Accrington Benjamin Hargreaves Church of England Primary School



Our vision is the centre of everything we do. It underpins our teaching and learning, interactions, and environment.

Document Title

Computing Curriculum Progression

2026-2027

Vertical Strands

1) Digital Literacy

2) Software

3) Computer Science

	Unit Title	Introduction to Purple Mash	Making Beats	Coding	Coding
Cl.2	**Disciplinary Knowledge**	• To know it is important to log in to a site, the importance of keeping passwords safe and the need to log out at the end of a session. • To know an avatar is a virtual representation of a person suitable for use online. • To know the 2Do system allows teachers to assign tasks to children within Purple Mash. • To know online sites have a main page called the homepage. • To know online sites often use an alert system to communicate with the user. • To know how to move to a different activity in Purple Mash, you must first close the current activity. • To know many online sites, including Purple Mash, have an area for an individual's work that is	• To know there are differences between music played using musical instruments and digitally created Music. • To know digital tools such as 2Explore and 2Beat can be used to compose music. • To know digital music tools can be used to change aspects of the composition. • To know digital music tools can be used to create the sound of multiple instruments at once. • To know different digital tools are better for different purposes.	• To know computer programs work by following instructions called algorithms. These are written as computer code that the computer can interpret. • To know that in 2Code, code is created using coloured code blocks. • To know code view is the place in 2Code where you see and use the blocks of code. • To know each single instruction such as 'Object Right' is called a command. • To know to make an algorithm happen you must execute, or run, the code. • To know an event is something that makes a block of code run in response to an action such as a user pressing a key or clicking a screen.	• To know that in computing, a set of instructions is known as an algorithm. • To know steps in an algorithm must be followed to achieve the intended outcome. • To know the collision detection event is a way to cause an action when two objects collide with each other. • To know programs follow a sequence of instructions (commands) in order. • To know timer commands can be used to make parts of the program run after a set time. • To know different object types in 2Code have different attributes that can be changed. • To know event commands in 2Code are used to create blocks of code that are run when an event happens.

		accessible only to the individual (and in Purple Mash to their teacher as well). • To know how to access Purple Mash programs, you use the Tools area. • To know you can access non-visible parts of a screen using scrolling. • To know you can use a physical or on-screen keyboard to type upper and lower-case letters and spaces.		• To know debugging is the name for fixing code that isn't working how it was designed to work. • To know the look of a program in 2Code is created in the Design View using backgrounds and objects. • To know program design is the first stage to making a well thought out program.	• To know there are different event command blocks in 2Code. • To know debugging is the process of testing that a program works as intended, looking for any bugs and fixing the problems when found.
	Procedural Knowledge	• Access Purple Mash from home and school. • Log out of Purple Mash. • Give reasons why it is important to keep a password safe and not share it with other people. • Make and edit their own avatar. • Open 2Dos. • Save 2Dos. • Hand in 2Dos and communicate with their teacher via the 2Do. • Access the Purple Mash homepage when on the site. • Access alerts within Purple Mash. • Close activities in Purple Mash. • Access their work area. • Save work in their work area. • Locate and open work they have done previously in their work folder.	• Identify differences and similarities between music played using instruments and digitally composed music. • Compose a melody using 2Explore. • Select sounds from the available libraries. • Compose a beat using 2Beat. • Use digital music tools to change the tempo, volume, looping and length of music compositions. • Use 2Beat to compose an interaction of different instrument sounds. • Select the best music tool for their compositions.	• Create instructions in the form of simple algorithms with attention to the order and the level of detail. • Interpret what a piece of code means. • Recognise object code blocks in 2Code are light blue. • Recognise action code blocks in 2Code are dark blue. • Recognise event code blocks in 2Code are green. • Recognise output code blocks in 2Code are purple. • Switch to code view. • Code blocks are dragged into the coding area to create commands. • Understand how code blocks fit together to create a command.	• Follow a written algorithm. • Identify required events, objects and actions from a program algorithm. • Identify the collision detection block within the event category blocks. • Use a collision detection block in a program. • Assign two objects within the collision detection command. Assign an event for when the two objects collide. • Run code and see the sequential effect that occurs. • Identify the timer command within the control category blocks. • Use the time after command and set a specified number of seconds.

