

EYFS - Reception

The children will be working toward the Early Learning Goals – please see EYFS Curriculum page

Understanding the World	• Talk about the lives of people around them and their roles in society, including how technology is used in everyday life. • Explore similarities and differences between their own lives and those of others, including the use of technology in different communities and cultures.
Physical Development	• Develop the control needed to use touchscreens, keyboards, a mouse, cameras, and programmable toys safely and accurately.
Communication and Language	• Follow instructions, explain how technology works, ask questions, and use new vocabulary related to digital devices.
Personal, Social and Emotional Development	• Learn to use technology responsibly, take turns, care for equipment, and understand simple online safety rules with adult guidance.
Mathematics	• Develop sequencing, pattern recognition, positional language, and logical thinking through coding games and programmable toys.

	Sunflowers		Honeysuckle		Forget Me Not	
	KS1		LKS2		UKS2	
Computing Systems and Networks	Technology around us	IT around us	Connecting Computers	The Internet	Systems and Searching	Communication and collaboration
	Choose a piece of technology to do a job. Show how to use technology safely. Explain that technology is something that can help us – give examples of how. Identify: - examples of technology and how technology helps us. - the main parts of a computer (i.e. mouse, keyboard) - be able to use them and edit text. Understand why rules are needed when using technology. Recognise that: - a computer is an example of technology; - choices are made when using technology. - some technology can be used in different ways.	Describe some uses of computers. Identify information technology in and beyond school. Show how to use information technology safely. Say how rules for using information technology can help us. Explain how information technology benefits us. Identify that a computer is a part of information technology. Recognise: - different types of computers used in school; - features and uses of information technology; - that choices are made when using information technology.	To explain: - that a computer system accepts an input and processes it to produce an output; - how a computer network can be used to share information; - the role of a switch server, and wireless access point in a network. Identify: - input and output devices; - networks devices around me and how networks can be connected to other networks. Describe what an input is. Identify the benefits of computer networks Identify how: - changing the process can affect the output; - devices in a network are connected with one another Explain: - that a process acts on the inputs;	Outline how information can be shared via the World Wide Web (www). Evaluate the reliability of content and the consequences of unreliable content. Describe: - how networks connect to other networks. - how to access the www; - the types of content/media that can be added, created, and shared on the www; - the current limitations of www media. Explain: - how the content of the www is created, owned, and shared by people; - that the global interconnection of networks is the internet. - the benefits of the www. Recognise: - that the www is part of the internet; - the need for security	Describe the input and output of a search engine. Demonstrate that different search terms produce different results. Evaluate the results of search terms Describe the role of a particular IT system in their lives Relate that search engines are examples of large IT systems. Recognise: - that a system is a set of interconnected parts which work together; - inputs, processes, and outputs in large IT systems. Explain: - that computers can be connected together to form IT systems; - why search engines create indices, and that they are different for each search engine; - that ranking orders search results to make them more	Outline methods of communicating and collaborating using the internet. Choose methods of internet communication and collaboration for given purposes. Evaluate different methods of online communication and collaboration. Decide what you should and should not share online Discuss the opportunities that technology offers for communication and collaboration. Recognise that: - data is transferred across networks using agreed protocols (methods); - connections between computers allow access to shared stored files; - computers connected to the internet allow people in different places to work together.

			- that an output is produced by the process; - how computer systems can change the way that we work; - how information is passed through multiple connections Recognise that: - a digital device is made up of several ports; - computers can be connected to each other; - a network is made up of a number of components.	on the internet.	useful; - how ranking is determined by rules, and that different search engines use different rules; - why the order of results is important and to whom; - how search engines make money by selling targeted advertising space; - the role of web crawlers in creating an index; - how search results are selected. Identify that: - that data can be transferred between IT systems; - some of the limitations of search engines.	Explain: - that data is transferred in packets; - which types of media can be shared through the internet; - that communicating and collaboration using the internet can be public or private.
--	--	--	--	------------------	--	---

