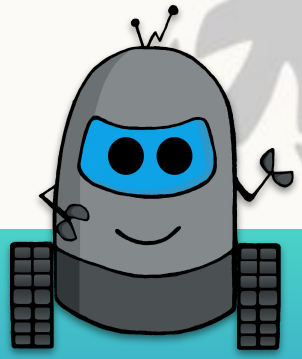


Play Code Learn

DINOSAUR COMMANDS

Lesson Three:

Writing efficient algorithms

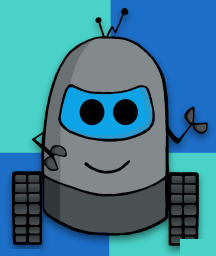
Lesson Three Learning Outcomes

Learning Intention:

...how to create efficient algorithms.

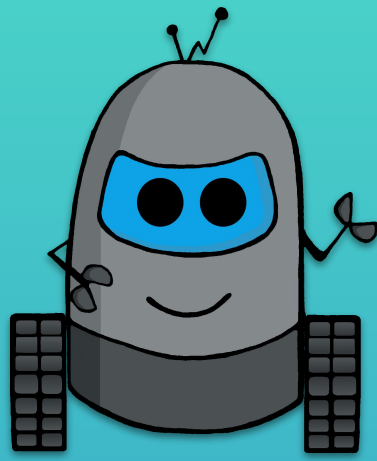
To use algorithmic thinking.

To use pattern recognition to complete a task.



To practise using repeats & loops.

To write, follow, test and debug algorithms.



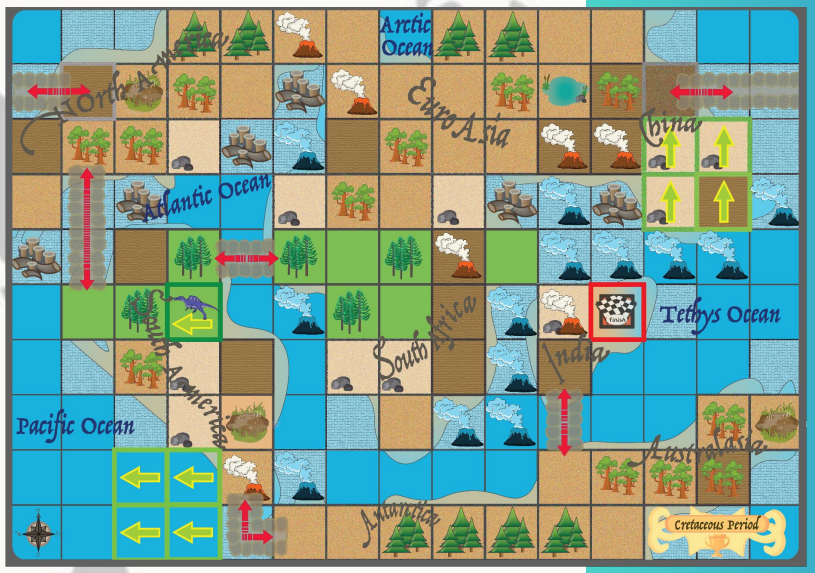
Using repetition

Activity: Decorating Dinosaur Trees Art

In this activity you are going to follow an **algorithm** to complete a task!

Did you know?

During the Cretaceous period the climate was temperate (mild) and so in the polar regions there was lots of conifer forest!

