Next.js React web framework

START HERE

`npx create-next-app@latest` *`<name>`*
Scaffold an app

`npx next dev -p` *`<port>`*
Use another port

`npm run dev`
Dev server on :3000

`npx next upgrade`
Move to latest version

`npm run build`
Production build

`npm run start`
Serve the build

`npx next typegen`
Generate route types

`npx next info`
Env details for bug reports

CONCEPTS

Folders under app/ are URL segments; a route goes public only once it has a page or route file.

layout shared shell
template re-mounts on nav
error error boundary
loading suspense fallback
not-found 404 boundary
page the route UI

Nesting order of special files in a segment

npm run forwards flags only after --, e.g. npm run dev -- -p 4000. pnpm, yarn and bun do not need it.

next with no command is the same as next dev.

CREATE AN APP

`npx create-next-app@latest` *`<name>`*
Create a new project

`create-next-app --yes`
Reuse saved prefs or defaults, no prompts

`create-next-app --js`
JavaScript instead of TypeScript

`create-next-app --src-dir`
Put code in src/

`create-next-app --biome`
Biome as linter

CREATE AN APP (cont.)

`create-next-app --eslint`
ESLint as linter

`create-next-app --no-tailwind`
Skip Tailwind CSS

`create-next-app --react-compiler`
Enable React Compiler

`create-next-app --use-pnpm`
Pick the package manager (npm, yarn, bun too)

`create-next-app --example` *`<name>`*
Start from an official example

DEV, BUILD, START

`next dev`
Dev server with hot reload

`next dev -p` *`<port>`*
Set the port (or PORT env var)
PORT cannot be set in .env: the server starts before env files load.

`next dev -H` *`<host>`*
Bind a hostname (default 0.0.0.0)

`next dev --webpack`
Use Webpack instead of Turbopack

`next dev --experimental-https`
HTTPS with a self-signed cert

`next build`
Optimized production build

`next build --debug`
Verbose output: rewrites, redirects, headers

DEV, BUILD, START (cont.)

`next build --debug-prerender`
Readable prerender errors, keeps going
Debugging only: do not deploy a build made with this flag.

`next build --debug-build-paths="`*`<glob>`***`"`**
Build only matching routes

`next build --webpack`
Build with Webpack

`next start`
Serve the production build

`next start -p` *`<port>`*
Port for the production server

`next start --keepAliveTimeout` *`<ms>`*
Keep-alive longer than your proxy's

TOOLS

`next --help`
List commands and options

`next --version`
Print the Next.js version

`next info`
Print system and package versions

`next typegen`
Generate route types without a build

`next typegen && tsc --noEmit`
Type-check routes in CI

TOOLS (cont.)

`next upgrade`
Upgrade Next.js and its codemods

`next upgrade --revision` *`<tag>`*
Target a version, latest or canary

`next analyze`
Explore bundle size in the browser

`next telemetry --disable`
Opt out of telemetry

There is no next lint here: run your linter (ESLint or Biome) via its own script.

APP ROUTER FILES

`app/layout.tsx`
Shared shell; root one needs <html> and <body>

`app/page.tsx`
UI that makes a route public

`app/loading.tsx`
Fallback shown while a segment streams

`app/error.tsx`
Error boundary; must be a client component

`app/not-found.tsx`
UI for notFound() and unmatched URLs

`app/global-error.tsx`
Catches errors in the root layout

APP ROUTER FILES (cont.)

`app/template.tsx`
Like layout but remounts on navigation

`app/api/`*`<name>`***`/route.ts`**
Route handler (GET, POST, ...)

`proxy.ts`
Runs before requests; successor of middleware
Renamed from middleware.ts in Next.js 16.

`next.config.ts`
Framework config (.js and .mjs also work)

`public/`
Static files served from /

ROUTE FOLDERS

`app/blog/page.tsx`
Serves /blog

`app/blog/[slug]/page.tsx`
Dynamic segment

`app/shop/[...slug]/page.tsx`
Catch-all segment

`app/docs/[[...slug]]/page.tsx`
Optional catch-all, also matches /docs

`app/(marketing)/page.tsx`
Route group: name not in the URL

`app/blog/_lib/`
Private folder, never routed

`app/@sidebar/page.tsx`
Named slot (parallel route)

ENVIRONMENT

`.env`
Defaults for every environment

`.env.local`
Local secrets; keep out of git

`.env.development`
Loaded by next dev

`.env.production`
Loaded by next build and start

`NEXT_PUBLIC_`*`<NAME>`***`=value`**
Inlined into browser code at build time
Changing it later needs a rebuild.

`process.env.`*`<NAME>`*
Read a variable on the server

Variables without the NEXT_PUBLIC_ prefix stay server-side.

RECIPES

New app, ready to code

```
npx create-next-app@latest my-app --yes
cd my-app
npm run dev
# open http://localhost:3000
```

CI check without a full build

```
npm ci
npx next typegen && npx tsc --noEmit
npx next build
```

Debug a failing prerender

```
npx next build --debug-prerender
npx next build --debug-build-paths="app/blog/**/page.tsx"
```