# VIM QUICK REFERENCE CARD

Basic movement
$\begin{array}{llllllllllllllllllllllllllllllllllll$
beginning, middle of line ^
\$ first, last character of line
$n$ G $n$ gg line $n$ , default the last, first $n$ % percentage $n$ of the file $(n \ must \ be \ provided)$
$n \dots$ column $n$ of current line%match of next brace, bracket, comment,
#define $n H n L \dots$ line $n$ from start, bottom of window M middle
line of window Insertion $\mathfrak G$ replace $\to insert$ mode
insertion $\mathfrak{G}$ replace $\rightarrow$ insert mode i a insert before, after
cursor I A insert at beginning,
end of line gI insert text
in first column a new line below, above the current line
rc replace character under cursor with
$c \ \mathtt{gr} c \ldots \ldots$ like $\mathtt{r},$ but without affecting
layout $R \ldots \ldots$ replace characters starting at the
cursor gR like R, but without
affecting layout $cm \dots change text of$
movement command $m$ cc $^{or}$
Schange current line
${\tt C}$
insert ~ switch case and
advance cursor $g^m \dots \dots \dots$ switch case of
movement command $m$ gu $m$ gu $m$ lowercase,
uppercase text of movement $m < m > m \dots \dots$ shift
left, right text of movement $m$ $n <<$
$n >> \dots \dots$
${\tt x} \; {\tt X} \ldots \ldots$ delete character under, before cursor
$dm \dots delete$ text of movement command
$m \text{ dd D} \dots$ delete current line, to the end of line $J gJ \dots$ join current line with next, without
space $: rd \leftarrow$ delete
range $r$ lines : $rdx \leftarrow$ · delete range $r$ lines
into register $x$

## Insert mode

${\bf \hat{v}}c$ ${\bf \hat{v}}n$ insert char $c$ literally, decimal value $n$
${\bf \hat{A}}$ insert previously inserted
text ^@same as ^A and stop insert $\rightarrow$ command
mode ^R $x$ ^R^R $x$ insert content of register $x$ ,
literally ${\bf ^{}P}$ text completion before,
after cursor ${\bf \hat{W}}$ delete
word before cursor ${\bf \hat{U}}$ delete all inserted
character in current line ^D ^T $\dots \dots \dots \dots$ .
shift left, right one shift width ${\rm \hat{k}}c_1c_2$ or
$c_1 \leftarrow c_2 \ldots \ldots$ enter digraph $\{c_1, c_2\}$
$\hat{\ }$ 0 $c$ execute $c$ in temporary command
$\bmod e \ ^X^E \ ^X^Y \dots \dots$
up, down $\langle esc \rangle$ or ^[ abandon edition $\rightarrow$
$ \begin{array}{c} \text{command mode} \\ \textit{Copying} \end{array} $

### Advanced insertion

### $Visual\ mode$

v V ^V ... start/stop highlighting characters, lines, block o ... exchange cursor position with start of highlighting gv ....... start highlighting on previous visual are aw as ap ...... select a word, a sentence, a paragrap ab aB ...... select a block ( ), a block { }

# Undoing, repeating & registers

u U undo last command, restore last changed
line . ${\bf \hat{R}}$ repeat last changes, redo last
$\mathbf{n}$ ndo repeat last changes with count replaced by $n$
$qc \ qC \dots$ record, append typed characters in register $c$
$\mathtt{q}\ldots\ldots\ldots\ldots \mathtt{stop}$
recording $@c \dots \dots$ execute the content
of register $c$ @0 repeat
previous @ command : @ $c \leftarrow$ ^ execute register
$c$ as an $Ex$ command $: rg/p/c \leftarrow$ $^{\circ}$ execute $Ex$
command $c$ on range $r \mid$ where pattern $p$ matches

# Complex movement

- + line up, down on first non-blank character
BWspace-separated word left,
right gE E end of space-separated word left,
right $n \dots \dots$ down $n-1$ line on first non-blank
character g0 beginning
of screen line g^g\$ first, last
character of screen line gk
gj screen line up, down
$fc Fc \dots \dots$ next, previous occurence of character $c$
t $c$ T $c$ before next, previous occurence of
c ; , repeat last fFtT, in opposite
direction [[]] start of section
backward, forward [] ] [ end of section
backward, forward [(])unclosed
(, ) backward, forward [{]}unclosed
$\{,\}$ backward, forward $[m]m$ start of
backward, forward Java method [#]#.unclosed #if,
<pre>#else, #endif backward, forward [*]*</pre>
start, end of /* */ backward, forward Search & substitution
$/s \leftarrow ?s \leftarrow $ search forward, backward for

# patterns into an atom\& $n \dots$ the whole matched

Tags
ta $t\leftarrow$ '
Scrolling & multi-windowing  ^E ^Y
Ex cômmands $(\leftarrow)$ : e f edit file f, unless changes have been made :e! f edit file f always (by default reload current): wn: wn write file and edit next, previous one: n: N edit next, previous file in list: rw write range r to current file: rw f

Ex ranges	
, ; separates two lines numbers, so $n = 1 - 1$	$ab solute \ line$
number $n$ . \$ the current	
line in file% *	
file, visual	area
$t \dots \dots$	line where $p$
matches+ $n - n + \dots + n, -n$ to the p	preceding line
$\begin{array}{c} \text{number} \\ Folding \end{array}$	
zfm create fold	
m:rfocreate f	
$r \text{ zd } zE \dots \dots$ delete fold at $c$	
window zo zc zo zcopen, cl	
recursively [z ]zmove to	
current open fold zj zk move	down, up to
start, end of next fold	
Miscellaneous	
$:$ sh $\leftarrow$ ' $:$ ! $c$ $\leftarrow$ ' $\dots$ start shell, execute comm	
Klookup keyword unde	
man :make←' start make, read errors	
first $: cn \leftarrow : cp \leftarrow : cp \leftarrow : display the n$	
error :cl←' :cf←' list all errors	
from file ^L ^G redraw screen, show	
position g^G show cursor column, line, a	and character

This card may be freely distributed under the terms of the GNU general public licence — Copyright c 2003 by Laurent Gre goire (laurent.gregoire@icam.fr) — v1.7 — The author assumes no responsibility for any errors on this card. The latest corrigin can be found at http://tporpul.origing.gr.fr/

under cursor gf.....open file which filename is