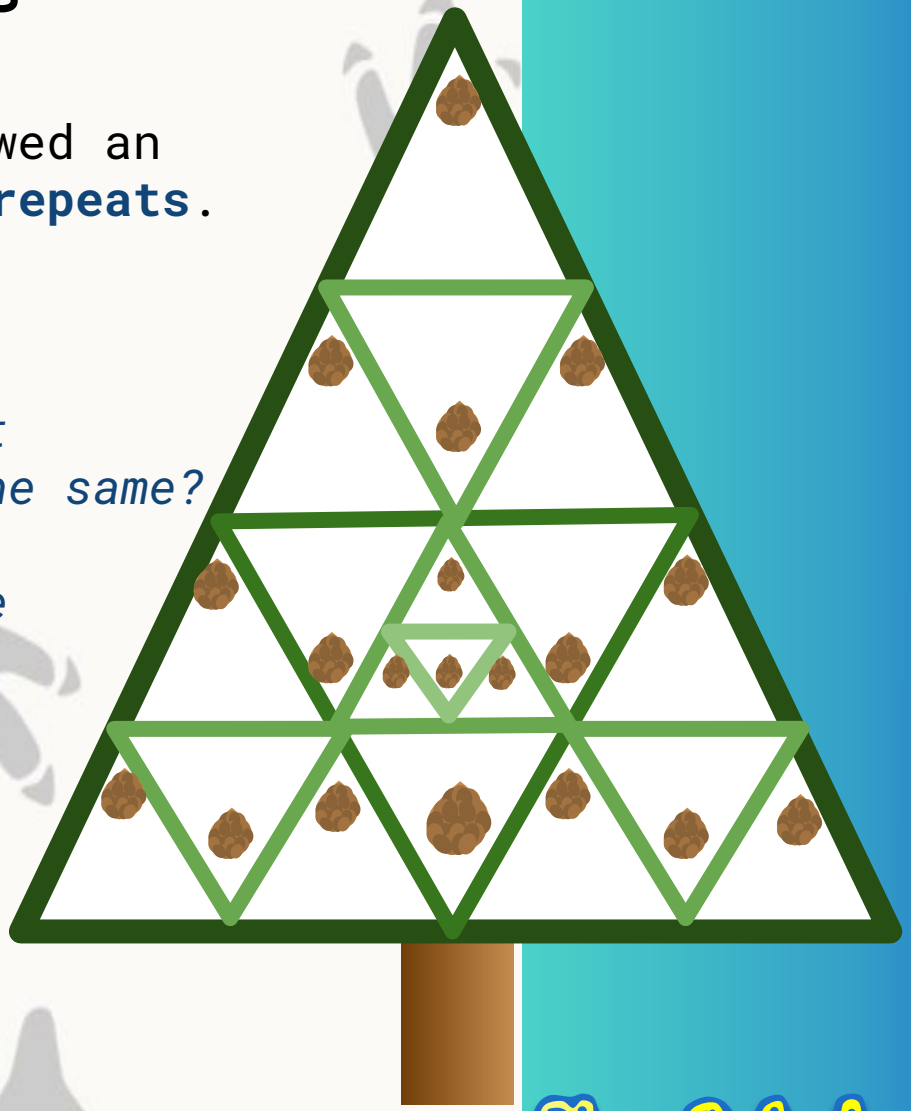


Look at the map - can you spot the conifer forests?

Discussion: Using repeats

In the tree activity you followed an **algorithms** and started to use **repeats**.

- *What happened?*
- *Did you all create different patterns or were they all the same?*
- *Did you find an error in the instructions?*

