Computing Scheme of Work Unit 1.7 – Coding – New From 2021

Contents

Introduction 4
Program Design 4
Levels of Scaffolded coding tasks 5
Medium-Term Plan 6
Lesson 1 - Instructions 7
Aims 7
Success Criteria 7
Resources 7
Preparation 7
Activities 7
Lesson 2 – Objects and Actions 9
Aims 9
Success criteria 9
Resources 9
Preparation 9
Activities 9
Lesson 3 - Events 12
Aim 12
Success Criteria 12
Resources 12
Preparation 12
Activities 12
Lesson 4 – When Code Executes 15
Aim 15
Success criteria 15
Resources 15
Preparation 15
Activities 15
Lesson 5 – Setting the Scene 17
Aims 17
Success Criteria 17
Resources 17

Preparation 17
Activities 17
Lesson 6 – Using a Plan 19
Aims 19
Success criteria 19
Resources 19
Preparation 19
Activities 19
Appendix 1: Display Boards 21
Assessment Guidance 25

Introduction

This unit consists of six lessons that assume no prior coding knowledge. Children will be coding using the 2Code tool.

Key coding vocabulary is shown in **bold** within the lesson plans, use these new words in context to help children understand the meaning of them and start to build up, their vocabulary of coding words.

Children will need to be able to drag and drop to move code blocks around. If children have not had much practice with this then there are several example activities within the Activities section of 2DIY that help children to practise these skills in preparation: 2DIY activities to practise drag and drop. Within each category of activity, look for the example file then press the Play button. If the children have not used Purple Mash before, spend some time showing them how to log in and how to get to 2Code.

The Chimp guided activities provide further practice of the concepts that the children will be learning and can be used as extension activities. More able children can be encouraged to explore other things that they can change in their programs and experiment with the options available, such as image and scale in 2Code.

When children get stuck, they will often be able to solve their own problems either by reading through their code again or by asking their peers; this models the way that coding work is really done. More able children can be encouraged to support their peers, if necessary, helping them to understand but without doing the work for them.

Note: To force links within this document to open in a new tab, right-click on the link then select 'Open link in new tab'.

Program Design

To master coding skills, children need to have the opportunity to explore program design and put computational thinking into practice. The lesson plans introduce children to designing and planning a program before coding in some lessons.

During the design process, children should be encouraged to clarify:

- The characters (objects).
- What they will do (actions and events).
- What order things will happen (the algorithm).

Levels of Scaffolded coding tasks

You can support children's learning and understanding by using different degrees of scaffolding when teaching children to code. The lessons provide many of these levels of scaffolding within them and using Free Code Chimp, Gibbon and Gorilla enables children to clarify their thinking and practise their skills. These are not progressive levels; children can benefit from all the levels of activities at whatever coding skill level they are:

Scaffolding	Task type	Examples of how to provide these opportunities
Most scaffolded ↓ Least scaffolded	Copying code	By giving children examples of code to copy.
	Targeted tasks	• Read and understand code • Remix code to achieve a particular outcome. • Debugging. • Use printed code snippets so that children can't run the code but must read it. • Include unplugged activities and 'explaining' tasks e.g. 'how do variables work?'
	Shared coding	• Sharing Challenge activities as a class or group on the whiteboard. • Complete guided activity challenges as a class. • After completing challenges; share methods to create a class version of the challenge. • Free coding as a class
	Guided exploration	• Exploring a limited repertoire of commands • Remixing code • Explore commands in free code before being taught what they do. • Use questioning to support children's learning. • PRIMM approach; Predict – Run – Investigate – Modify - Make
	Project design and code	**Projects (imitate, innovate, invent, remix)** There are different ways to scaffold learning in projects. This process can be applied to programming projects; • Using example projects e.g. the Guided 2Code activities. • Completing the challenges at the end of each guided activity. • Free code✓ • Create a project that imitates a high-quality exemplar. • Remixing ideas. • Independently creating a brand-new program.
	Tinkering	Use Free code Gorilla to access the full suite of 2Code objects and commands ✓ Use Free code to play and explore freely.

Adapted from work by Jane Waite - Computing at Schools https://www.computingatschool.org.uk/

In Literacy, some teachers follow a progression that scaffolds learning to write texts. At first children read lots of examples of the genre of text they are going to create. Then they create an ***imitation*** of an example text. Next, they create a variation of the text (***remix and innovate***). Finally, they get to ***inventing*** a brand-new version.

