Programming from an early age - with Bee-Bots!
First, a polite request...

Please note that Kinderlabor owns the copyright for this script. In the case of use / distribution outside your school or beyond the copyright law, we ask that you obtain permission from us beforehand. Thank you 😊
How do Bee-Bots have to do with IT?

Computer science revolves around the automatic solution of problems. For example, we try to find the quickest connection from A to B at a specific time with the help of a train timetable. People used to do this by looking it up in a printed timetable. Nowadays computers are supposed to relieve us of this task by referring to a digital timetable and showing us the results of the search on a web page or in an app.

The job of IT specialists is to teach the computer how to do that. This is done in two steps. First, an algorithm ("recipe") must be designed, then the computer must be programmed (fed with the recipe) so that it can solve the problem automatically for us by using the recipe.

By using the Bee-Bots, children can find out exactly how this process works in a very simple form. For example, the problem could be that the bee has to get to the flower. The children first consider a suitable path and then program the Bee-Bot so that it runs down that path automatically. The skills required to achieve this are the same as for "big" computer science: forward thinking, accurate work, creativity and teamwork. The Bee-Bot works very much like LOGO, a "real" programming language, which can therefore also be taught with greater ease after programming with the Bee-Bot. Programming with Bee-Bots can therefore be seen as an introduction to computer science.
What is programming?

**Programming** means giving a computer (or a robot) a sequence of commands so that it does exactly what we want it to do.

A **programming language** is a language that the computer understands. It consists of commands that can be used when using the computer.

A **program** consists of one or more command sequences. Programs are written in a programming language and then run by the computer.

**Example:** Bee-Bot has to "fly" to the flower. It understands four different commands: one tells it to move one field forward, one to move backwards, another to turn 90 degrees to the left and the last instructs him to turn 90 degrees to the right.

The path to the flower (for example, forward, left turn, forward, forward, forward) is recorded as a program with the help of the corresponding command button.

After pressing the GO button, the Bee-Bot follows the program thereby reaching the flower as directed.

The programming language of the Bee-Bot is explained in detail on the next page.
The Bee-Bot Language

Bee-Bot understands - via buttons - four different commands:

1️⃣ Move one field forwards!
2️⃣ Turn on the spot to the right!
3️⃣ Move one field backwards!
4️⃣ Turn on the spot to the left!
Learning the Bee-Bot commands

**Preparation:** The children all sit in a row (all facing the same direction and with enough space around them). They can now play robot and carry out commands themselves.

**Procedure:** The instructor presents him- or herself self in front of the children (facing the same direction), shows them the four big command cards one after another, making sure everyone can see them and demonstrates what they mean (one step forward, one step backward, quarter turn to the right, quarter turn to the left).

Then it is the children’s turn. The instructor stands in front of them and shows commands in succession, which each child then has to carry out. After each movement, all the children stop in the new position, and then the procedure continues with the next command. If they all do it right, they should all move together at the same time; at the beginning, the children often get confused by which direction they should be going in, and the children get out of sync. Before going any further, all errors should be identified and corrected together.

**Learning objective:** The children should know the four commands and be able to carry them out.
Using the Bee-Bot commands

**Preparation:** An array made up of 4x4 fields is affixed to the floor with masking tape, so that a child can easily stand on a field. A flower card is placed on one of the fields.

**Procedure:** The instructor explains the task: A child acts as a robot and is steered to the flower by the other children without using words. The instructor sets the starting field and one of the four directions the child is to face. The other children give the "robot" a sequence of command cards that tell it what to do in order to steer it. The robot immediately carries out each command and then the next command is selected.

At each step, the other children check whether the robot is "working correctly" (i.e. carrying out the commands properly) and whether the commands themselves are correct. Commands that steer the robot outside the field are not correct; these commands must be revoked. Commands that send the robot in the wrong direction can be corrected again with the next commands. If the robot reaches the flower, it must be "charged", that is to say another child takes its place (facing the same direction). Another child can then place the flower in a new field and the procedure continues as before.

**Learning objective:** The children can use the four commands for a specific purpose and determine whether the move made by the "robot" matches the command they gave.
First programming

**Procedure:** This time everything is done as before but the robot gets all the commands at once, in the form of a program – i.e. a batch of commands that are executed in a sequence. The instructor explains that it is not practical to have go to the robot after each step. It would be better to give it all the commands at once and then sit back and watch how it does its job. With the help of the children, the instructor creates a suitable pile of cards, as in the example shown on the right, using the following five commands:

1.  
2.  
3.  
4.  
5.

Since the robot is not running while the program is being created, the children have to work out where it would be. This process can be supported by laying out the cards on the field (see next page). The cards are then put into a pile (start with the first card, then the next card under the pile). The robot now gets the finished pile – the program. To carry out the task, it takes the top card from the pile, carries out the corresponding movement and places the card back under the pile facing down (i.e. "command executed"). In the first round, the instructor can act out the role of the robot and demonstrate the sequence. If all the commands are executed, the children check to see if the robot has reached the target destination.
Laying out the cards:

Forwards/backwards cards: *between* the start and target field of each step

Turning cards: *on* the field on which the robot should turn

Learning objective: The children can create a program - a sequence of commands that the robot can use to solve a pre-given task.
First programming

**Variants:** The "robot" is blindfolded before the flower is placed on a field. The children now create the program, and before the robot can see again, the flower is removed but the children remember where it was. The robot then carries out the program, and when it has landed on the correct field at the end, everyone claps “well done” to each other. Alternatively, the robot can also wait in another room and come in when the program is finished. When creating the program, the children should work as quietly as possible (even without words!) so that the robot cannot hear anything about where the flower is being positioned.

