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.

Config lookup goes up the tree from each file; there is no global config. CLI flags override the file by default.

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

`prettier --no-error-on-unmatched-pattern <glob>`
No error when nothing matches

`prettier --log-level silent <path>`
Quieter or louder output

`prettier --debug-check <file>`
Test that formatting is safe

INPUT

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

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

INPUT (cont.)

`prettier --range-start <n> --range-end <n>`
Format only a character range

`prettier --require-pragma <path>`
Only format files with the pragma

COMMON OPTIONS

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

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

`--use-tabs`
Indent with tabs

`--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)

`--arrow-parens <always|avoid>`
Parens around lone arrow param

`--quote-props <as-needed|consistent|preserve>`
Quote object keys

`--no-bracket-spacing`
No spaces in { a }

`--bracket-same-line`
Put > of multiline tag on last line

`--jsx-single-quote`
Single quotes in JSX

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

CONFIGURATION (cont.)

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

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

`prettier.config.js / .mjs / .ts`
Config as code, export default

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

IGNORING

`.prettierignore`
Gitignore syntax, project root

`prettier --ignore-path <file>`
Use another ignore file

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

`<!-- prettier-ignore -->`
Skip next HTML or Markdown block

node_modules and VCS folders are skipped by default, and .gitignore is honoured.

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.

RECIPES

CI gate

```
# fails the job on any unformatted file
npx prettier --check .
```

Format only staged files

```
git diff --name-only --cached --diff-filter=d -z \
  | xargs -0 prettier --write --ignore-unknown
git add -u
```

Minimal .prettierrc

```
{
  "singleQuote": true,
  "printWidth": 100,
  "overrides": [{ "files": "*.md", "options": { "proseWrap": "always" } }]
}
```