Computing @bsjs

2023 - 2024

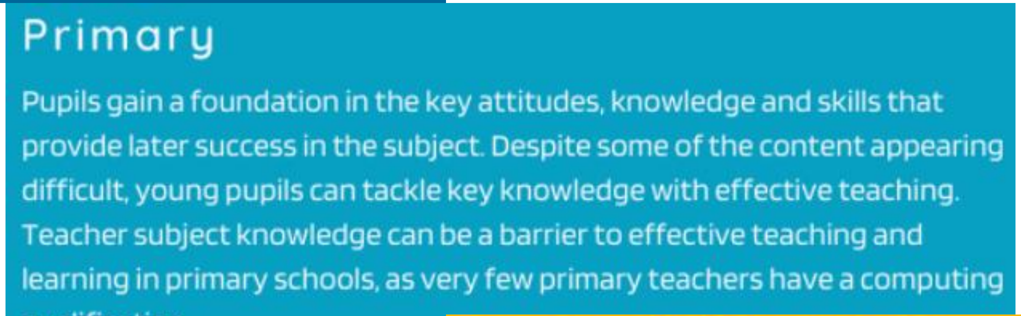
Intent

At BSJS we are preparing our children to be computational thinkers in order to be able to implement technology as an aid to learning and to stretch their understanding of how computers can be utilized for our changing needs.

It is important for us that children are safely able to navigate and make considerate choices in an increasingly connected digital society.

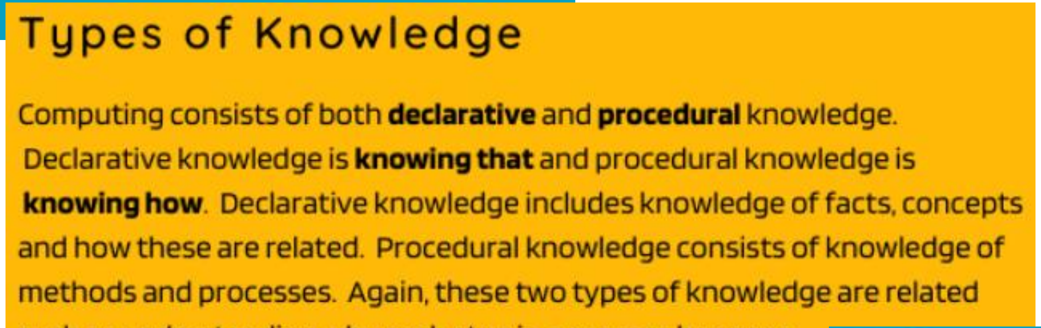
Rationale

Ofsted Research Review



Primary

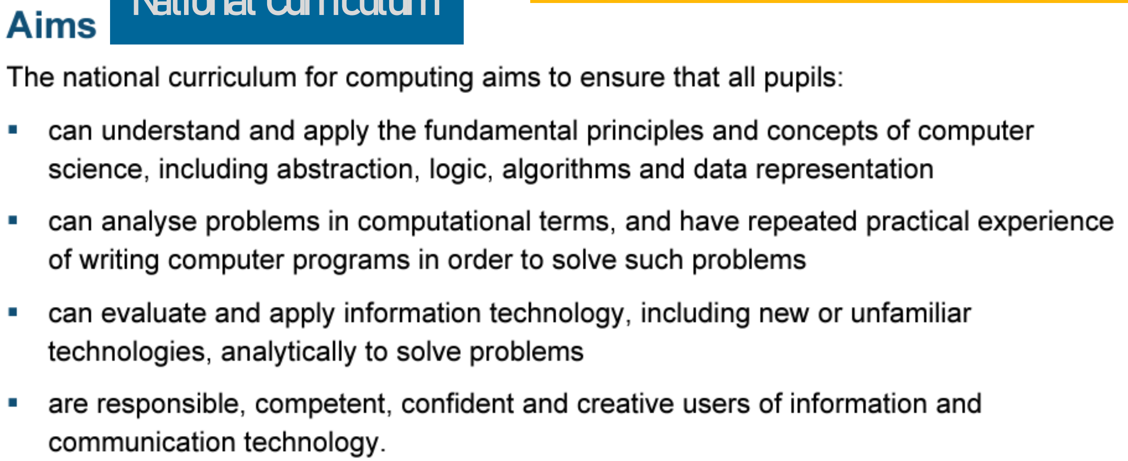
Pupils gain a foundation in the key attitudes, knowledge and skills that provide later success in the subject. Despite some of the content appearing difficult, young pupils can tackle key knowledge with effective teaching. Teacher subject knowledge can be a barrier to effective teaching and learning in primary schools, as very few primary teachers have a computing qualification.



Types of Knowledge

Computing consists of both **declarative** and **procedural** knowledge. Declarative knowledge is **knowing that** and procedural knowledge is **knowing how**. Declarative knowledge includes knowledge of facts, concepts and how these are related. Procedural knowledge consists of knowledge of methods and processes. Again, these two types of knowledge are related such as understanding why each step in a process happens.

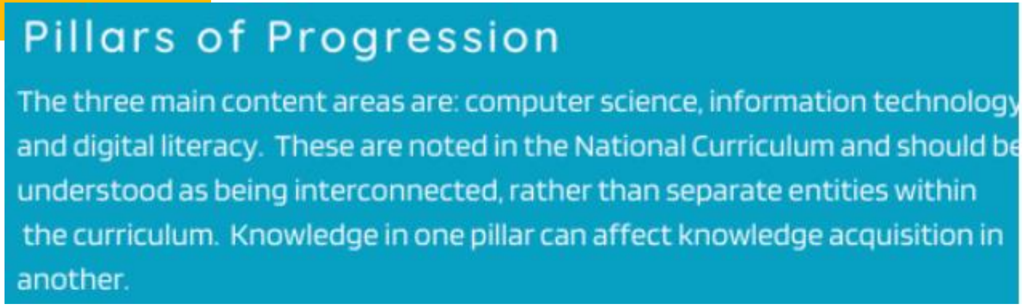
National Curriculum



Aims

The national curriculum for computing aims to ensure that all pupils:

- can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation
- can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
- can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems
- are responsible, competent, confident and creative users of information and communication technology.



Pillars of Progression

The three main content areas are: computer science, information technology and digital literacy. These are noted in the National Curriculum and should be understood as being interconnected, rather than separate entities within the curriculum. Knowledge in one pillar can affect knowledge acquisition in another.

End of KS2

PurpleMash Outcomes

	Computer Science				Information Technology		Digital Literacy
Statement	Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.	Use sequence, selection and repetition in programs; work with variables and various forms of input and output.	Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.	Understand computer networks, including the internet; how they can provide multiple services, such as the World Wide Web, and the opportunities they offer for communication and collaboration.	Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content.	Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information.	Use technology safely, respectfully and responsibly; recognise acceptable/ unacceptable behaviour; identify a range of ways to report concern about content and contact.
Outcome	*Children are able to turn a more complex programming task into an algorithm by identifying the important aspects of the task (abstraction) and then decomposing them in a logical way using their knowledge of possible coding structures and applying skills from previous programs. Children test and debug their program as they go and use logical methods to identify the cause of bugs, demonstrating a systematic approach to try to identify a particular line of code causing a* ***problem***.	*Children translate algorithms that include sequence, selection and repetition into code and their own designs show that they are thinking of how to accomplish the set task in code utilising such structures, including nesting structures within each other. Coding displays an improving understanding of variables in coding, outputs such as sound and movement, inputs from the user of the program such as button clicks and the* ***value of functions***.	*Children are able to interpret a program in parts and can make logical attempts to put the separate parts of a complex algorithm together to explain the* ***program as a whole***.	*Children understand and can explain in some depth the difference between the internet and the World Wide Web. Children know what a WAN and LAN are and can describe how they access the* ***Internet in school***.	*Children readily apply filters when searching for digital content. They are able to explain in detail how credible a webpage is and the information it contains. They compare a range of digital content sources and are able to rate them in terms of content quality and accuracy. Children use critical thinking skills in everyday use of online communication.*	*Children make clear connections to the audience when designing and creating digital content. The children design and create their own blogs to become a content creator on the Internet, e.g.* ***2Blog***. *They are able to use criteria to evaluate the quality of digital solutions and are able to identify improvements, making some refinements.*	*Children demonstrate the safe and respectful use of a range of different technologies and online services. They identify more discreet inappropriate behaviours through developing critical thinking, e.g.* ***2Respond*** *activities. They recognise the value in preserving their privacy when online for their own and other people's safety.*

Key stage 2

Pupils should be taught to:

- design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- use sequence, selection, and repetition in programs; work with variables and various forms of input and output
- use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
- understand computer networks including the internet; how they can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration
- use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content
- select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information
- use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.

National Curriculum

Implementation

Computer Science

This covers knowledge of computers and computation including data, system architecture, algorithms and programming and it is the core of computing, underpinning the whole subject. It provides the foundational knowledge required to understand and interpret the other areas of the computing curriculum. Computing curricula should therefore be rich in computer science knowledge.

CS

Information Technology

IT

Knowledge in this pillar refers to digital artefacts and computing contexts. Digital artefacts are the digital objects made by humans. Pupils need to acquire both declarative and procedural knowledge of how to create digital artefacts. Computing contexts refer to the knowledge about how computing has played a significant part in our history and how it can transform our daily lives. This is classed as 'empowering knowledge'.

Digital Literacy

DL

This pillar of progress consists of the knowledge pupils need to use digital devices safely, effectively and discerningly. Adults should not assume pupils are digital natives; pupils need to be taught how to use the devices intended by the curriculum. e-Safety should be carefully planned so that pupils learn age-appropriate content, building on prior knowledge.

Concepts

Computer Science

CS

Coding
Networking
Hardware

Information Technology

IT

Spreadsheets
Databases
Animation
Presenting and Publishing

Digital Literacy

DL

Being safe online
Email
Social Media

Declarative and Procedural Knowledge

Form of knowledge	Computer science	Information technology	Digital literacy
Declarative What? Facts	Programming syntax The purpose and function of different logic gates	Principles of effective multimedia design Spreadsheet formulae	Features of unreliable content
Procedural How? Skills	Performing binary addition Implementing a repeat in a programming language	Setting up a slide master Applying conditional formatting	How to perform an advanced web search

Computing is a constant blend of these concepts

Computing and SEND

We have a wide range of equipment:

- Tech lab – personalized suite of 30 desktop computers
- iPad trolley – 24
- Chromebook trolleys for Y6

Our equipment is accessible and portable for everyone:

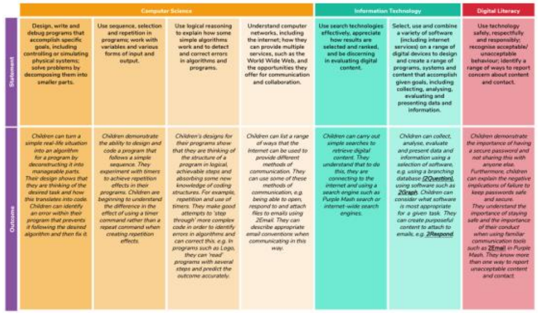
- Versatile iPads
- Lightweight Chromebooks

2Type encourages children to build typing skills and make computers more accessible in a fun and engaging way

2Type

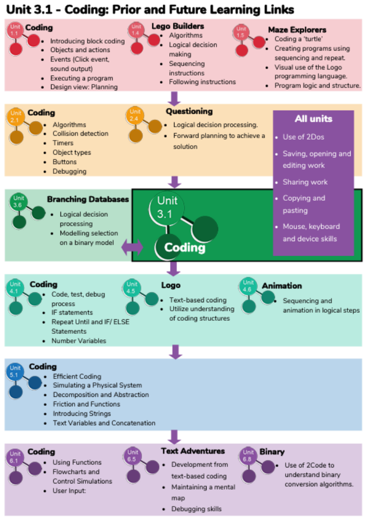
Lessons follow similar patterns and involve aspects that appeal to all learning styles

Progression – lessons are structured into smaller steps that build towards achieving the overall objective