**Checks:** If the robot lands on the wrong field, the children need to find out how this happened. Was the program not created properly or did the robot not carry it out properly? To do this, the pile of cards is placed on the field (starting from the top and turning over one card after the other) as shown on the previous page. Now, everyone has to try to find the problem together by running through the sequence again step by step. What is important is that no one is "blamed", the focus here should be on problem solving.

**Learning objective:** When running a program step by step, the children can tell if it is correct or not. They can identify errors (with the program or the robot).
Working with the Bee-Bot

**Preparation:** Every 2-3 children are given a Bee-Bot, a transparent mat (4x4 fields), a beehive, a red flower, a green flower, some obstacles as well as a set of command cards. The children are told that they now have a real robot, but that it works in exactly the same way as the child-robot did before. Instead of making the sequence of cards into a program and giving this to the robot, the commands are typed into the Bee-Bot one after the other. After pressing the green "GO" button, the Bee-Bot will run the program that has been entered. **Important:** The "X" button must always be pressed before the Bee-Bot is given any new commands, so that Bee-Bot "forgets" the old program.

"GO" (run the stored program)

"X" (delete the stored program)

"II" (1 second pause command)
Working with the Bee-Bot

**Procedure:** The instructor explains that Bee-Bot (just like the child-robot before) has to be steered to the flower. The Bee-Bot is placed on the mat (by the instructor or the children themselves). The flower is placed under the mat. The children from each group lay out a series of command cards for their respective task (just like before on the big field, but this time using the small command cards). Each child can take it in turns to type the program into the Bee-Bot (remember to press "X" beforehand!) and then start the Bee-Bot by pressing "GO". The checks and troubleshooting are carried out as before. The command sequence is still pending, so it is possible to check whether the program contains errors or if any errors have been made when typing. A typical mistake is that “X" has not been pressed before typing in the program and the Bee-Bot therefore starts with the "old" program from the previous round.

**Note:** The on/off switch is on the underside of the Bee-Bot as well as a switch that can be used to turn the sounds on and off. When the sound is on, the Bee-Bot provides audio feedback that helps the children (e.g. confirmation of the input). However, if lots of Bee-Bots are running at the same time, the sounds can also be confusing and disturbing. In this case, they should be turned off.

**Learning objective:** The children can use the Bee-Bot and steer it to specific destinations using programs that have been typed in.
Understanding the Bee-Bot

**Preparation:** This is a theory task for deepening the children’s understanding. The solution for this task can then be tested in a practical exercise. The instructor will hand out all or a selection of the following worksheets to the children. Each worksheet is designed to get the children to work out which of two flowers the Bee-Bot will “fly” to with the program shown (from top to bottom). The “backwards” command is also practiced here, as it is not used at all by some children when they play with the Bee-Bot.

**Procedure:** The children use a pencil to circle the flower on the worksheets that they think the Bee-Bot is going to fly to. The solution is then checked by setting up the situation shown on the worksheet and by typing in the program (remember to press "X" before!).

**Learning objective:** The children can correctly interpret short Bee-Bot programs (with "backwards" commands) as a whole.
Is the Bee-Bot going to go to the green flower or the red flower?
Is the Bee-Bot going to go to the green flower or the red flower?
Is the Bee-Bot going to go to the green flower or the red flower?
Setting and solving tasks

Procedure: Now the beehive, second flower and obstacles come into play. The aim is for the children to set tasks for themselves. The instructor explains that first of all the Bee-Bot has to move to one or two flowers from the chosen starting field and then "fly" to the beehive. Obstacle cards can be used to block certain fields, which means that the Bee-Bot must not enter these fields. The level of difficulty can be varied to suit the capabilities of the children. The second flower, obstacles and even the beehive should not really be used with children who still have problems solving the previous task of the Bee-Bot’s route to the flower. Advanced children can work without command cards and program the Bee-Bot "from their heads". Below are three example tasks (easy, medium, difficult). It is left up to the instructor/children to decide whether the beehive can be "flown over".

Learning objective: Children can use the Bee-Bot to solve complex tasks.
Tasks using two Bee-Bots

**Procedure:** Two children are given one Bee-Bot each and a transparent mat. Each child puts his or her Bee-Bot on a field. The aim is to program the two Bee-Bots in such a way that they swap places without colliding into each other. In this instance, the two children must work together and consider the corresponding commands for both Bee-Bots. The instructor can specify the starting fields at the beginning and then later the children can choose these themselves. Depending on the children’s capabilities, the children can also be asked not just to swap their places but to face them in the same direction as the other Bee-Bot at the beginning. Obstacles can also be used here. If there are collisions these can of course be fun and the Bee-Bots do not mind bumping into each other. Here are three example tasks:

![Example 1](image1.png)  ![Example 2](image2.png)  ![Example 3](image3.png)

**Learning objective:** The children can work together to solve complex tasks with the Bee-Bot.
Your turn! Bee-Bot parcours

Procedure: Using the template, the dimensions of which correspond exactly to a Bee-Bot field on the transparent mat, the children can paint a course for the Bee-Bot on the kraft paper and decorate the course how they want. The Bee-Bot can then also run off the course. The children can choose the starting and finishing points, set up obstacles, or, for example, draw a "snake" for the Bee-Bot, which it then has to run along:

Depending on the children’s capabilities, the instructor can point the following out to the children or let them discover it themselves: the Bee-Bot only stays in the fields if they fit together exactly. If there are gaps or offset fields, the Bee-Bot can no longer follow the fields.
Command cards to print off and cut out