

CAMBRIDGE

Coding  
Club

# Python: Programming



Supplement 1

Chris Roffey



**Python:**

**Programming**

**Art**

**Coding  
Club**

**Supplement 1**

**Chris Roffey**

# CAMBRIDGE UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning and research at the highest international levels of excellence.

[www.cambridge.org](http://www.cambridge.org)

Information on this title: [www.cambridge.org/9781107631090](http://www.cambridge.org/9781107631090)

© Cambridge University Press 2014

This publication is in copyright. Subject to statutory exception and to the provisions of relevant collective licensing agreements, no reproduction of any part may take place without the written permission of Cambridge University Press.

Printed in Poland by Opolgraf

*A catalogue record for this publication is available from the British Library*

ISBN 978-1-107-63109-0 Paperback

Cambridge University Press has no responsibility for the persistence or accuracy of URLs for external or third-party internet websites referred to in this publication, and does not guarantee that any content on such websites is, or will remain, accurate or appropriate. Information regarding prices, travel timetables, and other factual information given in this work is correct at the time of first printing but Cambridge University Press does not guarantee the accuracy of such information thereafter.

---

## NOTICE TO TEACHERS IN THE UK

It is illegal to reproduce any part of this work in material form (including photocopying and electronic storage) except under the following circumstances:

- (i) where you are abiding by a licence granted to your school or institution by the Copyright Licensing Agency;
- (ii) where no such licence exists, or where you wish to exceed the terms of a licence, and you have gained the written permission of Cambridge University Press;
- (iii) where you are allowed to reproduce without permission under the provisions of Chapter 3 of the Copyright, Designs and Patents Act 1988, which covers, for example, the reproduction of short passages within certain types of educational anthology and reproduction for the purposes of setting examination questions.

# Contents

Introduction	4
<b>Section One – Python's turtle module</b>	<b>7</b>
Chapter 1: Introducing turtle	8
Chapter 2: Shapes, colour and repetition	18
Chapter 3: Getting creative	29
<b>Section Two – Building an art app</b>	<b>42</b>
Chapter 4: From small beginnings	43
Chapter 5: Red lines are not enough	59
Chapter 6: Stamping and painting	73
Taking things further	95
Appendix	96
Glossary and index	101
The Quick Quiz answers	106
Acknowledgements	108

# Introduction

## Who is this book for?

This book is the first Level 1 supplementary project book in the *Coding Club* series of books. You should have read *Coding Club: Python Basics* before reading this or be familiar with Python 3 and know about, at the very least, variables, while loops and functions. It is aimed at 11–14 year olds with only a little previous programming experience. In *Programming Art* there is very little new theory introduced. The intention is to allow young coders to consolidate what they have learned in *Python Basics* in new situations while building some fun and challenging projects along the way. Although there is not much new theory, there are plenty of new techniques and a new module to add to your repertoire.



## Why should you choose this book?

Learning to program is not just about learning the syntax for while loops, functions, variables, etc. It is about knowing when to use them in new situations. This can take time. If you have just finished *Python Basics* but struggled with several of the challenges, it is likely that you need to practice what you have learned a bit more before moving straight to a Level 2 book. That is not a problem – it would be a shame to skip this book as it has some great projects in it.

## What you need?

Any computer can run Python 3. If your computer does not already have Python 3 installed there is a section on the companion website (<http://www.codingclub.co.uk>) that guides you through the download and installation. This takes about five minutes! That is all you need to get started.

Start files and (should you require them) finished files for the final three chapters are available from the companion website in one easily downloadable zip file, so you do not have to do too much typing or get lost in the *Art App* project. On the website, there are also answers to the puzzles and challenges.

## How to use this book

You should read this book carefully and build all the main projects in order. At the end of each chapter there are further ideas, and challenges that you can think of as ‘mini quests’. Some readers will want to work through them all so that they understand everything all the time. Some of you will probably prefer to rush through and get to the end. Which approach is best? The one you are most comfortable with is the best approach for you. If you are being

guided by a teacher, you should trust their judgement so that they can give you as much help as possible.

The explanations are shorter than in *Python Basics* as this book is about consolidation rather than learning lots of new things. There are four ways in which this book tries to help you to learn:

- 1 Typing in the code – this is important as it gets you to work through the code a line at a time (like computers do) and will help you to remember the details in the future.
- 2 Finding and fixing errors – error messages in Python give you some clues as to what has gone wrong. Solving these problems yourself will help you to be a better programmer. In the end though, this should not become boring so if you get stuck the code can be downloaded from the companion website ([www.codingclub.co.uk](http://www.codingclub.co.uk)).
- 3 Experimenting – feel free to experiment with the code we write. See what else you can make it do. If you try all the challenges, puzzles and ideas, and generally experiment with the code, this will help you learn how to write code like a professional.
- 4 Finally, this book will not only provide the code to build some pretty cool, short projects – it will also teach you how the programs were designed. You can then use the same methods to design your own applications.

