

Offline coding activities

To support coding objectives from the Computing curriculum

(with links to Discovery Education Coding units)



CODING™

Giving instructions	4
Robots	4
Do as I say, not as I do	4
Obstacle course	4
Barrier games	4
Knowing that programs execute by following instructions and understanding the link between algorithms, code, and programs	5
The 'Go!' game	5
Run robots, run!	5
Different types of input and output	5
Thought shower	5
Walkabout	6
Function machine	6
Sequence	6
Stand in order	6
Following instructions	7
Walking the playground	7
Selection (conditional events)	7
Simon Says	8
Getting if-y with it	8
Repetition and loops	8
Getting ready	8
Animation detective	8
Song lyrics	8
Dance routines	9
Variables	9
Crazy story generator	9
Understanding the difference between Events, Objects, and Actions.	9
Girls do / boys do	9
Ship, Shore, Submarine	10

Extension ideas — transition to KS3	10
Cup sort	10
Searching for cards	10
Patterns: Spelling sort activity Crazy Story Generator – introducing variables	12
Story 1	12
Story 2	12

THESE ACTIVITIES SUPPORT THE FOLLOWING:

Key Stage 1

Understand what algorithms are, how they are implemented as programs on digital devices, and that programs execute by following a sequence of instructions

Write and test simple programs

Use logical reasoning to predict the behaviour of simple programs

Key Stage 2

Design and write 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; generate appropriate inputs and predicted outputs to test programs

Use logical reasoning to explain how a simple algorithm works and to detect and correct errors in algorithms and programs

GIVING INSTRUCTIONS

Related Discovery Education Coding units:

Block coding > Unit 1a Moving and clicking

<https://central.espresso.co.uk/espresso/coding/unit.html#/unit1a/lesson1>

Block coding > Unit 2b Buttons and instructions

<https://central.espresso.co.uk/espresso/coding/unit.html#/unit1b/lesson1>

Simple games can help young children become familiar with giving and receiving instructions.

Robots

Tell children you would like them to be robots — for this you will give an instruction and they have to follow it. Start with simple instructions and extend to instructions that involve doing more than one thing at the same time, or a sequence of instructions in a row.

Children could sit opposite each other in pairs, and take it in turns to give / receive instructions (these can be anything they like, as long as both children remain seated!). Ask children to share some of the instructions they were given — for the whole class to follow.

Do as I say, not as I do

Play an instruction game similar to *Robots*, but explain to children that you are going to try and trick them — as you give the instruction, you will do a different action.

They need to listen and follow the instructions you give them; i.e. do as you say, not as you do.

Obstacle course

Set up a simple obstacle course in the classroom and ask pupils to give instructions to you, or another pupil, to complete the course.

Barrier games

Working in pairs, pupils sit back to back (or with some sort of barrier in place) so they can't see each other's work. One pupil, the 'sender', does a drawing (e.g. of a house, or a monster made of simple shapes) and then gives instructions to a 'receiver', who must replicate the same drawing simply by following the instructions given. Squared paper could help children to describe size and positions more accurately.

Pupils could play the game twice, once with the 'receiver' being given a chance to ask questions or clarify, and once where they are not allowed to speak, to see what difference this makes.

Alternatively, the 'sender' of the message could have a chance to see their partners' finished drawing, then try again, refining the instructions to make them clearer where they needed to be.

KNOWING THAT PROGRAMS EXECUTE BY FOLLOWING INSTRUCTIONS AND UNDERSTANDING THE LINK BETWEEN ALGORITHMS, CODE, AND PROGRAMS

The 'Go!' game

Repeat the *Robots* activity, but this time the robots only follow the actions after you say 'go'. Give several children a turn at being the one to say 'go' or being the robot. They could also play this in pairs.

Run robots, run!

Explain that computers don't understand the word 'go', but have a special word which means the same thing — run. Play *the 'Go!' game* as before, substituting the word 'go' for 'run'; i.e. give children an instruction, but they need to wait for the command 'run' before they follow it. Try and catch children out — say 'go' and see what they do.

You could also ask them why, they are doing the action that they are, instead of running. They should be able to explain that, to a computer, the word 'run' means something different.

