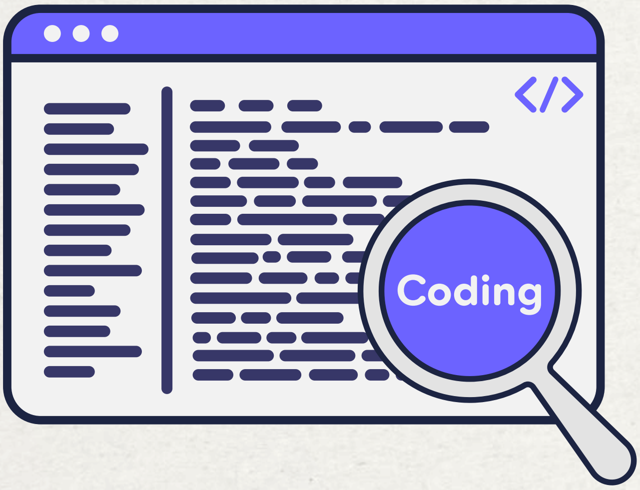
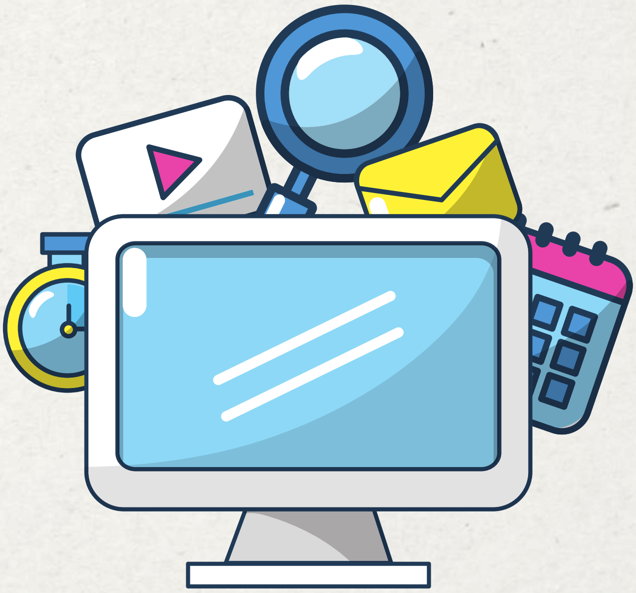
HARPER BELL
ADVENTIST
SCHOOL

COMPUTING





HARPER BELL ADVENTIST SCHOOL

COMPUTING ON A PAGE

NATIONAL REQUIREMENTS

- By the end of primary school, pupils should:
- Understand and apply principles of computer science (abstraction, logic, algorithms, data representation).
- Analyse problems computationally and gain practical experience writing programs.
- Evaluate and apply IT to solve problems.
- Be digitally literate for future work and active participation in a digital world

RESOURCES

- Teach Computing Curriculum
- The Computing Curriculum (Raspberry Pi Foundation)
- Barefoot Computing
- Computing at School (CAS)
- STEM Learning

WHERE IS OUR CURRICULUM NOW?

- Computing has not been taught for a number of years
- IT resources are poor within the school – all teaching must be completed from a limited number of ipads
- The DFE CTC is funding an upgrade school Wi-Fi and network infrastructure so pupils have fast, secure and reliable connectivity for digital learning
- IT support for the school has been changed to Bishop's Challenor and an action plan has been created for compliance following a thorough external review.

TRENDS IN PRIMARY COMPUTING

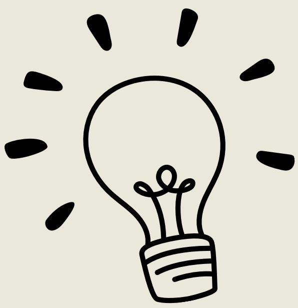
1. Greater Emphasis on Physical Computing
Tools like micro:bit, Crumble, and MaKey MaKey are increasingly used to make programming tangible and engaging for younger learners.
These hands-on experiences help pupils connect coding with real-world applications.
2. Focus on Inclusivity and Accessibility
Universal Design for Learning (UDL) principles are being applied to computing lessons to ensure all pupils, including those with SEND, can access and engage with content.
Strategies include multimodal resources, assistive technologies, and differentiated tasks.
3. Online Safety and Digital Ethics
Online safety is now a whole-school priority, integrated into computing lessons from KS1 onward.
Teaching digital etiquette and respect for digital creations is becoming standard practice.
4. Computational Thinking Beyond Coding
Schools are embedding problem-solving, logic, and abstraction into broader subjects, not just programming.
Gamified learning (e.g., Robot Turtles, Scratch-based games) is popular for teaching these skills.
5. Addressing Gender Imbalance
Initiatives to empower girls in computing are gaining traction, with targeted CPD and classroom strategies to close the gender gap.
6. Integration of Emerging Technologies
There is growing interest in introducing AI concepts and critical thinking about technology use, though this is still underdeveloped in many schools.
7. Curriculum Personalisation & Teacher Autonomy
While off-the-shelf schemes (like NCCE units) are widely used, there's a push for more creative, flexible approaches to avoid overly structured lessons.
8. Digital Infrastructure & Equity
Government investment aims to close the digital divide, ensuring all schools meet core digital standards (Wi-Fi, security, connectivity) by 2030.

HOW DO WE GET THE BIG PICTURE?

- Alongside the National Curriculum, the underlying pillars of our Computing Curriculum are taken from the Ofsted Research review.
- Early computing experiences are crucial
- Time in school for computing is a challenge and the school's technology needs to improve in quality and quantity
- Progression covers 3 pillars – Computer Science, Information Technology and Digital Literacy.
- Declarative (knowing that) and Procedural (knowing how) knowledge need to be sequenced and connected
- Our curriculum includes breadth in the 3 pillars, building skills gradually through sequence and connection.

SCHOOL PRIORITIES

- Clear sequences of learning in all wider curriculum subjects
- Ensure key knowledge is clear within lesson planning
- Regular check ins on how well pupils remember their learning and filling any gaps swiftly
- To ensure pupils with additional needs are supported



Computing - Aims and Intent

Aims
At Harper Bell SDA School, our intent is to provide every pupil with the knowledge, skills, and confidence to thrive in a digital world. We aim to develop computational thinking, problem-solving, and creativity through a progressive curriculum that builds from early years to upper key stages. Despite limited resources, we prioritise core computing concepts—such as algorithms, programming, and digital safety—using a blend of practical and unplugged activities. Our goal is to ensure pupils become responsible, competent, and innovative users of technology, prepared for future learning and employment.

Implementation

Our computing curriculum is designed to empower pupils to become confident, creative, and responsible users of technology. We follow the Raspberry Pi Foundation's Computing Curriculum, which is research-informed and structured around three strands: Computer Science, Information Technology, and Digital Literacy. Our intent is to develop computational thinking, problem-solving, and creativity through progressive units from EYFS to KS2. We prioritise online safety and inclusivity, ensuring all pupils can access high-quality computing education regardless of prior experience or resources.

Key Features of Intent:

- Based on national curriculum and Raspberry Pi progression framework.
- Ambitious yet realistic for a low baseline.
- Emphasises digital safety, creativity, and problem-solving.
- Builds logical progression across year groups.
- We use Raspberry Pi Foundation resources to deliver structured, engaging lessons:
- Curriculum Structure
- KS1: Unplugged activities, simple algorithms, introduction to Scratch.
- KS2: Programming with Scratch, data handling, creating media, and understanding networks.
- Cross-curricular links: Use computing in science (data logging), maths (logical reasoning), and art (digital media).

Teaching Strategies

- Unplugged activities for algorithmic thinking when devices are limited.
- Free tools: Scratch, BBC Micro:bit simulator, Raspberry Pi projects.
- Differentiation: Scaffolded tasks and extension challenges.
- CPD: Staff training via Raspberry Pi Foundation's free courses and Hello World magazine.

The National Curriculum

A high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world. It has strong links with mathematics, science, and design technology. The curriculum ensures pupils:

- Understand and apply principles of computer science (abstraction, logic, algorithms, data representation).
- Gain practical experience in programming to solve problems.
- Evaluate and apply information technology effectively.
- Become digitally literate, confident, and responsible users of technology

Three Pillars of Progression underpin the Computing curriculum:

1. Computer Science
    - Core theoretical knowledge: how computers and computation work, including algorithms, data, and programming concepts.
    - Provides the foundation for all other computing learning.
2. Information Technology (IT)
    - Practical application: using technology to create digital artefacts and solve real-world problems.
    - Includes understanding systems, networks, and digital tools in context.
3. Digital Literacy
    - Safe, responsible, and effective use of technology.
    - Enables pupils to communicate, collaborate, and participate in a digital society

Two Types of Knowledge are key:
Declarative Knowledge ("Knowing That")

- Facts and concepts about computing, e.g., what an algorithm is, how binary works, principles of networks.

Procedural Knowledge ("Knowing How")

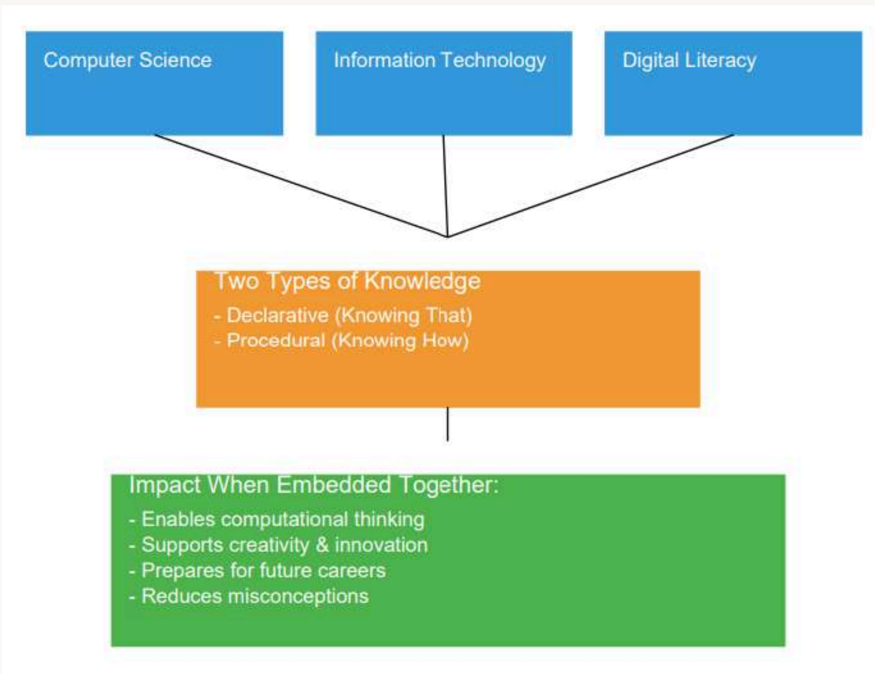
- Skills and processes, e.g., writing code, debugging, using software tools.

Why Both Matter:
When declarative and procedural knowledge are sequenced and connected, pupils can:

- Move from theory to practice (apply concepts in real tasks).
- Develop mental models for problem-solving.
- Progress from simple tasks (predicting program behaviour) to complex projects (designing modular systems).

Impact of Embedding Both Together

- Builds computational thinking: breaking problems into logical steps.
- Supports creativity and innovation: designing systems and digital artefacts.
- Prepares pupils for future careers and active participation in a digital world.
- Reduces misconceptions (e.g., writing code without understanding its purpose).



National Curriculum Content
Key stage 1
Pupils should be taught:
 understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions
 create and debug simple programs
 use logical reasoning to predict the behaviour of simple programs
 use technology purposefully to create, organise, store, manipulate and retrieve digital content
 recognise common uses of information technology beyond school
 use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.

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.

Pedagogy

Computing is a broad discipline, and computing teachers require a range of strategies to deliver effective lessons to their learners. Our pedagogical approach consists of 12 key principles underpinned by research: each principle has been shown to contribute to effective teaching and learning in computing.

It is recommended that computing teachers use their professional judgement to review, select, and apply relevant strategies for their learners.

These 12 principles are embodied by The Computing Curriculum, and you can find examples of their application throughout the units of work at every key stage. Beyond delivering these units, you can learn more about these principles and related strategies in *The Big Book of Computing Pedagogy* we have collated (the-cc.io/pedagogy).

Lead with concepts
Support learners in the acquisition of knowledge, through the use of key concepts, terms, and vocabulary, providing opportunities to build a shared and consistent understanding. Glossaries, concept maps (the-cc.io/qr07), and displays, along with regular recall and revision, can support this approach.

Work together
Encourage collaboration, specifically using pair programming (the-cc.io/qr03) and peer instruction (the-cc.io/qr04), and also structured group tasks. Working together stimulates classroom dialogue, articulation of concepts, and development of shared understanding.

Get hands-on
Use physical computing and making activities that offer tactile and sensory experiences to enhance learning. Combining electronics and programming with arts and crafts (especially through exploratory projects) provides learners with a creative, engaging context to explore and apply computing concepts.

Unplug, unpack, repack
Teach new concepts by first unpacking complex terms and ideas, exploring these ideas in unplugged and familiar contexts, then repacking this new understanding into the original concept. This approach, called 'semantic waves' (the-cc.io/qr06), can help learners develop a secure understanding of complex concepts.

Model everything
Model processes or practices — everything from debugging code to binary number conversions — using techniques such as worked examples (the-cc.io/qr02) and live coding (the-cc.io/qr05). Modelling is particularly beneficial to novices, providing scaffolding that can be gradually taken away.

Foster program comprehension
Use a variety of activities to consolidate knowledge and understanding of the function and structure of programs (the-cc.io/qr12), including debugging, tracing, and Parson's Problems. Regular comprehension activities will help secure understanding and build connections with new knowledge.

Create projects
Use project-based learning activities to provide learners with the opportunity to apply and consolidate their knowledge and understanding. Design is an important, often overlooked aspect of computing. Learners can consider how to develop an artefact for a particular user or function, and evaluate it against a set of criteria.

Add variety
Provide activities with different levels of direction, scaffolding, and support that promote learning, ranging from highly structured to more exploratory tasks. Adapting your instruction to suit different objectives will help keep all learners engaged and encourage greater independence.

Challenge misconceptions
Use formative questioning to uncover misconceptions and adapt teaching to address them as they occur. Awareness of common misconceptions alongside discussion, concept mapping, peer instruction, or simple quizzes can help identify areas of confusion.

Make concrete
Bring abstract concepts to life with real-world, contextual examples, and a focus on interdependencies with other curriculum subjects. This can be achieved through the use of unplugged activities, proposing analogies, storytelling around concepts, and finding examples of the concepts in pupils' lives.

Structure lessons
Use supportive frameworks when planning lessons, such as PRIMM (Predict, Run, Investigate, Modify, Make – the-cc.io/qr11) and UMC (Use-Modify-Create). These frameworks are based on research and ensure that differentiation can be built in at various stages of the lesson.

Read and explore code first
When teaching programming, focus first on code 'reading' activities, before code writing. With both block-based and text-based programming, encourage pupils to review and interpret blocks of code. Research has shown that being able to read, trace, and explain code augments pupils' ability to write code.

Early Years

The EYFS does not treat computing as a separate subject; instead, it embeds technology and computational thinking across seven areas of learning, particularly:

- Understanding the World – exploring how things work, using technology in everyday contexts.
- Expressive Arts and Design – using digital tools creatively (e.g., drawing apps, photography).
- Personal, Social and Emotional Development – resilience, independence, and sensible screen time.
- Physical Development – fine motor skills for using devices and tools confidently.

Examples of in practice:

- Children use programmable toys (e.g., Bee-Bots) to develop sequencing and directional language.
- Access to age-appropriate software for drawing, sorting, and problem-solving.
- Incorporating technology into role play (e.g., phones, cameras) to make learning meaningful.
- Using interactive whiteboards and apps like Purple Mash for creative and exploratory tasks.

Characteristics of Effective Learning

EYFS promotes:

- Playing and exploring – experimenting with digital tools and devices.
- Active learning – persisting with challenges (e.g., completing a simple program).
- Creating and thinking critically – making links, solving problems, and developing strategies

Research Review Insights

Ofsted emphasizes:

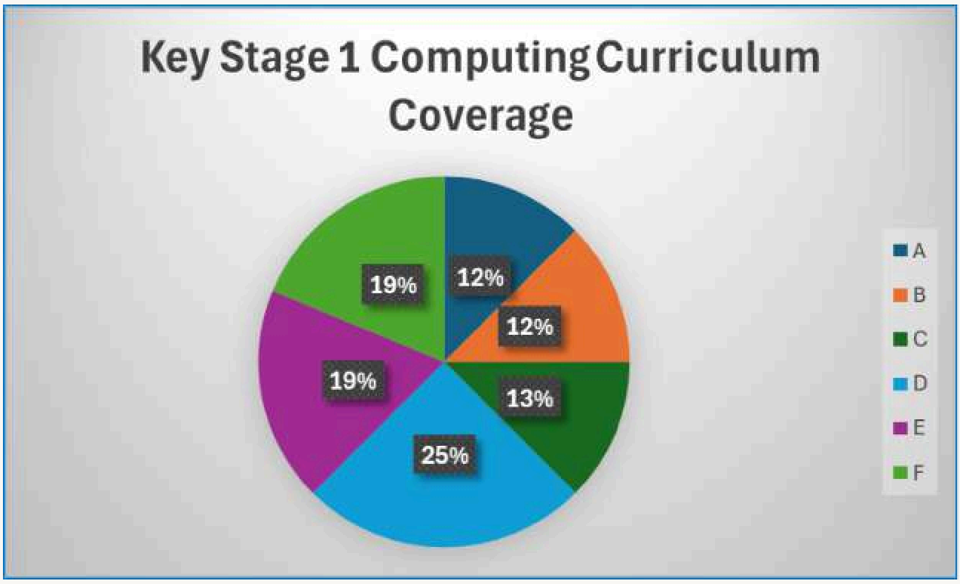
- Early exposure matters: Young children can grasp core computing concepts like algorithms and problem-solving when taught through play and exploration.
- Vocabulary development: Introduce terms like sequence, pattern, algorithm in everyday contexts.
- Cross-domain integration: Computing should not be siloed; link it with maths (patterns, sequencing), literacy (story sequencing), and physical development (fine motor control for devices).
- Knowledge building: Even in EYFS, start developing declarative knowledge (knowing what technology is and how it works) and procedural knowledge (knowing how to use it). These underpin later computational thinking.

When embedded effectively:

- Children develop problem-solving and logical reasoning early.
- They gain confidence with technology, preparing for KS1 computing.
- They learn safe and purposeful use of devices, supporting digital literacy.
- Builds a foundation for computational thinking, which is essential for progression.

