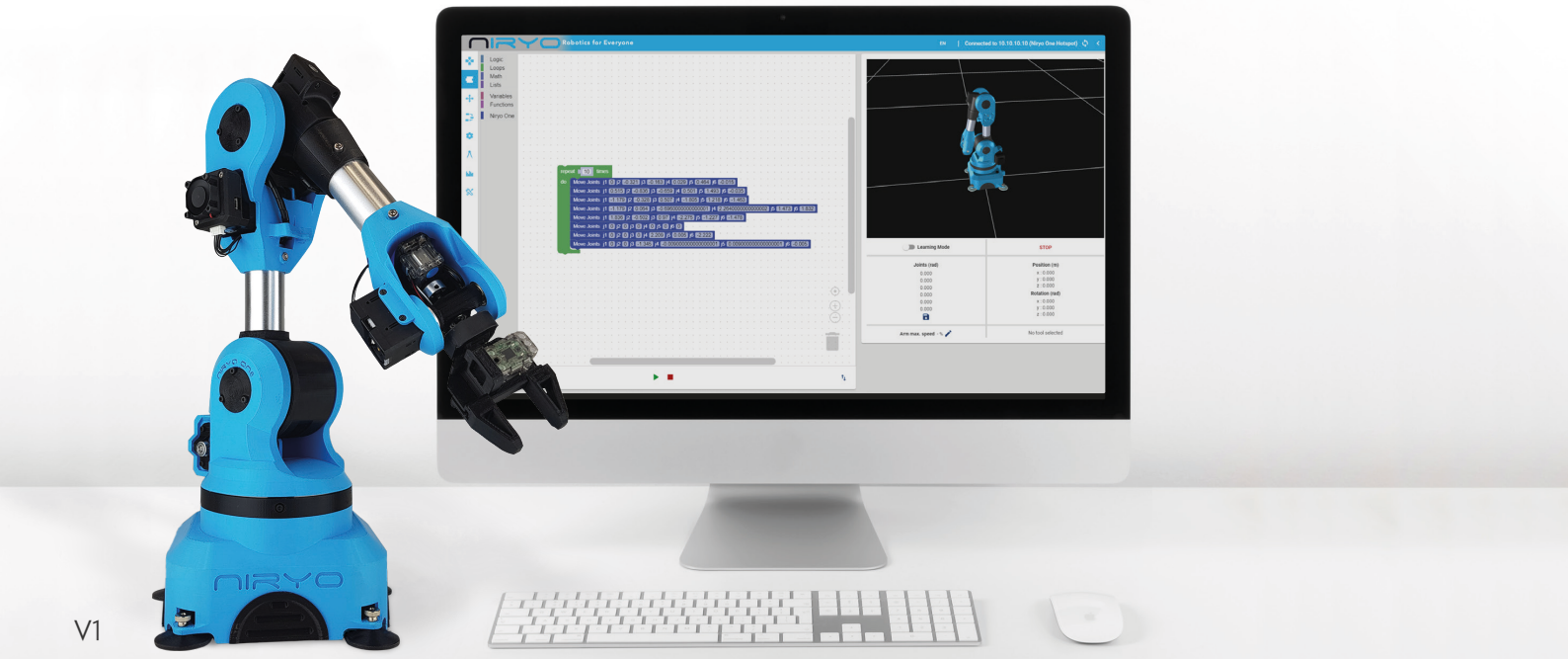
NIRYO

HUMAN - MOTION - ROBOT

BLOCKLY 1

Programming the Niryo One with Blockly



V1

OBJECTIVES

- Getting familiar with the basics of robotic programming
- Controlling the robot using Niryo One Studio and Blockly
- Being able to create algorithms to control the robot

REQUIREMENTS

- Starter guide

WHAT YOU WILL NEED

- Computer equipped with WIFI
- Niryo One Studio (Niryo One's graphical programming interface)
- A Niryo One

PROGRAMMING THE NIRYO ONE WITH THE LEARNING MODE

PRESENTATION OF THE LEARNING MODE

The « **learning mode** » is a programming method in which the **human operator manually places a robot in a position** and then saves it to let the robot execute it later.

It is possible to program the Niryo One using this programming method. To enable or disable this mode, click the « **Learning mode** » button on the Niryo One Studio interface.

EXERCISE

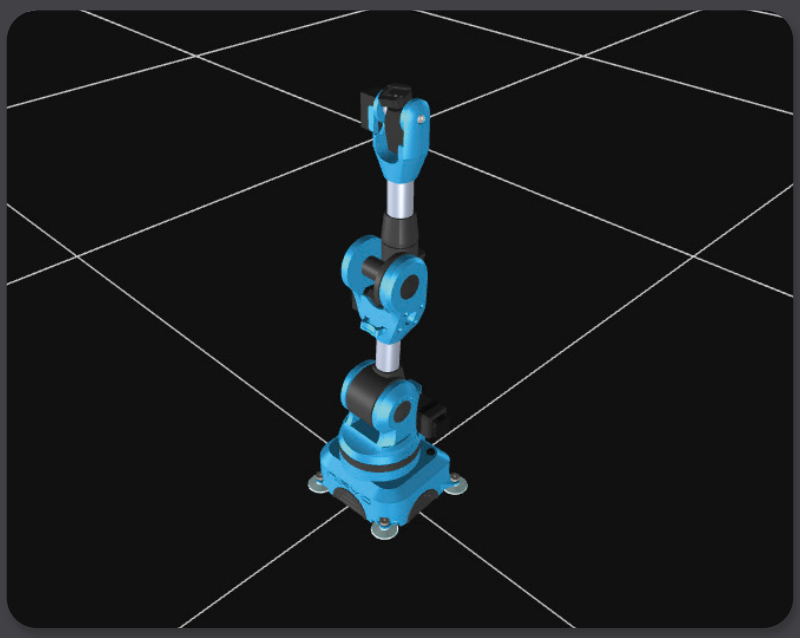
- Activate the learning mode of the robot
- Move the robot with your hand to a position of your choice
- Hold the robot in this position and click on « **UPDATE VALUES** » before releasing the robot

You can also press the upper button on the robot to save the value

- Deactivate the « **learning mode** »
- Execute a movement to this position by clicking on « **MOVE AXES** »

Did the robot make the desired movement?