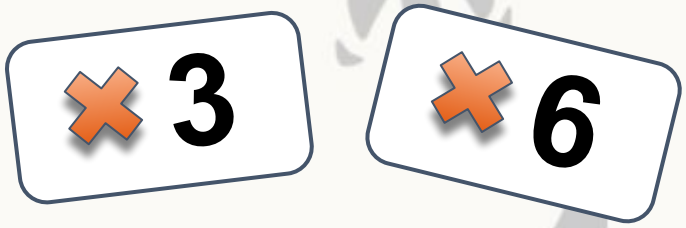


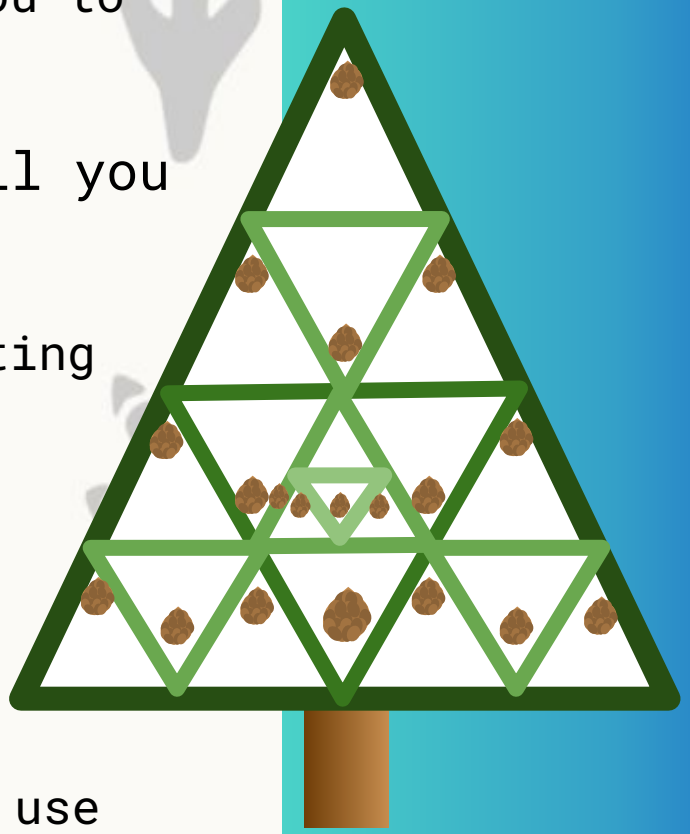
Discussion: Using repeats in algorithmic thinking

Did you spot that the **algorithm** told you to repeat the instructions ***forever***?

There was not an instruction to tell you to **stop**!

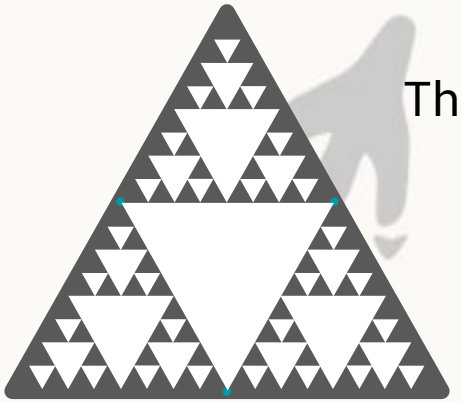
It is important to learn that when writing code you must always be very clear and specific.





In the Dinosaur Commands kit you will use these command cards to tell Explorer Ed how many times to repeat a **command** or a series of commands (**function**).

Extension Activity: Fun with fractals



The dinosaur tree art is a great way to explore fractals and repetition.

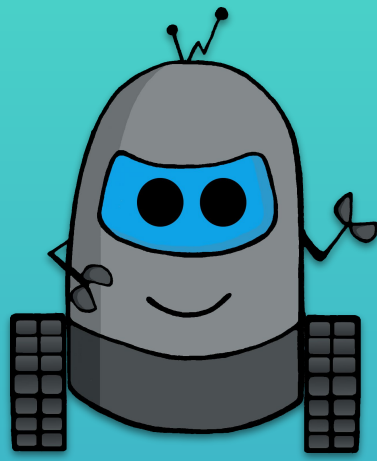
You can take this further by trying to make a Sierpinski triangle.

You will need:

- A triangle template
- A ruler and pencil
- Colouring pens, pencils or crayons.
- Scissors

Make your own Sierpinski triangle, cut it out and then join it to the class's triangles to make one huge Sierpinski triangle!

How big can you make a repeated pattern?



Using iteration to improve algorithms

Loops

Activity: Loopy dice

In this activity you are going to use one of the dice from the Dinosaur commands kit.

Let's practice following a **loop**!

Find a partner and follow this instruction:

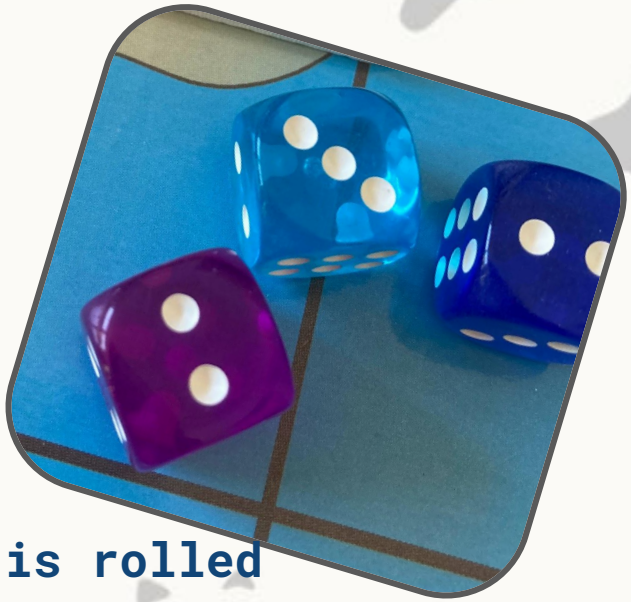
DO Star Jumps WHILE the dice roll is less than 6

What does this mean?
One person in the pair starts to do star jumps.
The other person rolls the die.

