

DINNER SELECTION



In this lesson, your Calliope mini becomes your guide in a really difficult question: what's for dinner today?

Shake the Calliope mini and you will get an answer to your question.

Tip: Ask each family member about his or her favourite food and write it down in a list - you'll need it later!



DINNER SELECTION



You need the following blocks and categories for this program:

Action



Show text

Displays a text on the screen.

Sensors



Position sensor

Returns "true" when the Calliope mini is shaken.

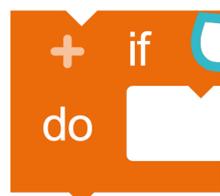
Control



Infinite Loop

Repeats the action indefinitely.

Logic



If/do condition

If a condition is true, then execute specific commands.

Math

Variables



DINNER SELECTION



You need the following blocks and categories for this program:

Action



Sensors



Control

Logic



Math



Variables

Logic comparison

Returns true if both inputs are equal.

Value

The input value is a number.

Random value generator

Outputs a random number between 1 and 100.

Set item

Assign a value to this variable, for example 0.



DINNER SELECTION



1

Click on the plus next to "Start" to create a new variable.



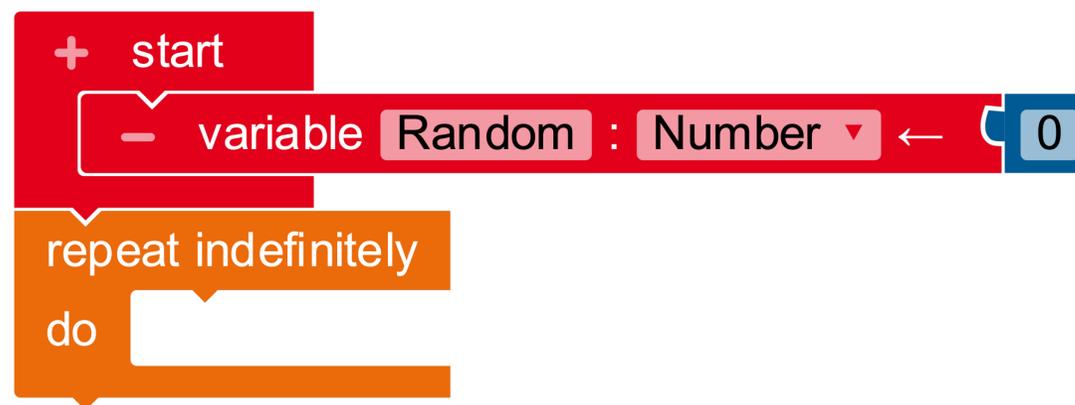
DINNER SELECTION



2

Change the name of the variable from "item" to "Random" and attach the block **repeat indefinitely** to the red start block.

Control



```
+ start
- variable Random : Number ← 0
repeat indefinitely
do
```

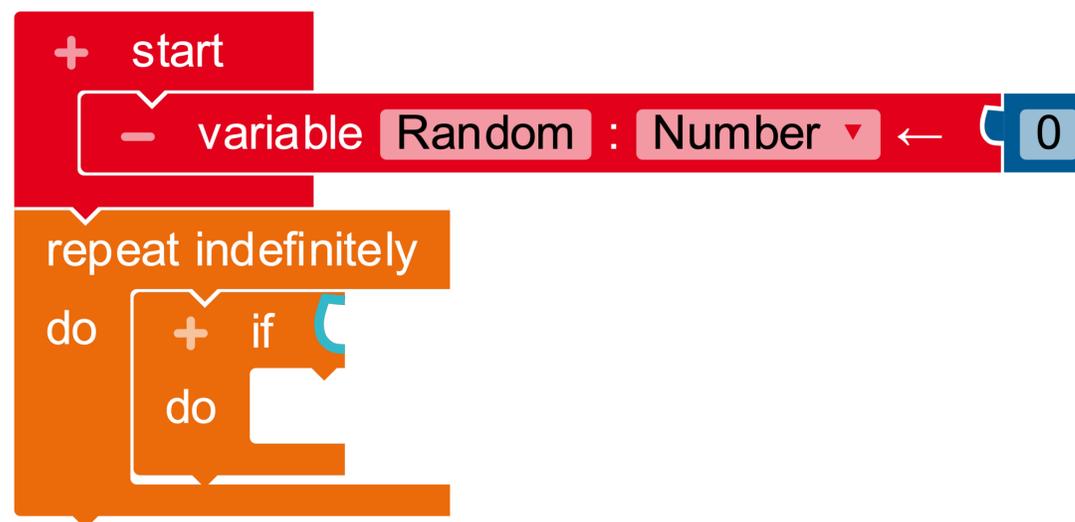
The image shows a Scratch code editor snippet. It starts with a red 'start' block. Attached to its bottom is a 'variable' block with the name 'Random' and type 'Number', set to the value '0'. Below the 'start' block is an orange 'repeat indefinitely' block with a 'do' block attached to its bottom.