- We would like to put the robot in a « top » position as shown in the figure below:



Which axis will reach its maximum angle?

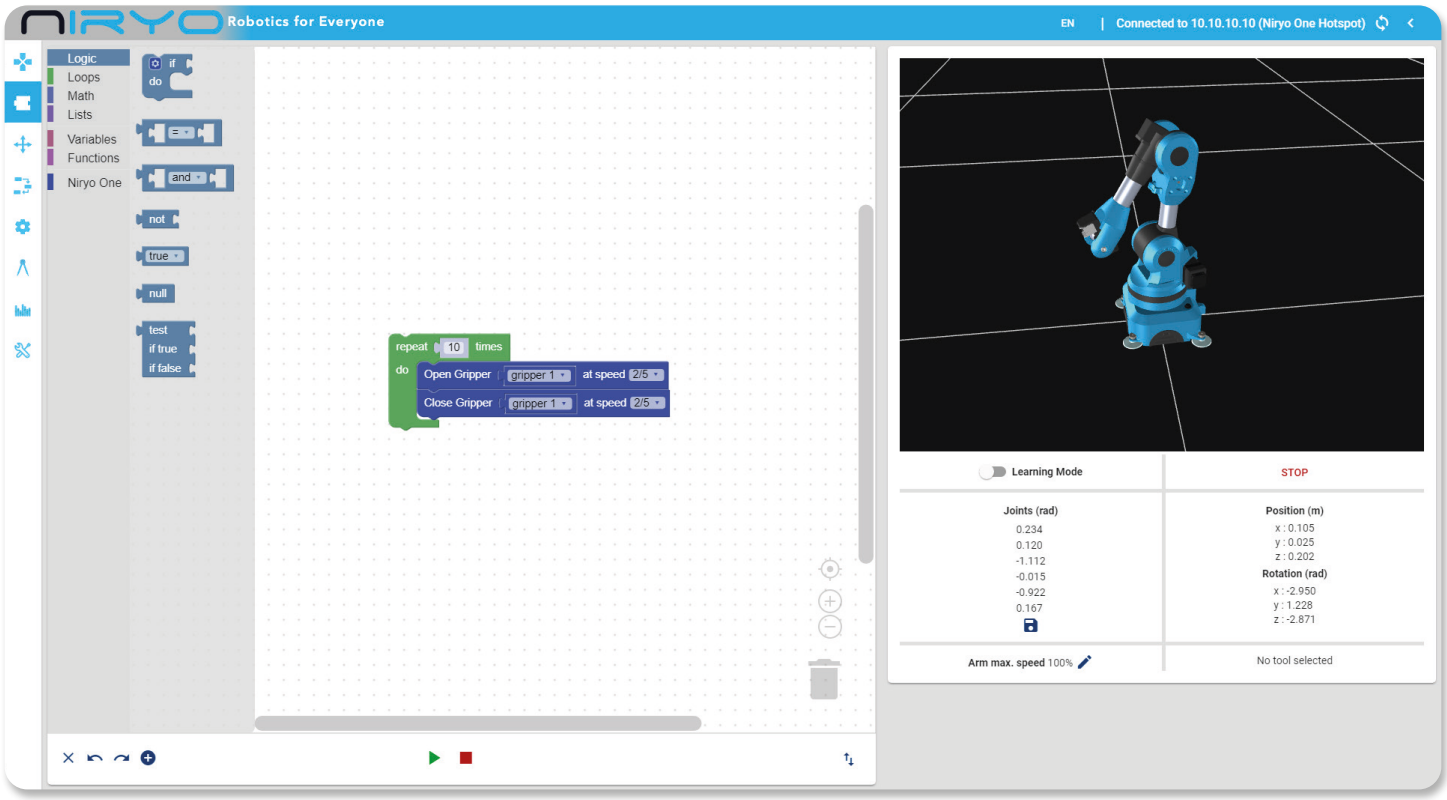
- Activate the « **learning mode** » and move the robot with your hand to this position
- Hold the robot in this position, select « **UPDATE VALUES** » and then save it with the name « TOP »
- Select this position and execute it by clicking on « **MOVE AXES** »

PROGRAMMING THE NIRYO ONE WITH BLOCKLY

In this section, we look at the basic functions of the visual programming software, based on Blockly.