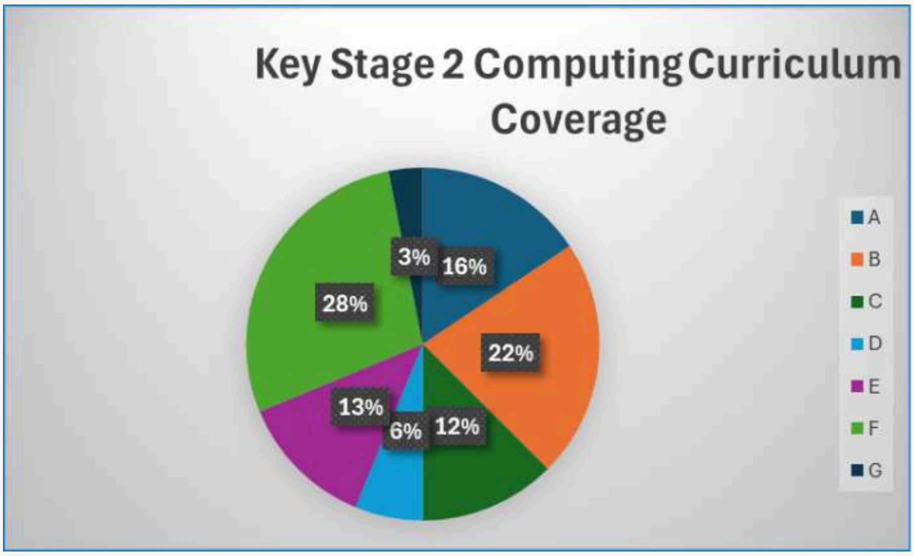
The National Curriculum Coverage and Compliance Key Stage 1

A- understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions
B- create and debug simple programs
C- use logical reasoning to predict the behaviour of simple programs
D- use technology purposefully to create, organise, store, manipulate and retrieve digital content
E- recognise common uses of information technology beyond school and IT
F- use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies



The National Curriculum Coverage and Compliance Key Stage 2

A - design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
B - use sequence, selection, and repetition in programs; work with variables and various forms of input and output
C - use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
D - 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
E - use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content
F - 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
G - use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact



How we support the learning of Computing

At Harper Bell SDA School, our Computing curriculum adopts a spiral model; this is where key knowledge and skills are revisited and built on at a level appropriate to children's development.

A spiral curriculum enables teachers to introduce complex topics to children in a way that is both accessible and engaging. This method builds a strong pedagogical foundation, which can be developed further as children progress through primary school. Children have multiple opportunities to master new skills and gain confidence in their abilities as they revisit and build upon previously taught concepts. Our 3 key areas rotate termly:

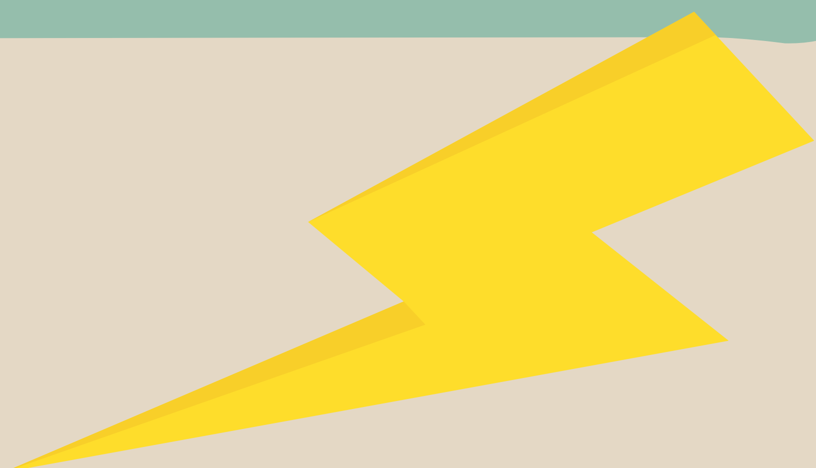
Autumn - Computer Systems and Networks or Data Handling

Spring - Programming

Summer - Creating Media

The use of Flashbacks

Research indicates that students must review information regularly to move it into long-term memory and form connections between ideas. At Harper Bell SDA, we are developing the consistent use of Flashbacks at the start of lessons. The Flashbacks provide an opportunity for children to revisit key learning from last lesson, last term, last year or last Key Stage. Teachers use this as an opportunity to check retention of knowledge and skills and select an appropriate action to close any gaps emerging.



OVERVIEW OF THEMES

Year 1

Computer Systems and Networks: Technology around us - cover Y1 and y2 (12 lessons)	Programming B: Progamming animations	Creating Media: Digital writing

Year 2

Data and Information: Pictograms	Programming B: Programming Quizzes	Creating Media: Digital Music

Year 3

Computer Systems and Networks: Connecting Computers Data and Information: Databases	Programming: Sequencing Sounds	Creating Media: Stop Frame Animation

Year 4

Data and Information: Data Logging	Programming: Repetition in Games	Creating Media: Audio Production

OVERVIEW OF THEMES

Year 5

Computer Systems and Networks: The Internet (y4) and Systems and Searching (12 lessons)	Programming: Selection in Physical Computing	Creating Media: Video Production

Year 6

Data and Information: Flat File Databases (y5) and Spreadsheets	Programming: Variables in Games	Creating Media: 3D Modelling

Computing Narrative: Spiral Approach from EYFS to Year 6
Our computing curriculum follows a spiral structure, ensuring that concepts introduced in EYFS are revisited and deepened through KS1 and KS2. This approach builds control, proficiency, and confidence over time.

EYFS Foundation

- Focus: Exploration through play (e.g., programmable toys, simple apps).
- Declarative Knowledge: Understanding what technology is and its purpose.
- Procedural Knowledge: Using devices to take photos, move a robot, or draw digitally.
- Goal: Develop curiosity and basic vocabulary (e.g., “algorithm,” “sequence”).

KS1 (Years 1–2)

- Autumn: Computer Systems & Networks – understanding technology around us.
- Spring: Programming – creating and debugging simple programs (Scratch Jr).
- Summer: Creating Media – digital painting and photography.
- Declarative: What algorithms are, how programs follow instructions.
- Procedural: Writing simple code, using tools to create digital content.
- Progression: Move from recognition to purposeful use of technology.

KS2 (Years 3–6)

- Year 3–4: Networks, data logging, stop-motion animation, repetition in code.
- Year 5–6: Search engines, variables, 3D modelling, video editing.
- Declarative: Understanding networks, data structures, programming concepts (sequence, selection, repetition, variables).
- Procedural: Designing, writing, and debugging increasingly complex programs; creating multimedia projects.
- Goal: Combine knowledge types to enable computational thinking and creativity.

ASSESSMENT

Two Main Types of Assessment in Computing

1. Formative Assessment
    - Occurs before and during learning.
    - Raspberry Pi recommends:
        - Flashbacks / Retrieval Practice: Revisiting prior concepts at the start of lessons (e.g., "What is an algorithm?" or "How do we debug a program?").
        - Live Questioning & Feedback: Teachers use probing questions during programming tasks and adapt instruction based on misconceptions.
        - Peer Discussion: Pupils explain their code or reasoning to others, reinforcing understanding.
        - Observation of Practical Tasks: Watching pupils use Scratch or micro:bits to identify gaps in procedural knowledge.
2. Summative Assessment
    - Occurs at the end of a unit or project.
    - Raspberry Pi uses:
        - Unit Markers / End-of-Unit Quizzes: Short, focused questions to check declarative knowledge (e.g., "What does a variable do?").
        - Project Outcomes: Completed programs or digital artefacts demonstrate procedural knowledge.
        - Teacher Judgement: Based on evidence from coding tasks and media projects.

Flashbacks in Computing:

Computing flashbacks revisit:

- Key vocabulary (algorithm, sequence, variable).
- Prior skills (e.g., debugging, using loops).
- Previous tools (Scratch Jr → Scratch → micro:bits).

Digital Portfolios:

- Pupils maintain:
    - Coding Journals: Screenshots of code, reflections on debugging.
    - Media Logs: Notes on design choices for animations, videos, or 3D models.
    - Self-Assessment: Pupils annotate what worked and what they would improve.

Unit Markers:

Computing unit markers include:

- Knowledge Checks: Multiple-choice or short-answer questions on concepts.
- Expected Outcomes: Example programs or media artefacts for comparison.

Declarative vs Procedural Assessment

- Declarative Knowledge:
- Assessed through quizzes, oral questioning, and flashbacks.
- Procedural Knowledge:
- Evident in projects, coding tasks, and portfolio entries.

KEY STAGE 1 DETAILED OVERVIEW

Year Group	Suggested Ord er	Unit Name	Lesson	Learning Objectives	Success Criteria	National Curriculum Links						RPF Computing Taxonomy										Cross Curricular Links	Education for a Connected World
						1.1	1.2	1.3	1.4	1.5	1.6	AL	CM	CS	DD	DI	ET	IT	NW	PG	SS		
1	1	Computing systems and networks – Technology around us	1	-To identify technology	-I can explain howthese technology examples help us - I can explain technology as something that helps us - I can locate examples of technology in the c lassro o m -I can name the main parts of a computer - I can switch on and log into a computer																		- Copyright and ownership - Health, well-being and lifestyle
1	1	Computing systems and networks – Technology around	2	-To identify a computer and its main parts	- I can use a mouse to click and drag -I can click and drag to make objects on a screen - I can use a mouse to create a picture																		- Copyright and ownership - Health, well-being and lifestyle
1	1	Computing systems and networks – Technology around	3	-To use a mouse in different ways	- I can use a mouse to open a program -I can save my work to a file - I can say what a keyboard is for																		- Copyright and ownership - Health, well-being and lifestyle
1	1	Computing systems and networks – Technology around	4	-To use a keyboard to type on a computer	- I can type my name on a computer -I can delete letters - I can open my work from a file																		- Copyright and ownership - Health, well-being and lifestyle
1	1	Computing systems and networks – Technology around	5	-To use the keyboard to edit text	- I can use the arrow keys to move the cursor -I can describe some uses of computers - I can identify examples of computers - I can identify that a computer is a part of IT																		- Copyright and ownership - Health, well-being and lifestyle
2	1	Computing systems and networks – IT	1	-To recognise the uses and features of information technology	-I can identify examples of IT - I can identify that some IT can be used in more than one way - I can sort school IT by what it's used for																		- Health, well-being and lifestyle
2	1	Computing systems and networks – IT around us	2	-To identify the uses of information technology in the school	-I can find examples of information technology - I can sort IT by where it is found - I can talk about uses of information technology																		- Health, well-being and lifestyle
2	1	Computing systems and networks – IT	3	-To identify information technology beyond school	-I can demonstrate how IT devices work together - I can recognise common types of technology - I can say why we use IT																		- Health, well-being and lifestyle
2	1	Computing systems and networks – IT	4	-To explain how information technology helps us	-I can list different uses of information technology - I can say how rules can help keep me safe - I can talk about different rules for using IT																		- Health, well-being and lifestyle
2	1	Computing systems and networks – IT	5	-To explain how to use information technology safely	-I can explain the need to use IT in different ways - I can identify the choices that I make when using IT - I can use IT for different types of activities																		- Health, well-being and lifestyle
2	1	Computing systems and networks – IT around us	6	-To recognise that choices are made when using information technology	-I can discuss how we benefit from these rules - I can give examples of some of these rules - I can identify rules to keep us safe and healthy when we are using technology in and beyond the home																		- Health, well-being and lifestyle
1	1	Computing systems and networks – Technology around us	6	-To create rules for using technology responsibly	-I can compare different programming tools - I can find which commands to move a sprite - I can use commands to move a sprite -I can run my program																		- Copyright and ownership - Health, well-being and lifestyle
1	6	Programming B - Programming	1	-To choose a command for a given purpose	- I can use a Start block in a program - I can use more than one block by joining them together -I can change the value																		
1	6	Programming B - Programming animations	2	-To show that a series of commands can be joined together	- I can find blocks that have numbers - I can say what happens when I change a value -I can add blocks to each of my sprites																		
1	6	Programming B - Programming	3	-To identify the effect of changing a value	- I can delete a sprite - I can show that a project can include more than one sprite -I can choose appropriate artwork for my project																		
1	6	Programming B - Programming animations	4	-To explain that each sprite has its own instructions	- I can create an algorithm for each sprite - I can decide how each sprite will move -I can add programming blocks based on my algorithm																		
1	6	Programming B - Programming	5	-To design the parts of a project	- I can test the programs I have created - I can use sprites that match my design -I can identify and find keys on a keyboard																		
1	6	Programming B - Programming animations	6	-To use my algorithm to create a program	- I can open a word processor - I can recognise keys on a keyboard -I can enter text into a computer - I can use backspace to remove text - I can use letter, number, and space keys																		
1	5	Creating media – Digital writing	1	-To use a computer to write	-I can explain what the keys that I have learnt about already do																		- Privacy and security
1	5	Creating media – Digital writing	2	-To add and remove text on a computer	- I can identify the toolbar and use bold, italic, and und e rline - I can type capital letters -I can change the font																		- Privacy and security
1	5	Creating media – Digital writing	3	-To identify that the look of text can be changed on a computer	- I can select all of the text by clicking and dragging - I can select a word by double-clicking -I can decide if my changes have improved my wr iting - I can say what tool I used to change the text																		- Privacy and security
1	5	Creating media – Digital writing	4	-To make careful choices when changing text	- I can use 'undo' to remove changes -I can explain the differences between typing and wr iting - I can make changes to text on a computer																		- Privacy and security
1	5	Creating media – Digital writing	5	-To explain why I used the tools that I chose	- I can say why I prefer typing or writing -I can compare totals in a tally chart - I can record data in a tally chart - I can represent a tally count as a total																		- Privacy and security
1	5	Creating media – Digital writing	6	-To compare typing on a computer to writing on paper	-I can enter data onto a computer - I can use a computer to view data in a different format - I can use pictograms to answer simple questions about objects																		- Privacy and security
2	4	Data and information – Pictograms	1	-To recognise that we can count and compare objects using tally charts	-I can explain what the pictogram shows - I can organise data in a tally chart - I can use a tally chart to create a pictogram																	Maths	- Privacy and security
2	4	Data and information – Pictograms	2	-To recognise that objects can be represented as p ic ture s	-I can answer 'more than'/'less than' and 'most/least' questions about an attribute - I can create a pictogram to arrange objects by an a ttr ib ute - I can tally objects using a common attribute																	Maths	- Privacy and security
2	4	Data and information – Pictograms	3	-To create a pictogram	-I can choose a suitable attribute to compare people - I can collect the data I need - I can create a pictogram and draw conclusions from it																	Maths	- Privacy and security
2	4	Data and information – Pictograms	4	-To select objects by attribute and make c o mp ariso ns	-I can give simple examples of why information should not be shared - I can share what I have found out using a computer																	Maths	- Privacy and security
2	4	Data and information – Pictograms	5	-To recognise that people can be described by a ttr ib ute s	- I can use a computer program to present information in different ways -I can describe music using adjectives - I can identify simple differences in pieces of music																	Maths	- Privacy and security
2	4	Data and information – Pictograms	6	-To explain that we can present information using a computer	- I can say what I do and don't like about a piece of m usic -I can create a rhythm pattern - I can explain that music is created and played by hum ans - I can play an instrument following a rhythm pattern																	Maths	- Privacy and security
2	5	Creating media - Digital music	1	-To say how music can make us feel	-I can connect images with sounds - I can relate an idea to a piece of music - I can use a computer to experiment with pitch -I can explain how my music can be played in different ways																		- Copyright and ownership
2	5	Creating media - Digital music	2	-To identify that there are patterns in music	- I can identify that music is a sequence of notes - I can refine my musical pattern on a computer -I can add a sequence of notes to my rhythm - I can create a rhythm which represents an animal I've chosen																		- Copyright and ownership
2	5	Creating media - Digital music	3	-To experiment with sound using a computer	- I can create my animal's rhythm on a computer -I can explain how I changed my work - I can listen to music and describe how it makes me feel																		- Copyright and ownership
2	5	Creating media - Digital music	4	-To use a computer to create a musical pattern	- I can review my work -I can identify that a program needs to be started - I can identify the start of a sequence - I can show how to run my program																		- Copyright and ownership
2	5	Creating media - Digital music	5	-To create music for a purpose	-I can change the outcome of a sequence of c o mm ands - I can match two sequences with the same o utc om e - I can predict the outcome of a sequence of c o mm ands																		- Copyright and ownership
2	5	Creating media - Digital music	6	-To review and refine our computer work	-I can build the sequences of blocks I need - I can decide which blocks to use to meet the design - I can work out the actions of a sprite in an algorithm																		- Copyright and ownership
2	6	Programming B - Programming	1	-To explain that a sequence of commands has a start	-I can choose backgrounds for the design - I can choose characters for the design - I can create a program based on the new design -I can build sequences of blocks to match my design																		
2	6	Programming B - Programming q uizze s	2	-To explain that a sequence of commands has an o utc om e	- I can choose the images for my own design - I can create an algorithm -I can compare my project to my design - I can debug my program - I can improve my project by adding features																		
2	6	Programming B - Programming q uizze s	3	-To create a program using a given design																			
2	6	Programming B - Programming	4	-To change a given design																			
2	6	Programming B - Programming q uizze s	5	-To create a program using my own design																			
2	6	Programming B - Programming	6	-To decide how my project can be improved																			