DINNER SELECTION



3 Insert an **if/do** condition.

Control



DINNER SELECTION



4

Add the green block "get gesture" from the sensors category. Select "shaking" in the menu.

Sensors

The image shows a Scratch script with the following blocks:

- start** block (red)
- variable Random : Number** block (red) with a value of **0** (blue)
- repeat indefinitely** block (orange)
- do** block (orange) containing:
 - if** block (orange) with the condition **get shaking** (green) **gesture** (green)
 - do** block (orange) containing a dropdown menu.

The dropdown menu lists the following gesture options:

- upright
- upside down
- at the front side
- at the back
- shaking
- freely falling



DINNER SELECTION



5

Insert the `set Random` block from the variables category into the "if/do" branch.

Variables

```
+ start
- variable Random : Number ← 0
repeat indefinitely
do
+ if get shaking gesture
do
set Random to
```



DINNER SELECTION



6

Assign a new value to your variable by appending the block random integer from 1 and 100.

Math

```
+ start
- variable Random : Number 0
repeat indefinitely
do
+ if get shaking gesture
do
set Random to random integer from 1 to 100
```



DINNER SELECTION



7

Change the range for the random value to "0 to 3". Your variable should only be able to contain four values.

Math

```
+ start
- variable Random : Number ← 0
repeat indefinitely
do
+ if get shaking gesture
do
set Random to random integer from 0 to 3
```



