

Coding Introduction

Overview

In this lesson, we introduce students to the beginner concepts of coding. This can be used as an introduction to coding, or the first steps into a broader exploration of coding as a topic.

Students play a game of 'simon says' to understand that computers follow coded instructions. We motivate students to learn about coding, and demonstrate how to code a character to move using coding blocks. Students participate in coding puzzles to apply these skills. Then the concept of loops is introduced and related back to the puzzles and the 'simon says' activity. With our new understanding of coding we brainstorm with students on how coding could be applied to the problems and challenges of Future Earth.

Minimum timing in the lesson can be applied strictly to achieve a single lesson, or activities can be fully explored to make up many individual lessons.

Outcomes

- Understand that computers follow coded instructions.
- Students are motivated to learn about Coding.
- Recall and apply the rules of pair programming.
- Understand what loops are and how they are useful.
- Apply coding concepts to the topic: Future Earth

Resources Required

Computers

Internet Access

Projector / Smart Board

Pedagogical approaches

DI	Direct Instruction	SGW	Small Group Work	PS	Problem Solving
DIS	Discussion	RP	Role Play		
CL	Cooperative Learning	WTG	Writing		

These pedagogical approaches are drawn from: Killen, R. (2007). Effective teaching strategies: Lessons from research and practice (4th Ed.). Australia: Thomson Social Science Press.

Focused Learning Episode Plan

Time	Outcome	Pedagogy	Topic	Activities
			Setup	<p><u>Student Computers</u> Get the students computers set up by going to this internet address: https://studio.code.org/hoc/1</p> <p><u>Teacher Computer</u> On the teacher computer connected to the projector, open up these internet links: https://studio.code.org/hoc/1 https://www.youtube.com/watch?v=vqkahOzFH2Q</p>
8-20 mins	Understand that computers follow coded instructions.	DI PS RP	What is Coding?	<p>Collect the students and guide them to sit on the floor as a group.</p> <p>Ask the students what they know about coding. Take their suggestions without defining coding as we will explore this.</p> <p>Play a game of Simon Says to help illustrate the idea of coding and following specific instructions. Explain the rules if you feel it's necessary. Don't forget to play the game properly and try to trick some students by omitting "Simon Says" and generally making mistakes. This is important so that you can later point out that instructions must be coded.</p> <p><i>Simon says...</i></p> <ul style="list-style-type: none"> - <i>Put your hands on your head</i> - <i>Put your left hand up</i> - <i>Put your left hand down</i> - <i>Blink your eyes</i> - <i>Touch the floor</i> - <i>Etc</i> <p>Now start to incorporate instructions more relevant to the puzzles we will be doing shortly (movement, turning, repeating).</p>

Time	Outcome	Pedagogy	Topic	Activities
				<p><i>Simon says...</i></p> <ul style="list-style-type: none"> - <i>Move one step forward</i> - <i>Move one step backward</i> - <i>Turn to your right</i> - <i>Turn to face teacher "Name"</i> - <i>Move one step forward and then turn to the left</i> - <i>Face the front</i> - <i>Do this <u>three</u> times, take a step forward</i> <p>You may need to laugh with the class at the confused attempts here. If need be, explain that you wanted them to take a step forward but repeat it three times. Then try this instruction again.</p> <p><i>Simon says...</i></p> <ul style="list-style-type: none"> - <i>Do this <u>three</u> times, take a step forward</i> <p><i>Does Simon Says work for a computer?</i></p> <p>Get a volunteer to come up to the projector screen to try:</p> <p><i>"Try saying this instruction, 'Computer, beep loudly'.</i></p> <p><i>Hmm, maybe we should get the whole class to try together. Class, say 'Computer beep loudly'</i></p> <p><i>Nothing happened! Something's not quite right here. What's wrong?"</i></p> <p><i>"Coding is like playing Simon Says with a computer.</i></p> <p><i>We can give a computers instructions to do things, and the instructions must be coded in a special way for the computer, or else the computer will not do it."</i></p> <p>Segway into a language analogy.</p> <p><i>"Who speaks a different language at home? What language do you speak? Or who knows someone who speaks another language?</i></p> <p><i>Just like with other languages, it is hard to understand if we don't speak a language, that's why the computer couldn't understand when we talked to it in English."</i></p> <p><i>"Coding is the language of computers."</i></p>

