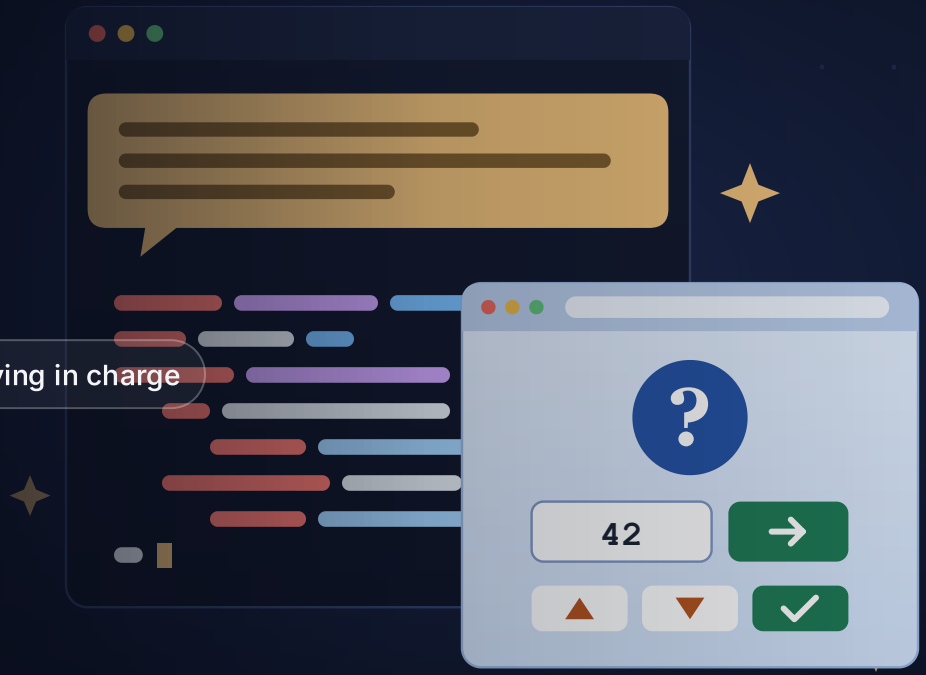
A PROGRAMMING LESSON

Vibe Coding

Programming by talking to an AI

You describe your game in plain English, and an AI writes the code. But who checks that it works, and that it does what you wanted?

1 The idea
2 OpenCode
3 Your first project
4 Staying in charge



A request to a coding agent, the code it writes, and the game it makes.

1 · THE IDEA

Programming by describing

In the JavaScript course, you wrote every line yourself. In **vibe coding**, you describe in plain English what you want, and an **artificial intelligence** (an AI) writes the code for you. You try it, then you ask for changes.

Coding by hand

- You write every instruction.
- You know exactly what your program does.
- It is slow for a big project, but you learn a lot.

Vibe coding

- You describe the result; the AI writes the instructions.
- You can try out lots of ideas quickly.
- The AI sometimes gets it wrong: testing and reading the code is your job.

WHERE THE WORD COMES FROM

In February 2025, the AI researcher Andrej Karpathy described “a new kind of coding I call ‘vibe coding’, where you fully give in to the vibes [...] and forget that the code even exists.” The Collins dictionary made it its word of the year for 2025.

For an afternoon game, you can go with the flow. For a program that matters, you don’t forget the code: you test it and you read it.