	Computer Science				Information Technology		Digital Literacy
Statement	Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.	Use sequence, selection and repetition in programs; work with variables and various forms of input and output.	Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.	Understand computer networks, including the internet; how they can provide multiple services, such as the World Wide Web, and the opportunities they offer for communication and collaboration.	Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content.	Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information.	Use technology safely, respectfully and responsibly; recognise acceptable/ unacceptable behaviour; identify a range of ways to report concern about content and contact.
Outcome	*Children can turn a simple real-life situation into an algorithm for a program by deconstructing it into manageable parts. Their design shows that they are thinking of the desired task and how this translates into code. Children can identify an error within their program that prevents it following the desired algorithm and then fix it.*	*Children demonstrate the ability to design and code a program that follows a simple sequence. They experiment with timers to achieve repetition effects in their programs. Children are beginning to understand the difference in the effect of using a timer command rather than a repeat command when creating repetition effects.*	*Children's designs for their programs show that they are thinking of the structure of a program in logical, achievable steps and absorbing some new knowledge of coding structures. For example, repetition and use of timers. They make good attempts to 'step through' more complex code in order to identify errors in algorithms and can correct this. e.g. In programs such as Logo, they can 'read' programs with several steps and predict the outcome accurately.*	*Children can list a range of ways that the internet can be used to provide different methods of communication. They can use some of these methods of communication, e.g. being able to open, respond to and attach files to emails using 2Email. They can describe appropriate email conventions when communicating in this way.*	*Children can carry out simple searches to retrieve digital content. They understand that to do this, they are connecting to the internet and using a search engine such as Purple Mash search or internet-wide search engines.*	*Children can collect, analyse, evaluate and present data and information using a selection of software, e.g. using a branching database (2Question), using software such as 2Graph. Children can consider what software is most appropriate for a given task. They can create purposeful content to attach to emails, e.g. 2Respond.*	*Children demonstrate the importance of having a secure password and not sharing this with anyone else. Furthermore, children can explain the negative implications of failure to keep passwords safe and secure. They understand the importance of staying safe and the importance of their conduct when using familiar communication tools such as 2Email in Purple Mash. They know more than one way to report unacceptable content and contact.*

A range of crucial documents can be used to aid accurate assessment and inform gaps, past and future learning



Unit 3.1 - Coding: Prior and Future Learning Links

Unit 1.1 Coding
- Introducing block coding
- Objects and actions
- Events (Click event, sound output)
- Executing a program
- Design view: Planning

Unit 1.4 Lego Builders
- Algorithms
- Logical decision making
- Sequencing instructions
- Following instructions

Unit 1.5 Maze Explorers
- Coding a 'turtle'
- Creating programs using sequencing and repeat.
- Visual use of the Logo programming language.
- Program logic and structure.

Unit 2.1 Coding
- Algorithms
- Collision detection
- Timers
- Object types
- Buttons
- Debugging

Unit 2.4 Questioning
- Logical decision processing.
- Forward planning to achieve a solution

All units
- Use of 2Dos
- Saving, opening and editing work
- Sharing work
- Copying and pasting
- Mouse, keyboard and device skills

Unit 3.6 Branching Databases
- Logical decision processing
- Modelling selection on a binary model

Unit 3.1 Coding

Unit 4.1 Coding
- Code, test, debug process
- IF statements
- Repeat Until and IF/ ELSE Statements
- Number Variables

Unit 4.5 Logo
- Text-based coding
- Utilize understanding of coding structures

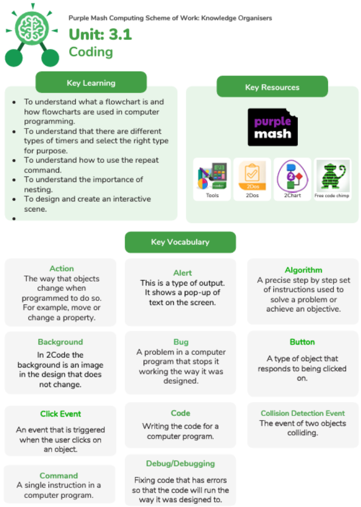
Unit 4.6 Animation
- Sequencing and animation in logical steps

Unit 5.1 Coding
- Efficient Coding
- Simulating a Physical System
- Decomposition and Abstraction
- Friction and Functions
- Introducing Strings
- Text Variables and Concatenation

Unit 6.1 Coding
- Using Functions
- Flowcharts and Control Simulations
- User Input:

Unit 6.5 Text Adventures
- Development from text-based coding
- Maintaining a mental map
- Debugging skills

Unit 6.8 Binary
- Use of 2Code to understand binary conversion algorithms.



Purple Mash Computing Scheme of Work: Knowledge Organisers

Unit: 3.1
Coding

Key Learning
- To understand what a flowchart is and how flowcharts are used in computer programming.
- To understand that there are different types of timers and select the right type for purpose.
- To understand how to use the repeat command.
- To understand the importance of nesting.
- To design and create an interactive scene.
-

Key Resources

purple mash

Tools · 2Dos · 2Chart · Free code chimp

Key Vocabulary

Action The way that objects change when programmed to do so. For example, move or change a property.

Alert This is a type of output. It shows a pop-up of text on the screen.

Algorithm A precise step by step set of instructions used to solve a problem or achieve an objective.

Background In 2Code the background is an image in the design that does not change.

Bug A problem in a computer program that stops it working the way it was designed.

Button A type of object that responds to being clicked on.

Click Event An event that is triggered when the user clicks on an object.

Code Writing the code for a computer program.

Collision Detection Event The event of two objects colliding.

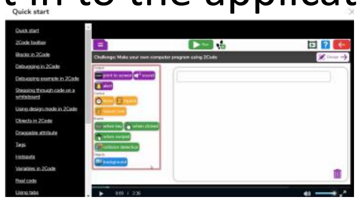
Command A single instruction in a computer program.

Debug/Debugging Fixing code that has errors so that the code will run the way it was designed to.

Learning Coach support

Hint

Helpful hints and videos built in to the applications

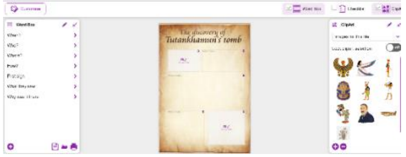


Quick start

Quick start
2Code toolbar
Blocks in 2Code
Debugging in 2Code
Debugging example in 2Code
Stepping through code on a whiteboard
Using design mode in 2Code
Objects in 2Code
Draggable attribute
Tags
Hotspots
Variables in 2Code
Real code
Using tabs

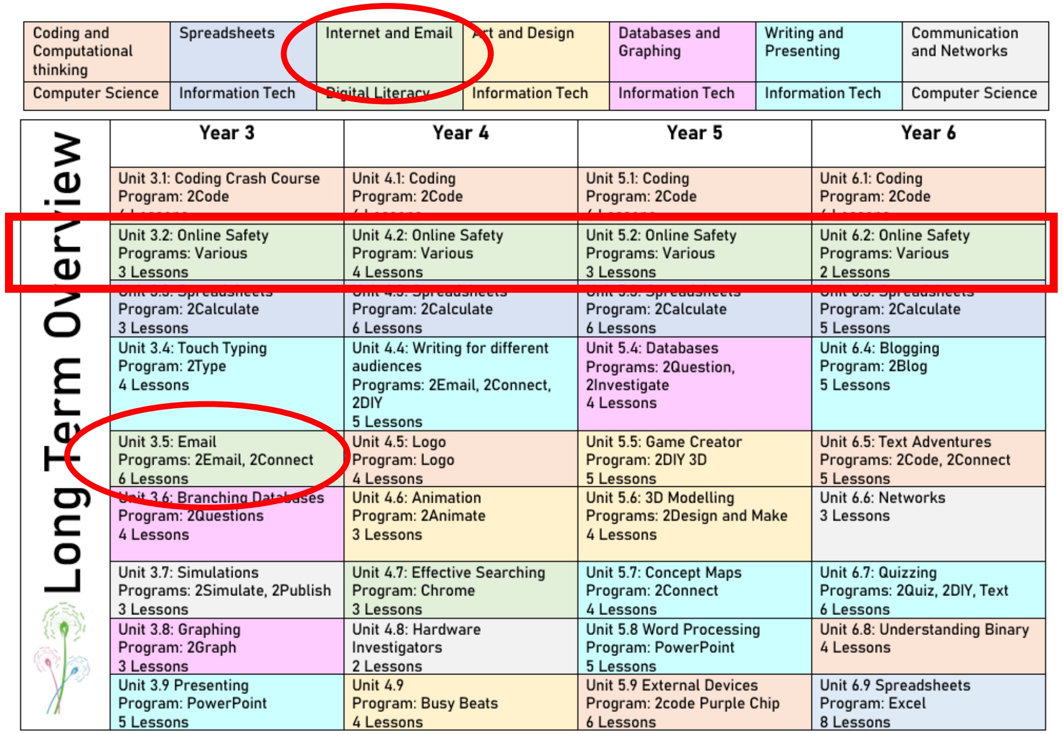
Challenge: Make your own computer program using 2Code

Templates and scaffolds are available for many applications



Simplified 'crash courses' designed for children who have missed or in need or overlearning from previous year/unit

Internet Safety



Coding and Computational thinking	Spreadsheets	Internet and Email	Art and Design	Databases and Graphing	Writing and Presenting	Communication and Networks
Computer Science	Information Tech	Digital Literacy	Information Tech	Information Tech	Information Tech	Computer Science

Long Term Overview

Year 3	Year 4	Year 5	Year 6
Unit 3.1: Coding Crash Course Program: 2Code 6 Lessons	Unit 4.1: Coding Program: 2Code 6 Lessons	Unit 5.1: Coding Program: 2Code 6 Lessons	Unit 6.1: Coding Program: 2Code 6 Lessons
Unit 3.2: Online Safety Programs: Various 3 Lessons	Unit 4.2: Online Safety Program: Various 4 Lessons	Unit 5.2: Online Safety Programs: Various 3 Lessons	Unit 6.2: Online Safety Programs: Various 2 Lessons
Unit 3.3: Spreadsheets Program: 2Calculate 3 Lessons	Unit 4.3: Spreadsheets Program: 2Calculate 6 Lessons	Unit 5.3: Spreadsheets Program: 2Calculate 6 Lessons	Unit 6.3: Spreadsheets Program: 2Calculate 5 Lessons
Unit 3.4: Touch Typing Program: 2Type 4 Lessons	Unit 4.4: Writing for different audiences Programs: 2Email, 2Connect, 2DIY 5 Lessons	Unit 5.4: Databases Programs: 2Question, 2Investigate 4 Lessons	Unit 6.4: Blogging Program: 2Blog 5 Lessons
Unit 3.5: Email Programs: 2Email, 2Connect 6 Lessons	Unit 4.5: Logo Program: Logo 4 Lessons	Unit 5.5: Game Creator Program: 2DIY 3D 5 Lessons	Unit 6.5: Text Adventures Programs: 2Code, 2Connect 5 Lessons
Unit 3.6: Branching Databases Program: 2Questions 4 Lessons	Unit 4.6: Animation Program: 2Animate 3 Lessons	Unit 5.6: 3D Modelling Programs: 2Design and Make 4 Lessons	Unit 6.6: Networks 3 Lessons
Unit 3.7: Simulations Programs: 2Simulate, 2Publish 3 Lessons	Unit 4.7: Effective Searching Program: Chrome 3 Lessons	Unit 5.7: Concept Maps Program: 2Connect 4 Lessons	Unit 6.7: Quizzing Programs: 2Quiz, 2DIY, Text 6 Lessons
Unit 3.8: Graphing Program: 2Graph 3 Lessons	Unit 4.8: Hardware Investigators 2 Lessons	Unit 5.8 Word Processing Program: PowerPoint 5 Lessons	Unit 6.8: Understanding Binary 4 Lessons
Unit 3.9 Presenting Program: PowerPoint 5 Lessons	Unit 4.9 Program: Busy Beats 4 Lessons	Unit 5.9 External Devices Program: 2code Purple Chip 6 Lessons	Unit 6.9 Spreadsheets Program: Excel 8 Lessons

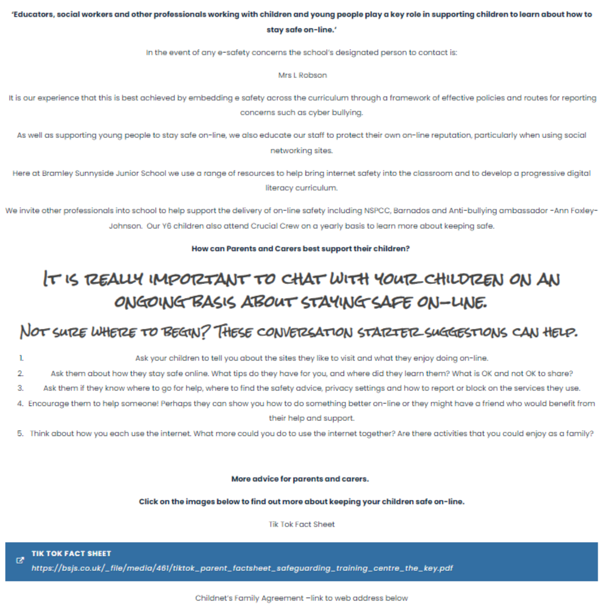
Internet safety is considered a vital part of the curriculum. It is taught within one unit in every year group.

