

# Vi Improved

© 2013 Martin Bruchanov, bruxy@regnet.cz

## Basic control and conventions

- Key combination** – several keys pressed together, e.g. `CTRL+Ae` means `CTRL+SHIFT+ae`.
- Key sequence** – several keys or combinations pressed gradually, e.g. `CTRL+wwe` means a press of combination `CTRL+W` and then a key `w`.
- Working mode** – it is displayed bottom right:
  - C – basic command mode, `ESCe`.
  - E – insert mode `INSe`, `ie`.
  - V – visual, used for highlighting of text blocks `ve`, `Ve`.
  - ex – command line ex-mode `:e`.
- Repeat command with typing number  $n \times$  of repetitions before command, e.g. `5xe` deletes 5 chars from cursor position. Option `:set showcmd` shows command sequence in bottom right.
- Help: `:help topic`, use `Tabe` for command completion.
- Similar commands uses same keys, e.g. `fe`/`Fe`, `te`/`Te` (see below). The difference is direction or range of operation.
- Repeat last:..... command `;e`, macro `@Qe`.
- Undo:**..... `ue`  $\approx$  `:undo`, redo `CTRL+re` / `:redo`
- Set or unset options with `:set option` a `:set nooption`. Alone `:set` displays current options.
- Shortening of options:..... `:set ignorecase`  $\approx$  `:se ic`
- Line addressing** of ex-commands:
  - `:xcommand`..... execute on line  $x$
  - `:x,ycommand`..... from  $x$  to line  $y$
  - `:$command`..... from actual position to the end
  - `:%command`  $\approx$  `:1,$command`..... for whole file
  - `:.,+xcomm`  $\approx$  `x:comm`..... from cursor to next  $x$  lines
  - `:$-x`..... from current position to  $x$ . line from the end
  - `:/pattern1/,/pattern2/`..... from the first occurrence of line with  $pattern_1$  to the first  $pattern_2$
  - More ex-commands together:..... `:command1 | command2`
  - Comparison of two files:..... `vimdiff file1 file2`

## 2. Edit and save

- Edit:..... `:edit file`
- Open new empty file:..... `:enew`
- Insert file to current position:..... `:read file`
- Insert output of system `command`:..... `:read !command`
- Save/append file:..... `:write file/ :w >> file`
- Save/append lines  $n$  to  $m$ :..... `:n,mw file/ :n,mw>> file`
- Save and quit:..... `:wq`, enforced `:wq!`
- Quit without save:..... `:q!`  $\approx$  `ZQe`
- Quit and save when file was changed:..... `:x`  $\approx$  `ZZe`
- Go to file under the cursors:..... `gfe`
- Change name of current file:..... `:f jméno`
- Insert before/after:..... char `ie`/`ae`, line..... `Ie`/`Ae`
- Insert new line above/below cursor:..... `Oe`/`oe`
- Rewrite:char `re`, word `cwe`, line `cce`, text `Re`  $\approx$  `INSINSe`
- Delete char/line and continue edit:..... `se`/`Se`

### 2.1. Moving thought file

- Open file on 23<sup>rd</sup> line:..... `vim file +23`
- Cursor:.....  $\rightarrow$   $\approx$  `le`,  $\leftarrow$   $\approx$  `he`,  $\uparrow$   $\approx$  `ke`,  $\downarrow$   $\approx$  `je`
- Paging:..... `CTRL+fe`  $\approx$  `PAGEUPe` / `CTRL+be`  $\approx$  `PAGEDOWNe`
- Half-paging up/down:..... `CTRL+ue` / `CTRL+de`
- Jump to  $n^{\text{th}}$  line:..... `:n`  $\approx$  `nGe`, the last `:$`  $\approx$  `Ge`
- Jump to percentual line position:..... `n%e`
- Display current line number:..... `CTRL+ge` `:set number`
- Display number of lines:..... `=`
- Jump to byte position:..... `:goto n`
- Jump in line before char  $a$  forward/backward:..... `tae`/`Tae`
- Jump in line to char  $a$  forward/backward:..... `fae`/`Fae`
- Repeat last line search forward/backward:..... `;/`/`;e`