YEAR 3 / 4 OVERVIEW

Year Gr oup	Suggested Or der	Unit Name	Lesson	Learning Objectives	Success Criteria	National Curriculum Links 2.1	2.2	2.3	2.4	2.5	2.6	2.7	RPF Computing Taxonomy AL	CM	CS	DD	DI	ET	IT	NW	PG	SS	Cross Curricular Links	Education for a Connected World
3	1	Computing systems and networks – Connecting computers	1	-To explain how digital devices function	-I can explain that digital devices accept inputs - I can explain that digital devices produce outputs - I can follow a process																			
3	1	Computing systems and networks – Connecting computers	2	-To identify input and output devices	-I can classify input and output devices - I can describe a simple process - I can design a digital device																			
3	1	Computing systems and networks – Connecting computers	3	-To recognise how digital devices can change the way we work	-I can explain how I use digital devices for different activities - I can recognise similarities between using digital devices and non-digital tools - I can suggest differences between using digital devices and non-digital tools																			
3	1	Computing systems and networks – Connecting computers	4	-To explain how a computer network can be used to share information	-I can discuss why we need a network switch - I can explain how messages are passed through multiple connections - I can recognise different connections -I can identify how devices in a network are connected together																			
3	1	Computing systems and networks – Connecting computers	6	-To recognise the physical components of a network	- I can identify networked devices around me - I can identify the benefits of computer networks -I can create two groups of objects separated by one attribute - I can investigate questions with yes/no answers																			
3	4	Data and information – Branching databases	1	-To create questions with yes/no answers	- I can make up a yes/no question about a collection of objects -I can arrange objects into a tree structure - I can create a group of objects within an existing gro up - I can select an attribute to separate objects into groups																			
3	4	Data and information – Branching databases	2	-To identify the attributes needed to collect data about an object	-I can group objects using my own yes/no questions - I can select objects to arrange in a branching database - I can test my branching database to see if it works																			
3	4	Data and information – Branching databases	3	-To create a branching database																				
3	4	Data and information – Branching databases	4	-To explain why it is helpful for a database to be well structured	-I can compare two branching database structures - I can create yes/no questions using given att ri but es - I can explain that questions need to be ordered carefully to split objects into similarly sized groups -I can create a physical version of a branching database																			
3	4	Data and information – Branching databases	5	-To plan the structure of a branching database	- I can create questions that will enable objects to be uniquely identified - I can independently create questions to use in a branching database																			
3	4	Data and information – Branching databases	6	-To independently create an identification tool	-I can create a branching database that reflects my plan - I can suggest real-world uses for branching databases - I can work with a partner to test my identification tool																			
3	3	Programming A - Sequencing sounds	1	-To explore a new programming environment	-I can explain that objects in Scratch have attributes (linked to) - I can identify the objects in a Scratch project (sprites, backdrops) - I can recognise that commands in Scratch are represented as blocks																			
3	3	Programming A - Sequencing sounds	2	-To identify that commands have an outcome	-I can choose a word which describes an on-screen action for my plan - I can create a program following a design - I can identify that each sprite is controlled by the commands I choose																			
3	3	Programming A - Sequencing sounds	3	-To explain that a program has a start	-I can create a sequence of connected commands - I can explain that the objects in my project will respond exactly to the code - I can start a program in different ways																			
3	3	Programming A - Sequencing sounds	4	-To recognise that a sequence of commands can have an order	-I can combine sound commands - I can explain what a sequence is - I can order notes into a sequence																			
3	3	Programming A - Sequencing sounds	6	-To create a project from a task description	-I can identify and name the objects I will need for a pro j ec t - I can implement my algorithm as code - I can relate a task description to a design																			
3	2	Creating media - Stop-frame animation	1	-To explain that animation is a sequence of drawings or photographs	-I can create an effective flip book—style animation - I can draw a sequence of pictures - I can explain how an animation/flip book works																			- Copyright and ownership - Managing online information
3	2	Creating media - Stop-frame animation	2	-To relate animated movement with a sequence of images	-I can create an effective stop-frame animation - I can explain why little changes are needed for each frame - I can predict what an animation will look like																			- Copyright and ownership - Managing online information
3	2	Creating media - Stop-frame animation	3	-To plan an animation	-I can break down a story into settings, characters and events - I can create a storyboard - I can describe an animation that is achievable on screen																			- Copyright and ownership - Managing online information
3	2	Creating media - Stop-frame animation	4	-To identify the need to work consistently and c arefu l l y	-I can evaluate the quality of my animation - I can review a sequence of frames to check my work - I can use onion skinning to help me make small changes between frames																			- Copyright and ownership - Managing online information
3	2	Creating media - Stop-frame animation	5	-To review and improve an animation	-I can evaluate another learner's animation - I can explain ways to make my animation better - I can improve my animation based on feedback																			- Copyright and ownership - Managing online information
3	2	Creating media - Stop-frame animation	6	-To evaluate the impact of adding other media to an animation	-I can add other media to my animation - I can evaluate my final film - I can explain why I added other media to my animation																			- Copyright and ownership - Managing online information
4	4	Data and information – Data logging	1	-To explain that data gathered over time can be used to answer questions	-I can choose a data set to answer a given question - I can identify data that can be gathered over time - I can suggest questions that can be answered using a given data set																			
4	4	Data and information – Data logging	2	-To use a digital device to collect data automatically	-I can explain what data can be collected using s en s ors - I can identify that data from sensors can be recorded - I can use data from a sensor to answer a given question																			
4	4	Data and information – Data logging	3	-To explain that a data logger collects 'data points' from sensors over time	-I can identify the intervals used to collect data - I can recognise that a data logger collects data at given points - I can talk about the data that I have captured -I can explain that there are different ways to view																			
4	4	Data and information – Data logging	4	-To recognise how a computer can help us analyse data	data - I can sort data to find information - I can view data at different levels of detail																			
4	4	Data and information – Data logging	5	-To identify the data needed to answer questions	-I can plan how to collect data using a data logger - I can propose a question that can be answered using logged data - I can use a data logger to collect data																			
4	6	Programming B – Repetition in games	1	-To develop the use of count-controlled loops in a different programming environment	-I can list an everyday task as a set of instructions including repetition -I can modify a snippet of code to create a given outcome - I can predict the outcome of a snippet of code																			
4	6	Programming B – Repetition in games	2	-To explain that in programming there are infinite loops and count controlled loops	-I can choose when to use a count-controlled and an infinite loop - I can modify loops to produce a given outcome - I can recognise that some programming languages enable more than one process to be run at once																			
4	6	Programming B – Repetition in games	3	-To develop a design that includes two or more loops which run at the same time	-I can choose which action will be repeated for each object - I can evaluate the effectiveness of the repeated sequences used in my program - I can explain what the outcome of the repeated action should be																			
4	6	Programming B – Repetition in games	4	-To modify an infinite loop in a given program	-I can explain the effect of my changes - I can identify which parts of a loop can be changed - I can re-use existing code snippets on new sprites																			
4	6	Programming B – Repetition in games	5	-To design a project that includes repetition	-I can develop my own design explaining what my project will do - I can evaluate the use of repetition in a project - I can select key parts of a given project to use in																			
4	6	Programming B – Repetition in games	6	-To create a project that includes repetition	-I can build a program that follows my design - I can evaluate the steps I followed when building my project - I can refine the algorithm in my design																			
4	2	Creating media - Audio production	6	-To evaluate the effective use of audio	-I can choose appropriate edits to improve my podcast - I can listen to an audio recording to identify its s tren gth s																			- Copyright and ownership

YEAR 5 / 6 DETAILED OVERVIEW

						National Curriculum Links							RPF Computing Taxonomy											
Year Gr oup	Suggested Or der	Unit Name	Lesson	LearningObjectives	Success Criteria	2.1	2.2	2.3	2.4	2.5	2.6	2.7	AL	CM	CS	DD	DI	ET	IT	NW	PG	SS	Cross Curricular Links	Education for a Connected World
5	1	Computing systems and networks – The Internet	1	-To describe how networks physically connect to other networks	-I can demonstrate how information is shared across the internet - I can describe the internet as a network of networks - I can discuss why a network needs protecting -I can describe networked devices and how they connect																			
5	1	Computing systems and networks – The Internet	2	-To recognise how networked devices make up the internet	- I can explain that the internet is used to provide many services - I can recognise that the World Wide Web contains websites and web pages -I can describe how to access websites on the WWW																			
5	1	Computing systems and networks – The Internet	3	-To outline how websites can be shared via the World Wide Web (WWW)	- I can describe where websites are stored when uploaded to the WWW - I can explain the types of media that can be shared on the WWW -I can explain that internet services can be used to create content online																			
5	1	Computing systems and networks – The Internet	4	-To describe how content can be added and accessed on the World Wide Web (WWW)	- I can explain what media can be found on websites - I can recognise that I can add content to the WWW																			
5	1	Computing systems and networks – The Internet	5	-To recognise how the content of the WWW is created by people	-I can explain that there are rules to protect content - I can explain that websites and their content are created by people - I can suggest who owns the content on websites																			
5	1	Computing systems and networks – The Internet	6	-To evaluate the consequences of unreliable content	-I can explain that not everything on the World Wide Web is true - I can explain why I need to think carefully before I share or reshare content - I can explain why some information I find online may not be honest, accurate, or legal																			
5	1	Computing systems and networks - Systems and s ea rc hi ng	1	-To explain that computers can be connected together to form systems	-I can describe that a computer system features inputs, processes, and outputs - I can explain that computer systems communicate with other devices - I can explain that systems are built using a number of parts																			- Copyright and ownership
5	1	Computing systems and networks - Systems and s ea rc hi ng	2	-To recognise the role of computer systems in our lives	-I can explain the benefits of a given computer system - I can identify tasks that are managed by computer systems - I can identify the human elements of a computer system																			- Copyright and ownership
5	1	Computing systems and networks - Systems and s ea rc hi ng	3	-To experiment with search engines	-I can compare results from different search engines - I can make use of a web search to find specific information - I can refine my web search -I can explain why we need tools to find things online																			- Copyright and ownership
5	1	Computing systems and networks - Systems and s ea rc hi ng	4	-To describe how search engines select results	- I can recognise the role of web crawlers in creating an index - I can relate a search term to the search engine's index -I can explain that a search engine follows rules to rank results																			- Copyright and ownership
5	1	Computing systems and networks - Systems and s ea rc hi ng	5	-To explain how search results are ranked	- I can give examples of criteria used by search engines to rank results - I can order a list by rank -I can describe some of the ways that search results can be influenced - I can explain how search engines make money																			- Copyright and ownership
5	1	Computing systems and networks - Systems and s ea rc hi ng	6	-To recognise why the order of results is important, and to whom	- I can recognise some of the limitations of search engines																			- Copyright and ownership
6+69:74	4	Data and information – Flat-file databases	1	-To use a form to record information	-I can create a database using cards - I can explain how information can be recorded - I can order, sort, and group my data cards -I can choose which field to sort data by to answer a given question - I can explain what a field and a record is in a database																			
6	4	Data and information – Flat-file databases	2	-To compare paper and computer-based databases	- I can navigate a flat-file database to compare different views of information -I can combine grouping and sorting to answer specific questions																			
6	4	Data and information – Flat-file databases	3	-To outline how you can answer questions by grouping and then sorting data	-I can explain that data can be grouped using chosen values - I can group information using a database -I can choose multiple criteria to answer a given question																			
6	4	Data and information – Flat-file databases	4	-To explain that tools can be used to select specific data	- I can choose which field and value are required to answer a given question - I can outline how 'AND' and 'OR' can be used to refine data selection																			
6	4	Data and information – Flat-file databases	5	-To explain that computer programs can be used to compare data visually	-I can explain the benefits of using a computer to create charts - I can refine a chart by selecting a particular filter - I can select an appropriate chart to visually compare data																			
6	4	Data and information – Spreadsheets	1	-To create a data set in a spreadsheet	-I can collect data - I can enter data into a spreadsheet - I can suggest how to structure my data																			
6	4	Data and information – Spreadsheets	2	-To build a data set in a spreadsheet	-I can apply an appropriate format to a cell - I can choose an appropriate format for a cell - I can explain what an item of data is																			
6	4	Data and information – Spreadsheets	3	-To explain that formulas can be used to produce calculated data	-I can construct a formula in a spreadsheet - I can explain which data types can be used in calculations - I can identify that changing inputs changes outputs																			
6	4	Data and information – Spreadsheets	4	-To apply formulas to data	-I can apply a formula to multiple cells by duplicating it - I can calculate data using different operations - I can create a formula which includes a range of cells																			
6	4	Data and information – Spreadsheets	5	-To create a spreadsheet to plan an event	-I can apply a formula to calculate the data I need to answer questions - I can explain why data should be organised - I can use a spreadsheet to answer questions																			
6	4	Data and information – Spreadsheets	6	-To choose suitable ways to present data	-I can produce a chart - I can suggest when to use a table or chart - I can use a chart to show the answer to questions																			
5	5	Creating media – Introduction to vector graphics	6	-To apply what I have learned about vector drawings	-I can compare vector drawings to freehand paint dra w i ngs - I can create a vector drawing for a specific purpose - I can reflect on the skills I have used and why I have used them																			- Copyright and ownership
5	6	Programming B – Selection in quizzes	1	-To explain how selection is used in computer programs	-I can identify conditions in a program - I can modify a condition in a program - I can recall how conditions are used in selection -I can create a program with different outcomes using selection																			
5	6	Programming B – Selection in quizzes	2	-To relate that a conditional statement connects a condition to an outcome	- I can identify the condition and outcomes in an 'if... then... else...' statement - I can use selection in an infinite loop to check a																			
5	6	Programming B – Selection in quizzes	3	-To explain how selection directs the flow of a program	-I can design the flow of a program which contains 'if... then... else...' - I can explain that program flow can branch according to a condition - I can show that a condition can direct program flow in one of two ways -I can identify the outcome of user input in an alg ori thm																			
5	6	Programming B – Selection in quizzes	4	-To design a program which uses selection	-I can outline a given task - I can use a design format to outline my project -I can implement my algorithm to create the first section of my program																			
5	6	Programming B – Selection in quizzes	5	-To create a program which uses selection	- I can share my program with others																			
5	6	Programming B – Selection in quizzes	6	-To evaluate my program	-I can extend my program further - I can identify the setup code I need in my program - I can identify ways the program could be improved																			
5	2	Creating media - Video production	1	-To explain what makes a video effective	-I can compare features in different videos - I can explain that video is a visual media format - I can identify features of videos -I can experiment with different camera angles																			- Managing online information - Online relationships - Online reputation - Self-image and identity
5	2	Creating media - Video production	2	-To identify digital devices that can record video	- I can identify and find features on a digital video recording device																			- Managing online information - Online relationships - Online reputation
5	2	Creating media - Video production	3	-To capture video using a range of techniques	-I can capture video using a range of filming techniques - I can review how effective my video is - I can suggest filming techniques for a given purpose																			- Managing online information - Online relationships - Online reputation - Self-image and identity
5	2	Creating media - Video production	4	-To create a storyboard	-I can create and save video content - I can decide which filming techniques I will use - I can outline the scenes of my video																			- Managing online information - Online relationships - Online reputation - Self-image and identity
5	2	Creating media - Video production	5	-To identify that video can be improved through reshooting and editing	-I can explain how to improve a video by reshooting and editing - I can select the correct tools to make edits to my video - I can store, retrieve, and export my recording to a																			- Managing online information - Online relationships - Online reputation - Self-image and identity
5	2	Creating media - Video production	6	-To consider the impact of the choices made when making and sharing a video	-I can evaluate my video and share my opinions - I can make edits to my video and improve the final outcome - I can recognise that my choices when making a video will impact on the quality of the final outcome																			- Managing online information - Online relationships - Online reputation - Self-image and identity

6	3	Programming A – Variables in games	1	-To define a 'variable' as something that is changeable	-I can explain that the way a variable changes can be defined - I can identify examples of information that is variable - I can identify that variables can hold numbers or l ett ers -I can explain that a variable has a name and a value	
6	3	Programming A – Variables in games	2	-To explain why a variable is used in a program	- I can identify a program variable as a placeholder in memory for a single value - I can recognise that the value of a variable can be changed -I can decide where in a program to change a variable	
6	3	Programming A – Variables in games	3	-To choose how to improve a game by using variables	- I can make use of an event in a program to set a variable - I can recognise that the value of a variable can be used by a program -I can choose the artwork for my project - I can create algorithms for my project - I can explain my design choices	
6	3	Programming A – Variables in games	4	-To design a project that builds on a given example	-I can choose a name that identifies the role of a variable - I can create the artwork for my project - I can test the code that I have written	
6	3	Programming A – Variables in games	5	-To use my design to create a project	-I can identify ways that my game could be im proved - I can share my game with others - I can use variables to extend my game	
6	3	Programming A – Variables in games	6	-To evaluate my project	-I can create a database using cards - I can explain how information can be recorded - I can order, sort, and group my data cards -I can choose which field to sort data by to answer a given question	
6	4	Data and information – Flat-file databases	1	-To use a form to record information	- I can explain what a field and a record is in a database - I can navigate a flat-file database to compare different views of information	
6	4	Data and information – Flat-file databases	2	-To compare paper and computer-based databases	-I can combine grouping and sorting to answer specific questions -I can explain that data can be grouped using chosen values - I can group information using a database -I can choose multiple criteria to answer a given question	
6	4	Data and information – Flat-file databases	3	-To outline how you can answer questions by grouping and then sorting data		
6	4	Data and information – Flat-file databases	4	-To explain that tools can be used to select specific data	- I can choose which field and value are required to answer a given question - I can outline how 'AND' and 'OR' can be used to refine data selection	
6	4	Data and information – Flat-file databases	5	-To explain that computer programs can be used to compare data visually	-I can explain the benefits of using a computer to create charts - I can refine a chart by selecting a particular filter - I can select an appropriate chart to visually compare data -I can ask questions that will need more than one field to answer	
6	4	Data and information – Flat-file databases	6	-To use a real-world database to answer questions	- I can present my findings to a group - I can refine a search in a real-world context	
6	5	Creating media – 3D Modelling	1	-To recognise that you can work in three dimensions on a computer	-I can add 3D shapes to a project - I can move 3D shapes relative to one another - I can view 3D shapes from different perspectives -I can lift/lower 3D objects	- Privacy and security
6	5	Creating media – 3D Modelling	2	-To identify that digital 3D objects can be modified	- I can recolour a 3D object - I can resize an object in three dimensions -I can duplicate 3D objects	- Privacy and security
6	5	Creating media – 3D Modelling	3	-To recognise that objects can be combined in a 3D model	- I can group 3D objects - I can rotate objects in three dimensions -I can accurately size 3D objects - I can combine a number of 3D objects	- Privacy and security
6	5	Creating media – 3D Modelling	4	-To create a 3D model for a given purpose	- I can show that placeholders can create holes in 3D objects -I can analyse a 3D model - I can choose objects to use in a 3D model - I can combine objects in a design	- Privacy and security
6	5	Creating media – 3D Modelling	5	-To plan my own 3D model		- Privacy and security
6	5	Creating media – 3D Modelling	6	-To create my own digital 3D model	-I can construct a 3D model based on a design - I can explain how my 3D model could be improved - I can modify my 3D model to improve it	- Privacy and security

VOCABULARY DEVELOPMENT

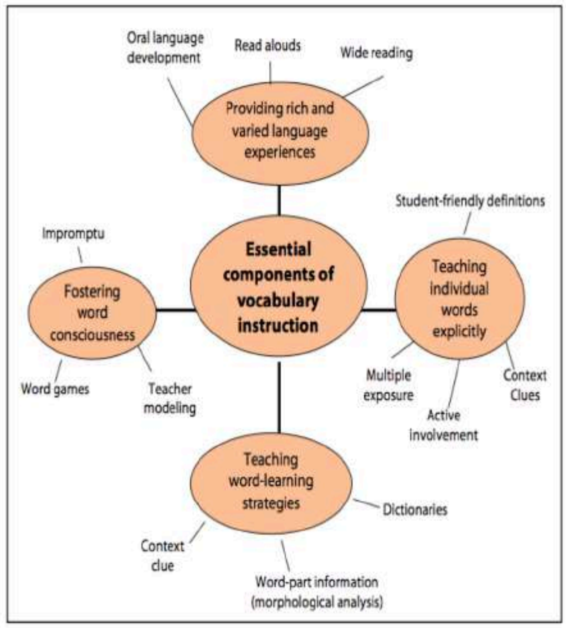


Figure 1. The four essential components of vocabulary instruction. Adapted from Graves, 2006.

The Three Pillars of Vocabulary Teaching
1. Explicit vocabulary teaching
2. Incidental vocabulary learning
3. Cultivating 'word consciousness'

Explicit vocabulary teaching
Explicit vocabulary teaching can provide a vital boost to our pupils' vocabulary development. At Harper Bell SDA School, our tier overview explicitly outlines vocabulary to be taught, retrieved and deeply embedded into the learning of our pupils. This vocabulary forms part of the vocabulary schemas developed across our Art curriculum. The Curriculum has been carefully aligned to promote opportunities for explicit vocabulary teaching.

Incidental vocabulary learning
We are aware that not all words can be taught! In Computing, we have carefully selected subject specific vocabulary and tier 2 words to enrich the language development of our pupils. To support this, we have developed a reading overview which links to the development of Computing knowledge and skills, whilst also providing a wealth of vocabulary to further develop and reinforce the vocabulary introduced within lessons. Reading rich texts, both in the classroom and beyond the school gates, is critical for language and vocabulary development. Our teachers flood their classrooms with vocabulary alongside the explicit teaching of vocabulary.