IF the die lands on 1, 2, 3, 4, or 5 **THEN** you continue to do star jumps.
IF the 6 is rolled **THEN** stop doing star jumps!

Play Code Learn

Loops



Activity: Loopy dice

Let's have another go!

Follow this loopy instruction:

DO a sit up each time the die is rolled
UNTIL you have done four.

What does this mean?
One person rolls the die. Each time the die is rolled the other person needs to do a sit up.

IF the die is rolled **THEN** do a sit up.
STOP doing a sit up after 4 rolls of the die.

Remember to swap roles :)

Extension activity: Make up your own loopy instructions with the dice. You can add more dice to make it more difficult! Test them out.

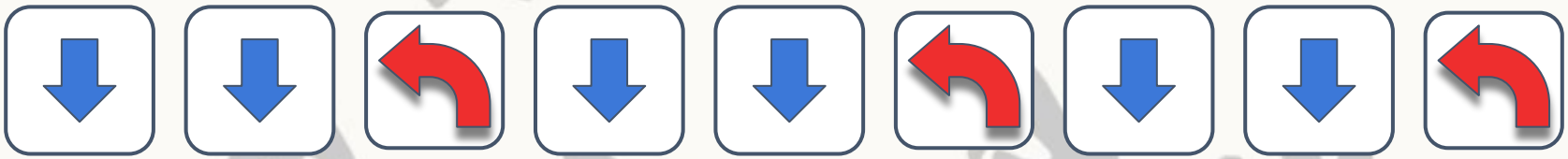
Where should we loop?

Activity: Loopy patterns

To be able to decide where you need to use a **loop** in an **algorithm** you need to use **pattern recognition** to help you.

Look at these examples - what is the repeated pattern? How many times does it repeat?

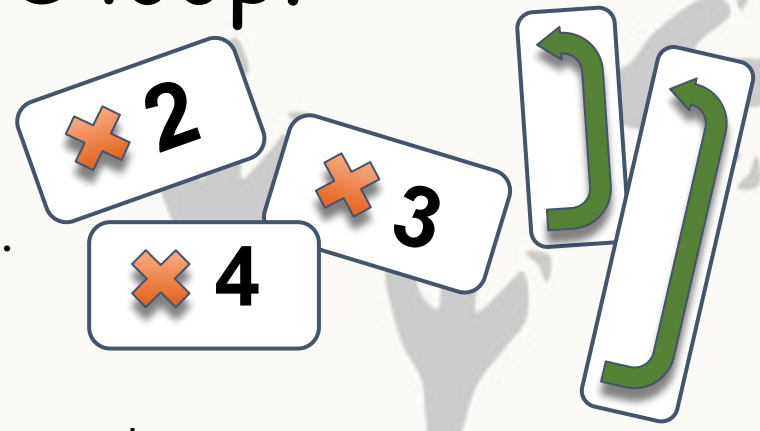
1



2

3

Where should we loop?