- Go to beginning of next/previous word:..... `we`/`be`
- Go to end of a word:..... `ee`..... with interpunction `Ee`
- Go to declaration:..... `gde`
- Go to begin/end of paragraph:..... `{e`/`]e`, block: `[{e`/`]}`
- Go to begin/end of line:..... `HOMEe`  $\approx$  `^e` / `ENDe`  $\approx$  `$e`
- Go to next/previous sentence:..... `>e`/`<e`
- Go to next/previous line:..... `ENDe`  $\approx$  `+e` / `-e`
- Go to column:..... first `Oe`  $\approx$  `|e`,  $n^{\text{th}}$  `n|e`
- Jump to pair bracket:..... `%e`
- Print position, number of lines and chars, etc.:..... `gCTRL+ge`

### 2.1.1. Marking of position

- Insert invisible mark  $a$ :..... `mae`
- Jump to mark  $a$ :..... `'ae`
- Print marks:..... `:marks`
- Erase mark  $a$ :..... `:delmarks a`
- Process text from mark  $a$  to  $b$ :..... `:'a,'b příkaz`, e.g. `:'a,'b !sort` will use external command for sort

### 2.2. Deleting, moving and copying of text

- Deletion automatically stores text in register (for moving)
- Delete:char `xe`  $\approx$  `DELe`, previous char `Xe`  $\approx$  `d<-e`  $\approx$  `dhe`, `CTRL+he`  $\approx$  `BACKSPACEe`, word `dwe`  $\approx$  `CTRL+we`, line `dde`
- Delete up to pair bracket:..... `d%e`
- Delete to end/beginning of sentence:..... `d]<e` / `d[<e`
- Delete the rest of line/file:..... `De`  $\approx$  `d$e` / `dGe`
- Delete from cursor position to beginning:..... `d0e`  $\approx$  `d^e`
- Join lines:..... `Je` add space,..... `gJe` without space
- Copy to register:..... line `Ye`  $\approx$  `yye`, word `ywe`
- Copy to the beginning/end of line:..... `y0e`/`y$e`
- More registers: press `"k...e` before command will save text to register  $k$ . E.g. `"kyye` copies line into  $k$
- Paste content of last used register or last deleted text: after `pe` before `Pe` cursor, `"kppe` pastes register  $k$
- Special register: `%` – current file name, `/` – last search,...
- Show content of registers:..... `:register a`
- Swaps the next two chars/lines:..... `xpe` / `ddpe`
- ex-mode:..... delete `:d`, move `:m`, copy `:co`  $\approx$  `:t`
- `:10,20m0` – move lines 10–20 to the beginning (0)
- `:.+5d a` – delete next 5 lines and store them to reg.  $a$

### 2.3. Formatting

- Set text width:..... `:set textwidth=n`
- Switch on line wrap:..... `:set wrap`
- Justify:..... `:center`, `:right`, `:left`
- Format line:..... `ggqe`
- Format paragraph:..... `ggape` jump to end, `gwape`
- Reverse case of char:..... `~e` line `g~e`
- Line to upper/lower:..... `gUUe` / `Vue`  $\approx$  `guue` / `VUe`
- Whole file to upper/lower case:..... `gggUGe` / `ggguGe`
- While line/file `ROT13` encoding:..... `g??e` / `ggg?Ge`

### 2.3.1. Indentation

- Shift line:..... `>>e`, `CTRL+te` of `:set shiftwidth=n`
- Shift line leftwards:..... `<<e`, `CTRL+de`
- Tabbing:..... `TABe`  $\approx$  `CTRL+ie`
- Shift line same as before:..... `=e`
- Indent block relatively:..... `=ae`
- Refresh indentation of whole file:..... `gg=Ge`

### 2.4. Complementation

- Complement word before/after cursor:..... `CTRL+pe` / `CTRL+ne`
- Name of file in working directory:..... `CTRL+x` `CTRL+fe`
- Name of directory:..... `CTRL+x` `CTRL+de`
- Whole line:..... `CTRL+x` `CTRL+le`
- Word from dictionary:..... `CTRL+x` `CTRL+ke`
- System command (reads \$PATH):..... `CTRL+x` `CTRL+ve`
- Increase/decrease number under cursor:..... `CTRL+ae` / `CTRL+xe`
- Name of current file:..... `CTRL+r%e`