Time	Outcome	Pedagogy	Topic	Activities
3-9 mins	Students are motivated to learn about Coding.	DIS	Motivation Coding	<p>Motivate students by relating coding to games: <i>"Who plays games or uses apps? What games do you play? What apps do you use?"</i></p> <p>Discuss and engage with students, especially if you have any common ground. <i>"Yeah I play that too. You play clash of clans? I love that game, I am level... That game sounds great, I will have to download it on my phone. Who plays minecraft? Wow, lots of minecraft players!"</i></p> <p>Relate these games and apps learning to code: <i>"All those games that you play, they didn't just appear. Someone created them. If we learn to code, we can make games and apps like that too. "</i></p> <p><i>"If you have ever played a game and thought 'this game would be better if...' or 'I wish there was a game like this.' If you learn to code, you can make those games for your family and friends to enjoy."</i></p>
8-20 mins	Understand that computers follow coded instructions.	DI DIS PS	Demo Coding	<p>Have open this link on your projector for the class to see: https://studio.code.org/hoc/1</p> <p><i>"Now while we are not making (some game they mentioned) today, I guarantee you that the people that created those games had to learn to code with the basics just like you will starting today."</i></p> <p>Point out and explain the sections:</p> <ul style="list-style-type: none"> - These blocks are your instructions, like in "Simon Says" - Here is the "workspace", this is where you put your block instructions. <p>Demonstrate the first three puzzles:</p> <p><u>Puzzle 1</u> <i>"The goal is to get the bird to the piggy. So far we have 'When run, move forward'. What do we think, that sounds right? Let's try it. Click Run."</i></p>

Time	Outcome	Pedagogy	Topic	Activities
				<p>Let the program fail on purpose for this demonstration. <i>“What have we done wrong? Computers are very picky, they need exact instructions. You can’t say ‘move forward a little bit’. Computers need to know exactly how many times to move forward. How can we fix this code? Yes, we need another move forward block.”</i> Demonstrate now how to drag a block in and “click” it into place.</p> <p><u>Puzzle 2</u> Let students try to tell you how to do this puzzle. Follow their instructions, but leave the third move forward block just slightly out of place so that it’s not connected. <i>“Is this good to run?”</i> If students say it is OK then run it and let it fail. Then ask students how to correct the mistake. <i>“Why didn’t it work? Because the block was not clicked in. You need to hear that ‘click’.”</i></p> <p><u>Puzzle 3</u> <i>“What’s different about this puzzle?”</i> Students will reply with either “different blocks”, “TNT”, or “there is a turn”. <i>“Should we touch the TNT? No!”</i> As students guide you through the puzzle, when they get to the turn, give a helpful thinking strategy regardless of whether they use the correct block. <i>“It’s really interesting you said ‘move forward’ (or ‘turn left’ or ‘turn right’). It’s important to know, the bird will move forward in the direction of its eyes. If you get stuck, what you can do is stand up, pretend to be the bird, and think like a robot. Think where the bird’s eyes are looking and pretend you are the robot bird and decide which way you would need to turn, left or right, if you were the bird.”</i> Students may also fail here at the end by missing the last “move forward” after the turn. Let them fail and then discuss how to fix their code.</p>

