

PRIMO

Teaching Guide

Introducing Cubetto in the classroom

Visit our website:

www.primotoys.com

Get in touch:

edu@primotoys.com



Table of content

How do I get started?	3
Research and value	4
Coding and computational thinking concepts in early years	5
Development and Learning in other key areas	6
First things first	7
Introducing Cubetto	8
Introducing the Board	8
Introducing the Blocks	9
Action causality	10
Unambiguous instructions	10
Setting up your classroom	10
The first challenge	11
The queue	12
Debugging	12
Introducing the function	13
Solving problems with the function	14
Additional help and resources	15

How do I get started?

Hello!

The Cubetto Playset is a Montessori inspired coding toy that allows children ages 3 to 6 to program a friendly wooden robot without screens or literacy, and it is powered by a programming language you can touch.

What teachers love most is its versatility in cross curricular applications. It fosters student learning in key areas like Spatial Reasoning, Creative Thinking, Socio-Emotional Communication, Numeracy, Literacy. It's born to be the perfect STEM tool to use with your youngest students, and it is.

The activities contained in this guide were created by educators for educators. New technology can sometimes be overwhelming to understand and adopt. We want to make it simple for you to integrate the Cubetto Playset and its tangible programming language into your teaching.

Where can I use Cubetto?

Here at Primo Toys, we are great fans of unplugged activities. Many of ambassadors start with an open conversation on "what is a robot", showing them examples of true robots (such as Cubettos) and robot toys that look like robots but aren't. It's extremely beneficial for the children to learn as young as possible that a robot is a machine we can programme to execute instructions. There are loads of inspiring unplugged activities most of which involve asking the children to program each other to perform a simple task like washing their hands, putting on a coat, clearing up a table.

Setting up your playset

You'll find instructions inside each playset that make setup fast and easy.

For more information, free lesson plans and resources, you can also visit:
www.primotoys.com/education



Help and support

We're here to help, and you can contact us anytime.

Educators support and purchases:
edu@primotoys.com

Research and value

The Cubetto Playset is a Logo Turtle inspired, Montessori led programming system. It is powered by a coding language you can touch, and an interface specifically designed for ages 3 to 6.

This age group is ideal to begin a child's journey in computer programming, but one should not have to do so at the expense of important educational areas traditionally learned in hands on play.

LOGO (not Lego), was a milestone in coding education. The goal of Seymour Papert, who created LOGO at MIT in the 1960's, was not just to teach

programming, but also to help children discover their own personal way of solving problems.

Cubetto's coding blocks can be considered an extreme simplification of LOGO. We limited the instructions to their purest form, avoiding any kind of textual or numerical language. The material choice is important: the shell of the interface, and of Cubetto are made of wood, a natural material.

During development, observations were conducted in traditional Swiss kindergartens, where toys and games

made of wood are the most loved by children.

Wooden toys are durable, they have memory, they collect history through marks and scratches, signs of past love and usage.

Wood was also chosen as a material because of the stark contrast it creates with technology, hiding the complexity of the circuit boards beneath the shell.



Coding and computational thinking concepts in early years

Hands on coding

The tangible Blocks have the potential and scalability of any real procedural programming language, and children can learn and play with a variety of core programming concepts.

Algorithms

Algorithms are sets of precise instructions that form a program. Cubetto's Blocks are a physical representation of an instruction that combine to create a program.

The queue

Instructions in programs are executed following a precise order. On Cubetto's Board, they are put together following a line, also a physical representation of the queue.

Debugging

The instructions are laid on the Board. Fixing mistakes is as easy as swapping a block if Cubetto doesn't arrive where he needs to. This is called debugging.

Recursions

Create a subroutine by "packaging" a sequence in the function line, and call it in the queue with a blue block when you need it.



Development and learning in other key areas

Beyond coding

The tactile and collaborative nature of Cubetto makes it an extremely versatile tool for the classroom. Cubetto fosters learning in key development areas that go beyond programming.

Communication

Children practice listening through a range of stories and narratives in relation to Cubetto, accurately anticipating key events and responding with relevant comments, questions or actions. They also develop their own narratives and explanations.

Physical exercise

Children master control and coordination in large and small movements around the playset. They negotiate the placement of obstacles around the world map, and place blocks on our tangible interface.

Social-Emotional

Children become confident by trying new, open ended activities that remove “wrong” outcomes, and easily encourage group work. The open nature of the maps allows them to choose the resources they need for their play session.