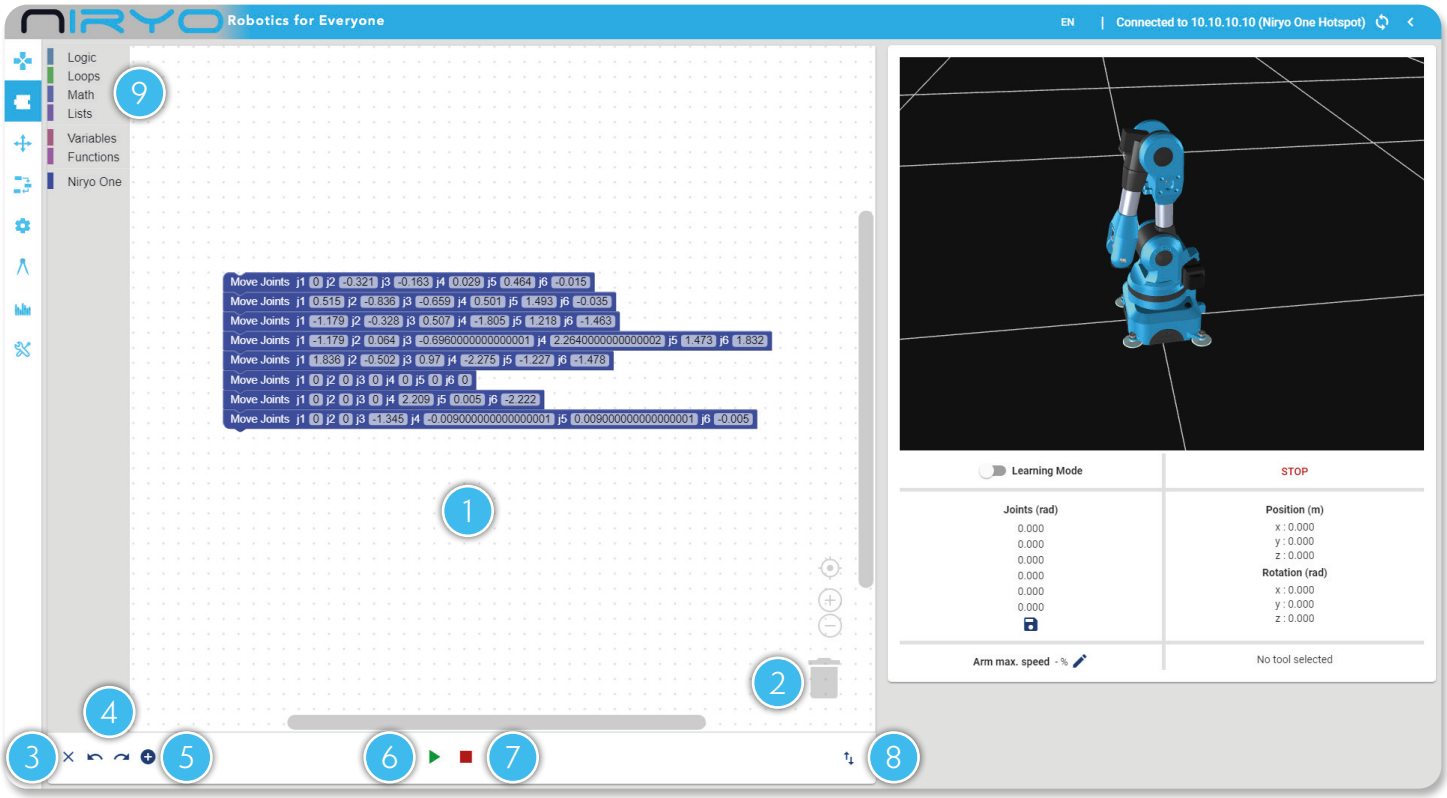
PRESENTATION OF BLOCKLY

The visual programming interface of Niryo One Studio is based on Blockly, a Google library, which is also used for the Scratch project made by the MIT.



THE PROGRAMMING ENVIRONMENT

The different parts of the environment and their functionalities are :



1. **The workspace that will contain your code**

2. **To delete a block, simply drag it to the trash can**
   You can also select it then press the « Del » key on your keypad

3. **Delete the current workspace**

4. **Undo / Redo**

5. **Add a position block**

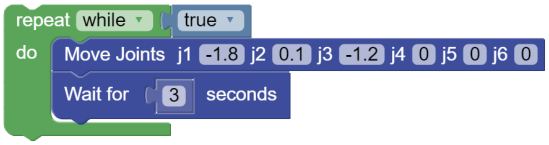
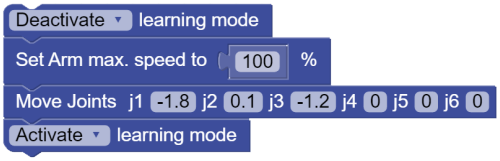
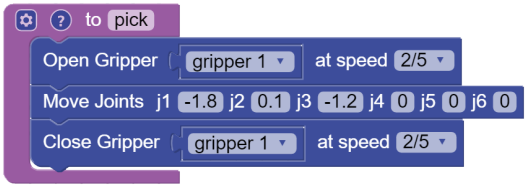
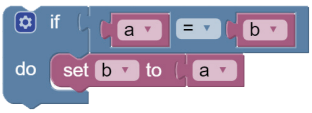
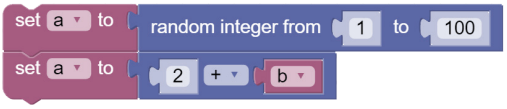
6. **Execute the sequence displayed in the workspace**

7. **Stop the execution of the current sequence in theworkspace**

8. **Import or export a sequence**

9. **The different programming blocks**

THE PROGRAMMING BLOCKS

CONCEPT	EXPLICATION	EXEMPLE
Variables	Variable blocks are used to create variables and to use them in the program. *A variable is a symbol that associates a name with a value.*	
Lists	List blocks are used to store and then access a list of numbers and strings. *A list is a structure that allows data to be grouped together in such a way that it can be accessed freely.* *It can be considered as an array.*	
Loops	Loop blocks can be used for iteration (repetition of a series of instructions). *A loop is a programming control structure that allows the repeated execution of a sequence of instructions.*	
Niryo One	Niryo One blocks are used to control the robot, its tools and its I/O interfaces.	
Functions	Function blocks are used to create and call up functions and procedures. *A function is a section that encloses a part of code performing a specific task that can be reused in the program.*	
Logics	Logic blocks allow either to initiate an action according to a condition or to manipulate logic variables.	
Maths	Math blocks are used for mathematical operations.	

Refer to the appendix for more details on the different blocks.

