



# CODE WITH SCRATCH

FROM BEGINNER TO INNOVATOR:  
A 2-YEAR PATHWAY



## WHAT IS SCRATCH?

Scratch is a visual-based programming language targeted primarily at people who have little to no experience with coding. In our Scratch course, you will learn how to program simple games, animations, and much more.

## WHY SCRATCH?

Scratch is a great way for students to develop their understanding of fundamental computing concepts. Its drag-and-drop coding blocks also removes the immediate need for newcomers to pay attention to fine details, such as proper syntax and background technical knowledge. Scratch's easy-to-use and brightly-coloured coding blocks makes it a powerful, accessible, and user-friendly tool to get started with coding.

## OVERVIEW OF OUR SCRATCH COURSE

Below is an overview of our 2-year Scratch programming pathway. It is made up of four 8-week courses to be delivered across four school terms.

Each term covers a theme: past examples of themes include **Art and Animation**, and **Space and Exploration**.

We build new projects each term that fit into one of following three categories:

- Computer games
- Interactive artwork
- Animations

# BEGINNER SCRATCH COURSE

Our Beginner Scratch course is a great place for those who have little to no experience with programming. There are four Beginner Scratch courses, over the course of which we learn the following fundamental programming concepts:

Commands	Positive and negative numbers	Maths (addition, subtraction, equality)
Sequencing	Animations	Simple Boolean logic
Iteration (loops)	Collision detection	Testing and debugging
Variables	Branching (decision making)	
XY coordinates		

ART AND ANIMATION	SPACE AND EXPLORATION	SPORTS AND GAMES	FANTASY AND STORIES
Use technology as a tool for creative design, artistic expression, and making animations.	Explore the wonders of space through game design and creative graphics.	Inspire yourself to create your own games and traditional games.	Create imaginary worlds and characters. Design your universe, your story, and your hero.
<ul style="list-style-type: none"> <li>❖ What is coding?</li> <li>❖ Intro to commands</li> <li>❖ Sequencing</li> <li>❖ Simple maths operations</li> <li>❖ XY coordinates</li> <li>❖ Positive and negative numbers</li> <li>❖ Trial and error</li> <li>❖ Creating animations &amp; graphics</li> </ul>	<ul style="list-style-type: none"> <li>❖ Writing linear algorithms</li> <li>❖ Branching: 'if' block</li> <li>❖ Collision detection: using the "touching" block</li> <li>❖ Debugging code: correcting errors</li> <li>❖ XY coordinates</li> <li>❖ Positive &amp; negative numbers</li> </ul>	<ul style="list-style-type: none"> <li>❖ Designing AI for computer games</li> <li>❖ Iteration: 'forever' vs 'repeat' blocks</li> <li>❖ Data storage: using variables to keep score</li> <li>❖ Using random numbers</li> <li>❖ Simple Boolean logic</li> </ul>	<ul style="list-style-type: none"> <li>❖ Problem decomposition: breaking it down into small achievable steps</li> <li>❖ Adding sound effects</li> <li>❖ Creating buttons that trigger actions</li> <li>❖ Asking for and using user input</li> <li>❖ Graphic design</li> </ul>

# INTERMEDIATE SCRATCH COURSE

Our Intermediate Scratch course is designed for students who are confident using the skills outlined in the aforementioned Beginner course. There are four Intermediate Scratch courses, over the course of which we learn the follow programming concepts:

- |  |   |
|--|---|
| Branching<br>(if blocks vs. if-else blocks)                        | Player dashboards                                   |
| Iteration<br>(forever vs. repeat vs. repeat until)                 | Controlling objects in 2D space<br>(XY coordinates) |
| Maths<br>(angles, geometry, random number generation, comparisons) | Designing AI algorithms                             |
| Joining strings  | More complex Boolean logic                          |
| User input box   | Gravity effects and trajectory physics              |
|  | Multi-player and multi-level game design            |

SCIENCE AND FUTURE	IMAGINARY WORLDS	GAME DESIGN	ARCADE CLASSICS
Using technology as a tool for creative design, artistic expression, and making animations.	Create your own imaginary world and creatures. Design the environment and the characters that live in it.	We love playing computer games! Now, design your own to understand what happens under the hood.	Let's discover classic arcade games. Learn how to build classics like Pong, Tank Wars, Space Invaders, and Breakout!
<ul style="list-style-type: none"> <li>❖ Algorithms: linear and branching</li> <li>❖ Animating &amp; graphic designs</li> <li>❖ Creating sound effects</li> <li>❖ Testing and debugging</li> <li>❖ Creating own sprites and backgrounds</li> <li>❖ Player input box</li> <li>❖ Joining strings and variables</li> </ul>	<ul style="list-style-type: none"> <li>❖ Arithmetic: simple maths operations</li> <li>❖ Problem decomposition</li> <li>❖ Data inputs and outputs</li> <li>❖ User inputs</li> <li>❖ Boolean logic in more complex applications</li> <li>❖ Branching: if vs if-else</li> <li>❖ Iteration: forever vs repeat vs repeat until</li> </ul>	<ul style="list-style-type: none"> <li>❖ Broadcast messages</li> <li>❖ Player dashboards</li> <li>❖ Creating multi-player controls</li> <li>❖ Creating multi-level games</li> <li>❖ Creating sound effects</li> <li>❖ Designing AI algorithms</li> <li>❖ Game design elements</li> </ul>	<ul style="list-style-type: none"> <li>❖ Iterative design, from a simple prototype to evolving the design</li> <li>❖ Arithmetic: comparisons</li> <li>❖ Using variables for data inputs &amp; outputs</li> <li>❖ Designing AI algorithms</li> <li>❖ Gravity effects &amp; trajectory physics</li> </ul>