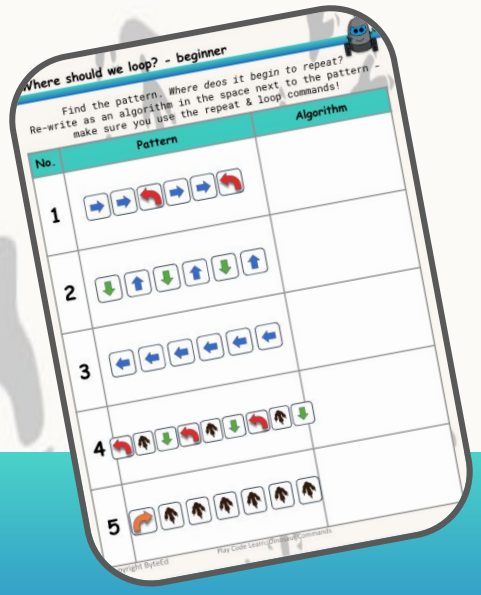
Activity: Loopy patterns

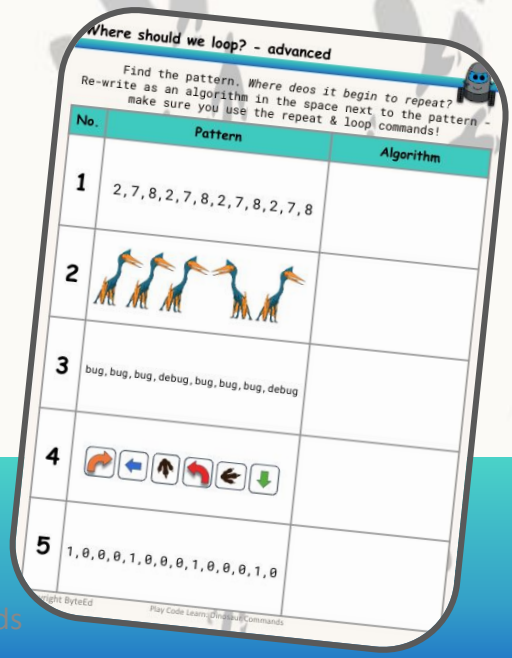
Have a go yourself at **pattern recognition**.

Look carefully to spot the pattern.

Check and see how many times the pattern repeats.

Re-write the pattern as an algorithm using the **repeat** & **loop** commands card symbols from the Dinosaur Commands kit.

