

# [vimtraining.com](http://vimtraining.com)

---

## Creating new files in Vim

### Terminal:

- `vim -u NONE` : Open Vim without any configurations

### Normal Mode:

- `<C-G>` : See current file stats in status bar
- `i` : Enter Insert Mode

### Command Line Mode:

- `:h / :help` : Open Vim help
- `:only / :on` : Close all but the current split
- `:q / :quit / ZZ` : Quit Vim or use the shortcut ZZ to save the buffer and quit vim
- `:w <filename>` : Save a new file with a name
- `:e <filename>` : Open a file using its path. Sift through the files in the current root folder by using `Shift-Tab`

### Insert Mode:

- `<Escape>` : Exit any mode and go back to the Normal Mode
- 

## Say hello to Vimrc

### Normal Mode:

- `dd` : Delete the line under the cursor

### Command Line Mode:

- `:e $MYVIMRC` : Opens the local vimrc file at `~/.vimrc`
- `:source $MYVIMRC` : Reloads the `.vimrc` file

### Vimrc

- `set number / set nu` : Show line numbers
- `set relativenumber / set rnu` : Show relative line numbers
- `syntax on` : Adds syntax based highlight
- `augroup, autocmd, BufWritePost` : Event based triggers for Vim

```
# Auto source vim files on save
augroup Reload_vimrc
  autocmd!
  autocmd BufWritePost *.vim* source $MYVIMRC | echom "Reloaded vim file..."
augroup END
```

---

## Operators in Vim

### Pre-requisite

git clone <https://github.com/typicode/json-server.git>

### Command Line Mode

- `:e .` : Opens netrw for exploring files
- `:vsplit/:vsp` : Opens a vertical split pane

### Normal Mode

- `h / j / k / l` : Navigate left/bottom/top/right respectively
- `u` : Undo
- `p` : Put
- `w` : Move by a word to the right

#### Operator Pending Mode

- `c{motion}` : Change using a motion
- `dd` : Delete a line
- `d{motion}` : Delete using a motion
- `yy` : Yank a line
- `y{motion}` : Yank using motion
- `gU{motion}` : Uppercase by motion, e.g. `gUw` Uppercases a Word
- `gu{motion}` : Lowercase by motion
- `>{motion}` : Indent right with a motion
- `<{motion}` : Indent left with a motion
- `>>` : Indent current line to right
- `<<` : Indent current line to left

#### Vimrc

- `set expandtab` : Adds spaces when using indent operator
- `set softtabstop=2` : Number of spaces added in edit mode while using Tab
- `set shiftwidth=2` : Number of spaces added when using indent operator

## Normal motions in Vim

#### Normal Mode:

- `0` : Go to start of the line
- `$` : Go to end of the line
- `^` : Go to the first non blank character in a line
- `g_` : Go to the last non blank character in a line
- `gm` : Go to middle of the screen width
- `w` : Forward by a Word
- `b` : Back by a Word
- `e` : End of Word
- `ge` : End of previous Word
- `W` : Forward by a WORD
- `B` : Back by WORD
- `E` : End of WORD
- `gE` : End of previous WORD
- `gg` : First line of the file
- `G` : Last line of the file
- `{line-number}gg` : Go to a specific line number
- `f{char}` : Find a character to the right
- `F{char}` : Find a character to the left
- `t{char}` : Find until a character to the right
- `T{char}` : Find until a character to the left
- `;` : Repeat the last find command
- `,` : Repeat the last find command in alternate direction

## Text-objects in Vim

#### Text Objects (In Normal mode)

Prefix:

- `i` : Inner
- `a` : All/around

Suffix:

- `w` : Word
- `W` : WORD
- `s` : Sentence
- `p` : Paragraph
- `"` : Quotes
- `[] / () / {}` : Braces, brackets and paranthesis
- `</>` : Indent
- `t` : Tag

E.g. Delete inside a word >> `diw`

---

## Scrolling in Vim

**Normal mode**

- `<C-d>` : Scroll down half a page
  - `<C-u>` : Scroll up half a page
  - `<C-u>` : Scroll up half a page
  - `<C-u>` : Scroll up half a page
  - `<C-f>` : Scroll full page forward
  - `<C-b>` : Scroll full page backwards
  - `zt` : Scroll cursor to the top of the screen
  - `zz` : Scroll cursor to the middle of the screen
  - `zb` : Scroll cursor to the bottom of the screen
  - `H` : Move cursor the the top of the screen
  - `M` : Move cursor to the middle of the screen
  - `L` : Move cursor to the bottom of the screen
- 

## Marks, jumps and changes in Vim

**Command Line Mode:**

- `:marks` : List the currently set marks
- `:view ~/.viminfo` : Look at numbered marks stored by Vim
- `:jumps` : List of jumps set by Vim
- `:changes` : List of changes

**Normal Mode:**

- `m{char}` : Set a local mark
  - `m{CHAR}` : Set a global mark
  - `\{char} / `{CHAR}`` : Column-wise Jump to local/global marks
  - `'{char} / '{CHAR}` : Line-wise Jump to local/global marks
  - `\\. `` : Jump to last edit position
  - `<C-o>` : Previous jump position
  - `<C-i>` : Next jump position
  - `g;` : Previous change position
  - `g,` : Next change position
- 