We also promote Internet safety through the Safer Internet Day scheme each year.

Anne Foxley Johnson advises and runs workshops for children on behaviour whilst online .

Our Internet filtering system is excellent and can be adapted to meet new challenges

Our website hosts a detailed parent guide with many links to information to extend Internet safety away from school



'Educators, social workers and other professionals working with children and young people play a key role in supporting children to learn about how to stay safe on-line.'

In the event of any e-safety concerns the school's designated person to contact is:

Mrs L Robson

It is our experience that this is best achieved by embedding e safety across the curriculum through a framework of effective policies and routes for reporting concerns such as cyber bullying.

As well as supporting young people to stay safe on-line, we also educate our staff to protect their own on-line reputation, particularly when using social networking sites.

Here at Bramley Sunnyside Junior School we use a range of resources to help bring internet safety into the classroom and to develop a progressive digital literacy curriculum.

We invite other professionals into school to help support the delivery of on-line safety including NSPCC, Barnados and Anti-bullying ambassador -Ann Foxley-Johnson. Our Y6 children also attend Crucial Crew on a yearly basis to learn more about keeping safe.

How can Parents and Carers best support their children?

IT IS REALLY IMPORTANT TO CHAT WITH YOUR CHILDREN ON AN ONGOING BASIS ABOUT STAYING SAFE ON-LINE.

NOT SURE WHERE TO BEGIN? THESE CONVERSATION STARTER SUGGESTIONS CAN HELP.

1. Ask your children to tell you about the sites they like to visit and what they enjoy doing on-line.
2. Ask them about how they stay safe online. What tips do they have for you, and where did they learn them? What is OK and not OK to share?
3. Ask them if they know where to go for help, where to find the safety advice, privacy settings and how to report or block on the services they use.
4. Encourage them to help someone! Perhaps they can show you how to do something better on-line or they might have a friend who would benefit from their help and support.
5. Think about how you each use the internet. What more could you do to use the internet together? Are there activities that you could enjoy as a family?

More advice for parents and carers.

Click on the images below to find out more about keeping your children safe on-line.

Tik Tok Fact Sheet

TIK TOK FACT SHEET
https://bsjs.co.uk/_file/media/461/tiktok_parent_factsheet_safeguarding_training_centre_the_key.pdf

Childnet's Family Agreement –link to web address below

https://bsjs.co.uk/e-safety

20th Edition
Safer Internet Day 2023 | Tuesday 7 February
Together for a better internet
www.saferinternetday.org

Whole-School

Coding and Computational thinking	Spreadsheets	Internet and Email	Art and Design	Databases and Graphing	Writing and Presenting	Communication and Networks
Computer Science	Information Tech	Digital Literacy	Information Tech	Information Tech	Information Tech	Computer Science

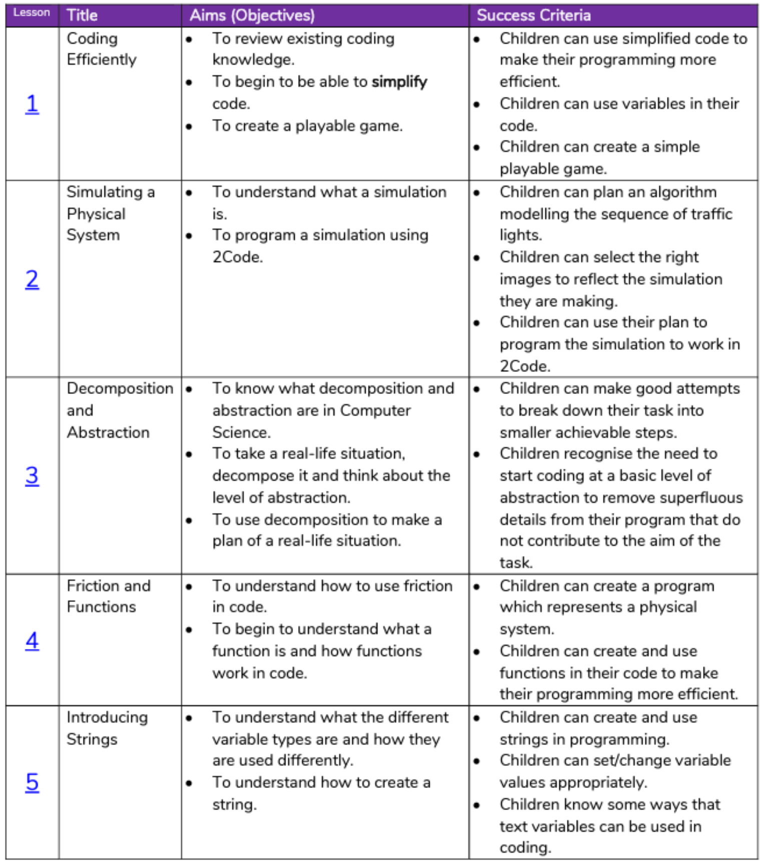
Long Term Overview

Year 3	Year 4	Year 5	Year 6
Unit 3.1: Coding Crash Course Program: 2Code 6 Lessons	Unit 4.1: Coding Program: 2Code 6 Lessons	Unit 5.1: Coding Program: 2Code 6 Lessons	Unit 6.1: Coding Program: 2Code 6 Lessons
Unit 3.2: Online Safety Programs: Various 3 Lessons	Unit 4.2: Online Safety Program: Various 4 Lessons	Unit 5.2: Online Safety Programs: Various 3 Lessons	Unit 6.2: Online Safety Programs: Various 2 Lessons
Unit 3.3: Spreadsheets Program: 2Calculate 3 Lessons	Unit 4.3: Spreadsheets Program: 2Calculate 6 Lessons	Unit 5.3: Spreadsheets Program: 2Calculate 6 Lessons	Unit 6.3: Spreadsheets Program: 2Calculate 5 Lessons
Unit 3.4: Touch Typing Program: 2Type 4 Lessons	Unit 4.4: Writing for different audiences Programs: 2Email, 2Connect, 2DIY 5 Lessons	Unit 5.4: Databases Programs: 2Question, 2Investigate 4 Lessons	Unit 6.4: Blogging Program: 2Blog 5 Lessons
Unit 3.5: Email Programs: 2Email, 2Connect 6 Lessons	Unit 4.5: Logo Program: Logo 4 Lessons	Unit 5.5: Game Creator Program: 2DIY 3D 5 Lessons	Unit 6.5: Text Adventures Programs: 2Code, 2Connect 5 Lessons
Unit 3.6: Branching Databases Program: 2Questions 4 Lessons	Unit 4.6: Animation Program: 2Animate 3 Lessons	Unit 5.6: 3D Modelling Programs: 2Design and Make 4 Lessons	Unit 6.6: Networks 3 Lessons
Unit 3.7: Simulations Programs: 2Simulate, 2Publish 3 Lessons	Unit 4.7: Effective Searching Program: Chrome 3 Lessons	Unit 5.7: Concept Maps Program: 2Connect 4 Lessons	Unit 6.7: Quizzing Programs: 2Quiz, 2DIY, Text 6 Lessons
Unit 3.8: Graphing Program: 2Graph 3 Lessons	Unit 4.8: Hardware Investigators 2 Lessons	Unit 5.8 Word Processing Program: PowerPoint 5 Lessons	Unit 6.8: Understanding Binary 4 Lessons
Unit 3.9 Presenting Program: PowerPoint 5 Lessons	Unit 4.9 Program: Busy Beats 4 Lessons	Unit 5.9 External Devices Program: 2code Purple Chip 6 Lessons	Unit 6.9 Spreadsheets Program: Excel 8 Lessons

Long Term

Each unit is carefully planned and sequenced to allow learners to progress by building the necessary skills and knowledge to meet the demands of the national curriculum

Unit 5.1 Example

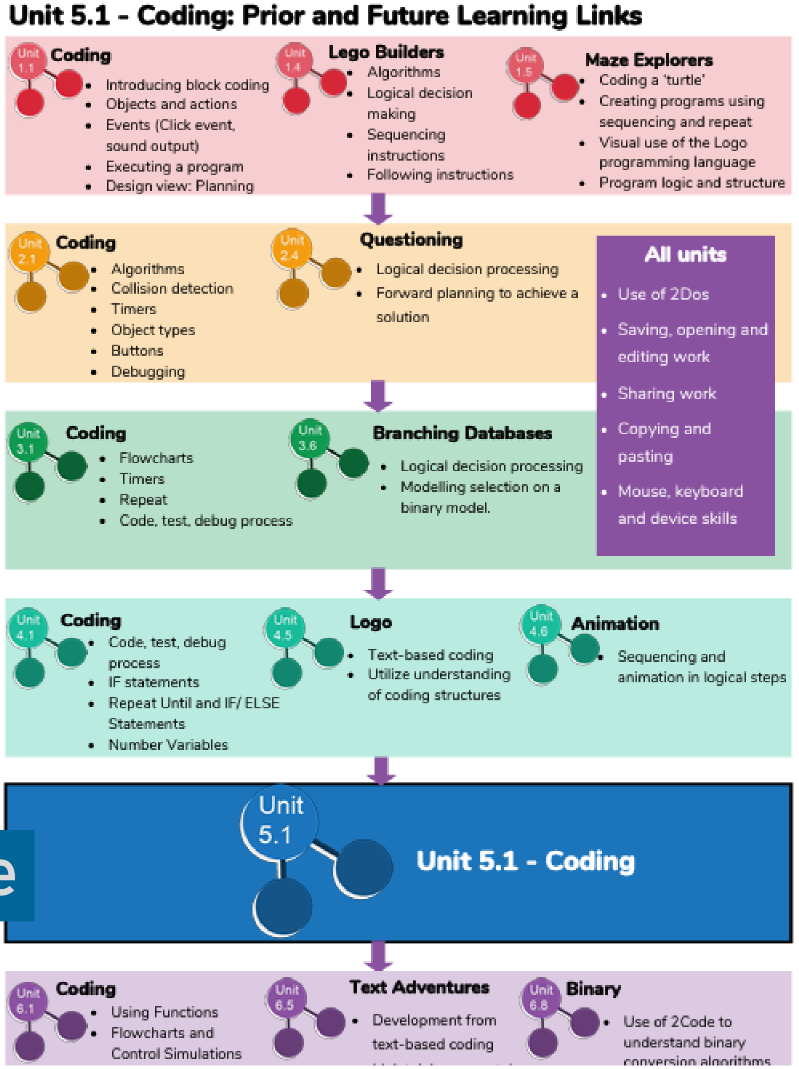


Lesson	Title	Aims (Objectives)	Success Criteria
1	Coding Efficiently	• To review existing coding knowledge. • To begin to be able to **simplify** code. • To create a playable game.	• Children can use simplified code to make their programming more efficient. • Children can use variables in their code. • Children can create a simple playable game.
2	Simulating a Physical System	• To understand what a simulation is. • To program a simulation using 2Code.	• Children can plan an algorithm modelling the sequence of traffic lights. • Children can select the right images to reflect the simulation they are making. • Children can use their plan to program the simulation to work in 2Code.
3	Decomposition and Abstraction	• To know what decomposition and abstraction are in Computer Science. • To take a real-life situation, decompose it and think about the level of abstraction. • To use decomposition to make a plan of a real-life situation.	• Children can make good attempts to break down their task into smaller achievable steps. • Children recognise the need to start coding at a basic level of abstraction to remove superfluous details from their program that do not contribute to the aim of the task.
4	Friction and Functions	• To understand how to use friction in code. • To begin to understand what a function is and how functions work in code.	• Children can create a program which represents a physical system. • Children can create and use functions in their code to make their programming more efficient.
5	Introducing Strings	• To understand what the different variable types are and how they are used differently. • To understand how to create a string.	• Children can create and use strings in programming. • Children can set/change variable values appropriately. • Children know some ways that text variables can be used in coding.

