

# Vim and Emacs

---

Seongjin Lee

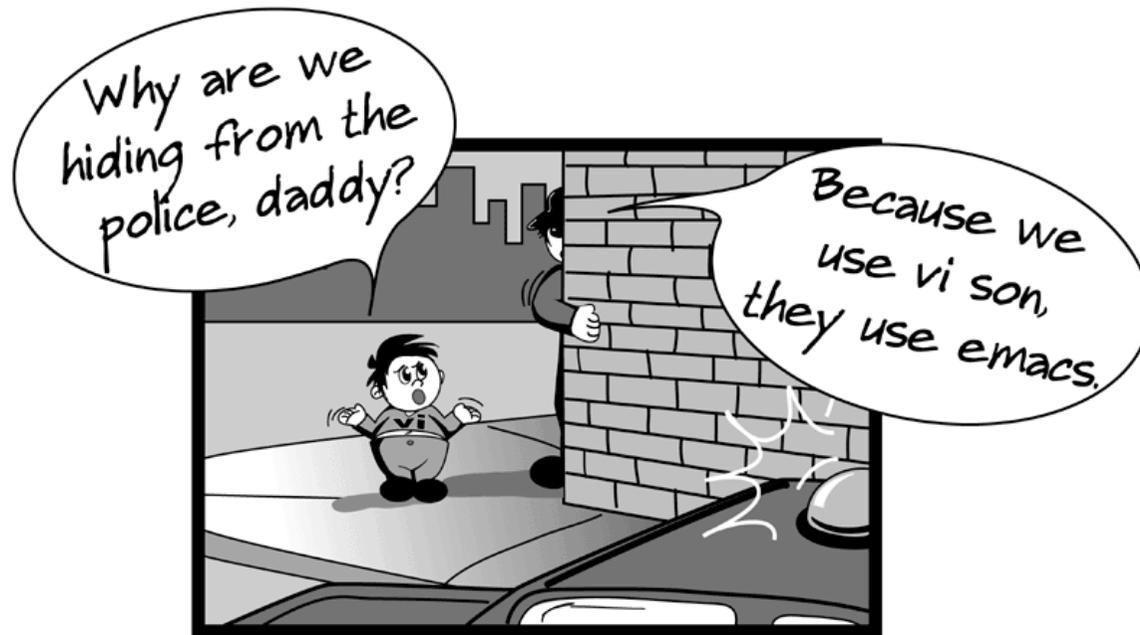
# Vim

---

# Contents

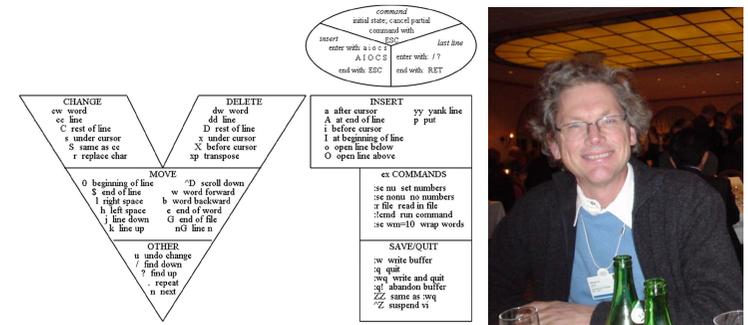
---

- What is vi
- Modes of Vi
- Moving Around and Deleting in vi
- Replacing, Searching, and Substituting in vi
- Undo and Redo in vi
- Simple Tutorial: From Starting to Quitting vi
- Learn by Doing
- References



# What is vi

- **Vi** is a visual screen text editor developed by Bill Joy, who later becomes co-founder of Sun Micro Systems.
  - It is visual version of **ex**, a Unix line editor
  - Vi is available on most Unix Systems
  - Works with a variety of terminals
  - Allows **ex** command from **vi**



The image contains several cheat sheets for the vi editor. On the left is a large 'V' shape containing commands for CHANGE, DELETE, MOVE, and OTHER. On the right is a smaller cheat sheet with sections for INSERT, ex COMMANDS, and SAVE/QUIT. Above the smaller cheat sheet is a diagram showing cursor movement with terms like 'first line', 'last line', and 'end with ESC'. To the right of the diagrams is a photograph of Bill Joy, the creator of vi.