## Splits in Vim

#### Command Line Mode:

- `:split` : Create a horizontal split
- `:vsplit` : Create a vertical split
- `:resize N` : Resize split horizontally
- `:vertical resize N` : Resize split vertically

#### Normal Mode:

- `<C-w> <h/j/k/l>` : Move around within splits
- `<C-w> w` : Toggle between splits

#### Vimrc:

- `let mapleader = "<Space>"` : Defines the leader
  - `nnoremap` : Map keys in Normal mode in a non recursive way
- 

## Visual mode in Vim

#### Normal Mode;

- `v` : Enter Visual mode
- `<C-v>` : Enter visual block mode

#### Visual Mode:

- `o` : Go to other end of the visual selection
- 

## Autocomplete in Vim

#### Insert Mode:

- `<C-p> / <C-n>` : Scroll through autocomplete suggestions from open buffers
- `<C-x> <C-o>` : Open omni complete menu

#### Vimrc:

- `filetype plugin on` : Sets up plugins based on filetype
  - `set omnifunc=syntaxcomplete#Complete` : Sets omnifunc to show syntax based suggestions
- 

## CTags in Vim

#### Pre-requisite

Install CTags `brew install ctags`

#### Terminal

- `ctags -R .` - Generate ctags in the root folder

#### Normal

- `<C-]>` - Go to a definition

#### Command Line Mode

- `:ts` - List down definitions from a tag
  - `:tnext` : Next definition from the tags file
  - `:tprev` : Previous definition from the tags file
- 

## Pattern search in Vim

#### Normal Mode:

- `*` : Search word under the cursor
- `n` : Move to the next search result

- N : Move to the previous search result

#### Command Line Mode:

- :nohl : Remove highlight from search results
- /{pattern} : Start searching for a pattern forward
- ?{pattern} : Start searching for a pattern backwards

#### Vimrc:

- set hlsearch : Highlight search results
  - set incsearch : Sets incremental search to show results as we type
  - set ignorecase : Ignore case while searching
- 

## Global search using Ack in Vim

### Pre-requisites

Install Plug

```
curl -fLo ~/.vim/autoload/plug.vim --create-dirs
https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim
```

Install Ack `brew install ack`

#### Command Line Mode:

- :silent grep -r {pattern} {files/path} : Recursively grep for a pattern in a file/folder
  - :vimgrep /{pattern} {files} : Use vimgrep to search for a word
  - :clist : Shows list of errors in quickfix
  - :copen : Opens the quickfix list
  - :cclose : Close Quickfix Window
  - :colder : Open older quickfix list
  - :cnewer : Open newer quickfix list
  - :PlugInstall : Installs plugins using Plug for vim
  - nnoremap <C-f> :Ack<space> : Map to Ack
- 

## Fuzzy find files in vim

### Pre-requisites

Install fzf `brew install fzf`

#### Vimrc:

- Plug 'junegunn/fzf', { 'dir': '~/fzf', 'do': './install --all' } : Install fzf for vim
  - nnoremap <leader><leader> :FZF<CR> : Map fuzzy find
- 

## Repeating commands in Vim

### Normal Mode:

- . : Repeat the last command with the Dot command
- "y{char/CHAR} : Yank in a register
- "{char/CHAR}p : Paste from a register
- q{0-9a-zA-Z} : Record a command
- @{0-9a-z.=\*+} : Repeat the command
- @@ : Repeat the last played command

#### Command Line Mode:

- `:[range]s/{pattern}/{substitute}/{flags}` : Substitute in a file
  - `:registers` : Show current registers
  - `:[range]g/{pattern}/{command}` : Execute a command over a pattern in a range
  - `:[range]norm![command]` : Run a normal command multiple times over a range
- 

## Folding code in Vim

#### Normal Mode

- `zf{motion}` : Create a fold using a motion
- `zd` : Delete a fold
- `zD` : Delete folds recursively
- `zE` : Eliminate all folds in the window
- `zo` : Open a fold
- `zc` : Close a fold
- `za` : Toggle a fold
- `zM` : Close all folds in a file
- `zR` : Open all folds in a file

#### Visual Mode

- `zf` : Create a fold in the selected area

#### Vimrc

- `set foldmethod=indent` : Folds a file based on indentation
  - `set foldlevel=0` : Folds from the first indent
- 

## Shell commands and clipboard access in Vim

#### Command Line Mode

- `:%w !pbcopy` - Copy the visually selected text to system clipboard, % here would select the entire file
  - `:r !pbpaste` - Paste from the system clipboard
- 

## Workflow examples and conclusion

#### Command Line Mode

- `:[range]s/{pattern}/{substitute}/{flags}` : Substitute in a file
- `:buffers` - List all the open buffers
- `:b{number}` - Go to a buffer

#### Normal Mode

- `%` - Finds the matching brace
  - `gg=G` - Indent the entire file
- 

Checkout the full video course at [vimtraining.com](http://vimtraining.com)