Astro Web framework CLI

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

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

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

`astro info`
Print environment details

PROJECT (cont.)

`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 --open`
Open the browser on start

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

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

BUILD AND PREVIEW

`astro build`
Build the site into dist/

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

BUILD AND PREVIEW (cont.)

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

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

`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

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

`astro sync`
Generate .astro/ types and content types

INTEGRATIONS

`astro add <name>`
Install and wire up an integration

`astro add react tailwind`
Add several integrations at once

INTEGRATIONS (cont.)

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

`astro add mdx`
Add MDX page and content support

`astro add sitemap`
Generate sitemap files on build

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

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

CONFIGURATION (cont.)

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

`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
```