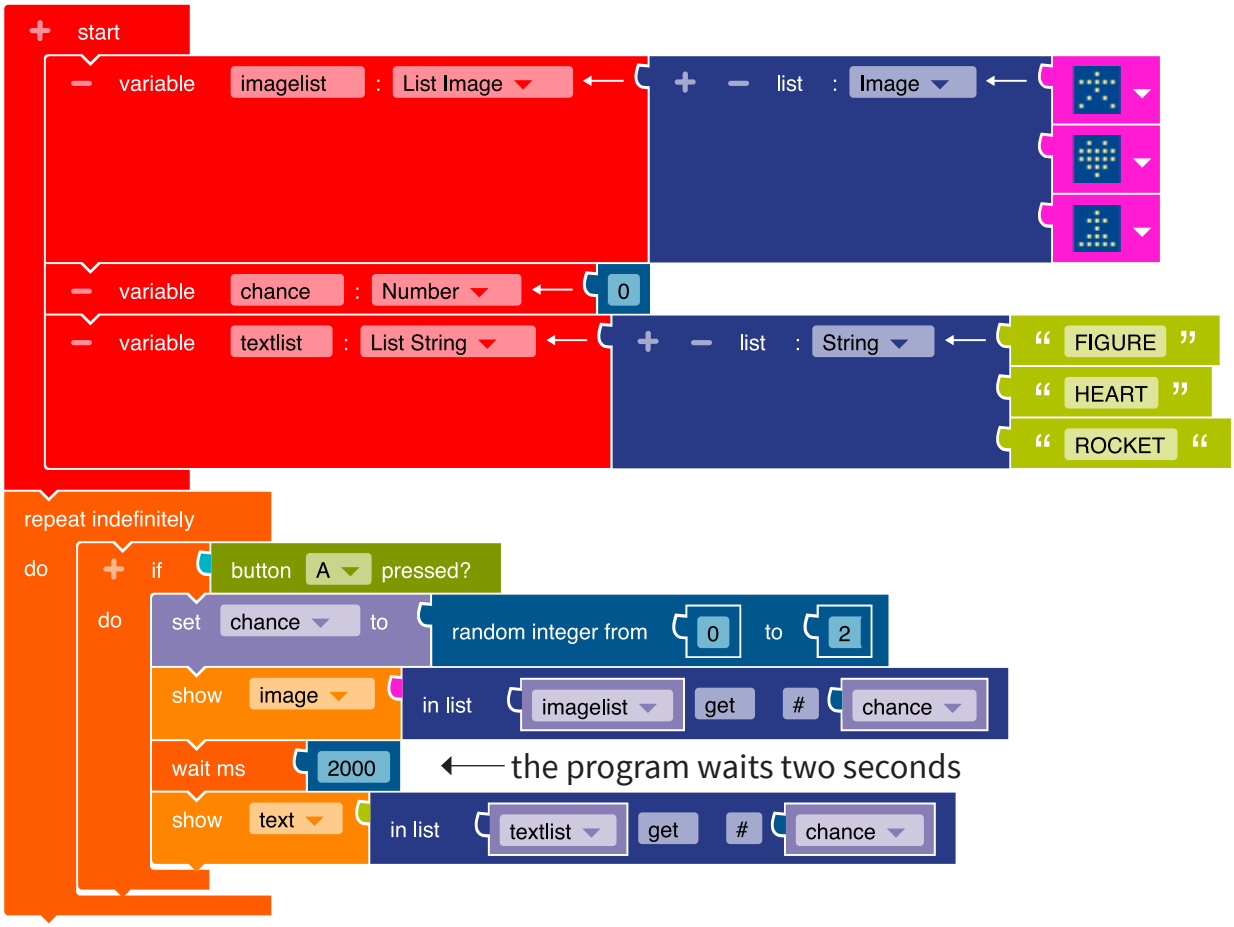
Click in the blue bock and enter the number 2000.

• The word should match the image from the related image list.

"chance" "image list" "text list"

At which position on the picture list should Lio‘s smiley be?

Remember how the computers count. _____



9. ▶ Transfer the code to the Calliope mini and run the program.