		• Open a specified tool. • Scroll up and down and from side to side where applicable. • Type upper and lower-case letters and spaces using the device available.		• Make a simple command in 2Code by using an object and action together. • Execute the code to see the effect by clicking Run. • Use the 'When Clicked' event code block. • Position a command inside a 'When Clicked' event code block. • Give an object an action that occurs when it is clicked. • Test 'When Clicked' events. • Begin to use logical reasoning to find where bugs in the code are. • Fix bugs in code. • Test whether bugs have been fixed. • Switch to design view. • Choose background images. • Add objects and modify some attributes such as scale. • Plan what objects in a scene will do. • Recognise that this is the algorithm for the program. • Use own design to code a program. • Debug the program against the design specifications.	• Nest actions within the timer command to delay their running. • Use a variety of object types. • Identify attributes which appear under some objects and not in others e.g. a turtle type and a character type. • Alter attributes of an object, such as the image and scale within Design View. • Recognise the event command blocks – When Key Event, When Swiped Event, When Clicked Event and Collision Detection. • Use a variety of event blocks in different circumstances. • Use specific event commands for specific objects, such as a click event for a button object. • Use logical reasoning to identify the source of bugs and attempt to fix them. • Test and debug code in a systematic way.
	LOs	• To login and create my own avatar on Purple Mash.	• To compare music made using computers and music played by real instruments.	• To understand that computer programs work by following instructions called code.	• To create a computer program using an algorithm. • To understand ways that the collision detection

		• To open an activity and then save the work to the 'My Work' area. • To understand how to complete work in the 2Dos area in Purple Mash.	• To explore combining the sounds of different instruments using the 2Beat tool. • To compose a tune to match a picture of a scene. • To compose a tune to match a picture of a scene.	• To use object and action code blocks to make a computer program. • To understand what an event is in coding. • To understand the purpose of an output and to be able to 'read' code to find out what it does. • To change aspects of the design view. • To write code for a computer program.	event can be used in a program. • To design a program that follows a timed sequence. • To understand that different objects have different attributes (properties). • To understand the function of button objects in a program. • To understand the importance of testing and debugging.
	Knowledge Organiser	Knowledge Organiser	Knowledge Organiser	Knowledge Organiser	Knowledge Organiser
	Assessment	Assessment Quiz	Assessment Quiz	Assessment Quiz	Assessment Quiz
Cl.3	**Unit Title**	**Email**	**Route Planners**	**Coding**	
	Disciplinary Knowledge	• To know there are different methods of communication and they each have strengths and weaknesses. • To know emails are a form of digital communication. • To know emails can be sent and received almost instantly to anyone with an email address. • To know common features of email software are the inbox, the 'To' address field, the sender email address, the subject, the message text, and the compose and reply functions. • To know alerts can be used to notify a person that they have unread email.	• To know the combinations of a direction and a distance is known as a command in 2Go. • To know commands can be input into 2Go to control the movements of a screen turtle in four directions. • To know the turtle can be set to rotate by angles of 90 or 45 degrees both clockwise and anticlockwise. • To know planning a route is important to ensure the correct commands are input. • To know a list of instructions for a route is called an algorithm. • To know routes can be programmes to perform	• To know flowcharts are a type of diagram that use specifically shaped label boxes and arrows to represent an algorithm as a diagram. • To know timers can control when a block of commands run. • To know timer commands can be run after a timed delay or at regular intervals. • To know in 2Code there are two timer options, timer every or timer after. These can be altered by changing the number of seconds/quarter seconds. • To know repeat is a control block allowing blocks of commands to	

