Modes

Vim offers multiple modes, including Normal, Insert, Command and Visual.
Normal mode is the mode Vim starts in and which is used for navigation and text manipulation. You can return to normal mode by pressing ESC.
Insert mode is used for inserting text and can be accessed several ways, the most common being "i" to Insert, "a" to Append, "c" to change.
Command mode is to enter commands such as searching, reading files, saving and exiting. It is accessed by ":" key.
Visual mode like normal mode is used for navigation and text manipulation, there are additional sub-modes - visual, visual-line and visual-block modes. Visual mode is accessed by pressing a variety of "v" key combinations.
In the various modes most commands execute immediately, the exception to this is Command mode where you need to press the Enter key to execute.

Insert Mode

Key	Action
`i`	insert at cursor
`I`	insert at the beginning of the line
`a`	append after the cursor
`A`	append at the end of the line
`o`	open blank line below current line (no need to press return)
`O`	open blank line above current line
`ESC`	to exit insert mode (and return to normal mode)

Cursor movements can be combined with insert mode commands
* `Bi` - insert at the start of the current word
* `ea` - append at the end of the current word
* `Wi` - insert at the start of next word
* `jI` - insert at the start of next line
* `jA` - append at the end of next line

Command Mode

Key	Action
`:`	enter command mode
`Enter`	to action command
`Esc` x2	to exit command mode (and return to normal mode)

The command is only executed after pressing the `Enter` key.

Visual Mode

Key	Action
`v`	enter visual mode
`V`	enter visual-line mode
`Ctrl-v`	enter visual-block mode
`Esc` x2	to exit visual mode (and return to normal mode)

Examples of usage, after making visual selection perform the following actions:
visual-line mode
* delete complete lines - `d`
* yank (copy) - `y`
visual-block mode
* Insert text to start of multiple lines - `I`, <text,whitespace>, `Esc`, `Esc`
* Delete characters from the start of multiple lines - `d`

Cursor Movement

Key	Action
`h`	move left (also left cursor)
`j`	move down (also down cursor)
`k`	move up (also up cursor)
`l`	move right (also right cursor)
`w`	jump forward to start of words (punctuation considered words)
`W`	jump forward to start of words (spaces separate words)
`e`	jump forward to end of words (punctuation considered words)
`E`	jump forward to end of words (spaces separate words)

Cursor Movement (cont)

`b`	jump backward to start of words (punctuation considered words)
`B`	jump backward to start of words (spaces separate words)
`0`	jump to start of line (zero)
`^`	jump to first non-blank character of line
`$`	jump to end of line
`G`	go to end of file
`gg`	go to start of file
`H`	move to top of screen (no screen movement)
`M`	move to middle of screen (no screen movement)
`L`	move to bottom of screen (no screen movement)

Prefix a movement command with a number to repeat the move that number of times.

Editing

Key	Action
`x`	delete character from right
`X`	delete character to left
`r`	replace a single character (does not use insert mode)
`R`	replace multiple characters
change c+option	start change command (will enter Insert mode)
`cw`	change to the end of word
`cc`	change an entire line
`c$` or `C`	change to the end of line
`cG`	change to end of file
delete d+option	start delete command (will stay in Normal mode)
`dw`	delete word
`dd`	delete entire line

Editing (cont)

`d$` or `D`	delete to the end of the line
`dG`	delete to the end of the file
`s`	delete character at cursor and substitute text (enters Insert mode)
`S`	delete line at cursor and substitute text (same as cc)
`xp`	transpose two letters (technically a combo to delete and paste)
`u`	undo
`Crtl-r`	redo
`.`	repeat last command
`J`	join current line with line below

Commands seem similar but there are slight differences. Some command will execute and can be repeated by pressing the same key. Other commands will perform the action and then automatically switch to Insert mode. Also, a command can be performed in multiple modes with slightly different keystrokes.
Commands are combined with cursor movement commands, so it is possible to:
* `dgg` - delete to beginning for file
* `c5w` - change 5 words
* `4dj` - delete current and 4 other lines