Medium-Term Plan

Lesson	Title	Success Criteria
1	Instructions	• Children can give and follow **instructions**. • Children can draw symbols to represent **instructions**. • Children can arrange code blocks to create a set of **instructions**.
2	Objects and Actions	• Children can create a program using **code** blocks. • Children can use **object** and **action code** blocks.
3	Events	• Children can create a simple program using **code** blocks. • Children can use **event**, **object** and **action** code blocks.
4	When Code Executes	• Children can create a simple program using **code** blocks. • Children can use **event**, **object** and **action** code blocks. • Children can notice when their code **executes** when their program is **run**.
5	Setting the Scene	• Children can edit a **scene** by adding, deleting and moving **objects**. • Children can change the size of **objects** using the **properties** table.
6	Using a Plan	• Children can create a design plan for their Free Code Scene program. • Children can use code to make the program they have designed work.

Lesson 1 - Instructions

Aims

- To understand what **instructions** are.
- To predict what will happen when **instructions** are followed.
- To understand that computer programs work by following **instructions** called **code**.

Success Criteria

- Children can give and follow **instructions**.
- Children can draw symbols to represent **instructions**.
- Children can arrange **code** blocks to create a set of **instructions**.

Resources

Unless otherwise stated, all resources can be found on the unit main page. From here, they can be set as 2dos by clicking on the icon. To preview resources linked to here, right-click and '**open in new tab'** so you do not navigate away from this page.

- Code Block Cards. Children will need to use a few copies of each picture to create code away from the computer.
- Optional: Individual Whiteboards

Preparation

- Print and copy the Code Block Cards, enough for 1 between 2.

Activities:

Introduction	Display **slide 2** and outline the lesson aims. Display **slide 3** and outline the success criteria.
Introducing Programming	Use **Slide 4** to explain to the children that they are going to learn about Computer Programming, which is sometimes also known as Coding. Ask them if they know what this is. Discuss briefly that it is the way that computer programmers input instructions into computers to create programs. Can they give any examples of computer programs that they have used?
Activity 1; Giving and Following Instructions	Use **slide 5** to share the first activity - Choose two children; one is a robot, and the other is a programmer. The programmer needs to direct the robot to walk from one place in the classroom to another. How can they give the instructions so that the robot does not crash into objects in the way? Repeat a few times in different locations.

Activity 2: Teacher is the Programmer	Use **slide 6** to explain to the children that you are now going to be the programmer and they are all the robots. Reveal the instructions on the board as symbols. Get children to 'act' out/ follow the instructions you have displayed as symbols – a twirl, a hand next to a toe and a hand next to an ear - the children should twirl, touch their toes then touch their ears.
Activity 3: Using Symbols	Use **slide 7** to display a hand next to an up arrow and see if the children can see that this would be 'hand up'! Ask children to use small whiteboards draw symbols for 'hand down', what about touch nose?
Computers Follow Instructions	Display **slide 8**. Now that children have explored instructions at a symbol level, reiterate the introduction using this slide. Explain to children that computers understand programming languages that use symbols rather than whole sentences. Use slide 8 to help explain that a coder writes instructions called code for the computer to follow, this is called the input. These instructions make our programs work, our programs are the output.
Fish in the Sea	Display **slide 9** Fish in the Sea and discuss what you can see – 3 fish in the sea. Ask children what they think those fish could be programmed to do.
Activity 4: Code Blocks	Show the children the code on **slide 10**. Explain that these are examples of **code** used to program a computer. Can they suggest ways to combine the cards to make **instructions**? Pair up the children and give the pairs some printed out Code Block Cards. Challenge them to join two blocks to give one clear **instruction** (i.e. tuna - left). Child one should lay out a line of code by joining blocks, then Child two should 'read' the line of code and explain what the **code** would do. Repeat, swapping roles.
Activity 6: Extension – Improving our quiz	Display **slide 11.** Explain to the children that the light blue code blocks represent **objects**, and the dark blue code blocks represent **actions.** This set of instructions can be called an **algorithm**. Can they say the word **algorithm?**
Review Success Criteria	Display **slide 12.** Review the success criteria from **slide 3.** Children could rate how well they achieved this using a show of hands.

Lesson 2 – Objects and Actions

Aims

- To use **code** to make a computer program.
- To understand what **objects** and **actions** are.

Success criteria

- Children can create a program using **code** blocks.
- Children can use **object** and **action code** blocks.

Resources

Unless otherwise stated, all resources can be found on the unit main page. From here, they can be set as 2Dos by clicking on the icon. To preview resources linked to here, right-click and '**open in new tab'** so you do not navigate away from this page.

- Code block cards from Lesson 1.
- Fun with Fish Activity. This is on the main 2Code page in the Chimp section.
- Optional: Exercise books to be used as 2Code workbooks for recording coding exercises and designs.

Preparation

- Set Fun with Fish as a 2Do.