Mathematics

Children add and subtract blocks from a sequence. They solve problems, including doubling, halving and sharing to get Cubetto from A to B. They discuss size, identify shapes and patterns, distance, position, and time to solve problems.

Logical reasoning

The blocks allow children to create and debug simple programs with their hands. They use technology purposefully to create, organise, store, manipulate and retrieve meaningful sequences.



First things first

The goal is to get children to create programs for Cubetto by arranging sequences of instructions.

The more time spent playing with Cubetto, the more children develop computational thinking skills. This can be measured by observing them create longer sequences to solve and build more complex algorithms and

problems. The speed with which a child can progress through “mission difficulty” varies from child to child, but it is always important to not skip the introductory steps, no matter how fast a child goes through them.



Introducing Cubetto

Introduce Cubetto as a friendly robot that children can program. Children should be told that Cubetto cannot think for himself, and can only move as programmed by the child, just like any other machine. It's fine at this point to have students sitting in a large circle around a map, and pass a couple of Cubettos around to one another, saying hello or acknowledging its presence. Doing so provides for a great opportunity for the teacher to ask open questions such as "how do you think it moves?" and contributes to create a powerful bond with it which help engage them in solving problems through narratives, which we will see later on.

Introducing the Board

It is important for children to understand Cubetto is only able to move with a human's command. This is not only empowering, but also key to understanding computing.

Introduce the Board as a sort of remote control that children can use to send instructions to Cubetto. Without the Board, there is no way of sending Cubetto his instructions..

Ask the children to explain what other objects in their homes and lives function with a similar paradigm. A television needs a human to change its channels for example, or a washing machine needs a human to select its

settings. All of these examples, like Cubetto, are machines that need human programming to work and do their job.



Introducing the Blocks

Different blocks represent different instructions, and an unambiguous, distinct command.

These blocks are what make up Cubetto hands-on coding language, and are key in the learning of computational thinking.

After the children have been given time to think about how to make Cubetto move and about its connection to the board, show them the blocks. Ask them

to sort them, spot differences and guess how they work. This is a great way for them to learn about formulating an hypothesis and validating a theory, which is part of the exploratory process of writing code.

You can decide to introduce the instruction blocks formally as the directions Cubetto follows when inserted in the Board and sent by pressing the

action button. When each block is inserted in the Board, a child should be encouraged to predict what Cubetto will execute before pressing the “Go” button.

Prediction is key in understanding concepts like program design, and it helps them develop abstraction.



Action causality

The aim of the first session is to introduce the simple notion that sending a command to Cubetto will result in an action. Take the Green Block (Forward), and have a child insert it in the first slot in the Board.

Then have the child press the big blue button on the Interface Board (Go), and observe Cubetto executing the command.

Make sure the child clearly associates the colour of the block with the action performed.

