Do kids still need to learn coding now that AI writes code?

Peabody blog · AI for kids · updated 2026-11-06

Yes, but not the typing. Keep the six ideas behind code and the five skills of working with an AI, and skip syntax drills. What to do at 4 to 7, 8 to 11 and 12 to 14.

Yes, kids still need to learn coding, but not the typing. Keep the ideas behind code (order, cases, repeating, remembering, events and breaking a big thing into small ones) and the skills of reading, describing and testing what a program does. Skip the drills on brackets and semicolons. An AI can write the syntax. It can't know what your child meant, and it can't tell them whether the result is right. Those two jobs stay with the person. Peabody's coding is free through the end of 2026, and there is a hub page for grown-ups that gathers everything below in one place.

Why the ideas still matter

Say a child asks an AI, "make a game where a dragon chases me." They get a game, and it is only as good as their picture of what a game is. A child who has built even one loop by hand asks the next questions on their own: what happens when it catches me? Does it get faster each level? What if two dragons touch? Those are if and for questions said in plain English.

A child who has never seen that picture writes a vague ask, gets something odd, and can't say why it is broken. The ideas are what let them steer.

The six ideas

- Idea: Sequence · What it teaches: Order matters. · The question to ask in kid words: "What happens first?"
- Idea: Conditionals (if) · What it teaches: Thinking in cases. · The question to ask in kid words: "What should happen when... and when not?"
- Idea: Loops (repeat, for) · What it teaches: Spotting a pattern and saying it once. · The question to ask in kid words: "Is this the same thing again and again?"
- Idea: Variables and state · What it teaches: Something remembers, and it changes. · The question to ask in kid words: "What does the game need to remember?"
- Idea: Events · What it teaches: Things happen because of other things. · The question to ask in kid words: "What starts it?"
- Idea: Decomposition · What it teaches: Big into small, each piece easy to check. · The question to ask in kid words: "What are the pieces?"

You don't need a screen for any of them. Making a sandwich is a sequence. "If it's raining, boots" is a conditional. Brushing teeth, two minutes, is a loop.

The five skills that matter more now

1. Read and judge. Look at a program, or watch it run, and say "that's wrong, and here is why." Finding what is off is worth more than typing it out.
2. Say it exactly. Describe what you want so precisely that nobody could misread it: the rules, the odd cases, what "done" means. This is the new way of writing code, and it is conditionals and loops in plain words.
3. Try to break it. Try zero. Try it twice. Try the silly input. Children love this, because it is breaking things on purpose.
4. Predict, then run. Guess what a change will do before you see it. It is the fastest way to build a picture of how a program works.
5. Work with an AI partner. Ask, look, notice what is off, ask better. The first answer is a draft.

What to do at each age

Ages 4 to 7: no screen needed

Play the ideas. Give directions to a "robot" (a grown-up who only does exactly what is said) and watch what happens when you leave a step out. Sort things by "if it's red, here; if not, there." Sing a song and notice the chorus is a loop. Ask "what comes first?" at bedtime and in the kitchen, or while reading a story from the library together: "what happened first, and what made the next thing happen?" If you want a screen, Robot Steps lets a child line up step cards and watch the robot walk them out, Peabody Blocks starts with a single step, and the unplugged coding lessons need no screen at all.

Ages 8 to 11: build small things, then poke at them

This is the age for block coding. Peabody Blocks has eight short lessons, from one step to a small game, with no typing and nothing sent anywhere. After each build, ask the prediction question: "what will happen if we change this number?" Then run it. Ask "how could we break it?" and let them try. When they ask an AI for help, have them say the rules aloud first.

Ages 12 to 14: read real code, and write specifications

Now it is worth meeting a real language, mostly for reading. Have an AI write something small, then read it together and have your child explain each part in plain words. Ask them to write down, before asking, what the program must do and what it must never do. Then test it against that list. Debugging a program they did not write is the skill they will use most.

What Peabody has today, and what it doesn't

- Robot Steps: a tap-only game from one step to repeat and if cards. A wrong program never costs a life.
- Coding lessons: ten unplugged lessons with printables, for ages 5 to 14.
- Peabody Blocks: eight coding lessons with blocks a child taps, from one step to a game. Open to anyone, no sign-up, nothing leaves the device.
- Learning games: reading and math practice that adapts to the child. Not coding, but the same habits of noticing a pattern and trying again.
- For educators: where teachers can find what is free through the end of 2026.

We don't have a full coding course,. We would rather say so than pad the list.

Questions grown-ups ask

Should my child learn to code at all?

If they are curious, yes. The ideas are useful whether or not they ever become a programmer, because they are how you describe any process exactly. If they are not curious, play the unplugged games and wait.

Which programming language should they start with?

Start with none. Blocks first, then a language their teacher or a project calls for. The first language matters much less than having built the picture of a program.

Will AI replace programmers?

Nobody knows. What looks safe is that someone has to say what to build, check that it works and fix it when it doesn't. That person needs the ideas above.

How much screen time does this need?

Very little. The ideas can be learned away from a screen, and a few minutes a day with Blocks is plenty.

Is it too late to start at 12?

No. A twelve-year-old can learn the six ideas in a few weeks, and they bring the patience for reading real code.

Free at peabody.kids/blog/do-kids-still-need-to-learn-coding-when-ai-writes-code