

CLI Editor

nano

cli parameters/flags

- `-c`, `--const` - Constantly display the cursor position and line number on the statusbar. Note that this overrides `-U`.
- `-U`, `--quickblank` - Do quick statusbar blanking. Statusbar messages will disappear after 1 keystroke instead of 25. Note that `-c` overrides this.
- `-S`, `--smooth` - Enable smooth scrolling. Text will scroll line-by-line, instead of the usual chunk-by-chunk behavior.
- `-m`, `--mouse` - Enable mouse support, if available for your system. When enabled, mouse clicks can be used to place the cursor, set the mark (with a double click), and execute shortcuts. The mouse will work in the X Window System, and on the console when `gpm` is running.

set defaults by editing `/etc/nanorc` or in `~/.nanorc`: (some recommendations)

```
## Constantly display the cursor position in the status bar or minibar.
set constantshow

## Display line numbers to the left (and any anchors in the margin).
set linenumbers

## Enable mouse support, if available for your system.  When enabled,
## mouse clicks can be used to place the cursor, set the mark (with a
## double click), and execute shortcuts.  The mouse will work in the
## X Window System, and on the console when gpm is running.
set mouse

## Make the Home key smarter: when Home is pressed anywhere but at the
## very beginning of non-whitespace characters on a line, the cursor
## will jump to that beginning (either forwards or backwards).  If the
## cursor is already at that position, it will jump to the true start
```

```
## of the line (the left edge).  
# set smarthome  
  
## Spread overlong lines over multiple screen lines.  
# set softwrap
```

useful short cuts

shortcuts are displayed on the bottom, maximize to see more

- `Ctrl` + `k` cut selection or current line
- `Ctrl` + `u` paste at cursor position
-

vim

tutorials

- [VimTutor](#)
- <https://danielmiessler.com/study/vim/>

vim modes

- command mode (default)
You can switch to any mode from this mode. You can't do this in any other modes. To switch from one mode to another, you have to come to Command Mode first and then navigate to the other mode. The commands that you run without any prefix (colon) indicate that you're running the command in command mode.
- `i` - insert mode
This mode is used to edit the contents of the file. You can switch to insert mode by pressing `i` from command mode. You can use the `Esc` key to switch back to command mode.
- `:` - command-line mode
You can use this mode to play around with some commands. But the commands in this mode are prefixed with a colon (`:`). You can switch to this mode by pressing `:` (colon) in command mode.
- `v` - visual mode
You use this mode to visually select some text and run commands over that section of code. You can switch to this mode by pressing `v` from the command mode.

base commands

- create/edit and save/quit
 - `:edit sample.txt` create/edit (new file)
 - `:w` - save and continue
 - `:q` - quit
 - `:q!` - quit without save
 - `:wq` - save and quit
- modes
 - `i` - insert
 - `v` - visual
 - `ESC` - return to command mode (default start point)
- undo/redo
 - `u` - undo
 - `U` - undo a whole line
 - `3u` - undo 3 times
 - `:undolist` - list available undo options
 - `CTRL + R` - redo

movement

- general

```
^
k[] Hint: The h key is at the left and moves left.
< h l >[] The l key is at the right and moves right.
j[] The j key looks like a down arrow.
v
```

- jump to lines
 - `gg` - first line
 - `G` - end of file
 - `42G` - to line 42
- counted motions
 - `2w` move cursor to words forward
 - `3e` move cursor to the end of the third word
 - `0` move to line start

text editing

- motion list
 - `w` - until the start of the next word, EXCLUDING its first character.

- `e` - to the end of the current word, INCLUDING the last character.
- `$` - to the end of the line, INCLUDING the last character.
- inserting
 - `i` - insert before cursor
 - `A` - append after the line
- deletion
 - `x` - delete current character
 - `dw` - delete from cursor up to the next word
 - `de` - delete from cursor up to the end of the current word
 - `d$` - delete from cursor to the end of the line
 - `d0` - delete to the beginning of line
 - `dd` - delete whole line
 - `d2w` - delete two words
 - `d2d` - delete two lines
- copy/cut/paste
 - `y` - copy selected text/paste
 - `yy` - copy current line
 - `y5y` - copy 5 lines
 - `d` - cut selected text
 - `p` - paste copied text
 - `:t.` - duplicate current line
 - `yy`, `p` - duplicate current line

find and replace

searching only

- `/` search forward (forward slash)
- `?` search backward
- case insensitiv: append `\c`

example: `/Linux\c` + Enter

- after first search press `n` to search for the next occurrence or uppercase `N` to search backwards

find and replace

`:[range]s/{pattern}/{string}/[flags]`

- `[range]` indicates that you can pass the range of lines. The range is separated by a comma.
 - Pass `%` to find and replace in all lines
 - `5,10` To find and replace between lines 5 to 10
 - `.` current line
 - `$` the last line of the file

- `{pattern}` indicates the pattern to find the text. You can pass regex patterns here.
- `{string}` is the string to replace in the found text.
- `[flags]` indicates if you wish to pass any additional flags. By default, this does a case-sensitive search.
 - `i` case-insensitive search
 - `c` confirm before replacing
 - `g` indicates making the change globally

example

`:2,3s/Hi/Hello and Welcome/gci` -> replace with Hello and Welcome (y/n/a/q/l/^E/^Y)?

- `y` - Replace the match
 - `n` - Skip the match
 - `a` - Substitutes the match and all remaining occurrences of the match
 - `q` or `Esc` - Quit substitution
 - `l` - Replace the match and quit
 - `CTRL+Y` - Scroll the screen down
 - `CTRL+E` - Scroll the screen up
-

Created 2023-02-06 09:32:56 UTC by Daniel Raab
Updated 2025-07-08 11:56:35 UTC by Daniel Raab