		• To know address books can be saved in the email software. This provides a convenient way to send emails without typing the full email address each time. • To know pictures, documents and other file types can be attached to emails. • To know there are risks related to use of email.	more than one command in a sequence. • To know routes can be programmed to repeat a sequence of commands a set number of times. • To know errors (bugs) can occur because commands may have been input incorrectly. • To now fixing the errors is called debugging.	repeat a specified number of times. • To know testing a debugging are an important part of the process of making computer programs. • To know what nesting is and the effect it has on a program can help when trying to debug a program.	
	Procedural Knowledge	• Explain the advantages and disadvantages of different communication methods. • Choose an appropriate communication method for a task. • Make use of 2Email to communicate within school. • Recognise the differences between digital and non-digital communication methods. • Explain different digital communication methods such as instant messaging, email and video calls. • Identify the advantages of instant communication. • Explain the differences between email and traditional communication methods which are not instant. • Identify the common features of the email screen.	• Input commands in 2Go. • Input purposeful commands in 2Go to make the turtle move in a particular direction towards a goal. • Input commands that rotate the turtle to face the desired direction for movement using both 90 and 45 degree angles. • Use techniques such as finger movements to plan a route. • Plan the route by first writing the algorithm and then inputting the code commands. • Input several commands into a sequential algorithm layout and run this code to move the turtle along the programmed route. • Reset the turtle to the starting position to re-run the code. • Use the repeat algorithm layout for 2Go commands.	• Identify features of a flowchart including the starts, inputs, outputs, delays such as a timer, actions and events. • Follow the flow of the chart and interpret what it is representing. • Use a flowchart as part of a planning process for designing a program. • Code the features of timer after command including the interval, actions, correct nesting of commands. • Distinguish between the use of the timer every and timer after commands. • Use a timer every command to make an event happen repeatedly. • Identify situations when a repeat command can make the code more efficient. • Use a repeat command to make other commands repeat.	

		• Make use of these common features of email software to communicate digitally. • Check alerts for new messages and respond to these. • Use the address book within 2Email to find contacts. • Send emails to multiple contacts using the address book. • Identify the attachment icon. • Select files to attach to an email and send. • Be cautious of emails that have attachments. • Discuss the advantages and disadvantages of being able to send attachments with emails. • Recognise a concerning email/contact. • Identify who a trusted contact is. • Report any concern to a trusted adult and use the report to teacher feature in 2Email. • Consider consent when sharing content digitally. • Recognise personal and private information and know what is not appropriate to share with a recipient.	• Anticipate the effect of the repeat and use logical reasoning to decide upon when this layout is useful to accomplish a task. • Make logical attempts to debug code for routes. • Reset, debug and re-run the code to test routes.	• Read the code of example programs with nesting to determine the outcomes of running the code. • Test what happens when changing how a program is nested. Examples could include moving a timer so that it is nested within a timer that is currently nested within a when clicked command and comparing how it executes. • Use the knowledge of nesting to help debug a program that isn't working as intended.	
	LOs	• To understand how people communicate with each other.	• To create simple commands in 2Go to move the turtle along the route.	• To understand what a flowchart is and how flowcharts are used in computer programming.	

		• To understand and respond to an email. • To compose and send an email. • To send an email attachment. • To learn how to use email safely. • To explore simulated email scenarios.	• To create commands in 2Go in which the turtle turns using rotation. • To plan algorithms and write 2Go code that uses angle of turn. • To use 90 and 45 degree angles in 2Go. • To use the repeat algorithm and coding in 2Go.	• To understand that there are different types of timers. • To understand how to use the repeat command. • To understand the importance of nesting when coding, testing and debugging. • To design a program. • To use design documentation to code a program.	
	Knowledge Organiser	Knowledge Organiser	Knowledge Organiser	Knowledge Organiser	
	Assessment	Assessment Quiz	Assessment Quiz	Assessment Quiz	
Cl.4	**Unit Title**	**Effective Searching**	**Game Creator**	**Coding**	**Unpacking Hardware and Software**
	Disciplinary Knowledge	• To know the internet is a global network of computers that share information from one device to another, anywhere in the world. • To know we can search for information on the internet using a tool called a search engine. • To know a search engine works by collecting and organising vast amounts of data from webpages across the internet. • To know typing in keywords rather than a whole question can be a quicker and more effective way of finding the correct information. • To know search engines send out programs called web crawlers or spiders to	• To know there are criteria which determine the playability of a video game. • To know 3D games can be made using the 2DIY 3D tool. • To know to maximise playability, it is important to plan and design a game before making it. • To know the visual properties of a game should fit with the theme and add a feeling of immersion to the experience. • To know the audio properties of a game should fit with the theme and add a feeling of immersion to the experience.	• To know the way that code is written and structured can improve the ability to debug it, simplify the code and run it more efficiently. • To know that a simulation is a model that represents a real or imaginary situation. • To know abstraction is a way to think about a situation without unnecessary details, enabling a focus on the function of a program. • To know decomposition is a method of breaking down a task into manageable components. This makes coding easier as the components can then be coded separately and then brought back	• To know the word 'technology' describes using scientific knowledge to design and make tools, systems or machines that help solve problems or make tasks easier. • To know electrical, digital and smart technology are sub-sets of technology. • To know hardware describes the physical parts of a computer. • To know software describes the programs that instruct a computer to complete computational tasks. • To know software and hardware operate together to follow processes that assist in completing tasks.