Cultivating 'word consciousness'
Our aim is to develop a curiosity about words and how they have developed by exploring them. We aim for our pupils to make connections between words and roots of words.
.
"Education is the process of preparing us for the big world and the big world has big words. The more big words I know, the better I will survive in it. Because there are hundreds of thousands of big words in English, I cannot learn them all. But this does not mean that I shouldn't try to learn some."
David Crystal, 'Words, Words, Words'

VOCABULARY OVERVIEW

*WVOCABULARY FROM PREVIOUS UNITS ARE REVISITED AND RETRIEVED TO PRMOTE THESE BEING EMBEDDED IN THE LONG TERM MEMORY AND TO FURTHER DEVELOP THE SCHEMA BETIWEEN LEARNING

	Year 1		
Computing systems and networks - Technology around us	**Computing systems and networks - Information technology around us**	**Programming B – Programming animations**	**Creating media - Digital writing**
technology, computer, mouse, trackpad, keyboard, screen, double-click, typing.	Information technology (IT), computer, barcode, scanner/scan	ScratchJr, command, sprite, compare, programming, area, block, joining, start, run, program, background, delete, reset, algorithm, predict, effect, change, value, instructions, design.	word processor, keyboard, keys, letters, type, numbers, space, backspace, text cursor, capital letters, toolbar, bold, italic, underline, mouse, select, font, undo, redo, format, compare, typing, writing.

Year 2		
Data and information – Pictograms	**Programming B - Programming quizzes**	**Creating media - Digital music**
more than, less than, most, least, common, popular,	sequence, command, program, run, start, outcome, predict, blocks, design, actions, sprite, project, modify, change, algorithm,	music, quiet, loud, feelings, emotions, pattern, rhythm, pulse, pitch, tempo, rhythm, notes, create, emotion, beat, instrument, open, edit.

Year 3

Computing systems and networks - Connecting computers	**Programming A - Sequencing sounds**	**Creating Media - Stop-frame animation**
digital device, input, process, output, program, digital, non-digital, connection, network, switch, server, wireless access point, cables, sockets	Scratch, programming, blocks, commands, code, sprite, costume, stage, backdrop, motion, turn, point in direction,go to, glide, sequence,event, task, design, runthecode, order, note, chord,algorithm, bug, debug,code.	animation, flip book, stop-frame, frame, sequence, image, photograph, setting, character, events, onion skinning, consistency, evaluation, delete, media, import, transition.
Data and Information - Branching databases		
attribute,value, questions, table, objects, branching, database, objects, equal, even, separate, structure, compare, order, organise, selecting, information, decision tree.		

Year 4

Data and Information - Data logging	**Year 4** **Creating Media - Audio production**	**Programming B - Repetition in games**
data, table, layout, input device, sensor, logger, logging, data point, interval, analyse, dataset, import, export, logged, collection, review, conclusion.	audio, microphone, speaker, headphones, input device, output device, sound, podcast, edit, trim, align, layer, import, record, playback, selection, load, save, export,MP3, evaluate, feedback.	Scratch,programming, sprite, blocks, code, loop, repeat, value, infinite loop, count-controlled loop, costume, repetition, forever, animate, event block, duplicate, modify, design, algorithm, debug, refine, evaluate.

Year 5

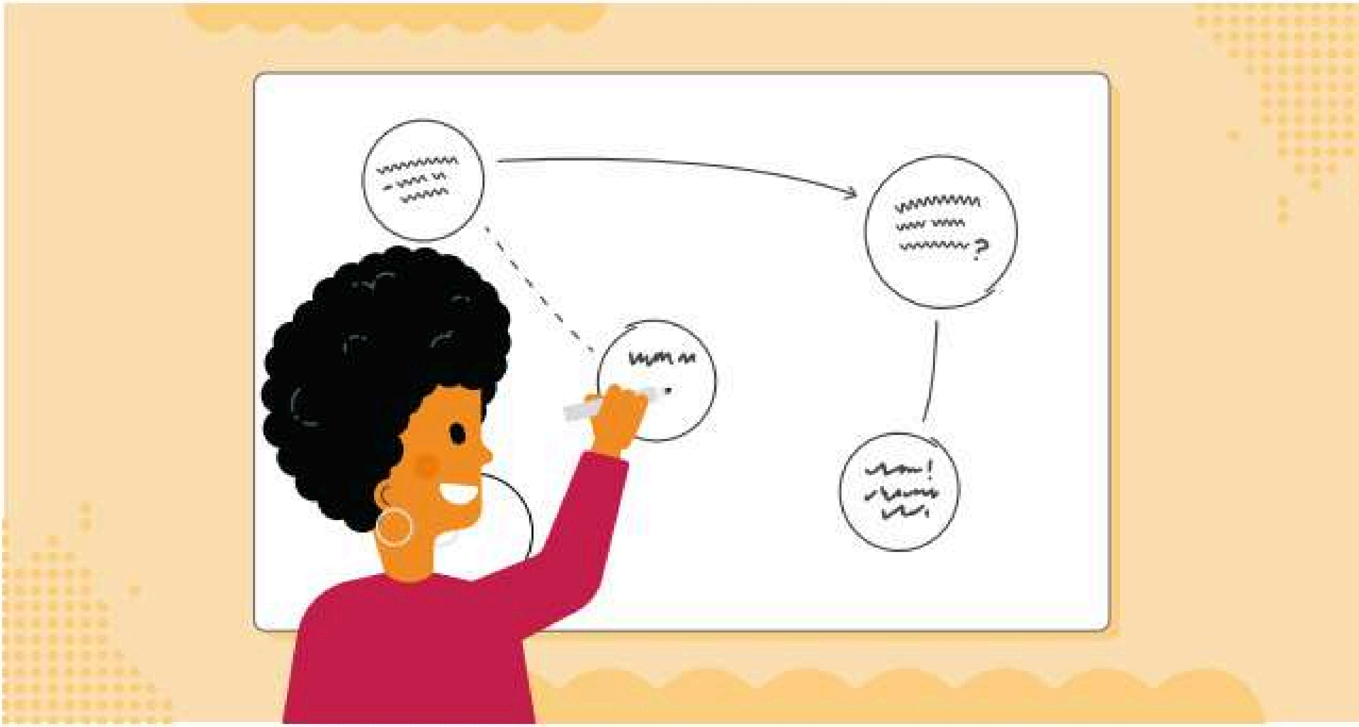
Computing systems and networks - systems and searching	**Computing systems and networks - Connecting computers - The internet**	**Creating Media – Video production**
system, connection, digital, input, process, storage, output, search, search engine, refine, index, bot, ordering, links, algorithm, search engine optimisation (SEO), web crawler, content creator, selection, ranking.	internet, network, router, security, switch, server, wireless access point (WAP), website,web page, web address,routing, web browser, WorldWide Web, content, links,files, use, download,sharing, ownership,permission, information,accurate, honest, content, adverts	video, audio, camera, talking head, panning, close up, video camera, microphone, lens, mid-range, long shot, moving subject, side by side, angle (high, low, normal), static, zoom, pan, tilt, storyboard, filming, review, import, split, trim, clip, edit, reshoot, delete, reorder, export, evaluate, share.
Programming A - Selection in physical computing		
microcontroller, USB, components, connection, infinite loop, output component, motor, repetition, count-controlled loop, Crumble controller, switch, LED, Sparkle, crocodile clips, connect, battery box, program, condition, Input, output, selection, action, debug, circuit, power,cell, buzzer		

Year 6

Data and Information – Flat-file databases	**Data and Information -Introduction to spreadsheets**	**Creating Media 3D Modelling**
database, data,information, record, field,sort, order, group, search,value, criteria, graph, chart,axis, compare, filter, presentation.	data,collecting,table, structure, spreadsheet, cell, cell reference, data item, format, formula, calculation, spreadsheet, input, output, operation, range, duplicate, sigma, propose, question, data set, organised, chart, evaluate, results, sum, comparison, software, tools.	TinkerCAD, 2D, 3D, shapes, select, move, perspective, view, handles, resize, lift, lower, recolour, rotate, duplicate, group, cylinder, cube, cuboid, sphere, cone, prism, pyramid, placeholder, hollow, choose, combine, construct, evaluate, modify.
Programming – Variables in games		
variable, change, name, value, set, design, event, algorithm, code, task, artwork, program, project, code, test, debug, improve, evaluate, share, assign, declare		

Progression within a unit — learning graphs

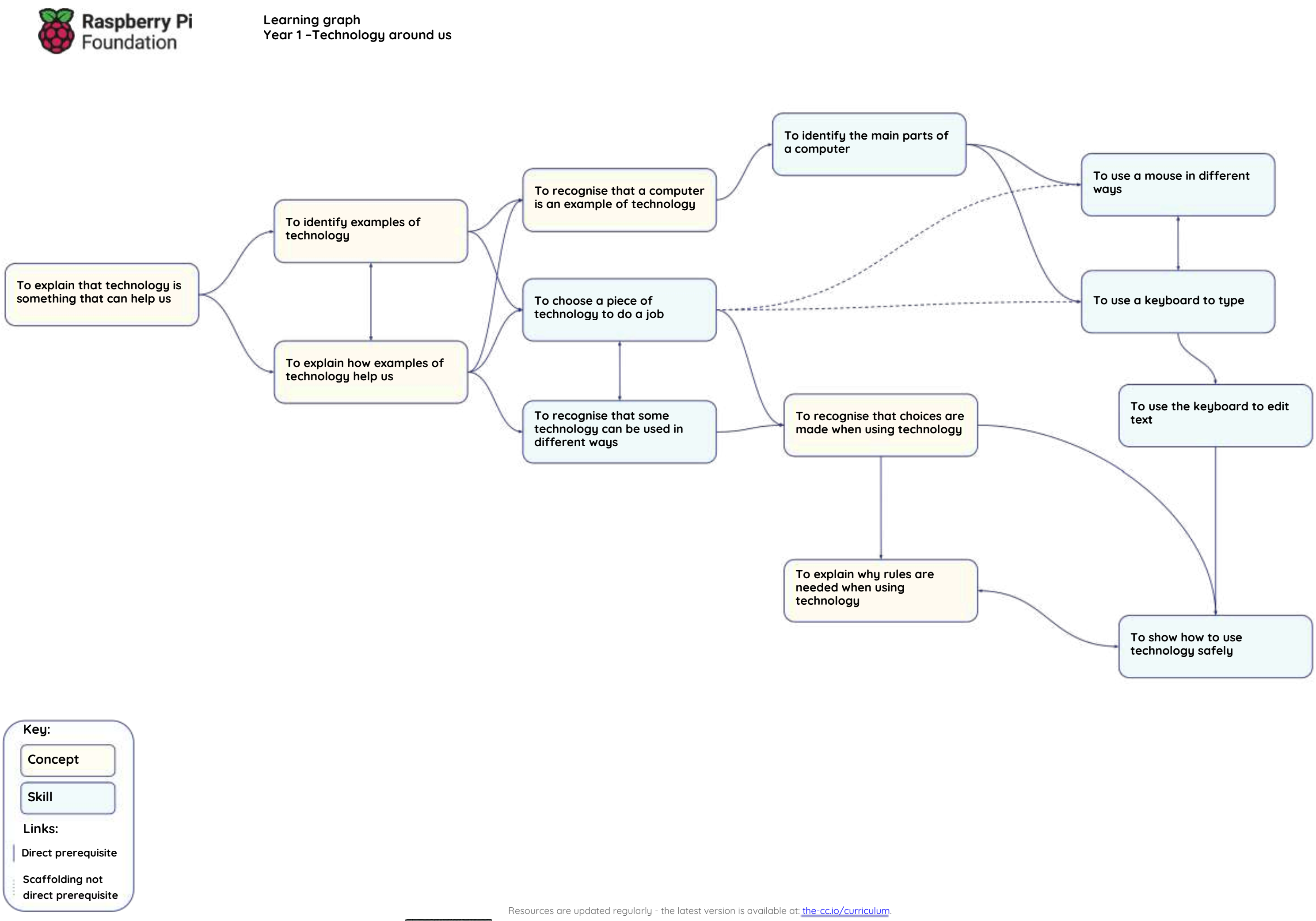
Learning graphs are provided as part of each unit and demonstrate progression through concepts and skills. In order to learn some of those concepts and skills, learners need prior knowledge of others, so the learning graphs show which concepts and skills need to be taught first and which could be taught at a different time.



The learning graphs often show more statements than there are learning objectives. All of the skills and concepts learnt are included in the learning graphs. Some of these skills and concepts are milestones, which form learning objectives, while others are smaller steps towards these milestones, which form success criteria. **Please note** that the wording of the statements may be different in the learning graphs than in the lessons, as the learning graphs are designed for teachers, whereas the learning objectives and success criteria are age-appropriate so that they can be understood by learners. In each year group, there are two 'Programming' units of work, but only one 'Programming' learning graph. The second 'Programming' unit builds on the content in the first 'Programming' unit so closely that there is no specific divide where one ends and the other begins.

LEARNING GRAPHS

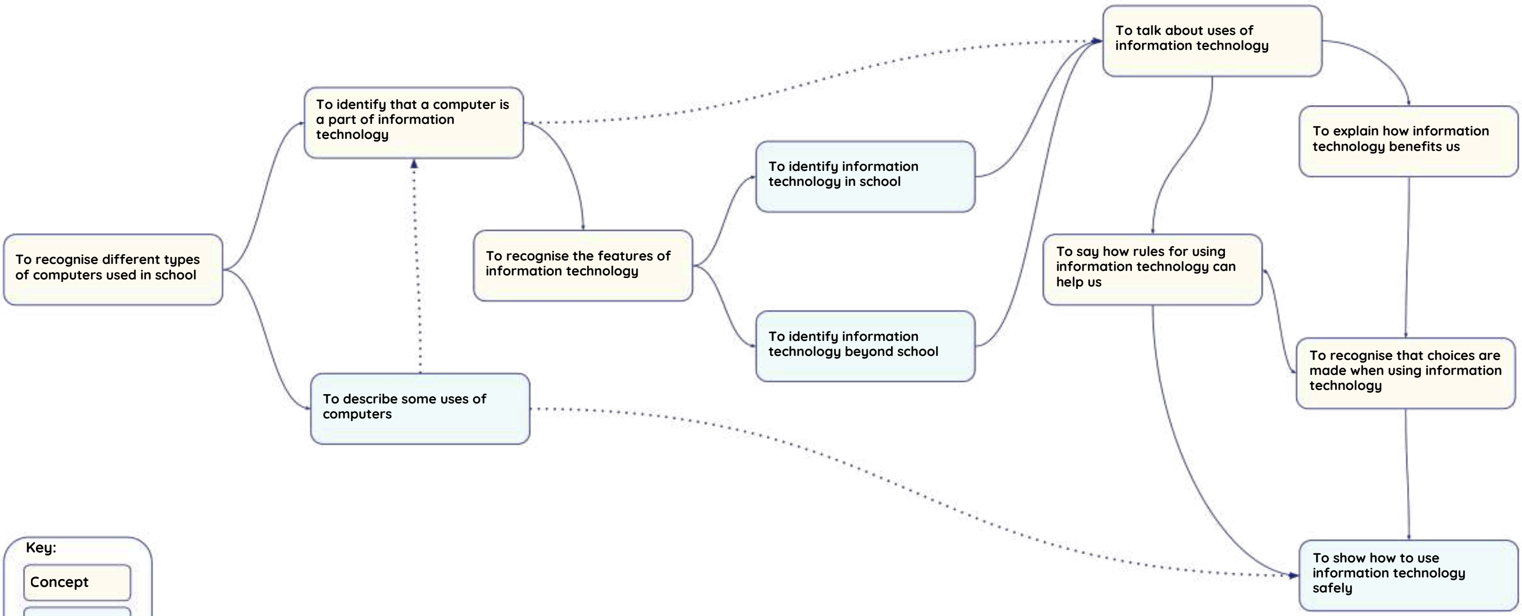
COMPUTER SYSTEMS AND NETWORKS

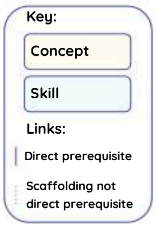


Resources are updated regularly - the latest version is available at: the-cc.io/curriculum.
This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of this license, visit, see creativecommons.org/licenses/by-nc-sa/4.0/

Raspberry Pi Foundation

Learning graph
Year 2 – Information technology around us

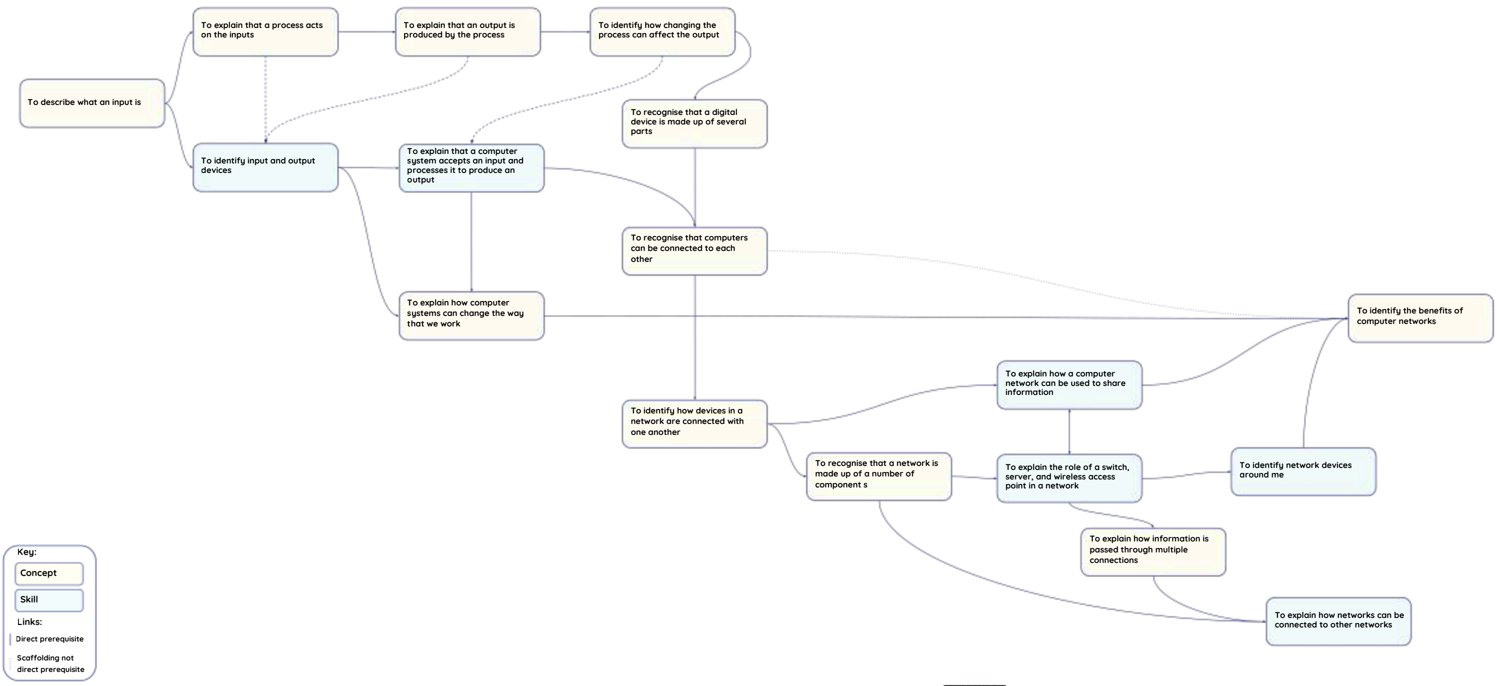




Resources are updated regularly - the latest version is available at: the-cc.io/curriculum.
This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of this license, visit, see creativecommons.org/licenses/by-nc-sa/4.0/