DIFFERENT TYPES OF INPUT AND OUTPUT

Related Discovery Education Coding units:

Block coding > Unit 1b Obey my command

<https://central.espresso.co.uk/espresso/coding/unit.html#/unit1b/lesson1>

Block coding > Unit 2a Different sorts of inputs

<https://central.espresso.co.uk/espresso/coding/unit.html#/unit2a/lesson1>

Thought shower

Thought shower a list of inputs and outputs, remembering to include things like 'QR codes', 'scanners' and 'sensors' as input devices, and physical control systems as output. Or, show pupils an iPad/tablet, and ask them to think about how they can give it information — swipe, touch, double touch, long touch,

pinch, tilting, speaking into the microphone, shaking, attaching a keyboard, etc. You could keep a working wall of input and output devices and add to this as you discover or identify new ones.

Walkabout

Go for a walk around the school and identify computers, particularly those embedded into other objects. Identify inputs and outputs (e.g. a movement sensor would provide input as part of the school alarm system.)

Function machine

To introduce the terms 'input' and 'output', draw a machine on the board with space for an input and an output. Decide on a 'rule' that the machine will apply; e.g. if you input a number, it multiplies it by 10, or adds 200.

Ask pupils to suggest a number to 'input' and give them the appropriate 'output' — repeat this several times and see whether they can guess the rule. They could also make up rules of their own, and test these out on each other.

This works well for maths, but could also be used across other curriculum areas — for example, if you input a word, it tells you the number of syllables in that word, if you input the name of a country, it will tell you the continent it is on, etc.

SEQUENCE

Related Discovery Education Coding unit:

Block coding > Unit 3a Sequence and animation

<https://central.espresso.co.uk/espresso/coding/unit.html#/unit3a/lesson1>

English and Maths offer plenty of opportunities to develop sequencing ability, such as when ordering numbers on a number line; sorting objects from lightest to heaviest, or recounting the events in a story in order.

In Discovery Education Coding, the concept of elapsed time is used to sequence actions. Give pupils plenty of practice at physically sequencing events first, before introducing this more abstract idea.

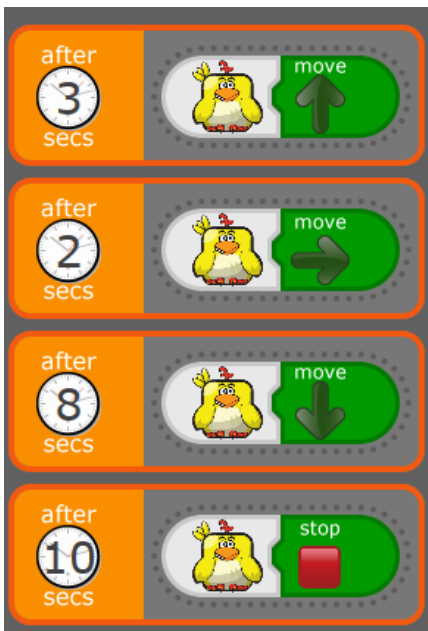
Stand in order

Challenge children to line up in a particular order; e.g. tallest to shortest, earliest birthday to latest birthday, etc. Add an extra element of challenge by seeing whether they can do this without talking.

Following instructions

Using one of the lessons in unit 3a, create and print out a screenshot of a set of code blocks like the ones below. Pupils can cut these out and then physically put them in order, according to instructions for what the bird will do and when (i.e. it will move to the right before it moves up).

Children could even act the movements out themselves.



Walking the playground

Challenge pupils to plan a route through the school playground, and then ask them to walk the route with a stopwatch to see how long each section takes. They can then use this information to write instructions (start at the gate and walk straight ahead, after 10 seconds turn left, after 15 seconds turn right, etc.) They could jumble these instructions up and give them to someone else to follow.

SELECTION (CONDITIONAL EVENTS)

Related Discovery Education Coding unit:

Block coding > Unit 3b Conditional events (selection)

<https://central.espresso.co.uk/espresso/coding/unit.html#/unit3b/lesson1>

Simon Says

Play a game of Simon Says — children must listen and follow your instructions, but only if you say 'Simon Says' first.