EXERCISE

We want to move the Niryo One to the two positions P1 and P2 :

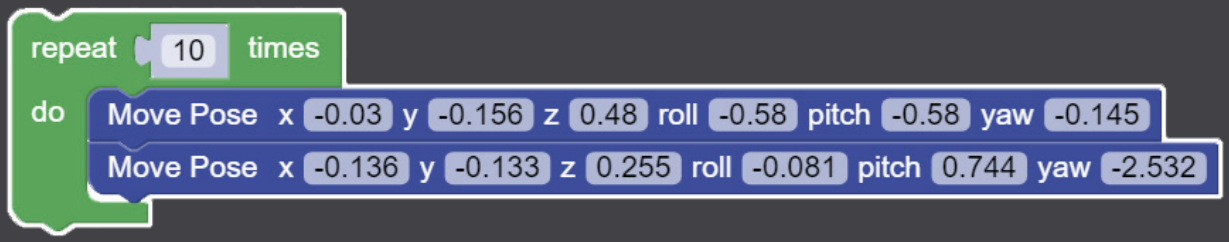
P1 = [x = -0.03 ; y = -0.156 ; z= 0.48 ; roll= -0.58 ; pitch = -0.58 ; yaw = -0.145]

P2 = [x = -0.136 ; y = -0.133 ; z = 0.255 ; roll= -0.081 ; pitch = 0.744 ; yaw = -2.535]

After reading the appendix, identify the block that allows you to move the robot to a position with Cartesian coodinate x y and z.

- Reproduce the following sequence in Niryo One Studio

- Deactivate the « learning mode » and execute the program
- Add a new block to your sequence as shown in the illustration below:



- Deactivate the « learning mode » and execute this new sequence

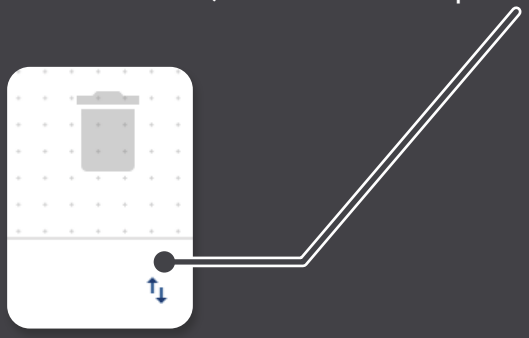
Explain the difference between the first and the second sequence and conclude on the role of the added block.

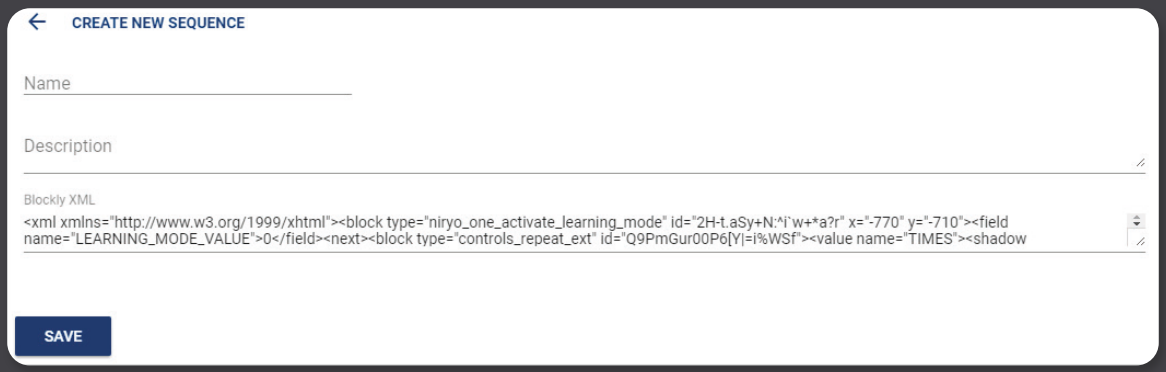
We want to automate the task of activating and deactivating the « learning mode ».

Which block should be used to perform this operation? Add this block at the beginning and at the end of your program, then execute it.

A sequence of instructions represents the program to be executed by the robot.

Keeping the program developed in the previous sections, click on the « import/export » button.