Raspberry Pi Foundation

Learning graph
Year 3 –Connecting computers

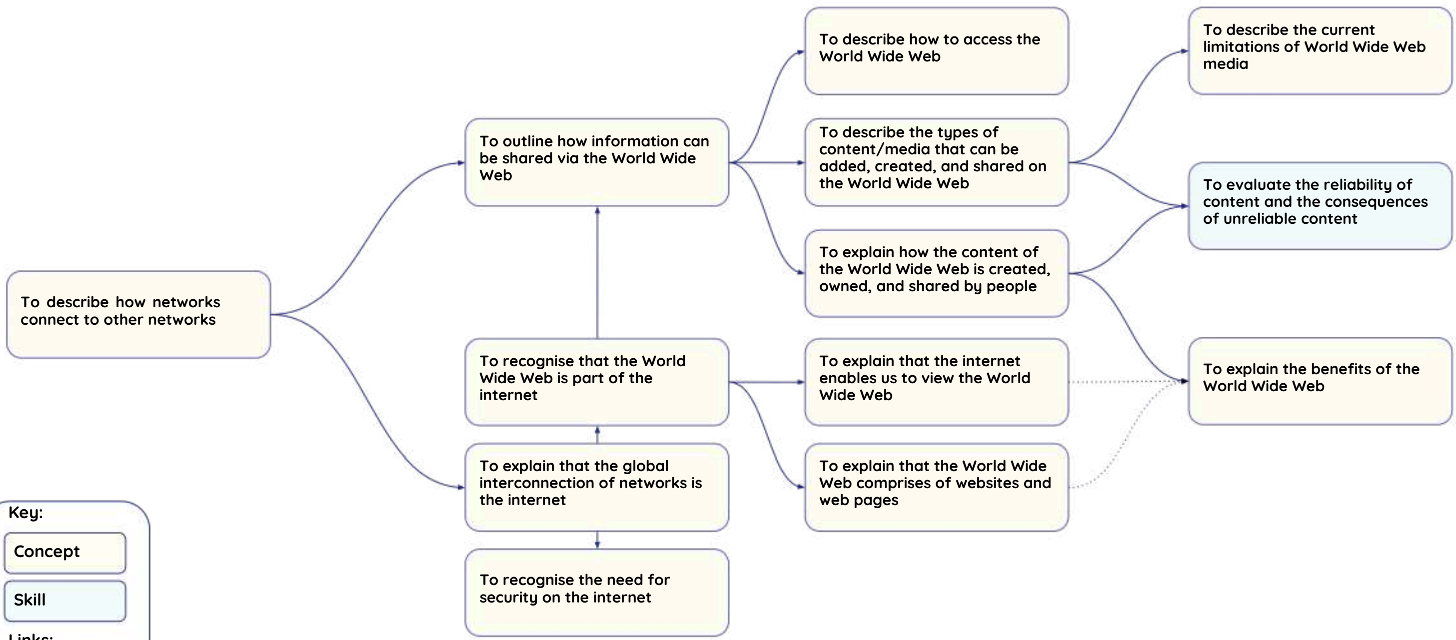


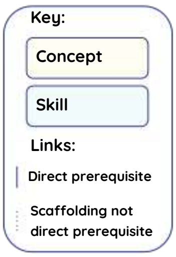
This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution 4.0 International licence. For more information on this licence, see creativecommons.org/licenses/by-sa/4.0.

Raspberry Pi Foundation

Learning graph

Year 4 – Computing systems and networks – The internet

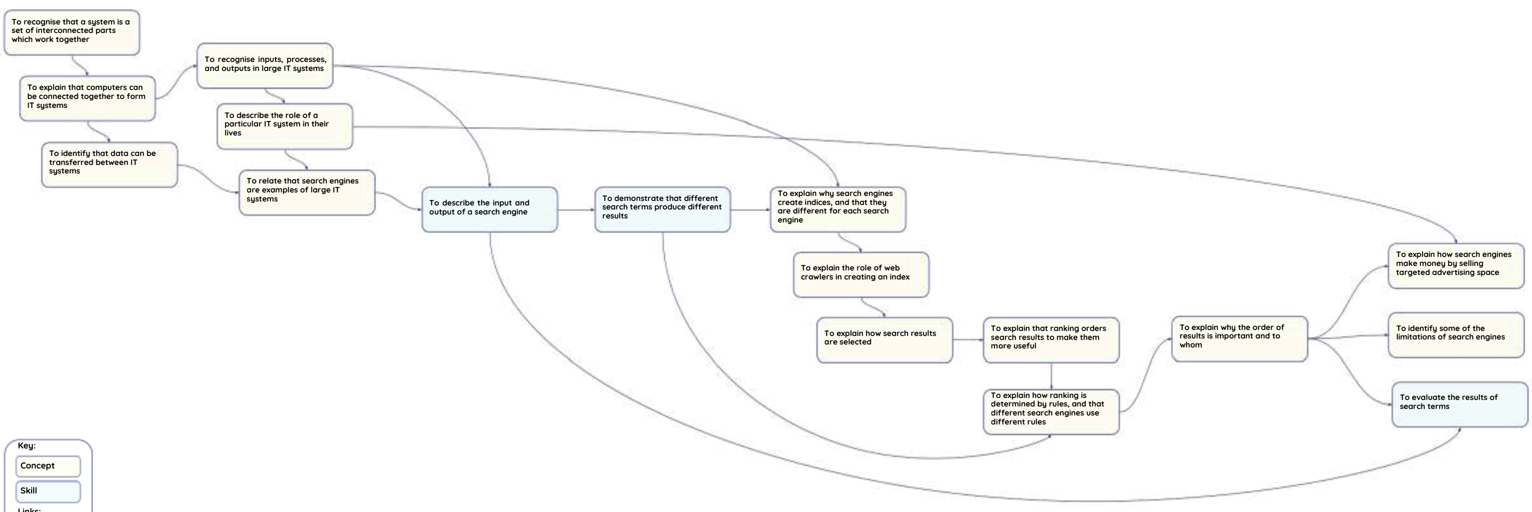




Resources are updated regularly - the latest version is available at: the-cc.io/curriculum.
This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of this license, visit, see creativecommons.org/licenses/by-nc-sa/4.0/

Raspberry Pi Foundation

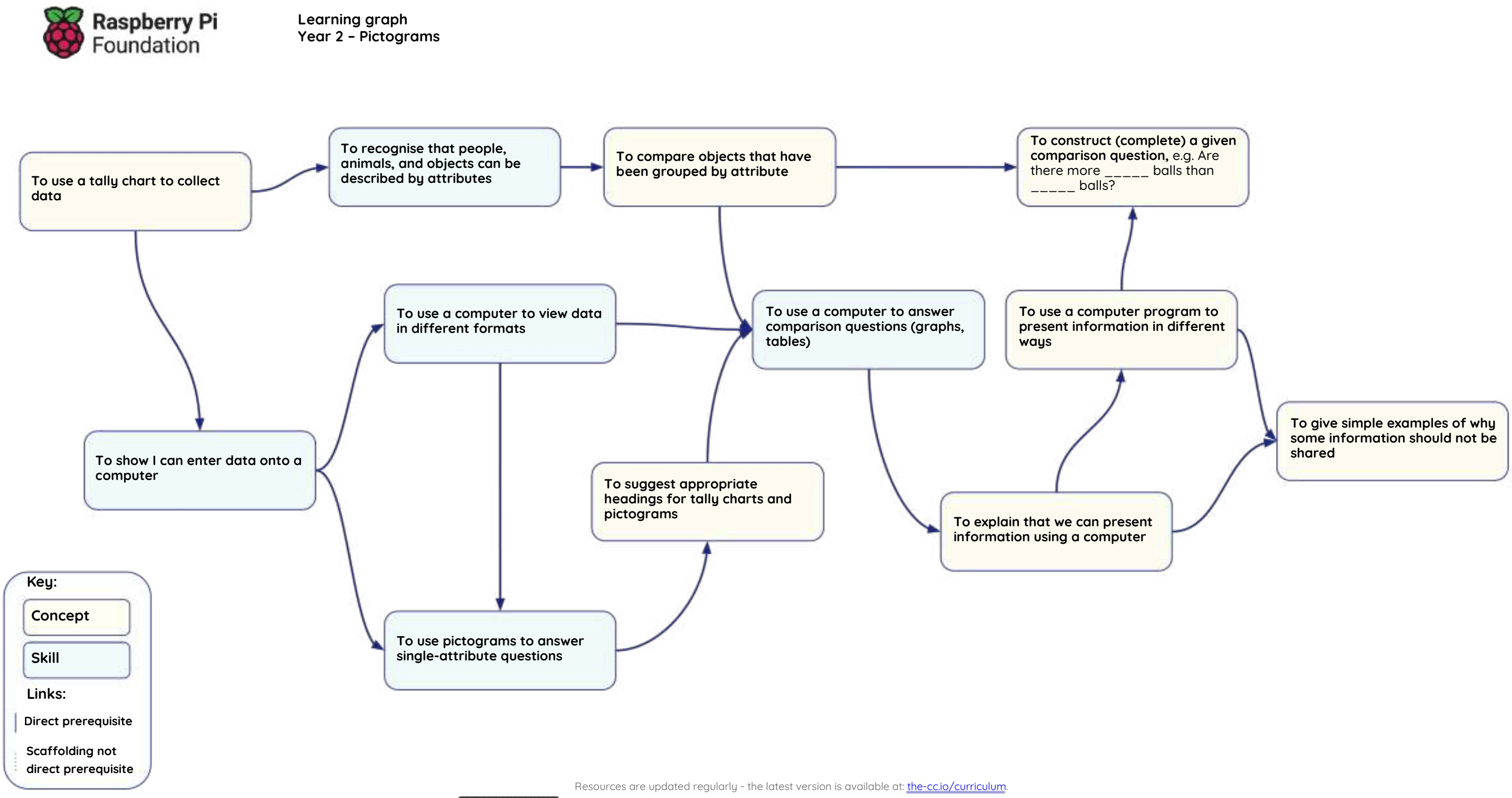
Learning graph
Year 5 – Systems and searching



Resources are updated regularly - the latest version is available at: the-cc.io/curriculum.
This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of this license, visit, see creativecommons.org/licenses/by-nc-sa/4.0/

LEARNING GRAPHS

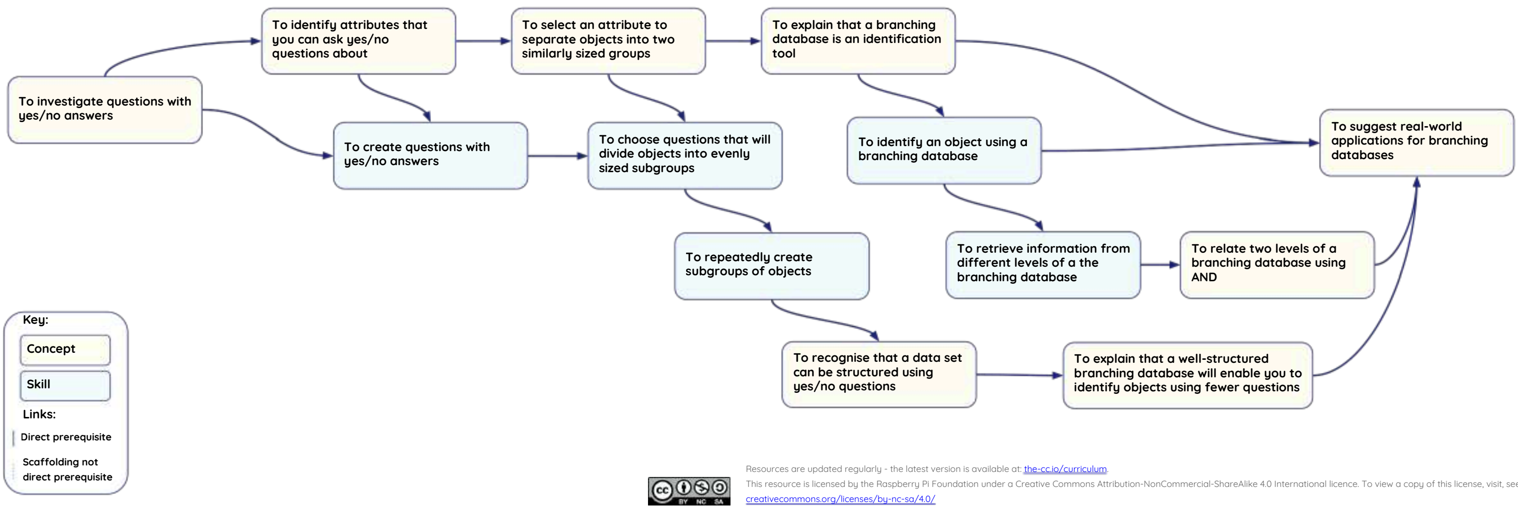
DATA AND INFORMATION



Resources are updated regularly - the latest version is available at: the-cc.io/curriculum.
This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of this license, visit, see creativecommons.org/licenses/by-nc-sa/4.0/

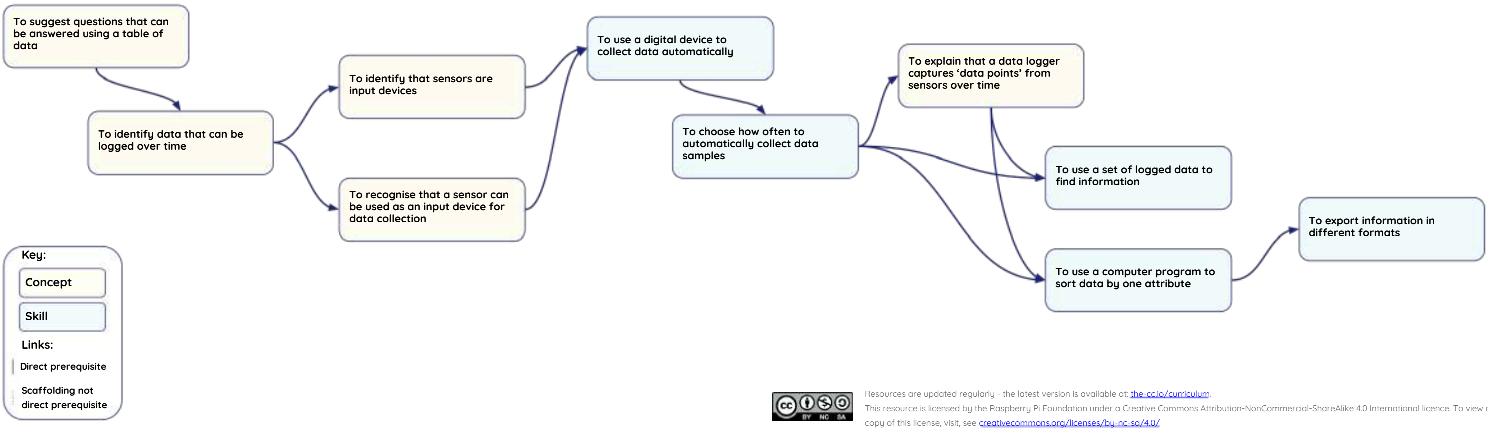
Raspberry Pi Foundation

Learning graph
Year 3 – Branching databases



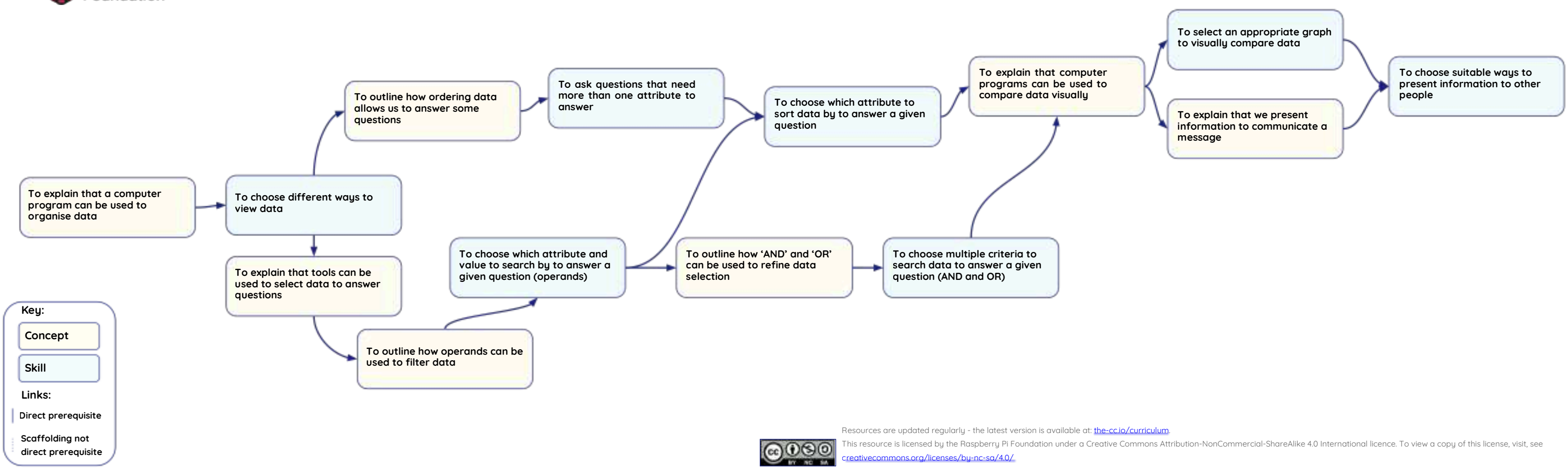
Resources are updated regularly - the latest version is available at: the-cc.io/curriculum.
This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of this license, visit, see creativecommons.org/licenses/by-nc-sa/4.0/

Learning graph
Year 4 – Data and information – Data logging



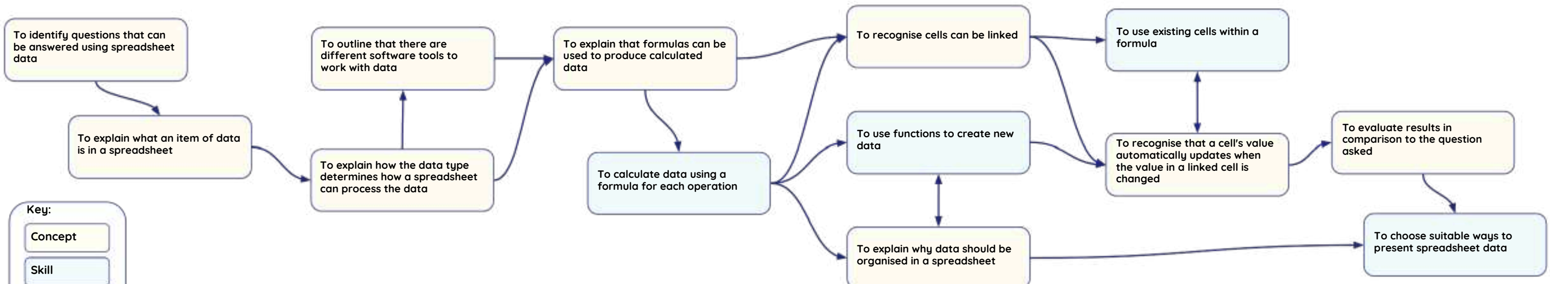
Resources are updated regularly - the latest version is available at: the-cc.io/curriculum.
This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of this license, visit, see creativecommons.org/licenses/by-nc-sa/4.0/

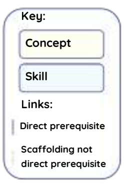
Learning graph
Year 5 - Data and information - Flat-file Databases



Resources are updated regularly - the latest version is available at: the-cc.io/curriculum.
This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of this license, visit, see creativecommons.org/licenses/by-nc-sa/4.0/.

