Beginner Block

i	start insert
<esc>	escape to normal mode
:w	in normal mode, write to current file
:q	in normal mode, quit the editor

Description in BOLD means overwrite default configuration
[https://github.com/Piping/dotfiles] has my configuration

Notation: `<leader>mm` means `,mm`
`<C-a>` means `Ctrl + A`, `<S-x>` means `Shift + x`
`a/b` means press keyboard either a or b

Motions: command that moves the cursor

hjkl	left/down/up/right
f/F	search and move to next typed character
t/T	similar to `f/F`. but stop before character
0/$	start/end of current line
w/e	next start/end of word, `W/E` for word with punctuation
b/B	previous start of word `B` for word with punctuation
gg	go to line {count}, default first line
G	go to line {count}, default last line
-/<enter>	previous/next start of the line
H/M/L	cursor go to TOP/MIDDLE/BOTTOM of the screen
''	(single quote twice) previous location in jumplist
'{a-z}/`{a-z}	previous marked position using m{a-z}, E.g. ma `a
'.	To the position where the last change was made.
%	move to closing pair [], {}, ()
{1-9}+	a number type before motions, repeat {count} motion
:help motion	check more on documents inside vim

Vim Concept: Motions - command that moves the cursor, depends on current cursor position. Above list is not complete! It can be used with `OPERATOR` to efficiently editing text in Textual User Interface

Operator - editing the text efficiently

d	delete
c	change
y	copy/yank
~	reverse case
!	filter with external program, E.g. format doc
: h operator	more on help document inside vim

Two way to use operator (before selection, after selection):
1. <operator><motion> or <operator><text-object-motion> etc.
2. <selection from visual mode><operator>

Visual Mode Commands

o	switch between two ends of selection (anchor)
<c-v>	switch to visual block mode
V	switch to visual line mode
v	switch to visual mode
<leader>a	alignment by character type

Command Mode :

:%	{range}, equal to 1,$ (the entire file)
:! <external cmd>	range of text is being pipe to cmd to be replaced
:%! xxd	edit binary file
:%! xxd -r	save the file into binary form

Page Movement

J/K	**move buffer down/up and keep cursor position**
zz	bring current line and cursor to center
zb	move current line and cursor to bottom screen
zt	move current line and cursor to top screen

CSCOPE MAPPING (My Configuration)

:cs add <path to cscope.out> <path to workspace>

<leader>gs	search the C symbol under cursor
<leader>gg	search global defintion
<leader>gd	search funtions called by this function
<leader>gc	search who called this function
<leader>gt	search this string
<leader>ge	search this egrep pattern
<leader>gi	search for files include the filename under cursor
<leader>gii	search for files include the current file

Insert Mode (My configuration)

<C-a>	Jump to the beginning of the line
<C-e>	Jump to the end of the line
<C-w>	Backward-Delete Word
<C-d>	Forward-Delete Word
<C-z>	backward move cursor one word
<C-x>	forward move cursor one word
<C-u>	delete/cut current line
<C-y>	paste/yank to current line
<C-k>	delete the rest line after cursor

Normal Commands (My Configuration)

\\	**format and intent the file** E.g. clang-format
=	**go to end of line**
<leader>nn	open file browser panel
<leader>mm	open ctags symbol list panel
<leader><cr>	Disable highlights when you press
<leader>f	use fzf to fuzzy search recuently used files
<leader>q	close current window
<leader> w	save the file
<leader>u	open the undotree
<leader>zz	Distraction-Free Display

fuzzy-search-panel (My configuration)

<c-f>	select all
<c-g>	deselect all
tab/shift-tab	select-deselect current line
<C-t>	open file in new tab
<C-x>	split horizontaly
<C-v>	split vertically
<esc>/<c-d><c-c>	quit

Commentary (External Plugin Default)

gcc	comment/uncomment current line
10gcc	comment next 10 lines
gcu	comment block
gcap	comment the paragrah
gc	comment selection (visual mode)