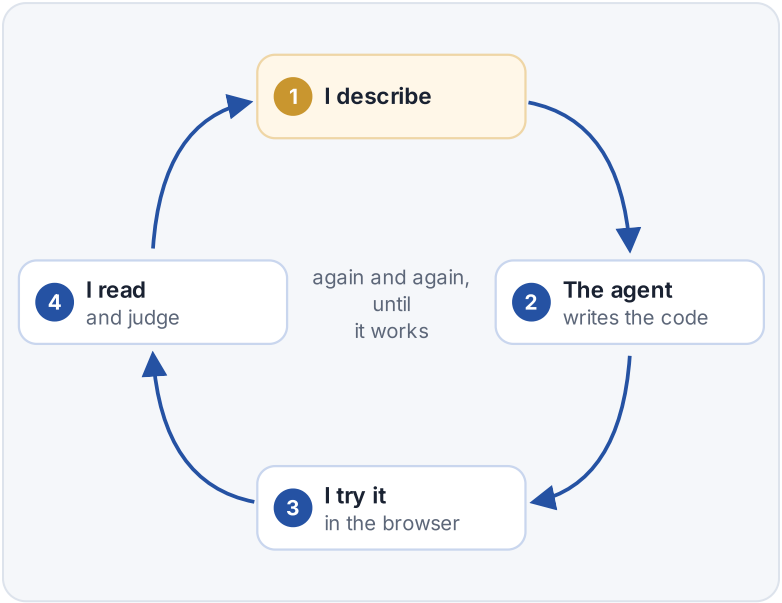
1 · THE IDEA

The vibe coding loop

The AI that codes is an **agent**: a **language model**, which reads your request and writes an answer, plus **tools** that let it act in your folder.

- The **model** learned from huge amounts of text and programs. It doesn't "understand" the way you do: it predicts, word after word, the most likely next part. It is often right, not always.
- The **tools** let it read your files, write new ones and run commands on your computer.
- **You** choose the goal, try the result and judge whether it is good.

> **YOU ARE THE PROJECT LEAD**
>
> The agent writes fast, but you are the one who knows what you want. One turn of the loop often takes less than a minute.



OpenCode, a coding agent

OpenCode is an **open source** coding agent: its code is public, and anyone can install it for free. It works in a terminal or as a desktop app, on macOS, Windows and Linux.

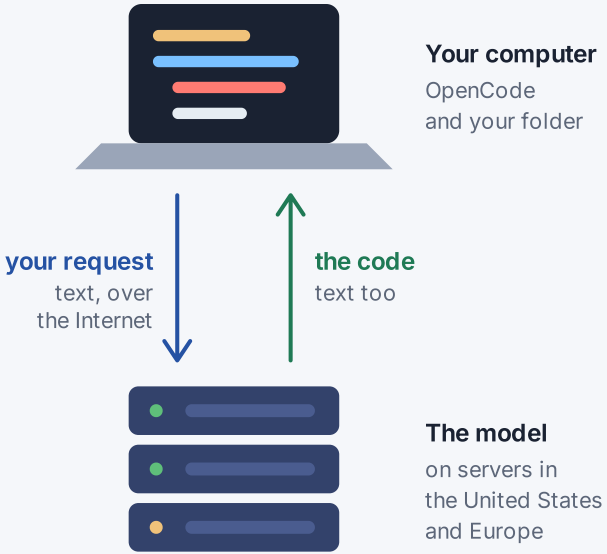
- OpenCode is not an AI itself: it puts the **model** of your choice to work, and gives it its tools.
- Some models must be paid for. Others are **free for a limited time**, in OpenCode Zen, OpenCode's model service. In November 2026, for example, **DeepSeek V4.1 Flash Free**, a free version of a model by the Chinese company DeepSeek.
- Free models have the word **Free** in their name.

Mode	What the agent does
Build	Writes and changes files, and runs commands. It is the mode OpenCode starts in.
Plan	Explores your folder and proposes a plan, without changing the project's files.

Switch from one to the other with `Shift`+`Tab`.

Free, on any computer?

The model doesn’t run on your computer, but on its provider’s servers. Your computer only sends and receives text: it doesn’t need to be powerful.



Computer	macOS, Windows or Linux. An old laptop will do, with no graphics card.
Internet	Needed. And the free model only works inside OpenCode.
Price	$0 “for a limited time”, with no end date announced. The previous offer, DeepSeek V4 Flash Free, ended in August 2026.
Limits	Not published. If the model stops answering or disappears from the list, choose another one marked Free.
Your data	For this model, OpenCode says nothing is kept or used to train the model. Other free models, like Big Pickle, may keep what you write.

From the OpenCode Zen documentation, in November 2026. Free models come and go often.