Learning graph
Year 6 – Introduction to spreadsheets

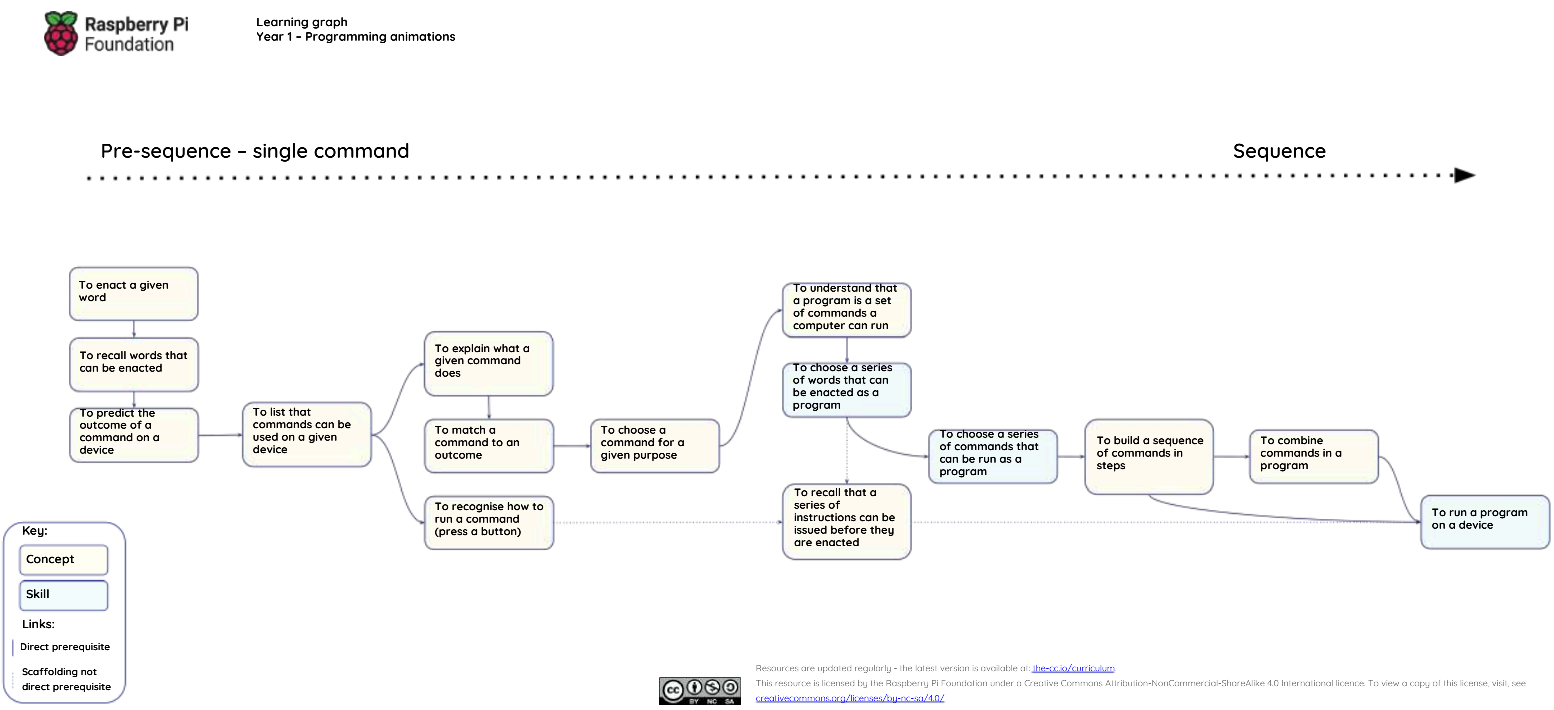




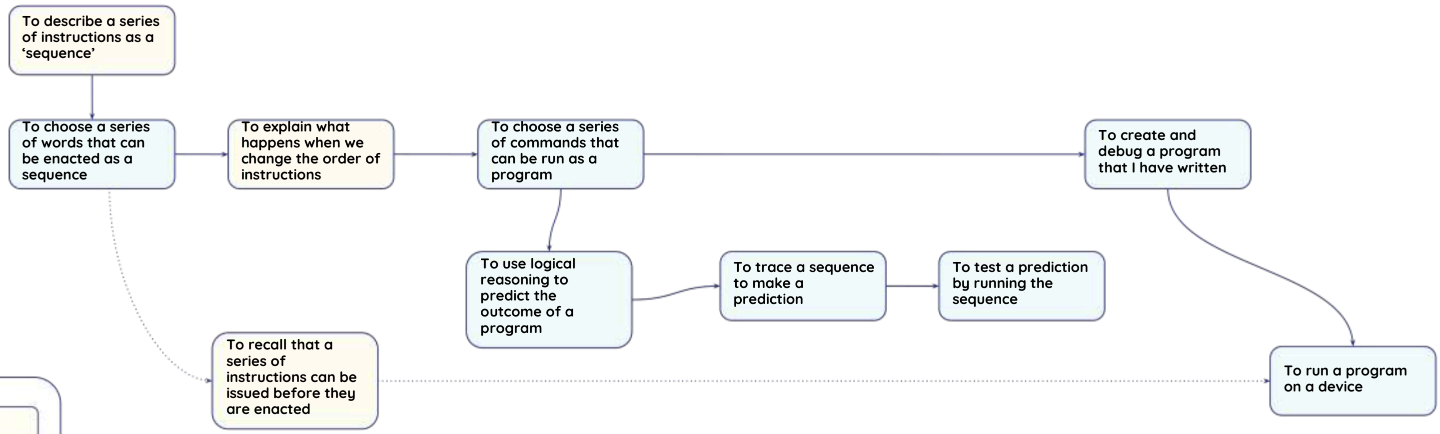
Resources are updated regularly - the latest version is available at: the-cc.io/curriculum
This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of this license, visit, see creativecommons.org/licenses/by-nc-sa/4.0/

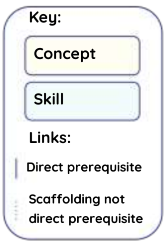
LEARNING GRAPHS

PROGRAMMING



Learning graph
Year 2 – Programming quizzes

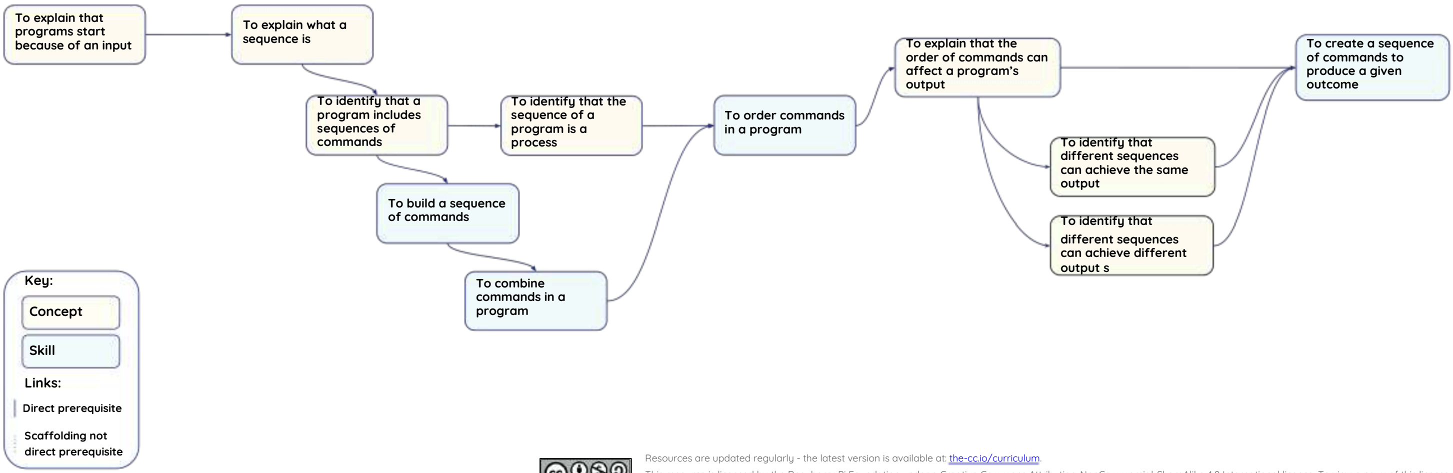




Resources are updated regularly - the latest version is available at: the-cc.io/curriculum.
This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of this license, visit, see creativecommons.org/licenses/by-nc-sa/4.0/

Raspberry Pi Foundation

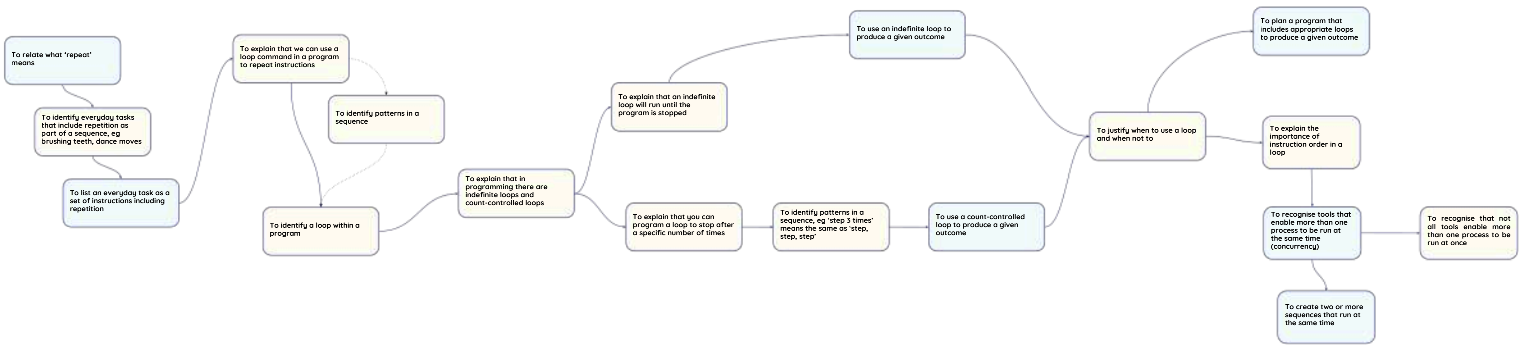
Learning graph
Year 3 – Sequencing sounds



Resources are updated regularly - the latest version is available at: the-cc.io/curriculum
This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of this license, visit, see creativecommons.org/licenses/by-nc-sa/4.0/

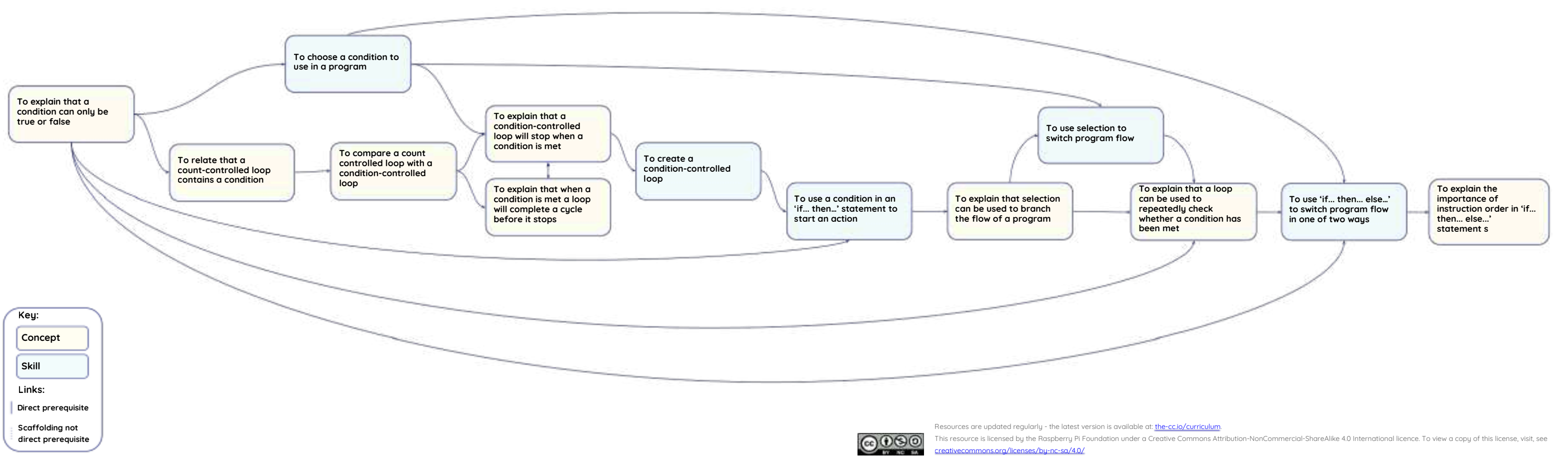
Raspberry Pi Foundation

Learning graph
Year 4 -Repetition in games



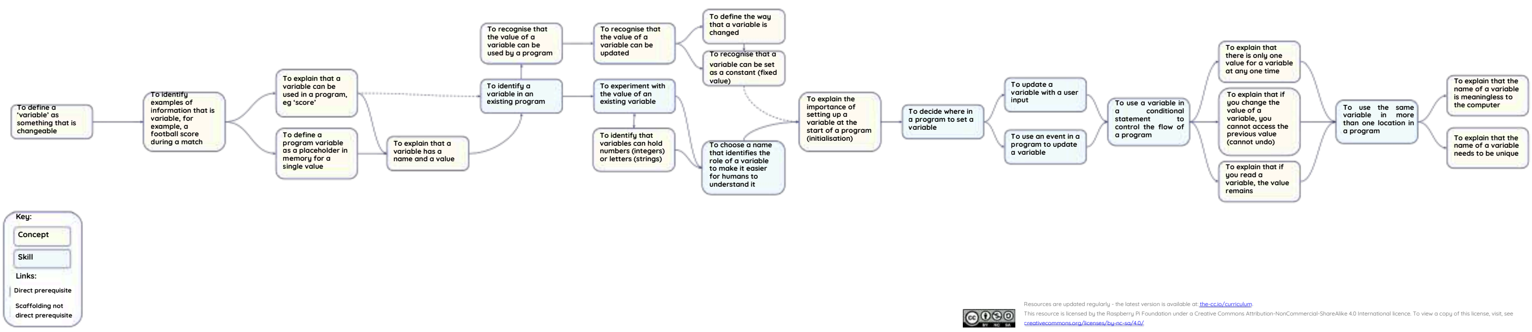
Resources are updated regularly - the latest version is available at: the-cc.io/curriculum.
This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of this license, visit, see creativecommons.org/licenses/by-nc-sa/4.0/

Learning graph
Year 5 –Selection in physical computing



Resources are updated regularly - the latest version is available at: the-cc.io/curriculum
This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of this license, visit, see creativecommons.org/licenses/by-nc-sa/4.0/

Learning graph
Year 6 – Variables in games



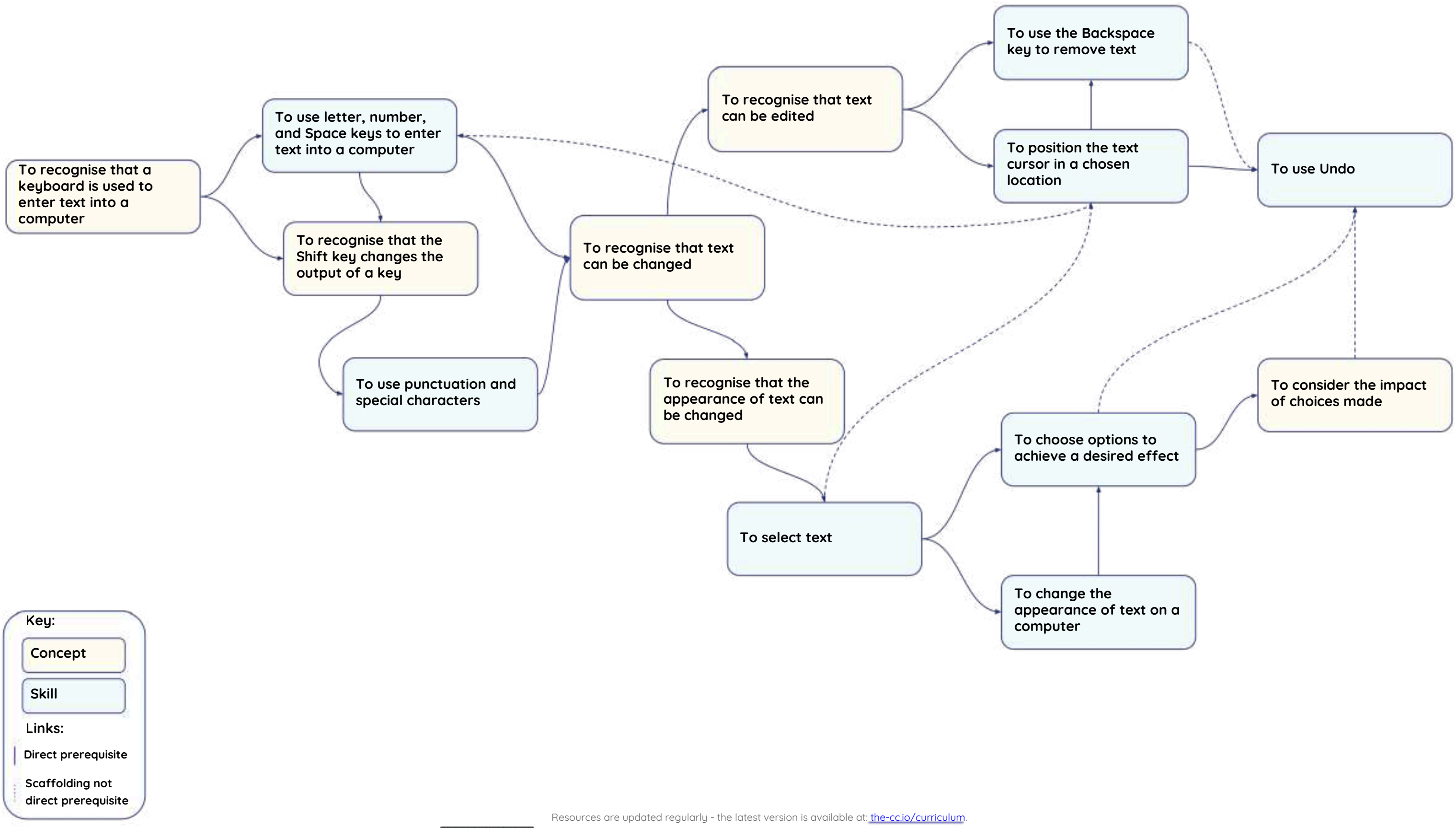
Resources are updated regularly - the latest version is available at: the-cc.io/curriculum.
This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of this license, visit, see creativecommons.org/licenses/by-nc-sa/4.0/.

