

<p>find</p> <p>"find": "f", "bind": "F", "(until till)": "t", "bill": "T",</p> <p>search</p> <p>"search": "slash", "birch": "question",</p> <p>letter</p> <p>'langle': 'langle', 'lace': 'lbrace', 'lack': 'lbracket', 'laip': 'lparen', 'rangle': 'rangle', 'race': 'rbrace', 'rack': 'rbracket', 'raip': 'rparen',</p> <p>modifier</p> <p>'(in inside inner)': 'i', '(a around outer)': 'a',</p> <p>verb</p> <p>"delete": "d", "(copy yank)": "y", "(select visual)": "v", "case lower": "g,u", "case upper": "g,U", "format": "g,q", "comment": "g,c", "(indent reindent)": "equal", "(flow reflow)": "g,q", "shift left": "langle", "shift right": "rangle",</p>	<p>object</p> <p>'word': 'w', 'big word': 'W', 'sentence': 's', '(paragraph pare)': 'p', 'block': 'b', '(paren laip)': 'rparen', '(brackets rack)': 'rbracket', '(brace race)': 'rbrace', 'quote': 'dquote', '(post troth)': 'apostrophe',</p> <p>simple_verb</p> <p>"case (swap toggle)": "tilde", "X.": "x",</p> <p>"Pete macro": "at,at", "Pete": "dot",</p> <p>"join": "J",</p> <p>"paste up": "P", "paste down": "p",</p> <p>"(increment increase)": "c-a", "(decrement decrease)": "c-x",</p> <p>"undo": "u", "redo": "c-r",</p>	<p>surround</p> <p>"tag": "t", 'ampersand': 'ampersand', '(post apostrophe)': 'apostrophe', '(starling asterisk)': 'asterisk', 'backtick': 'backtick', '(bar pipe)': 'bar', 'dollar': 'dollar', 'quote': 'dquote', 'hashtag': 'hash', 'modulo': 'percent', 'single quote': 'squote', '(tilde strike)': 'tilde', '(underscore score)': 'underscore',</p> <p>'langle': 'langle', 'lace': 'lbrace', 'lack': 'lbracket', 'laip': 'lparen',</p> <p>'rangle': 'rangle', '(race brace curly)': 'rbrace', '(rack bracket)': 'rbracket', '(raip paren)': 'rparen',</p>	<p>motion</p> <p>"up": "k", "down": "j", "left": "h", "right": "l", "word": "w", "big word": "W", "board": "b", "word end": "e", "big end [word]": "E", "sent up": "lparen", "sent down": "rparen", "pare up": "lbrace", "pare down": "rbrace", "next": "n", "pecks": "N",</p> <p>uncountableMotion</p> <p>"start": "0", "front": "caret", "(end rest)": "dollar", "match": "percent", "top": "g,g", "bottom": "G",</p>
--	--	--	--

ActionRule

```
"<verb> <motion_rule>": Key("%(verb)s") + execute_rule('motion'),
"<verb> <object_rule>": Key("%(verb)s") + execute_rule('object'),
"[<n>] <simple_verb>": Key("%(n)s, %(simple_verb)s"),
"[<n>] <verb> line": Key("%(n)s, %(verb)s:2"),

"replace spell <letter_sequence_rule>": Key("R") +
execute_rule('letter_sequence') + Key("escape"),
"[<n>] replace <letter>": Key("%(n)s, r, %(letter)s"),
"record <letter>": Key("q, %(letter)s"),

"select line": Key("V"),
"select last": Key("g, v"),
"visual other": Key("o"), # switch to other end of visual selection
"(visual | select) block": Key("c-v"),
"duplicate line": Key("y, y, p"),
```

BufferRule

```
"file save": Key("colon, w, exclamation, enter"),
"file save all": Key("colon, w, a, exclamation, enter"),
"file quit": Key("colon, q, exclamation, enter"),
"file quit all": Key("colon, q, a, exclamation, enter"),
"file done": Key("colon, x, exclamation, enter"),
"file done all": Key("colon, x, a, exclamation, enter"),
"file reload": Key("colon, e, exclamation, enter"),
"file open [<text>]": Key("colon, e") + Text("<text>"),
```

CommandModeStartRule

DiffRule

GeneralRule

```
"cancel": Key("escape, u"),
"abort": Key("c-c"),
"hide light": Text(':nohlsearch\n'),
"local directory": Text('lcd %:h'),
```

MarkMotionRule

```
"go <letter>": Key('apostrophe, %(letter)s'),
"go line <letter>": Key('backtick, %(letter)s'),
```

MarkOpRule

```
"mark <letter>": Key("m, %(letter)s"),
"show marks": Text(":marks") + Key("enter"),
"clear mark <letter>": Text(':delmark %(letter)s'),
```

MotionRule

```
exported = True
"[<n>] <motion>": Key("%(n)d, %(motion)s"),
"<quick_search_rule>": execute_rule('quick_search'),
"<mark_motion_rule>": execute_rule('mark'),
"go <uncountableMotion>": Key("%(uncountableMotion)s"),
```

ObjectRule

```
"[<n>] <modifier> <object>": Key("%(n)d, %(modifier)s, %(object)s")
```