Getting if-y with it

Play any of the games (i.e. Robots, The 'Go!' game, Run robots, run!) using an if/then statement; e.g. 'If you have blue eyes, take two steps forward'; 'If you are a girl, clap three times'.

Extend this to include if/then/else statements; e.g. 'If you have blue eyes, take two steps forward, otherwise, sit down'; 'If you walked to school today, clap three times, otherwise put your hands on your head'.

REPETITION AND LOOPS

Related Discovery Education Coding unit:

Block coding > Unit 4b Repetition and loops

<https://central.espresso.co.uk/espresso/coding/unit.html#/unit4b/lesson1>

Getting ready

Write algorithms for brushing teeth, washing hands, combing hair, etc., which involve repeated actions. Include specific details about how many times these need to happen.

Animation detective

Watch some animated gifs or a front screen from a Discovery Education Espresso module. These tend to include 'forever' loops, so pupils could watch to spot the pattern of events and when the sequence starts again.

This could make a good precursor to planning and programming an animation of their own.

Song lyrics

Ask pupils to write the lyrics of their favourite songs, or look at some of these online. Identify examples of repetition, and ways of representing this; e.g. identifying the chorus, adding (repeat 2), (x3) at the end of lines, etc.

Choose a familiar song or poem with repeated lines, and challenge pupils to find an economical way to write this, using similar notation.

Dance routines

Ask pupils to work in groups to create a simple dance routine, where the same steps are repeated several times. (This could be a simple series of hand actions, if you don't have a lot of space). Talk about how repetition makes it easy to remember these routines, and show them how they could keep a record of the steps, such as by drawing a simple flow chart, with a repeated loop in it. They could pass instructions onto another group to follow and test to see whether they work.

VARIABLES

Related Discovery Education Coding unit:

Block coding > Unit 4a Introduction to variables

<https://central.espresso.co.uk/espresso/coding/unit.html#/unit4a/lesson1>

Crazy story generator

Introduce pupils to the idea of a 'crazy story generator', where different stories can be generated from a template, by filling in the blank spaces. For the story to work, the right kind of word (a colour, a number, an animal, etc.) needs to be chosen each time.

Each blank space has a label, to show what kind of word should be used. This is similar to naming a 'variable' in a program, which will hold a particular value.

You could ask pupils to think of a word to fill each space, and compare the different stories they end up with. Or, roll a dice each time to select a word from the lists given. You could also work backwards — give pupils several different versions of the same finished story, and ask them to identify what the 'variables' are, and give them names. After they've explored a pre-written story, pupils could try writing crazy stories of their own.

Supporting resources: *Crazy Story Generator – introducing variables* on **Page 12**.

UNDERSTANDING THE DIFFERENCE BETWEEN EVENTS, OBJECTS, AND ACTIONS.

Girls do / boys do

Divide pupils into groups that are easily identifiable, such as boys and girls; jumpers / no jumpers, etc. It is important that children know and understand which group they belong to before you start the game.

Play Robots (page 4), but give instructions that only apply to a specific group, e.g. girls stand up; non-jumpers clap your hands. The key teaching point here is that children should only follow the instruction

if it has been given to their group. Extend to include multiple instructions for different groups (e.g. girls hold up 5 fingers, boys put your hands on your head). You could also try combining this with Simon Says, The 'Go!' game or Run robots, Run!

Ship, Shore, Submarine

Teach children a game where they listen and perform certain actions when specific commands are called.

Ship — run to the centre of the room

Shore — run to one wall

Submarine — run to the other wall

Captain's coming — stand at attention and salute

Climb the rigging — make a climbing action

Scrub the deck — drop to your knees and scrub

You could also make up actions / phrases of your own.

After playing, write an algorithm using a set of statements following the same format: When..... the..... should..... (e.g. When I say 'ship' the children should run to the centre of the room).

This introduces a structure can then be used to help pupils form algorithms when they write their code. When (event) the (object) should (action).

EXTENSION IDEAS — TRANSITION TO KS3

Searching and sorting algorithms.