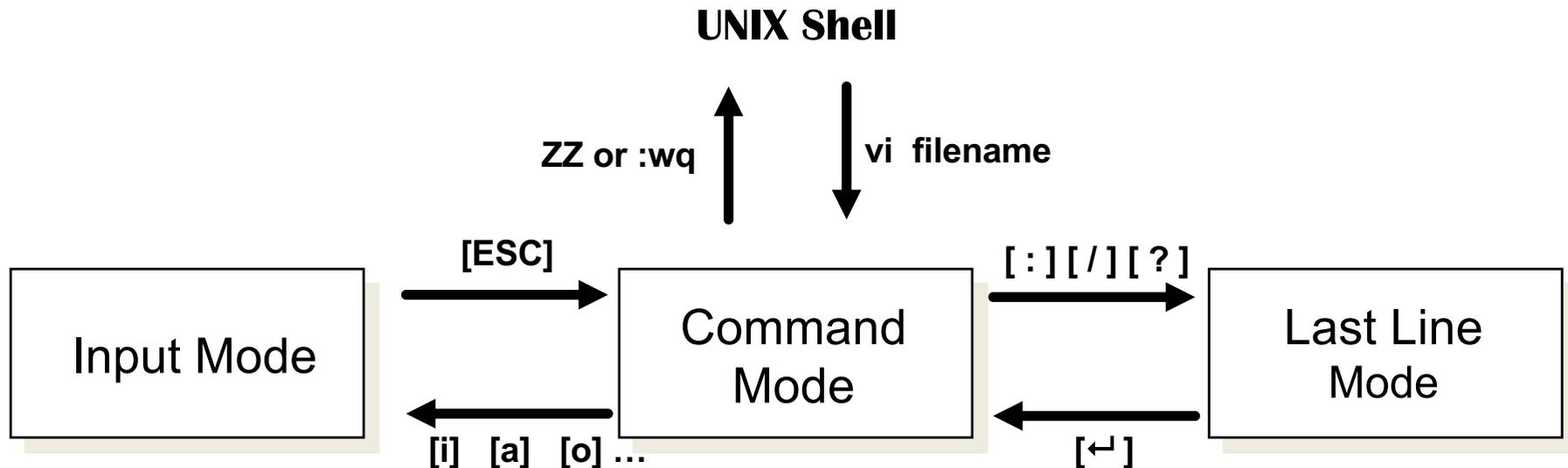
- **VIM** is acronym for **Vi** Improved, developed by Bram Moolenaar, a extended version of vi and some of enhancements include
  - Completion, comparison, and merging of files
  - Split and tabbed windows
  - Command histories
- All editing session before saving is done in buffer area
  - Nothing is saved as hard data, until you save it



# Modes of vi

---

- There are three mode in vi
  - Command Mode – A default mode in vi
    - Everything is command before you enter into other modes
  - Input Mode – What you type is what you see
    - Anything typed in this mode is considered as data
    - Pressing [ESC] always leads to Command mode
  - Last Line Mode – Only can be accessed from Command mode
    - Three ways to enter Last Line Mode – : (Colon) / (Back Slash) ? (Question Mark)



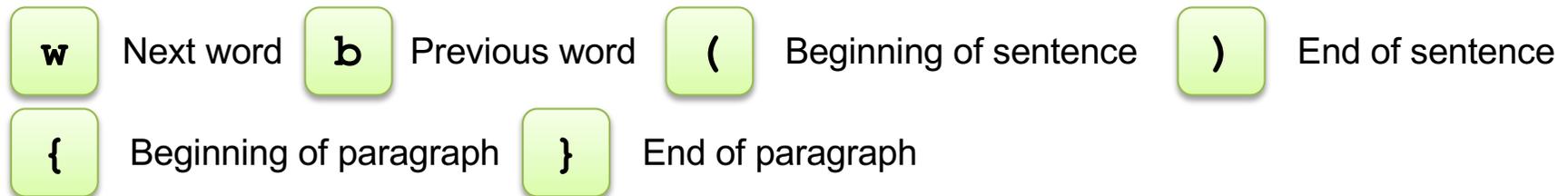
# Moving Around

---

- Moving cursor to a direction
  - VI uses four characters to move around each mapped to a direction