The small coding encyclopedia

instruction (= command)

When you receive an instruction, you can execute it. For example:
"Hang the wet socks on the clothes horse to dry."
The same is true for the computer. It executes instructions that clearly describe what it should do. A code/program is built from instructions.

loop with a condition

A loop allows a sequence of instructions to be executed over and over again.
For example:
"Hang up socks as long as there‘s laundry in the basket."
The **loop** is: "Hang up socks as long as (repeat) …"
The **condition** of the loop is: "Is there still laundry in the basket?"
Answer: "Yes!"
In the loop, **four instructions** are executed one after the other:
1. Take a wet piece of laundry
2. Hang the piece of laundry on the clothes horse
3. Use two clothespins
4. Fasten the piece of laundry with the clamps

If the answer to the condition "Is there still laundry in the basket?" is "No!", the program continues behind the loop:
"Bring the basket to the bathroom."

infintite loop

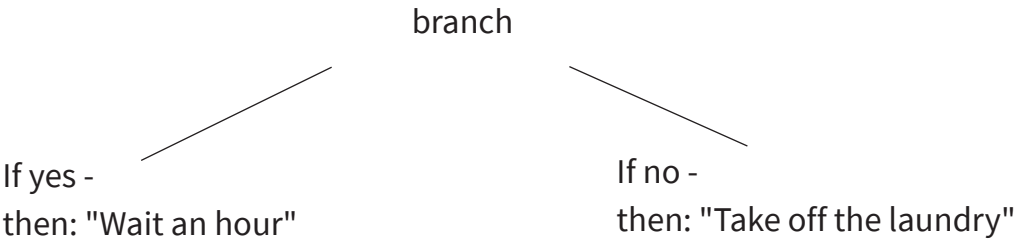
An infinite loop **has no condition** und and will run until the Calliope mini is switched off.

variable

A variable is a container for a specific value (number, word, etc.), image or something else that is set at the beginning of the program. Each variable needs a unique name and you have to decide if the variable should store a number, a word (→ string), an image or something else.

branch with a condition

Every branch in a program needs a condition.
The condition defines the next instructions in the program.
There are two ways of doing this, for example:
Condition: "Is the laundry on the clothes horse still wet?"



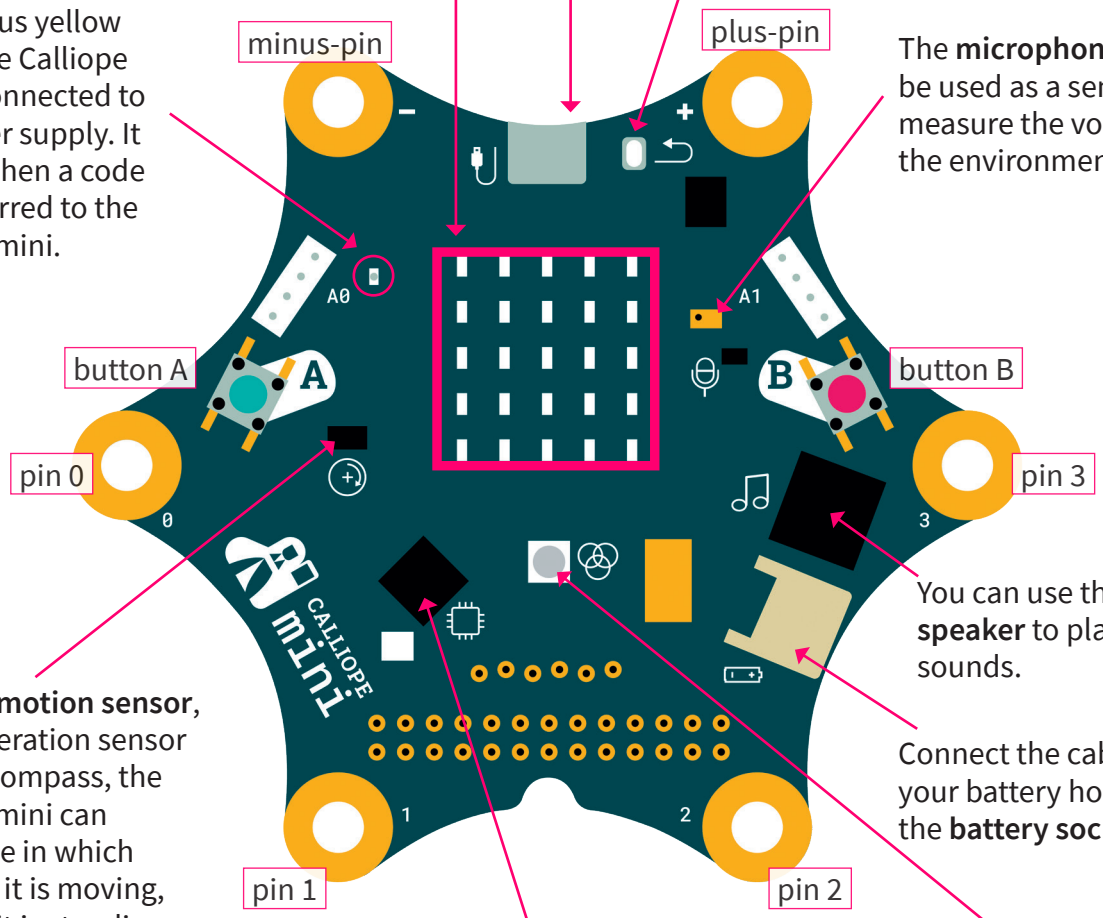
By pressing the **A** and **B buttons**, you enter data so that the Calliope mini executes the programed commands.

