Vim Cheat Sheet

Walk (normal mode)

← , l	– one char left
→ , h	– one char right
↑ , k	– one line up
↓ , j	– one line down
PgUp , Ctrl + B	– page up
PgDn , Ctrl + F	– page down
0	– to beginning of line
^	– to beginning of string
$	– to end of line
gg , :0	– to first line
G , :$	– to last line
:17	– to line 17 (use any number)
w / W	– to next word/big-word
e / E	– to end of word/big-word
b / B	– to prev word/big-word
(/)	– to next/prev sentence
{ / }	– to next/prev paragraph
%	– to next pair of brackets (try this)

Walk marks (normal mode)

ma	– set mark 'a' (use any char)
'a	– go to mark 'a'
'.	– go to last edit symbol
:delm a	– del mark 'a'
:delm!	– del all marks

Commands (normal mode)

u	– undo
Ctrl + R	– redo
.	– do last action again
x / X	– del char under/before cursor
~	– change char case under cursor
J	– merge two lines together
D	– del all to end of line
rX	– replace char under cursor by X
c_	– replace ...
d_	– delete ...
y_	– copy ...
dd	– del line
cc	– replace line
yy	– copy line
>>	– add tab at the beginning of line
<<	– del tab at the beginning of line

Here is the magic:

c3w	– replace next 3 words
40k	– go 40 lines up

Insert mode

i / a	– go edit before/after cursor
I / A	– go edit beginning/end of text
o / O	– add line under/upper and go edit
Ctrl + ^	– while insert, switch input lang!
Arrows	– move cursor
Del	– del char under cursor
Esc	– leave insert mode (back to normal)

Find (normal mode)

fG	– to G char in line (use any char)
/substring	– find substring in text
#	– find word under cursor
n / N	– find prev/next

Find and Replace (normal mode)

:[%]s/<what>/<with>/[<mod>] wher:

%	– replace in all lines
<mod>	– replacement modifiers

For example:

:s/foo/bar/	– replace 1st foo with bar in line
:s/foo/bar/g	– replace all foo with bar in line
:%s/foo/bar/g	– replace all foo with bar in all lines

Modifiers:

g	– search all entries in a line
c	– ask before replace
i	– case insensitive
l	– case sensitive replacement

Modifiers can be combined:

:%s/foo/bar/gic	– ignoring case replace all foo with bar in all lines and ask before replacement

Copy and Paste

First case is to use c_/d_/y_ commands and then paste data under cursor using p / P .
Second case is to use select mode:

1. v – go to select mode OR Ctrl + v – box select.
2. Use 'walk' keys to select a piece of a text.
3. Press y / d / c to copy/delete/replace that text. r and ~ also work but not copy data to buffer.
4. Use p / P to paste data from buffer.

Named Registers

" – named register's qualifier. Next letter after " specifies a name of register. For example:

"add	– remove curent line and put it to register 'a'
v3l"by	– copy 3 chars to register 'b'
"cp	– paste from register 'c'

Special named registers:

:, ., %, #	– read-only registers
=	– expression register
*, +, ~	– selection and drop registers
/	– last search pattern register

Macros (normal mode)

Macros – a sequence of actions that can be repeated.

1. qx – start record macro with name 'x'. Any char can be used.
2. Do something your need to repeat (be carefull).
3. Press q to stop recording.
4. Call macro:
   - @x – call macro 'x' one time
   - 10@x – call macro 'x' ten times

Session (normal mode)

:q	– quit
:!q	– force quit
:w/:w filename	– save/save to file 'filename'
:x	– save and quit
:mks filename	– save session to file 'filename'. Use vim -S filename to restore a session.
:e ++enc=ENC	– set current buffer encoding to ENC

Windows and Tabs

:sp filename	– split current window horizontally by two and show file 'filename' in second window
:vs filename	– same as previous but vertically
Ctrl + w and Arrows	– move cursor from window to window
:tabnew file	– open file in new tab
gt/gT	– go to next/prev tab
:windo ...	– exec command in all windows
:tabdo ...	– exec command in all tabs

Spelling

:set spell/:set nospell	– turn on/off spelling marks
:set spell spl=ru	– turn on spelling using 'ru' lang
z=	– open spell menu for a word under cursor

Folding

zo/zc	– open/close fold
zR/zM	– open/close all folds

Variables:

:set foldlevel=...	– integer value
:set foldmethod=...	– syntax, indent or manual

For manual foldmethod:

1. v – select a part of text for folding.
2. zf – mark selected text as folding.

Useful variables

:set var	– set boolean to True OR print current value
:set novar	– set boolean var to False
:set var=foo	– set value to foo

Variables:

hls	– boolean, search highlight on/off
ts	– integer, number of spaces to show TAB
ai	– boolean, copy current indent for new line
paste	– boolean, fixes ugly insert into vim-in-terminal
nu	– boolean, line numbers on/off
wrap	– boolean, wrap long lines on/off
ss	– integer, number of columns to scroll horizontally