- Moving by units of word, sentence, paragraph
  - E.g., 3w moves to three words after the current cursor



## Deleting in vi

---

- Deleting a character, words, sentence, line, and paragraph
  - “x” erases a character
  - Combination of direction commands with “d” erases a word, sentence, and paragraph.
    - E.g., dw erases a word before the cursor
  - “dd” erases a line
  - D to delete rest of line
  - X to delete before the cursor
  - Xp to transpose

# Searching and Replace

---

- Searching in vi is done in last line mode
  - “/” lets you search a character, word, and words
    - E.g., “/abc” moves the cursor to the location of the pattern
  - Search pattern in forward direction: n, backward direction: N
  - Regular expressions can be also used in searching
  
- “r” replaces a character
  - Suppose the cursor is on “b”, and by “[r] p” we can change it to “preview”

This is breview.



This is preview.

# Substitution

---

- Substituting in vi is done in last line mode
- Find i and substitute with X once



- Find i and substitute with X in the same line



- Find i and substitute with X in all the lines



# Undo and Redo in vi

---

- Undo in vi is done by “u”
  - Or to do in last line mode you could type in “:undo”
  - “U” undo all latest changes on one line
- Redo in vi is done by “CTRL-R”
  - Or to do in last line mode you could type in “:redo”

## Simple Tutorial: From Starting to Quitting vi

---

- This simple tutorial illustrates how to write, delete, copy, paste, replace, save, and quit
- Start vi by ***vi newfile.txt*** and type the following

```
[ i ] This is how we write [ esc ] [ o ] and copy lines  
[ esc ] [ k ] [ y ] [ y ] [ j ] [ p ] [ k ] [ y ] [ 3 ]  
[ w ] [ j ] [ ) ] [ a ] [ space bar ] [ esc ] [ p ]  
[ o ] ummm [ esc ] [ b ] [ x ] [ r ] E [ l ] [ r ] N  
[ l ] [ r ] D [ : ] [ w ] [ q ] [ enter ]
```

- This will produce following and goes back to command prompt

```
This is how we write  
and copy lines  
This is how we write and copy lines  
END
```

# Simple Tutorial: From Starting to Quitting vi

---

```
[ i ] This is how we write [ esc ] [ o ] and copy lines
[ esc ] [ k ] [ y ] [ y ] [ j ] [ p ] [ k ] [ y ] [ 3 ]
[ w ] [ j ] [ ) ] [ a ] [ space bar ] [ esc ] [ p ]
[ o ] ummm [ esc ] [ b ] [ x ] [ r ] E [ l ] [ r ] N
[ l ] [ r ] D [ : ] [ w ] [ q ] [ enter ]
```

- Explaining the commands in the tutorial

**i** Insert      **o** Add new line after current line      **a** Append

**k** Moving cursor up      **j** Down      **l** Right      h: moves cursor left

**y** Yank a line      **p** Paste after cursor point

**w** Moves cursor to next word      **b** Previous word      **)** End of sentence

**r** Replace a character      **x** Erase a character      **3** Number Repeats command

**:** Enters last line mode      **w** Writes to data file      **q** Quits vi (only in last line mode)

## Learn by Exercise

---

[Line 1]: The five boxing wizards jump quickly.

[Line 2]: See the quick brown fox jump over the lazy dog.

[Line 3]: A mad boxer shot a quick, gloved jab to the jaw of his dizzy opponent.

[Line 4]: We promptly judged antique ivory buckles for the next prize.

[Line 5]: A quart jar of oil mixed with zinc oxide makes a very bright paint.