LEARNING GRAPHS

CREATING MEDIA

Raspberry Pi Foundation

Learning graph
Year 1 – Creating media – Digital writing

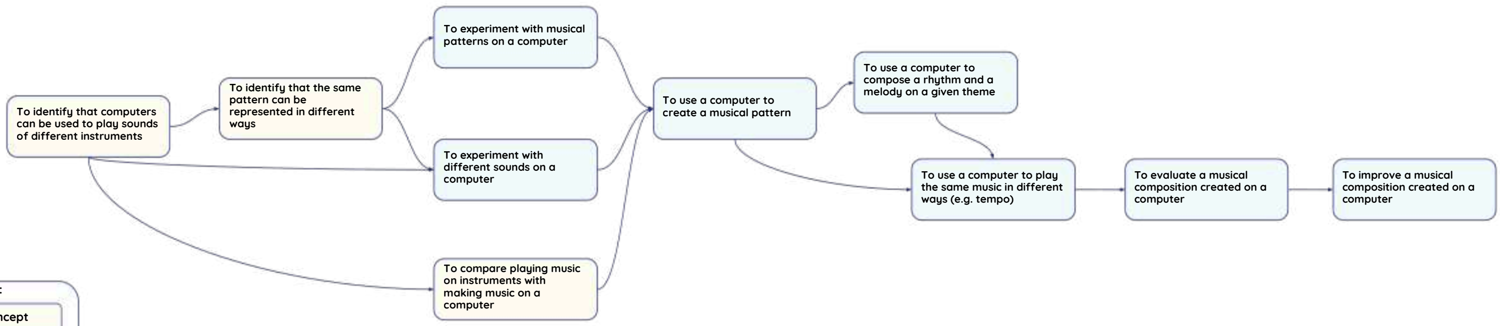


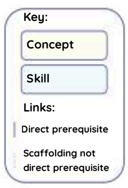
Resources are updated regularly - the latest version is available at: the-cc.io/curriculum.

This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of this license, visit, see creativecommons.org/licenses/by-nc-sa/4.0/

Raspberry Pi Foundation

Learning graph
Year 2 – Creating Media – Making music

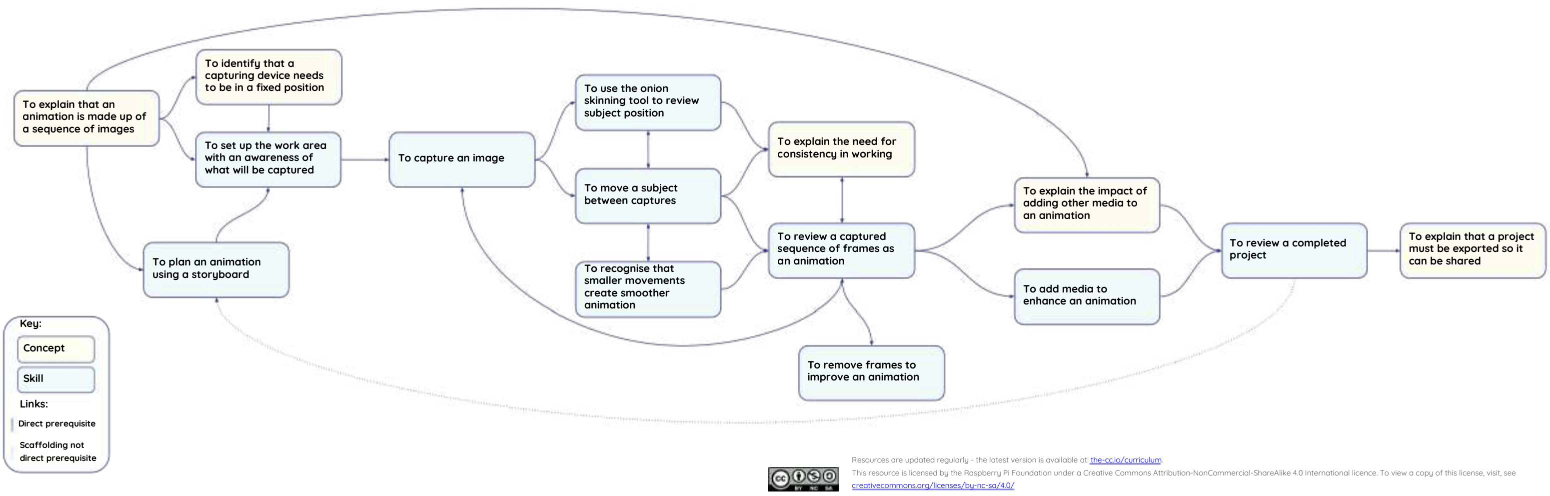




Resources are updated regularly - the latest version is available at: the-cc.io/curriculum.
This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of this license, visit, see creativecommons.org/licenses/by-nc-sa/4.0/

Raspberry Pi Foundation

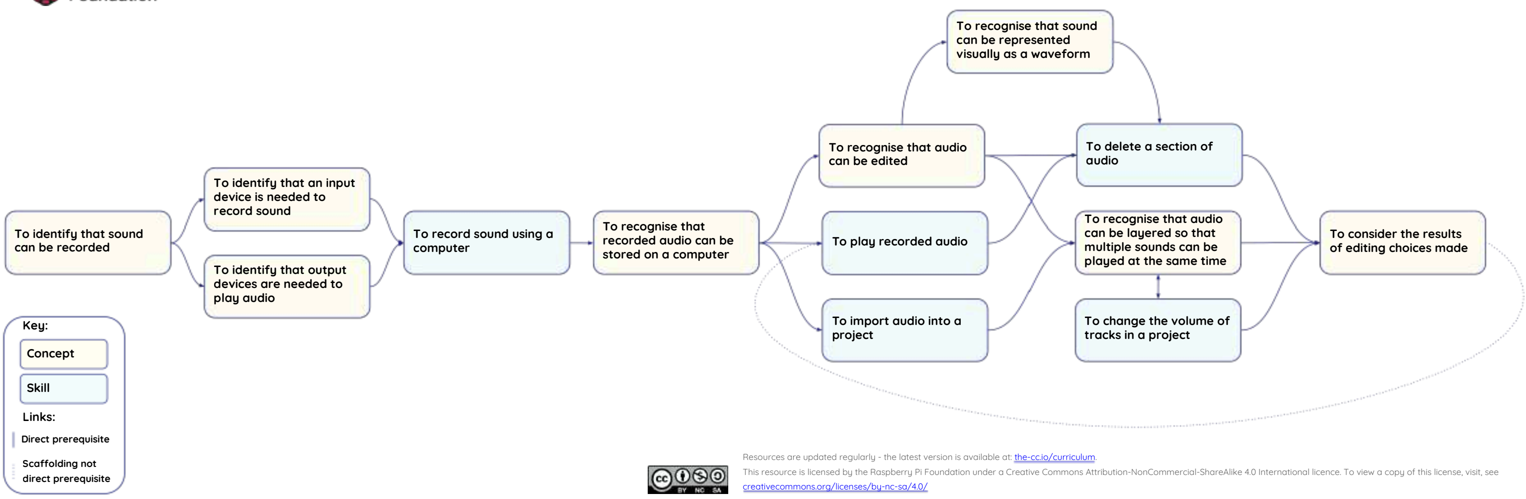
Learning graph
Year 3 – Stop-frame animation



Resources are updated regularly - the latest version is available at: the-cc.io/curriculum.
This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of this license, visit, see creativecommons.org/licenses/by-nc-sa/4.0/

Raspberry Pi Foundation

Learning graph
Year 4 – Creating media – Audio production

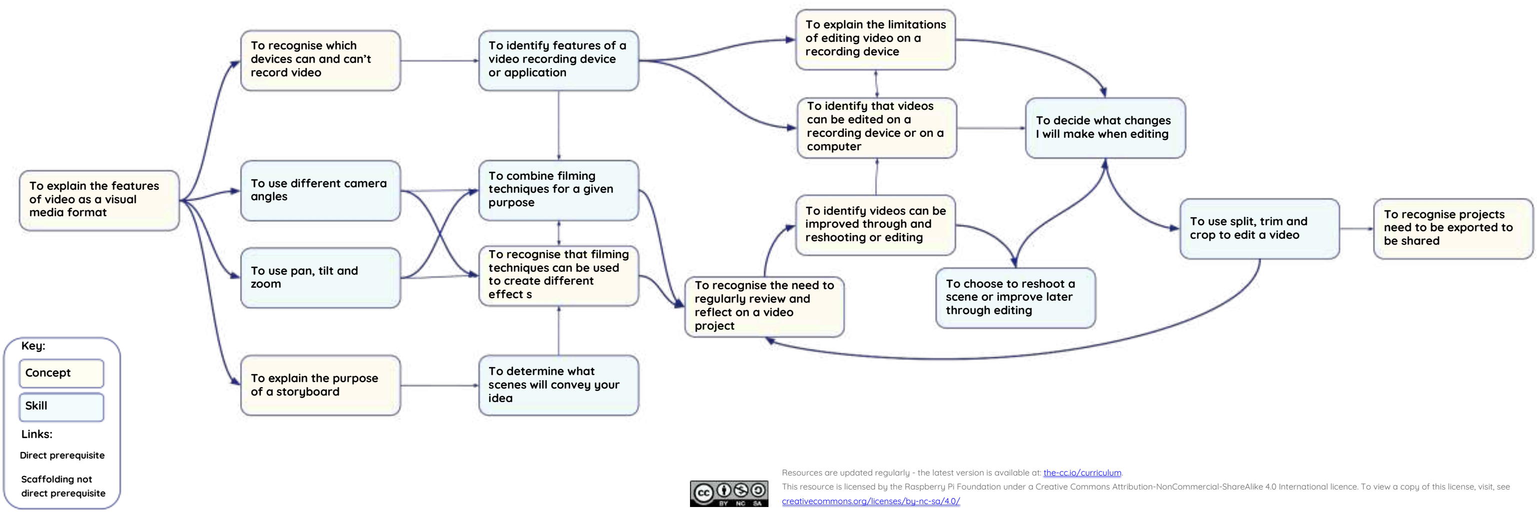


Resources are updated regularly - the latest version is available at: the-cc.io/curriculum
This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of this license, visit, see creativecommons.org/licenses/by-nc-sa/4.0/

Raspberry Pi Foundation

Learning graph

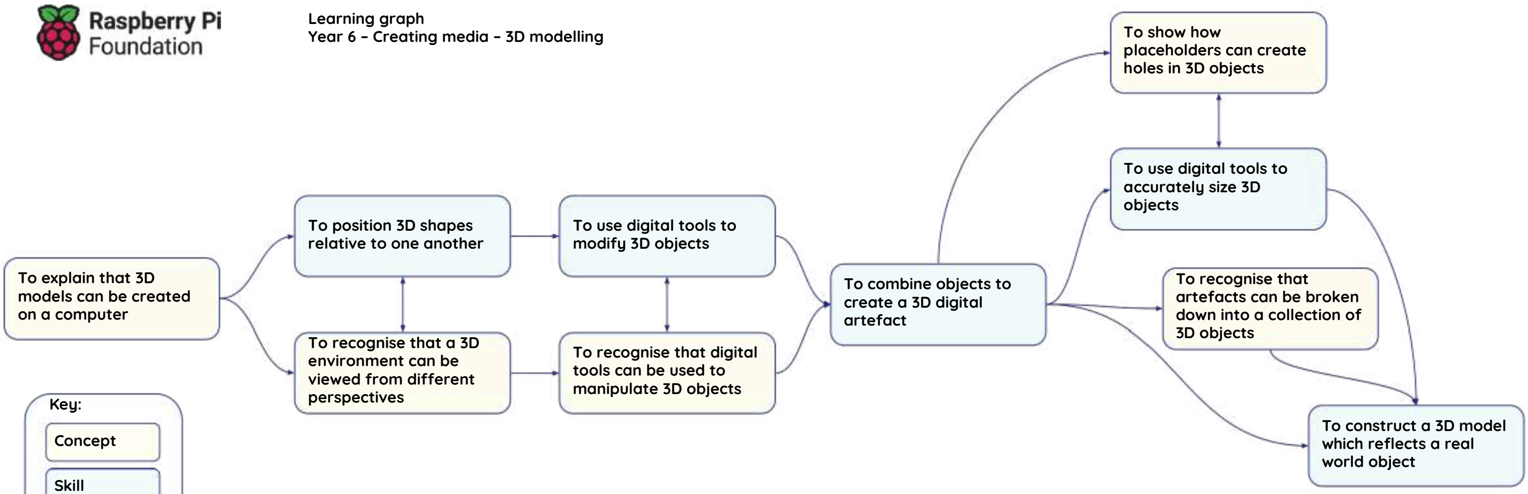
Year5–Creatingmedia – Video production



Resources are updated regularly - the latest version is available at: the-cc.io/curriculum

This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of this license, visit, see creativecommons.org/licenses/by-nc-sa/4.0/

Learning graph
Year 6 – Creating media – 3D modelling



Key:
Concept
Skill
Links:
Direct prerequisite
Scaffolding not direct prerequisite

Resources are updated regularly - the latest version is available at: the-cc.io/curriculum.
This resource is licensed by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of this license, visit, see creativecommons.org/licenses/by-nc-sa/4.0/

Software and hardware overview

Requirements for pupils — below

✓ Used for the unit — reflected in screenshots ● Could be used as an alternative

	Desktop or laptop	Chromebook	Tablet	Software or hardware
1.1 Technology around us	✓	✓	●	paintz.app
1.2 Digital painting	✓	✓	●	Microsoft Paint or similar
1.3 Moving a robot				Bee-Bot, Blue-Bot, or other fixed-movement floor robot
1.4 Grouping data	✓	✓		Google Slides or Microsoft PowerPoint
1.5 Digital writing	✓	✓	●	Google Docs or Microsoft Word
1.6 Programming animations	●	●	✓	ScratchJr
2.1 Information technology around us	✓	✓		Google Slides or Microsoft PowerPoint
2.2 Digital photography	✓		●	Digital camera
2.3 Robot algorithms				Bee-Bot, Blue-Bot, or other fixed-movement floor robot
2.4 Pictograms	✓	✓	●	j2data Pictogram
2.5 Digital music	✓	✓	●	Chrome Music Lab
2.6 Programming quizzes	●	●	✓	ScratchJr

Software and hardware overview

Requirements for learners — below

	Desktop or laptop	Chromebook	Tablet	Software or hardware
3.1 Connecting computers	✓	●	●	Painting program (any)
3.2 Stop-frame animation	●	●	✓	iMotion (app for iOS)
3.3 Sequencing sounds	✓	✓	●	Scratch
3.4 Branching databases	✓	✓	●	j2data Branch and Pictogram
3.5 Desktop publishing	✓	●	●	Canva.com
3.6 Events and actions in programs	✓	✓	●	Scratch
4.1 The internet	✓	✓	✓	Various websites
4.2 Audio production	✓			Audacity
4.3 Repetition in shapes	✓	●	●	FMSLogo
4.4 Data logging	✓	+	+	Data logger and associated software
4.5 Photo editing	✓	●		Paint.NET (for Microsoft Windows)
4.6 Repetition in games	✓	✓	●	Scratch

✓ Used for the unit — reflected in screenshots ● Could be used as an alternative + Data loggers that work with Chromebooks or tablets are available. Check with suppliers.

Software and hardware overview, cont.

Requirements for learners — below

	Desktop or laptop	Chromebook	Tablet	Software or hardware
5.1 Systems and Searching	✓	✓		Google Slides
5.2 Video production	✓	●	●	Microsoft Photos (for Microsoft Windows 10)
5.3 Selection in physical computing	✓	✓		Crumble controller + starter kit + motor
5.4 Flat-file databases	✓	✓	●	j2data Database
5.5 Introduction to vector graphics	✓	●		Google Drawings
5.6 Selection in quizzes	✓	✓		Scratch
6.1 Communication and collaboration	✓	✓		Google Slides
6.2 Webpage creation	✓	✓		Google Sites
6.3 Variables in games	✓	✓		Scratch
6.4 Introduction to spreadsheets	✓	✓	●	Google Sheets or Microsoft Excel
6.5 3D modelling	✓	✓	●	Tinkercad
6.6 Sensing movement	✓	✓	●	micro:bit and Microsoft MakeCode

✓ Used for the unit — reflected in screenshots ● Could be used as an alternative

READING OPPORTUNITIES

✅ For Key Stage 1 (Years 1–2, Ages 5–7)

My First Coding Book – DK
 Interactive lift-the-flap book introducing loops, conditionals, and sequencing without screens.
 Great for unplugged activities and early logic skills.

I Can Code: If/Then – Vicky Fang
 Simple board book teaching basic logic through everyday examples.
 Perfect for Reception and Year 1 learners.

Lift-the-Flap Computers and Coding – Rosie Dickins
 Explains computing concepts in a fun, visual way.
 Ideal for introducing vocabulary and ideas.

✅ For Lower Key Stage 2 (Years 3–4, Ages 7–9)

Hello Ruby: Adventures in Coding – Linda Liukas
 Story-based approach to coding concepts with creative activities.
 Encourages problem-solving and imagination.

Coding for Kids: Scratch – Matthew Highland
 Step-by-step guide to creating games in Scratch.
 Aligns with block-based programming in the Raspberry Pi curriculum.

Help Your Kids with Computer Coding – DK
 Uses Scratch and Python with clear visuals and projects.
 Great for bridging block-based to text-based coding.

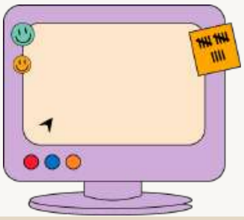
✅ For Upper Key Stage 2 (Years 5–6, Ages 9–11)

Coding Projects in Python – DK
 Visual guide to creating Python projects.
 Supports transition to text-based programming. Hello Raspberry Pi!: Python Programming for Kids and Other Beginners – Ryan Heitz
 Introduces Raspberry Pi setup, Python basics, and fun projects.
 Perfect for physical computing and creative tasks.

Raspberry Pi Projects for Kids – Dan Aldred
 Hands-on projects like LED night-lights, Minecraft mods, and home automation.
 Encourages creativity and links to real-world applications.

✅ Why These Books?

They match progression in the Raspberry Pi Foundation curriculum:
 KS1 → unplugged logic → Scratch → Python → physical computing.
They combine fun, storytelling, and hands-on projects to keep engagement high.
They support computational thinking and digital literacy from early years.



Computing - supporting the provision for pupils with SEND within the Wider Curriculum

SEND Code of Practice (DfE, 2015)
‘ High quality teaching that is differentiated and personalised will meet the individual needs of the majority of children.’

At Harper Bell SDA School, the SEND Code of Practice is at the heart of our curriculum for all areas. For us, this means:

✓ Ensuring that all children are able to make the most of learning opportunities given to them in the Wider Curriculum.