All units are sequenced with a clear progression in skills between lessons

Medium Term

Prior & Future



Unit 5.1 - Coding: Prior and Future Learning Links

Unit 1.1 Coding
- Introducing block coding
- Objects and actions
- Events (Click event, sound output)
- Executing a program
- Design view: Planning

Unit 1.4 Lego Builders
- Algorithms
- Logical decision making
- Sequencing instructions
- Following instructions

Unit 1.5 Maze Explorers
- Coding a 'turtle'
- Creating programs using sequencing and repeat
- Visual use of the Logo programming language
- Program logic and structure

Unit 2.1 Coding
- Algorithms
- Collision detection
- Timers
- Object types
- Buttons
- Debugging

Unit 2.4 Questioning
- Logical decision processing
- Forward planning to achieve a solution

All units
- Use of 2Dos
- Saving, opening and editing work
- Sharing work
- Copying and pasting
- Mouse, keyboard and device skills

Unit 3.1 Coding
- Flowcharts
- Timers
- Repeat
- Code, test, debug process

Unit 3.6 Branching Databases
- Logical decision processing
- Modelling selection on a binary model.

Unit 4.1 Coding
- Code, test, debug process
- IF statements
- Repeat Until and IF/ ELSE Statements
- Number Variables

Unit 4.5 Logo
- Text-based coding
- Utilize understanding of coding structures

Unit 4.6 Animation
- Sequencing and animation in logical steps

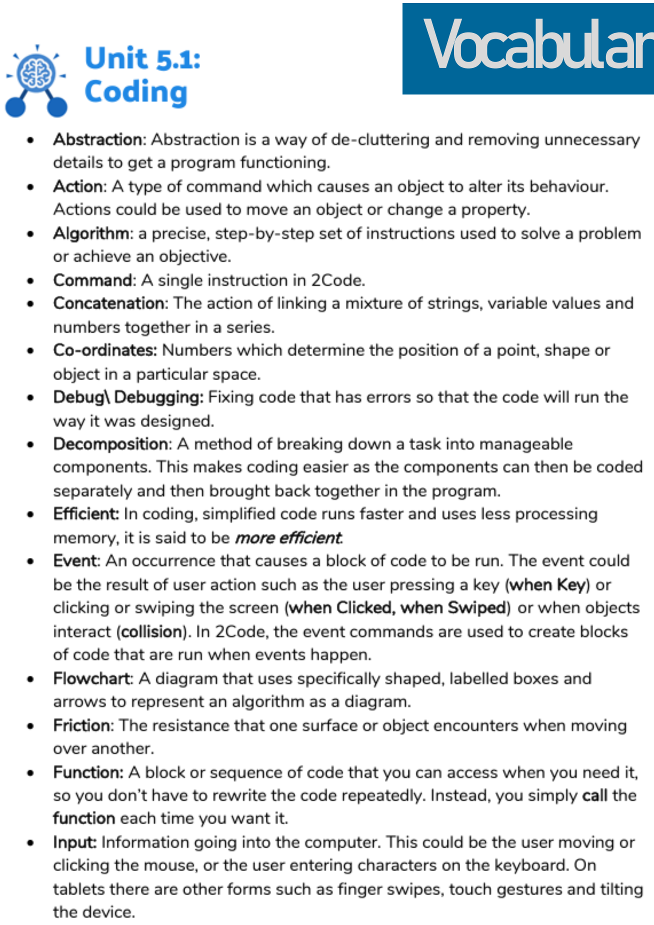
Unit 5.1 Unit 5.1 - Coding

Unit 6.1 Coding
- Using Functions
- Flowcharts and Control Simulations

Unit 6.5 Text Adventures
- Development from text-based coding

Unit 6.8 Binary
- Use of 2Code to understand binary conversion algorithms

Vocabulary

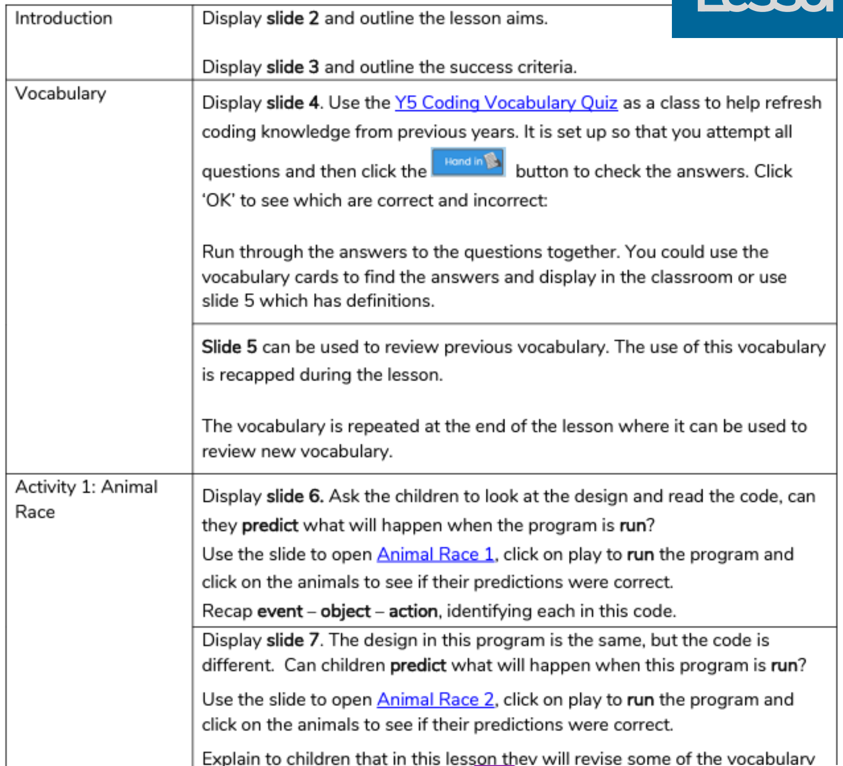


Unit 5.1: Coding

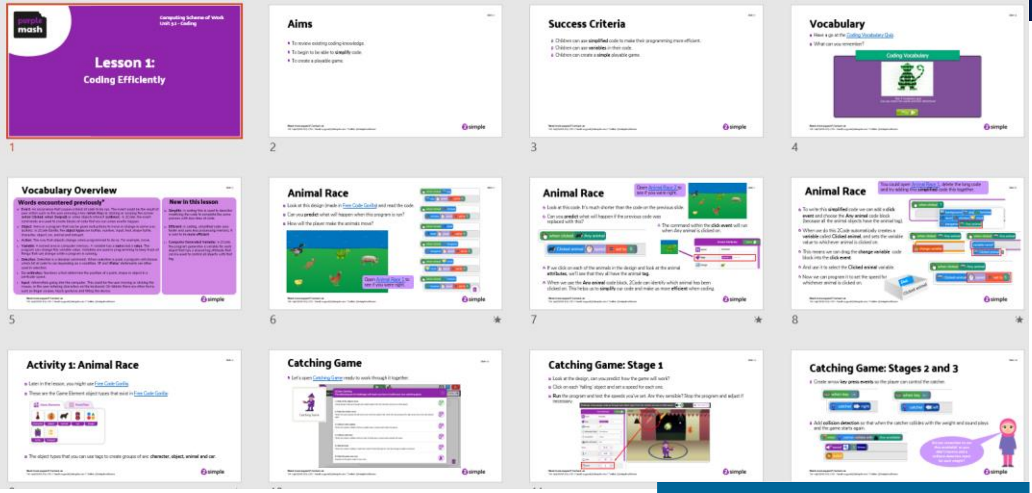
- **Abstraction**: Abstraction is a way of de-cluttering and removing unnecessary details to get a program functioning.
- **Action**: A type of command which causes an object to alter its behaviour. Actions could be used to move an object or change a property.
- **Algorithm**: a precise, step-by-step set of instructions used to solve a problem or achieve an objective.
- **Command**: A single instruction in 2Code.
- **Concatenation**: The action of linking a mixture of strings, variable values and numbers together in a series.
- **Co-ordinates:** Numbers which determine the position of a point, shape or object in a particular space.
- **Debug\ Debugging:** Fixing code that has errors so that the code will run the way it was designed.
- **Decomposition**: A method of breaking down a task into manageable components. This makes coding easier as the components can then be coded separately and then brought back together in the program.
- **Efficient:** In coding, simplified code runs faster and uses less processing memory, it is said to be ***more efficient.***
- **Event**: An occurrence that causes a block of code to be run. The event could be the result of user action such as the user pressing a key (**when Key**) or clicking or swiping the screen (**when Clicked, when Swiped**) or when objects interact (**collision**). In 2Code, the event commands are used to create blocks of code that are run when events happen.
- **Flowchart**: A diagram that uses specifically shaped, labelled boxes and arrows to represent an algorithm as a diagram.
- **Friction**: The resistance that one surface or object encounters when moving over another.
- **Function:** A block or sequence of code that you can access when you need it, so you don't have to rewrite the code repeatedly. Instead, you simply **call** the **function** each time you want it.
- **Input:** Information going into the computer. This could be the user moving or clicking the mouse, or the user entering characters on the keyboard. On tablets there are other forms such as finger swipes, touch gestures and tilting the device.

Lesson by lesson

Lesson plans



Introduction	Display **slide 2** and outline the lesson aims. Display **slide 3** and outline the success criteria.
Vocabulary	Display **slide 4**. Use the Y5 Coding Vocabulary Quiz as a class to help refresh coding knowledge from previous years. It is set up so that you attempt all questions and then click the Hand in button to check the answers. Click 'OK' to see which are correct and incorrect: Run through the answers to the questions together. You could use the vocabulary cards to find the answers and display in the classroom or use slide 5 which has definitions.
	Slide 5 can be used to review previous vocabulary. The use of this vocabulary is recapped during the lesson. The vocabulary is repeated at the end of the lesson where it can be used to review new vocabulary.
Activity 1: Animal Race	Display **slide 6.** Ask the children to look at the design and read the code, can they **predict** what will happen when the program is **run**? Use the slide to open Animal Race 1, click on play to **run** the program and click on the animals to see if their predictions were correct. Recap **event** – **object** – **action**, identifying each in this code.
	Display **slide 7**. The design in this program is the same, but the code is different. Can children **predict** what will happen when this program is **run**? Use the slide to open Animal Race 2, click on play to **run** the program and click on the animals to see if their predictions were correct.
	Explain to children that in this lesson they will revise some of the vocabulary and concepts they have learnt in to make their programming more
	Discuss what it might mean to ma
	Return to **slide 6** and begin to lool



1 Computing Scheme of Work Unit 5.1 – Coding

purple mash

Lesson 1:
Coding Efficiently

2 Aims

- To review existing coding knowledge.
- To begin to be able to **simplify** code.
- To create a playable game.

3 Success Criteria

- Children can use **simplified** code to make their programming more efficient.
- Children can use **variables** in their code.
- Children can create a **simple** playable game.

4 Vocabulary

- Have a go at the Coding Vocabulary Quiz
- What can you remember?

Coding Vocabulary

5 Vocabulary Overview

Words encountered previously*

New in this lesson

6 Animal Race

- Look at this design (made in Free Code Gorilla) and read the code.
- Can you **predict** what will happen when this program is run?
- How will the player make the animals move?

Open Animal Race 1 to see if you were right.

7 Animal Race

- Look at this code. It's much shorter than the code on the previous slide.
- Can you **predict** what will happen if the previous code was replaced with this?

Open Animal Race 2 to see if you were right.

