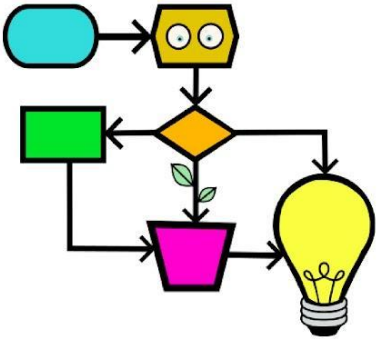
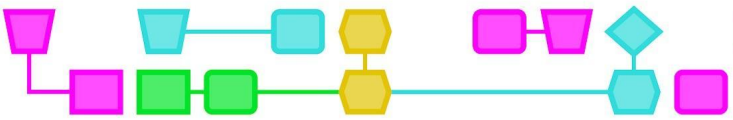
CTprimED

Programming - Offline programming



Co-funded by the Erasmus+ Programme of the European Union



Summary:
An algorithm can be understood as step-by-step instructions on how to solve a problem. Think of recipes, a script for a play, or a program for a computer. In this lesson, students will learn precisely what an algorithm is by taking part in a game that shows them what instructions a computer can understand. After they have done this, they will program each other as dance computers.

Target group: 6-12 (we have added a differentiation for children who cannot easily read and write)

Duration: 55 minutes

Learning goals: By the end of this exercise students will:

- know what an algorithm is
- learn how to use an algorithm in everyday life
- learn what a loop and condition are and how to use them in a program

Online/offline: offline.

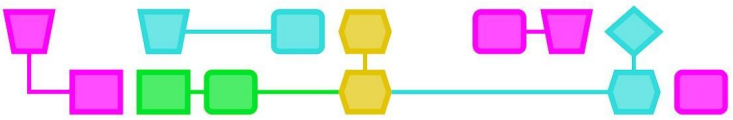
Computational Thinking:

- General skills: creativity, collaboration, logical thinking
- CT foundations: Algorithm, Pattern Recognition
- CT concepts**:** variables, sequences, loops, and conditions

Materials:

- Presentation: EN CTPrimED: Programming - Offline Programming - Presentation
- Dance pictograms: EN CTPrimED: Programming - Offline Programming - Dance Pictograms
- Pens
- Paper

Preparation:
Read through the lesson plan and prepare the presentation.



Lesson description for Offline programming

Introduction - Going from A to B (10 min)
Explain to the students that they are going to make their own algorithm. An algorithm is a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

Tell the students that for the first exercise, they will be the computer and they will have to 'program' you to walk from point A to point B in the classroom. Pick a place and label it A and then label the endpoint as B. Make sure there are a few obstacles in the way, such as, for example, a table that you would have to climb over or crawl under.

Pick 2-3 students who, in turn, will try to guide you from point A to point B. They will have to give you instructions such as 'take two steps forward' and 'turn right.' It is important that you take the students' commands incredibly literally: for example, if they say, 'go straight ahead', keep going straight ahead until you bump into something, and even then, keep pretending to walk forward. This will teach them that they also have to say 'stop' at some point.

Once you begin, you will find that students often find it difficult to explain everything as literally as possible. Tell the students that they have just made an algorithm: a process or set of rules to be followed in calculations or other problem-solving operations, to go from point A to point B. They instructed you step-by-step and explained every step to you literally.

Choreography - Algorithm (20 min)
Goal: Algorithm, clear instruction.

Explain that an algorithm can also be used for many different things: tying your shoes, following a recipe, or choreographing a dance. In this task, students will devise an algorithm for a dance that another group will perform. Divide the students into groups of 3-4 students. Each group will create a dance (choreography) lasting around 1-2 minutes.

Step 1: The students should think about what dance or movements they want to use. This can be a pre-existing dance, such as head–shoulders-knees-and-toes or a new dance they make up themselves.

Step 2: The students must then write out the dance very precisely on paper, so another group is able to perform their dance when they read the instructions.

Differentiation: use the dance pictograms for students who cannot easily read and write.

Step 3: Students exchange their choreography with another group. The groups will have five minutes to read and rehearse the choreography from the other group and then dance it in front of the whole group. Emphasize that students can only perform what is written on the paper.

