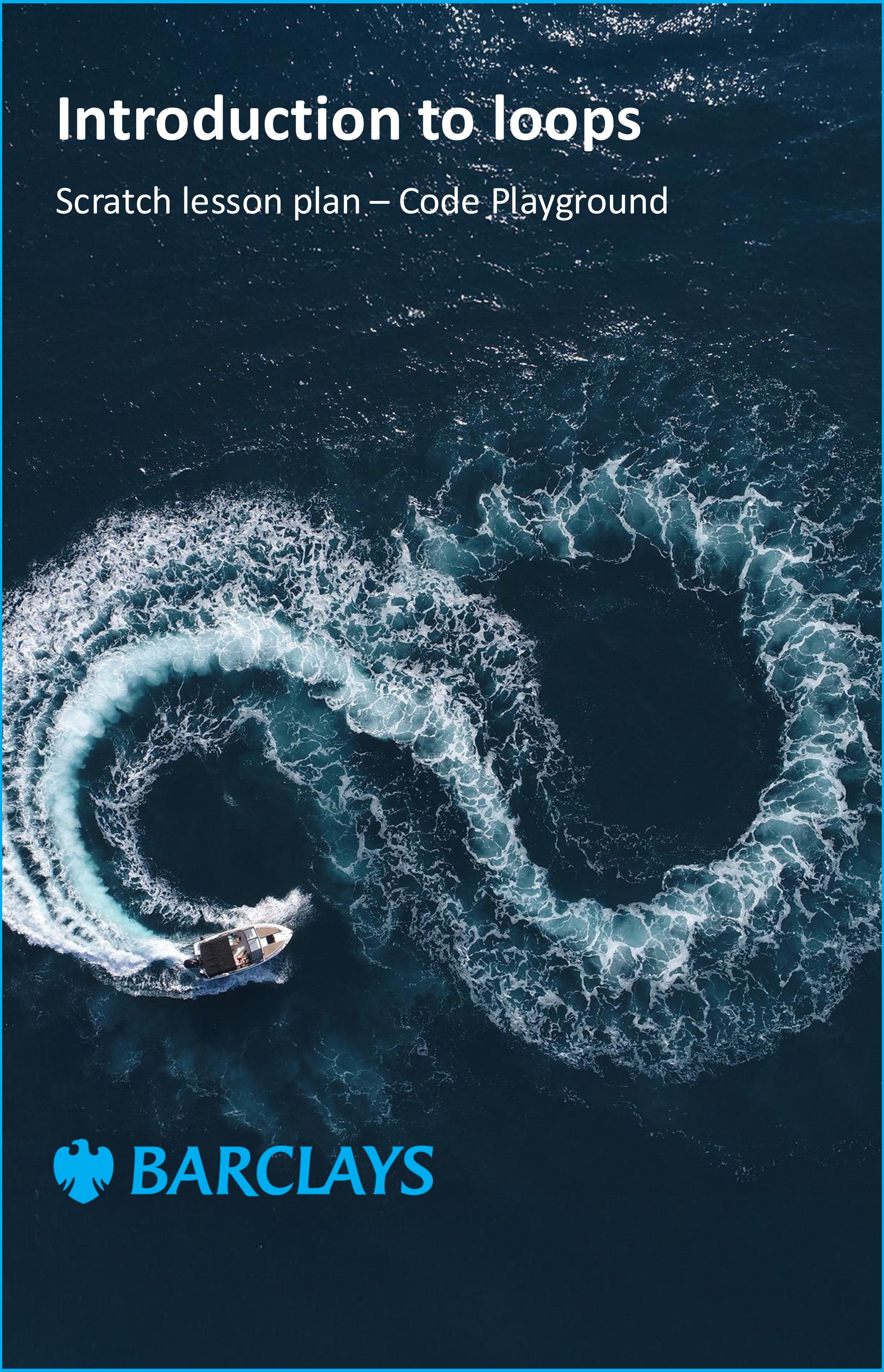


Introduction to loops

Scratch lesson plan – Code Playground

BARCLAYS

Lesson overview

Introduce students to loops in Scratch, helping them understand how loop blocks like '**forever**', '**repeat until**', and '**repeat x times**' work. Students will learn to simplify code by repeating actions and creating patterns with loops.