8 Animal Race

Activity 1: Animal Race

- Later in the lesson, you might use Free Code Gorilla
- These are the Game Element object types that exist in Free Code Gorilla

Catching Game

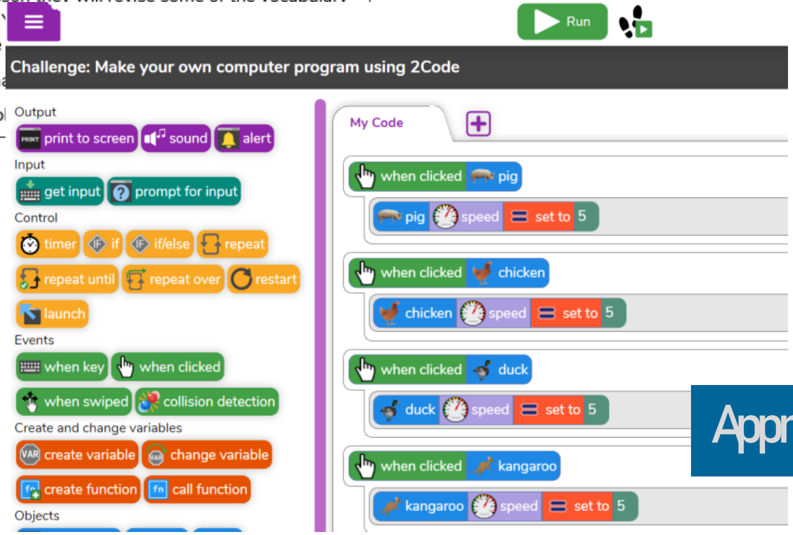
- Let's open Catching Game ready to work through it together.

Catching Game: Stage 1

Catching Game: Stages 2 and 3

Supporting PP

Guidance PowerPoints provided by PurpleMash allow teachers of all levels of subject knowledge to follow or adapt their lessons

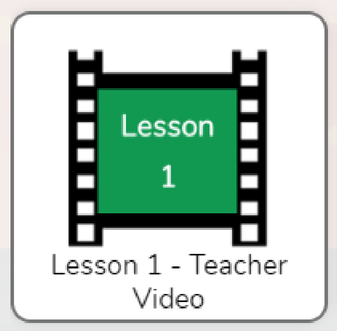


Challenge: Make your own computer program using 2Code

```
when clicked pig
    pig speed = set to 5
when clicked chicken
    chicken speed = set to 5
when clicked duck
    duck speed = set to 5
when clicked kangaroo
    kangaroo speed = set to 5
```

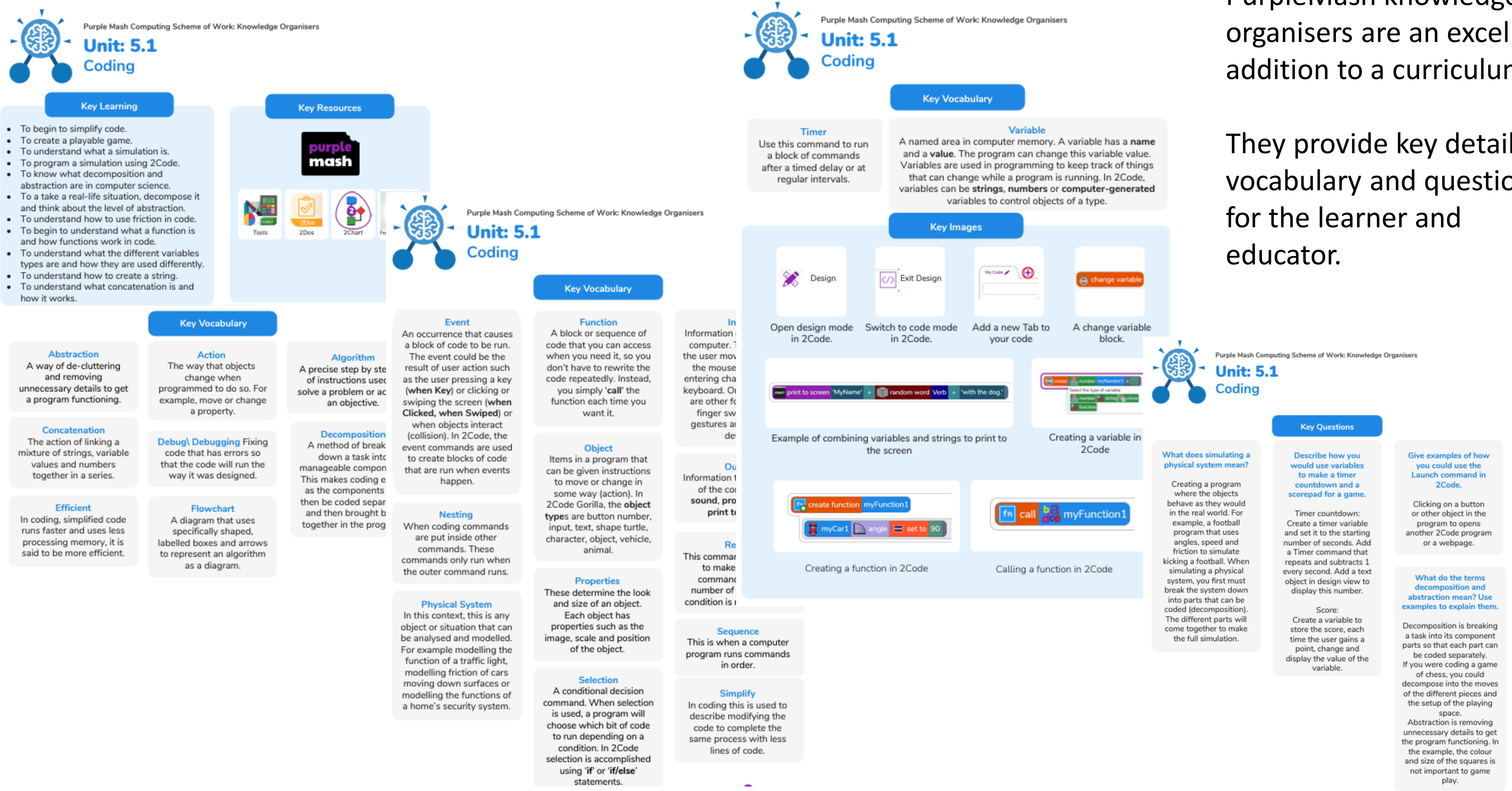
Appropriate Software

High quality, cloud-based software creates an opportunity to achieve all necessary skills in a safe and practical environment



Teacher videos further explain PowerPoints and support teacher knowledge and confidence

Concept Progression – Knowledge Organisers



Purple Mash Computing Scheme of Work: Knowledge Organisers

Unit: 5.1

Coding

Key Learning

- To begin to simplify code.
- To create a playable game.
- To understand what a simulation is.
- To program a simulation using 2Code.
- To know what decomposition and abstraction are in computer science.
- To a take a real-life situation, decompose it and think about the level of abstraction.
- To understand how to use friction in code.
- To begin to understand what a function is and how functions work in code.
- To understand what the different variables types are and how they are used differently.
- To understand how to create a string.
- To understand what concatenation is and how it works.

Key Resources

Key Vocabulary

Abstraction A way of de-cluttering and removing unnecessary details to get a program functioning.

Action The way that objects change when programmed to do so. For example, move or change a property.

Algorithm A precise step by ste of instructions used solve a problem or ac an objective.

Concatenation The action of linking a mixture of strings, variable values and numbers together in a series.

Debug\ Debugging Fixing code that has errors so that the code will run the way it was designed.

Decomposition A method of break down a task into manageable compon This makes coding e as the components then be coded separ and then brought b together in the prog

Efficient In coding, simplified code runs faster and uses less processing memory, it is said to be more efficient.

Flowchart A diagram that uses specifically shaped, labelled boxes and arrows to represent an algorithm as a diagram.

Purple Mash Computing Scheme of Work: Knowledge Organisers

Unit: 5.1

Coding

Key Vocabulary

Event An occurrence that causes a block of code to be run. The event could be the result of user action such as the user pressing a key (**when Key**) or clicking or swiping the screen (**when Clicked, when Swiped**) or when objects interact (collision). In 2Code, the event commands are used to create blocks of code that are run when events happen.

Function A block or sequence of code that you can access when you need it, so you don't have to rewrite the code repeatedly. Instead, you simply '**call**' the function each time you want it.

In Information computer. T the user mov the mouse entering cha keyboard. O are other fo finger sw gestures a de

Nesting When coding commands are put inside other commands. These commands only run when the outer command runs.

Object Items in a program that can be given instructions to move or change in some way (action). In 2Code Gorilla, the **object types** are button number, input, text, shape turtle, character, object, vehicle, animal.

Ou Information t of the co **sound, pro print t**

Physical System In this context, this is any object or situation that can be analysed and modelled. For example modelling the function of a traffic light, modelling friction of cars moving down surfaces or modelling the functions of a home's security system.

Properties These determine the look and size of an object. Each object has properties such as the image, scale and position of the object.

Re This comman to make comman number of condition is

Selection A conditional decision command. When selection is used, a program will choose which bit of code to run depending on a condition. In 2Code selection is accomplished using '**if**' or '**if/else**' statements.

Sequence This is when a computer program runs commands in order.

Simplify In coding this is used to describe modifying the code to complete the same process with less lines of code.

Purple Mash Computing Scheme of Work: Knowledge Organisers

Unit: 5.1

Coding

Key Vocabulary

Timer Use this command to run a block of commands after a timed delay or at regular intervals.

Variable A named area in computer memory. A variable has a **name** and a **value**. The program can change this variable value. Variables are used in programming to keep track of things that can change while a program is running. In 2Code, variables can be **strings**, **numbers** or **computer-generated** variables to control objects of a type.

Key Images

Open design mode in 2Code.

Switch to code mode in 2Code.

Add a new Tab to your code

A change variable block.

Example of combining variables and strings to print to the screen

Creating a variable in 2Code

Creating a function in 2Code

Calling a function in 2Code

Purple Mash Computing Scheme of Work: Knowledge Organisers

Unit: 5.1

Coding

Key Questions

What does simulating a physical system mean?

Creating a program where the objects behave as they would in the real world. For example, a football program that uses angles, speed and friction to simulate kicking a football. When simulating a physical system, you first must break the system down into parts that can be coded (decomposition). The different parts will come together to make the full simulation.

Describe how you would use variables to make a timer countdown and a scorepad for a game.

Timer countdown: Create a timer variable and set it to the starting number of seconds. Add a Timer command that repeats and subtracts 1 every second. Add a text object in design view to display this number.

Score: Create a variable to store the score, each time the user gains a point, change and display the value of the variable.

Give examples of how you could use the Launch command in 2Code.

Clicking on a button or other object in the program to opens another 2Code program or a webpage.

What do the terms decomposition and abstraction mean? Use examples to explain them.

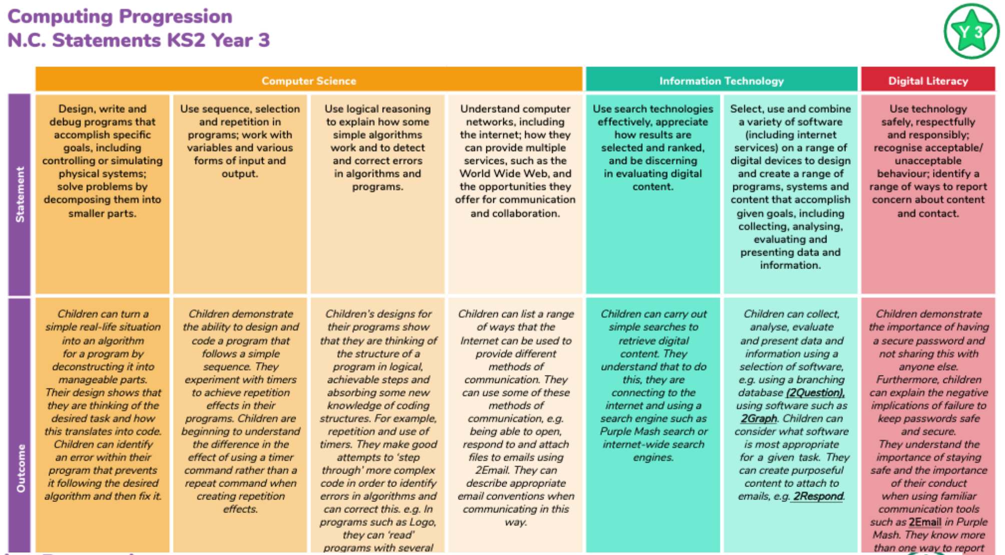
Decomposition is breaking a task into its component parts so that each part can be coded separately. If you were coding a game of chess, you could decompose into the moves of the different pieces and the setup of the playing space. Abstraction is removing unnecessary details to get the program functioning. In the example, the colour and size of the squares is not important to game play.

