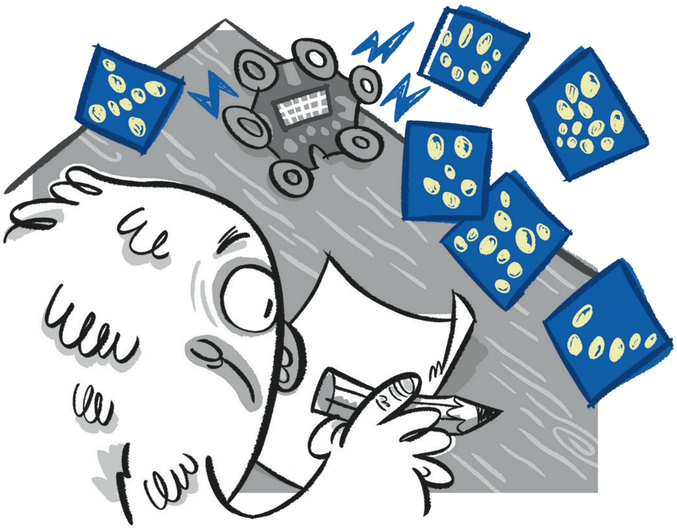
Create image impulses and stimulus words with the Calliope mini



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.

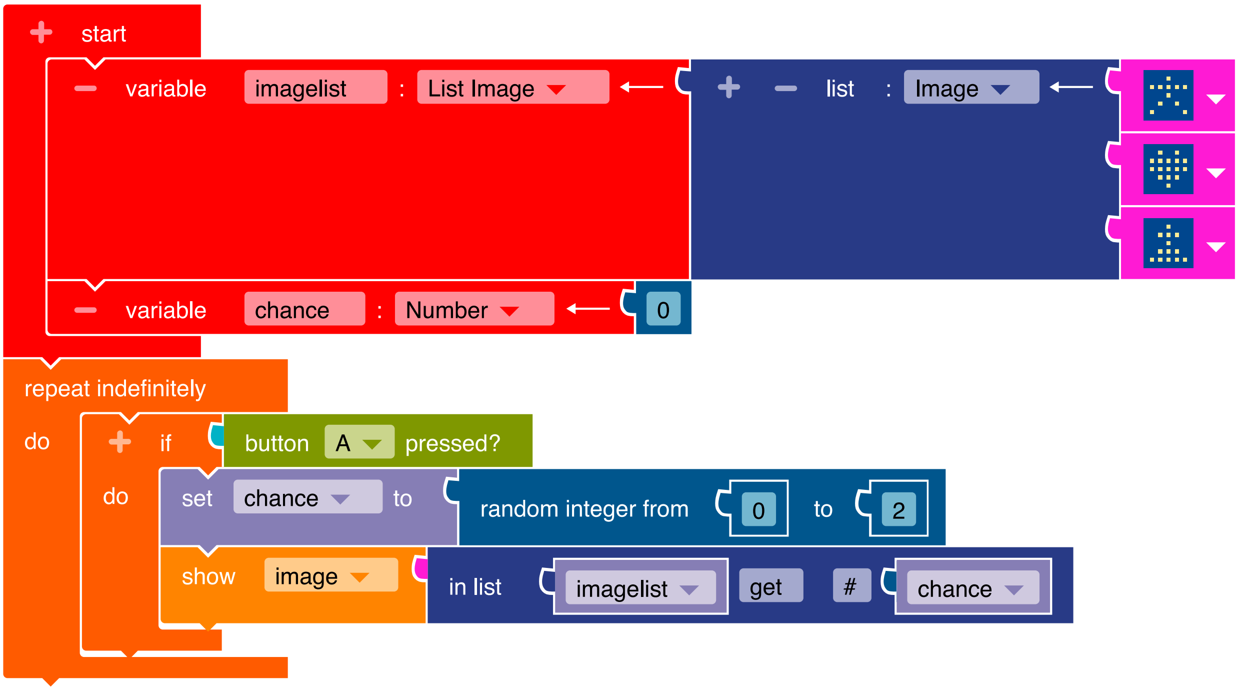
The idea generator

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.

