

Teaching Primary Programming with Scratch

Pupil Book – Year 3

PHIL BAGGE

A research informed scheme of work by Phil Bagge HIAS Computing Inspector/Advisor
Part of the HIAS Teaching Primary Programming from Scratch Series

Published in 2023 by University of Buckingham Press,
an imprint of Legend Times Group
51 Gower Street
London WC1E 6HJ
info@unibuckinghampress.com
www.unibuckinghampress.com

Copyright © Phil Bagge 2023

Published by arrangement with Hampshire Inspection and Advisory Service (part of Hampshire County Council)

All rights reserved. No reproduction, copy or transmission of this publication may be made without written permission.

Except for the quotation of short passages for the purposes of research or private study, or criticism and review, no part of this publication may be reproduced, stored in a retrieval system, copied or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, now known or hereafter invented, save with written permission or in accordance with the provisions of the Copyright, Design and Patents Act 1988, or under terms of any licence permitting limited copying issued by the publisher.

This book is sold subject to the condition that it shall not, by way of trade or otherwise, be lent, resold, hired out, or otherwise circulated without the publisher's prior consent in any form of binding or cover other than that in which it is published and without a similar condition including this condition being imposed on the subsequent purchaser.

Any person who does any unauthorised act in relation to this publication may be liable to criminal prosecution and civil claims for damages.

ISBN 978-1-91505-4-227

CONTENTS

Introduction & Progression

Introduction	6
--------------	---

Introducing New Concepts

1 Simple Sequence	11
2 Fast & Slow Blocks	14

Modules of Work

3 Animal Challenges*	19
4 Conversation	35
5 Fish Fun*	46
6 Animation	60
7 Ladybug Munch*	68

A good plan might involve doing two or three of these modules in Year 3.
A Game followed by a nongame with Ladybug Munch to finish if pupils reading is good.

*These are simple games

INTRODUCTION

What Does This Book Do?

This book is a complete scheme of work for teaching primary programming using Scratch in Year 3 for 7–8 year olds.

What Is Included?

It includes permission to photocopy the pupil's worksheets answersheets and other resources that include this block for your class or school.

It includes links to example code, project templates and slides to improve how you introduce new programming concepts.

Part of a Series

It is part of a five-book series. Three other books include projects for other year groups.

Teaching Primary Programming with Scratch, Year 4

Teaching Primary Programming with Scratch, Year 5

Teaching Primary Programming with Scratch, Year 6

Teaching Primary programming with Scratch – Research-Informed Approaches

The teacher book listed at the end explores methodology and pedagogy in detail helping you to understand why an approach is useful.

Progression

There is a clear, research-informed progression through the series, and the graphic on the next page on a grey background shows which programming concepts are introduced in this book.

Pedagogy in a Few Paragraphs

Introduction to Programming Concepts Away From Code

Pupils are taught key programming concepts away from programming to lower cognitive load and make it easier to transfer these ideas from one programming language to another.

I Do You Do

While pupils are finding their way around the Scratch environment, we employ this basic methodology as it allows more solo hands on access.

PRIMM

Pupils are encouraged to read and understand code before they create their own code. We use the PRIMM method in this book in the last module.

Predict

Run

Investigate

Modify (change)

Make

Parsons

Some modules include a Parsons exercise to build code from teacher-provided pre-selected code.

Creative

Each project provides time and stimulus to be creative in code within the zone of proximal development provided by the taught concepts and explored projects. In other words, it has reasonable projects that can be created independently or with minimum teacher support.