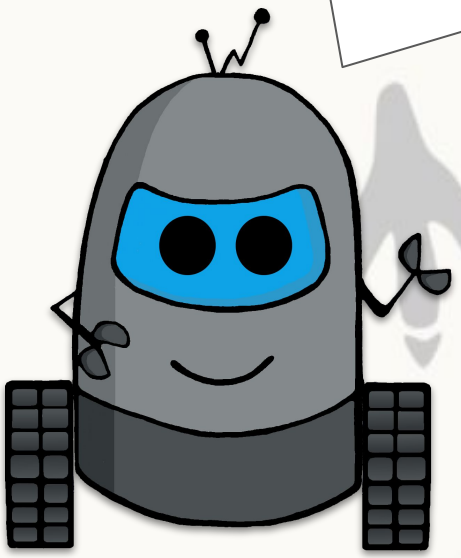


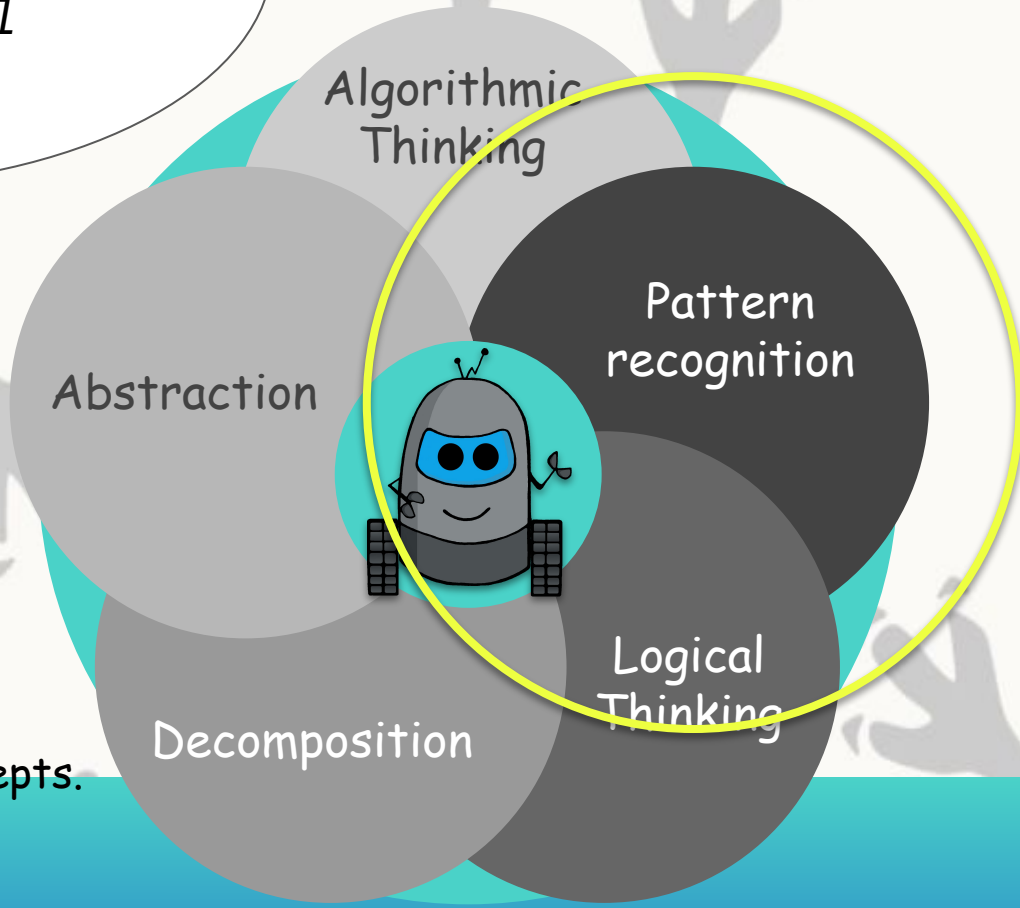


Play Code Learn

Pattern Recognition

Pattern recognition *is another important skill in computational thinking!*



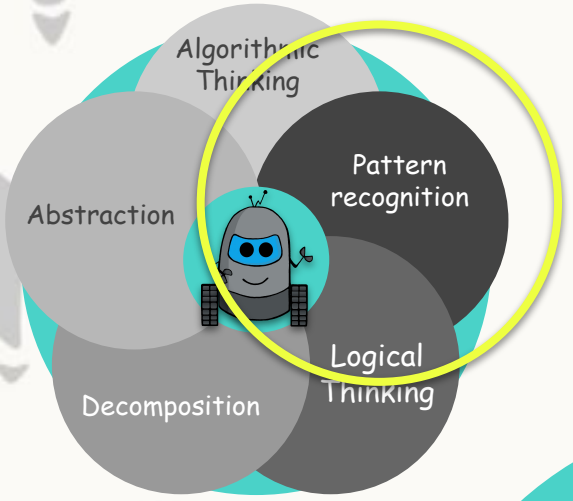


Computational thinking concepts.

Play Code Learn

Pattern Recognition

We can repeat **patterns** to help write programs.



The more **patterns** we find the quicker we can solve the problem.

Pattern recognition makes algorithms more efficient.

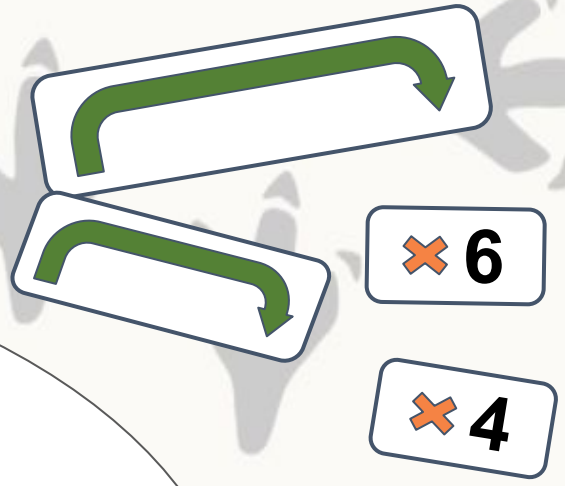
Patterns can make our task simpler.

Problems are easier to solve when they share **patterns** - we can use the same problem-solving solution.