Unambiguous instructions

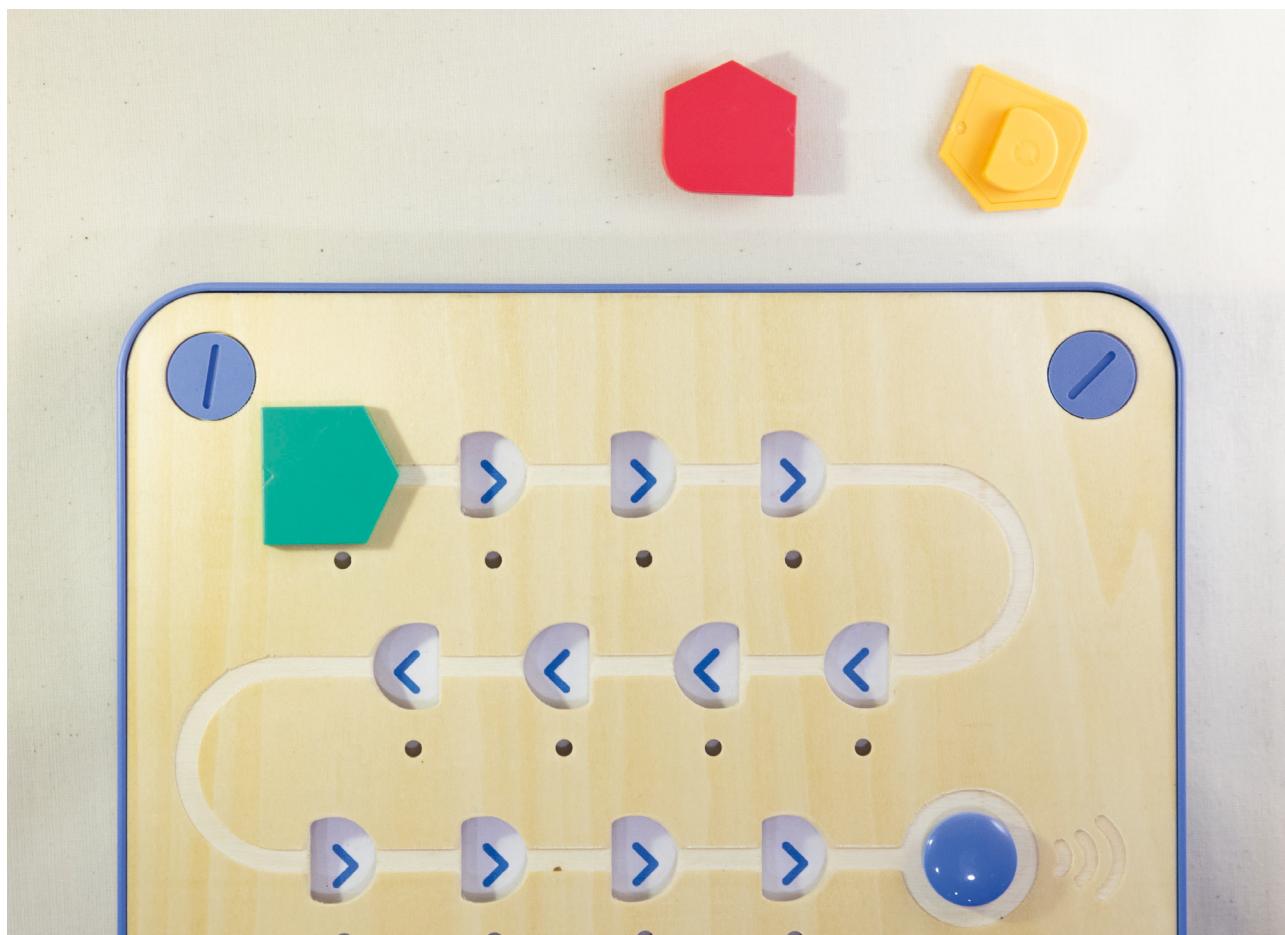
Repeat the previous step with each directional block (except the blue function block), until the children can confidently recognise each block as a distinct and unambiguous instruction.

This is an important step into understanding how a meaningful chain of commands, or a sequence if you will, can be later created to solve a specific problem.

Setting up your classroom

Communication skills are a fundamental, and sharing a Cubetto can give children lots of practice. We found small groups of 2-3 work best, up to 4 for older students who already have got the hang of sharing.

Aim to have as many children as many roles you can assign them. It's important they have a clear understanding of what is expected of them, how they should be working together and who is responsible for what.