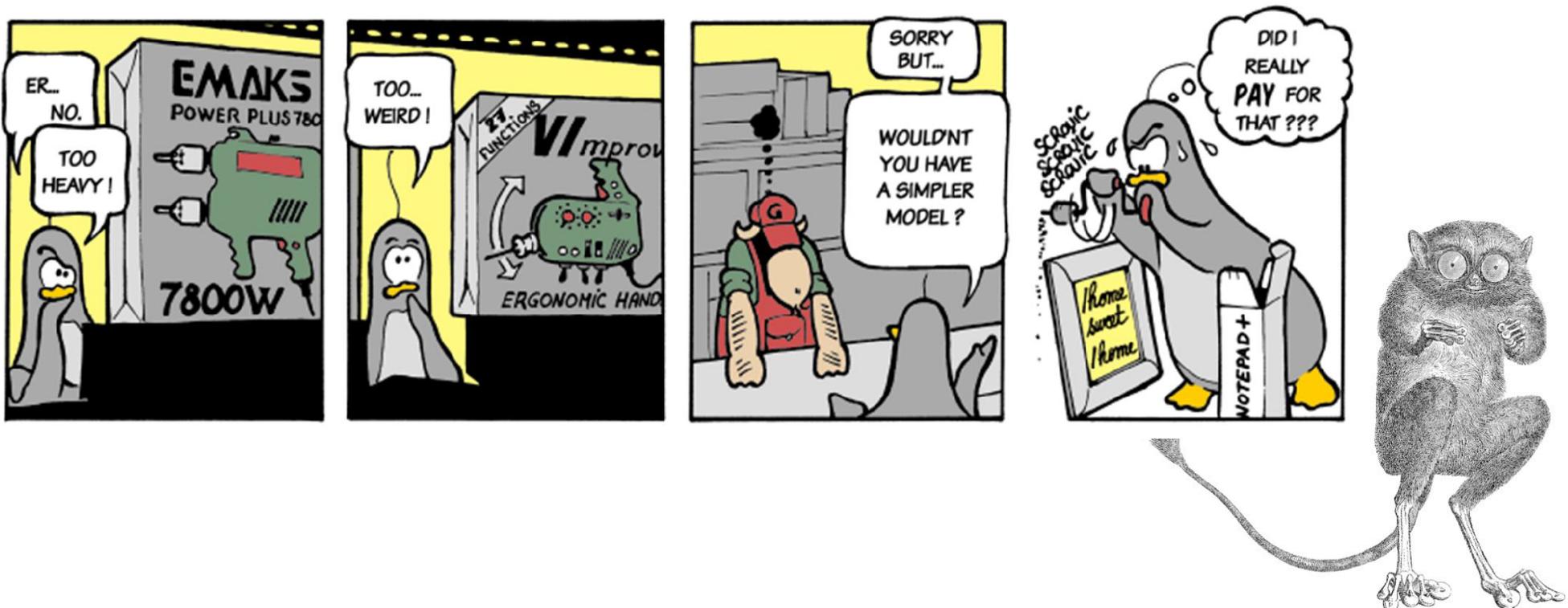
[Line 6]: The job requires extra pluck and zeal from every young wage earner.

- Complete all tasks with minimum number of retyping, but with commands
  1. Substitute all j's to z and all z's to j
  2. Copy lines 1, 3, 5, and 6, and make new paragraph with those lines
  3. Delete three words "requires extra pluck," and type in "need lot of money" in the place
  4. Add "caps" at the end of all words with "w", e.g., wizards to "wizardscaps"

# References

---

- Graphical cheat sheet of Vi and VIM
  - [http://www.viemu.com/a\\_vi\\_vim\\_graphical\\_cheat\\_sheet\\_tutorial.html](http://www.viemu.com/a_vi_vim_graphical_cheat_sheet_tutorial.html)
- Cursor movement Commands
  - <http://www.kcomputing.com/vi.html>
- List of Commands
  - <http://www.smashingmagazine.com/2010/05/03/vi-editor-linux-terminal-cheat-sheet-pdf/>