PurpleMash knowledge organisers are an excellent addition to a curriculum.

They provide key details, vocabulary and questions for the learner and educator.

Assessment

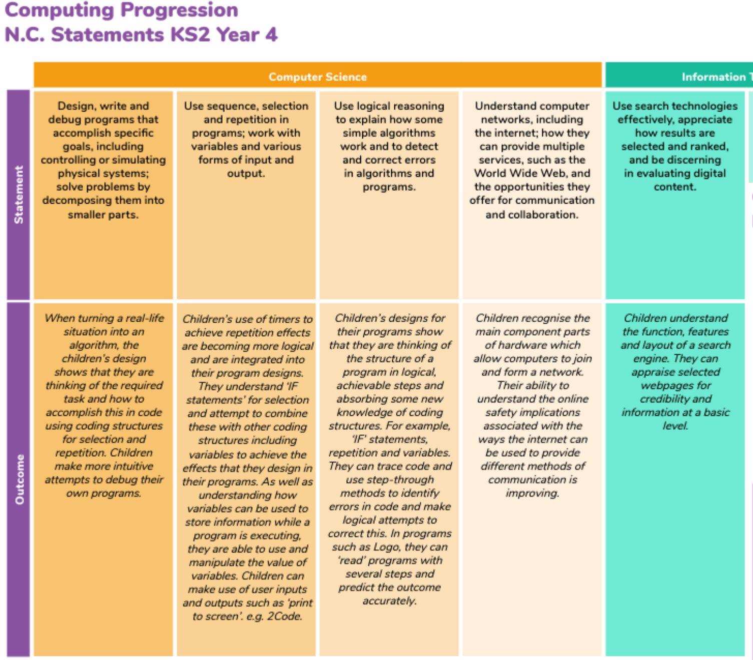
For each year group a progression document matched to the end of KS2 national curriculum expectations explains what the knowledge and skills of an expected level child might look like for each year group.



Computing Progression
N.C. Statements KS2 Year 3

Y 3

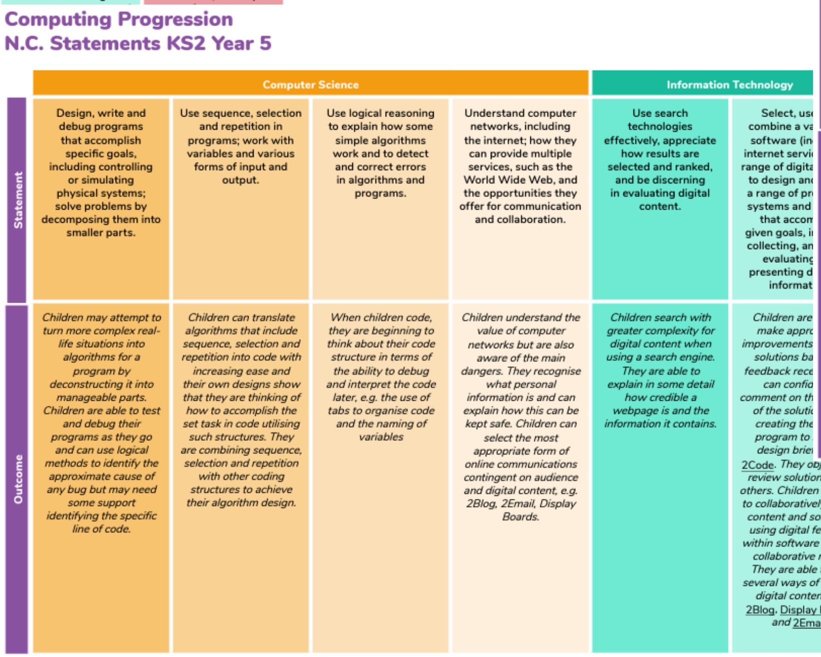
	Computer Science				Information Technology		Digital Literacy
Statement	Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.	Use sequence, selection and repetition in programs; work with variables and various forms of input and output.	Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.	Understand computer networks, including the internet; how they can provide multiple services, such as the World Wide Web, and the opportunities they offer for communication and collaboration.	Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content.	Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information.	Use technology safely, respectfully and responsibly; recognise acceptable/ unacceptable behaviour; identify a range of ways to report concern about content and contact.
Outcome	*Children can turn a simple real-life situation into an algorithm for a program by deconstructing it into manageable parts. Their design shows that they are thinking of the desired task and how this translates into code. Children can identify an error within their program that prevents it following the desired algorithm and then fix it.*	*Children demonstrate the ability to design and code a program that follows a simple sequence. They experiment with timers to achieve repetition effects in their programs. Children are beginning to understand the difference in the effect of using a timer command rather than a repeat command when creating repetition effects.*	*Children's designs for their programs show that they are thinking of the structure of a program in logical, achievable steps and absorbing some new knowledge of coding structures. For example, repetition and use of timers. They make good attempts to 'step through' more complex code in order to identify errors in algorithms and can correct this. e.g. In programs such as Logo, they can 'read' programs with several*	*Children can list a range of ways that the Internet can be used to provide different methods of communication. They can use some of these methods of communication, e.g. being able to open, respond to and attach files to emails using 2Email. They can describe appropriate email conventions when communicating in this way.*	*Children can carry out simple searches to retrieve digital content. They understand that to do this, they are connecting to the internet and using a search engine such as Purple Mash search or internet-wide search engines.*	*Children can collect, analyse, evaluate and present data and information using a selection of software, e.g. using a branching database (2Question), using software such as 2Graph. Children can consider what software is most appropriate for a given task. They can create purposeful content to attach to emails, e.g. 2Respond.*	*Children demonstrate the importance of having a secure password and not sharing this with anyone else. Furthermore, children can explain the negative implications of failure to keep passwords safe and secure. They understand the importance of staying safe and the importance of their conduct when using familiar communication tools such as 2Email in Purple Mash. They know more than one way to report tent*



Computing Progression
N.C. Statements KS2 Year 4

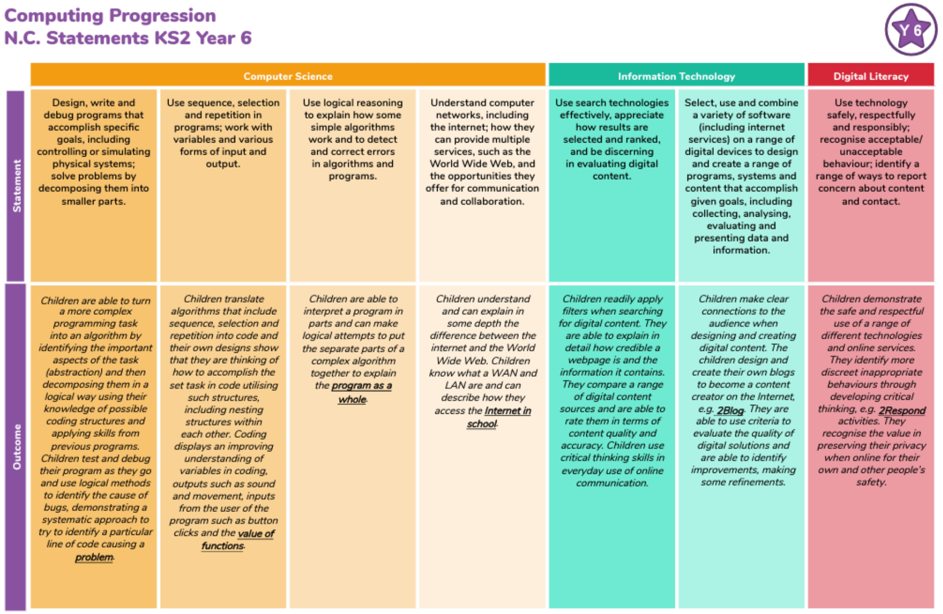
Y 4

	Computer Science				Information Technology		Digital Literacy
Statement	Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.	Use sequence, selection and repetition in programs; work with variables and various forms of input and output.	Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.	Understand computer networks, including the internet; how they can provide multiple services, such as the World Wide Web, and the opportunities they offer for communication and collaboration.	Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content.	Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of	Use technology safely, respectfully and responsibly; recognise acceptable/ unacceptable behaviour; identify a
Outcome	*When turning a real-life situation into an algorithm, the children's design shows that they are thinking of the required task and how to accomplish this in code using coding structures for selection and repetition. Children make more intuitive attempts to debug their own programs.*	*Children's use of timers to achieve repetition effects are becoming more logical and are integrated into their program designs. They understand 'IF statements' for selection and attempt to combine these with other coding structures including variables to achieve the effects that they design in their programs. As well as understanding how variables can be used to store information while a program is executing, they are able to use and manipulate the value of variables. Children can make use of user inputs and outputs such as 'print to screen'. e.g. 2Code.*	*Children's designs for their programs show that they are thinking of the structure of a program in logical, achievable steps and absorbing some new knowledge of coding structures. For example, 'IF' statements, repetition and variables. They can trace code and use step-through methods to identify errors in code and make logical attempts to correct this. In programs such as Logo, they can 'read' programs with several steps and predict the outcome accurately.*	*Children recognise the main component parts of hardware which allow computers to join and form a network. Their ability to understand the online safety implications associated with the ways the internet can be used to provide different methods of communication is improving.*	*Children understand the function, features and layout of a search engine. They can appraise selected webpages for credibility and information at a basic level.*		



Computing Progression
N.C. Statements KS2 Year 5

	Computer Science				Information Technology		
Statement	Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.	Use sequence, selection and repetition in programs; work with variables and various forms of input and output.	Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.	Understand computer networks, including the internet; how they can provide multiple services, such as the World Wide Web, and the opportunities they offer for communication and collaboration.	Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content.	Select, use combine a va software (in internet servi range of digita to design an a range of pr systems and that accom given goals, i collecting, an evaluating presenting d informat	
Outcome	*Children may attempt to turn more complex real-life situations into algorithms for a program by deconstructing it into manageable parts. Children are able to test and debug their programs as they go and can use logical methods to identify the approximate cause of any bug but may need some support identifying the specific line of code.*	*Children can translate algorithms that include sequence, selection and repetition into code with increasing ease and their own designs show that they are thinking of how to accomplish the set task in code utilising such structures. They are combining sequence, selection and repetition with other coding structures to achieve their algorithm design.*	*When children code, they are beginning to think about their code structure in terms of the ability to debug and interpret the code later, e.g. the use of tabs to organise code and the naming of variables*	*Children understand the value of computer networks but are also aware of the main dangers. They recognise what personal information is and can explain how this can be kept safe. Children can select the most appropriate form of online communications contingent on audience and digital content, e.g. 2Blog, 2Email, Display Boards.*	*Children search with greater complexity for digital content when using a search engine. They are able to explain in some detail how credible a webpage is and the information it contains.*	*Children are make appro improvements solutions ba feedback rece can confid comment on th of the soluti creating the program to design brie 2Code. They objectively review solutions from others. Children are able to collaboratively create content and solutions using digital features within software such as collaborative mode. They are able to use several ways of sharing digital content, i.e. 2Blog, Display Boards and 2Email.*	*behaviour to their right to personal privacy and mental wellbeing of themselves and others.*