		explore the web for information. They make a copy of it and store it on their database called an index. • To know when you type a query into a search engine it searched its index for relevant information and ranks it according to how useful it is. This is the order in which the information is displayed. • To know a search engine bases its ranking on the relevance of the information, the popularity of the website the information is from and whether the business has paid for the information rank higher. This means the higher ranked results are not always the best. • To know there are a range of techniques you can use to refine a search query, including using quotes, using the minus sign and using filters. • To know information can be put online by anyone so we should not take everything we read as fact. • To know a fact is something that is true and provable, an opinion is something that someone feels and a belief is something that someone	• To know the design of baddie sprites and collectable quest items is a key aspect of game creation. • To know it is important to give the player good instructions to enable them to enjoy playing the game. • To know evaluation is important so a game can be improved.	together in the program. Each part is known as a sub-routine. • To know a function is a named group of commands that a program can run when called by name; reducing the need to rewrite code repeatedly. • To know a datatype defines the type of data, for example, a number or text, and what operations can be performed with it. • To know it is important to initialize variable values to avoid unanticipated bugs in the program. • The know concatenation is the name given to the action of linking things together in a series. For example, in programs we might want to link words together to form random phrases that are seen by a user of a program.	

		accepts to be true based on their faith or tradition. • To know fae news is information that might look real but isn't. Sometimes people make fake news for fun, to make money or try and change the way people think. • To know it is important to evaluate the reliability od any information that we find online.			
	Procedural Knowledge	• Recognise and find a search engine on the internet. • Explain the uses of a search engine. • Recognise and name some popular search engines. • Explain the different stages of how a search engine works to find relevant information. • Select appropriate key words to use in a search to answer a particular question. • Perform a keyword search when looking for information online. • Explain how information on the webpages is gathered, copied and stored by a search engine's program known as web crawlers or spiders. • Explain that when a search is made, a search engine searches through	• Evaluate the playability of a video game. • Make a 3D game using the key functions of the 2DIY 3D game creator tool. • Use a design document to plan the theme and aim, characters and game world, game features, timing, character movement and interaction for a 3D game. • Choose and design appropriate textures for the game world walls, floors, lighting, hazards and scenery. • Choose or compose appropriate sound effects and music for the game. • Design the collectable quest items and add movement, sound effects and actions. • Consider where to place the collectable quest items so it is possible to finish the game.	• Use tags to write code for several objects in one block of commands facilitated by the computer-generated variables for tagged objects. • Use tabs to organise code. • Use functions to reduce the need to repeat blocks of code and make the program easier to read. • Explain what a simulation is and provide examples of physical systems that could be made into a simulation. • Plan an algorithm for a physical system such as traffic light sequences using 2Chart. • Use the algorithm to write the code. • Test and evaluate the program. • Use the program to model changes to the physical system for instance	• Identify whether an item is an example of technology. • Decide whether an item is an example of electrical, digital or smart technology. • Define what is meant by hardware, components and peripherals. • Name hardware components of a computer system. • Describe the function of these different parts. • Identify the functions and common components of different software tools and relate them to the tasks those tools perform. • Describe a process in terms of inputs, hardware and software processing and outputs.

		its index rather than through the pages of the internet. • Explain how search results are selected and ranked. • Look through search results and evaluate which could be the most useful. • Use techniques such as quotes, the minus sign and filters to refine a search query. • Explain the importance of questioning where information online has actually come from. • Explain the differences and identify examples of facts, options and beliefs. • Identify the difference between real news and 'fake news'. • Understand the reasons why someone might want to put fake news online. • Check whether information online is reliable and true by checking if it sounds real, checking the sources, checking the date and checking if the news is also reported by other sources.	• Place sprites in the game in such a way as to provide challenge but not make it impossible to play. • Select appropriate penalties for encountering an enemy. • Create useful instruction screens for players. • Evaluate games made by their peers using given criteria. • Read evaluations of their game from others. • Make appropriate improvements to their game.	improvements if there are traffic jams at the lights. • Recognise what abstraction is and why it is important. • When planning a program, use abstraction to focus thinking on the task itself. • During planning, use decomposition to break down the plan into the key parts that are required to get the program functioning. • Use decomposition as a way to organise the code. For example, using tabs in 2Code. • Use 2Chart to create a flowchart that contains sub-routines. • Create functions in 2Code and call these functions appropriately instead of repeating code. • Refine existing code to include functions by recognising where code is repeated within the program. • Make sensible choices of datatype when creating variables. • Anticipate the effect of operators on the variable values given their datatype. • Use the Variable Watch window to monitor the changing value of the variables in a program.	