Cup sort

Give children five cards, or paper cups, with random numbers written on them. (e.g. 67, 18, 48, 4, 23), laid out in a straight line. Challenge them to sort these into order by taking it in turns to swap the positions of one pair of numbers. What is the least number of moves they can do this in?

Repeat this several times with different sets of numbers. Ask them to think about a general pattern or rule they could follow. What would they do differently if they could only swap pairs of numbers that are next to each other?

Searching for cards

Put children into teams of 2 – 4 and give each team a (shuffled) pack of playing cards. Challenge the teams to race each other to see who can be quickest to sort the cards so that each suit is together and

in the right order from Ace to King). Give them time to plan a strategy first, and after the race is over compare these to see which was most efficient. Repeat several times to give groups a chance to test out the different strategies suggested. As a variation, remove one card from each pack and challenge pupils to identify the card that is missing.

PATTERNS: SPELLING SORT ACTIVITY

CRAZY STORY GENERATOR – INTRODUCING VARIABLES

Story 1

One (*weather*) day, (*boy's name*) decided to take his (*animal*) on a trip to (*place*). They travelled by (*transport*) and it took (*number*) (*time*) to get there. On the way they snacked on (*food*) sandwiches and bottles of (*liquid*), and played (*game*). When they arrived, they went to a (*music act*) concert and rocked out until (*time of day*).

Story 2

Breaking news! Yesterday at (*time of day*), (*music act*) was attacked by an angry (*animal*) which had escaped from a nearby zoo. The (*animal*), which fled the scene, was found a few (*time*) later in a letterbox, where it is thought to have been sheltering from the (*weather*) conditions. By using some (*food*) as bait police were able to lure it out and return it to the zoo. After being taken straight to hospital by (*transport*) and being operated on by (*number*) doctors, (*music act*) is expected to make a full recovery. Tonight's concert in (*place*) has been cancelled, and anyone who has bought tickets is entitled to a refund of (*number*) pounds.

Variables on following page.

<p>Variable name: Weather</p> <ol style="list-style-type: none"> 1. balmy 2. grey and gloomy 3. windy 4. sunny 5. snowy 6. scorching hot 	<p>Variable name: transport</p> <ol style="list-style-type: none"> 1. train 2. rickshaw 3. unicycle 4. taxi 5. flying carpet 6. limousine 	<p>Variable name: liquid</p> <ol style="list-style-type: none"> 1. lemonade 2. washing up liquid 3. petrol 4. Milk 5. orange juice 6. melted chocolate
<p>Variable name: boy's name</p> <ol style="list-style-type: none"> 1. James 2. Abdul 3. Steven 4. Ayden 5. Sam 6. Josh 	<p>Variable name: number</p> <ol style="list-style-type: none"> 1. seven 2. four and three quarters 3. ninety-nine 4. six million 5. 3.14 6. a dozen 	<p>Variable name: game</p> <ol style="list-style-type: none"> 1. Hide and seek 2. I spy 3. Angry Birds 4. Minecraft 5. Hangman 6. Monopoly
<p>Variable name: animal</p> <ol style="list-style-type: none"> 1. caterpillar 2. frog 3. pig 4. triceratops 5. giraffe 6. whale 	<p>Variable name: time</p> <ol style="list-style-type: none"> 1. nanoseconds 2. years 3. days 4. seconds 5. millenia 6. hours 	<p>Variable name: music act</p> <ol style="list-style-type: none"> 1. One Direction 2. Lady Gaga 3. Justin Bieber 4. Sia 5. Elton John 6. Ed Sheeran
<p>Variable name: place</p> <ol style="list-style-type: none"> 1. The Houses of Parliament 2. Canada 3. Disneyland 4. Australia 5. New York 6. Mars 	<p>Variable name: food</p> <ol style="list-style-type: none"> 1. gravy 2. chicken 3. Spaghetti bolognaise 4. museli 5. bananas 6. fish fingers 	<p>Variable name: time of day</p> <ol style="list-style-type: none"> 1. midnight 2. tea-time 3. 6am 4. morning 5. 2 o'clock 6. sunset