Activities

Introduction	Display **slide 2** and outline the lesson aims. Display **slide 3** and outline the success criteria.
Vocabulary	Display **slide 4**. Remind the children about the blocks of **code** that they were using last lesson. Explain that today they are going to be coding on the computers using blocks of **code**. Remind them what an **algorithm** is. Ask them to identify **objects** and **actions**.
	Display **slide 5** to introduce the term **Command**. Explain that a single instruction is called a **command**.
Demonstrating: Fun with Fish	Display **slide 6**. Open Purple Mash and go to 2Dos, click on Preview within the Fun with Fish 2Do to show children the Fun with Fish lesson. Open Stage 1 and click on OK to close the instruction screen. Click on 'Design' (in the top right-hand corner) and discuss what can be seen – a fish in the sea. Explain to children that they will use 2Code to program the object (fish) to do an action (move right).

	Display **slide 7**. Click  to go back to the code view. Open the instruction screen by clicking on the  . Watch the video for stage 1.
	Display **slide 8**. Complete stage 1 as a class; emphasise the need to give the computer clear instructions for moving the fish. The available **actions** for the fish **object** pop-up as soon as the fish is dragged into the code window. Show the children what to do if they click on the wrong direction - click on the direction again and select the correct one. Show the children where the Play button is to **run** the code and emphasise that the code has programmed the **object** to do an **action.** Show them how to move to the next stage of the activity or stop the code running to make changes. Complete stage 2 together as a class.
Activity: Fun with Fish	Display **slide 9**. Ask children to log in to Purple Mash, go to their 2Dos and click on 'Start' on the Fun with Fish 2Do. Challenge them to complete stages 1 and 2. Ask them to use the **code** blocks to make their Tuna move, and then move onto the next challenge to make the Crab move.
	Use **slide 10**. Load stage 3 and explain that this is a stage where you must fix the code that the monkey has got wrong. We call this **debugging**. Complete this stage as a class (show children that if you want to change an action you can click on it). Children to then complete stage 3 In Purple mash independently. Review progress together - did they get lots of code monkey stars? The maximum is 5; they lose stars for using hints.
	Display **slide 11**. Look at stage 4 together – this is the challenge stage. All the guided activities have this challenge stage, and this is where children deepen their understanding of the **code** that they have been working on. Take a few suggestions from the class about how to improve the fish tank by adding new **objects** – fish/ crabs – add one new **object** then switch to the code screen to notice it then appears as a blue object code piece, show how to program it to move and test it out using the play button. Add one or two more **objects** and show children how to use the **event**. Ask children to complete the challenge stage and then save their work before they exit. Review children work together against the lesson aims – this could be done by sharing some good examples from the 2Dos folder.

	Did any children try using the 'Click **event**'? What did that do?
Review Success Criteria	Display **slide 12.** Review the success criteria from **slide 3.** Children could rate how well they achieved this using a show of hands.

Lesson 3 - Events

Aim

- To understand what an **event** is.
- To use an **event** to control an **object**.

Success Criteria

- Children can create a simple program using **code** blocks.
- Children can use **event**, **object** and **action** code blocks.

Resources

Unless otherwise stated, all resources can be found on the unit main page. From here, they can be set as 2Dos by clicking on the icon. To preview resources linked to here, right-click and '**open in new tab'** so you do not navigate away from this page.

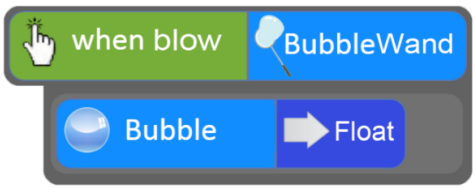
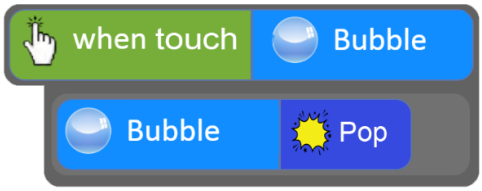
- A pot of bubbles and a bubble wand (usually part of the lid!)
- Bubble Coding
- Example Code

Preparation

- Set Bubble Coding as a 2Do.

Activities

Introduction	Display **slide 2** and outline the lesson aims. Display **slide 3** and outline the success criteria.
Events	Display **slide 4**. Start by telling children that in this lesson they will all use **events** in their coding. See if any of them can remember and describe how an event worked in the last stage of Fun with Fish? (you could bring up an example of a child's previous work to show them).
Activity 1: Bubbles	Use **slide 5.** Blow bubbles. Ask children: What is the **event**? (What do you do to make something happen?) (blow bubble wand – if they just say 'blow', blow the air nowhere near the bubble wand and ask them if that works! Talk about the need for **precise instructions**). What are the **objects**? (bubble wand, bubble) What is the **action**? (float)