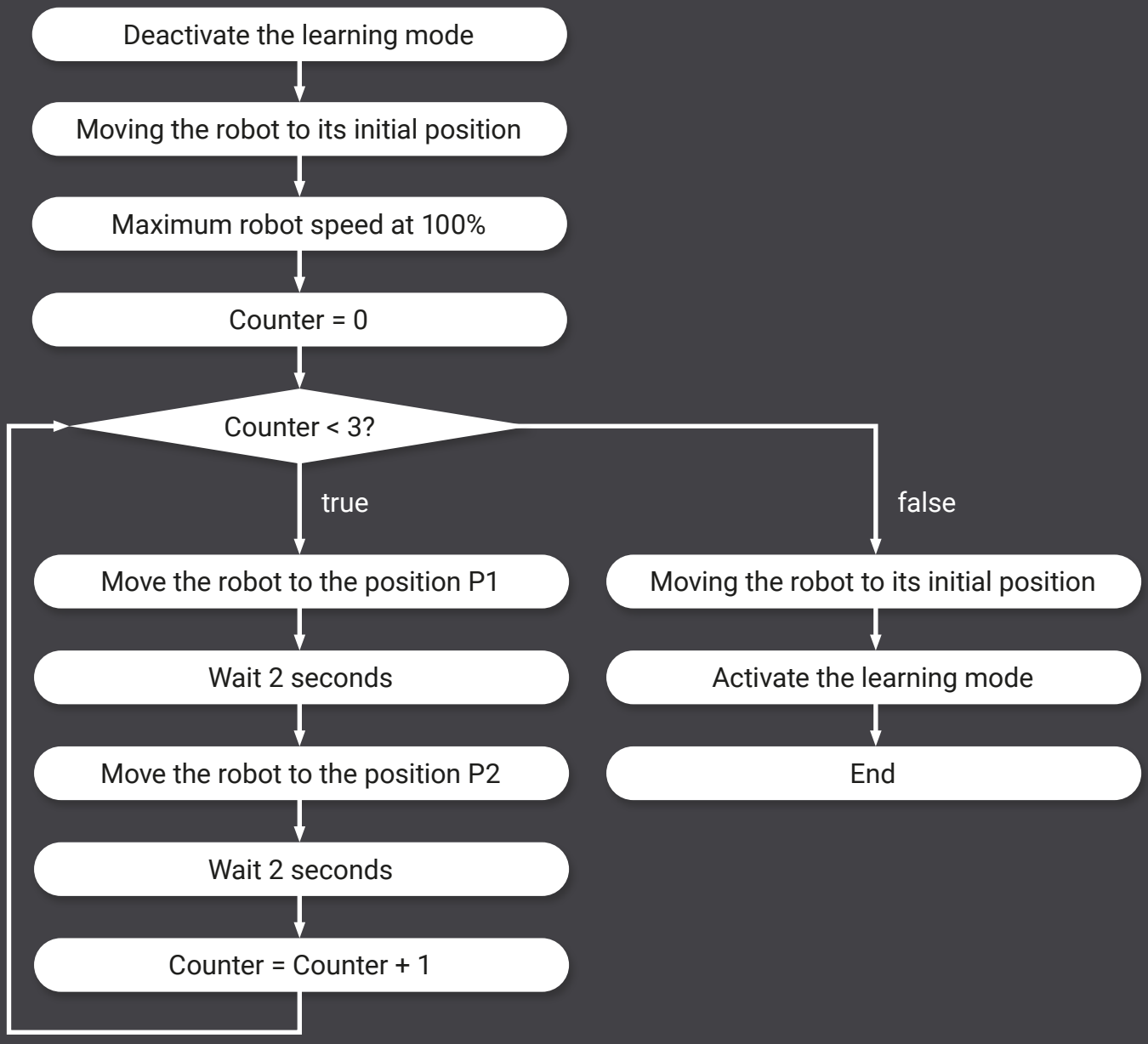




A sequence is characterized by:

- Its name (by default, it will be « sequence » + ID)
- Its description

We want to execute the following program:

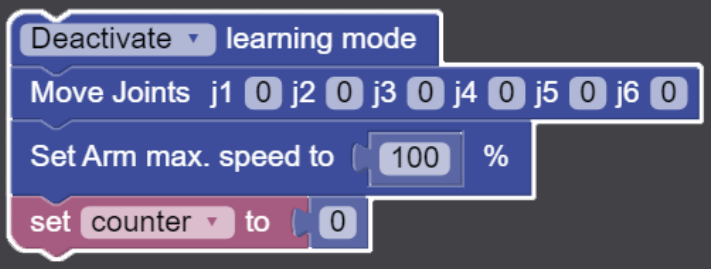


Knowing that:

- The positions of the robot are in joint coordinates
- The initial position of the robot is [0 ; 0 ; 0 ; 0 ; 0 ; 0]
- P1 = [-1.215 ; -0.333 ; -0.867 ; -3.053 ; -0.014 ; 0.127]
- P2 = [-1.377 ; 0.22 ; -0.281 ; -3.053 ; -0.031 ; 1.549]

Translate the previous diagram into text.

- Reproduce the program below:



Choose the block that allows to make the loop of the program shown above.

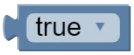
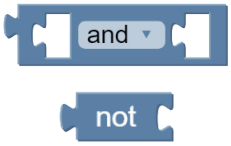
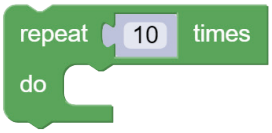
Using the appendix, complete the Blockly sequence so that it reproduces the previous diagram.

Taking into account the counter variable and the loop block, propose a block to replace these two blocks.

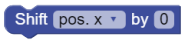
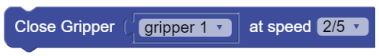
- Replace this variable and loop blocks with this new block
- Execute the sequence

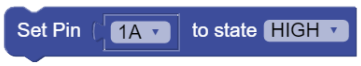
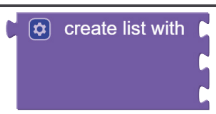
Comment.

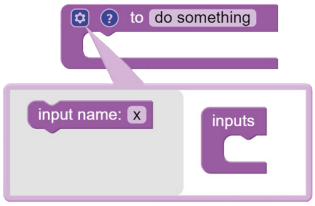
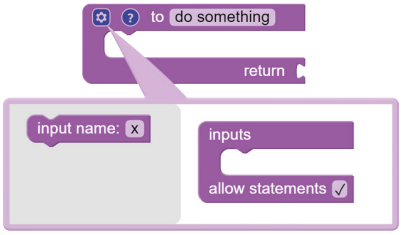
APPENDIX

TYPE	CONCEPT	BLOCK	EXPLANATION
Logic	Conditional instructions		Perform a specific task based on the evaluation of a condition.
Logic	Comparison operators		To compare certain values: • The equality test is performed with the = operator while the difference test is performed with the ≠ operator • Strict comparisons are made with the < and > operators. • Broad comparisons are made ith the <= and => operators.
Logic	Boolean types		The boolean type defines two states: true and false. The associated keyword are therefore «true» and «false».
Logic	Boolean logic operators		• And : return «true» if both values are «true» • Or : return «true» if at least one of the two values is «true» • Not : return the inverse of the boolean.
Logic	Value		A variable is «null» if it is null or contains no value.
Loops	Iteration		Repeat a sequence of instructions x times, where x is the number of iterations.
Loops	Iteration		Create a loop that executes an instruction as long as a condition is not checked.
Loops	Iteration		Repeat a specific sequence a number of times. *Example : count i from 1 to 10 by 1 (this instruction allows to count the variable i from 1 to 10 by incrementing by 1).*

TYPE	CONCEPT	BLOCK	EXPLANATION
Loops	Iteration		Repeat certain instructions for each item in a list.
Loops	Iteration		Exit the current loop (for/repeat/count) and go directly to the next instruction.
Math	Parameter setting		Specify a number for a function.
Math	Operator		• + : Add two numbers • - : Substract the second number from the first number • * : Multiply the two numbers • / : Divide the first number by the second number.
Math	Operator		Checks if a number is • Even: exactly divisible by 2 • Odd: not exactly divisible by 2 • Prime: a prime number (2, 3, 5...) • Whole: an integer • Positive: >= 0 • Negative: <= 0 • Divisible by: un nombre divisible par...
Math	Operator		Return the lower integer (round down), upper integer (round up) or the nearest integer (round).
Math	Operator		Return : • Sum: the sum of a list • Min: the minimum of a list • Max: the maximum of a list • Average: the average of a list • Median: the central value of a list • Mode: the most frequent value in a list • Standard deviation: the standard deviation of a list • Random item: a random number from the list.
Math	Operator		Calculate the remainder in an integer division.
Math	Operator		Return a random integer in the interval.

