

Book corner

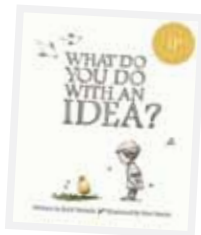


Lift-the-flap Computers and Coding by Rosie Dickins and Shaw Nielsen (Usborne)

A child-friendly intro to how computers are built, what they do and how they work.

What Do You Do With An Idea? by Kobi Yamada and Mae Besom (Compendium)

Lovely picture book about how a child brings a bright idea into being.



The Most Magnificent Thing by Ashley Spires (Kids Can Press)

A story about a young girl's determination to persevere and succeed.

Hello Ruby: Adventures in Coding by Linda Liukas (Feiweil & Friends)

Coding and computational thinking explained in a child-friendly way. A good book to refer to.



Coding for Beginners: Using Scratch by Rosie Dickins, Jonathan Melmoth and Louie Stowell (Usborne)

Helps demystify this aspect of computing for practitioners who would like to know more about what lies ahead for their children.

Using ICT in the Early Years by Alex Morgan and John Siraj-Blatchford (Practical Pre-School Books)

How ICT promotes creative thinking.

Your Guide to Outstanding Early Childhood Practice in ICT by Hui-Yun Sung, Natalia Kucirkova and John Siraj-Blatchford (Practical Pre-School Books)

Matched to the 2014 computing curriculum.

READING FOR CODING?

If any children in your setting are ready to be introduced to writing computer code, check out the computer science courses by Code Studio (<https://studio.code.org>) as these are suitable for children as young as four years old. The activities introduce the skills involved in 'computational thinking' as children write their own code and create games and stories.

Further reading

Embracing future developments, <http://bit.ly/1T2LlaO>

Computing at School's Barefoot programme, <http://bit.ly/ZsbKz3>

Let's explore... computation

Children can be introduced to the basics of coding at a very young age, says *Marianne Sargent*

The introduction of the computing programme of study in the 2014 English National Curriculum changed the landscape of ICT teaching in schools.

The teaching of ICT is no longer focused solely on helping children to become competent users of technical equipment and software. Children are now learning about ICT with a view to becoming the future creators of their own programs, systems and content.

One of the central aims of computing in the National Curriculum is to teach children about 'computational thinking', or in other words, ensuring that 'pupils can analyse problems in computational terms'. Children in Key Stage 1 are now being taught about the thought processes involved in the logical reasoning that is needed to solve problems, devise procedures and better understand systems.

This has implications for Foundation Stage practice. It is no longer enough to introduce young children to technology and exemplify its uses. Practitioners need to understand the skills that underpin 'computational thinking' and put in the groundwork so that children are prepared for what they will encounter at primary school.

However, as Professor John Siraj-Blatchford points out, early years practitioners can consider themselves to be 'ahead of the game'. The teaching of creative and critical thinking skills is already integral to Foundation Stage practice.

So, let's take a closer look at 'computational thinking' along with some examples and ideas for activities that will help children develop the skills that underpin it.

LOGICAL REASONING

Future computer programmers need to be able to identify and fix problems (bugs) in theirs and others' code. Logical reasoning involves being able to think logically – for instance, identifying patterns, making generalisations and making predictions.



Practise this skill

Get young children thinking logically by planning activities and challenges that involve making predictions, testing ideas and using what they learn. All the time, ask them what they think, why they think so and what they have seen and learned about this before. For example:

- test materials to see which float and sink and then challenge the children to build a boat that floats
- test materials to see which are waterproof and then challenge the children to build a rain shelter
- test materials to see which float in the breeze and then challenge the children to make parachutes for toys
- build ramps of different gradients on a range of surfaces and test different toy vehicles on them. Then challenge the children to work out which vehicle will travel the furthest on which gradient of ramp and landing on which floor surface.

DECOMPOSITION

This means breaking things down into more manageable steps or stages. In computing this is important because programmers need to devise systems that can tackle large tasks and solve complicated problems.

Practise this skill

Break learning tasks and activities into stages and explain how and why you are doing so to the children. Present

al thinking



them with tasks and challenge them to break each task down to make it more manageable. For example:

- **Plan a picnic** Decide what to eat, write a shopping list, buy the food and prepare it. Break the preparation part of the task up into separate activities such as making sandwiches, preparing a salad and making cupcakes. Take photos of each stage and use these to create pictorial instructions that take you through how each task is done.
- **Recreate a brick model** Challenge a group of children to build something out of bricks. Then ask another group to build something that is exactly the same. Encourage them to take the other group's model to pieces and help by using a digital camera to photograph each stage. Encourage the other group to refer back to these photos as they try to build their replica.

ALGORITHMIC THINKING

This is the ability to use your knowledge of sequences and rules to solve problems. Algorithms transfer knowledge of rules to different but similar situations. They are used to tackle the same or similar tasks over and over again.

Practise this skill

Children use algorithmic thinking when they solve mathematical problems and follow sequential steps to complete a task. In the early years, this

means teaching children to recognise, continue and create patterns in a variety of ways. The following activities will prepare children to be able to use algorithmic thinking later on:

- Make visual patterns out of objects of different sizes, shapes and colours.
- Correct and alter patterns that have been created by others.
- Learn about patterns in numbers – for example, doubling and halving, multiplying and dividing sets of objects.
- Build a series of towers that double in height.
- Challenge children to give sequential instructions – for instance, set out clues for a treasure hunt or create instructions to build a model out of Lego.

ABSTRACTION

Abstraction means to simplify a problem so that it is easier to think about and tackle. Computer programs are designed to tackle problems by deciding what details they need to consider and what they need to ignore.

Practise this skill

Teach children to home in on the information they need and ignore the superfluous. In the early years, this means starting with sorting and classifying. Try the following challenges:

- Provide a selection of coats, with and without hoods, waterproof or not, in different colours and with

Computational thinking means being creative and logical to solve problems and tackle tasks

Early years practitioners are 'ahead of the game'. The teaching of creative and critical thinking is integral to Foundation Stage practice

different fastenings. Ask the children to choose the best coat to wear out in the rain. Help the children simplify this problem by first discarding all coats that are not waterproof. Then discard those that do not have hoods.

- Set out a selection of pretend food containing fruit, vegetables and meat. Sit three bears on a picnic blanket and explain one of them is vegetarian. Challenge the children to divide up the food so that they all have a fair share but the vegetarian is not given meat. Help the children simplify this problem by first separating out the vegetarian food. Then divide the meat between two of the bears before giving the vegetarian bear an equivalent share in fruit and vegetables.

EVALUATION

Evaluation involves considering whether something has worked or not. It is about asking whether something is any good at serving the purpose for which it was intended. Computer programs are designed for specific purposes. Programmers need to be able to evaluate programs and decide whether they need improvement and in what way they should be improved.

Practise this skill

Encourage young children to reflect on their learning and consider how they would do something differently in the following ways:

- Challenge the children to build a bridge strong enough to support one child as they walk across. Does the bridge serve its purpose? If not, why not? What would they do to change it?
- Challenge the children to make bubble blowers out of different materials such as pipecleaners, coat hangers, wire and plastic bangles. Which work best and why?
- Provide a Bee-Bot, set out a landscape full of obstacles and challenge the children to program him to take a certain route to a particular destination. Did he follow the correct route? Did he arrive at the destination? If not, why not? What was wrong with the programming? What should the children do differently on the next attempt? ■

Marianne Sargent is a writer specialising in early years education and a former Foundation Stage teacher and primary and early years lecturer.