# Prettier Opinionated formatter

## CONCEPTS

Without --write or --check, Prettier prints the formatted code to stdout and touches nothing.

**Parse** source to AST
↓ discard layout
**Reprint** printWidth rules
↓ write / check
**Output** stdout, file or exit code

Prettier is opinionated: few options, and it reprints from the AST, so your original line breaks mostly do not survive.

## RUN

`prettier --write <path>`
Format files in place

`prettier --check <path>`
Report unformatted files, exit 1

`prettier --list-different <path>`
Print names of differing files

`prettier <file>`
Print formatted result to stdout

`prettier --ignore-unknown --write .`
Skip file types it cannot parse

## INPUT

`prettier --stdin-filepath <path>`
Pick parser and config from a name

`prettier --parser <name>`
Force a parser (babel, ts, json, md...)

## COMMON OPTIONS

`--print-width <n>`
Line length to wrap at (80)

`--tab-width <n>`
Spaces per indent level (2)

`--use-tabs`
Indent with tabs

## COMMON OPTIONS (cont.)

`--no-semi`
Omit semicolons

`--single-quote`
Single quotes in JS/TS

`--trailing-comma <all|es5|none>`
Trailing commas (default all)

`--prose-wrap <always|never|preserve>`
Wrap Markdown text

`--end-of-line <lf|crlf|cr|auto>`
Line ending style (lf)

## CONFIGURATION

`prettier --config <file> <path>`
Use a specific config file

`prettier --no-config <path>`
Ignore config files, use defaults

`prettier --find-config-path <file>`
Show which config applies

`package.json "prettier" key`
Config inside package.json

`.prettierrc / .prettierrc.json`
JSON or YAML options file

`"overrides": [{ "files", "options" }]`
Per-glob options

## IGNORING

`.prettierignore`
Gitignore syntax, project root

`// prettier-ignore`
Skip next JS/TS node

## CACHE AND CI

`prettier --cache --write .`
Only reformat changed files

Exit codes: 0 all formatted, 1 some files unformatted, 2 Prettier error. Use --check in CI.