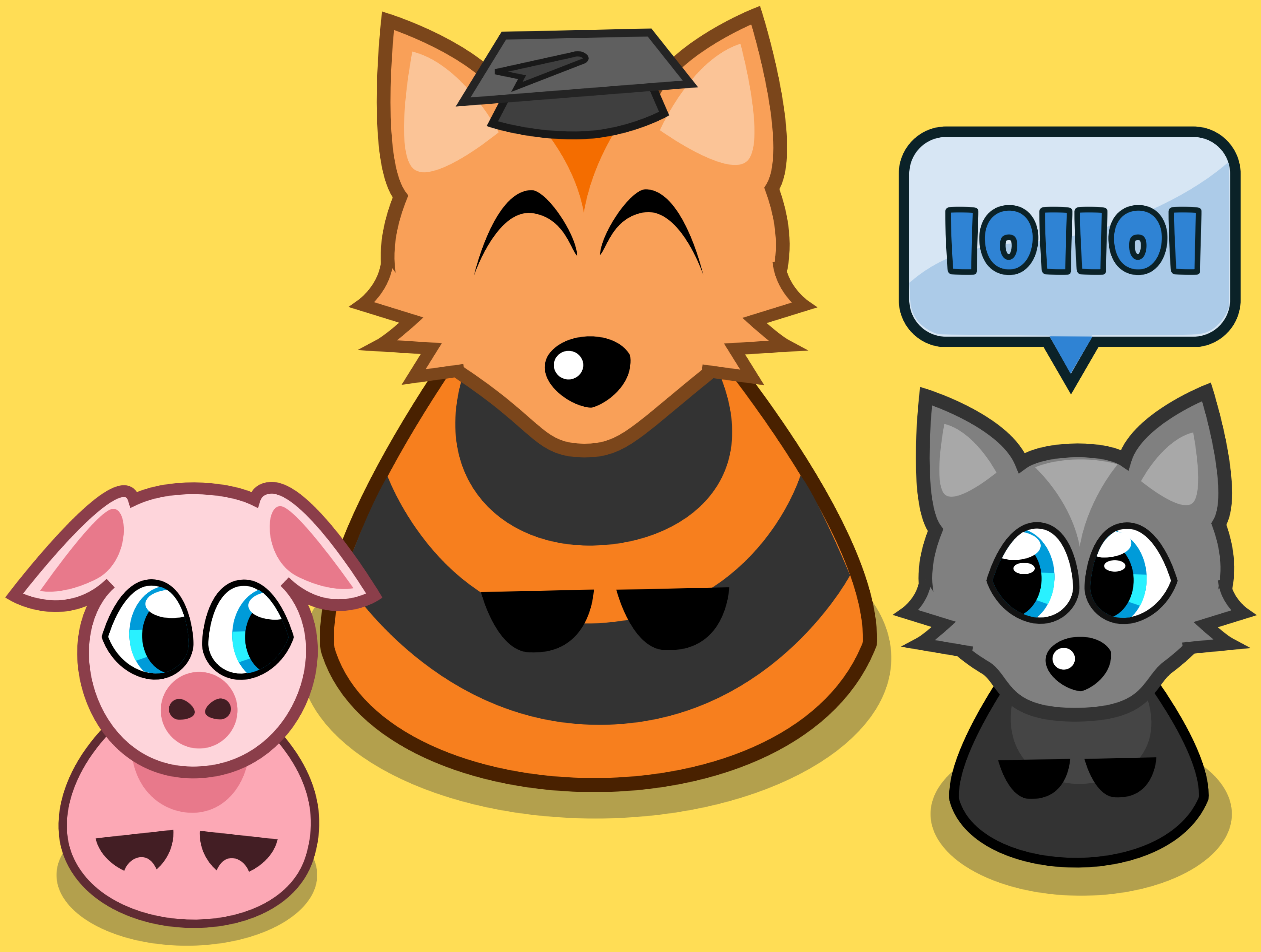
codekingdoms

Code Kingdoms
Teacher Guide



for kids, with kids, by kids.

Resources overview

We have produced a number of resources designed to help people use Code Kingdoms. There are introductory guides to all parts of the product and classroom materials to help teach lessons around Code Kingdoms.

codekingdoms
Code Kingdoms
Learning: What, where, when and how
for kids, with kids, by kids.

Code Kingdoms Learning: What, where, when and how

A summary of the Code Kingdoms approach to learning.

Teacher Guide

An overview for teachers. Describes the Code Kingdoms learning ethos and details the different parts of the product.

Dashboard Guide

A beginner's guide to using our group management tool. Describes everything from registering for an account to assessing the progress of your kids.

codekingdoms

Sandbox guide

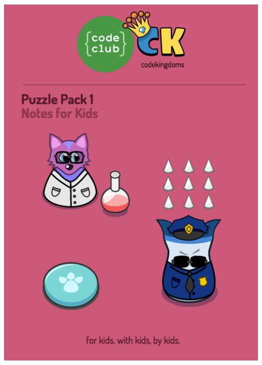
A guide to using our unstructured creation environment. Learn everything from using the menus to making great puzzles.

