CodeWeek
Build your digital future

Nature's Codebreakers: an unplugged family adventure – part 2

By European Parents' Association – EPA

OCT/2025 | EPA

Funded by the European Union

Objectives

Using the example of patterns in nature to introduce concepts of coding in an unplugged way, i.e. without the use of devices, to foster computational thinking and problem solving skills.

Duration

60 min – 2 hrs

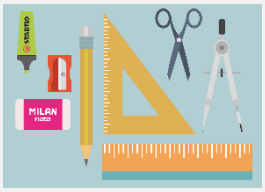
Target group

Different ages for different missions

Experience

Beginner and Intermediate

Materials



Paper and pencil, magnifying glass, camera for documentation, bag or basket to collect materials, outdoor clothing

Description

Introduction

At its heart, computer science is about identifying, analysing, and implementing patterns to solve problems or create systems. Algorithms are sequences of steps: Many natural phenomena follow predictable sequences or cycles, which can be seen as real-world algorithms. Think of the steps a seed takes to grow into a plant, or the way a spider builds its web.

Thinking about patterns in nature as a form of "unplugged coding" is a fantastic way to introduce computational thinking skills in your family without relying on computers. The following activities are designed to spark curiosity and coding skills in your littlest explorers or your grown-up adventurers.

Funded by
the European Union

©https://img.freepik.com/premium-vector/growing-plant-pot-guide-how-grow-flower-step-by-step-instruction_277904-609.jpg?w=2000

Part 2. Advanced level

Welcome back, Super Codebreakers! You've mastered the basics, now it's time to sharpen your skills and uncover even more amazing connections between nature and the world of coding! Get ready for some new missions that will challenge your observation, logical thinking, and creative coding abilities – all without plugging into a single device!

☞ What skills will your child develop in this section?

In addition to the foundational skills – Pattern Recognition, Decomposition, and Abstraction – that continue to be developed in Missions 4-6, these more advanced missions also layer on and emphasize these additional computational thinking skills:

Funded by the European Union

- **Logical Thinking:** Children need to apply logical rules to categorize and group items based on specific criteria.
- **Categorization:** This is a fundamental skill in computer science for organizing data and information efficiently.
- **Algorithmic Thinking:** Even in simple sorting, children are implicitly thinking about the steps involved in placing items into the correct groups. The "Invent Your Own Sort" activity encourages this in a basic way, while other missions directly engage children in designing and writing a sequence of precise instructions to achieve a specific outcome.
- **Binary Representation:** The concept of representing information using only two states is the bedrock of how computers store and process data.
- **Encoding and Decoding:** Children learn to translate information into a binary format and back again, a core skill in communication and data transfer.
- **Symbolic Representation:** They understand that natural elements can stand for abstract concepts (like 0 and 1 or letters).
- **Precision and Detail:** They learn the importance of clear and unambiguous instructions, as even small errors can lead to a different result.
- **Testing and Debugging (Implicit):** When someone else follows their algorithm, they can see if it works as intended and identify any "bugs" or unclear steps.
- **Conditional Logic (Introduction):** The last "Master Minds" challenge introduces the concept of "if-then" statements, a crucial part of programming logic.

Mission 4: The Great Sorting Expedition! (3+)

In the world of computers, sorting things into groups is super important! Programmers need to organize information so it's easy to find and use. Guess what? Nature is a master sorter too!

Your Challenge: Head outside and collect a variety of natural items. This could include different types of leaves, stones of various sizes or colors, different kinds of flowers, or even fallen seeds and twigs.
Your Task: Now, put on your sorting hats! Can you find different ways to group your collection?

Funded by
the European Union

- **Sorting by Shape:** Group all the round leaves together, the pointy leaves together, the long thin leaves together.
- **Sorting by Size:** Create groups of small, medium, and large stones or leaves.
- **Sorting by Colour:** Separate your collection by the different colours you find.
- **Creating Your Own Rules:** Can you come up with your own unique way to sort your items? Maybe by texture (smooth vs. rough) or by where you found them?

Codebreaker Tip for Little Explorers:

Start with just one or two sorting rules at a time. "Let's put all the green things together!"

Codebreaker Challenge for Junior Agents:

Try sorting your collection using two rules at once! For example, "Let's make groups of small green leaves and big brown leaves."

