Astro Web framework CLI

START HERE

`npm create astro@latest`
New project

`astro dev`
Dev server

`astro build`
Build to dist/

`astro preview`
Serve the build

`astro add <name>`
Add integration

`astro check`
Type-check

`astro sync`
Generate types

`astro info`
Bug-report info

src/pages .astro, .md routes → `astro dev` → **Dev server** HMR, port 4321 → `astro build` → **dist/** static output → `astro preview` → **Local preview** check before deploy

CONCEPTS

Install Astro per project and run it with npx astro <cmd> or npm scripts; it is not meant to be installed globally.

Pages are static HTML by default; add an adapter and output or per-page prerender settings for on-demand rendering.

Config lives in astro.config.mjs; CLI flags like --site, --base, --outDir override it for one run.

PROJECT

`npm create astro@latest`
Scaffold a project with the wizard

`npm create astro@latest -- --template <name>`
Start from an official template
Also accepts <user>/<repo>, optionally with #<branch>.

`npm create astro@latest -- --add <integration>`
Scaffold and add an integration

`npm run dev`
Run the dev script from package.json

`npx astro <command>`
Run the local CLI directly

`astro docs`
Open the docs in a browser

PROJECT (cont.)

`astro info`
Print environment details

`astro info --copy`
Copy details to clipboard, no prompt

`astro --version`
Print the Astro version

`astro --help`
List commands and flags

DEVELOP

`astro dev`
Start the dev server (HMR)

`astro dev --port <n>`
Use another port (default 4321)

`astro dev --host`
Listen on all network addresses

`astro dev --host <address>`
Listen on one address

`astro dev --open`
Open the browser on start

`astro dev --open <pathname>`
Open a specific page

`astro dev --force`
Clear content layer cache, full rebuild

`astro dev --allowed-hosts <hosts>`
Allow extra hostnames, or true for any

DEVELOP (cont.)

`astro dev --background`
Run as a detached process

`astro dev status`
Show the background server state

`astro dev logs --follow`
Tail background server logs

`astro dev stop`
Stop the background server

`s / o / q`
Dev hotkeys: restart / open / quit

BUILD AND PREVIEW

`astro build`
Build the site into dist/

`astro build --timings`
Show build time per phase

`astro build --outDir <path>`
Write output elsewhere

`astro build --site <url>`
Override the site URL

`astro build --base <path>`
Override the base path

`astro build --mode <name>`
Set the project mode

`astro preview`
Serve dist/ locally

`astro preview --port <n>`
Preview on another port

`astro preview --host`
Preview on the network

CHECK AND SYNC

`astro check`
Type-check .astro and TS files
Exits with code 1 on errors, so it works as a CI step.

`astro check --watch`
Re-check on every change

`astro check --tsconfig <path>`
Use a specific tsconfig

`astro check --minimumFailingSeverity <level>`
Fail on error, warning or hint

`astro check --minimumSeverity <level>`
Lowest severity to report

`astro sync`
Generate .astro/ types and content types
dev, build and check run this already; run it by hand after editing the config in an editor.

INTEGRATIONS

`astro add <name>`
Install and wire up an integration
Edits astro.config.mjs and package.json; review the diff.

`astro add react tailwind`
Add several integrations at once

`astro add node`
Add an SSR adapter (also vercel, netlify, cloudflare)

INTEGRATIONS (cont.)

`astro add mdx`
Add MDX page and content support

`astro add sitemap`
Generate sitemap files on build

`integrations: [react()]`
Config key listing active integrations

`adapter: node({ mode: 'standalone' })`
Config key for the hosting adapter

CONFIGURATION

`--config <path>`
Use another config file

`--root <path>`
Set the project root

`--verbose`
Verbose logging

`site: 'https://example.com'`
Final URL for sitemap and canonical links

`base: '/docs'`
Deploy under a subpath

`output: 'static' / 'server'`
Prerender by default or render on demand

`server: { port: 3000, host: true }`
Dev and preview port and host

CONFIGURATION (cont.)

`outDir: './dist'`
Build output directory

`trailingSlash: 'always' / 'never'`
URL trailing-slash rule

`redirects: { '/old': '/new' }`
Static redirects

`vite: { }`
Pass options through to Vite

RECIPES

Config with integration and site

```
// astro.config.mjs
import { defineConfig } from 'astro/config';
import react from '@astrojs/react';

export default defineConfig({
  site: 'https://example.com',
  integrations: [react()],
});
```

Test on a phone in the LAN

```
astro dev --host --port 3000
# open http://<your-ip>:3000 on the phone
```

CI pipeline

```
npm ci
astro check
astro build
```

Preview a build under a subpath

```
astro build --base /docs --site https://example.com
astro preview
```