# Emacs

---

# Contents

---

- What is Emacs
- What is Emacs and Why use it ?
- Emacs Preliminaries
- Command Combinations
- Moving Around and Deleting in Emacs
- Searching, and Substituting, Redo in Emacs
- Simple Tutorial: From Starting to Quitting Emacs
- Learn by Doing
- References



# What is Emacs and Why use it ?

---

- Emacs (Editor MACroS) is the extensible, customizable, self-documenting, real-time display editor
  - Richard Stallman is the author of Emacs; the author of GCC and GDB
  - Runs on LISP engines + lots of LISP libraries
- It is not the only good choice, there are options like VI, VIM
  - Works on many platforms and independent of GUI
  - Extremely powerful
  - **vi** often does things with fewer keystrokes, but **emacs** easily surpass vi when it comes to searching and replacing and using macros
- Some of assumptions of Emacs are
  - No mouse! – Much more reliable and much faster for experienced user
  - No particular keyboard; No particular GUI environment
  - Runs through telnet (as well as directly)



**Richard Stallman**  
The founder of GNU

# Emacs Preliminaries

- In the emacs documentation, key sequences described as:
  - C-e – This is [Ctrl-e]
  - C-x C-b – This is [Ctrl-x] [Ctrl-b]
  - C-x b – This is [Ctrl-x]b
  - M-e – This is [Meta-e]
- On the PC, you can use the [Alt] key or [Esc]-release to substitute [Meta] key
- When you press a valid key sequence, emacs executes a command associated with the key

The screenshot shows the Emacs interface with the 'Buffers' menu open. The menu lists several buffers: 'temp' (pointing to /home/james/), '\*scratch\*', '\*Messages\*', '\*GNU Emacs\*', and '\*Buffer List\*'. Below the menu, a table displays the buffer list with columns for Buffer Name, Size, Mode, and File. The table content is as follows:

Buffer Name	Size	Mode	File
temp	0	Fundamental	~/temp
*scratch*	191	Lisp Interaction	
*Messages*	1428	Fundamental	
*GNU Emacs*	679	Fundamental	

Annotations in the image include:

- Shortcut key**: Points to the 'List All Buffers C-x C-b' menu item.
- Pop up a window listing all Emacs buffers**: A text box pointing to the Buffer List window.
- Edit modes**: A section listing '- Not modified', '\* modified', and '% read-only'.
- WINDOW**: Points to the Buffer List window.
- Window Display position**: Points to the buffer name 'temp' in the table.
- Line Number**: Points to the 'All' column in the table.
- Buffer major/minor MODE name**: Points to the 'L1' column in the table.
- Modeline**: Points to the '(Buffer Menu)' text at the end of the line.
- MINIBUFFER**: Points to the bottom line of the Emacs window showing '\*Buffer List\* All L1 (Buffer Menu)'.

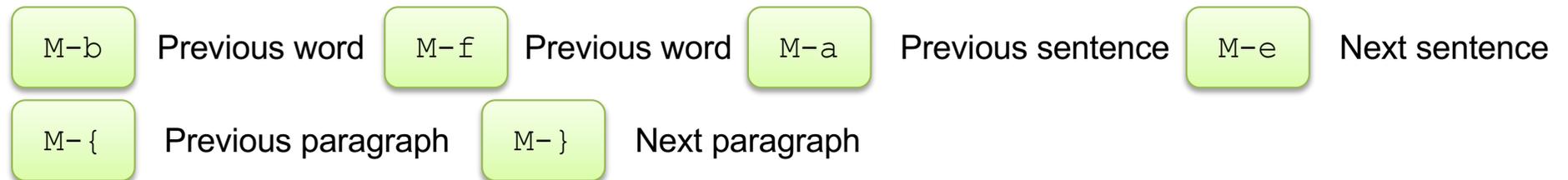
# Moving Around and Deleting in Emacs

---

- Emacs uses the control keys to move in the four directions



- To move by units of word, sentence, and paragraph



- Delete a word, line, and sentence



- When in Doubt, use “Get me out of here” command – [Ctrl- g], c-g

# Searching, Substituting, and Redo in Emacs

---

- “`^s`” search asks for search pattern
  - “`^s`” again to search next pattern “`^r`” for pervious pattern
  - Regular expressions can be also used in searching with “`^M-s`”
- Substituting in emacs “`M-%`”
  - `M-%` requests for search pattern; press enter to enter substituting pattern
  - Replacing to the substituting pattern this once [SPC]
  - Skipping to the next without replacing [DEL]
  - Replace all remaining matches [ ! ]
  - Exiting replace command by [RET]
- Undo and Redo
  - Undo an unwanted change is done by [`^_`]
  - Redo is reverse of undo, undo direction is reversed by [`^x ^_`]