Time	Key learning outcomes	Resources
30 mins	• Understand the purpose of **forever**, **repeat until**, and **repeat [x] times** in Scratch • Use loops to create shapes and movement patterns • Change loop parameters to affect sprite behaviour • Recognise how loops simplify code by reducing repetition.	• Access to scratch.mit.edu • Projector or smartboard for demonstrating code examples

Content

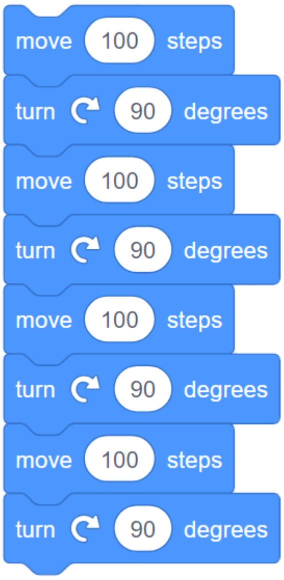
Introduction - What are loops?

Introduce the concept of loops as instructions that repeat actions in coding. Imagine you're walking in a circle. Each time you finish the circle, you come back to the start and begin again. This is what loops do in coding — they keep repeating actions until we tell them to stop.

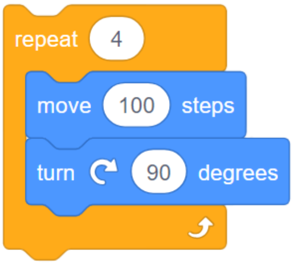
Types of loops in scratch

- **Forever**: Describe this loop as something that continues without stopping, like blinking or breathing. Explain that in Scratch, **forever** repeats an action until the program is stopped
- **Repeat until**: Compare this to a timer that keeps counting until it reaches zero. Explain that **repeat until** does the same, repeating an action until something specific happens
- **Repeat [x] times**: Compare this to skipping a rope, where each jump could be counted. Explain that **repeat [x] times lets us choose how many times we want an action to repeat.**

Show them an example that moves a sprite in a square without a loop (using **move 100 steps** and **turn ↻ 90 degrees** four times), then show how the **repeat [4]** block simplifies the code by reducing repetitive instructions.



```
move 100 steps
turn ↻ 90 degrees
move 100 steps
turn ↻ 90 degrees
move 100 steps
turn ↻ 90 degrees
move 100 steps
turn ↻ 90 degrees
```



```
repeat 4
    move 100 steps
    turn ↻ 90 degrees
```

Ask your class

- Why do you think loops are useful?
- How does it help to repeat actions with fewer steps?

Without a loop, we have to repeat the same instructions again and again. With a loop, we tell the computer to do the same thing multiple times in a row without extra work.

Activity 1: Drawing a square with loops (15 minutes)

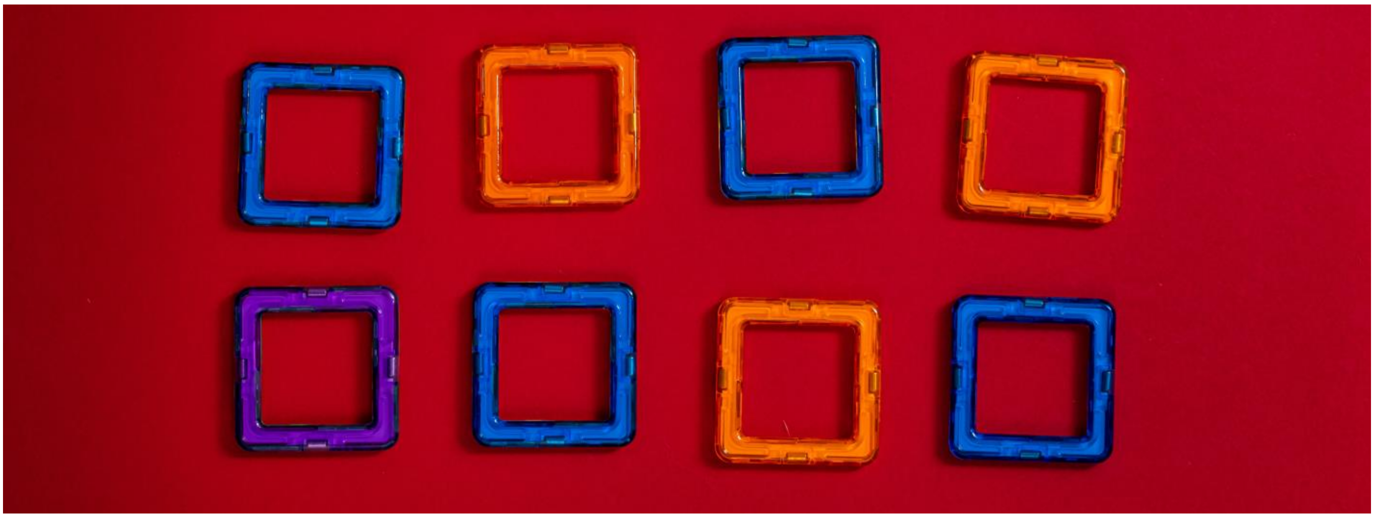
Guide students in using the **repeat [4]** loop to draw a square.