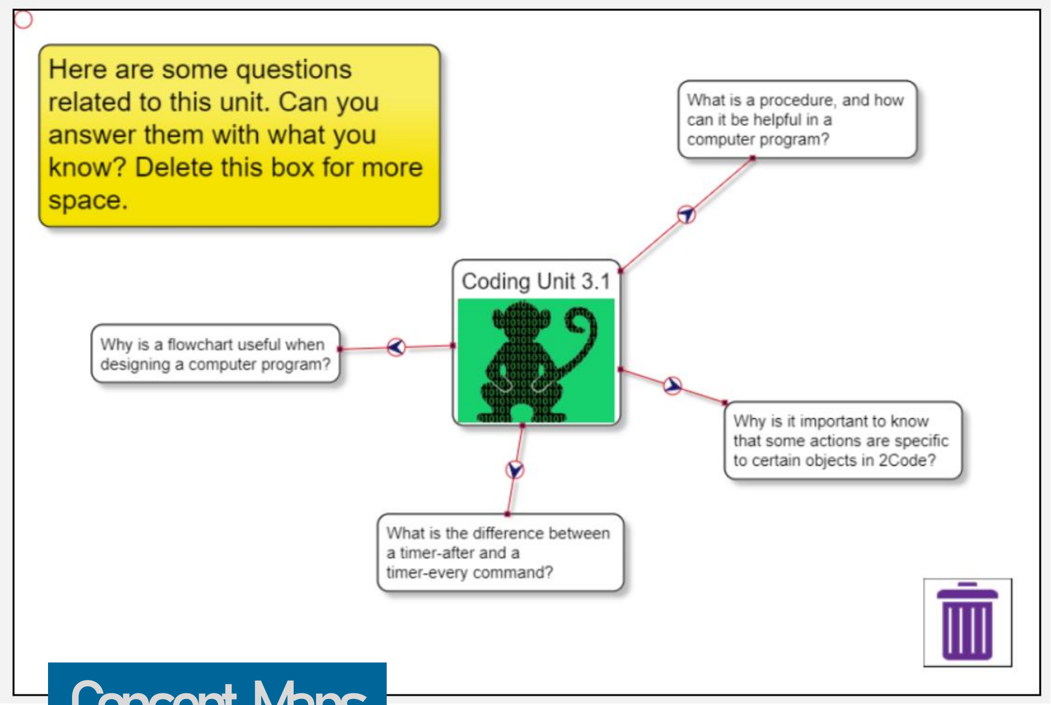
Computing Progression
N.C. Statements KS2 Year 6

Y 6

	Computer Science				Information Technology		Digital Literacy
Statement	Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.	Use sequence, selection and repetition in programs; work with variables and various forms of input and output.	Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.	Understand computer networks, including the internet; how they can provide multiple services, such as the World Wide Web, and the opportunities they offer for communication and collaboration.	Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content.	Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information.	Use technology safely, respectfully and responsibly; recognise acceptable/ unacceptable behaviour; identify a range of ways to report concern about content and contact.
Outcome	*Children are able to turn a more complex programming task into an algorithm by identifying the important aspects of the task (abstraction) and then decomposing them in a logical way using their knowledge of possible coding structures and applying skills from previous programs. Children test and debug their program as they go and use logical methods to identify the cause of bugs, demonstrating a systematic approach to try to identify a particular line of code causing a problem.*	*Children translate algorithms that include sequence, selection and repetition into code and their own designs show that they are thinking of how to accomplish the set task in code utilising such structures, including nesting structures within each other. Coding displays an improving understanding of variables in coding, outputs such as sound and movement, inputs from the user of the program such as button clicks and the value of functions.*	*Children are able to interpret a program in parts and can make logical attempts to put the separate parts of a complex algorithm together to explain the program as a whole.*	*Children understand and can explain in some depth the difference between the internet and the World Wide Web. Children know what a WAN and LAN are and can describe how they access the Internet in school.*	*Children readily apply filters when searching for digital content. They are able to explain in detail how credible a webpage is and the information it contains. They compare a range of digital content sources and are able to rate them in terms of content quality and accuracy. Children use critical thinking skills in everyday use of online communication.*	*Children make clear connections to the audience when designing and creating digital content. The children design and create their own blogs to become a content creator on the Internet, e.g. 2Blog. They are able to use criteria to evaluate the quality of digital solutions and are able to identify improvements, making some refinements.*	*Children demonstrate the safe and respectful use of a range of different technologies and online services. They identify more discreet inappropriate behaviours through developing critical thinking, e.g. 2Respond activities. They recognise the value in preserving their privacy when online for their own and other people's safety.*

Assessment

Concept maps and quizzes can be used in order to assess a child's understanding for each unit and identify gaps for future learning. These are available for every unit and can be set as a 2Do



Here are some questions related to this unit. Can you answer them with what you know? Delete this box for more space.

Coding Unit 3.1

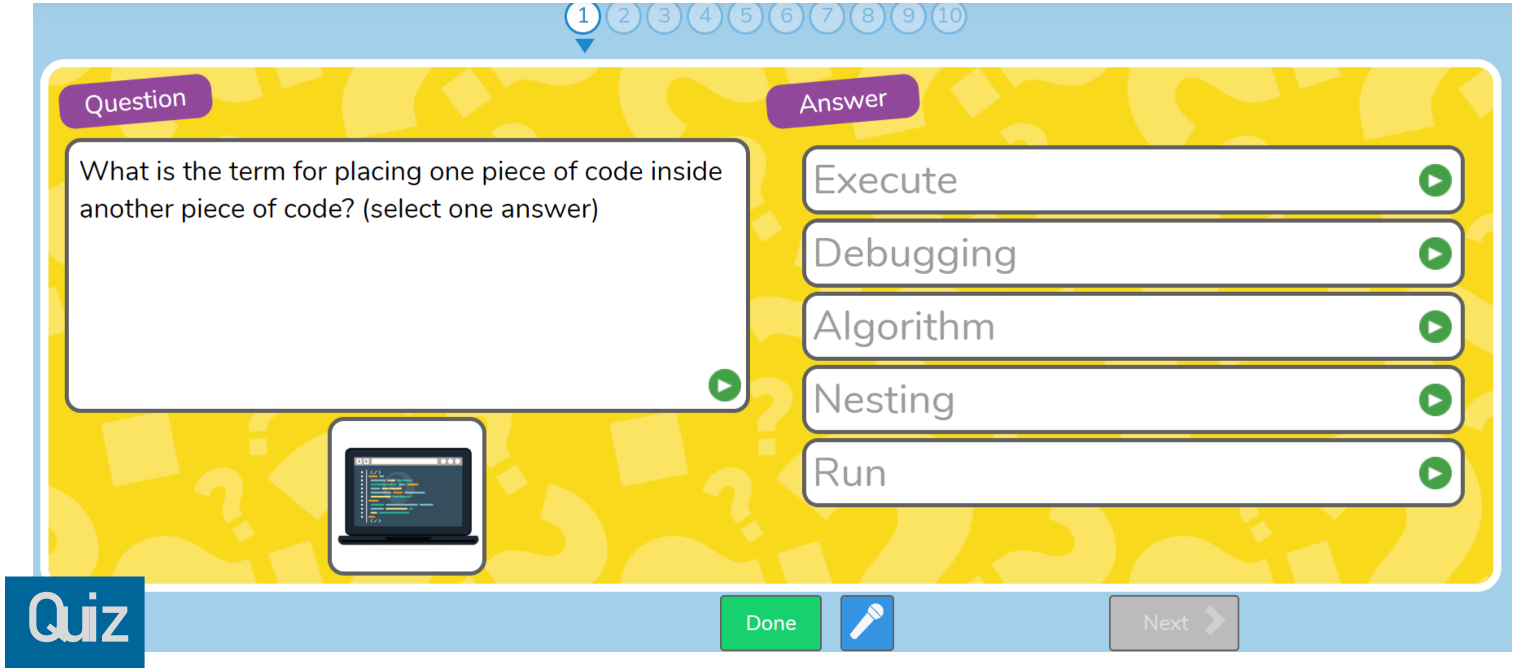
What is a procedure, and how can it be helpful in a computer program?

Why is a flowchart useful when designing a computer program?

Why is it important to know that some actions are specific to certain objects in 2Code?

What is the difference between a timer-after and a timer-every command?

Concept Maps



Question

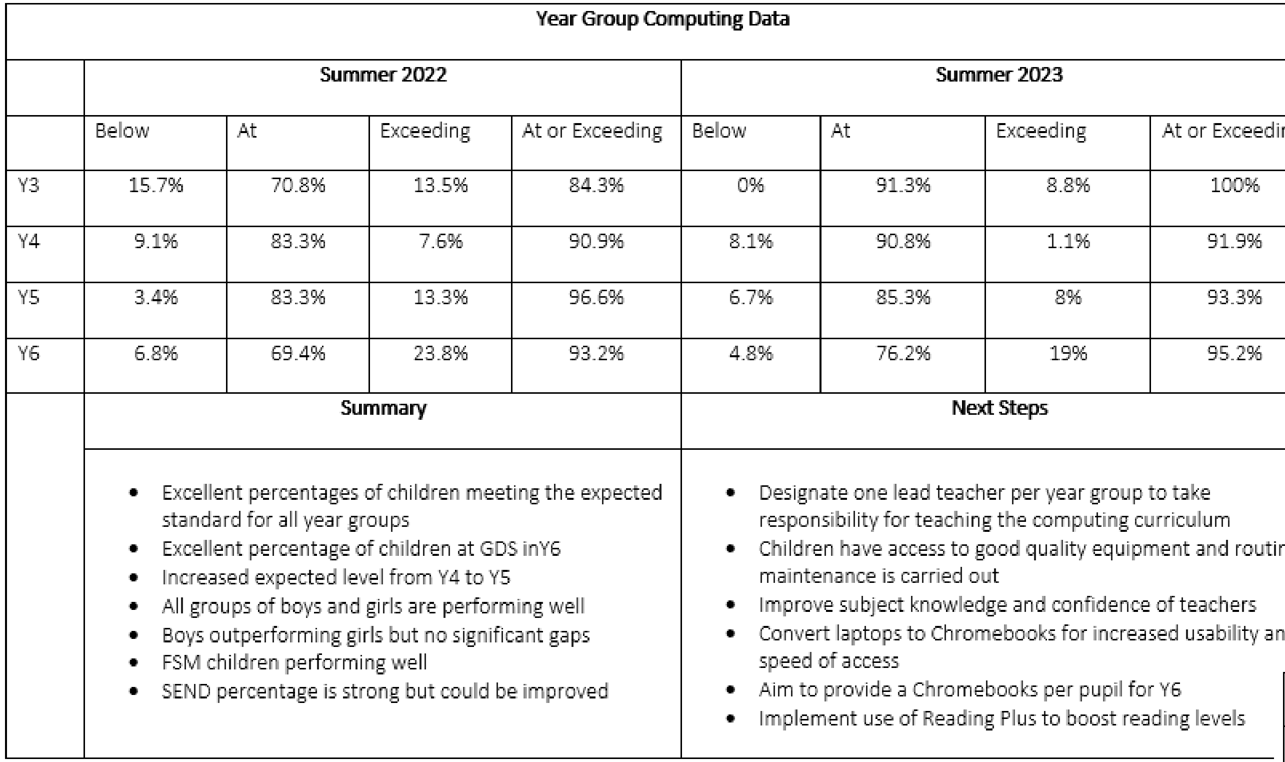
What is the term for placing one piece of code inside another piece of code? (select one answer)