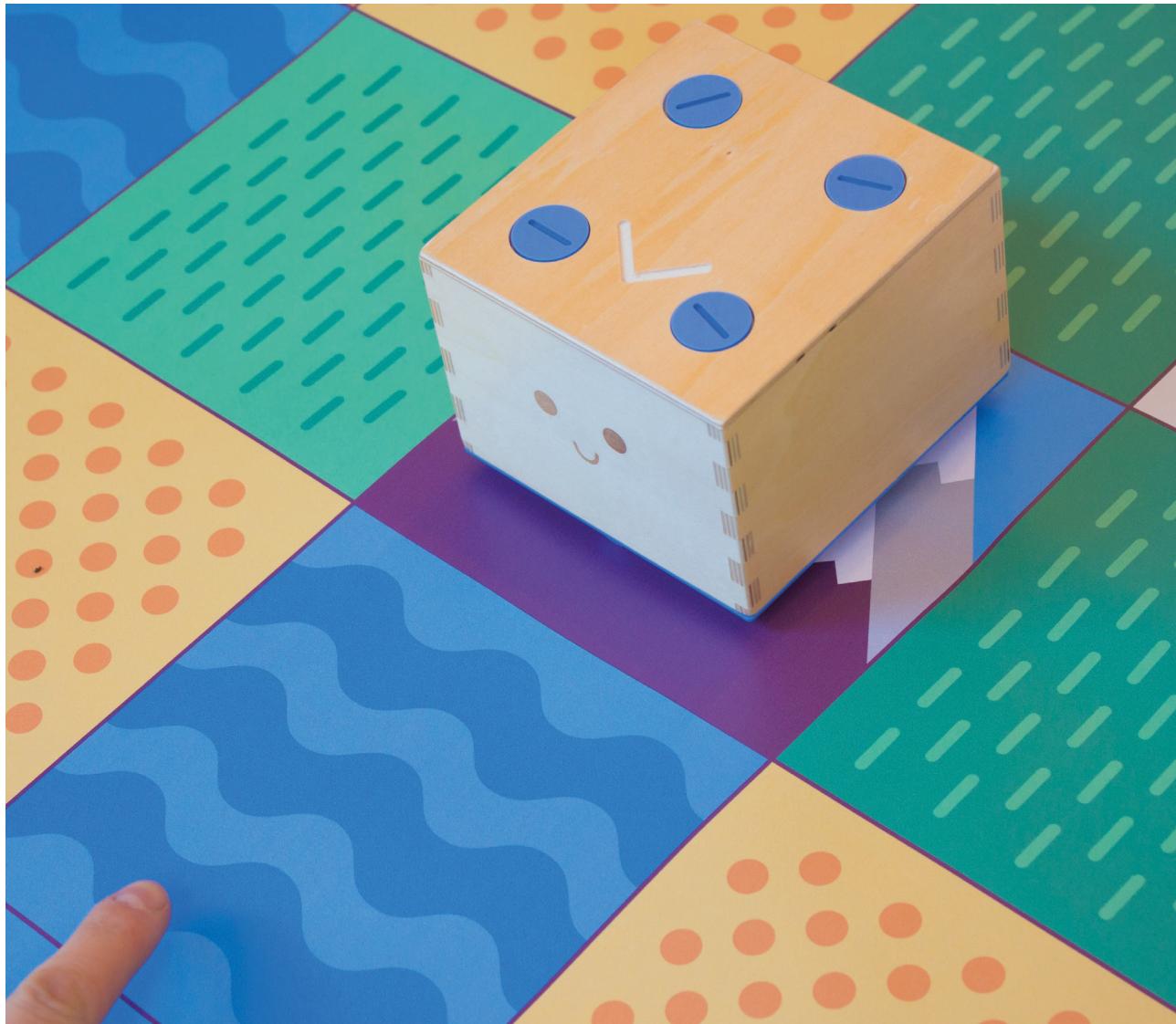
The first challenge

Unfold the map, and place Cubetto on a square. Ask the children to create a program that will get Cubetto to the square directly in front of it. They should be able to reason and tell which single instruction will allow Cubetto to reach his destination.

Let them insert the block in the first slot of the Interface Board, and press the action button.

Don't worry if the wrong block was selected. Just reset Cubetto's position, and encourage the children to reason their choice, voice out what didn't work and try new options.

One-block sequences are also a great way to help your children memorise different commands, so it's perfectly fine to give them a complex challenge for them to solve one block at time.



The queue

As your children will make progress and familiarise themselves with the idea of programming Cubetto, challenge them to compose longer and longer sequences, where the objective is to arrive to the objective with the directive of pressing the "Go" button only once.

As for the first challenge, let them operate the board autonomously. Give them plenty of time to examine the outcome and reason their choice of blocks and the result of that sequence.

Finally, encourage them to try new options.