Cut, Copy, Paste

Key	Action
delete	delete, i.e. cut (see previous)
yank (y+option)	yank, i.e. copy
`yw`	yank word
`yy`	yank line
`2yy`	yank 2 lines
`yj`	yank current and next line
put (p+option)	put, i.e. paste
`p`	put clipboard contents after cursor
`P`	put clipboard contents before cursor
`2p`	paste 2 copies

Vim Editor [Yet Another] Cheat Sheet

by PeterCeeAU via cheatography.com/58333/cs/15442/

Search and Replace

Key	Action
`/pattern`	search for *pattern* forward from cursor
`?pattern`	search for *pattern* backward from cursor
`n`	repeat search in same direction
`N`	repeat search in opposite direction
`:%s/old/new/g`	replace all *old* with *new* throughout file
`:%s/old/new/gc`	replace all *old* with *new* throughout file with confirmations
`:1,.%s/old/new/g`	search from start of file to current cursor position, replace all *old* with *new*
`:.,$%s/old/new/g`	search from current cursor position to end of file, replace all *old* with *new*

Open, Save and Exit

Key	Action
`:w`	write (save) the file, but don't exit
`:w file`	write to new *file*
`:1,5w file`	write lines 1-5 to new *file*
`:wq` or `ZZ`	write (save) and quit
`:q`	quit (fails if anything has changed)
`:q!`	quit and discard changes
`:r file`	read in *file* to current cursor position
`:r !head -n5 file`	read in the first 5 lines of *file* to current cursor position
`:e file`	edit *file* in new buffer
`:sp file`	edit *file* in new buffer and split window
`ctrl-ww`	switch between windows
`!! command`	perform shell command, output will overwrite current line

Marks

`:marks`	list of marks - includes mark name, line, column and text
`ma`	set current position for mark 'a'
`'a`	jump to position of mark 'a'
`` `a ``	jump start of line of mark 'a'
`y'a`	yank from current position to mark 'a'

Can use a variety of commands from current position to position of mark 'a'. Most often use this command to copy and paste multiple lines of text. Similar results can be achieved in Visual mode.

Ranges

`:.`	perform action on current line
`:1`	perform action on line 1
`:1,5`	perform action on lines 1-5
`:$`	perform action on last line
`:%`	perform action on all lines
`:5,$`	perform action on lines 5 to last
`:.,$`	perform action on current to last line
`:.+1,$`	perform action on line below current to last
`:.+1,.+5`	perform action on line below to 5 lines below current line
`:'a,'b`	peform action from mark `a` to mark `b`

Ranges are combined on the Command line to perform a task on the lines specified. The default range, for most commands, is the current line. While for other commands the default range is all lines.

`:1,5d` - delete lines 1-5

`:5,.y` - yank lines 5 to current

`:.,$s/old/new/g` - replace *old* with *new* from current line to end of file

`'a,'by` - yank from 'a' to 'b'

Miscellaneous

`:set number`	display line numbers
`:set number!` or `:set nonumber`	turn off line numbers
`:colorscheme <colorscheme>`	change colorscheme
`:version`	show version and other information
`:set list`	show hidden characters
`:set nolist`	do not show hidden characters
`Ctrl-V`, cursor highlight, `I`, `#`, `Esc`, `Esc`	multi-line comment
`Ctrl-V`, cursor highlight, `x`	delete multi-line comment
`:nohl`	remove search highlight
`:%s/^M//g`	replace DOS carriage returns (if `dos2unix` is not installed)

Commands can be abbreviated, e.g. `:nohl` is shorthand for `:nohl-search`

Configuration

Global defaults can be configured in `/etc/vimrc` but setting personal defaults is preferred.
Shared colour scheme files are located in `/usr/share/vim/vim`*`XX`*`/colors`
Personal defaults can be configured in `~/.vimrc`
Personal colour scheme files added to `~/.vim/colors`
Specifiy default colour scheme in `~/.vimrc`
`colorscheme <colorscheme>`

Useful Links

https://en.wikibooks.org/wiki/Learning_the_vi_Editor/Vim/Modes
http://vim.wikia.com/wiki/Vim_Tips_Wiki
http://vim.wikia.com/wiki/Search_and_replace
http://vimcolors.com/
https://www.howtoforge.com/vim-basics