Answer

- Execute
- Debugging
- Algorithm
- Nesting
- Run

Quiz

Current Assessment



Year Group Computing Data

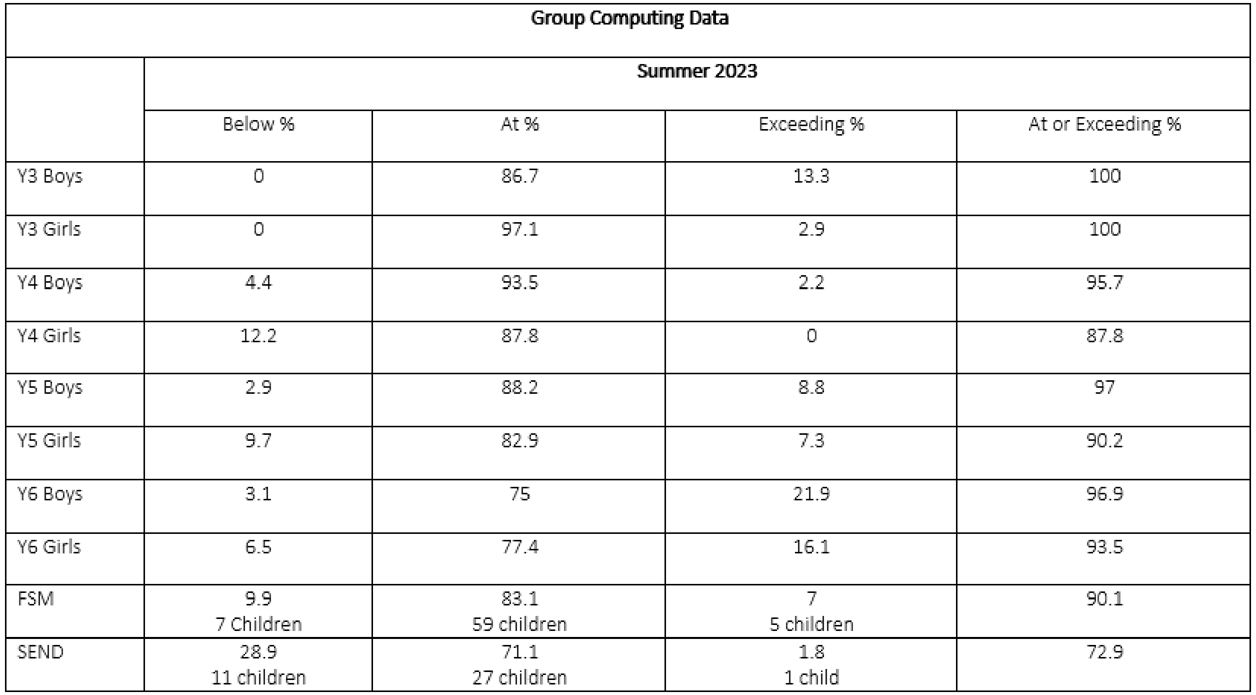
	Summer 2022				Summer 2023			
	Below	At	Exceeding	At or Exceeding	Below	At	Exceeding	At or Exceeding
Y3	15.7%	70.8%	13.5%	84.3%	0%	91.3%	8.8%	100%
Y4	9.1%	83.3%	7.6%	90.9%	8.1%	90.8%	1.1%	91.9%
Y5	3.4%	83.3%	13.3%	96.6%	6.7%	85.3%	8%	93.3%
Y6	6.8%	69.4%	23.8%	93.2%	4.8%	76.2%	19%	95.2%

Summary

- Excellent percentages of children meeting the expected standard for all year groups
- Excellent percentage of children at GDS inY6
- Increased expected level from Y4 to Y5
- All groups of boys and girls are performing well
- Boys outperforming girls but no significant gaps
- FSM children performing well
- SEND percentage is strong but could be improved

Next Steps

- Designate one lead teacher per year group to take responsibility for teaching the computing curriculum
- Children have access to good quality equipment and routine maintenance is carried out
- Improve subject knowledge and confidence of teachers
- Convert laptops to Chromebooks for increased usability and speed of access
- Aim to provide a Chromebooks per pupil for Y6
- Implement use of Reading Plus to boost reading levels

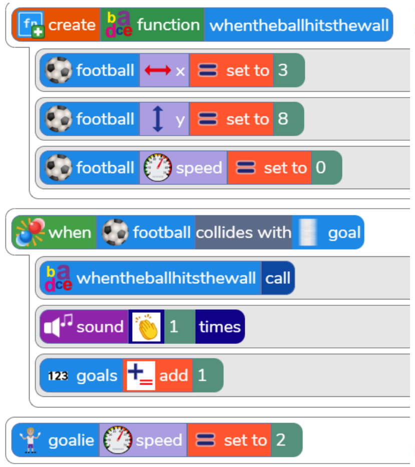


Group Computing Data

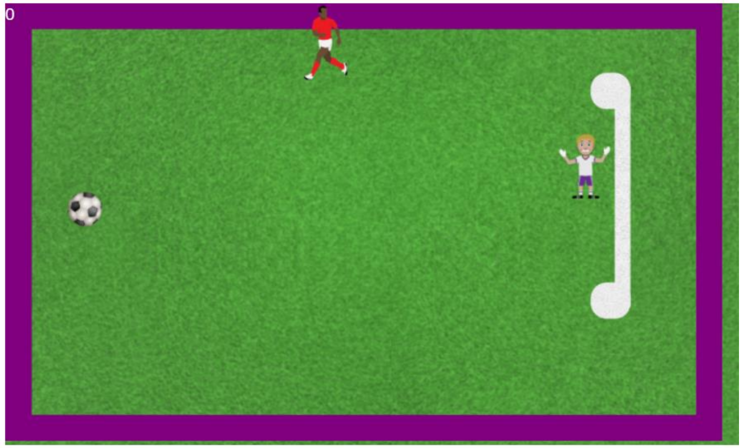
	Summer 2023			
	Below %	At %	Exceeding %	At or Exceeding %
Y3 Boys	0	86.7	13.3	100
Y3 Girls	0	97.1	2.9	100
Y4 Boys	4.4	93.5	2.2	95.7
Y4 Girls	12.2	87.8	0	87.8
Y5 Boys	2.9	88.2	8.8	97
Y5 Girls	9.7	82.9	7.3	90.2
Y6 Boys	3.1	75	21.9	96.9
Y6 Girls	6.5	77.4	16.1	93.5
FSM	9.9 7 Children	83.1 59 children	7 5 children	90.1
SEND	28.9 11 children	71.1 27 children	1.8 1 child	72.9

Work Showcase

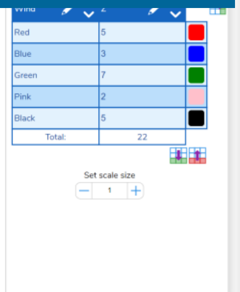
Coding and game creating



```
create function whentheballhitsthewall
    football ↔ x = set to 3
    football ↕ y = set to 8
    football speed = set to 0
when football collides with goal
    whentheballhitsthewall call
    sound 1 times
    goals + add 1
goalie speed = set to 2
```



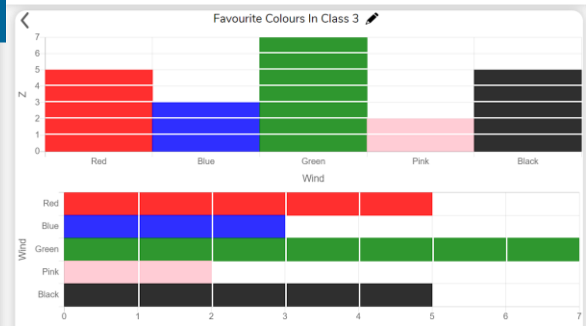
Data collection and presentation



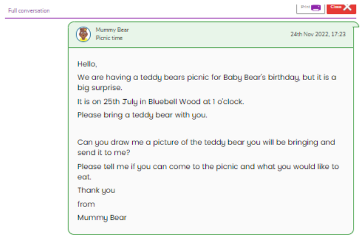
Wind	Z	
Red	5	🟥
Blue	3	🟦
Green	7	🟩
Pink	2	▨
Black	5	⬛
Total:	22	

Set scale size

− 1 +



Simulated Email threads



Mummy Bear
Picnic time
24th Nov 2022, 17:23

Hello,

We are having a teddy bears picnic for Baby Bear's birthday, but it is a big surprise.

It is on 25th July in Bluebell Wood at 1 o'clock.

Please bring a teddy bear with you.

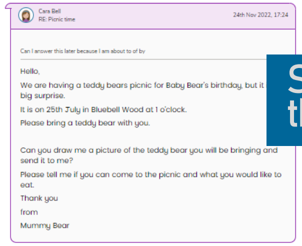
Can you draw me a picture of the teddy bear you will be bringing and send it to me?

Please tell me if you can come to the picnic and what you would like to eat.

Thank you

from

Mummy Bear



Cara Bell
RE: Picnic time
24th Nov 2022, 17:24

Can I answer this later because I am about to of by

Hello,

We are having a teddy bears picnic for Baby Bear's birthday, but it big surprise.

It is on 25th July in Bluebell Wood at 1 o'clock.

Please bring a teddy bear with you.

Can you draw me a picture of the teddy bear you will be bringing and send it to me?

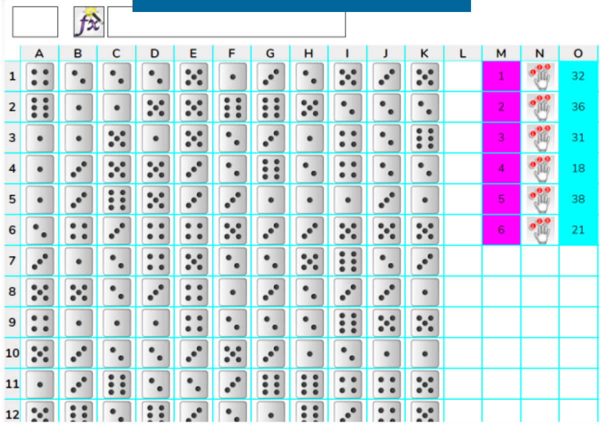
Please tell me if you can come to the picnic and what you would like to eat.

Thank you

from

Mummy Bear

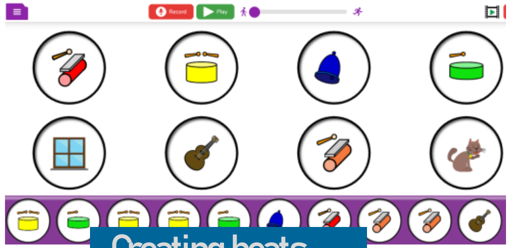
Spreadsheets and simulations



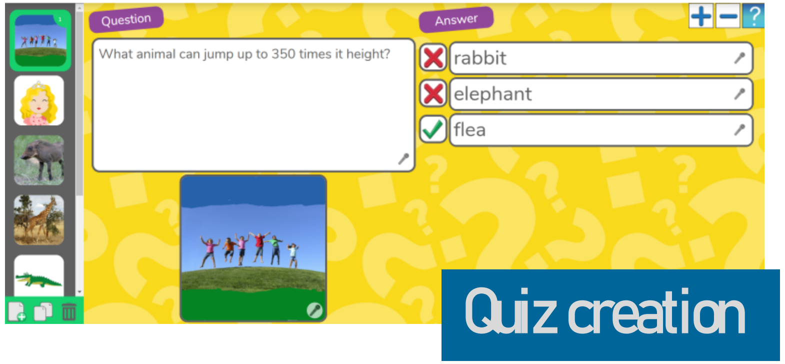
	M	O
1	1	32
2	2	36
3	3	31
4	4	18
5	5	38
6	6	21

Name	Number of teddies		Number of cars		Number of toys
Grace	15	+	12	=	27
Finlay	17	+	25	=	42
Hadas	6	+	13	=	19
Mia	10	+	4	=	14
Kelly	12	+	12	=	24

Creating beats and melodies



Quiz creation



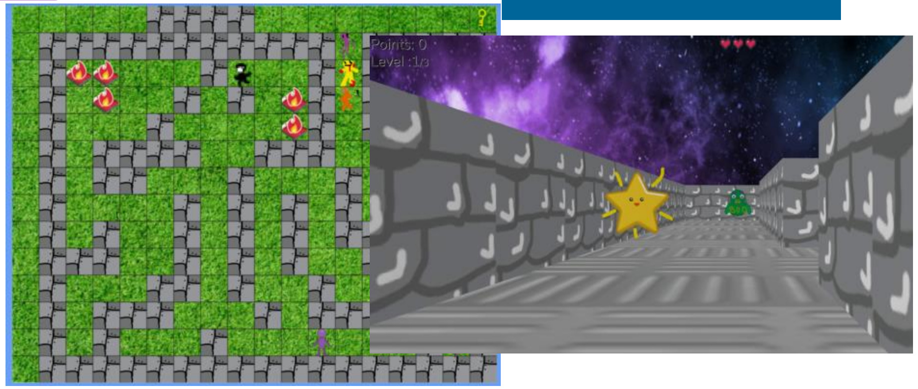
What animal can jump up to 350 times it height?

☒ rabbit

☒ elephant

☑ flea

3D game design


Points: 0
Level :1a