vim

Basics

```
ESC ............ Enter normal mode (exiting current
mode/command)
:w .............................. Save (don't exit)
:w !sudo tee % ................ Save using sudo
:saveas file ........................... Save as
:q ................................... Quit
:q! ..................... Quit discarding changes
:wq | :x .....|.|{}............... Save and exit
n<command> ..|.|{}........ Repeat command n times
u ................................... Undo
C-r .................................. Redo
```

Movement

```
h | j | k | l  Move one char left | up | down | right
w | e .......... Jump forward to start | end of word
W | E ... Jump forward to start | end of word+punct
b .................. Jump backward to start of word
0 | $ ................. Jump to start | end of line
∧ | g_ ..... Jump to first | last non-blank char in line

% ........ Jump to matching char of pair '()' | '{}' | '[]'
} | { ... Jump to next | previous para/function/block

H | M | L . Move to top | middle | bottom of screen
C-f | C-b ............ Move down | up one screen
C-d | C-u ............ Move down | up half screen
zz ...................... Center cursor on screen
gg | G ......... Move to first | last line of document
nG ............................ Jump to line n
```

Enter/exit Insert mode

```
i | a .......... Start inserting, before | after cursor
I | A ............ Start inserting, start | end of line
o | O ..... Start inserting, below | above current line
ESC ........ Exit insert mode (enter normal mode)
```

In insert mode:

```
C-h | C-w | C-u .... Del backwards char|word|line
C-k .......................... Delete forwards line
C-t | C-d ............... Tab | de-tab current line
C-r 0 ....................... Paste last 'y'anked
C-r C-p 0 ............ Paste last 'y'anked, indenting
C-n | C-p .... Start auto-complete (C-n/C-p scrolls)
```

Editing

```
rx ............. Replace current (single) char with x
s .................... Delete char + insert mode
S | cc ................. Delete line + insert mode
cw .......... Delete to end-of-word + insert mode
c$ ........... Delete to end-of-line + insert mode

J ......................... Join line below to current
R ............. Start replacing (enter replace mode)
ESC ....... Exit replace mode (enter normal mode)
```

Cut/Copy/Paste

```
yy | Y ........................ Copy ('yank') a line
yw ................ Copy from char to end-of-word
y$ ................. Copy from char to end-of-line
yb ............. Copy before char to start-of-word
y0 ............. Copy before char to start-of-line

x ................................... Cut one char
dw ............................ Cut to end-of-word
d$ | D ........................ Cut to end-of-line
dd ................................. Cut entire line
db ............................ Cut to start-of-word
d0 .......................... Cut to start-of-line

p | P .......... Paste clipboard after | before cursor
xp ............................... Transpose chars
```

Visual Mode

```
v .................... Start visual mode, mark chars
C-v ........ Start visual BLOCK mode, mark block
gv .......................... Visual select last paste
```

In visual mode:

```
o ............... Jump to other end of marked area
O ........ Move horizintally to other corner of block
aw .................................. Mark word
ab | aB ................... Mark block with '()'|'{}'
ib | iB ............. Mark inner block with '()'|'{}'
< | > ........ Shift marked left|right by shiftwidth
y | d ......................... Copy | Cut marked
```

Search and Replace

```
/pattern .............. Search forwards for pattern
?pattern ............ Search backwards for pattern
n | N ...... Repeat search in same|reverse direction
:%s/old/new/g ..... Replace old with new (g)lobally
:noh ........... (Noh)ighlighting of search matches
# ............. Search backwards word under cursor
```

Macros

```
@x ................................ Run macro x
@@ .......................... Re-run last run macro
qx ................. Start recording macro named x
q .................................. Stop recording
```

Registers

```
"xp ................... Paste contents of register x
"x<yank-command> ........... Yank into register x
:reg ................... Show contents of registers
```

Register 0 contains the value of the last yank command.

vim

Windows

```
:split ................ Split window horizontally
:vsplit ................ Split window vertically
:split filename .. Split horizontally, load filename
:vsplit filename ..... Like :split but vertically
:close .................... Close current window
:only ............. Keep only this window open
C-h|j|k|l ...... Move to left|up|down|right window
C-w = ................ Make all windows equal size
n C-w + ........ Increase size of window by n lines
```

Buffers

```
:ls ............................... List all buffers
:b n ............... Open buffer n in this window
:sb n .......... Split, open buffer n in this window
```

Reformatting

```
V<movement>= ..... Visual select, reformat selected
== ......................... Reformat current line
gqap ...................... Reformat paragraph
:set textwidth  For reformat, margin to wrap lines
```

netrw

```
:Explore ......... Explore directory of current file
:Vexplore ....... Explore directory in vertical split
I ................................. Toggle banner
i .................... Cycle directory listing mode
gn ........ Go i(n)to the directory under the cursor
- ............................... (Go up a directory
<CR> ....... Expand/Collapse a directory, open a file
o|v .................... Open file in horiz|vert split
c ................. Make this the current directory
d ............................... Create directory
% .................................... Create file
<DEL>|D .................... Delete file/directory
R .......................... Rename file/directory
```

Org Mode

Show/Hide:

```
<TAB> .................... Cycle visibility, in subtree
zm ............. Collapse headings one level, globaly
zr ............. Expand headings one level, globally

} ................................... Next heading
]] ..................... Next heading, same level
g} ..................... Next heading, higher level
} .............................. Previous heading
[[ ................... Previous heading, same level
g{ ................. Previous heading, higher level
```

Headings:

```
<CR>|\hn ....... New heading, below, after children
\hN .............. New heading, above (Not below)
\hh  (hijack) New heading, below, taking over children

m]] ......................... Move subtree down
m[[ ............................ Move subtree up
>> .............................. Demote heading
<< .............................. Promote heading
>ar ............................. Demote subtree
<ar ............................. Promote subtree

dar .............................. Delete subtree
dah .............................. Delete heading
p ........................................ Paste
```

TODO lists:

```
S-left|S-right ...... Cycle keyword (for current
heading)
```

Checkboxes:

```
\cn ...................... Insert checkbox, below
\cN .......... Insert checkbox, above (Not below)
\cc ................................. Toggle status
```

Fugitive

% in commands refers to current file

```
:Git ........ Run arbitrary command on current file
:Git! ... Run arbitrary git command, capture output
:Gedit ................... Edit file/commit/tag/tree
:Gwrite ........................... Like git-add
:Gread ... git checkout – filename, but in buffer only
:Gstatus .......................... git status
:Gmove ................... Rename file and buffer
:Gcommit ...................... Commit changes
:Gdiff ..................... Interactive visual diff
```