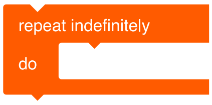
1. a) Take a good look at the program.



```
+ start
    − variable  imagelist : List Image ←  + − list : Image ←
    − variable  chance : Number ←  0
repeat indefinitely
do  + if  button A pressed?
    do  set chance to  random integer from 0 to 2
        show image  in list  imagelist get # chance
```

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

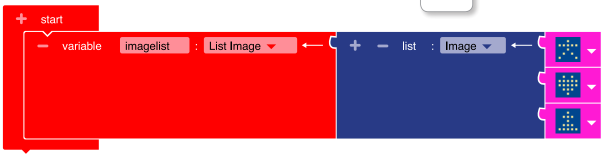
block



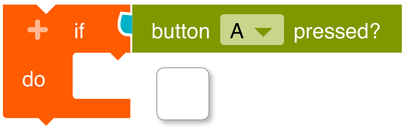
```
repeat indefinitely
do
```

□

□



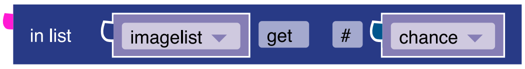
```
+ start
  − variable  imagelist : List Image ←  + − list : Image ←
```



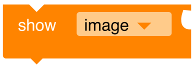
```
+ if   button A pressed?
do
```

□

□



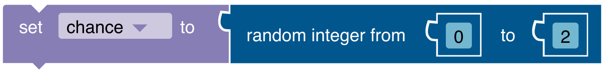
```
in list  imagelist  get  #  chance
```



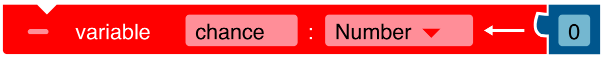
```
show  image
```

□

□



```
set  chance  to  random integer from  0  to  2
```



```
−  variable  chance : Number ←  0
```

□

description

1 request pressed button (input)

2 show an image (output)

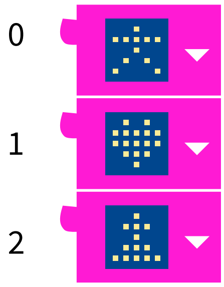
3 randomly select an image from a list of images in a fixed order

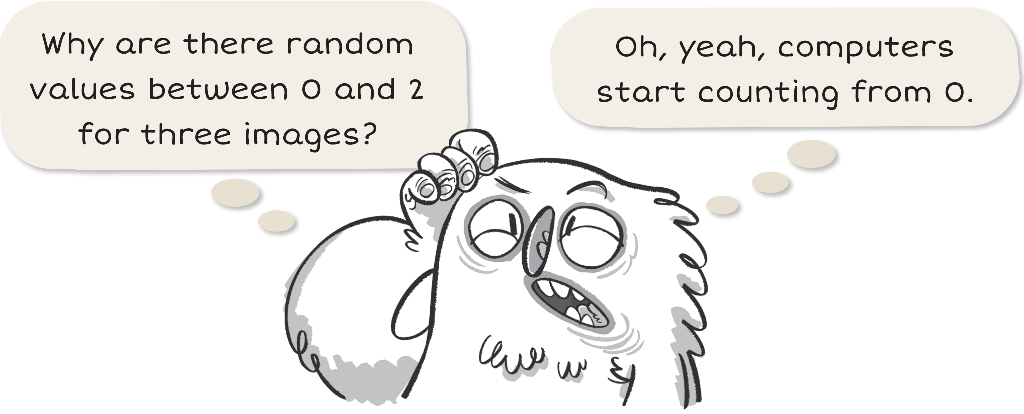
4 repetition (infinite loop*)

5 generate random numbers in a specified range

6 creating the "chance" variable

7 creating an image list with a fixed order




Why are there random values between 0 and 2 for three images?
Oh, yeah, computers start counting from 0.

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)

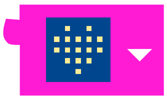
generate random

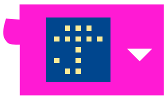
~~creating an image list with a fixed order~~

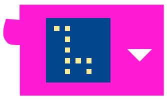
creating the "chance" variable

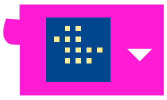
work steps		once	repeatedly
1.	creating an image list with a fixed order	X	
2.			
3.			
4.			
5.			
6.			
7.			