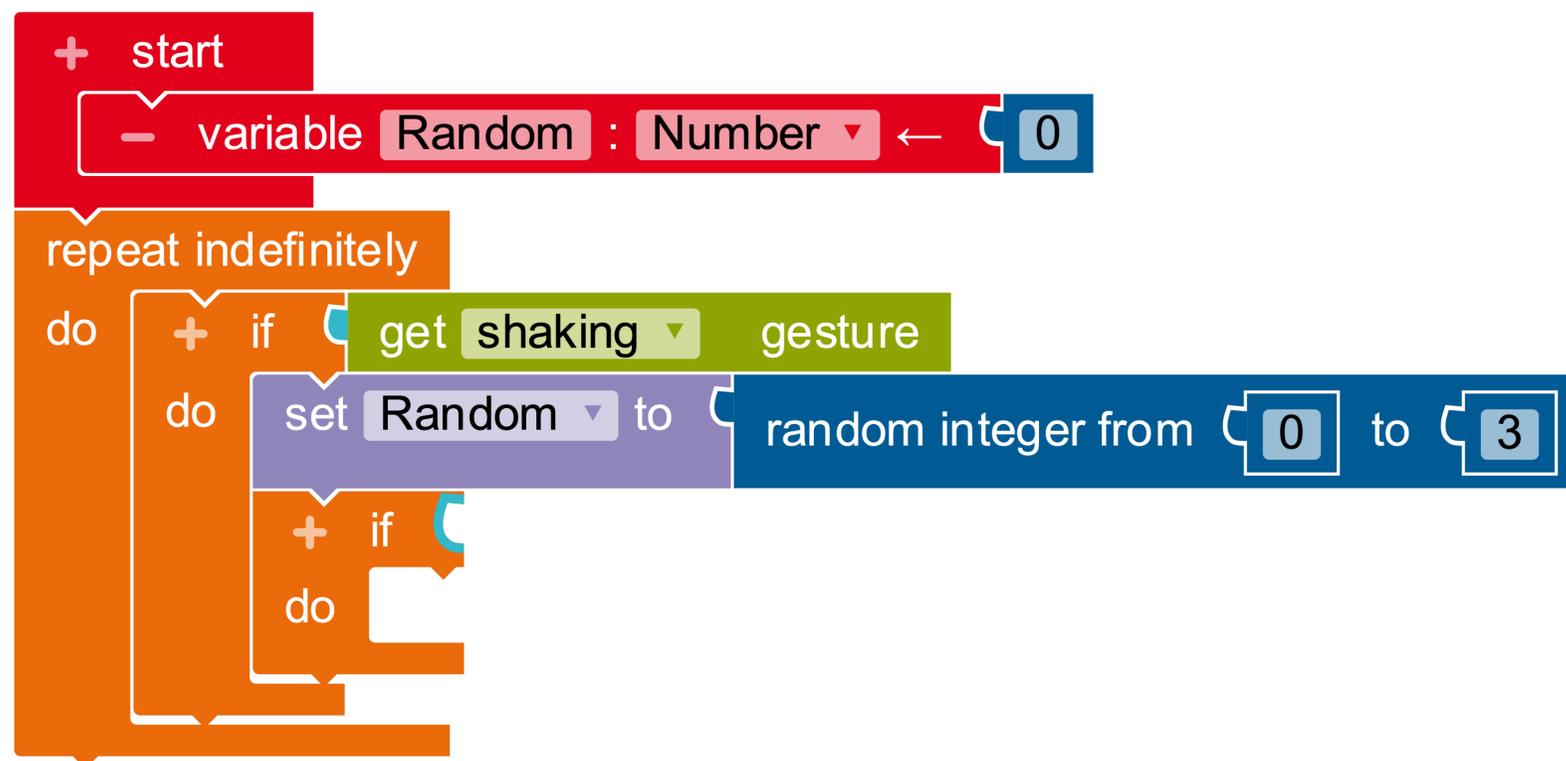
DINNER SELECTION



8

Insert another **if/do** condition below the „set Random to“ block.

Control



DINNER SELECTION



9 Insert a comparison block "=" from the logic category.

Logic

```
+ start
- variable Random : Number ← 0
repeat indefinitely
do
+ if get shaking gesture
do
set Random to random integer from 0 to 3
+ if Random = Random
do
```

The image shows a Scratch script for a dinner selection. It starts with a 'start' block, followed by a 'variable Random : Number ← 0' block. A 'repeat indefinitely' loop contains three nested 'do' blocks. The first 'do' block is an 'if' block with the condition 'get shaking gesture'. Inside this 'if' block is another 'do' block containing a 'set Random to random integer from 0 to 3' block. Below that is a second 'if' block with the condition 'Random = Random', which is currently empty.



DINNER SELECTION



10

In the first part of the code, the program selects a random number and stores it in the variable "Random". With the if/do condition you can assign dishes to the numbers. Set your variable "Random" to the left of the "=".

Variables

```
+ start
- variable Random : Number ← 0
repeat indefinitely
do
+ if get shaking gesture
do
set Random to random integer from 0 to 3
+ if Random =
do
```

The code consists of the following blocks:

- A red **start** block.
- A red **variable** block: `variable Random : Number ← 0`.
- An orange **repeat indefinitely** loop block.
- Inside the loop, an orange **do** block.
- Inside the **do** block, a green **if** block: `if get shaking gesture`.
- Inside the **if** block, a purple **do** block.
- Inside the **do** block, a blue **set** block: `set Random to random integer from 0 to 3`.
- Below the **set** block, a cyan **if** block: `if Random =`.
- Below the **if** block, an orange **do** block.



DINNER SELECTION



11 Place the number "0" from the math category to the right of the "=".

Math

```
+ start
- variable Random : Number ← 0
repeat indefinitely
do
+ if get shaking gesture
do
set Random to random integer from 0 to 3
+ if Random = 0
do
```



DINNER SELECTION



12

By using the **show text** block you can now add the first dish to the menu and assign it to the number "0".

Action

```
+ start
- variable Random : Number ← 0
repeat indefinitely
do
+ if get shaking gesture
do
set Random to random integer from 0 to 3
+ if Random = 0
do
show text " Pesto pasta "
```



DINNER SELECTION



13

Repeat the last three steps and match more dishes to the numbers. Change the [number in the comparison block](#) of the if/do condition.

Action

Logic

Math

Variables

```
+ start
- variable Random : Number ← 0
repeat indefinitely
do
+ if get shaking gesture
do
set Random to random integer from 0 to 3
+ if Random = 0
do show text "Pesto Pasta"
+ if Random = 1
do show text "Sausages and mashed potatoes"
+ if Random = 2
do show text "Rice with mushrooms"
+ if Random = 3
do show text "Greek salad"
```