	Use the Example Code or display **slide 6.** Show the children what the code could look like for blowing bubbles. 
	Use **slide 7** to talk about the **event**, **object**, **action** (touch – bubble – pop). Ask children to rearrange the following **code** for that (this could be done physically with Example code printed on paper or as a drag into place activity on the board using the code on slide 7). 
Activity 2: Other Events	Use **slide 8** to discuss other **event** – **object** – **action** examples children might be familiar with (e.g. push – swing – swing - forward, kick – football - football – roll). Rearrange the **code** for the football example. (this could be done physically using Example Code printed on paper or as a drag into place activity on the board using slide 8). 
Activity 3: Bubbles Coding	Use **slide 9** to demonstrate click events activity before they have a go (slide 10).Open Purple Mash and go to 2Dos, click on 'Preview' within the Bubble Coding 2Do to show children the Bubbles lesson. Look at the available code blocks available and see if children can tell you what they might see when you click to 'Design'. Referring to **event**, **object**, **action**, add code that makes a bubble move up when it is clicked on, make another bubble pop when it is clicked on. Remind children of the play button to **run** the code. Test the **code** you have just added in together, discuss what other **code** they could add to make the other bubbles move.
	Use **slide 10** to set children off independently on Bubbles Coding from 2Dos.

	Use **slide 11** to review progress together as a class – using terminology **event**, **object**, **action**. Have any of the children noticed or tried to use the **sound** button? Demonstrate how it might be used so the bubble makes a **sound** when it pops. Challenge children to improve their program by adding **sounds** and then saving before exit.
Review Success Criteria	Display **slide 12**. Review the success criteria from slide 3. Children could rate how well they achieved this using a show of hands.

Lesson 4 – When Code Executes

Aim

- To understand what an **event** is.
- To begin to understand how **code executes** when a program is **run**.

Success criteria

- Children can create a simple program using **code** blocks.
- Children can use **event**, **object** and **action code** blocks.
- Children notice when their code **executes** when their program is **run**.

Resources

Unless otherwise stated, all resources can be found on the unit main page. From here, they can be set as 2dos by clicking on the icon. To preview resources linked to here, right-click and '**open in new tab'** so you do not navigate away from this page.

- Air Traffic Control. This is on the main 2Code Page (scroll down to the Chimp activities).
- Haunted Scene. This is on the main 2Code Page (scroll down to the Chimp activities).

Preparation

- Set Air Traffic Control as a 2Do.
- Set Haunted Scene as a 2Do for an extension task.

Activities

Introduction	Display **slide 2** and outline the lesson aims. Display **slide 3** and outline the success criteria.
Activity 1: Air Traffic Control	Display **slide 4.** In Purple Mash, open Air Traffic Control by clicking on 'Preview' in the 2Dos area. Watch the video for stage 1 together as a class. Remind children that there are hints if they need them, and that once they've clicked on OK they can get back to the hint by clicking on the instruction at the top. Ask children to come up to the front of the class and use the code blocks to make the plane take off when it is clicked on: 

	Click on **run**  to test the code and see if it works as they were expecting. When you do this, notice with them that the **code** highlights orange when it **executes** (you may need to click on 'Stop' and re-run the program to point it out). Explain that if you click on the plane before the **code executes** it won't take off, they need to make sure the code **executes** first. Get children to watch the code and see what happens when you click on the planes, which bit of code **executes** when?
	Use **slide 5** to set children off on completing stage 1 and 2 of Air Traffic control themselves.
	Use **slide 6** to support looking at stage 3 together as a class – remind children what **debugging** is and show them how you can drag code around to move it into a different place and click on actions to change them. Fix the code together and click on Next Challenge. Remind children about the Challenge Stage and that they can add more **objects** (runways, planes, helicopters) to program. Set children off to complete the **debugging** and challenge stages. Discuss with the children what they have done this lesson using the key vocabulary.
Activity 2: Extension	Display **Slide 7**. For children who you feel are confident, ask them to try the Haunted Scene activity. You might want to explain what a Sprite is.
Review Success Criteria	Display **slide 8**. Review the success criteria from slide 3. Children could rate how well they achieved this using a show of hands.

Lesson 5 – Setting the Scene

Aims

- To understand what **backgrounds** and **objects** are.
- To understand how to use the **scale property**.

Success Criteria

- Children can edit a **scene** by adding, deleting and moving **objects**.
- Children can change the size of **objects** using the **properties** table.

Resources

Unless otherwise stated, all resources can be found on the unit main page. From here, they can be set as 2Dos by clicking on the icon. To preview resources linked to here, right-click and '**open in new tab'** so you do not navigate away from this page.

- Free Code Scenes. This is on the main 2Code Page (scroll down to the Chimp activities).
- Air Traffic Control. This is on the main 2Code Page (scroll down to the Chimp activities).

Preparation

- Set Free Code Scenes as a 2Do.