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

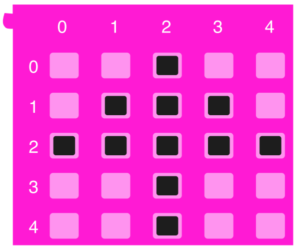




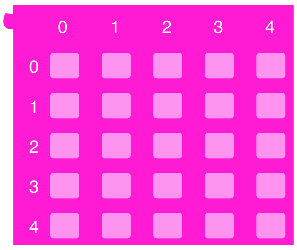




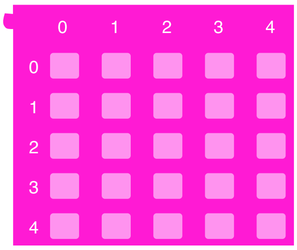
5. Create your own images for your program. Draw them into the LED screens by coloring the corresponding boxes.



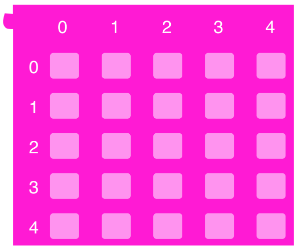
TREE



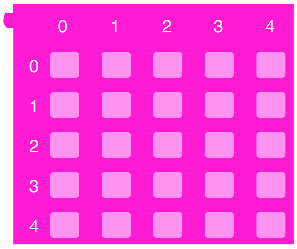
____



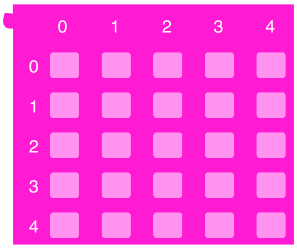
____



____



____



____

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".

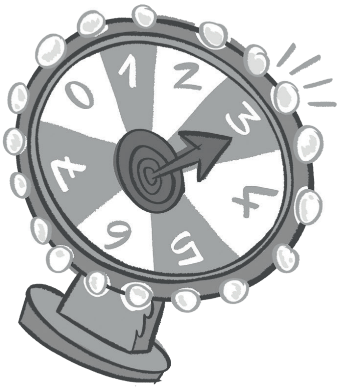
- We need a break to separate the image from the name.

Control ▶ Wait → "wait ms"

Click in the blue bock and enter the number 2000.

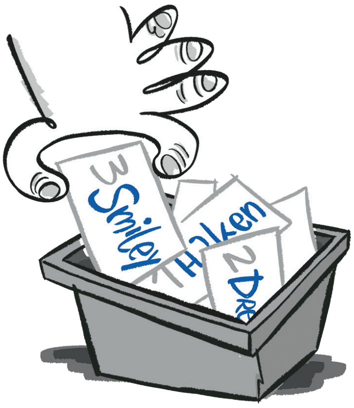
- The word should match the image from the related image list.

"chance"



