Auto Complete
Omni-Complete
CTRL-N

UNDO
:later 2h
:undolist
:earlier 2h
g- (undo)
g+ (redo)
u (undo last)
C-r (redo last)

Marks
m<letter> sets mark named <letter>
current location
`<letter> jumps precisely to that mark
'<letter> jumps to the line with the m
lowercase letter: mark is local to the buff
uppercase letter: mark is global
:marks shows you your current marks
c`a - change text from cursor to mark a
d`a - delete text from cursr to mark a
='a - reformat lines from current one to the one with mark a
`` -> toggle back and forth between 2 spots
'. and `. -> jump to the line or exact location of the last modifcation

copy/delete/paste
p -> paste after P -> paste before
]p ->paste after with current tabstop
]P ->paste before with current tabstop
yi” -> yank all inside of “ “
di] -> delete all inside of []
ci) -> cut all inside of ()
:m5 -> Move current line to line 5
gP -> Paste after retaining left margin formatting
gpv -> Select just pasted area and then go into visual mode

Registers
"wyy -> yank current line into register w
"Wyy -> append to register w using a capital "W"
"wp -> paste register w
:reg -> see what is in register
q<char> start recording in

Command Mode
e or w -> move to next word (right)
b -> move to previous word (left)
100j -> down 100 lines
20k -> up 20 lines
G -> End of file :1 or gg -> Top of file
:955 -> Go to line 55
25% -> Go to the 1/4 part of the page
^-f Down 1 screen
^-b -> Up 1 screen
^-d -> Down 1/2 screen
^-u -> Up 1/2 screen
^-e -> Scroll up w/out moving cursor
^-y -> Scroll down w/out moving cursor
0 -> beginning of line
$ -> End of line
(-> go back a sentence
) -> forward a sentence
. -> repeat

zt -> cursor stay: scroll screen so cursor falls @ top
zz -> "" @ middle
zb -> "" @ bottom
zL -> scroll half page right
zH -> scroll half page left
H -> cursor to highest part of the screen
M -> "" middle ""
L -> "" lowest ""
g; -> jump back to last position
gv -> reblock las visual block
gj -> down 1 line in word wrapped block
gk -> up 1 line in word wrapped block
g0 -> home for a word wrapped block
g$ -> end for a word wrapped block
ci< -> delete all inside of bracket ie <ok id-"ok">

Searching
?<text> -> search backwards
* -> find word underneath the cursor
:set hlsearch
:noh or :nohlsearch
Search within the line:
f/F<char> jumps to the first <char> to the right/left and places cursor on it
t/T<char> jumps does the same, but stops one character short of it
Next two are more in the spirit

Insert Mode
a -> Insert after the current character.
A -> Insert at the end of the line.
I -> Insert at the beginning of the line.
O -> empty line above the current one.
s -> erase current char
S -> erase current line
r -> replace one char and go to command mode
R -> replace mode
C -> erase from cursor to end put in Insert mode
D -> erase from cursor to end leave in command
Y -> short for yy
C-x C-l -> autocomplete line
C-x C-f -> autocomplete filesystem
----
gi -> go back to where you last where and go into insert mode

^-w -> delete word before the cursor
^-u -> delete line before cursor - good for joining
^-r (plus register) -> insert register
^-r ^-o (plus register) -> insert register and don't autoindent
^-r ^-p (plus register) -> insert reg. and fix indent
^-t -> insert shiftwidth at the beginning
^-d -> delete indent at the beginning
0 ^-d -> del all indents at the beginnig
^ ^-d -> del add indents and return to next ln.
^-e -> copy char below cursor here
^-y -> copy char above cursor here
^-x ^-e -> scroll up
^-x ^-y -> scroll down

Replace examples
:set noic -> case sensitive
:set ic -> not case sensitive
:%s/'//g -> delete all single quotes
:%s/, :force => true//g -> del all [, force => true]
:%s/\r$//e | set nomod -> gets rid of ^M
:%s/pels/pelz/g -> replace all pels with pelz
:%s/(\d\+)\s/863-/g -> repl all

Windows
^wn -> New Window
^ws -> split
^wv -> split vertically
----
^wc -> close
^wo -> close all except the active one

Moving between windows
^wj mv down ... ^wk mv up ...
^wh mv left ... ^wl mv right ..
^wR rotate up ^wr rotate down

Resizing Windows
^w 10+ (or 10) Resize horizontal split 10
^w 10> (or 10<) Resize vertical split 10
^w_ max height ^w1_ min height
^w| max width ^w1| min width
^w= equal heights
----
^wR -> swap splits
^wT -> break up win to tab
^wo -> close out all in tabview but this win

Buffers
:bn next buffer :bp previous
:b main -> goto buffer with main in the name
:sb main -> split buffer : ditto above
:sbm or :bm -> goto modified buffer
:sp %:h/<filename> -> %:h is current dir of the file ur currently editing

