webpack JS module bundler

START HERE

`npm i -D webpack webpack-cli`
Install locally

`npx webpack --analyze`
Inspect bundle size

`npx webpack`
Build with config

`npx webpack info`
Print system info

`npx webpack --mode production`
Minified build

`npx webpack watch`
Rebuild on change

`npx webpack serve --open`
Dev server

`npx webpack configtest`
Check the config

Entry ./src/index.js → **imports** → **Loaders** module.rules → **graph** → **Plugins** whole bundle → **emit** → **Output** dist/

CONCEPTS

Webpack starts at the entry, follows imports, runs loaders per file type and plugins over the result.

Install webpack-cli next to webpack; the webpack command comes from the CLI package.

COMMANDS

`npx webpack [entries...]`
Build (default command)

`npx webpack build`
Build, explicit form

`npx webpack watch`
Build and rebuild on changes

`npx webpack serve`
Run webpack-dev-server

`npx webpack configtest [path]`
Validate a config file
Defaults to ./webpack.config.js.

`npx webpack info`
Show system and package info

`npx webpack version`
Show webpack, CLI, dev-server versions

COMMANDS (cont.)

`npx webpack help [cmd]`
Help for a command or option

`npx create-webpack-app [dir]`
Scaffold a new project

`npx create-webpack-app plugin [dir]`
Scaffold a plugin

`npx create-webpack-app loader [dir]`
Scaffold a loader

BUILD OPTIONS

`--mode` *<mode>*
development, production or none

`-c, --config` *<file>*
Use this config file

`--entry` *<path>*
Add an entry point

`--entry-reset` *<path>*
Replace configured entries

`-o, --output-path` *<dir>*
Output directory

`-t, --target` *<target>*
Build target, e.g. web or node

`-d, --devtool` *<type>*
Source map style

BUILD OPTIONS (cont.)

`--no-devtool`
Disable source maps

`--name` *<name>*
Name this configuration

`--fail-on-warnings`
Exit non-zero on warnings

`-w, --watch`
Watch for changes

`--no-watch`
Do not watch

CONFIG AND ENV

`--env` *<key>*`[=`*<value>*`]`
Pass env to a function config
Repeatable; read as the first argument of a config function.

`--config-node-env` *<env>*
Set process.env.NODE_ENV

`--config-name` *<name>*
Run only the named config

`--merge`
Merge several --config files

`-e, --extends` *<file>*
Extend another config

CONFIG AND ENV (cont.)

`--disable-interpret`
Skip interpret when loading config

`WEBPACK_SERVE / WEBPACK_BUILD / WEBPACK_WATCH`
Built-in env flags

`NODE_OPTIONS=--max-old-space-size=4096 npx webpack`
Give the build more memory

OUTPUT AND ANALYSIS

`--stats` *<preset>*
Stats output detail

`--no-stats`
Silence stats output

`--json [file]`
Print or write stats as JSON

`--analyze`
Open webpack-bundle-analyzer

`--progress[=profile]`
Show progress, optionally profile

`--color / --no-color`
Toggle console colors

OUTPUT AND ANALYSIS (cont.)

`--help=verbose`
List every flag

DEV SERVER

`npx webpack serve --open`
Serve and open the browser

`npx webpack serve --static` *<dir>*
Serve static files from a dir

`npx webpack serve --no-hot`
Disable hot reloading

`npx webpack serve --mode development`
Serve in development mode

CONFIG FILES

Lookup order

Base name	Extensions
`webpack.config`	js mjs cjs ts cts mts
`.webpack/webpack.config`	same
`.webpack/webpackfile`	same

Exit codes: 0 success, 1 webpack errors, 2 bad options or internal error.

RECIPES

Minimal webpack.config.js

```
const path = require('path');
module.exports = {
  entry: './src/index.js',
  output: { path:
path.resolve(__dirname,
'dist'), filename:
'[name].js' },
  module: { rules: [{ test: /
\.css$/, use: ['style-loader',
'css-loader'] }] },
  plugins: [],
  devServer: { static: './
dist', hot: true },
};
```

Env-aware config

```
# webpack.config.js exports
(env, argv) ⇒ ({...})
npx webpack --env production --
mode production
npx webpack serve --env
port=3000
```

Size investigation

```
npx webpack --mode production
--json stats.json
# upload stats.json to an
analyzer, or:
npx webpack --mode production
--analyze
```

Verify before CI

```
npx webpack configtest
npx webpack --mode production
--fail-on-warnings
```