Step 4: Review the dances and discuss with the students: Did the groups perform the dance exactly the same? What went well and what did not go so well?
It is highly likely that the students did not perform the dance exactly the same. Acknowledge this and explain that algorithms are step-by-step instructions that you can use to solve a problem. You can also use an algorithm to get a computer to do something for you. In this case, the algorithm was the description of the dance, while the group performing the dance was the computer. When describing the steps, precision is of vital importance because any ambiguities can cause errors in the program (= lead to differences in the performance of the dance). Precise instructions are incredibly difficult because you have to write everything out very clearly. Explain that this is probably why the dances went slightly different than how they were choreographed.

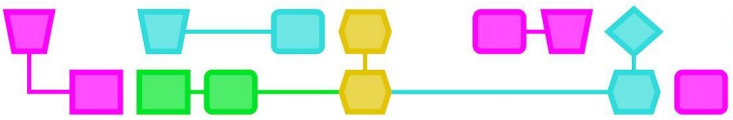
Tip: Turn on some music when performing the choreography.

Choreography - Loops and conditions (20 min):
Purpose: To learn what loops and conditions are.

Discuss with the students:
- What would you do differently now that you have seen the performance of the dance?
- How could we make sure the dance is performed more precisely?
    - Simplify the movements
    - Define the movements more clearly
    - Agree on which movements the dancers need to make
    - Repeat things

Start the presentation The Dance Computer. Show the first slide and tell the students that they are going to do the task again but now by using pictograms. Explain that each pictogram is a code for a move that the computer can perform.
- Show slide 2 and explain the pictograms.
- Show slide 3 and perform the dance with the class. Discuss how you could make them repeat this dance 3 times.
- Show on slide 4 that you can repeat all the steps but that it becomes very unclear. Discuss how this could be made shorter.
- Show slide 5. This is called a loop, a repetition. Discuss if there is another place to apply a loop (at the clapping of hands).
- After this, show slide 6. This is called a condition, an if-then rule. If something happens, then something else stops. Here, you see that if the music is on, then the dance is repeated. If the music is off, then the program stops.

Tell the students that they are now going to choreograph in groups using the pictograms to design their own dance. Make sure they include at least two loops and a condition.



After the assignment, discuss with the students what they learnt from the lesson today. They can now program a dance computer, with a function, loop, and conditions. Although with the dance computer, the code language they used were pictograms, on the computer they can use other code languages to program things, such as Scratch, MakeCode or Kodetu. These programming languages also allow you to use functions, loops, and conditions. In computer science, algorithms are needed to design computer programs.

Algorithm	A set of logical instructions for performing a task. In computer science, algorithms are needed to design computer programs.
Program	A sequence of instructions for a computer.
Function	A block of organized, reusable code used to perform a single, related action.
Instruction	A specific task that can be executed by a human or a computer.
Loop	A repetition of one (series of) instruction(s).
Condition	An if-then rule that states that something must happen before the other thing happens.

Tip: Show what a function, loop and condition look like in a programming language using Scratch Kodetu or Makecode.

Closing (5 min):
Tell the students that they created their own algorithm in this lesson! Explain that there are lots of algorithms in the world, including, for example, at traffic lights, or on social media like TikTok. Indeed, even when they are brushing their teeth, setting the table, and doing math, they use step-by-step instructions that make things go easier. You also see loops and conditions in everyday life, such as in a recipe where you have to repeat something, that if you have been to the bathroom then you have to wash your hands, or that if you travel you have to pack a suitcase. Algorithms, loops, and conditionals are everywhere!

Colophon

This publication is a product of CTPrimED (2021-1-NL01-KA210-SCH-000031319), funded with support from the Erasmus+ Programme of the European Union. This publication reflects the views of the authors, and the Commission cannot be held responsible for any use that may be made of the information contained therein.

Project Coordinator

NEMO Science Museum, The Netherlands

Partners

Universidad de la Iglesia de Deusto Entidad Religiosa, Spain

Stichting Children's Science Museum Curacao, Curacao

Co-funded by the Erasmus+ Programme of the European Union