

NAME

`git-column` – Display data in columns

SYNOPSIS

```
git column [--command=<name>] [--[raw-]mode=<mode>] [--width=<width>]
            [--indent=<string>] [--nl=<string>] [--padding=<n>]
```

DESCRIPTION

This command formats the lines of its standard input into a table with multiple columns. Each input line occupies one cell of the table. It is used internally by other git commands to format output into columns.

OPTIONS

`--command=<name>`

Look up layout mode using configuration variable `column.<name>` and `column.ui`.

`--mode=<mode>`

Specify layout mode. See configuration variable `column.ui` for option syntax in **git-config(1)**.

`--raw-mode=<n>`

Same as `--mode` but take mode encoded as a number. This is mainly used by other commands that have already parsed layout mode.

`--width=<width>`

Specify the terminal width. By default *git column* will detect the terminal width, or fall back to 80 if it is unable to do so.

`--indent=<string>`

String to be printed at the beginning of each line.

`--nl=<N>`

String to be printed at the end of each line, including newline character.

`--padding=<N>`

The number of spaces between columns. One space by default.

EXAMPLES

Format data by columns:

```
$ seq 1 24 | git column --mode=column --padding=5
1  4  7  10  13  16  19  22
2  5  8  11  14  17  20  23
3  6  9  12  15  18  21  24
```

Format data by rows:

```
$ seq 1 21 | git column --mode=row --padding=5
1  2  3  4  5  6  7
8  9  10 11 12 13 14
15 16 17 18 19 20 21
```

List some tags in a table with unequal column widths:

```
$ git tag --list 'v2.4.*' --column=row,dense
v2.4.0 v2.4.0-rc0 v2.4.0-rc1 v2.4.0-rc2 v2.4.0-rc3
v2.4.1 v2.4.10  v2.4.11  v2.4.12  v2.4.2
v2.4.3 v2.4.4   v2.4.5   v2.4.6   v2.4.7
v2.4.8 v2.4.9
```

GIT

Part of the **git**(1) suite