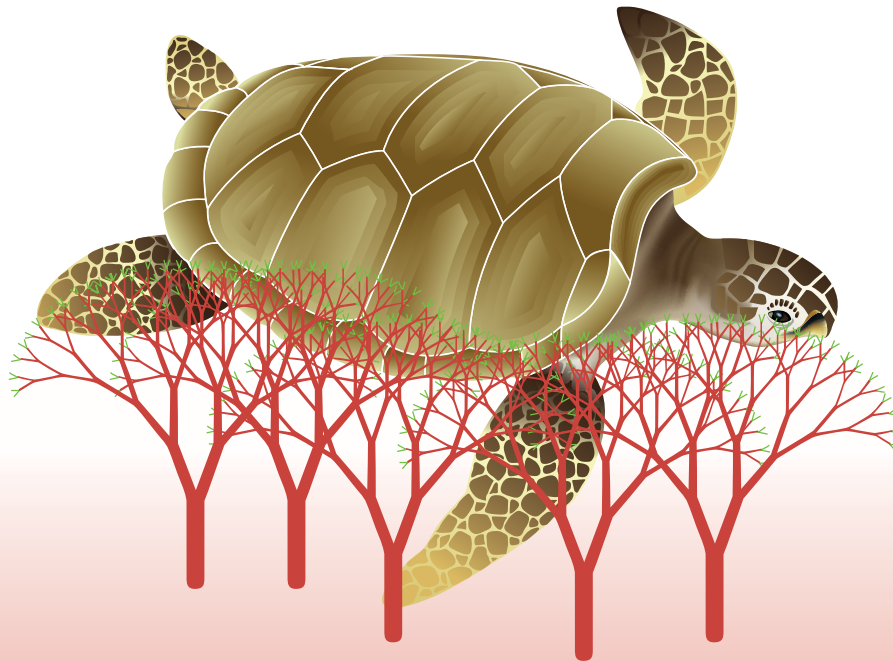
## A word of warning

You may be tempted to simply get the code off the website instead of typing it in yourself. If you do this you will probably find that you cannot remember how to write code so easily later. In this book you will only be asked to type small chunks of code at a time – remember that this will help you understand every detail of each of your programs.



# Section One

## Python's turtle module



# Chapter 1

## Introducing turtle

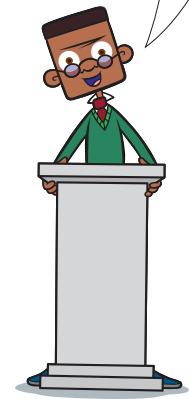
In *Coding Club: Python Basics* you learned the fundamentals of programming using Python 3. In this book, you will use that hard-won knowledge to have some fun making some little applications while re-enforcing your knowledge and learning a few more tricks.

Python 3 comes with some great, ready-built **modules** some of which we have already used such as `tkinter` and `random`. Another module we can use is `turtle`. This is an implementation of the turtle graphics part of a complete programming language called Logo which was created for educational use; schools often used it to drive a toy turtle around classrooms. The **commands** available in Python's `turtle` module are very easy to learn. The fantastic thing about this Python module is that there is nothing new to install and we can combine the turtle commands with the Python language we have already learned.

In this chapter, you will learn how to:

- import the turtle module
- make your turtle move around in all directions
- change what the turtle looks like.

The original Logo programming language was developed by Daniel G. Bobrow, Wally Feurzeig, Seymour Papert and Cynthia Solomon in 1967.



# Hello World

We are introducing a new module, so why not start with a Hello World program to test and see how turtle graphics works? Well, one reason would be that we are going to use our turtle to draw the letters rather than use text! However, here we go anyway.

Open **IDLE** from your Python 3 install and then open a new window by selecting *File* then *New Window*. Type out the code from Code Box 1.1 and save the file as `hello_world.py` into your *Python Code* folder which you created when reading *Python Basics*. Run the program and see what happens. It really is, almost, a 'hello world' program!

## Delving Deeper

Turtle is not a new language; it is a Python module. What this module does, though, is give access to many of the commands of the turtle graphics part of the Logo language. This does not mean we cannot write a "Hello World" program. It is quite common to do so when learning new aspects of a language. For example, when learning about the **tkinter** module many programmers would write a short program that opens a window and displays the text: "Hello World".

### Code Box 1.1

```
# hello_world.py introduces the turtle module
from turtle import *

# change line width
pensize(5)
```

(continues on the next page)

Hi, I'm Sam. Do you remember from *Python Basics* that when we are working in **script mode** we write code in a new window and save it?

