A PARENT'S FIELD GUIDE

# The Coding *Starter Kit*

Everything a parent needs to start a child in coding without guesswork: the honest age map, the readiness signals, the first ten projects, the free tools, and the checklist for judging any class, including ours.

**Modern Age Coders** · live coding & maths classes, ages 6-65
learn.modernagecoders.com · rated 4.9 across 547 Google reviews

# The honest age map

Most guides answer "what age?" with whatever the guide's author sells. This map includes the rows that lose us business, because they are true.

| AGE | RECOMMENDATION | WHY |
|---|---|---|
| **4-5** | Not formal classes yet | Puzzle play, pattern games and being read to build the pre-coding brain better than any curriculum. Wait. |
| **5-6** | Borderline: picture-blocks IF signals are strong | Some children are ready; most gain more by waiting a year. Judge by the signals on the next page, not the birthday. |
| **6-9** | The sweet spot: Scratch, taught for ideas | Reading arrives, imagination peaks, and block coding carries real computer science without typing in the way. |
| **9-12** | Either door: fast Scratch arc, or straight to Python | Typing and abstraction mature. The single best age range for a text-code start. |
| **13-16** | Python directly, real projects fast | Teens need respect: a real language with visible results. Blocks feel babyish, and rightly so. |
| **17+** | Start anywhere, anytime | Motivation replaces the head start. There is no closing window; adult beginners prove it weekly. |

> **The one-line summary:** earlier starts buy pressure-free runway, not superiority. Late starters with appetite catch up startlingly fast. Starting before readiness is the only real mistake.

# The six readiness signals

Worth more than age, every time. Tick what you see; four or more usually means ready now.

- [ ] **Asks how things work.** "How do they make games?" is the strongest single signal at any age.
- [ ] **Builds and tinkers.** LEGO, Minecraft, crafts, dismantled remote controls: the constructor instinct converts directly into programming.
- [ ] **Tolerates not-working-yet.** Can they retry a fallen tower without melting down? Debugging is that, daily. (Good teaching also trains this.)
- [ ] **Reads fluently enough for the tool.** Scratch needs early reading; Python needs comfortable reading plus some typing. The tool follows fluency, not age.
- [ ] **Holds a 20-minute focus on chosen things.** On chosen things is the key phrase. A child who locks onto their own projects will thrive in an interactive class hour.
- [ ] **Wants it, or would if shown.** Dragging an uninterested child to coding builds resentment, not skill. A single trial class answers this at zero cost.

> **Fewer than four ticks?** Wait three to six months and feed the signals instead: building toys, puzzle games, and "how do you think that works?" conversations. Readiness grows fast at these ages.

# The first ten projects

Projects beat exercises: each one below teaches a real idea while producing something a child wants to show off. Tick them off as they ship.

## In Scratch (ages 6-10)

- [ ] **Animated greeting card** · sprites, looks, sounds
- [ ] **Dancing pet** · loops and rhythm
- [ ] **Chase game** · events, conditionals, score variable
- [ ] **Quiz game** · questions, variables, if/else
- [ ] **Two-level platformer** · broadcasting, cloning, win state

## In Python (ages 9-14)

- [ ] **Mad-libs story machine** · input, strings, print
- [ ] **Number guessing game** · while loops, comparisons
- [ ] **Dice argument settler** · randomness, 10,000 rolls, probability
- [ ] **Flash-card quizzer** · lists, dictionaries, score tracking
- [ ] **Turtle art generator** · geometry, angles, loops that draw

> **The house rule that makes these work:** the child types and the adult (or mentor) only asks questions: "what do you think this line does?" Copy-paste teaches copying; predicting teaches programming.

# The free tools shortlist

Everything on this list costs nothing. If a program tries to sell you software before teaching, that tells you something.

| TOOL | AGES | WHAT IT IS |
|---|---|---|
| **ScratchJr** | 5-7 | Picture-block coding on tablets, made for pre-readers. Gentle, wordless, brilliant. |
| **Scratch** | 6-12 | MIT's block language: real loops, events and logic inside games and stories. Runs free in any browser. |
| **MakeCode for Minecraft** | 7-11 | Block coding that commands the Minecraft world, for families who already own the game. |
| **Python + Thonny** | 9+ | The real language taught everywhere from schools to labs, with Thonny as the friendliest free beginner editor. |
| **Roblox Studio** | 9-14 | The professional-grade (and free) editor behind Roblox, scripted in the Luau language. The development tool, distinct from the social game. |
| **VS Code** | 12+ | The free editor most working programmers use. Teens graduate here naturally. |

> **Safety note for the game platforms:** creation tools (Roblox Studio, Minecraft coding) are separate from public multiplayer. Building is solitary or supervised; it is public servers and open chat that need parental controls, set those once, and separately.

# How to judge any coding class

Take this checklist into any trial class, anywhere, including ours. Six things to watch for in the hour:

- [ ] **Who talks more?** In a good class the child talks, builds and explains more than the teacher. Lecture-heavy hours are a warning sign.
- [ ] **Questions before answers?** Watch whether the teacher asks "what do you think happens?" before showing. That one habit predicts everything else.
- [ ] **Does the level adjust live?** Too easy should get harder within minutes; too hard should get gentler. Scripts cannot do this. Teachers can.
- [ ] **How are mistakes treated?** Errors should become the most interesting moments of the hour, examined with curiosity, never rushed past.
- [ ] **Is your child lighter afterward?** The walk-away test: more confident and talkative about the subject than they entered.
- [ ] **Was the placement honest?** A read that flatters everything is a sales script. An honest one names a real starting point, and sometimes says "wait".

> **Also ask any provider:** Is every class recorded? Does the same teacher stay with my child? Is billing monthly with no lock-in? "No" to any of these is worth noticing.

# The screen-time peace treaty

The hours are being spent either way. Five house rules that convert consumption into creation, from families who made it work:

### 1 · Name the two kinds of screen time

Watching and making. The vocabulary alone changes negotiations: "you have had lots of watching time, want some making time?"

### 2 · Maker time is never the punishment

Never remove building time as a consequence; it teaches the child that creating is a privilege of mood. Protect it like you protect reading.

### 3 · Ship on Sundays

A weekly five-minute family demo of whatever got built, however small. An audience is the strongest motivation engine ever discovered.

### 4 · The adult asks, never grabs

When helping, hands off the mouse and keyboard. Questions only. Slower tonight, transformative by next month.

### 5 · Follow the obsession, do not fight it

Minecraft obsession becomes Minecraft coding; Roblox hours become Studio hours; YouTube fandom becomes a first video script. Redirect the current, do not dam it.

THE ROAD AHEAD

# From first sprite to first job offer

The whole ladder, so you know where any starting rung leads:

| | |
|---|---|
| **Ages 6-9** | Scratch: ideas inside games and stories |
| **Ages 9-12** | Game building (Roblox, Minecraft) and first text code |
| **Ages 11-14** | Real Python: apps, art, AI literacy |
| **Ages 14-18** | Deeper tracks: web, AI/ML, AP Computer Science, competition coding |
| **Beyond** | College and professional tracks, portfolios that open doors |

## About Modern Age Coders

We teach live, genuinely interactive coding and mathematics classes, one full hour, twice a week, with a dedicated mentor who stays with your child. Every mentor teaches both subjects; every class is recorded; billing is monthly with no lock-in. Rated 4.9 across 547 Google reviews.

**Every student starts with a free trial class:** a real one-hour lesson plus an honest placement read, no card details, no obligation. If this kit was useful, the trial is the natural next page.

**learn.modernagecoders.com/free-trial**

WhatsApp: +91 91233 66161