VIBE CODING · START HERE · 10 MIN READ

What is vibe coding?

Building software by describing it to AI: where the term came from, the build loop, and when to slow down.

Pass the check (4 of 5) online to earn the “Vibe Coder” badge:
https://digitalbuilders757.com/university/study/what-is-vibe-coding

Where the term came from

In February 2025, AI researcher Andrej Karpathy (a co-founder of OpenAI) described a new way he was building small projects: talking to an AI coding tool, accepting what it wrote, pasting error messages back in, and barely looking at the code. He called it vibe coding, and the name stuck.

Today people use the term more loosely: building software mainly by describing what you want in plain language and letting AI write most of the code. You steer by intent and by testing results instead of typing every line.

The vibe coding loop

Every build, big or small, runs the same loop:

- Describe: say what the user should see and be able to do.
- Generate: the AI writes or changes code.
- Run: open the preview and use it like a real user.
- Check: does it do what you asked? Anything broken, ugly or unsafe?
- Refine: describe the next change or paste the exact error. Repeat.
- Save: commit working versions so you can always go back (see the GitHub guide).

Two kinds of tools

Prompt-to-app builders (Lovable, Bolt, v0, Replit and others) run in your browser and are made for people who don't write code. You describe an app; they build it, show a live preview and can host it.

Coding agents (Cursor, Claude Code, GitHub Copilot, OpenAI Codex and others) work inside a real codebase. They are made for builders who want an AI that reads, edits and runs code across a whole project. Many people start with a builder and move to an agent as their project grows.

Where vibe coding shines, and where to slow down

It shines for prototypes, personal tools, internal dashboards, landing pages, class projects and testing a business idea quickly.

Slow down when real people's data, money or safety is involved. Security researchers have repeatedly found vibe-coded apps with secret keys in the browser code and databases left open to anyone. Those mistakes are easy to prevent once you know to look (see Build safely). Pure 'accept everything, never look' vibe coding is for throwaway projects; anything you ship to others needs review.

The mindset

You are the product owner and the tester. The AI is a fast builder that needs clear direction. You don't need to memorise syntax, but you do need to understand the parts of an app well enough to describe them, test them and notice when something is off. That is exactly what this library teaches.

Key terms

Vibe coding
Building software mainly by describing what you want to an AI and testing the results. Coined by Andrej Karpathy in 2025.

Prompt
What you give the AI: instructions, a question and any material to work from.

Preview
A live, working view of your app while you build it.

Prompt-to-app builder
A browser tool that builds, previews and often hosts an app from your chat description.

Coding agent
An AI that reads, edits and runs code across a whole project.

Bug
A mistake that makes software behave wrongly.

Debugging
Finding and fixing bugs, often by reading the exact error message.

Prototype
A rough early version made to test whether an idea works.

Try it: Your first 15-minute build

1. Open a prompt-to-app builder such as Lovable, Bolt, v0 or Replit (free tiers are fine).
2. Prompt: 'Build a one-page tip calculator. Inputs: bill amount and tip percent (10, 15, 20). Show tip and total. Big friendly buttons, works on phones.'
3. Use the preview like a customer. Try a weird input like a negative number.
4. Ask for one improvement based on what you found. You just ran the loop.

KNOWLEDGE CHECK

What is vibe coding?

5 questions. Circle one answer for each; 4 or more correct passes. Answers are at the back.

1. Who popularised the term 'vibe coding' in early 2025?

A. Bill Gates
B. Andrej Karpathy
C. Mark Zuckerberg
D. Ada Lovelace

2. What is the right order of the vibe coding loop?

A. Publish, describe, hope
B. Generate, publish, then describe what you wanted
C. Describe, generate, run, check, refine
D. Run, run, run

3. Which tool is a prompt-to-app builder made for non-coders?

A. A terminal
B. Git
C. DNS
D. Lovable

4. When should you slow down and review AI-written code carefully?

A. When the app handles real people's data, money or safety
B. Never, because AI-written code is tested automatically
C. Only when the app first fails to load
D. Only when the colours look wrong

5. In vibe coding, what is your main job?

A. Typing every line of code yourself, so you understand all of it first
B. Giving clear direction and testing that the result really works
C. Letting the AI decide what to build, then publishing whatever it makes
D. Choosing fonts

Official sites

Lovable: https://lovable.dev
Bolt: https://bolt.new
v0 by Vercel: https://v0.app
Replit: https://replit.com

Last reviewed 2026-11-25. AI tools change fast; check each tool's own site for current features, plans and prices.

ANSWER KEY

What is vibe coding?

1. B: Andrej Karpathy
Andrej Karpathy coined it in February 2025 to describe building by talking to AI and accepting its code.

2. C: Describe, generate, run, check, refine
Describe what you want, let AI generate, run it, check it like a user, then refine.

3. D: Lovable
Lovable (like Bolt, v0 and Replit) builds and previews an app from your description in the browser.

4. A: When the app handles real people's data, money or safety
Real data, payments and safety need review. Common vibe-coding mistakes include exposed keys and open databases.

5. B: Giving clear direction and testing that the result really works
You're the product owner and tester: describe clearly, test like a user, and refine.