

FOR SCHOOLS

Coding, AI and Robotics for ages 6 to 12

A complete curriculum your students work through in the browser — with progress you can see, and certificates that carry your school's name.



4
COURSES

103
LESSONS

6–12
AGES

0
SETUP

Why schools come to us

Coding, AI and robotics are being asked for. The usual options are a teacher building a course from nothing, or an outside tutor charging per child — and a video playlist proves nothing at all.

Completion is earned, not clicked

We measure the seconds a child actually watched. Skipping ahead earns nothing, so a certificate means the lessons were genuinely covered.

Your name on every certificate

Your logo and signatory print alongside Coderboy's, and each certificate carries a number anyone can verify online.

Safeguarding by design

Children get no accounts and no email addresses. A public certificate shows only a first name and an initial.

Nothing for staff to install

It runs in a browser on the computers you already have. No software, no licences, no hardware.

The curriculum at a glance

COURSE	AGES	LESSONS	STATUS
Scratch	6–10	25	Available
Python — Part 1	7–10	26	Available
Python — Part 2	9–11	24	Available
Artificial Intelligence	8–11	28	Available
Robotics with Arduino	8–11	24	Planned

Every lesson is printed in full from page 4 onwards.



Start with a pilot. One class, one course, an agreed number of weeks. You see real completion data and real certificates before committing to anything wider.

How it works

#	STEP	WHO
1	We set up your school and your classes	Coderboy
2	Teacher pastes the class register	Teacher, 5 minutes
3	Students work through lessons in class or at home	Students
4	Teacher watches progress and exports a report	Teacher
5	Certificates issue automatically on completion	Automatic

What the teacher sees

- The whole class on one screen — lessons complete, percentage, minutes watched
- Who has stalled, and when each student was last active
- **CSV export** for the register, a department report or a parents' evening
- A class **join code**, so students enrol without needing email addresses

What you need – and do not

You need

Browsers and internet · a supervising teacher · a period a week, or homework time

You do not need

To buy software or hardware · a specialist coding teacher · to write any curriculum

Data and safeguarding

- Children have **no login, no email address and no password**.
- We store a first name, progress and a certificate record. Nothing else.
- Public certificate checks show **a first name and an initial only**.
- Records deleted on request; a data-processing agreement is available.



Let's start with one class

A twenty-minute call to pick a year group and a starting course. We'll follow it with a short written proposal for your school — scope, timeline and cost.

hello@ibranext.com · fanwaar.com/coderboy

Every lesson, in order

Lessons run three to five minutes. Courses are listed in the order a child would take them.



Scratch AGES 6–10 · 25 LESSONS · AVAILABLE

Drag-and-drop coding. The first taste of logic, loops and events — no typing required.

- | | |
|--------------------------|------------------------------|
| 1 Welcome to Scratch | 14 Events Blocks — Part 1 |
| 2 Create Your Account | 15 Events Blocks — Part 2 |
| 3 Scratch Basics | 16 Your First Animated Story |
| 4 What is Coding? | 17 Balloon Pop-Up Game |
| 5 X and Y Coordinates | 18 Jumping Game |
| 6 Coordinates Demo | 19 Ping Pong Game |
| 7 Sprite Costumes | 20 Catch Game |
| 8 Motion Blocks — Part 1 | 21 Space Invader Game |
| 9 Motion Blocks — Part 2 | 22 Boat Game |
| 10 Sprite Sounds | 23 Snake Game |
| 11 Sound Blocks | 24 Football Penalty Shootout |
| 12 Looks Blocks — Part 1 | 25 Coding Success Stories |
| 13 Looks Blocks — Part 2 | |



Python — Part 1 AGES 7–10 · 26 LESSONS · AVAILABLE

Real typed code in the browser. Printing, variables, loops, lists and nine build-it projects.