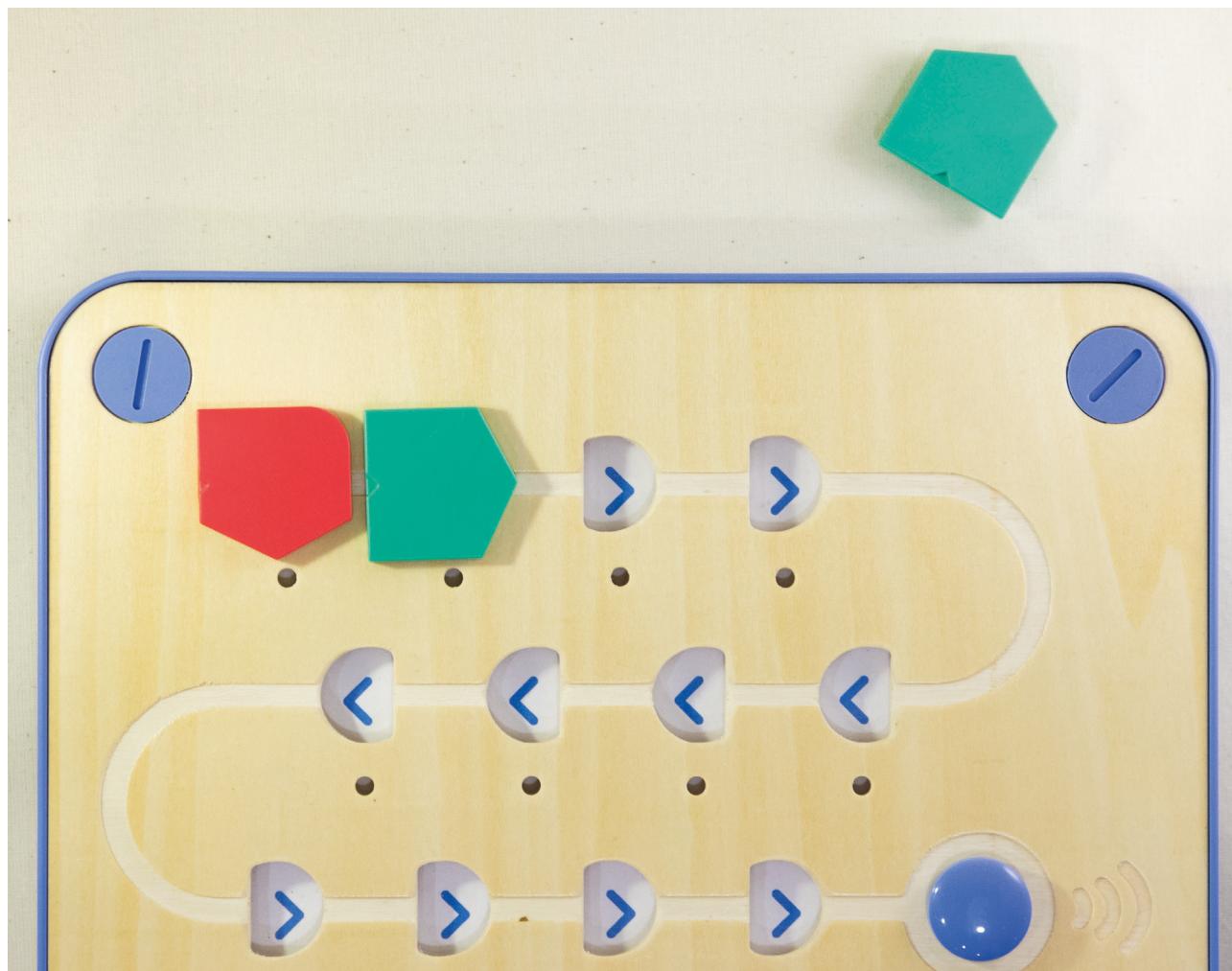
A good example is a L-shaped route, setting the arrival square one square ahead of Cubetto and one square to his left (or right). Let them reason and create the sequence that drives Cubetto to his destination.

Just reset Cubetto's position, and encourage the child to reason his/her choice, and try new options.

Debugging

Debugging is an integral part of learning to code.

As the challenges increase in difficulty your children will find that often they need to go back to their sequence and debug their code to obtain the desired result. The blocks will be a great help, as their algorithm is not only visible, it's tangible too.