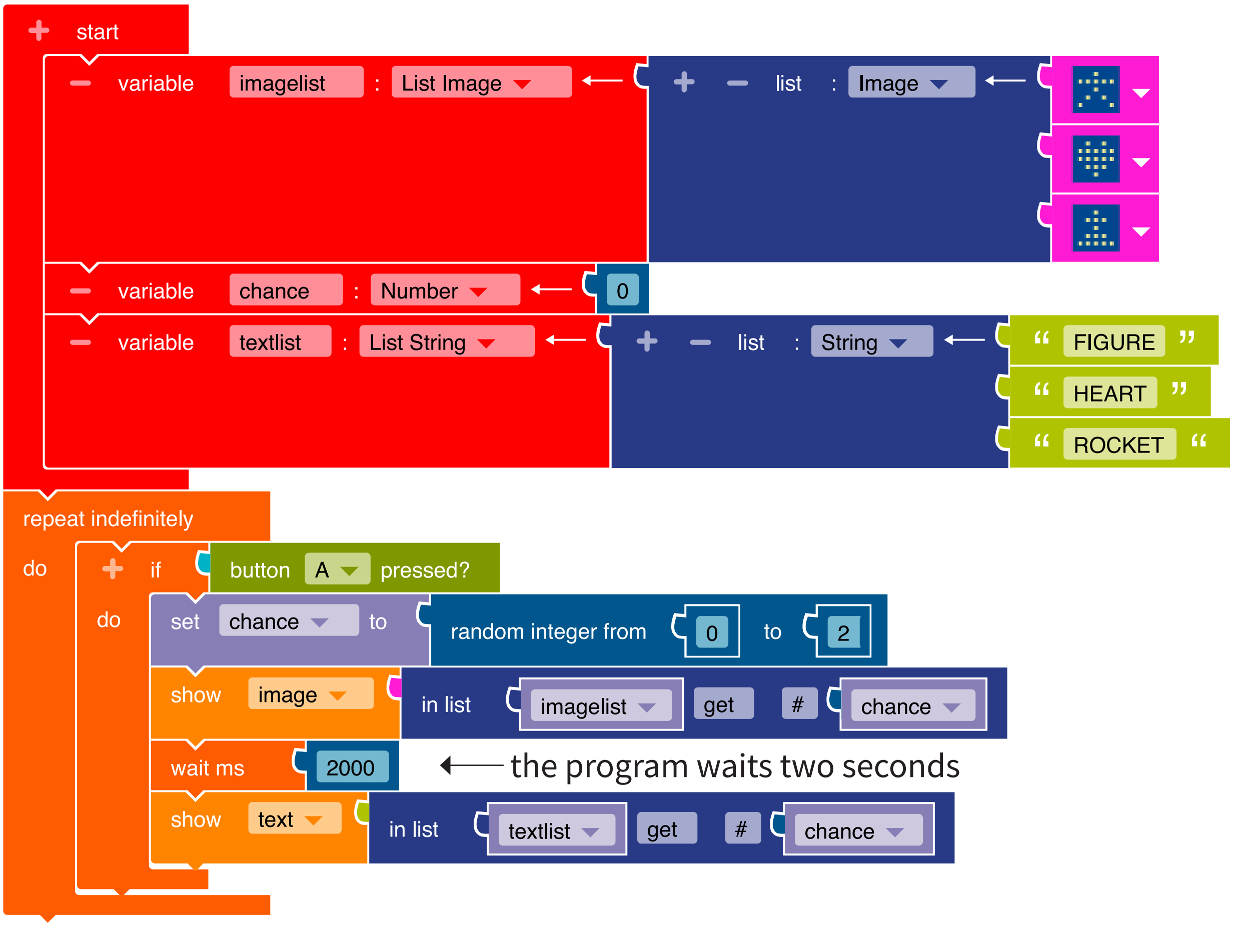
"image list"

"text list"



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

Remember how the computers count. ______________________________



```
start
  variable imagelist : List Image ← list : Image ←
  variable chance : Number ← 0
  variable textlist : List String ← list : String ← " FIGURE "
                                                     " HEART "
                                                     " ROCKET "
repeat indefinitely
do
  if button A pressed?
  do
    set chance to random integer from 0 to 2
    show image in list imagelist get # chance
    wait ms 2000
    show text in list textlist get # chance
```

← the program waits two seconds

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

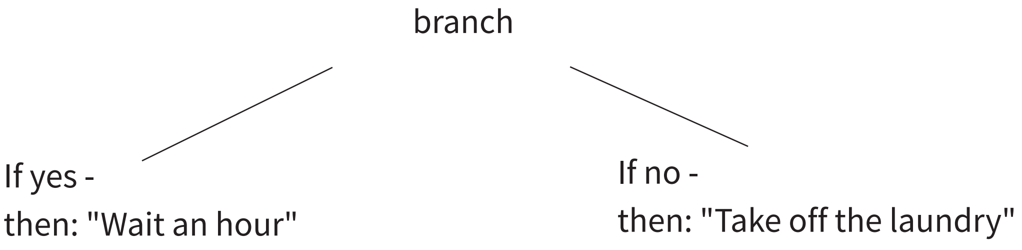
An infinite loop **has no condition** und and will run until the Calliope mini is switched off.

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?"


branch
If yes -
then: "Wait an hour"
If no -
then: "Take off the laundry"