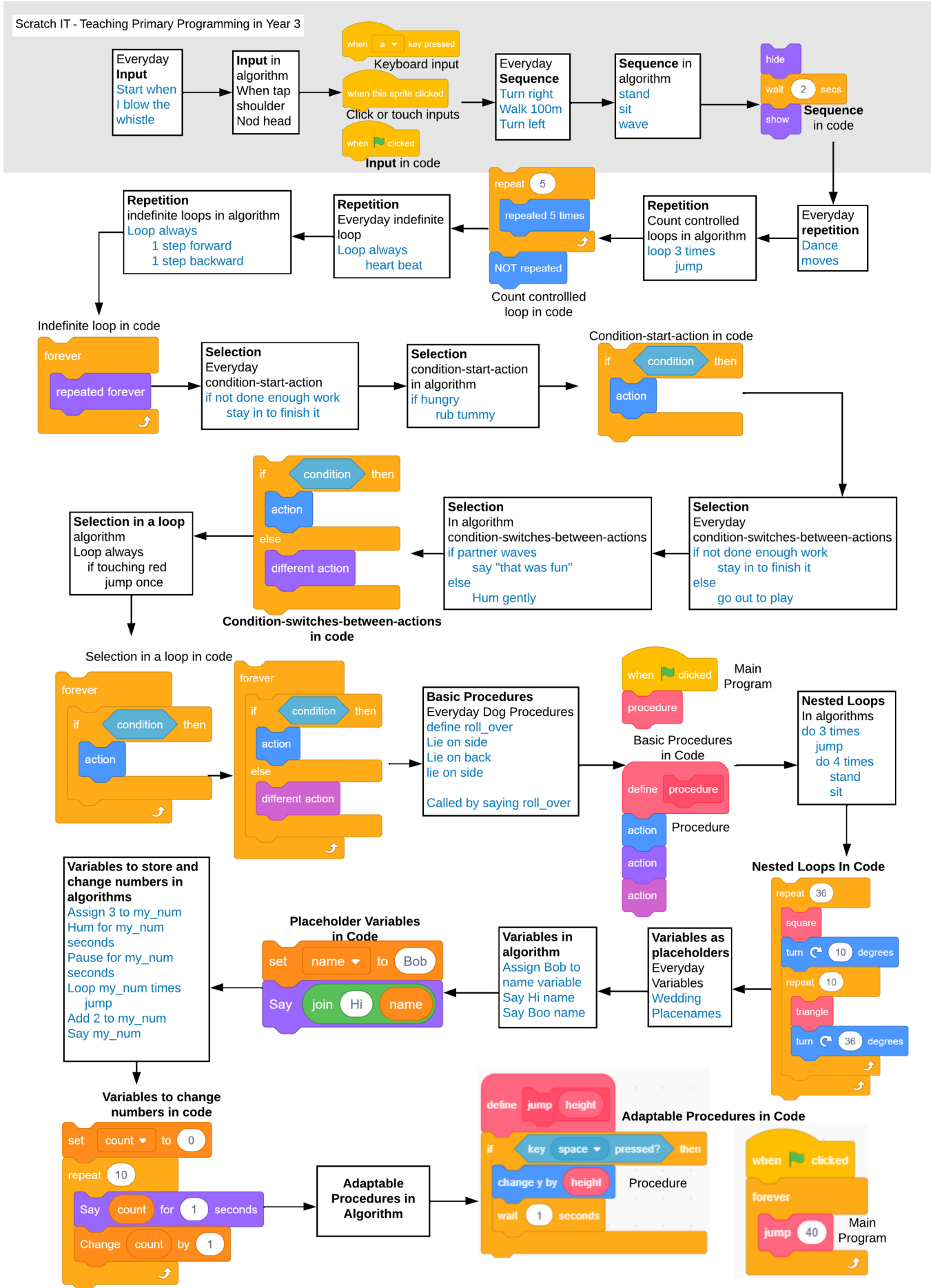
Knowledge

Key knowledge is introduced in the concept introductions and reinforced in each of the activities.

Hints & Tips

The last module also includes a copy of the pupil resource annotated with extra information to further teachers programming knowledge, hints and formative assessment opportunities in case pupils are stuck and tips to adapt or support whole class teaching.

Many of these extra hints and tips will not be needed, but the more informed the teacher is, the better quality learning opportunity pupils will have.



Committed to Improvements

HIAS, Hampshire’s Inspection & Advisory Service, is committed to developing and improving these resources. We recognize that primary programming is still its infancy in comparison with other

subjects and that new research and primary practice will refine and improve teaching and learning in this area. All royalties earned from this series will be used to write more computing books and revise these resources as needed.

Scratch IT 2

Come Back Doggy!
INVESTIGATE

Start Scratch and load the Come Back Doggy! program

Work with a partner

Count controlled loop

Play Come Back Doggy! a few times. The green flag starting block will start the program.

Mark your reading code and predicting what it will do questions from the last sheet

Investigate the code
 Run the programs lots of times to help you answer the questions but don't change the code

Look at the code inside Maria
 Maria sprite questions

- Which block starts the code?
- What block makes Maria go back to the start? (Initialization) *HINT* go to
- Which block rubs out any old lines before Maria searches for her dog? (Initialization)
- In the first repeat loop (count-controlled loop) how many times will move 1 step be run?
- Which loop draws the shortest line?
- Which block changes Marias direction?

Look at the code inside the dog
 Dog sprite questions

- Which line of code makes the dog wait until Maria arrives?
- Which blocks get repeated 21 times?
- What direction (up, down, right or left) does point in direction 180 make the dog go?

Now mark the investigate questions using the answer sheet

72

page can be photocopied

Photocopiable resource for pupils

Come Back Doggy!

Come Back Doggy!
Supporting INVESTIGATE

Play Come Back Doggy! a few times. The green flag starting block will start the program.
 Mark your reading code and predicting what it will do questions from the last sheet

Investigate the code
 Run the programs lots of times to help you answer the questions but don't change the code.