Introducing the function

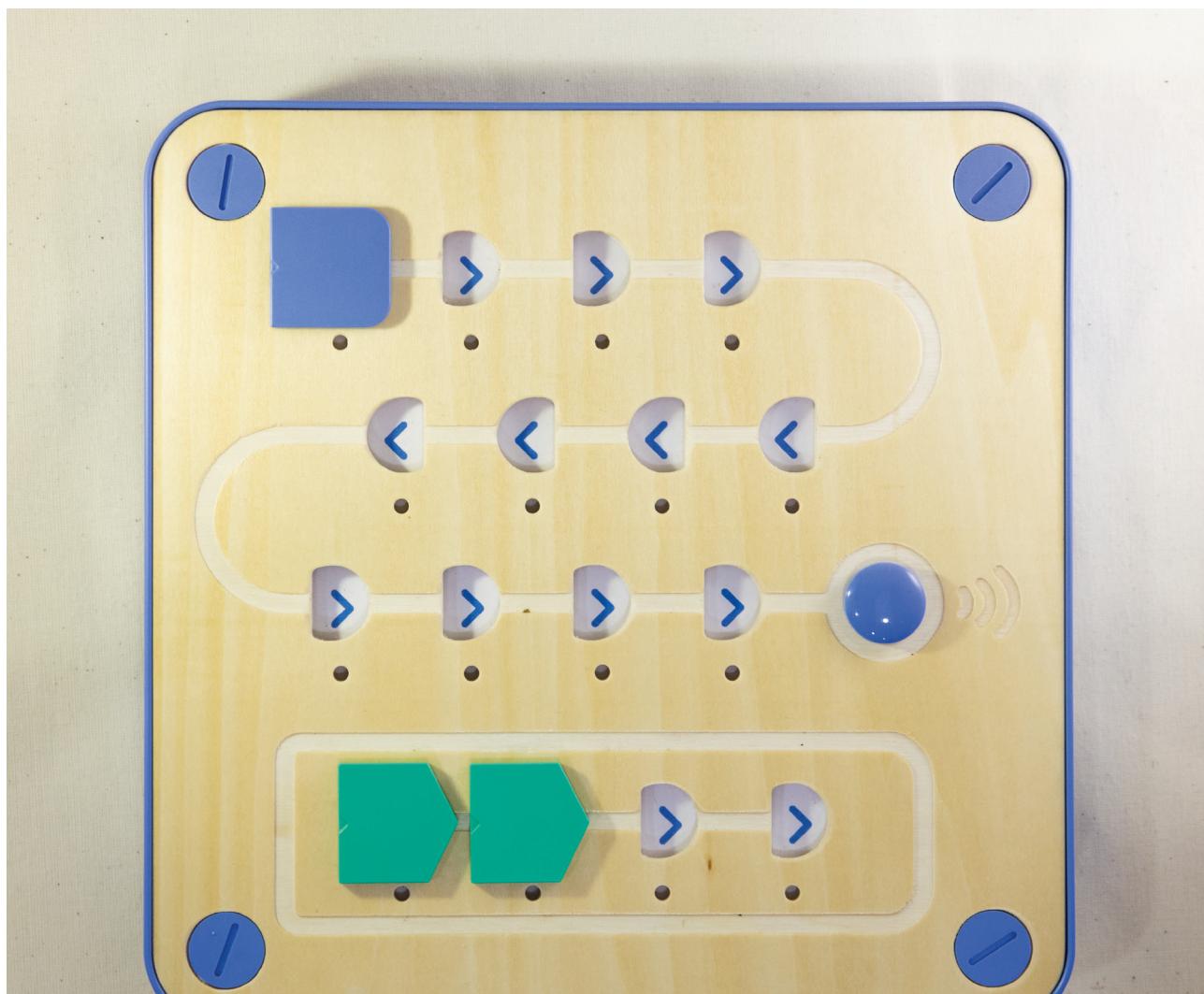
After the child is familiar with the basic blocks, and the idea of an algorithm, it is time to introduce the function block. To explain how it works, you can use the “../pack” metaphor, explaining that it is possible to pack more instructions inside a blue block.

To show this, first place two green blocks in the main sequence and press the Go button.

This will move Cubetto forward by two grid tiles on the map. Now clear the interface Board, and place the two forward blocks in the function line (The last line in the Board) instead, while placing a blue block in the main sequence.

Let the children observe that Cubetto can perform the same actions with two different sequences.

Leaving the same sequence in the function line, place two blue blocks in the main sequence, and let the children observe how two blue blocks cause Cubetto to repeat the function twice.