Installing and starting, with an adult

OpenCode's terms of use require you to be old enough to sign a contract, or to have a parent's permission: the adult installs OpenCode and makes the decisions.

1 **Install.** The desktop app is at opencode.ai/download. In a macOS or Linux terminal, one command is enough: `curl -fsSL https://opencode.ai/v2/install | bash`

2 **Open an empty folder.** Make a folder just for your project, such as `guess-the-number`. In the terminal: `cd guess-the-number`, then `opencode`. In the app, open that folder.

3 **Choose the model.** Type `/models` (or `Ctrl`+`X`, then `M`) and pick DeepSeek V4.1 Flash Free, or another model marked Free.

`Enter`	sends your request
`Shift`+`Tab`	switches from Build to Plan, and back
`/undo` · `Ctrl`+`X`, `U`	undoes your last request and the work that followed it
`/redo` · `Ctrl`+`X`, `R`	brings back what was undone
`/new` · `Ctrl`+`X`, `N`	starts a new conversation
`Ctrl`+`P`	shows every command

The commands of OpenCode 2, in November 2026. They sometimes change: `Ctrl`+`P` shows the ones that exist.

A good request

The first project: a game where you guess a number between 1 and 100. The agent can't read your mind: the clearer your request, the closer the result to what you want.

1. **The goal:** what the program does.
2. **The rules:** what happens when you play, and when you win.
3. **The look:** the language, the colours, the size of the text.
4. **The limits:** a single file, nothing to download, simple code with comments.

> **A PLAN FIRST**
>
> Switch to Plan (`Shift` + `Tab`) and send your request: the agent proposes a plan without writing anything. Read it, fix what is wrong, then switch back to Build and type "Go ahead".

YOUR REQUEST

Make a "Guess the number" game in a single file, `index.html`, in HTML, CSS and JavaScript, without downloading anything from the Internet.

The computer picks a random whole number between 1 and 100. The player types a number in a box and presses "Try". The game answers "Too low!", "Too high!" or "Well done!", and counts the tries. A "Play again" button starts a new game.

The text is in English, in big letters, on a light blue background.

I am learning JavaScript: write simple code, with comments in English.

Try it, then ask for a change

When the agent has finished, open `index.html` in your browser (double-click the file) and play. Then tell the agent what you saw, like a tester.

× Too vague

- “It doesn’t work.”
- “Make it better.”
- “Add levels, sounds, a high-score table and animations.”

✓ Precise

- “When I type 80, the game says ‘Too low!’, but the secret number was 35.”
- “Make the Try button bigger and put it to the right of the box.”
- “Add only a counter of tries at the top of the page.”

ONE THING AT A TIME

If a change breaks the game, you know which one. And if you don’t like it, `/undo` undoes your last request and the work that followed it.

A BACKUP

When your game works, make a copy of the folder (`guess-the-number-v1`). If the agent breaks everything later, you start again from the copy.

Reading the AI's code

Thanks to the course, you can read the code the agent writes: variables (lesson 2), conditions (lesson 3), a function (lesson 5). Here is a piece of the game… with a mistake in it.

```
// The secret number, a whole number from 1 to 100
const secret = Math.floor(Math.random() * 100) + 1;
let tries = 0;

function check(guess) {
  tries = tries + 1;
  if (guess < secret) {
    return "Too high!";
  }
  if (guess > secret) {
    return "Too low!";
  }
  return "Well done! Found in " + tries + " tries.";
}
```

1 **The secret is 35 and the player types 20. What does the game answer? Is it right?**

"Too high!", which is wrong: 20 is lower than 35. The two messages are swapped: when `guess < secret`, the game must answer "Too low!".

2 **The player tries 50, then 25, then 35, and wins. What is `tries` at the end?**

3: each call to `check` adds 1. The game prints "Well done! Found in 3 tries."

3 **`Math.random()` is not in the course. What do you do?**

Ask the agent: "Explain the line that picks the secret number, as if to a beginner." `Math.random()` gives a random number from 0 to 1 (never 1 itself); times 100, it goes from 0 to 99.99…; `Math.floor` drops what is after the point (0 to 99); plus 1: from 1 to 100.