Tabs
:tabe <file>
:tabc -> close
:tabn or :tabp (or gt, gT (for prev.))
:tabs -> list of all tabs and view ports
:tabm n -> n i.e. 4 is 4th tab order

Opening files
:sav -> save as
:sav %:h/new.html.erb
:e -> open
:sp -> open split
:vsp -> open split vertically
:rightbelow vnew spec/factories.rb -> vert. split to the right

register q again 2 stop
@<char> to play the macro
@@ to repeat last macro

Misc
:set syn=ruby -> set syntx
:se iskeyword+=46
viw -> to select
mail.body.encoded
:color <tab> -> change colors
gqap -> reformat a para to textwidth
gq -> reformat selected block

Ctags
ctags -R .
ctags -R -f ./.git/tags .
<C-]> follow tag under cursor
<C-t> climb back up tag tree

:tag <e.g. function_name>
:ta <e.g. function_name>
:ts or :tselect shows the list
:tn or :tnext goes to the next tag in that list
:tp or :tprev goes to the previous tag in that list
:tf or :tfirst goes to the first tag of the list
:tl or :tlast goes to the last tag of the list

Folding
Example: Folding XML within a Rails log.
First:
set foldmethod=indent
then:
zo -> to open 1 level
zO -> open all levels
zc -> close 1 level
zC -> close all levels

of Insert mode
df; - delete text from cursor to the first ; to
the right
cT$ - change text from cursor up to the first
$ to the left and go into Insert Mode

Often you want to find other instances of word under the cursor
*/# -> find next/previous instance of whole word
g*/g# -> find next/previous instance of partial word or find lines with a certain word.
[I and]I -> list lines with word under the cursor
/clock\C -> case sensitive search

Visual Mode
< : shiftwidth to left
> : shiftwidth to right
3< : 3 shiftwidth to left
3> : 3 shiftwidth to right
// comment out a block
^-v then select ur block then type "I" and "#" and then ESC
If you need to replace the selection use 's' to delete and then type the replacement.
vi" -> select all inside of " "
vi} -> select all inside of {}
vi) -> select all inside of ()
g Cntl-G -> byte char count
Vap -> select current paragraph
V2ap -> select current and next paragraph}V-2ap -> go to the end of the current and then select it and the preceding paragraph
Vap[d,c,y] -> del,cut,yank current paragraph
'<,'>s/\%V_/ /g -> sub _ for spaces for visually selected block

Diff
vimdiff <file1> <file2>
]c -> move to next difference
[c -> to prev. difference
// Note dp and dg are Not preceded with a :
dp -> put to diff. to other side
dg -> get the diff from the other side
^W l -> switch to other window
^W h -> switch to left window

(234)[space] with 863-
:%s/\s\+$//e -> Get rid of trailing white space
:%s/\$/\$j/g -> replace all $ with $j
:1,9 s/a\./employer\. -> replace only first instance in lines 1 thru 9
:150,170 s/</\&lt;/g -> remove certain lines '<' with html entity
% -> shortcut for rainge of 1,$ or the entire document
%S/Sir/Mr/g -> case sensitive
Substitute in visual mode? See that section

Spelling:
:set spell
:set nospell
z= -> list suggestions for current word cursor is on
zg -> add good word in spell file
zG -> add word to internal list
]s -> next spelling mistake
[s -> prev spelling mistake

Undo and Redo:
:earlier 5m -> go back 5m
:later 5s -> redo 5 seconds
:undolist -> undo list
:undo *n*

Auto-indenting
:set nocin or nocindent
:set noautoindent
:set nosmartindent
:set paste | :set nopaste

Matching
Use built-in matchit plugin which is '%'

:vertical resize -20 -> resize
:Sex -> open split window explorer
:lcd %:p:h -> change dir to current file path
:!echo % -> display current filename
:sp %:h/_subnav -> open another file in the same directory as the file you are currently editting.
Commands
:set ai // set autoindent
:mksession <name of file> // save a work session
:source <name of file> //
:so ~/.vimrc // reload vimrc
restore session
vim -S <name of file>
:set number // line number
:set ignorecase // ignore case in search
:set autoindent
:set hlsearch
:set nohlsearch
:set ea // will toggle expandalways to off
:set ea ? // see the current value of

Plugins

Surround.vim
yat -> yank entire tag contents
yit -> yank inside tag contents
[instead of 'y', 'x', or 'c', or 'v' can be used]
cs"' -> Turn double quotes to single
dst -> Only delete the tag not content
csw" -> double quotes around World
cs({ -> Put cursor right on the '(' e.g. expect**(** ….()).to
yi" -> yank contents inside of quotes
To use surround in visual mode, highlight the block and hit 'S' followed by char.

Text Object Plugins
prefix = v|c|d
(prefix)ar, (prefix)ir for Ruby text object
(prefix)ai, (prefix)ii for Indent text object
(prefix)aj, (prefix)ij for {} () [] Brace text object