Look at the code inside Maria
 Maria sprite questions

- Which block starts the code?
 Green flag (1 mark)
- What block makes Maria go back to the start? (Initialization)
 HINT go to
 go to x and y (1 mark)
- Which block rubs out any old lines before Maria searches for her dog? (Initialization)
 Erase all (1 mark)
- In the first repeat loop (count-controlled loop), how many times will move 1 step be run?
 250 (1 mark)
- Which loop draws the shortest line?
 Repeat 160 or the second repeat loop (1 mark)
- Which block changes Marias direction?
 Point in direction (1 mark)

Look at the code inside the Dog
 Dog Sprite Questions

- Which line of code makes the dog wait until Maria arrives?
 Wait until touching Maria (1 mark)
- Which blocks get repeated 21 times?
 Next costume (1 mark) wait 0.4 seconds (1 mark)
- What direction (up, down, right or left) does point in direction 180 make the dog go?
 Down (1 mark)

Now mark the investigate questions using the answer sheet

73

Whole class advice
 Work in pairs, one device between the pair. Take it in turns every question to swap who runs code. You must work at the same pace as your partner and not move on to the next question until you have both written your answer down. If you disagree write a different answer. You must mark your work before moving on to the next section.

Notes on the activity
 Investigating the code encourages pupils to think deeply about how it works. Check that every pupil is filling in and marking the questions individually but at the pace of the slowest in the pair. Sometimes a pair decides not to mark to speed up their efforts. Marking gives valuable information so I recommend sending them back to mark their work. A class instruction to come and talk to you if they have over half of the questions wrong or they do not understand the answer after they have marked it helps to check progress is being made correctly. There is real value in collecting these scores to build up a summative picture of pupil progress.

Q2 Code initialization—The idea that we need to write code to make sure the program resets itself before running again is a hard concept so it is important to drip feed this in every project. Why not add it to your spellings or word wall.

Q2 Pupils don't need to understand x and y at this moment it is enough to know that these numbers make the code go to a place on the screen. Dragging a sprite to the place you want it to start from and then dragging an x and y block will give it the correct coordinate reference points.

Q4 You can sometimes help by simplifying
 Say you are inside a loop 5 times move 1 step algorithm. How many steps will you take?

Q5 Repeat 10 with move 1 inside would move 10 steps. Repeat 50 with move 1 would move 50 steps.

Q6 Key word direction.

Q7 Wait until touching is a condition which we will explore in much more depth next year. In Year 3 and 4, we are sticking to wait until <insert condition> blocks wait until a key is pressed or wait until a colour is touched are other common ones. These are simple enough to understand in a concrete way.

Q9 Click on the direction block to show a direction dial.

Send advice
 Support pairs of pupils who are poor readers by reading questions, reading code samples and covering up questions until they get to them.

Teacher Hints & Tips on the same resource

CHAPTER 1

Simple Sequence

Introducing Simple Sequence

These slides can be downloaded from <https://computing.hias.hants.gov.uk/course/view.php?id=51>.

Part One: Simple Sequence

They are designed to be delivered to the whole class before pupils move on to using simple sequence in any module of work in this book. If pupils have used the slides before a previous module of work omit them.

Part Two: Fast & Slow

Is designed to be delivered before the conversation or animation modules. If pupils have not viewed sequence or fast and slow, then deliver one in the first session and the second in a further lesson.

They can also be delivered to a small group of pupils if they are working independently through resources in pairs.

Format

Slides are provided in PDF and PowerPoint Formats and teachers who purchased the book are authorized to adapt the resources within their school or on closed learning platforms such as Seesaw, Google Classroom or Teams as long as they are not shared outside the school community.