QuickReplaceRule

```
"strip <haystack>": Text(":%s/%(haystack)s/c") + Key('enter'),
"replace <replace>": Text(":%s//%(replace)s/c") + Key('enter'),
"sub <haystack> shark <replace>": Text(":%s/%(haystack)s/%(replace)s/c") + Key('enter'),
```

For the confirmation dialog (due to the C flag)

```
"sub (this | it | yes)": Key("y"),
"sub (no | not)": Key("y"),
"sub all": Key("a"),
"sub last": Key("l"),
"sub quit": Key("q"),
```

QuickSearchRule

```
"[<n>] <search> <text>": Key("%(n)d, %(search)s") + Text('%(text)s') +
Key('enter'),
"[<n>] <find> <letter>": Key("%(n)d, %(find)s") + Key('%(letter)s'),
```

InsertModeStartRule

```
exported = True
"change <motion_rule> [<text>]": Key("c") + execute_rule('motion') +
Text('%(text)s'),
"change <object_rule> [<text>]": Key("c") + execute_rule('object') +
Text('%(text)s'),
"change line [<text>]": Key("c,c") + Text('%(text)s'),

"insert [<text>]": Key("i") + Text('%(text)s'),
"prepend [<text>]": Key("I") + Text('%(text)s'),
"after [<text>]": Key("a") + Text('%(text)s'),
"append [<text>]": Key("A") + Text('%(text)s'),
"oh [<text>]": Key("o") + Text('%(text)s'),
"bo [<text>]": Key("O") + Text('%(text)s'),

"insert last [<text>]": Key("g, i") + Text('%(text)s'),
```

InsertModeFinishRule

```
"okay | kay | abort": Key("escape"),
"cancel | oops": Key("escape, u"),
```

InsertModeCommands

```
# Insertion
"<text>": Key('c-g, u') + Text("%(text)s"), # create undo point
"comp line": Key("c-x, c-l"),
"comp file": Key("c-x, c-f"),
"shark paste <letter>": Key("c-r, %(letter)s"),
"shark paste literal <letter>": Key("c-r:2, %(letter)s"),
"shark paste sys": Key("c-r, asterisk"),
"shark paste file": Key("c-r, percent"),
"shark paste search": Key("c-r, slash"),
"shark shift left": Key("c-t"),
"shark shift right": Key("c-d"),

# Deletion
"scratch [last]": Key("escape, u, i"), # undo last utterance & re-enter insert
mode
"[<n>] scratch word": Key("c-w:%(n)d"),
"[<n>] scratch line": Key("c-u:%(n)d"),
"[<n>] scratch back": Key("c-h:%(n)d"),

# Motion
"[<n>] up": Key("up:%(n)d"),
"[<n>] down": Key("down:%(n)d"),
"[<n>] left": Key("left:%(n)d"),
"[<n>] right": Key("right:%(n)d"),
"go end": Key("c-e"),
"go start": Key("escape, l"), # avoiding c-a because of tmux conflict

"breathe <letter>": Key("space, %(letter)s, space"),
"assign": Key("space, equal, space"),
```

NavigationRule

"go old | gist up": Key("c-o"),
"go new | gist down": Key("c-i"),

"go old file | gist file up": Key("c-o:2/150, enter"),
"go new file | gist file down": Key("c-i:2/150, enter"),

"go alternate": Key("colon, e, hash, enter"),

"go file": Key("g, f"),
"go tag": Key("c-rbracket"),
"go tag back": Key("c-t"),

Cursor:

"cursor top": Key("s-h"),
"cursor middle": Key("s-m"),
"cursor (low | bottom)": Key("s-l"),

Windows/viewports:

"win west": Key(c.system.windowSwitchPrefix + "h"),
"win east": Key(c.system.windowSwitchPrefix + "l"),
"win north": Key(c.system.windowSwitchPrefix + "k"),
"win south": Key(c.system.windowSwitchPrefix + "j"),
"win switch": Key("c-w, c-w"),

"win split": Key("c-w, s"),
"win (vault | fault)": Key("c-w, v"),

"win equal": Key("c-w, equal"),

Tabs:

"[<n>] tab right": Key("g, t"),
"[<n>] tab left": Key("g, T"),

New:

"new tab": Key("colon") + Text("tabnew") + Key("enter"),
"new split": Key("colon") + Text("split") + Key("enter"),
"new fault": Key("colon") + Text("vsplit") + Key("enter"),

QuickSettingsRule

RegisterSaveRule

"to <letter>": Key('dquote, %(letter)s')

RegisterLoadRule

"from <letter>": Key('dquote, %(letter)s')

NetrwRule

exported = True

"explore here": Text(":"Explore") + Key("enter"),
"explore fault": Text(":"Vexplore") + Key("enter"),
"explore split": Text(":"Sexplore") + Key("enter"),
"explore tab": Text(":"Texplore") + Key("enter"),
"explore back": Text(":"Rexplore") + Key("enter"),

SnipMateRule

"snip <text>": Text("%(text)s") + Key("tab"),
"snip alias <alias>": Text("%(alias)s") + Key("tab"),
"snip letters <letter_sequence>": ...

extras = [

Choice("alias", {
Whatever snippets become useful but hard to say?
"fixture": "fix",
"method": "defs",
"function": "def",
"class": "cl",
"while loop": "wh",
"for loop": "for",
"mark code block": "codeblock",