				• Initialize all variables appropriately. • When debugging, look at the variable data type and initialization as possible sources of bugs. • Recognise where concatenation can be useful in programs. • Use the print screen command in combination with random words that are joined together to demonstrate concatenation. • Demonstrate concatenation in other programs created such as linking a variable (score) with text (well done) for a game. • Debug programs that involve concatenation errors in strings.	
	LOs	• To understand what a search engine is, how it works and how to use simple keywords to find information online. • To understand how search engines collect, sort and rank results, and why some results appear first. • To learn advanced ways to improve searches so we can fine the most accurate and useful information quickly. • To develop strategies to judge whether information	• To evaluate the features of a successful video game. • To plan a game in 2DIY3D. • To design and use game sprites. • To add features to a game world and check playability. • To evaluate games created by others.	• To understand some ways that code can be simplified so that it is easier to read and runs more efficiently. • To program a computer simulation using 2Code. • To know what decomposition and abstraction are in Computer Science. • To understand what a function is and how functions work in code. • To understand what datatypes are and how they are used when coding with variables.	• To understand what technology is, where we see it in everyday life and how it helps us at school, home and beyond. • To identify and describe the different pieces of computer hardware and understand their functions. • To understand what software is and how we use it to complete tasks. • To explore how hardware and software interact to complete everyday tasks.

		online is true and if sources can be trusted.		• To read code, predict outcomes and identify and fix bugs.	
	Knowledge Organiser	Knowledge Organiser	Knowledge Organiser	Knowledge Organiser	Knowledge Organiser
	Assessment	Assessment Quiz	Assessment Quiz	Assessment Quiz	Assessment Quiz
Cl.5	**Unit Title**	**Coding**	**Coding**	**Networks**	
	Disciplinary Knowledge	• To know the way that code is written and structured can improve the ability to debug it, simplify the code and run it more efficiently. • To know that a simulation is a model that represents a real or imaginary situation. • To know abstraction is a way to think about a situation without unnecessary details, enabling a focus on the function of a program. • To know decomposition is a method of breaking down a task into manageable components. This makes coding easier as the components can then be coded separately and then brought back together in the program. Each part is known as a sub-routine. • To know a function is a named group of commands that a program can run when called by name; reducing the need to rewrite code repeatedly.	• To know cloning is a process where additional objects can be created while the program is running. • To know objects can be draggable in 2Code to provide another way for the user to interact with them. • To know hotspots can be created in 2Code. These are a type of object that can be used to make a specific area of the screen react to actions such as objects entering or leaving the area. • To know using a combination of coding structures and objects allows the design of interactive and creative programs. • To know variables can have different designated datatypes. The datatype will impact how the value of the variable can be used. • To know there are a variety of ways of receiving input from users and then using this input.	• To know a network describes a group of connected computers that can share information and hardware resources. • To know LAN and WAN are different kinds of networks. • To know certain hardware is required to create a network. • To know networks can be wired or wireless or a combination of both. • To know there is a difference between the World Wide Web and the Internet. • To know web browsers are used to access the World Wide Web. • To know the existence of networks has opened online communication. • To know internet filtering and censorship are both used to make parts of the internet less accessible for different reasons.	

		• To know a datatype defines the type of data, for example, a number or text, and what operations can be performed with it. • To know it is important to initialize variable values to avoid unanticipated bugs in the program. • The know concatenation is the name given to the action of linking things together in a series. For example, in programs we might want to link words together to form random phrases that are seen by a user of a program.	• To know a flowchart is a diagram that uses specifically shaped, labelled boxes and arrows to represent an algorithm as a diagram. • To know using functions helps with making programs more efficient. Instead of writing the same sequence of code repeatedly. • To know design documentation is vital when producing more complex programs. The process of designing supports abstraction and decomposition of the task at hand. • To know text-based adventures are computer games that have been created using text instead of graphics. Players use text commands to control characters and influence the environment.		
	Procedural Knowledge	• Use tags to write code for several objects in one block of commands facilitated by the computer-generated variables for tagged objects. • Use tabs to organise code. • Use functions to reduce the need to repeat blocks of code and make the program easier to read.	• Write code to clone objects and loop through the clones to change their attributes. • Make objects draggable in a program. • Create hotspots and write code that controls the flow of the program when objects enter and leave hotspots. • Use code to create effects when combining the use of draggable objects,	• Identify types of computer networks locally and globally. • Explain the hardware resources that a network might share. • Explain the difference between LAN and WAN. • Create a network diagram that includes hardware such as router and different connected devices and peripherals.	