Activities

Introduction	Display **slide 2** and outline the lesson aims. Display **slide 3** and outline the success criteria.
Vocabulary	Display **slide 4**. Load Air Traffic Control. Ask children to help you recap stages 1and 2 using key vocabulary **code, event, object, action**, **execute**.
The Air Traffic Control Scene	Display **slide 5**. Stop at Stage 2 and notice with the children that there are 2 planes and 2 runways and the runways have some grass around them. Explain that the runways and grass are the **background**, and that the planes are **objects** – and all together they make the **scene**.
Your Scene	Use **slide 6** to explain to children that in this lesson they will be designing a **scene**. Ask children "If you could design a scene, what would your **background** and **objects** be?" Suggest maybe… ○ The **background** could be the woods and a rabbit and a squirrel might be my **objects** ○ The **background** could be space and a rocket and an alien could be my **objects**

	Give the children some time with a talking partner to discuss ideas for a **scene** they might choose if they could make any program they wanted.
Free Code Scenes	Display **slide 7**. Open Free Code Scenes by going to your 2Dos and clicking on 'Preview'. Choose a **scene** and click on 'Design' to show children the **backgrounds** and possible **objects**. Click on the menu and then 'New' to choose a different one, and look through the designs of each: - Space (Aliens, astronauts, rockets, robots) - Flowers and Fairies (Fairies, bee, ladybird) - Sky High Code (Balloon, Helicopter, Plane, Cloud, Rain, Snow, Sun, Thunder, Wind) - Spooky Code (Bat, ghost, skeleton, witch)
	Display **slide 8**. Choose one together. Show children how to: - Delete an **object** (click on an **object** and then click on the bin). - Move an **object** (click on an **object** and then and drag it). - Change the size of an **object** (click on an **object** and then on the value of the **scale** to increase or decrease it).
Activity: Design a Scene	Display **Slide 9**. Ask children to log on to Purple Mash, go to their 2Dos and start the Free Code Scenes 2Do you've set them. Ask them to choose a **scene** from the list and add, delete, move, change, copy objects until they are happy with their design. Ask children to save their **scene** when they have finished. Use the 2Dos folder to open a selection of children' designs and share and celebrate achievement.
Review Success Criteria	Review the success criteria from slide 3. Children could rate how well they achieved this using a show of hands.

*Children will continue their work next lesson, so leave the 2DO open.

Lesson 6 – Using a Plan

Aims

- To plan a computer program.
- To make a computer program.

Success criteria

- Children can create a design plan for their Free Code Scenes program.
- Children can use **code** to make the program they have designed work.

Resources

Unless otherwise stated, all resources can be found on the unit main page. From here, they can be set as 2Dos by clicking on the icon. To preview resources linked to here, right-click and '**open in new tab'** so you do not navigate away from this page.

- Free Code Scenes Backgrounds
- Free Code Scenes

Preparation

- Print/ copy Free Code Scenes Backgrounds – enough for each child to have a copy of the background of the scene they designed last lesson.
- Create a display board for the class to share their programs to. Details of how to do this are given in Appendix 1

Activities

Introduction	Display **slide 2** and outline the lesson aims. Display **slide 3** and outline the success criteria.
Review Scene Design	Display **slide 4**. Ask children to recap what they did last lesson: - *Create a **scene** using **backgrounds** and **objects.*** - *Learn how to add, delete, change the size of and move **objects.***
Scene Plan	Show children **slide 5** and ask them to suggest what they think will happen when the program is **run** and the **code executes**. • Open Free Code Scenes and select Flowers and Fairies. Use the code blocks to program the **objects** to move, hide and speak when they are clicked on, in a way that reflects the plan.

	• Remind children of the click **event**, introduce children to the 'speak' action. • **Run** the program and interact with it, does it reflect the plan? Is any **debugging** needed? • Explain to children that in this lesson they are going to plan **actions** for the **objects** in their **scene**.
Activity: Plan your Program	Display **slide 6**. In Purple Mash, impersonate a child and show them how to open 2Dos and click on 'Continue Work' to open their scene from last lesson. Give children a copy of the background they chose last lesson and ask them to draw on their **objects** and symbols to indicate what **actions** they will program them to do.
Activity: Make your Program	Display **slide 7**. Once they have finished their plan, children can use 2Code to program the **objects** in their **scenes** to reflect their designs. Challenge them to use the click **event**. Share children's work 2Displayboard (see Appendix 1) and have a coding show to share children's programs and celebrate achievements. Use key vocabulary from the unit to discuss how the **code** is making the program work.
Review Success Criteria	Review the success criteria from slide 3. Children could rate how well they achieved this using a show of hands.