Play Code Learn

Iteration

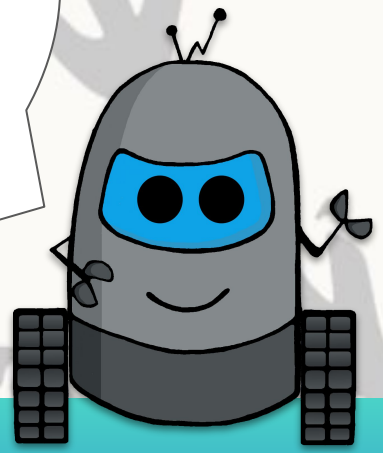
Discussion: **Why use iteration?**



Programmers need to consider the users of their programs and algorithms.

They need to make algorithms and programs fast and efficient to save time and energy use.This can be done by adding **repeats** and **loops** into their code.

By doing this they are adding **iteration**.



Iteration

Activity: Play Code Learn: Dinosaur Commands

“Going Loopy!”

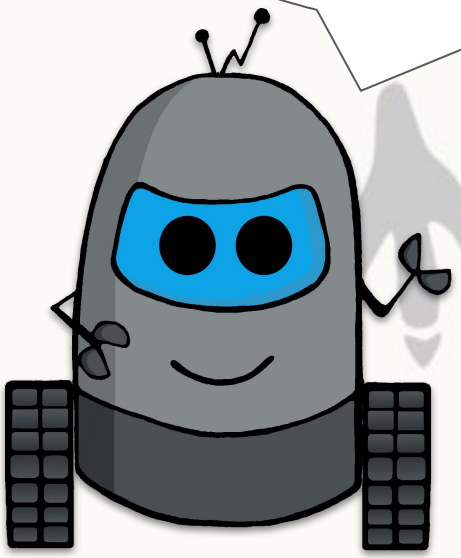
Read the Dinosaur Commands booklet pages 9 to 16.

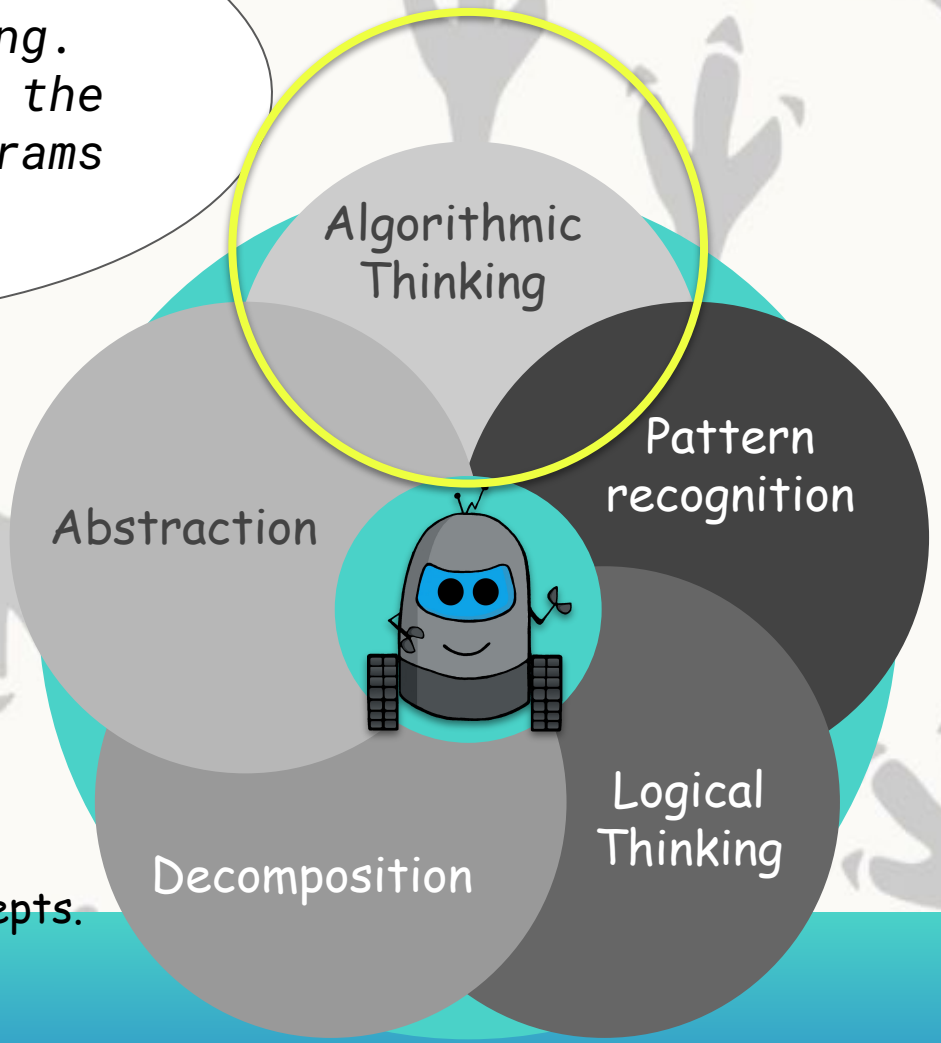
Complete the challenges in the unplugged challenge booklet to practice writing algorithms using **repeats** & **loops**.