Creating Media	Digital Painting	Digital Photography	Stop-frame animation	Audio Production	Video production	Webpage creation
	Create a picture using freehand tools.	Capture a digital image / photograph on digital devices in both landscape and portrait format.	Plan an animation using a story board.	Record sound using a computer.	Identify features of a video recording device or application.	Review an existing website (navigation bars, header).
	Use a range of paint colours and art tools when precision is needed (i.e. shape, line, colour).	Hold a camera still (and use zoom) to take clear photographs.	Set up the work area with an awareness of what will be captured.	Play recorded audio.	Combine filming techniques for a given purpose.	Create a new blank web page.
	Use the undo button to correct a mistake.	Consider the lighting and use filters to edit photographs.	Capture an image.	Import audio into a project.	Determine what scenes to use to convey ideas.	Add text to a web page.
	Combine tools to create artwork.	To decide which photographs to keep or improve by retaking.	Use the onion skinning tool to review subject position.	Delete a selection of audio.	Decide what changes I will make when editing.	Set the style of text on a web page.
	Explain what different freehand tools do and decide when it is appropriate to use each tool.	Recognise:	Move subject between capture.	Change the volume of tracks in a project.	Choose to reshoot a scene or improve later through editing.	Change the appearance of text.
	Recognise that computers can be used to create art.	that digital devices can capture images using a camera and that these photos can be saved and viewed later.	Review a captured sequence of frames as an animation.	Consider the results of editing choices made.	Use:	Embed media in a web page.
	Decide when it's appropriate to use each tool – consider impact of choices made.	features of a 'good' photograph.	Add / remove media to enhance animation.	Identify that:	- different camera angles; - pan, tilt and zoom; - split, trim and crop to edit a video.	Add web pages to a website.
	Compare painting using a computer with using brushes.	- that some images are not accurate. - that photographs can be changed after they have been taken.	Identify that a capturing device needs to be in a fixed position.	- sound can be recorded; - an input device is needed to record sound; - output devices are needed to play audio.	Explain the:	Preview a web page (different screen sizes).
		Identify how to improve, retake and change a photograph.	Recognise that smaller movements create smoother animations.	Recognise that:	- features of video as a visual media format; - purpose of a storyboard; - limitations of editing video on a recording device.	Insert hyperlinks between pages / to another site

			up of a sequence of images.		review and reflect on a video project; - projects need to be exported to be shared. Identify: - that videos can be edited on a recording device or on a computer; - videos can be improved through and reshooting or editing.	pages (different screens / devices); - the need for a navigation path; - the implications of linking to content owned by others.
	Digital Writing To select, position and change the appearance of text to achieve a desired effect. To use: - a range of keys (i.e. letter, number, space, backspace and punctuation) to enter / remove text; - undo. To recognise that: - a keyboard is used to enter text into a computer; - the shift key changes the output of a key; - text can be changed – visual appearance and editing.	Digital Music Experiment with musical patterns and different sounds on a computer. Evaluate and improve a musical composition created on a computer. Use a computer to: - compose a rhythm and melody based on a theme; - play the same music in different ways (i.e. tempo). Identify that computers can be used to play sounds of different instruments. Identify that the same pattern can be represented in different	Desktop Publishing Change page orientation. Organise, add and remove text and image to and from placeholders. Edit text and images, including moving and resizing. Choose fonts and apply effects to text. Review a document Define landscape and portrait as two different page orientations. To recognise: how text and images can be used together to convey	Photo Editing To use an application to change the whole and part of a digital image Change the composition of an image (rotate, flip, arrange, crop or cut) Apply a change globally (adjust colours, apply filters, add effects) Apply changes locally (adjust colour, retouch, reuse) Make additions (draw, add text, add an element) Use clone, copy, and paste to change the composition of a digital image Use cloning to retouch a digital	Introduction to Vector Graphics Add an object to a vector drawing. Select, duplicate, modify, reposition and delete objects. Move objects between layers of a drawing. Group and ungroup selected objects. To identify that a vector drawing comprises separate objects; To explain how alignment and size guides can help create a more consistent drawing. To consider the impact of	3D Modelling Position 3D shapes relative to one another Combine objects to create a 3D digital artefact Construct a 3D model which reflects a real world object To use digital tools to: - modify 3D objects - accurately size 3D objects Explain that 3D models can be created on a computer Show how placeholders can create holes in 3D objects To recognise that

