Now you\_

Draw your robot:

**1\_Data**

Name your robot. The name can be numbers, letters, any key on the keyboard (+, \_, #, …), for example: “Robo\_Me“

A name of numbers, letters or any key on the keyboard (+, \_, #, …) is called a **string.** Strings are always put in quotation marks (“ “).

Now you\_

Name your robot:

“ “

There are different types of data:

* numbers,
* strings and
* **Boolean variables**.

You have already gotten to know numbers and strings. Boolean variables are data, which have one of two possible values - they can be either true or false).

Describe your robot.

Review\_

Forming negative forms

Now you\_

* Create Boolean variables: Write down 5 statements, which are true or false about your robot.
* Extra\_challenge: use *and/or/doesn’t have/isn’t …*
* Let a partner find out which statements are true.

|  |  |  |
| --- | --- | --- |
| for example | true | false |
| My robot has two arms. |  |  |
| My robot has arms *and* legs.  |  |  |
| My robot has arms *or* legs. |  |  |
| My robot *doesn’t have* a green nose. |  |  |

|  |  |  |
| --- | --- | --- |
| Statement | true | false |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**2\_Computational Thinking**

Every big task can be made into smaller ones; this means: one big problem really is many little problems. To solve a problem, computers follow many little steps. If computers have solved a difficult problem, they can help other computers with similar problems, too.

This is how a computer thinks; this is called computational thinking.

The only way to learn something new is to try and make mistakes!

**3\_Programming**

Making difficult problems, events or tasks into smaller ones.

This is called: **decomposition.**

A list of small steps to finish a difficult task is called an **algorithm.**

Writing this list of smaller steps is called **programming** or **coding.**

The people who write these lists are **programmers.**

The computer has to follow every step; it cannot leave out any steps. The steps have to be in a specific order.

This specific order is called a **sequence.**

Programming an algorithm could look like this, for example:

Task: performing Queen’s “We will rock you“

Things you need: hands, thighs

Steps:

Step 1: when the music starts playing, clap your hands on your thighs;

Step 2: clap your hands on your thighs;

Step 3: clap your hands;

Step 4: if the music continues, then go back to step 1; if the music stops, then stop performing.

useful vocabulary for ‘dancing':

verbs: to bend, to clap, to dab, to drag, to freeze, to glide, to lift, to move, to play the air guitar, to push, to put up, to roll, to shimmy, to slide, to shuffle, to step out, to take one step, to tap, to twist, to wave, …

directions: backward, forward, to the front, sideways, on the spot, turn, 90 degrees to the left, …

bodyparts: head, shoulders, elbows, fingers, hips, knees, legs, toes, …

Now you\_

Decompose a dance choreography. Program an algorithm.

Task:

Things you need:

Steps:

Step 1:

…

Now you\_

Get up. Get together with a partner. Dance!

One of you is the programmer; one of you is the robot.

The programmer reads out his/her dance sequences of his/her algorithm,

The robot completes this difficult task, following your programmer’s algorithm. Make mistakes, if the steps are not clear enough.

Change roles.

**4a\_Programming: loops**

In a **loop** the computer does a sequence of steps again and again; it **repeats** the step(s) until one specific piece of the difficult task is finished. For example, let’s take a closer look at steps 1-2:

“Step 1: clap your hands on your thighs;

Step 2: clap your hands on your thighs; “

The loop can look like this:

*Step 1: “repeat twice: clap your hands on your thighs;“*

2

In the programming tool ‘Scratch’ the loop looks like this:

Now you\_

Go through your algorithm and program loops, if possible.

**4b\_Programming: selection**

If there are different conditions, and also different options for the computer to react, this is called a **selection.** For example, let’s take a closer look at step 4:

*“Step 4: if the music continues, then go back to step 1; if the music stops, then stop performing.“*

There are two conditions:

condition 1: “the music continues“

condition 2: “the music stops“

There are two options to react:

option 1: “go back to step 1“

option 2: “stop performing“

one condition matches one option:

condition 1 matches option 1;

condition 2 matches option 2.

If condition 1 happens, then option 1 follows.

If condition 2 happens, then option 2 follows.

If “the music continues“, then “go back to step 1“.

If “the music stops“, then “stop performing“.

A selection puts together the conditions, and options for the computer to react in one conditional sentence, using: if, then, or else.

The selection for step 4 can look like this:

*Step 4:* ***if*** *the music continues,* ***then*** *go back to step 1;* ***or else:*** *stop performing.*

*or*

*Step 4:* ***if*** *the music has stopped,* ***then*** *stop performing;* ***or else:*** *go back to step 1.*

**

In the programming tool ‘Scratch’ the selection looks like this:

Now you\_

Go through your algorithm and program a selection, if possible.

Review\_

* conditional sentences 1
* present perfect



|  |  |  |
| --- | --- | --- |
| description | link | QR code |
| video “Candy and coding“ - further information about programming basics: | [https://candy.codes](http://candy.codes) |  |
| First steps in programming with the program ‘Scratch‘: | <https://scratch.mit.edu> |  |
| First steps in programming hip hop dance moves with the program ‘Scratch‘: | <https://scratch.mit.edu/pathways/hiphop/> |  |

Extra\_Now you\_

* For more information on coding basics visit: [https://candy.codes](http://candy.codes)
* Visit <https://scratch.mit.edu> or <https://scratch.mit.edu/pathways/hiphop/> to program your dance moves.

Extra\_Now you\_

* Put together dances of two or more robots. Write new algorithms.

Define the following vocabulary for your robot.

|  |  |
| --- | --- |
| vocabulary | English definition |
| data |  |
| string |  |
| Boolean variables |  |
| decomposition |  |
| algorithm |  |
| programming |  |
| sequence |  |
| selection |  |
| loop |  |