Vite Frontend build tool

START HERE

`npm create vite@latest`
Scaffold a project

`npx vite --host`
Expose on the network

`npm install`
Install dependencies

`npx vite --force`
Re-bundle dependencies

`npx vite`
Start dev server

`npx vite --open`
Dev server, open browser

`npx vite build`
Production build to dist/

`npx vite preview`
Serve the built output

index.html entry point → `vite` **Dev server** localhost:5173, HMR → `vite build` **dist/** bundled assets → `vite preview` **Local check** not for production

CONCEPTS

Dev serves your source as native ES modules with no bundling; build bundles for production. Use preview only to check the build locally, not to host it.

Only variables prefixed VITE_ reach client code, and their values are baked into the bundle. Never put secrets in them.

DEV SERVER

`vite`
Start the dev server (aliases: dev, serve)

`vite <root>`
Serve a different project root
The config file must live in that root.

`vite --port <port>`
Set the port (default 5173)

`vite --strictPort`
Fail instead of picking another port

`vite --host [host]`
Listen on all addresses or a given host
On WSL2 --host alone may not be enough for LAN access.

`vite --open [path]`
Open the browser on start

`vite --force`
Ignore the optimizer cache

DEV SERVER (cont.)

`vite --mode <mode>`
Set the env mode (default development)

`vite --base <path>`
Set the public base path

`vite --config <file>`
Use a specific config file

BUILD

`vite build`
Production build into dist/

`vite build --outDir <dir>`
Change the output directory

`vite build --sourcemap [output]`
Emit source maps (true, inline, hidden)

`vite build --mode <mode>`
Build with another env mode
Mode and NODE_ENV are separate: build keeps NODE_ENV=production.

`vite build --watch`
Rebuild when files change

`vite build --minify <minifier>`
Pick minifier: oxc, terser, esbuild, false

`vite build --target <target>`
Set the transpile target

`vite build --base <path>`
Build for a sub-path deployment

BUILD (cont.)

`vite build --emptyOutDir`
Empty outDir even outside the root

`vite build --manifest [name]`
Emit a manifest of hashed assets

`vite build --ssr [entry]`
Build an entry for server rendering

PREVIEW

`vite preview`
Serve the built dist/ locally

`vite preview --port <port>`
Set the preview port

`vite preview --host [host]`
Expose the preview on the network

`vite preview --open [path]`
Open the browser on start

`vite preview --outDir <dir>`
Serve a non-default build folder

`vite optimize`
Pre-bundle dependencies (deprecated)
Deprecated: this already runs automatically.

DEV SERVER KEYS

Type the key in the terminal running the dev server, then press Enter.

`h`
Show the shortcut list

`r`
Restart the server

`u`
Print the server URL

`o`
Open in the browser

`c`
Clear the console

`q`
Quit the server

SCAFFOLDING

`npm create vite@latest <name>`
Create a project interactively

`npm create vite@latest <name> -- --template <tpl>`
Scaffold a template (npm 7+ needs --)

`pnpm create vite <name> --template <tpl>`
Same with pnpm (no extra --)

`npm create vite@latest . -- --template <tpl>`
Scaffold in the current directory

SCAFFOLDING (cont.)

`npm create vite@latest <name> -- --no-interactive`
Scaffold without prompts

Template names

Framework	Plain JS	TypeScript
`vanilla`	vanilla	vanilla-ts
`vue`	vue	vue-ts
`react`	react	react-ts
`react (compiler)`	react-compiler	react-compiler-ts
`preact`	preact	preact-ts
`svelte`	svelte	svelte-ts
`solid`	solid	solid-ts
`lit`	lit	lit-ts
`qwik`	qwik	qwik-ts

ENV AND DEBUGGING

Env files (later rows win)

File	Loaded	Git
`.env`	always	commit
`.env.local`	always	ignored
`.env.<mode>`	that mode only	commit
`.env.<mode>.local`	that mode only	ignored

Read values with import.meta.env.VITE_NAME. Built-ins: MODE, BASE_URL, PROD, DEV, SSR. Shell variables beat env files; restart after editing them.

ENV AND DEBUGGING (cont.)

`vite --debug [feat]`
Show debug logs (e.g. vite:*)

`vite --version`
Print the Vite version

`vite --help`
Show CLI help

`npx vite@<sha>`
Try an unreleased build via pkg.pr.new

RECIPES

New React + TypeScript app

```
npm create vite@latest my-app
-- --template react-ts
cd my-app
npm install
npm run dev
```

Staging build with its own env

```
# loads .env and .env.staging
npx vite build --mode staging
npx vite preview --outDir dist
```

Proxy API calls in vite.config.js

```
export default defineConfig({
  server: { proxy: {
    '/api': { target: 'http://
localhost:4567', changeOrigin:
true }
  } }
})
```

Test the build on a phone

```
npx vite build
npx vite preview --host --port
4173
```