		ways. Compare playing music with instruments to making music on a computer.	- information; - that DTP pages can be structured with placeholders; - how different font styles and effects are used for particular purposes; To consider: - how different layouts can suit different purposes; - the benefits of using DTP applications.	image Add text to a digital image Choose the most appropriate tool for a particular purpose Consider the impact of changes made on the quality of the image. To recognise that digital images can be manipulated and changed for different purposes;	choices made. To recognise that: - each object in a drawing is in its own layer; - vector images can be scaled without impact on quality - objects can be modified in groups.	- a 3D environment can be viewed from different perspectives; - digital tools can be used to manipulate 3D objects; - artefacts can be broken down into a collection of 3D objects.
Programming	Moving a robot Understand that programs are made up of a sequence of commands Follow simple instructions and sequences of commands. Match commands to their expected outcomes. Predict the outcome of individual commands and sequences of commands. Create and order commands to achieve a desired goal. Use movement and turn commands to control a programmable device or robot. Debug simple programs by	Robot Algorithms Follow and give clear, sequenced instructions. Create simple algorithms to achieve a specific goal. Use a sequence of commands to program a floor robot. Predict the outcome of algorithms and programs. Compare predicted outcomes with actual program behaviour. Plan and identify different routes or solutions to a problem. Design and test a simple navigation mat or environment. Translate algorithms into	Sequencing Sounds Identify and use sprites, backdrops, and programming blocks within Scratch. Create programs by following a planned design. Select appropriate commands to control the actions of sprites. Build sequences of connected programming blocks. Run programs using different start events. Create and organise sequences of actions, including sound and movement. Plan algorithms to achieve a specific task or outcome.	Repetition in Shapes Write simple programs using a text-based programming language. Create code snippets to achieve a specific purpose. Design and test algorithms before implementing them in code. Apply variables and values to control program behaviour. Recognise patterns and identify opportunities to use repetition. Use count-controlled loops to create efficient programs. Predict and evaluate the behaviour of programs	Selection in Physical Computing Build and connect simple electronic circuits to a microcontroller. Program output devices, such as LEDs, using text-based code. Apply count-controlled loops to automate hardware outputs. Use inputs and conditional statements to control program flow. Design algorithms that respond to real-world events and conditions. Integrate inputs and outputs to create interactive physical computing systems.	Variables in Games Identify and use variables to store and manage data within programs. Create, name, and update variables appropriately for their purpose. Apply variables to control and influence program behaviour. Use events to initialise and modify variable values. Design algorithms that incorporate variables to solve problems. Plan and develop interactive programs or games using variables. Create and select appropriate

identifying and correcting errors.	executable programs.	Make appropriate design choices for sprites, artwork, and other project assets.	containing loops.	Plan and communicate solutions through annotated designs and algorithms.	digital assets to support program design.
Explain what a program is designed to do.	Test, evaluate, and refine programs by identifying and correcting errors.	Translate a design or algorithm into executable Scratch code.	Decompose problems into reusable procedures or sub-programs.	Predict and explain the behaviour of programs controlling physical devices.	Test programs to ensure variables function as intended.
Recognise that different sequences of commands can achieve the same result.	Explain the purpose and effectiveness of algorithms and programs.	Test, evaluate, and refine programs to ensure they behave as intended.	Integrate loops and procedures to develop more complex programs.	Test and debug physical computing projects to achieve intended outcomes.	Extend programs by using variables to add new features and functionality.
Plan, compare, and evaluate different solutions to the same problem.			Refine and debug text-based programs to improve efficiency and functionality.	Evaluate and refine physical computing solutions to improve functionality.	Evaluate and refine programs to improve gameplay, usability, and performance.
Develop simple algorithms to solve navigation or movement tasks.					
Programming animations	Programming quizzes	Events and actions in programs	Repetition in games	Selection in quizzes	Sensing movement
Compare different programming tools and identify their features.	Run and control simple programs using appropriate start events.	Select appropriate events and inputs to control program actions.	Analyse algorithms and predict the behaviour of code containing repetition.	Identify, modify, and apply conditions within programs.	Apply programming knowledge in a physical computing environment.
Create programs using start blocks and connected sequences of commands.	Predict and modify the outcome of sequences of programming commands.	Program character movement using suitable commands.	Select appropriate loop structures to solve different programming tasks.	Use selection (if...then...else) to create programs with multiple outcomes.	Program and test applications using emulators and controllable devices.
Select and use appropriate movement commands to control sprites.	Select and combine programming blocks to create sequences.	Choose appropriate sprites and adjust their properties to suit a design.	Modify and adapt loops to achieve a desired outcome.	Design algorithms and program flow using conditional branching.	Use variables, conditions, and selection to control program flow.
Modify values in blocks and predict how changes affect program outcomes.	Create algorithms to control the actions of sprites.	Use programming blocks and extensions to add functionality.	Reuse and adapt existing code to improve program development.	Incorporate user input to influence program behaviour.	Apply comparison operators to create complex decision-making processes.
Add, remove, and manage multiple sprites within a project.	Design projects by selecting appropriate backgrounds, characters, and images.	Build sequences of commands to create interactive programs.	Apply repetition to control multiple objects or processes within a program.	Plan program structure using appropriate design methods.	Integrate physical inputs to produce responsive programs.
Plan and create algorithms for	Translate design ideas into	Add and customise features using appropriate controls and inputs.	Evaluate the effectiveness and efficiency of repeated	Implement algorithms as structured programs.	Design algorithms and program flow for physical computing
				Test programs to ensure they	