#### 2.4.1. Completion at ex-command line

- Insert content of register  $a$ :..... `CTRL+rae`
- Insert word under the cursor:..... `CTRL+r` `CTRL+ae`

- Cancel command:..... `CTRL+ue`

### 2.5. Insert special sumbols

- Display unprintable characters:..... `:set list`
- Insert ASCII char:..... `CTRL+v codeENDe`
- Show ASCII/UTF-8 of char under cursor:..... `gae` / `g8e`
- Suppress control character, e.g. `TAB` (`^I`):..... `CTRL+v TABe`
- Table of Unicode digraphs:..... `:digraphs`
- Insert digraphs:..... `CTRL+k K1 K2`
- Digraphs:  $\pm$ +- °DG ×\*X ÷-: --N --M „:9 “:6
- Define digraphs `.3` for „...“ (U+2026):..... `:dig .3 8230`

## 3. Search and replace

- Ignore case:..... `:set ignorecase`, opposite `:set noic`
- Search forward/backward:..... `/pattern/` / `?pattern/`
- Repeat last search:..... `/` / `?`
- Ignore case:..... `/\cpattern/`
- Jump to next searched place:..... `ne` forward, `Ne` backward
- Search for word under the cursor:..... `#e` forward, `*e` backward
- Search by ASCII code of „M“:..... `%d77, \%x4d, \%o0115`
- Search Unicode character „a“ (U+03B1):..... `%u03b1`
- Substitution:..... `:substitute/from/to/flag`, abbr. `:s`
- Repeat last substitution:..... `:s`  $\approx$  `:s//~`
- The delimiter `/` can be any char instead `\`, `"` and `|`
- Flags: `c` – confirmation, `g` – for whole line, `p` – print, without flag changes only first occurrence
- Cancel highlight:..... `:nohlsearch`
- Search in more files:..... `:vimgrep /pattern/g f1 f2 ...fn`
- Execute command on line with: `:global/pattern/command`
- Execute where pattern not match:..... `:g!/pattern/command`

### 3.1. Regular expressions

- Begin/end of:..... line `^` / `$`, word `\<` / `\>`, file `%^` / `\/%`
- Last searched pattern/substitution:..... `&` / `~`
- Matches any character:..... (dot)
- Or/and:..... `|` / `&`
- Suppress next special symbol:..... `\.`, `\&`, `!`, `^`, `\\`, `\\*`
- Atom:..... `\(pattern\)`
- $N^{\text{th}}$  reference:..... `\N`,..... first `\(pattern\)` = `\1`
- Ranges:..... `[abcd]`  $\approx$  `[a-d]`, `[a-zA-Z]`  $\approx$  `[a, [0-9]`  $\approx$  `d`
- Ignored chars and ranges:..... `[^abcd]`  $\approx$  `[^a-d]`
- Zero or multiple  $z$ :..... `z*`  $\approx$  `z\{0,\}`
- One or multiple  $z$ :..... `z+`  $\approx$  `z\{1,\}`
- One or zero times:..... `z?`  $\approx$  `z\{0,1\}`
- Repetition  $z$ :..... `z\{n\}` accurate  $n \times$ , `z\{n,m\}` minimal  $n \times$ , `z\{n,m\}` min.  $n$  and max.  $m$  times

## 4. Folding of text blocks (:help folding)

- Fold text:..... `zfe`, selected `zfve`
- Jump to beginning/end of fold:..... `[ze` / `]ze`
- Open fold under the cursor:..... `zoe`, recursively `zOe`
- Open/close all:..... `zRe` / `zMe`
- Switch fold-unfold:..... `zae`, recursively `zAe`
- Highlight folded parts:..... `:set foldcolumn=5`

## 5. Edit more files, windows and tabs

### 5.1. More files (edit buffers)

- Start vim:..... `vim file1 file2...`
- Display buffers:..... `:ls`  $\approx$  `:files` (numbered)
- Switch to next/previous file:..... `:next` / `:prev`
- Switch to first/last file:..... `:rewind`  $\approx$  `:first` / `:last`
- Perform mass `command` for all files:..... `:bufdo command`

