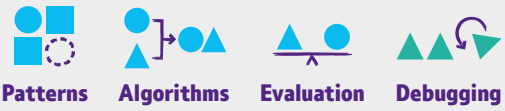
Creating patterns

Overview

In this activity, children spot and create patterns. Children develop their understanding of algorithms (a specific sequence of instructions or rules) by writing instructions for others to recreate their patterns. They then check if their algorithms work through evaluation and debugging.

Concepts:



Age group: 5 – 7

Duration: 15 – 45 minutes

Materials you will need:

- ☑ Printable pattern cards (Example and challenge cards)
- ☑ First, then, repeat cards

These cards can be downloaded, but you may prefer to use coloured bricks, counters, beads, pens and paper or anything you have round the house.

What will your child/children learn?

Algorithms – An algorithm is a precise sequence of instructions, or set of rules, for performing a task.

Debugging – Debugging is about finding out what is wrong in an algorithm or program and fixing it.

Patterns – By spotting patterns we can make predictions, create rules and solve other problems.

Evaluation – We use evaluation when we make judgements based on different factors, including the end result.

The behaviours **creating, persevering, collaborating** and **tinkering** (changing things to see what happens), are approaches to learning that are encouraged throughout our home activities.

Getting started

Explain to your child/children that they are looking for patterns.
Patterns can be similar colours, repeating combinations or groupings of whichever item you have chosen.

1) Show a simple pattern using two colours. **See example patterns.pdf**

2) Ask: What colour will come next?

3) Ask: Can they explain how to make this pattern?

4) Repeat with another pattern to check their understanding.

5) Explain: A pattern is a sequence of things that are arranged following a rule – if we can predict what comes next, it is a pattern.

You could look around the house for something with a pattern, perhaps a cushion or wallpaper to help bring patterns to life.

Their turn

1) Give your child/children a number of patterns and ask them to carry on the sequence. Encourage them to use the cards or coloured items you have chosen.

2) Ask your child/children to explain the patterns they have made. How did they know what came next? If it is out of sequence, can they look and explain why.

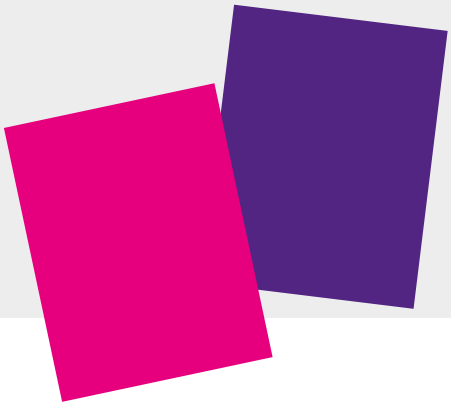
3) Ask your child/children to make their own pattern. They can use beads, counters, toys, building bricks or other items you have at home, or they can draw shapes on paper, or different coloured circles to create a pattern.

4) Show them what to do first. Use 5-8 items to start and ask them to create a short sequence to begin with. It's important to remind children that the sequence is a pattern i.e. If there are two or three beads on the string, they should repeat the same order with the next two or three beads.

Creating an algorithm for their pattern

1) It's now time to use the **'first'**, **'then'** and **'repeat'** instruction cards (see **'materials you will need'**) plus the objects (e.g. beads) to develop an algorithm (a precise set of instructions) that someone else will use to recreate the pattern they have made.

Can you or another child follow the instructions to replicate the pattern?



2) Ask: your child/children to evaluate their instructions: Did they work? How do they know?

If not, can they find any errors in their instructions and correct it – this is called debugging.

Time to talk

1) Look at successful examples of your child's/children's patterns.

2) Use the key vocabulary for this activity: pattern, sequence, repeat, first, then

3) Explain that the instructions they have made are algorithms – a precise sequence of instructions for performing a task.

More ideas

- Keep patterns short and simple to start with, for example alternating colours or shapes. Children may need support creating their own pattern.
- Encourage children to create longer sequences (patterns) and write more complex algorithms to stretch their understanding.

This activity is based on a resource that was designed in conjunction with Michael Haigh, teacher of computing.