shfmt Shell script formatter

CONCEPTS

Default output uses tabs. Pick your style once with flags or an .editorconfig, then let CI enforce it with -d.

Script *.sh or shebang
↓ parse
Syntax tree dialect rules
↓ print
Formatted code flags / .editorconfig
↓ -w / -d / -l
Result file, diff or list

RUN

`shfmt <file>`
Print the formatted script

`shfmt -w <file>`
Write the result back to the file

`shfmt -d <path>`
Show a diff, exit non-zero if different

`shfmt -l <path>`
List files whose formatting differs

`shfmt -l -w .`
Format everything under here

`shfmt -`
Read from stdin

`shfmt --version`
Show version

STYLE (PRINTER)

`shfmt -i <n>`
Indent with n spaces (0 = tabs)

`shfmt -ci`
Indent case branches

STYLE (PRINTER) (cont.)

`shfmt -bn`
Binary ops like && and | may start a line

`shfmt -sr`
Space after redirect operators

`shfmt -i 2 -ci -bn`
Roughly Google's shell style

DIALECT AND PARSING

`shfmt -ln <dialect>`
Set dialect: bash, posix, mksh, bats, zsh

`shfmt -p`
Shorthand for -ln=posix

`shfmt -s`
Simplify code where safe

`shfmt --filename <name>`
Name for stdin input

`shfmt -f <path>`
Print shell files found, no formatting

CONFIG (.EDITORCONFIG)

.editorconfig

```
[*.sh]
indent_style = space
indent_size = 2
case_indent = true
binary_next_line = true
space_redirects = true
language_dialect = bash
```

indent_style / indent_size
Same as -i

case_indent, binary_next_line
Same as -ci and -bn