Vim Cheatsheet

Mandatory get out of Vim joke

Ughhh how do I escape Vim?

```
# <ESC> :wq, write the changes to the file and quit
# <ESC> :q, quit if there is no changes
# <ESC> :q!, quit without saving
```

Basics

```
            ^
            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
```

This is how you would be moving your cursors around in the file

They can be preceeded by a number to tell how many lines to say go down or go to the left.

For example: Pressing 10 and then h will move your cursors to the left 10 characters.

Command	**Description**
`x`	Delete the character at the cursor
`i, a, A`	Insert character at the cursor, insert character after the cursor (append), and append to the end of the line respectively.
`<ESC>`	Put you into normal mode if you're in insertion mode, replace mode, or other modes
`w, e, b`	Use it to traverse skip through each word to the start of each other, or to the end of each word. Use `b` to go back a word.

`dw, de, d$, dd`	Delete from cursor up to the next word Delete from cursor up to the end of the word only Delete from cursor to the end of the line Delete the entire line regardless where your cursor is
`operator [number] motion`	Operator such as `d` can be specified together with a number to tell it how many times to repeat the motion. Motion tells which text to operate on, `w, e, $`
`u, U, CTRL-R`	Undo previous action Undo all the changes on the current line Redo, (undo the undo)
`p`	Paste what was deleted with the `d` operator
`rx, R`	Replace the current character with x, where x can be any character Enter in replace mode where every character you type will be replacing the one that's on the cursor.
`CTRL-G`	Display current line number
`G, <NUMBER> G`	Move to the end of the file Move to the line number specified
`gg`	Move to the first line of the file

`/, ?`
`n, N`
`CTRL-o, CTRL-i`
	Used for searching patters forward and backward respectively
	Used to find the next occurrence of the pattern in the direction specified, `N` is used to find the previous occurrence.
	`CTRL-o, CTRL-i` take you back to the older/newer position so you don't have to scroll back to where you are respectively

`%`	Used to find the matching ( ), [ ], { }

`:!command`	Will execute external command, like `ls`
`:w FILENAME`	Write the current Vim file to another file called FILENAME
`v`	Visual mode, let you see what you are highlighting and then you can apply other operator such as deleting the highlighted text.
`:r FILENAME`	Read in the specified file and then paste it below the cursor Can be used together with command to paste the output of the command. `:r !ls`

Some more command

Command	**Description**
`o, O`	To insert a empty line below and above the current line
`y`	Copies text, `yy` copies the entire line. Can be used together with `v` to highlight a good section of the file and then paste it with `p`
`CTRL-W CTRL-W`	To jump to another opened window
`:terminal`	Open up a terminal within Vim this is pretty cool

Set command

Some settings in Vim can be changed for example if you want to show line numbers you can do `:set number` to toggle on displaying the line number on the side.

You can toggle it off by prepending a `no` to the original toggle, for example `:set nonumber` will disable displaying the line number on the side.

Revision #2
Created 2023-05-07 16:08:34 UTC by Tamarine
Updated 2023-05-07 17:07:09 UTC by Tamarine