TYPE	CONCEPT	BLOCK	EXPLANATION
Niryo One	Movement		Move the robot axes at the specified angles.
Niryo One	Movement		Move the robot to the specified position and orientation.
Niryo One	Movement		Offset a coordinate (orientation or position) to a value.
Niryo One	Calibration		Calibrate the robot automatically.
Niryo One	Calibration		Calibrate the robot manually.
Niryo One	Parameter setting		Set the maximum speed of the robot.
Niryo One	Programming		Pause execution for the specified number of seconds.
Niryo One	Programming		Activate or deactivate the learning mode.
Niryo One	Tools		Change the tool type.
Niryo One	Tools		A variable that contains the grippers available for Niryo One.
Niryo One	Tools		Detach the gripper.
Niryo One	Tools		Open the gripper at a specific speed.
Niryo One	Tools		Close the gripper at a specific speed.
Niryo One	Tools		Pull air with the vacuum pump to catch an object.
Niryo One	Tools		Push air from the vacuum pump to release the object.
Niryo One	Tools		Configure the electromagnet by specifying the connection pin.
Niryo One	Tools		Activate the electromagnet by specifying the connection pin.
Niryo One	Tools		Deactivate the electromagnet by specifying the connection pin.

TYPE	CONCEPT	BLOCK	EXPLANATION
Niryo One	Inputs/Outputs		A variable that contains the pins available on the Niryo One.
Niryo One	Inputs/Outputs		Set the digital pin to HIGH or LOW.
Niryo One	Inputs/Outputs		Set the pin as an: • INPUT • OUTPUT.
Niryo One	Inputs/Outputs		Set the digital pin to HIGH or LOW.
Niryo One	Inputs/Outputs		Return the value of the pin.
Niryo One	Inputs/Outputs		Set the switch (SW1/SW2) to 1 (HIGH) or 0 (LOW).
Niryo One	Programming		Add a comment.
Niryo One	Programming		Add a break point. This will pause the program. To resume execution, press the «play» button.
Variables	Variables		Create and name a new variable. When you create a variable, the blocks corresponding to the variable are displayed.
Variables	Variables		Assign a value to a variable.
Variables	Variables		Change the value of a variable.
Variables	Variables		Return the variable.
Lists	Lists		Create an empty list.
Lists	Lists		Create a list by adding elements.
Lists	Lists		Create a list with an item repeated x times.
Lists	Lists		Return the lenght of the list.
Lists	Lists		Return the position of the: • First occurence of the item in a list • Last occurence of the item in a list.

TYPE	CONCEPT	BLOCK	EXPLANATION
Lists	Lists		Return: • # : the n item in the list • # from end : the n item in the list from the end of it • First : the first item in the list • Last : the last item in the list • Random : a random item in the list.
Functions	Functions		Create a procedure with arguments (input parameters).
Functions	Functions		Create a procedure with arguments (input parameters) and an output (return).

Find all our teaching materials on

www.niryo.com

LEARNING MODE: CORRECTION

- Activate the learning mode of the robot
- Move the robot with your hand to a position of your choice
- Hold the robot in this position and click on « **UPDATE VALUES** » before releasing the robot

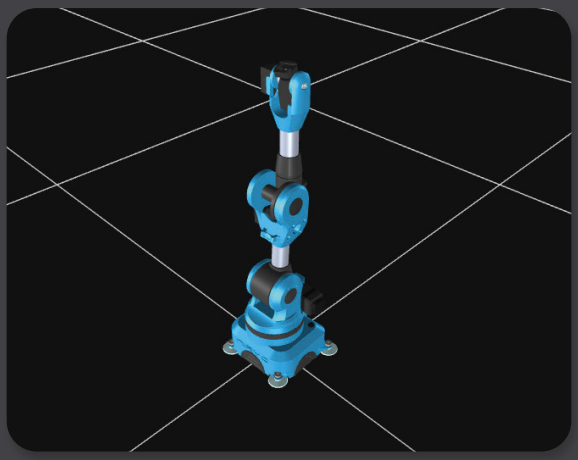
You can also press the upper button on the robot to save the value

- Deactivate the « **learning mode** »
- Execute a movement to this position by clicking on « **MOVE AXES** »

Did the robot make the desired movement?

The robot made the desired movement: indeed, the learning mode is designed to save the positions and sequences manually determined by the operator and then reproduce them.

- We would like to put the robot in a « top » position as shown in the figure below:



Which axis will reach its maximum angle?

The joint 3 will reach its maximum angle of 90°.