		• Explain what a simulation is and provide examples of physical systems that could be made into a simulation. • Plan an algorithm for a physical system such as traffic light sequences using 2Chart. • Use the algorithm to write the code. • Test and evaluate the program. • Use the program to model changes to the physical system for instance improvements if there are traffic jams at the lights. • Recognise what abstraction is and why it is important. • When planning a program, use abstraction to focus thinking on the task itself. • During planning, use decomposition to break down the plan into the key parts that are required to get the program functioning. • Use decomposition as a way to organise the code. For example, using tabs in 2Code. • Use 2Chart to create a flowchart that contains sub-routines. • Create functions in 2Code and call these functions appropriately instead of repeating code.	hotspots and cloning in conjunction with sequences, repetition and selection. • Make conscious choices about datatype when creating variables. • Anticipate effects such as calculating using variable values, concatenation of string variables and bugs that could occur due to datatypes when user input is used. • Use 'Prompt for input' and 'get input' commands to receive input from the user of a program and write code using variables to work with user input. • Use if statements that include checking of a value of a user input and changing a variable if user input has met the if statement condition. • Interpret flowcharts and use these as a basis to begin coding or debugging a program. • Create flowcharts to represent algorithms. • Include sub-routines within flowcharts to help with more complex program ideas. • Create functions and write code for function calls. • Interpret programs that include functions to explain the purpose of the function.	• Identify the terms Wi-Fi, mobile data and 5G as pertaining to wireless network connections. • Describe the difference between the Internet and World Wide Web giving examples of the services that both provide. • Give examples of web browser tools. • Give examples of online communication. • Give safety-tips related to online communication. • Explain the differences between internet filtering and censorship and why they are used.	

		• Refine existing code to include functions by recognising where code is repeated within the program. • Make sensible choices of datatype when creating variables. • Anticipate the effect of operators on the variable values given their datatype. • Use the Variable Watch window to monitor the changing value of the variables in a program. • Initialize all variables appropriately. • When debugging, look at the variable data type and initialization as possible sources of bugs. • Recognise where concatenation can be useful in programs. • Use the print screen command in combination with random words that are joined together to demonstrate concatenation. • Demonstrate concatenation in other programs created such as linking a variable (score) with text (well done) for a game. • Debug programs that involve concatenation errors in strings.	• Interpret design documentation, commenting on the level of abstraction and the way that a task has been decomposed. • Use user interface designs to create the look of a program. • Use flowcharts to decide upon the structure of a program and what functions might be required. • Review the code of a text-based adventure game and relate this to game play when the code is running. • Identify how variables, functions, selection and repeat loops are used to make the program function. • Adapt the code maintaining the structure but changing the game play.		
	LOs	• To understand some ways that code can be	• To understand ways to use cloning in 2Code.	• To understand what aomputer network is and	

		simplified so that it is easier to read and runs more efficiently. • To program a computer simulation using 2Code. • To know what decomposition and abstraction are in Computer Science. • To understand what a function is and how functions work in code. • To understand what datatypes are and how they are used when coding with variables. • To read code, predict outcomes and identify and fix bugs.	• To explore the use of hotspots in 2Code. • To understand the different options for generating and using user input in 2Code. • To use flowcharts to test and debug a simulation. • To use program design documentation to produce a program. To understand how 2Code can be used to make a text-based adventure game.	identify examples of networks at home, school and in the wider world. • To understand the difference between the internet and World Wide Web and explore the services they provide. • To explore how the internet can be used for communication and collaboration, and how to do this safely and respectfully. • To explore who is in charge of the internet and how rules and website blocking can affect people, society and online platforms.	
	Knowledge Organiser	Knowledge Organiser	Knowledge Organiser	Knowledge Organiser	
	Assessment	Assessment Quiz	Assessment Quiz	Assessment Quiz	