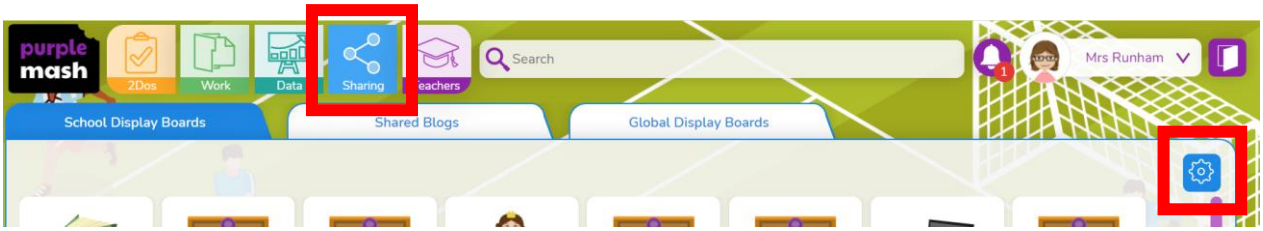
Remember to close your 2Dos when you have finished the lesson.

Appendix 1: Display Boards

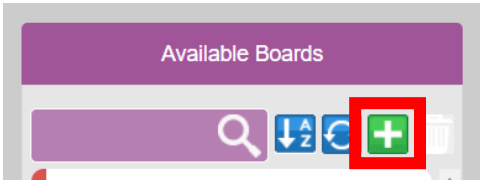
Create the Display Board

Creating the display board is usually something you do before the lesson.

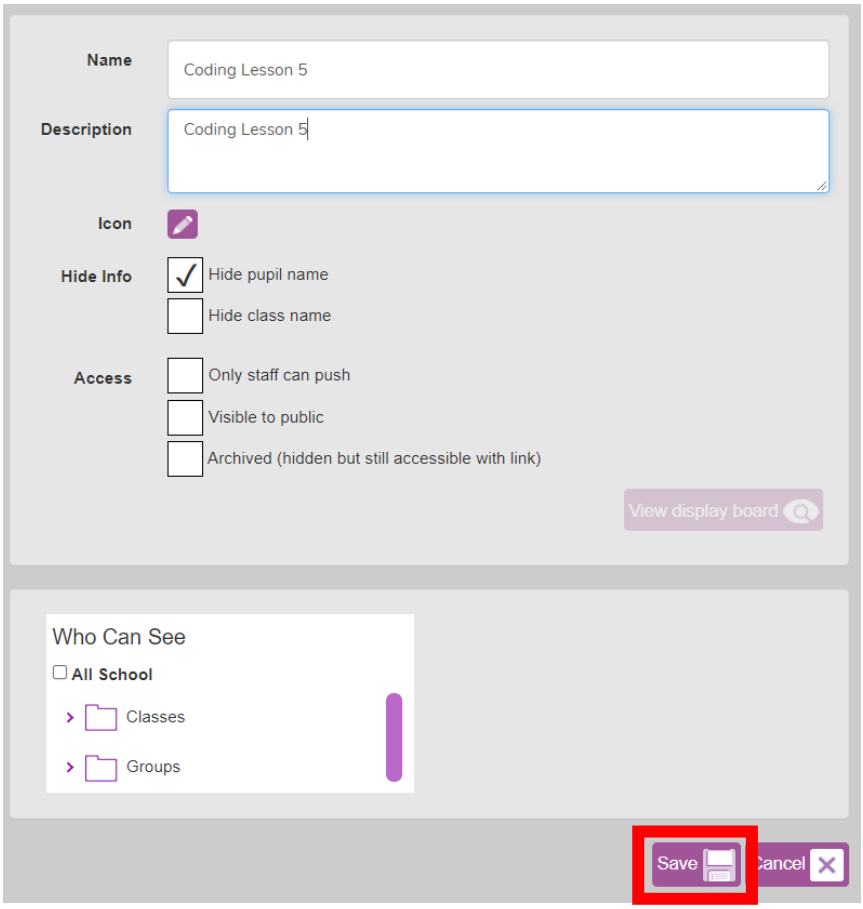
1. Click on the 'Sharing' button to find the Display Board tab, and then click on the settings cog:



