

# Notes

## Text Editing

Vim has many tools for editing large text files. Some examples below.

[Idiomatic vimrc](#)

[OG Idiomatic vimrc](#)

[Revamped vim](#)

[Thoughtbot/blogs/vim](#)

## Binds

Either in normal mode after pressing `:`, or inserted into your `.vimrc`, map the input directly by starting the mapping:

```
nnoremap
```

then hitting `Ctrl + V` then the keystroke to record, e.g. `Alt + ←`. This will show as something like this:

```
nnoremap ^[[1;3D
```

but each `^[]` is a single, literal escape character and syntax highlighting should show this. Replace the literal escape(s) with the text `<Esc>`, and append the command you wish to use:

```
nnoremap <Esc>[1;3D <C-w>h
```

## Commands

These commands can be used directly in vim, or bound to a key within a custom `~/.vimrc`

You can bind `CTRL-B` to build a cmake project

```
nnoremap <C-b> :!cmake -S . -B ./build/ && cmake --build ./build
```

`C-I-(` Changes the contents of parenthesis `()` we are within

`C-I-{'` Changes the contents of brackets `{}` we are within

## Search / Replace

To search and replace text within an active vim session

```
:%s/search/replace/g
```

## Split Windows in Vim

Run the commands below to split windows while within a Vim session -

```
:split /path/to/file # To split horizontally
:vsplit /path/to/file # To split vertically

OR

:sp /path/to/file # To split horizontally
:vs /path/to/file # To split vertically

:open /path/to/file # To open a file within the active tab

:retab # To resize tabs in this session to your .vimrc configuration

# Split horizontal windows from bash
vim -o file.txt file2.txt file3.txt

# Split vertical windows from bash
vim -0 file.txt file2.txt file3.txt

# Tabbed windows from bash
vim -p file.txt file2.txt file3.txt
```

## Window Navigation

Use `Ctrl-w <Arrow Keys>` or `Ctrl-w <h j k l>` to move between split windows.

Use `Ctrl-w w` to move to the next window, `Ctrl-w W` to move to the previous.

Use `Ctrl-w s` to split active window horizontally, `Ctrl-w v` to split active window vertically.

Use `Ctrl-w z` to close an active preview or scratch window `Ctrl-w c`, `:q`, `:close`, or `:clo` to close the active window.

Close all other windows with `Ctrl-w o`, `:only`, or `:on`.

## Tab Navigation

Use `Ctrl-w PgUp` and `Ctrl-w PgDwn` to move between tabs within vim

# Visual Block Mode

Inserting text at the beginning of multiple lines

```
press Esc (to leave editing or other mode)
hit ctrl + v (visual block mode)
use the up/down arrow keys to select lines you want (it won't highlight everything - it's OK!)
Shift + i (capital I)
insert the text you want, i.e. %
press Esc Esc.
```

# Vim History

`less .viminfo` to view recent history in vim. could possibly recover lost files / information if needed.

# Runtime Path

“ The `.vim` directory under your home directory is used by Vim as the first place to search for vim scripts after starting up.

It's fine to add your own files, in fact a lot of plugins or plugin managers already do under their own subdirectory. Just be careful not to name your files or subdirectories anything that is already used by Vim. Take a look at the ['runtimepath'](#) documentation for a list.

- [Stack Exchange User](#)

---

Revision #18

Created 2019-07-21 05:15:19 UTC by Shaun Reed

Updated 2021-03-03 15:42:16 UTC by Shaun Reed