- Activate the « **learning mode** » and move the robot with your hand to this position
- Hold the robot in this position, select « **UPDATE VALUES** » and then save it with the name « TOP »
- Select this position and execute it by clicking on « **MOVE AXES** »

BLOCKLY: CORRECTION

We want to move the Niryo One to the two positions P1 and P2 :

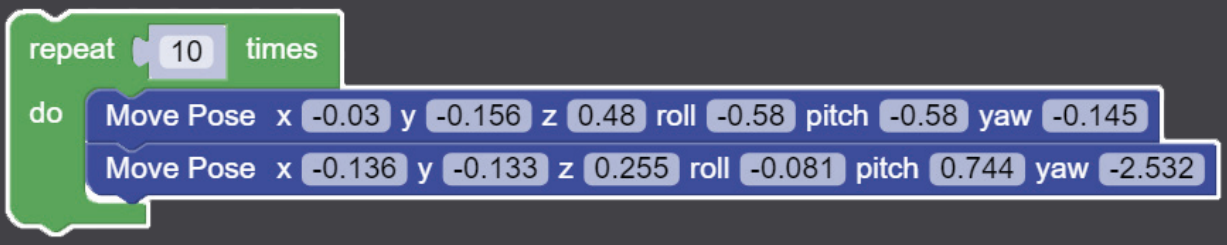
P1 = [x = -0.03 ; y = -0.156 ; z= 0.48 ; roll= -0.58 ; pitch = -0.58 ; yaw = -0.145]

P2 = [x = -0.136 ; y = -0.133 ; z = 0.255 ; roll= -0.081 ; pitch = 0.744 ; yaw = -2.535]

After reading the appendix, identify the block that allows you to move the robot to a position with Cartesian coodinate x y and z.

- Reproduce the following sequence in Niryo One Studio

- Deactivate the « learning mode » and execute the program
- Add the block to your sequence as shown in the illustration below :



- Deactivate the « learning mode » and execute the is new sequence

Explain the difference between the first and the second sequence and conclude on the role of the added block.

In the first code, the robot moved to a first position then to a second one. In the second code, the robot followed the same path and repeated this sequence 10 times.

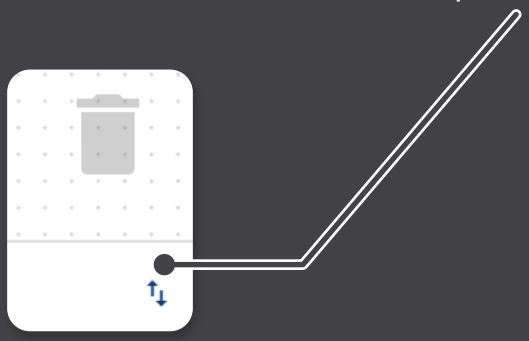
We want to automate the task of activating and deactivating the « learning mode ».

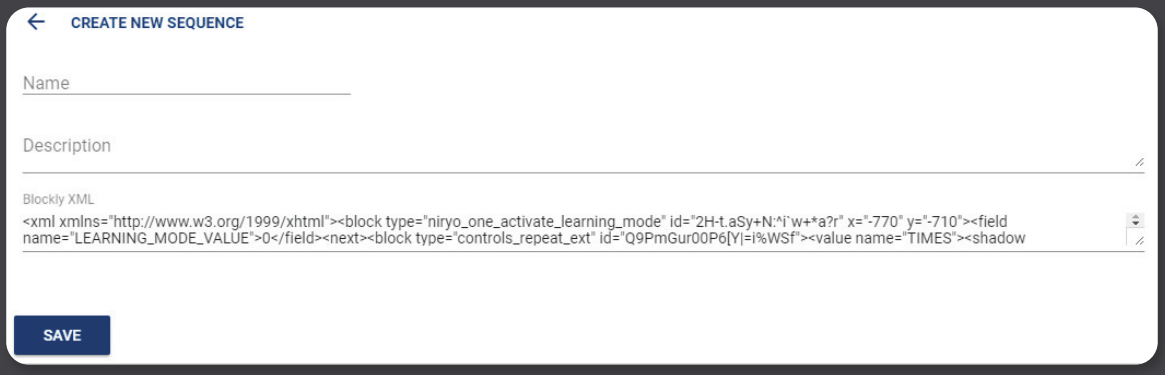
Which block should be used to perform this operation? Add this block at the beginning and at the end of your program, then execute it.

As explained in the appendix, the function of this block is to activate or deactivate the learning mode.

A sequence of instructions represents the program to be executed by the robot.

Keeping the program developed in the previous sections, click on the « import/export » button.