Play Code Learn

Iteration

Iteration *is part of algorithmic thinking. It helps to improve the algorithms and programs that you write.*

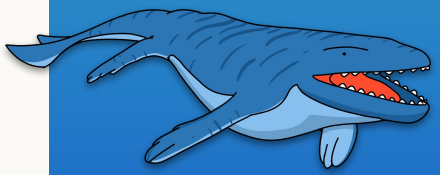




Computational thinking concepts.

Play Code Learn

Extension Activity

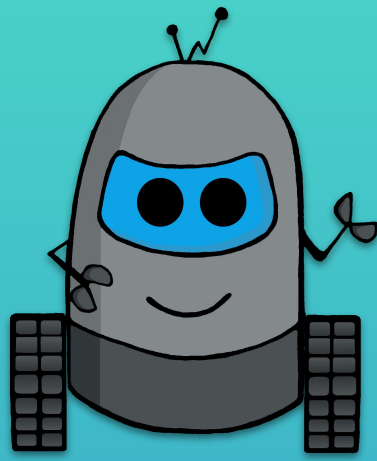


Design your own challenges!

Practice writing iterative algorithms using the Dinosaur Commands kit by setting your own challenges for your partner to complete.

Or, if you are working on your own, see how many different algorithms you can write for EACH unplugged challenge - there is always more than one way to solve the problem!

Which one is the MOST efficient?

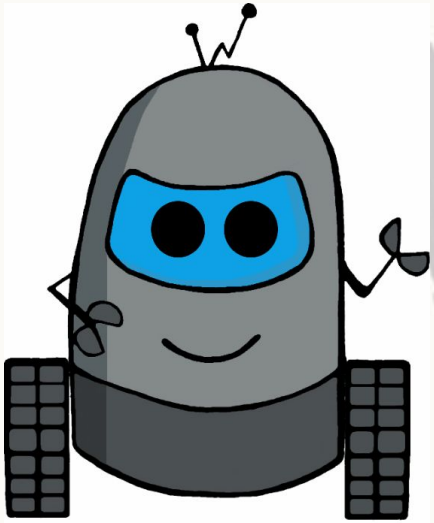


Reflection

Reflection: Lesson Three

Learning Intention:

...how to create efficient algorithms.



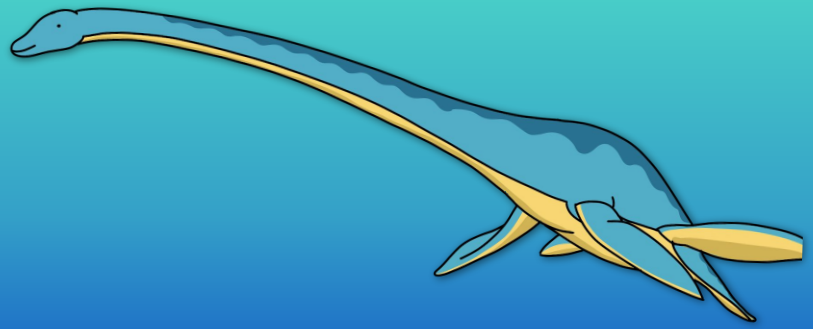
How do you feel about today's lesson?



What were the key takeaways from the lesson today?

What would you like to learn more about?

Play Code Learn



Thank you!