2. Click on the '+' in the menu on the left:



3. Edit the settings (don't forget to add an icon by clicking on the), select the class and then click on 'Save':



4. Exit Display Board settings:

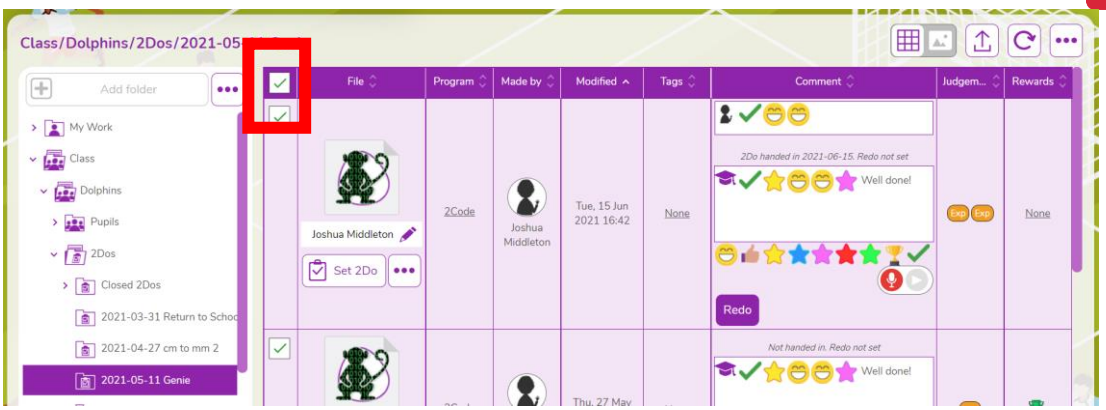
The Display Board will now be visible under the 'Sharing' button to all those you've selected to have access to it.

Adding work to a Display Board:

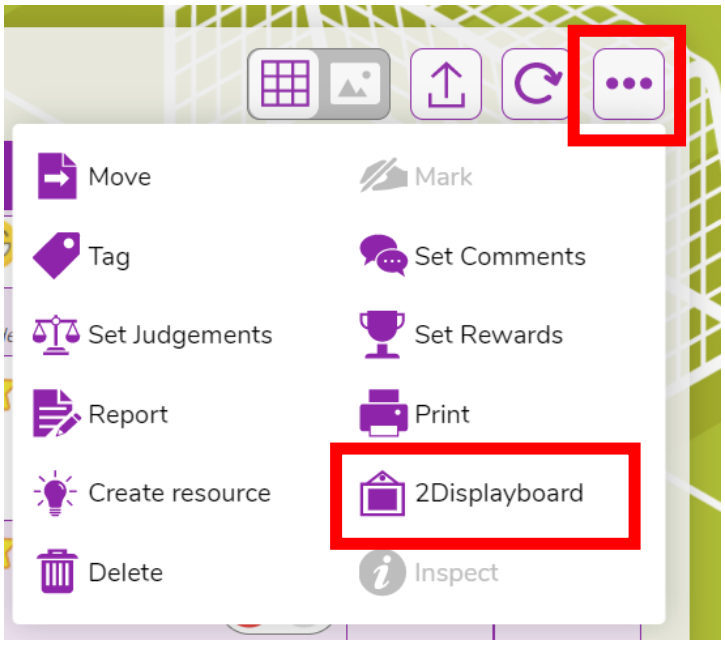
1. Click on 'View Folder' from the 2Do:

(or navigate to the work you want to share in the Work area).

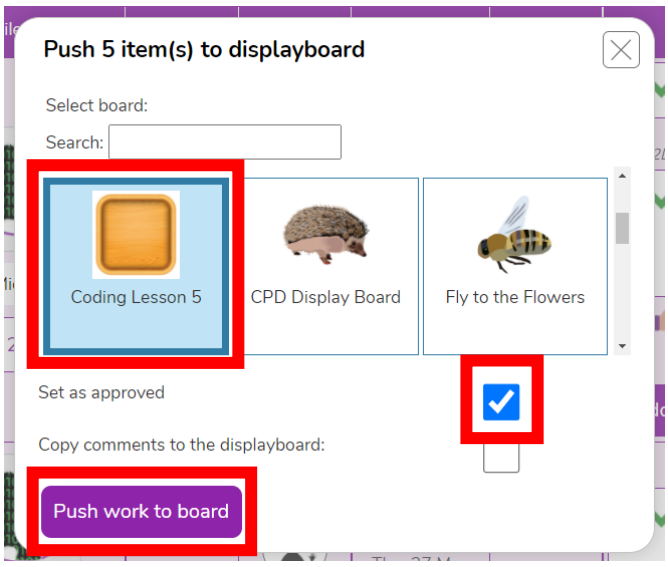
2. Select the files you want to add to the display board or select all files in the folder using the tick at the top.



