Biome Lint and format JS/TS

CONCEPTS

biome check is the one command to remember: it runs the formatter, the linter and import sorting in one pass.

biome check one pass, all three
- **Formatter** layout
- **Linter** rules, safe and unsafe fixes
- **Assist** organise imports

Without --write nothing on disk changes; biome only reports. Fixes marked unsafe also need --unsafe.

Biome supports JS, TS, JSX, JSON, CSS and more, but is not a drop-in for every ESLint plugin or Prettier option.

SETUP

npm i -D -E @biomejs/biome
Install pinned in a project

biome init
Create biome.json

biome version
Print the CLI version

biome migrate --write
Update config after an upgrade

biome migrate prettier --write
Import Prettier settings

biome migrate eslint --write
Import ESLint settings

CHECK

biome check *<path>*
Format, lint and assist, report only

biome check --write *<path>*
Apply formatting and safe fixes

biome check --write --unsafe *<path>*
Also apply unsafe fixes

biome check --staged
Only staged files

biome check --changed
Files changed versus the default branch

CHECK (cont.)

biome check --changed --since=*<ref>*
Changed since a given ref

biome check --linter-enabled=false
Skip the linter for this run

biome check --formatter-enabled=false
Skip the formatter for this run

biome check --stdin-file-path=*<file>*
Process stdin to stdout

FORMAT

biome format *<path>*
Show formatting differences

biome format --write *<path>*
Format files in place

biome format --staged --write
Format staged files

biome format --indent-style=space
Use spaces instead of tabs

biome format --line-width=*<n>*
Set the line width

LINT

biome lint *<path>*
Lint only

biome lint --write *<path>*
Apply safe lint fixes

biome lint --only=*<rule>*
Run a single rule or group

biome explain *<rule>*
Show docs for a rule

CI & OUTPUT

biome ci .
Check without modifying files

biome ci --changed
Only files changed on this branch

biome ci --reporter=github
Annotate a GitHub run

--reporter=*<name>*
Pick the output format

--error-on-warnings
Fail the run on warnings

CONFIGURATION

biome.json basics

```
{
  "vcs": { "enabled": true,
"clientKind": "git",
          "useIgnoreFile": true,
"defaultBranch": "main" },
  "files": { "includes": ["src/
**"] },
  "formatter": { "indentStyle":
"space", "lineWidth": 100 },
  "linter": { "rules": { "preset":
"recommended" } }
}
```

Config sections

Key	Controls
`formatter`	indent, width, line ending
`linter`	rules and their severity
`javascript.formatter`	quotes, semicolons, commas
`assist`	import organising and other actions
`overrides`	per-glob settings

Use files.includes with ! patterns to exclude paths; there is no separate ignore file.

A rule can be silenced inline with // biome-ignore lint/<group>/<rule>: <reason>.

RECIPES

Adopt in an existing project

```
npm i -D -E @biomejs/biome
npx biome init
npx biome migrate prettier --write
npx biome migrate eslint --write
npx biome check --write .
```

Pre-commit on staged files

```
# .git/hooks/pre-commit
npx biome check --staged --write
--no-errors-on-unmatched
git update-index --again
```

CI job

```
npx biome ci --reporter=github
# or only this branch
npx biome ci --changed --
since=origin/main
```

package.json scripts

```
"lint": "biome lint .",
"format": "biome format --
write .",
"check": "biome check --write ."
```