Imagine you're walking in a square on the playground. Each time you finish one side, you turn, walk the next side, and so on. By repeating this four times, you complete a square

Instructions

1. Ask students to tell you what they'd do to walk around this square
2. Explain that in Scratch, we can make the sprite do the same thing by combining **move** and **turn** blocks with a **repeat [4]** loop to create a full square
3. Encourage them to add the Pen extension and use **pen down** so their sprite draws as it moves.

Just like you'd turn at each corner of a square, the sprite uses **turn ↻ 90 degrees** in Scratch to create right angles.



Ask your class

- What would happen if we changed repeat [4] to repeat [5]?
- What would happen if we changed **turn ↻ 90 degrees** to **turn 60 degrees?** What shape would it create?

Activity two – Make the sprite dance

Students will learn how to use the **forever** loop to make a sprite dance by changing costumes repeatedly.

If you're doing a dance where you repeat the same steps again and again. A **forever** loop in Scratch lets our sprite keep changing its costume, which looks like it's dancing.

Instructions

1. Explain that some sprites have multiple costumes that they can switch between, like dance poses
2. Show students how to set up a **forever** loop with **next costume** to keep changing costumes
3. Explain that adding **wait 0.5 seconds** makes the costume change slower, creating a dance rhythm.

Think of a flashing light that goes on and off at regular intervals. Changing the costume in a loop is like making the sprite 'flash' through different dance moves.

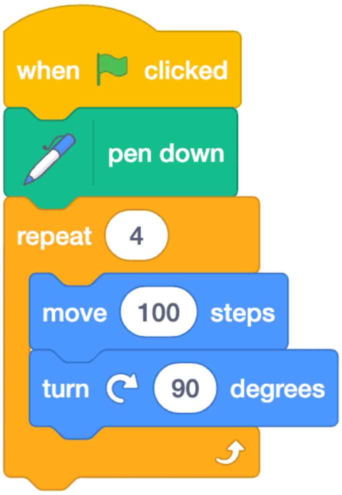
Ask your class

- How does changing the wait time affect how fast or slow the dance looks?
- What would happen if we added **move 10 steps** inside the **forever** loop? Would the sprite dance across the screen?

Code snippets

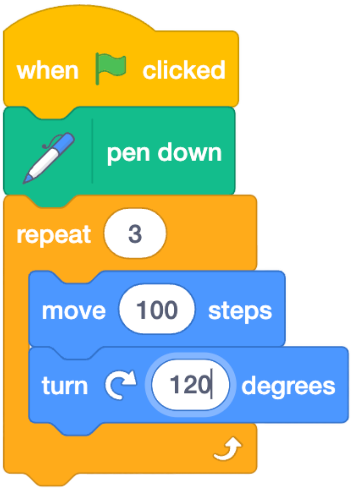
Shapes

Square



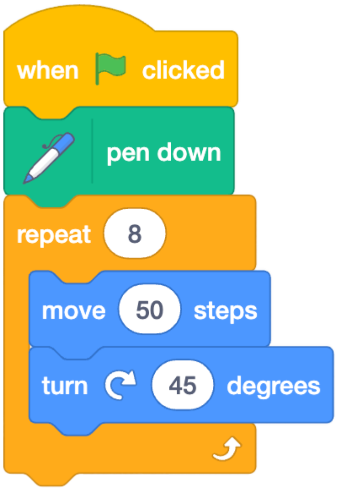
```
when ⚑ clicked
pen down
repeat (4)
    move (100) steps
    turn ↻ (90) degrees
```

