clang-format C/C++ code formatter

CONCEPTS

Put a .clang-format file in the repo root and use --style=file: every tool and editor then agrees on one style.

Without a .clang-format file, --style=file falls back to LLVM style; change that with --fallback-style.

RUN

clang-format *<file>*
Print formatted code to stdout

clang-format -i *<file>*...
Rewrite files in place

clang-format --style=*<style>* *<file>*
Pick a preset or file

clang-format --version
Show the version

clang-format -i src/*.cpp include/*.h
Format many files

clang-format --lines=*<start>*:*<end>* **-i** *<file>*
Format only a line range

<cmd> **| clang-format --assume-filename=***<name>*
Format stdin as that file type

CHECK (CI)

clang-format --dry-run *<file>*
Report violations, change nothing

clang-format -n -Werror *<file>*
Exit non-zero if unformatted

git ls-files '*.c' '*.h' | xargs clang-format -n -Werror
Check all tracked sources

STYLES & CONFIG

--style=LLVM
Preset: LLVM (default)

STYLES & CONFIG (cont.)

--style=Google
Preset: Google

--style=file
Use nearest .clang-format

--style="{BasedOnStyle: Google, IndentWidth: 4}"
Inline settings

clang-format -style=*<style>* **--dump-config > .clang-format**
Write a full config file

.CLANG-FORMAT KEYS

BasedOnStyle: Google
Start from a preset

IndentWidth: 4
Spaces per indent level

ColumnLimit: 100
Maximum line width (0 = none)

BreakBeforeBraces: Allman
Brace style: Attach, Linux, Allman

PointerAlignment: Left
Align * and & left, right or middle

SortIncludes: CaseSensitive
Sort #include lines (or Never)

GIT INTEGRATION

git clang-format
Format staged changes

git clang-format *<commit>*
Format lines changed since a commit

git clang-format main
Format all changes on a branch

git clang-format --diff
Show the diff, apply nothing

IN CODE & EDITORS

// clang-format off
Stop formatting from here

// clang-format on
Resume formatting