Plug a micro-**USB** cable into the USB port to connect the Calliope mini to a computer.

The **LED screen** is a 5 x 5 grid with red LEDs.

Use the reset button to restart the program on the Calliope mini.

The **status light** indicates with a continuous yellow light if the Calliope mini is connected to the power supply. It flashes when a code is transferred to the Calliope mini.



The **microphone** can be used as a sensor to measure the volume in the environment.

With the **motion sensor**, the acceleration sensor and the compass, the Calliope mini can determine in which direction it is moving, whether it is standing on its head, whether it is lying, whether it is upright, in which direction it is turned or whether it is not moving.

You can use the **speaker** to play sounds.

Connect the cable of your battery holder to the **battery socket**.

The **processor** is the heart of the Calliope mini! The processor combines all functions and processes all information.

The **RGB-LED** can flash brightly and light up in different colours.

By touching the **pins** 0, 1, 2, or 3, you can also enter information for the Calliope mini to execute commands. You must simultaneously use the other hand and touch the minus pin (-).

CALLIOPE

Calliope mini is a product of the Calliope gGmbH

With the Calliope mini microcontroller, school children should be able to gain playful access to the digital world. Because digital knowledge is essential if we want to actively participate in society and help shape it. The Calliope team is made up of experts from the IT and education sectors who work together on an interdisciplinary basis.

More information about the initiative can be found at calliope.cc

OPEN ROBERTA

The Open Roberta Lab is a freely available cloud-based programming platform on which children, teenagers and adults can program microcontrollers and robots - even without previous knowledge. Students bring the Calliope mini to life intuitively via drag and drop using the NEPO® graphical programming language. Open Roberta® is a technological open source development of the Fraunhofer IAIS initiative „Roberta® - Learning with Robots“, which has been promoting digital education in Germany since 2002. The development of Open Roberta® has been supported by Google.org since 2014. Roberta®, Open Roberta® and NEPO® are registered trademarks of Fraunhofer-Gesellschaft für angewandte Forschung e. V.

Click here to go to the Open Roberta Lab: lab.open-roberta.org

Coding with the Calliope mini

With the workbook “Coding with the Calliope mini” you learn about six programming examples for the subjects English, Science and Math from a new angle:

- Is morse code suitable to transmit secret messages?
- How does it work that a scooter light turns on automatically in darkness?
- Does anybody notice that I have programmed a cheat dice?

Step by step you will learn what coding is, what an editor is needed for, how to program the Calliope mini board and how to use it. You’ll code your first programs and will get to know the basic principles of programing.

Let‘s go - give it a try and get creative!

CALLIOPE.CC

CALLIOPE