- | | |
|---|--|
| 1 What Is Python? | 14 The while Loop — Repeat Until a Condition Changes |
| 2 Python Editor Tour — How to Run Python Code | 15 String Tricks — Join, Repeat and len() |
| 3 What Is a Program? | 16 Lists — Indexing, append() and for Loops |
| 4 The print() Function — Your First Python Code | 17 Random Numbers — randint() & choice() |
| 5 Variables — Store Values in Python | 18 Project: Greeting Story |
| 6 Numbers and Strings — Data Types Explained | 19 Project: Number Guessing Game |
| 7 Math Operators Part 1 — Add, Subtract, Multiply, Divide | 20 Project: Rock Paper Scissors |
| 8 Math Operators Part 2 — Remainder, Power & Order | 21 Project: Mad Libs Story Generator |
| 9 The input() Function — Ask the User a Question | 22 Project: Simple Calculator |
| 10 The int() Function — Convert Text to Number | 23 Project: Multiplication Tables |
| 11 The if Statement — Decisions and Indentation | 24 Project: Quiz Game |
| 12 if, else, elif — Comparison Operators | 25 Project: Password Strength Checker |
| 13 The for Loop — range() and Repeating Code | 26 Project: Tic-Tac-Toe |

Python – Part 2 AGES 9–11 · 24 LESSONS · AVAILABLE

Functions, dictionaries, files and error handling — the step from scripts to programs.

- | | |
|--------------------------------------|--|
| 1 Welcome to Python Part 2 | 13 String Tricks (upper, lower, split, join) |
| 2 Quick Recap of What You Know | 14 What is a File? |
| 3 What is a Function? | 15 Reading a File |
| 4 Making Your Own Function — Part 1 | 16 Writing to a File |
| 5 Making Your Own Function — Part 2 | 17 Errors and Why They're Helpful |
| 6 Functions with Inputs (Parameters) | 18 try and except |
| 7 Functions that Give Back (return) | 19 Hangman |
| 8 What is a Dictionary? | 20 Word Counter |
| 9 Reading and Writing Dictionaries | 21 Mini Chatbot (no AI) |
| 10 Looping Over a Dictionary | 22 Quiz with Saved Scores |
| 11 Lists of Dictionaries | 23 Text Adventure Game |
| 12 f-strings — Smart Strings | 24 Build Your Own (Capstone) |

Artificial Intelligence AGES 8–11 · 28 LESSONS · AVAILABLE

What AI actually is, how it learns from examples, where it gets things wrong, and how to use it well.

- | | |
|--|--|
| 1 You Already Use AI | 15 Giving the AI a Role |
| 2 Two Kinds of Problems | 16 AI is Just a Super Guesser |
| 3 How AI Learns from Examples | 17 AI's Safety Fences |
| 4 Train Your Own AI! (Teachable Machine) | 18 Garbage In, Garbage Out |
| 5 When AI Gets It Wrong | 19 Homework Helper, Not Homework Doer |
| 6 Why YouTube Knows What You Like | 20 Make Your Own Storybook with AI |
| 7 The AI Family — AI, ML, DL | 21 Train an AI to Know Your Toys (Teachable Machine — sound/gesture) |
| 8 Meet the AI Helpers | 22 Build a Vocabulary Helper |
| 9 Staying Safe with AI | 23 Make AI Quiz You for a Test |
| 10 The Magic Word — Prompt | 24 AI Drawing Game (you draw, AI guesses) |
| 11 Asking Great Questions — Part 1 | 25 Plan Something Fun with AI |
| 12 Asking Great Questions — Part 2 | 26 Spot the AI Mistake |
| 13 Asking AI to Make Things — Part 1 | 27 Be the Boss — Make a Mini Comic |
| 14 Asking AI to Make Things — Part 2 | 28 A Real AI Tool, Made by Real People (WriteTone) |



Robotics with Arduino AGES 8-11 · 24 LESSONS · PLANNED

LEDs, motors and buttons in a browser simulator — real hardware logic with no kit to buy.

- | | |
|--------------------------------------|-------------------------------------|
| 1 Welcome to Arduino | 13 for loop in Hardware — Part 2 |
| 2 Open Your Hardware Lab | 14 Buttons and digitalRead — Part 1 |
| 3 The Wokwi Editor Tour | 15 Buttons and digitalRead — Part 2 |
| 4 What is a Microcontroller? | 16 Build a Traffic Light |
| 5 The LED | 17 Push Button Lamp |
| 6 The Resistor — One Sentence | 18 Reaction Time Game |
| 7 Pins and Power | 19 Morse Code Sender |
| 8 digitalWrite HIGH and LOW — Part 1 | 20 Light Dimmer (PWM) |
| 9 digitalWrite HIGH and LOW — Part 2 | 21 Temperature Display |
| 10 delay() — Part 1 | 22 Simple Servo Wave |
| 11 delay() — Part 2 | 23 LED Dice |
| 12 for loop in Hardware — Part 1 | 24 Mini Synthesizer |