✓ Children are able to develop their knowledge and skills within the Wider Curriculum through appropriate levels of curricular adaptations.

✓ Being cautious that:
- There is not a ceiling being put on children’s learning.
- Wider Curriculum differentiation and scaffolding is not always focused on writing expectations or level of adult support.
- Differentiation by outcome is used minimally - not an effective strategy for long term learning and retention (Nasen Guidance).
- There isn’t a reliance on using worksheet

1. Planning and Preparation
- Know your learners: Review individual education plans (IEPs) and language proficiency levels.
- Set clear, achievable objectives: Break down complex computing concepts into smaller, manageable steps.
- Use visual timetables and lesson outlines: Helps SEND and EAL learners anticipate what’s coming.

2. Language and Instruction
- Simplify language: Use short sentences and avoid jargon. Explain technical terms with visuals.
- Pre-teach key vocabulary: Provide word banks with images (e.g., “algorithm,” “debug”).
- Model instructions: Demonstrate tasks step-by-step before asking pupils to try.

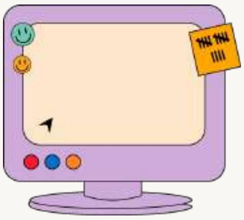
3. Multi-Sensory Approaches
- Visual aids: Diagrams, flowcharts, and icons for coding concepts.
- Physical activities: Use unplugged computing (e.g., sequencing cards) to reinforce logic before coding.
- Audio support: Record instructions or use text-to-speech tools for learners with reading difficulties.

4. Scaffolded Learning
- Chunk tasks: Break activities into smaller stages with checkpoints.
- Provide templates: For coding tasks, give partially completed code or block-based environments (Scratch).
- Offer sentence starters: For EAL learners when explaining algorithms or reflecting on work.

5. Assistive Technology
- Screen readers and magnifiers: For visually impaired learners.
- Speech-to-text tools: For learners with writing difficulties.
- Translation tools: For EAL learners to access instructions in their home language.

6. Collaborative Learning
- Pair work or small groups: Mix abilities so peers can support each other.
- Role allocation: Assign roles like “coder,” “tester,” “explainer” to reduce cognitive load.

Extra Tips
- Incorporate culturally relevant examples for EAL learners.
- Use color coding in code blocks for SEND learners with processing difficulties.
- Provide clear exit points for learners who may fatigue easily.



What general strategies can we use to help support children with SEND to access the Wider Curriculum?

The EEF outlines five key principles for supporting children with SEND in mainstream settings.

Differentiated Learning Objectives:
Learning objectives which are focused on the same learning concept, but are pitched at a different level can be effectively used, for example a lower year group objective or different key stage (depending on a child's level of ability).

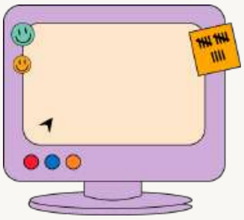
Writing Supports:
The following scaffolds can be used, however this shouldn't be the only source of differentiation for pupils:
- Word Banks to help support spellings of key words
- Using technology to record ideas - for example using Slides to record the different steps in a Science investigation or to create a presentation about a historical time period.
- Reducing writing demands - could verbal responses be scribed or recorded and captured through a QR code?
- Writing Frames to help structure written work

Use of Visuals and Dual Coding:
Visual prompts and Dual Coding (presenting visual and written/verbal prompts together) can help to support children's understanding of learning concepts, task instructions, teacher modelling and independent work. When used effectively, visuals can help to reduce the reliance on a pupil's memory, for example using photographs to help support understanding of a task and the repetition of images to support learning
and recall. Helpful Dual Coding symbols can be found at: https://thenounproject.com/term/code/

Differentiated Knowledge Organisers:
Knowledge Organisers and can be differentiated to help support children's engagement with this resource to deepen knowledge and recall. It could be that such resources draw on differentiated key vocabulary or learning concepts from lower year groups or Key Stages (as appropriate for a child's needs).

Pre-Tutoring:
For certain topics or learning concepts, Pre-Tutoring may be helpful to build children's understanding of key topic words or tasks which involve more complex steps, for example a Science investigation or D.T. project.

Flexible Grouping and Active Learning Methods:
Flexible Grouping is one of the key best-practice approaches for supporting children with SEND, asoutlined by EEF. This approach involves children being grouped in different ways, according to the learning and tasks being focused on. Within the Wider Curriculum, this approach can be used as a layer of support for children with SEND, for example working in a group or partner to undertake a Science investigation or to find out more about a historical event.



What general strategies can we use to help support children with SEND to access the Wider Curriculum?

Preparing pupils for change:
Some Wider Curriculum lessons can be very practical, using novel equipment, with an emphasis on group work. For certain pupils, this can be overwhelming and prior preparation, perhaps through Pre-Tutoring, may be beneficial.

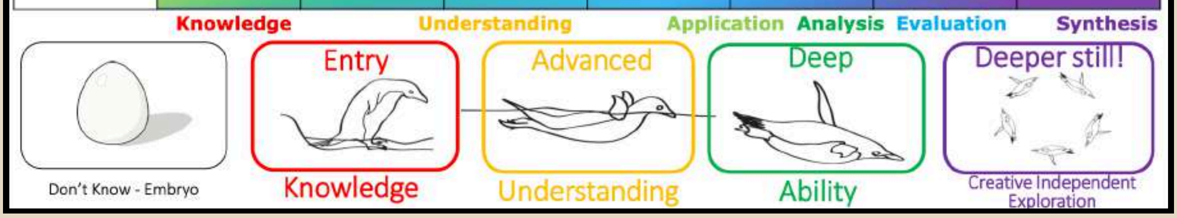
Use of Additional Adult Support:
It is important that this support strategy is not always used within the Wider Curriculum and when used,
is, not just to facilitate task completion.
When adult support is provided, it is important that they:
- Are clear about what Learning Objectives are
- Know the sequence of the lesson
- Understand the lesson content
- Know how to break the lesson into manageable chunks and provide opportunities for independent learning
- Are provided with key questions to help support Formative Assessment
- Are familiar with how ICT can be used to enhance learning, if appropriate

Differentiated Questioning:
Layers of questioning can be used to help support children with SEND's engagement and participation
in whole-class learning opportunities. Consider using the following Question Complexity Grid to help plan specific questions, targeted at certain pupil groups:

Questioning Grid

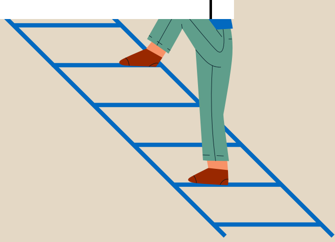
SOCRATIC - DIALOGIC - HIGHER ORDER THINKING

?	Is? (present)	Did? (past)	Can? (possibility)	Should? (opinion)	Would? (probability)	Will? (prediction)	Might? (imagination)
What? (event)	What is?	What did?	What can?	What should?	What would?	What will?	What might?
Where? (location)	Where is?	Where did?	Where can?	Where should?	Where would?	Where will?	Where might?
When? (sequence, location)	When is?	When did?	When can?	When should?	When would?	When will?	When might?
Which? (choice)	Which is	Which did?	Which can?	Which should?	Which would?	Which will	Which might?
Who? (person)	Who is?	Who did?	Who can?	Who should?	Who would?	Who will?	Who might?
Why? (reason)	Why is?	Why did?	Why can?	Why should?	Why would?	Why will?	Why might?
How? (meaning)	How is?	How did?	How can?	How should?	How would?	How will?	How might?



Supporting the provision for pupils with SEND within Computing

Tier 1 Strategies	Tier 2 Strategies	Tier 3 Strategies
• Ensure that key vocabulary is shared and explained with class. Vocabulary is displayed. Practise saying vocabulary together. • Unit markers at start of new topics to support key vocabulary and sticky knowledge questions (with images to support in EYFS/KS1). • Instructions are clear and teachers use appropriate strategies to check understanding throughout the lesson. • Questioning and AFL strategies used throughout the lesson for teacher to check understanding and then address misconceptions where and when they happen. • Use of praise and verbal feedback given throughout lesson to support progress of all learners. • Use of retrieval practise at start of lessons through use of Flashbacks. This will ensure that prior learning is activated and that questions accessed in the Flashback will also support learning that will happen in the coming lesson. • Use of sentence stems to support verbalisation of key concepts. • Use of visual aids to support children with knowledge of artists and works of art. • Use of teacher sketchbooks and modelling, demonstrating and imitating where appropriate. • Allow appropriate time for children to practise and develop skills. • All chn to be encouraged to experiment and explore with a range of media and tools. Provide a range of scales for children to work on. • Ensure that independence is encouraged in terms of selecting materials, but also in tidying equipment away.	• Provide topical word cards and word banks with pictures where necessary to support learners. • Additional staff support (where appropriate) to record key words with the child to support learning. • Use of 'check in' with students who will benefit from use of name to ensure that they know instructions are for them. • Teacher to target and scaffold questioning to support the needs of all learners. • Adaptation of task where appropriate – subject matter of task may be adapted as long as skills are followed. • Allow chn to use tools that enable them to create their art e.g. use of fingers instead of paintbrush, playdoh instead of clay, etc. • Use of concrete resources where needed to support learning of abstract concepts. • Ensure that children who need adapted tools (pencil size, larger / easy grip scissors, etc) have access to these and know where to get them. •	• Pre-teaching of specific scientific vocabulary. • Small group teaching activities (if possible). • Providing students with worked examples to use as a model whilst completing independent work. • Additional adult to scribe/note take to support recording of answers. • Allow additional time for children to practise skills and build and develop fine motor skills e.g. use of modelling clay / playdough



Our journey so far 2025

1. New IT support has been sought for the school from Bishop Challenor.

2. There is a clearer vision for IT from an action plan baseline uindertken with DRB Ignite MAT Support.

3. DFE Connect the Classroom funding is being used to improve the network and wifi speed.

4. IT resources have been reconfigured to support apps for computing which has not been taught for some years. Micro:bits have been purchased.

5. The HT has taken leadership in the interim to ensure strategic oversight of the resources required.

6. STEM funding has been used for working with a Subject Matter Expert National Centre for Computing Education to discuss the best way forward with limited resources.

7. Raspberry Pi Foundation lesson plans and curriculum have been mapped out ensuring NC coverage

8. Close the gap focus days are being organised to bring pupils up to speed to relaunch the curriculum as appropriate to age.

9. Training and support for staff has been built in to the relaunch of the subject

10. Initially, the curriculum is able to be taught from ipads as the only resource we have. This can be upgraded as IT resources are invested in.

So what?

Raspberry Pi has been introduced as a curriculum mapping tool .

To ensure that the National Curriculum is planned for and have designed our units to allow children to build schemas. This will support children in being able to know more and do more.

Key domains and knowledge threads have been selected and mapped out

To ensure that pupils gain declarative and procedural knowledge in small sequenced steps. Teachers are clear about what to teach and when and know where their lesson fits in the wider schema.

Computing is now being taught each week ensuring compliance and coverage.

Opportunity to dedicate the learning time to the subject and to practise skills over time. Compliance and coverage is ensured.

Progression coverage checked for compliance

Progression should cover three pillars:

- Computer Science (core concepts like algorithms, data, programming)
- Information Technology (creating digital content, using tools)
- Digital Literacy (safe, responsible, effective technology use)

Progression coverage checked for compliance

Programming in Primary

- Programming is central and appears throughout the curriculum.
- Pupils need structured progression:
  - Start with algorithms and simple programs.
  - Develop understanding of sequence, selection, repetition.
- Avoid rushing into coding without conceptual understanding to prevent misconceptions.

So what?

MTPs are being created to support prior learning and next steps .

This ensures teachers are clear about what they are teaching and the prior knowledge the children should to both check retention and build upon this.

Flasbacks are emerging into regular practice

Deeper learning happens when we transfer information from our short-term memory stores into our long-term memory stores. By planning for and providing retrieval practice we support this process because it requires pupils to recall previously-learned information.

The profile of computing has been raised across the school

This ensures a broad and balanced curriculum and pupils can explore both hobbies and career pathways which can come from their enjoyment of the subject. Pupil voice shows our pupils appreciate the mental health benefits of the subject.

Staff CPD has been held on the use of Raspberry Pi planning- this aids teacher develpment

Staff understand the purpose and support available in the planning documents to enable them to see the bigger picture, access training and get support with assessment

Next steps

Reading material to support curriculum to be purchased and made available

Development of IT resources to further support opportunities and variety of hard/software - see 5 year plan

Specific work on development of AI and future tech within the Computing curriculum and other areas from policy to embedding

See plan for regular CPD for all staff to ensure high quality teaching over 5 year development

Detailed bridgeing activities to support transition from EYFS to KS1

New staff from September so assessment of knowledge base and training impact to be part of action plan

Ensure that transition between EYFS and KS1 is progressive

Consider how STEM themed opportunities can enhance the curriculum

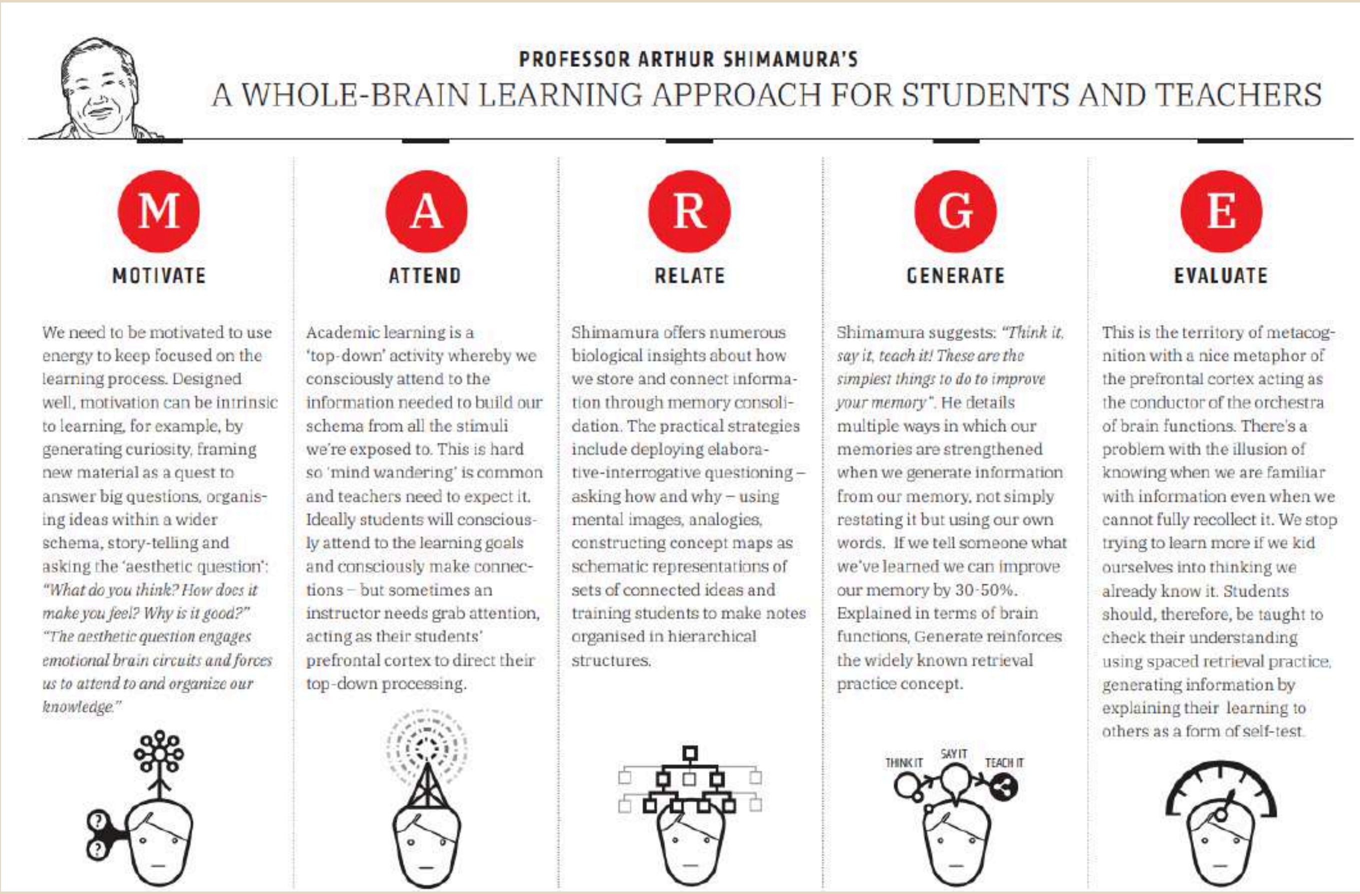
Review and adapt vocabulary progression document to align with the needs of the pupils.

To look at opportunities to further embed digital safeguarding within the Computing curriculum and beyond

Develop cross curricular resources for Art and resources which can be accessed outside of Art lessons.

How we support the development of schemas in Computing

Relate - linking to prior knowledge. This is what we are working so hard to create in our curriculum - the links to previous knowledge for children to "pin" their new knowledge on to.
Generate - generating information from our memories helps to consolidate knowledge.
Evaluate - checking the understanding of children and encouraging them to check their own understanding. A good way to check is to see if they can explain their knowledge to others for example.



PROFESSOR ARTHUR SHIMAMURA'S

A WHOLE-BRAIN LEARNING APPROACH FOR STUDENTS AND TEACHERS

M — MOTIVATE

We need to be motivated to use energy to keep focused on the learning process. Designed well, motivation can be intrinsic to learning, for example, by generating curiosity, framing new material as a quest to answer big questions, organising ideas within a wider schema, story-telling and asking the 'aesthetic question': *"What do you think? How does it make you feel? Why is it good?" "The aesthetic question engages emotional brain circuits and forces us to attend to and organize our knowledge."*

A — ATTEND

Academic learning is a 'top-down' activity whereby we consciously attend to the information needed to build our schema from all the stimuli we're exposed to. This is hard so 'mind wandering' is common and teachers need to expect it. Ideally students will consciously attend to the learning goals and consciously make connections – but sometimes an instructor needs grab attention, acting as their students' prefrontal cortex to direct their top-down processing.

R — RELATE

Shimamura offers numerous biological insights about how we store and connect information through memory consolidation. The practical strategies include deploying elaborative-interrogative questioning – asking how and why – using mental images, analogies, constructing concept maps as schematic representations of sets of connected ideas and training students to make notes organised in hierarchical structures.

G — GENERATE

Shimamura suggests: *"Think it, say it, teach it! These are the simplest things to do to improve your memory"*. He details multiple ways in which our memories are strengthened when we generate information from our memory, not simply restating it but using our own words. If we tell someone what we've learned we can improve our memory by 30-50%. Explained in terms of brain functions, Generate reinforces the widely known retrieval practice concept.

E — EVALUATE

This is the territory of metacognition with a nice metaphor of the prefrontal cortex acting as the conductor of the orchestra of brain functions. There's a problem with the illusion of knowing when we are familiar with information even when we cannot fully recollect it. We stop trying to learn more if we kid ourselves into thinking we already know it. Students should, therefore, be taught to check their understanding using spaced retrieval practice, generating information by explaining their learning to others as a form of self-test.