Unit 1: Introducing Code Kingdoms

An introductory unit of six 'off-the-shelf' lesson plans. Targeted at KS2 kids.

Unit 2: Learning a language

Six 'off-the-shelf' lesson plans designed to teach kids the basic of JavaScript



Puzzle Packs

A guide to building specific puzzles in Creative mode. Step-by-step instructions from start to finish. Four puzzles per pack.

What is Code Kingdoms?

Code Kingdoms teaches programming and computational thinking in a way that's fun for kids. They build and protect their lands with puzzles coded in real JavaScript.

Why Code Kingdoms?

Code Kingdoms wants to inspire kids to learn one of the essential creative skills of the 21st Century - coding.

We want kids to have the freedom to have fun and be creative with code, so we built a game that enables just that. In Code Kingdoms, kids build and protect their own worlds and share them with friends.

All kids in UK schools are now expected to learn to code. We believe games are a highly effective platform for learning to code because kids relate to them. Of the thousands of kids we worked with to develop Code Kingdoms, a great number said they wanted to use code to build their own games.

Computing is about more than coding

Code Kingdoms allows kids aged 6 to 13 to learn computational thinking alongside a real coding language, whilst promoting soft skills like problem-solving, teamwork and time management. We encourage kids to experiment with code knowing that it won't always work - breaking things is OK! Getting stuck, debugging and further problem solving are all part of how programming works in the real world.

As their skills in coding and computational thinking develop, so does their ability to create challenging and exciting puzzles. A lot of the satisfaction for kids and teachers comes from discovering how they can use code to create really fun games.

CK School

The school environment is browser-based (Google Chrome is preferred) and allows kids to solve puzzles using computational thinking and build their own coded puzzles with integrated classroom management tools. Each activity comes with its own summary card that describes the puzzle mechanics and the learning outcomes associated with solving or building the puzzle.

As kids progress through the activities, they will develop skills in computational thinking, coding and game design. Good coding and puzzle design are closely correlated with robust defences to prevent Glitches attacking!

Dashboard

The dashboard is designed to make using Code Kingdoms with a group of kids easy. As group leader, you can manage user accounts, set lesson activities and assess kids' progress against national standards. We have a dashboard-specific guide which can be found at codekingdoms.com/teachers. If you'd like to access the dashboard directly it can be found at dashboard.codekingdoms.com.

Sequencer

The Sequencer is the programming interface where kids write lines of real JavaScript code to build puzzles and control their lands. Kids begin by dragging chunks of code, allowing them to focus on what they can achieve with JavaScript, before a seamless transition to text-based programming which allows them to grasp syntax and formatting code. Our unique slider allows kids to experience both inputs in a consistent environment and fully scaffolds the progression to text-based programming.

In the Classroom - The 3Ws

To assist students in thinking about structuring lines of code, we recommend using the 3Ws **(When? Who? What?)**

When you are trying to write a line of code to carry out a command break down what you are trying to do using the 3Ws.

When?

What event will trigger the action occurring? E.g. when a button is pressed or onPress

Who?

Who should the command control e.g.CatapultA. Hint: this could be an object or a character.

What?

What action do you want to happen? E.g.Direction=NORTH

Further Support

The 'Contact Us' page is a staple of any company website, inviting stakeholders to be in touch with queries and for assistance. As a growing company we take supporting our users especially seriously. We want kids, teachers and parents to have the best experience using our product and the feedback they share is invaluable.

Our team of educators and developers regularly schedule calls to help with set up, discuss learning approaches and resolve technical issues – so please use the channels below if you'd like to talk to us.

Email: team@codekingdoms.com
Twitter: @CodeKingdoms
Facebook: facebook.com/CodeKingdom

codekingdoms

Glossary

1. ALGORITHM - a precise step-by-step guide to achieve a particular task.

2. DEBUG - spot the errors and correct them.

3. DECOMPOSITION - splitting a program up into a number of smaller parts.

4. INPUT - how a computer receives data (e.g. mouse, keyboard, touch screen).

5. LOGICAL REASONING - use of an appropriate system of rules to plan and evaluate your work.

6. OUTPUT - the information produced by the computer for its user (on a screen, through speakers, on a printer

7. PROGRAM - a set of instructions written within a language that can be understood by the computer.

8. REPETITION - one or more instructions are repeated a number of times, until a condition has been satisfied or until the program is stopped.

9. SELECTION - the instructions that are executed are determined by whether a particular condition is met.

10. SEQUENCE - to place programming instructions in order, with each one executed one after another.

11. SIMULATION - using computers to model the real world.

12. UNAMBIGUOUS - clear and cannot be understood wrongly.

13. VARIABLE - a part of a program that can be changed or change (e.g. a name of an Animal).