Time	Outcome	Pedagogy	Topic	Activities
4-10 mins	Recall and apply the rules of pair programming.	DI	Pair Programming	<p>You may want your students to work as a team of two on the one computer. If so, use this activity to introduce the concept of pair programming.</p> <p>Display this video on pair programming: https://www.youtube.com/watch?v=vgkahOzFH2Q Stop video at 1:14</p> <p>Recap the roles.</p> <p>Driver:</p> <ul style="list-style-type: none"> - Controls the mouse and keyboard. <p>Navigator:</p> <ul style="list-style-type: none"> - Watching for mistakes. - Giving suggestions. - The only person allowed to stand up and be the robot when you get stuck. <p>Encourage students that the navigator gets up and pretend to be a robot to help solve the problems.</p> <ul style="list-style-type: none"> - Not to touch the keyboard or mouse <p><i>"You will be swapping roles every two puzzles."</i></p>
20-60 mins	Understand that computers follow coded instructions.	SGW PS	Coding Puzzles	<p>During setup, student computers should have been opened to: https://studio.code.org/hoc/1 You will need student computers at this internet address to conduct this activity.</p> <p>Introduce an expectation about appropriate handling of the computers. For example: <i>"Are we going to use the computers like this (big smashing motions) or like this (posh gentle motions)?"</i></p> <p>Students are sent to sit at computers, either in pairs or individually.</p> <p>Let the students play through the puzzles and offer support and guidance as necessary. Remind students about driver/navigator roles if their behaviour requires.</p>

Time	Outcome	Pedagogy	Topic	Activities
9-25 mins	Understand what loops are and how they are useful.	DI DIS	Loops	<p>As students were doing the puzzles, they would have encountered a “repeat” block. In this activity, discuss and explain repeats as a “loop”.</p> <p>Define a “Loop”: <i>“The action of doing something over and over again.”</i></p> <p>Guide their thoughts by providing some examples where they might find loops. For differentiation, alternate the between these descriptions: “loops”, “do something over and over”, and “repeat.”</p> <p><i>Examples:</i></p> <ul style="list-style-type: none"> - <i>Something that they do <u>over and over again</u>.</i> - <i><u>Loops</u> that other people do.</i> - <i>Actions that machines <u>repeat</u>.</i> - <i>Something that animals do <u>over and over again</u>.</i> - <i><u>Loops</u> in nature.</i> - <i>Jobs that do <u>repetitive</u> tasks.</i> <p>Ask the students to provide examples of a loop from their lives, or the lives of their friends and family, or in the world around them. Here’s some ideas;</p> <ul style="list-style-type: none"> - Dance moves - Skipping - Practicing something - Brushing teeth - Planet orbits - Day/night - Job: “Lollipop person” - Eating/chewing - Bees collecting nectar/pollen <p>Can students think of a ‘Future Earth’ loop?</p>

Time	Outcome	Pedagogy	Topic	Activities
				<p>Now relate this concept back to the Simon Say's game where you asked students to do a repeated command, "Do this <u>three</u> times, take a step forward". That was a loop.</p> <p>Finally bring it all back to the puzzles students did where they got the character to repeat an instruction.</p> <p>If you have time at this point and students had found the puzzles with repeats in them difficult, then give students some time to do these puzzles again now that they understand loops better.</p>
10-25 mins	Apply coding concepts to the topic Future Earth	DIS	Future Earth	<p>Now that students have learnt some basics of coding, let's apply it to the topic of "Future Earth".</p> <p>Remind students about some of the Future Earth concepts students may have been learning in school. Eg sustainability, human health, resources for everyone, etc.</p> <p>Using our new understanding of coding, brainstorm with the class how the power and usefulness of coding can be applied to Future Earth applications.</p> <p>Here are some examples:</p> <ul style="list-style-type: none"> - Solar panels are most efficient when they are pointing at the sun. Coding can be used to move the panels so that they track the sun's motion to be as efficient as possible. - Delivering water or food to everyone on the planet is tricky and coding can be used to calculate the systems needed to distribute resources on a global scale. - As human health problems in the future get solved with mechanical implants, for example an artificial heart, these devices will be running on code. - Building the cities of the future will be modelled using coding of big data.

Time	Outcome	Pedagogy	Topic	Activities
5-15 mins		DIS	Synthesis	Bring students back to the floor as a group. Review and discuss the concepts learned today: <ul style="list-style-type: none">- Computers follow coded instructions- Motivations for coding- Coding blocks- Using blocks to solve a puzzle- Loops- Future Earth applications of coding
Extension 30-60 mins			EXTENSIONS	<u>Extension work:</u> If you still have time remaining or wish to practice these concepts further: Send students to do some more coding puzzles, which are “Star Wars” themed: https://studio.code.org/s/starwarsblocks/stage/1/puzzle/1