### 5.2. Windows

- New window:..... `:new`  $\approx$  `CTRL+wne`
- Split window:..... `:split`  $\approx$  `CTRL+wse`
- Vertically split window:..... `:vsplit`  $\approx$  `CTRL+vwe`
- Close window:..... `:close`  $\approx$  `CTRL+wce`
- Close all except current:..... `:only`  $\approx$  `CTRL+woe`
- Jump to next window:..... `CTRL+wwe`
- Shrink/large window by  $n$  lines:..... `CTRL+wn-<e` / `CTRL+wn+>e`

- Shrink/large window by  $n$  columns:..... `CTRL+wn<e` / `CTRL+wn+>e`
- Same size for all windows:..... `CTRL+w=<e`
- Mass:..... write `:wall`, quit `:qall`, `:wqall`
- Perform mass `command` for all windows:..... `:windo command`
- Open name under cursor in new window:..... `CTRL+wfe`

### 5.2.1. Moving in window

- Jump to window line:..... top `He`, middle `Me`, last `Le`
- Jump to  $n^{\text{th}}$  line from top/bottom:..... `nHe` / `nLe`
- Set actual line to:..... top `zENDe`, center `z.ce`, bottom `z-e`
- Scroll window up/down by:line `CTRL+ee` / `CTRL+ye`, page `CTRL+be` / `CTRL+fe`, half-page `CTRL+ue` / `CTRL+de`
- Redraw window without scrolling:..... `CTRL+le`

### 5.3. Tabs

- Command:..... `vim -p f1, f2 ...` open files in tabs
- Create/close tab:..... `:tabnew` / `:tabc`
- Close all, except current tab:..... `:tabo`
- Go to next/previous:..... `:tabn`  $\approx$  `CTRL+PAGEDOWNe` / `:tabp`  $\approx$  `CTRL+PAGEUPe`
- Move after the fold  $n$ :..... `:tabm n`
- Perform mass `command`:..... `:tabdo command`
- Change tab design:..... `:set tabline`
- Open name of file under cursor in new tab:..... `CTRL+wgfe`

## 6. Visual mode

- Select:..... chars `ve`, lines `Ve`, cursor sets range
- Block text selection:..... `CTRL+Ve`
- Change to upper/lower case:..... `U` / `u`
- Select:..... word `awe`, `()/{}|`block `abe` / `aBe`, no brackets `ibe` / `iBe`, XML block with marks `ate`, inside marks `itve`
- Perform `command` in selected range:..... `:'<,'> command`

## 7. Abbreviations, key maps and macros

- Define abbreviation:..... `:abbreviate abbr full expansion`
- Abbreviations for insert/ex-mode:..... `:iabbrev` / `:cabbrev`
- List/delete abbreviations:..... `:ab` / `:unab abbreviation`
- Map key  $x$ :..... `:map x commands`
- Map by mode:..... `:map` `:imap` `:cmap` `:nmap` `:vmap`
- Map key for insert mode:..... `:map!` `x commands`
- List key mapping:..... `:map` / `:map!`
- Delete mapping:..... `:unmap x` / `:unmap! x`
- Write macro to register  $a$ :..... `qae`
- End macro definition:..... `qe`
- Perform macro in register  $a$ :..... `@ae`
- Repeat last called macro:..... `@@e`
- Execute macro  $a$  as ex-command:..... `:@a`

## 8. Spell checking

- Switch on spelling:..... `:set spell spelllang=cs,en`
- Search next/previous mistype:..... `lse` / `lSe`
- Add to dictionary a word under cursor:..... `zge`
- Add temporary to dictionary:..... `zGe`
- Set word as bad:..... `zwe`, temporary `zWe`
- Suggest correctly spelled words:..... `z=e`
- Always take the first suggestion:..... `lz=e`

## 9. Programmer support

- Run make:..... `:make`
- Jump to next/previous error:..... `:cprevious` / `:cnext`
- List all errors:..... `:clist`
- Add window with list of errors:..... `:copen`

## 10. Other

- Start shell:..... `:shell`
- Open manpage of command under cursor:..... `Ke`
- Open file under the cursor:..... `gfe`
- Save as color HTML:..... `:runtime! syntax/2html.vim`
- Wildcard of current file  $%$ , e.g. `!:context %` process current file with `ConTeXt`