3. Click on the '...' menu button top right, then click on '2Displayboard':



4. Choose the display board you've made for the work, tick 'Set as approved' and 'Push work to board':

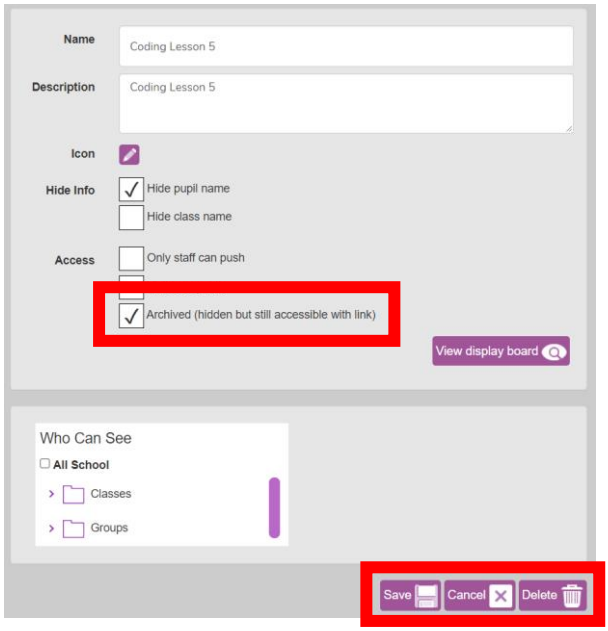


5. Click on 'Sharing' button and then on the display board, you should see the work you've added. It can be deleted by clicking on 'Edit' at the top of the board, then clicking work and then delete. This will remove it from the display board, it won't delete it from Purple Mash.

Deleting or Archiving a Display Board:

When you've finished the lesson you can return to the Displayboard settings and either delete it or archive it to stop it appearing under the 'Sharing' button.

1. Click on 'Sharing' and then on the settings cog.
2. Tick 'Archive', and then 'Save' OR 'Delete'
   Clicking on 'Delete' will delete the display board but the work will still be available in the work area, it doesn't not delete the files.



Assessment Guidance

The unit overview for year 1 contains details of national curricula mapped to the Purple Mash Units. The following information is an exemplar of what a child at an expected level would be able to demonstrate when completing this unit with additional exemplars to demonstrate how this would vary for a child with emerging or exceeding achievements.

Assessment Guidance	
Emerging	Children have a basic understanding that coding involves writing instructions that a computer can follow. They are developing their understanding that these instructions must be precise and carefully structured through their work making simple one and two Stage programs (for example in programs where they make an object move when clicked on). With support, children can create a simple one Stage program that achieves a specific purpose. (In Unit 1.7 Lesson 2, they can make a fish object move). With support, they can create a scene (Unit 1.7 Lesson 5) and plan to make objects move (Unit 1.7 Lesson 6). Children are beginning to understand that they can correct unexpected outcomes by changing the code and they make attempts to identify the source of bugs. With support, children can explain the possible actions of objects including movement and sound. When looking at a simple program they can 'read' the code one line at a time and predict what will happen but might not be able to envision the bigger picture of the overall effect of the program. With support, children can manipulate how their program looks using the 2Code design mode, by adding and changing objects (Unit 1.7 Lesson 5). They can create a program that controls an object.
Expected	Children can both give and receive verbal instruction to achieve a simple outcome such as getting from one point of the classroom to the other whilst avoiding obstacles. Furthermore, they can use printed block-based code to also articulate a simple set of instructions (Unit 1.7 Lesson 1). Children can apply off-screen block code to on-screen block code within 2Code (Unit 1.7 Lessons 2, 3). Children can consider a variety of factors when coding, including the way that the program is designed (Unit 1.7 Lesson 5). They can then design programs that control the look and the actions of objects.

Assessment Guidance	
	Their designs show that they have thought about the need for precise, purposeful, ordered instructions. For example, (Unit 1.7 Lesson 6), they consider the kinds of actions they know to be possible when designing their program. Children think about the program they are making with reference to the objects, the actions and the output e.g. they know that an object will get clicked on and then an object will do something in response (Unit 1.7 Lesson 6). They can then construct their code purposefully to make objects interact. Using the 2Code design mode, children can manipulate how their program looks by adding and changing objects. They can break a problem down into small chunks and then combine it to see an outcome e.g. combine two parts of code "When we click the red bubble, red bubble hides." They know that any unexpected outcome is due to the code that they have created and make logical attempts to try to fix this code rather than attributing it to a fault with the computer understanding the instructions. When looking at a program they can 'read' the code one line at a time and make good attempts to envision the bigger picture of the overall effect of the program.
Exceeding	Children can consider a variety of factors when coding including the way that the program is designed. They can then design programs that control the look and the actions of objects. Children will try, test and often succeed in using code blocks that are new to them and that haven't been explained (e.g. event in Unit 1.7 Lesson 2 and Sound in Unit 1.7 Lesson 3). They intuitively debug their code knowing that any unexpected outcome is down to the code and not the computer's understanding. When looking at a program they can 'read' the code one line at a time and take account of the nesting of lines of code inside each other e.g. the action of a click event. Children can explain the possible actions of objects including moving and responding to being clicked on. They can also use their prior coding experience to recognise whole blocks of familiar code instead of reading line by line which helps them to envision the bigger picture of the overall effect of the program. Using the 2Code design mode and the properties table, children can manipulate how their program looks (Unit 1.7 Lesson 5). They can create a program that controls multiple objects (Unit 1.7 Lesson 6).