# Macro in Emacs

---

- Macros are useful for repeatable key sequences that may include commands.
- Common macro commands
  - C-x - ( – begin macro definition (after this, type whatever actions you would like repeated and stored)
  - C-x - ) – end macro definition
  - C-x - e – execute stored macro
  - C-u 5 C-e – execute stored macro 5 times (Note: C-u 5 can prefix any emacs cmd, even a non-cmd)

# Simple Tutorial: From Starting to Quitting Emacs

---

- One can type without having to use complex commands but here we go
- Start emacs by *emacs newfile.txt* and type the following

```
This is how we write [ c-q-j ] and copy lines [ c-p ]  
[ c-a ] [ c-spac ] [ c-e ] [ m-w ] [ m-} ] [ c-q-j ]  
[ c-y ] [ c-p ] [ c-a ] [ c-spac ] [ c-u ] [ 3 ] [ m-f ]  
[ m-w ] [ m-} ] [ spac ] [ c-y ] [ enter ] ummm [ c-a ]  
[ c-u ] [ 4 ] [ c-d ] END [ c-x-s ] [c-x-c]
```

- This will produce following and goes back to command prompt

```
This is how we write  
and copy lines  
This is how we write and copy lines  
END
```

# Simple Tutorial: From Starting to Quitting Emacs

```
This is how we write [ c-q-j ] and copy lines [ c-p ]  
[ c-a ] [ c-spac ] [ c-e ] [ m-w ] [ m-} ] [ c-q-j ]  
[ c-y ] [ c-p ] [ c-a ] [ c-spac ] [ c-u ] [ 3 ] [ m-f ]  
[ m-w ] [ m-} ] [ spac ] [ c-y ] [ enter ] ummm [ c-a ]  
[ c-u ] [ 4 ] [ c-d ] END [ c-x-s ] [c-x-c]
```

- Explaining the commands in the tutorial

C- is denoted as ^ in this example

^qj

Add a new line

^p

Move to previous line

^a

Beginning of the sentence

^spc

Begin highlighting

^e

End of the sentence

M-w

Copy highlighted

M-}

End of the paragraph

^y

Paste copied text

^u

Repeat command with following number

M-f

Move to next word

^d

Delete a character

^xs

Save the document

^xc

Quit the document

## Learn by Exercise

---

[Line 1]: The five boxing wizards jump quickly.

[Line 2]: See the quick brown fox jump over the lazy dog.

[Line 3]: A mad boxer shot a quick, gloved jab to the jaw of his dizzy opponent.

[Line 4]: We promptly judged antique ivory buckles for the next prize.

[Line 5]: A quart jar of oil mixed with zinc oxide makes a very bright paint.

[Line 6]: The job requires extra pluck and zeal from every young wage earner.

- Complete all tasks with minimum number of retyping, but with commands
  1. Substitute all j's to z and all z's to j
  2. Copy lines 1, 3, 5, and 6, and make new paragraph with those lines
  3. Delete three words "requires extra pluck," and type in "need lot of money" in the place
  4. Add "caps" at the end of all words with "w", e.g., wizards to "wizardscaps"

# References

---

- Reference card with most commands you'll ever need
  - <http://home.uchicago.edu/~gan/file/emacs.pdf>
- Official GNU emacs site
  - <http://www.gnu.org/software/emacs/>
- An emacs HowTo
  - <http://jeremy.zawodny.com/emacs/emacs.html>
  - <http://www2.lib.uchicago.edu/keith/tcl-course/emacs-tutorial.html>