The AI makes mistakes too

A language model always writes confidently, even when it is wrong. Four traps to know:

“It’s fixed!”

The agent says it has solved the problem… without having tried it.

→ **Always try it yourself.**

It makes things up

It can use a function that doesn’t exist, or talk about a file it has never read.

→ **If an error appears, paste the message to the agent.**

It breaks what worked

While changing one thing, it changes another one too.

→ **After each change, play the whole game again.**

It does too much

A hundred lines where ten would do, or things you never asked for.

→ **Ask for “simpler”, or “only this”.**

WHAT YOU KNOW MATTERS

The more you know about programming, the better you vibe code: you spot the mistakes, you describe what you want more clearly, and you can tell when the AI is right.

Vibe coding safely

The agent acts on your computer, and what you type travels over the Internet. A few rules keep you safe.

✓ Good ideas

- Work in an empty folder, just for the project.
- Start in Plan mode, which doesn't change the project's files.
- Have an adult nearby when the agent wants to run a command or install something.
- Ask the agent what a command does before letting it run.
- Make a copy of the folder when things work.

× Bad ideas

- Typing your name, your address, your school or your phone number, or giving photos of yourself.
- Giving a password, or a key you got from a website.
- Starting OpenCode in a folder with important files: your photos, your parents' documents.
- Agreeing to install a program you don't know.

IT DOESN'T ASK FIRST

Out of the box, in Build mode, OpenCode creates, changes and deletes files and runs commands without asking, even outside your folder. An adult can set it to ask for permission first: see the Permissions page of its documentation.

REVIEW

Test yourself

1 **In a sentence, what is vibe coding?**

Describing in plain language what you want, letting an AI write the code, trying it, and asking for changes, again and again.

2 **Will an old laptop without a graphics card do for DeepSeek V4.1 Flash Free?**

Yes, if it runs macOS, Windows or Linux and has Internet: the model runs on servers, not on your computer.

3 **Your free model has disappeared from the list. What do you do?**

Pick another model marked Free with `/models`: free models are only free for a limited time.

4 **Plan or Build: which one changes your files?**

Build. Plan explores and proposes, without changing the project's files.

5 **The agent says "It's fixed!". What do you do?**

You try it yourself, and play the whole game again: maybe something else broke.

6 **You want a high-score table. Do you give the AI your real name and your school for it?**

No: you choose a nickname. You never give personal information.

YOUR TURN

Project ideas

Start small, get a first version working, then add one thing at a time. For each project, the same method: a plan first, a clear request, then the loop.

The times-table quiz

Ten random multiplications and a score at the end. Then: a stopwatch.

Euclid, animated

Two numbers, and each step of the algorithm shown one after the other. Compare the AI's code with yours from lesson 7.

The bouncing ball

A ball that bounces off the edges of a drawing (a canvas). Then: a paddle to hit it back.

The memory game

Sixteen cards face down, turned over two at a time to find the pairs. Then: count the moves.

The dice lab

Roll two dice 1,000 times and draw how often each total comes up. Which total wins?

Your project page

A page about your favourite animal, with a three-question quiz at the end. No photo of yourself!

A CHALLENGE

Write "Guess the number" again by hand, with what you learned in the course, then compare your code with the AI's. Which one is shorter? Which one is clearer?

TO SUM UP

The big ideas

1. In **vibe coding**, you describe what you want, an **agent** writes the code, you try it and you ask for changes: that is the loop.

2. **OpenCode** puts to work a model that runs on servers: any computer with macOS, Windows or Linux will do, with Internet. Models marked **Free**, like DeepSeek V4.1 Flash Free, are free for a limited time.

3. You **stay in charge**: a clear request, one thing at a time, you test and you read the code, in a folder of its own, and you never give personal information.

Go further

▶ The JavaScript course | Lesson 7: Euclid's algorithm | Read code in the Practice Room | The OpenCode documentation

Illustrations drawn for this lesson. The information on OpenCode and its free models dates from November 2026 (the OpenCode and OpenCode Zen documentation, and OpenCode's terms of use): it changes often.