Triangle



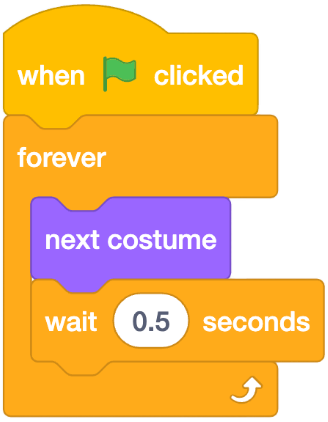
```
when ⚑ clicked
pen down
repeat (3)
    move (100) steps
    turn ↻ (120) degrees
```

Octagon

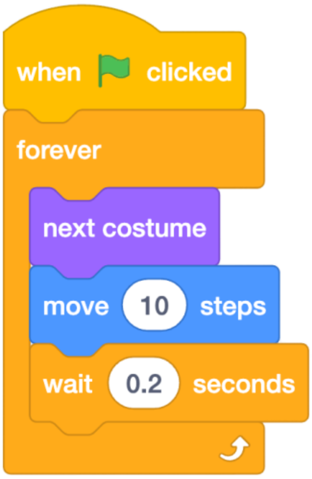


```
when ⚑ clicked
pen down
repeat (8)
    move (50) steps
    turn ↻ (45) degrees
```

Dance

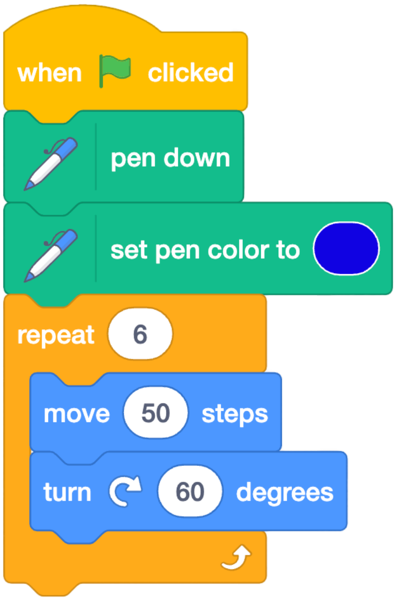


```
when ⚑ clicked
forever
    next costume
    wait (0.5) seconds
```



```
when ⚑ clicked
forever
    next costume
    move (10) steps
    wait (0.2) seconds
```

Extension



```
when ⚑ clicked
pen down
set pen color to (●)
repeat (6)
    move (50) steps
    turn ↻ (60) degrees
```

Summary

The following information is an example of what a child at an expected level would be able to demonstrate when completing these activities with additional examples to demonstrate how this would vary for a child with emerging or exceeding achievements.

Assessment guidance

Differentiation – Lower ability/ASN

- Use visual aids or printed screenshots of each Scratch block to help students recognise and locate the correct blocks
- Simplify each activity into smaller steps, focusing on just one or two loops, such as **repeat [x] times** to make it easier to follow
- Before coding, have students physically act out walking in a square to understand the pattern. Ask them to take four steps, with a turn after each one, reinforcing the concept of repetition
- Pair lower ability students with a peer who can offer additional support, guiding them through selecting blocks and connecting them.

Differentiation – Higher ability/extension

- Encourage higher ability students to adjust the **repeat** count and **turn** angles to create different shapes, like triangles (repeat 3, turn 120 degrees) or hexagons (repeat 6, turn 60 degrees)
- Introduce **repeat until** in combination with other blocks, such as using it to make a simple sprite move until it reaches a certain position or touches an edge
- Challenge students to add **change pen colour** inside loops, creating colourful, multi-layered designs
- Suggest that students use loops to play sounds in rhythm or set up a **forever** loop to respond to keyboard inputs, like making the sprite dance to the beat by pressing keys.

Plenary

- Gather the class to discuss what they've created with loops. Ask students to describe how loops made coding simpler and more efficient
- Encourage students to share the shapes, dances, or patterns they've coded and how adjusting the loop values affected the results.

Assessment questions

- What is a loop in coding and why do we use it?
- Can you name the three main types of loops we used today? – Forever, repeat [x] times and repeat until
- Can you give an example of a loop in real life, like something you repeat regularly?