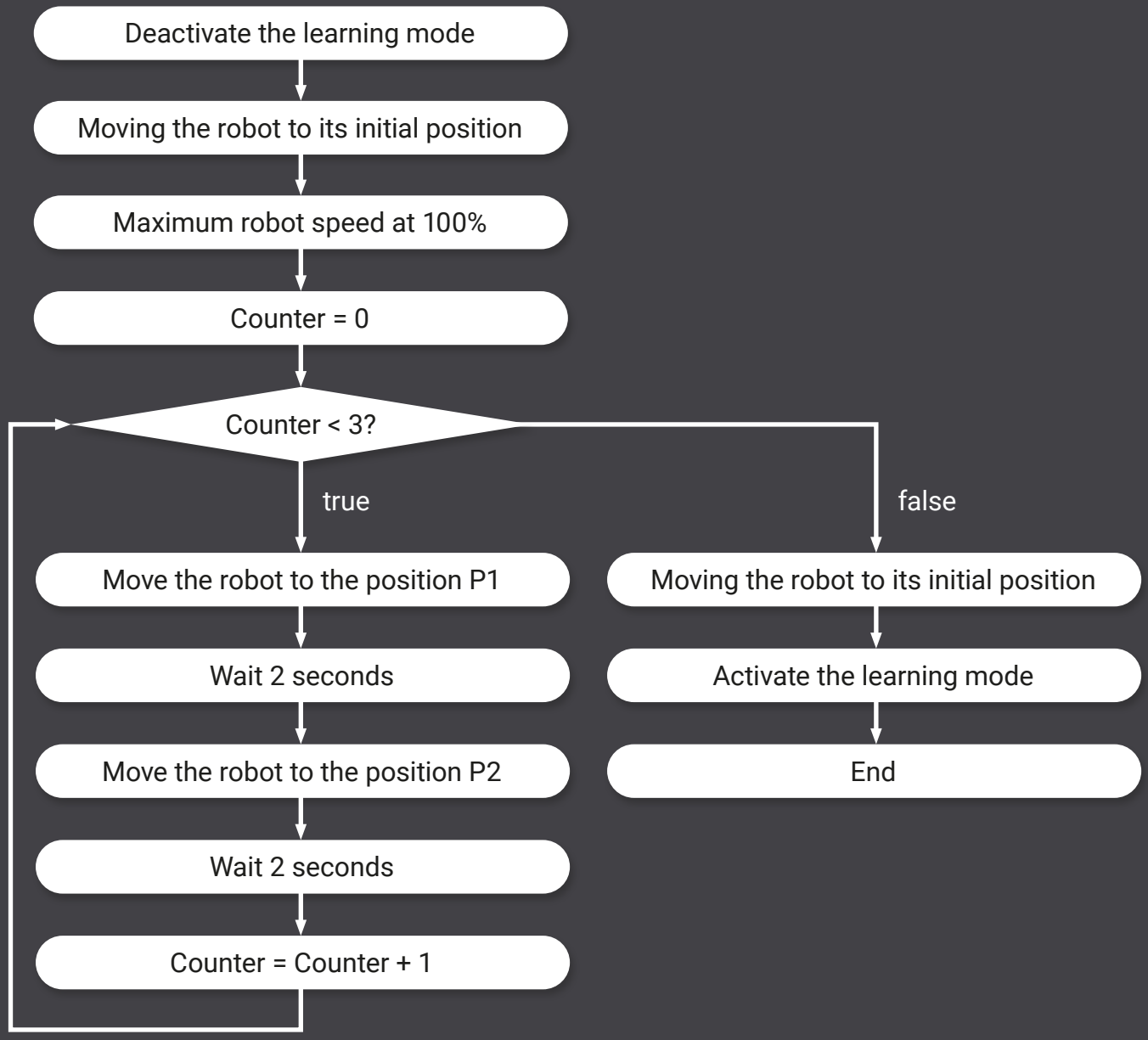




A sequence is characterized by:

- Its name (by default, it will be « sequence » + ID)
- Its description

We want to execute the following program:



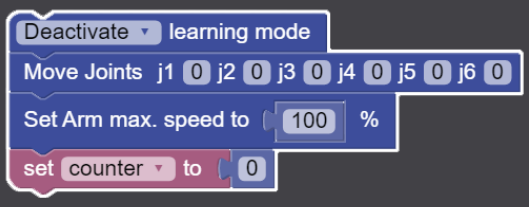
Knowing that:

- The positions of the robot are in joint coordinates
- The initial position of the robot is [0 ; 0 ; 0 ; 0 ; 0 ; 0]
- P1 = [-1.215 ; -0.333 ; -0.867 ; -3.053 ; -0.014 ; 0.127]
- P2 = [-1.377 ; 0.22 ; -0.281 ; -3.053 ; -0.031 ; 1.549]

Translate the previous diagram into text.

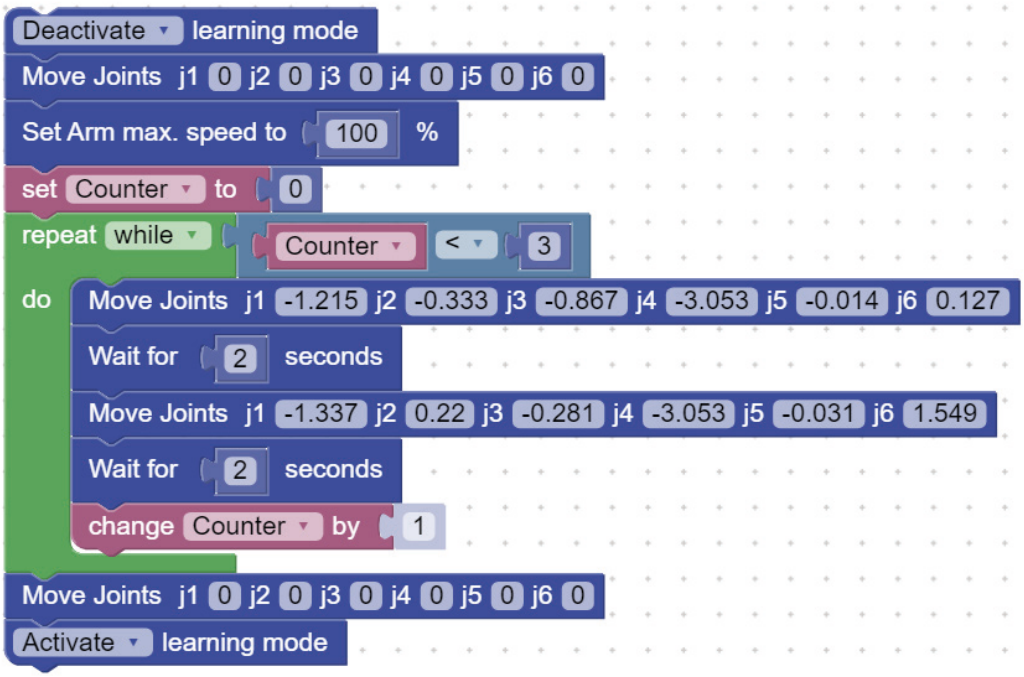
First, the learning mode is deactivated. Then, the speed of the robot is set at 100% and the robot goes to its initial position.
The robot goes to P1, waits 2 seconds, then goes to P2, and repeats 3 times this operation. Once the 3 repetitions achieved, the robot will go to its initial position and the learning mode will be activated.

- Reproduce the program below:

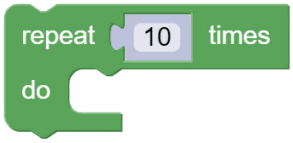


Choose the block that allows to make the loop of the program shown above.

Using the appendix, complete the Blockly sequence so that it reproduces the previous diagram.



Taking into account the counter variable and the loop block, propose a block to replace these two blocks.



or

- Replace this variable and loop blocks with this new block
- Execute the sequence and **comment.**

The variable Counter acts as a counter for iteration. Loop blocks such as «repeat» or «count with» can replace this variable and the «repeat while».