Codebreaker Quest for Master Minds:

Can you create a "flowchart" using arrows and your natural items to show the steps someone would take to sort your collection based on your rules?

Debrief:

How many different ways could you sort your collection? Why is sorting important in everyday life and in computers?

Funded by
the European Union

©avashist1998.github.io

Mission 5: Nature's Binary Messages! (ages 6+)

Did you know that computers talk using only two signals: on and off, which we often represent as 0 and 1? This is called binary code! We can find simple "on/off" or "yes/no" patterns in nature too!

Your Challenge: Look for things in nature that have two clear states or options. Here are some ideas:

- **Leaves:** Present or absent on a branch.
- **Flowers:** Open or closed.
- **Stones:** Light-colored or dark-colored.
- **Twigs:** Pointing left or pointing right.

Your Task: Create secret messages using these binary choices!

- **Simple Sequences:** Assign one state to "0" and the other to "1". For example, a leaf present = 1, leaf absent = 0. Can you create a short sequence like 101 (leaf, no leaf, leaf)?
- **Picture Messages:** Use your chosen natural "bits" to represent letters or simple pictures. You'll need to agree on a simple code beforehand (e.g., maybe 3 "bits" can represent a direction: 001 = up, 010 = down, 100 = left, 011 = right).

Codebreaker Tip for Little Explorers:

Start by just identifying things that have two clear states. "The flower is open!" or "There's no leaf on this spot!"

Funded by
the European Union

Codebreaker Challenge for Junior Agents:

Try to create a short message using a simple binary code you invent. Can you get someone else to decode it?

Codebreaker Quest for Master Minds:

Research the Morse code! It uses short and long signals to represent letters. Can you adapt this idea using natural sounds (like tapping a stone lightly or heavily) or visual cues (like a small leaf or a big leaf)?

Debrief:

Was it easy or hard to create and decode your messages? How is this similar to how computers store and transmit information?

Mission 6: The Amazing Algorithm Trail! (8+)

An algorithm is a set of step-by-step instructions to solve a problem or achieve a goal. We saw some of nature's algorithms earlier, but now it's your turn to create one in nature!

Your Challenge: Design a simple "obstacle course" or a path in your outdoor space using natural elements. This could involve going around a tree, stepping over a log, touching a certain flower, etc.

Your Task: Write down the exact steps someone needs to follow to complete your trail. Be very precise! This is your algorithm!

- **Clear Starting Point:** Where does the trail begin?
- **Specific Actions:** Use action verbs to describe each step (e.g., "Walk three steps forward," "Turn left at the big rock," "Hop over the fallen branch").
- **Clear Ending Point:** Where does the trail end?

Funded by the European Union

Codebreaker Tip for Little Explorers:

Keep the trail very short and simple with only a few steps. Use pictures instead of words if that's easier!

Codebreaker Challenge for Junior Agents:

Make your trail a bit longer and include different types of actions. Test your algorithm by having someone else follow your instructions. Did they end up in the right place?

Codebreaker Quest for Master Minds:

Can you add "conditional" steps to your algorithm? For example, "If you see a red flower, turn right. Otherwise, continue straight." This is like an "if-then" statement in coding!

Debrief:

What was the hardest part about writing your algorithm? What happens if you miss a step or make a mistake in your instructions? How is this similar to writing code for a computer?

Remember, Nature's Codebreakers, every time you explore, observe, and create in the natural world, you're building important skills that are just like the ones used by computer programmers. Keep your curiosity alive, and who knows what other amazing

Here is the International Morse [illegible] for your "Nature's Binary Messages" mission:

Letter Morse Code

A ·–

Funded by the European Union

B	−···
C	−·−·
D	−··
E	·
F	··−·
G	−−·
H	····
I	··
J	·−−
K	−·−
L	·−··
M	−−
N	−·
O	−−−
P	·−−·
Q	−−·−
R	·−·
S	···
T	−
U	··−

V	···–
W	·––
X	–··–
Y	–·––
Z	––··

Funded by
the European Union

CodeWeek
Build your digital future

Legal Disclaimer

The European Commission's support to produce this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Project funded via a Digital Europe Programme grant 101158834

Funded by
the European Union