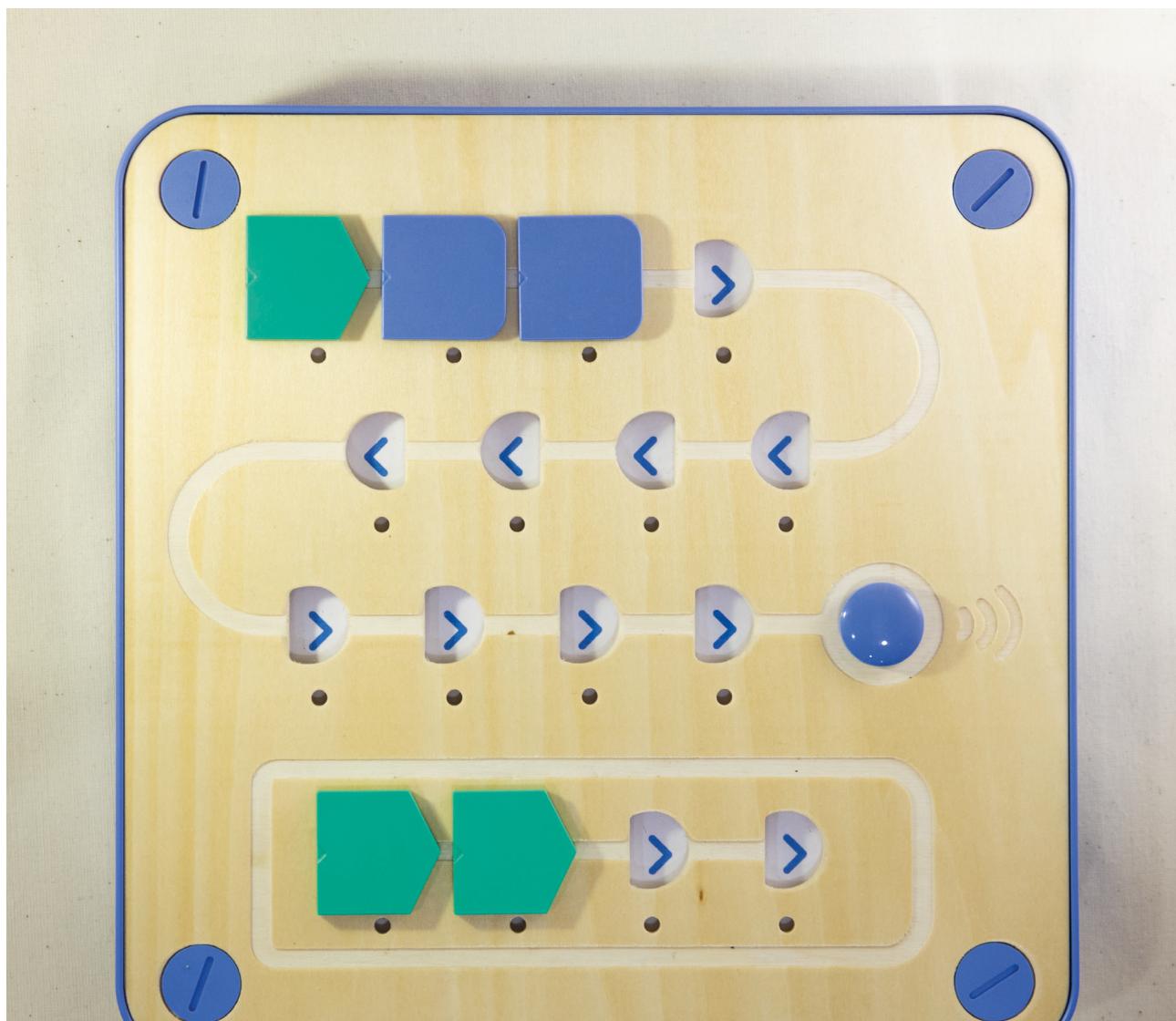
Solving problems with the function

Unfold the map, place Cubetto on the compass, and give the child only the following blocks: 3x Green blocks, and 2x Blue blocks (Function). Ask the child to create a program that will get Cubetto 5 squares ahead of him.

Since there are not enough forward blocks to make this happen, the child should be able to reason that a function

should be created in order to complete the project.

Let the child create the right sequence, including a function, and press the Go button. If the sequence is wrong, just reset Cubetto's position, and encourage the child to reason his/her choice, and try new options.



Additional help and resources

Here is a list of very helpful resources to effectively implement coding with your youngest students:

- Code.org (<https://code.org/>)
- Coding as a Playground, Marina Umaschi Bers, Routledge, 2018
- Computing At School (<https://www.computingatschool.org.uk/>)
- Cubetto Playset Lesson Plans, Primo Toys Education, 2017
- Mindstorms, Seymour Papert, Harvester Press, 1980
- Robotics for Young Children, Ann Gadzikowski, Redleaf Press, 2017
- Teaching in the Digital Age, Brian Puerling, Redleaf Press, 2012
- Teaching in the Digital Age for Preschool and Kindergarten, Brian Puerling, Redleaf Press, 2018

Ultimately, The best judge of what makes a child tick is you. You know your classroom, your environment and your group, which is why we only provide a framework for progression instead of a ready made solution. It's your story, your classroom, your Cubetto.



PRIMO

Visit our website:

www.primotoys.com

Get in touch:

edu@primotoys.com