	individual sprites. Build programs by converting algorithms into block-based code. Test, evaluate, and refine programs by identifying and correcting errors. Choose suitable sprites and artwork to match the intended design and purpose of a project	block-based programs. Build programs that match a planned design. Test programs to check they meet the intended outcome. Debug programs by identifying and correcting errors. Evaluate and improve projects by refining designs and adding new features.	Match code to its intended outcome and explain program behaviour. Translate a design into an executable program. Test and modify programs to meet design requirements. Evaluate projects and justify design decisions to improve functionality.	sequences. Explain how changes to loops affect program behaviour. Design programs that make purposeful use of repetition. Build programs by implementing and refining original designs. Evaluate and improve algorithms to create more effective solutions.	meet their intended purpose. Share and review programs to gather feedback. Extend programs by adding new features and functionality. Evaluate and refine programs to improve performance and user experience.	projects. Select and manage appropriate variables to support program functionality. Implement complete physical computing solutions from a planned design. Test and debug programs using a range of systematic approaches. Evaluate and refine physical computing projects to improve performance and reliability.
--	--	---	---	--	---	---

Data and information	Grouping Data	Pictograms	Branching Databases	Data Logging	Flat-file Databases	Introduction to Spreadsheets
	Collect simple data and show that collected data can be counted. Identify similarities (attributes) of an object; describe properties of an object and group objects (based on commonality) to answer questions. Identify that objects can be counted. Recognise that information can be presented and in different ways.	Recognise that people, animals and objects can be described by attributes Enter data onto a computer. Use a computer to view data in different forms. Use pictograms to answer single attribute questions. Use a computer to answer comparison questions (tables, graphs). Compare objects that have been grouped by attribute. Construct (complete) a given comparison question. Suggest heading for tally charts and pictograms. Understand how some information should not be shared. To use a: - tally chart to collect data. - computer program to present information in different ways.	Create questions with yes/no answers Choose questions that will divide objects into evenly sized subgroups Repeatedly create subgroups of objects Identify an object using a branching database Retrieve information from different levels of the branching database Investigate questions with yes/no answers. Identify attributes that you can ask yes/no questions about. Select an attribute to separate objects into two similarly sized groups. Recognise that a data set can be structured using yes/no questions. Relate two levels of a branching database using AND. Suggest real-world applications for branching databases. To explain that a: - branching database is an identification tool; - well-structured branching database	Export information in different formats. Choose how often to automatically collect data samples. To use: - a computer to sort data by one attribute; - a set of logged information to find information; - a digital device to collect data automatically; Suggest questions that can be answered using a table of data. Recognise that a sensor can be used as an input device for data collection. Explain that a data logger captures 'data points' from sensors over time. To identify: - data that can be logged over time; - that sensors are input devices.	Select an appropriate graph to visually compare data Ask questions that need more than one attribute to answer. To choose: - different ways to view data; - multiple criteria to search data to answer a given question (AND and OR); - suitable ways to present information to other people. To choose which attribute: - to sort data by to answer a given question; - and value to search by to answer a given question (operands). To explain that: - a computer program can be used to organise data; - tools can be used to select data to answer questions; - computer programs can be used to compare data visually; - we present information to communicate a	Calculate data using a formula for each operation. Choose suitable ways to present spreadsheet data. To use: - functions to create new data; - existing cells within a formula. Identify questions that can be answered using spreadsheet data. Evaluate results in comparison to the question asked. To explain: - what an item of data is in a spreadsheet; - outline that there are different software tools to work with data; - how the data type determines how a spreadsheet can process the data; - that formulas can be used to produce calculated data; - why data should be organised in a spreadsheet. To recognise - that a cell's value

			will enable you to identify objects using		message. To outline how: • ordering data allows us to answer some questions; • operands can be used to filter data; • AND' and 'OR' can be used to refine data selection.	automatically updates when the value in a linked cell is changed; • cells can be linked.
--	--	--	---	--	---	---

End Points – National Curriculum	
Key Stage 1	- 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	- 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.