Hints

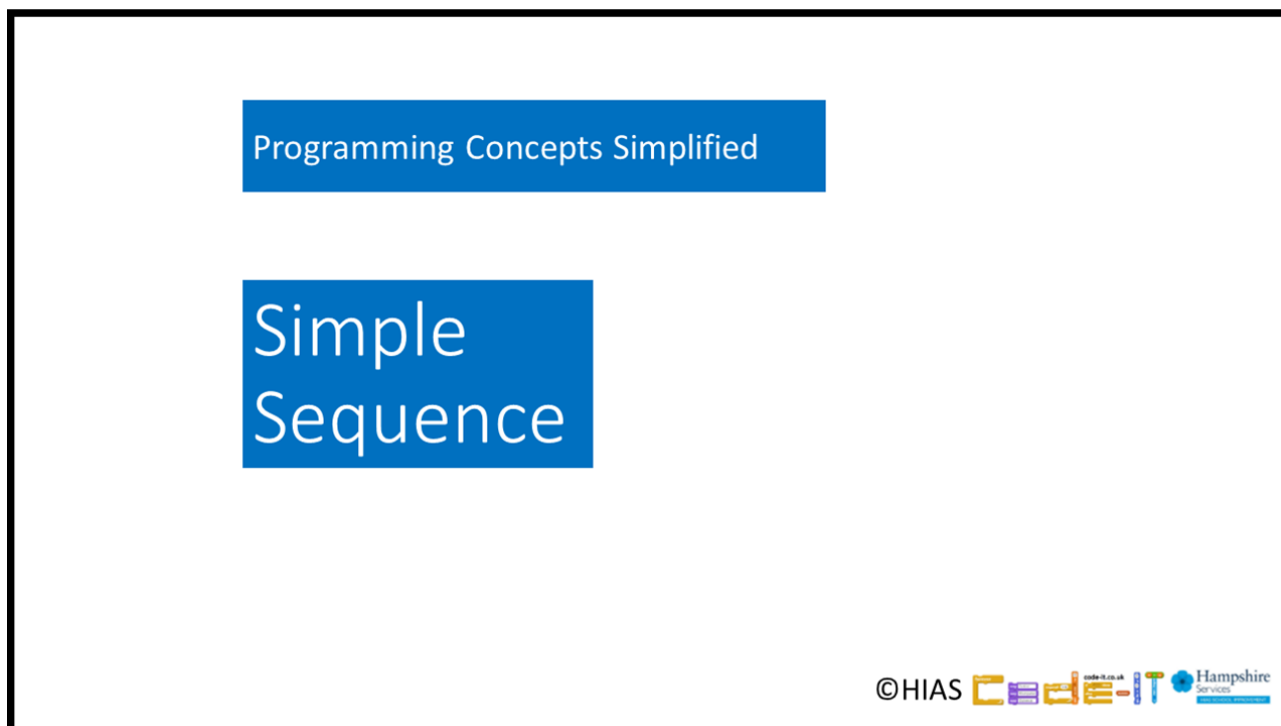
Extra hints and tips on usage are provided alongside each slide on the following pages.

Resources

Pupils will need whiteboards and pens or paper and pencils.

Summary Sheet

These is a summary sheet that follows this slide commentary that pupils can use to write their algorithms on and be reminded about key knowledge.



Simple Sequence

Stand
↓
Wave
↓
Bow
↓
Sit

One thing at a time in an order

A simple definition of sequence. NOTE sequences can include other concepts such as repetition or selection, but for this book these more complex ideas are omitted, hence simple sequence.

Simple Sequence

Stand
↓
Wave
↓
Bow
↓
Sit

Arrows show the order the algorithm takes

Point out that sequences start at the top and work their way down. They could also go from left to right. Pupils should know that an algorithm is a set of instructions to do something that can be understood by a human, and you may want to remind them of this.

Simple Sequence

Stand
Wave
Bow
Sit

Act it out

Ask pupils to act it out. You could point to the actions one at a time to help them.

Simple Sequence

Stand
Wave
Bow
Sit

We all waved differently

Point out that some people waved in different ways. This is fine as we have not given much detail.

Simple Sequence

Stand
Wave once slowly
Bow
Sit


Act it out

Ask pupils to act out this adapted algorithm.

Simple Sequence

Stand
Wave once slowly
Bow
Sit

What was different?



Ask them to either write down what is different or tell their neighbours.

Simple Sequence

Stand
Wave once slowly
Bow
Sit

What was different?

More detail telling us how to wave

Accept any answers that point out that wave has more detail telling us how to wave

Simple Sequence

Stand
Wave once slowly
Bow
Sit

Algorithms are easier to follow if they are precise with more detail

Read the slide and explain what precise is, if any need this.

Simple Sequence

Stand
Wave *once slowly*
Bow
Sit

Now write your own and act it out with a partner

Can you make it precise?

Simple Sequence

Stand
Wave *once slowly*
Bow
Sit

This is the flow of control, the order the algorithm takes

Instruct pupils to write their own algorithm and then give them to their neighbour to act out. They can have more actions. If any pupils struggled to act out the last algorithm check their work first. Offer to scribe for poor writers.

As you read through the algorithm, again point to the dots on the line. Explain that this is the flow of control, the order the instructions take in this algorithm.

Simple Sequence

Stand
Wave *once slowly*
Bow
Sit

Dots show actions

Simple Sequence

Stand
Wave *once slowly*
Bow
Sit

Draw a flow of control next to your algorithm

Dots show actions.

Ask pupils to draw the flow of control line and dots on top of their sequence algorithms.

Simple Sequence

when clicked

show

say Hello! for 2 seconds

play sound welcome until done

say Bye for 2 seconds

hide

Simple sequence in Scratch code

Simple Sequence

when clicked

show

say Hello! for 2 seconds

play sound welcome until done

say Bye for 2 seconds

hide

Simple sequence in Scratch code with flow of control

Point to the code blocks one at a time and read them. You could also explain what each command will do. Point out that they will be run in order from top to bottom.

Programming has the same flow of control as the algorithm. From top to bottom.