

VIM VIM VIM

НИКОЛАЙ СЛАДКИЙ
MULTIKEY.STUDIO

ДИСКЛЕЙМЕР

Все совпадения с реальными людьми - случайные. Это всё моя больная фантазия и немного юмора (kek). Я не считаю себя каким-то гуру, никого ничему не учу, не хочу никого обидеть и никого ни к чему не призываю. Я просто делясь своими мыслями в своей странной манере. Если что не так - сорян.

КТО СЛЫШАЛ ПРО VIM?

КТО ЗАХОДИЛ В VIM?

КТО ВЫХОДИЛ ИЗ VIM?

КТО ВЫХОДИЛ ИЗ VIM?



КТО ВЫХОДИЛ ИЗ VIM?

 **I Am Devloper** Following

@iamdevloper

I've been using Vim for about 2 years now, mostly because I can't figure out how to exit it.

Reply Retweet Favorite More

RETWEETS	FAVORITES
4,846	2,105

4:56 AM · 18 Feb 2014

Ваш первый раз...

Ваш первый раз...

if (Vim === боль) {

Ваш первый раз...

```
if (Vim === боль) { }
```



ЧТО ЭТО?

ЧТО ЭТО?

КОМУ ЭТО?

ЧТО ЭТО?

КОМУ ЭТО?

ЭТО БОЛЬНО?

ЧТО ЭТО?

КОМУ ЭТО?

ЭТО БОЛЬНО?

А ПРИЧЁМ ТУТ Я?

ЧТО ЭТО?

КОМУ ЭТО?

ЭТО БОЛЬНО?

А ПРИЧЁМ ТУТ Я?



Кто я такой?

Кто я такой?



Кто я такой?



Николай Сладкий

26 лет

2 года в уголовном розыске

3 года в веб-разработке

Где я работаю?

Chief Technical Officer

Где я работаю?

MULTIKEY

Сайты, которые продают. Трафик, который покупает

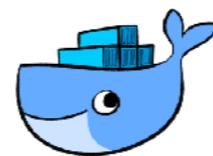
Где я работаю?

MULTIKEY

Сайты, которые продают. Трафик, который покупает

Chief Technical Officer

Где я работаю?

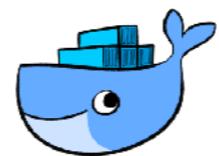


MULTIKEY

Сайты, которые продают. Трафик, который покупает

Chief Technical Officer

Где я работаю?

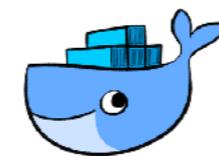


MULTIKEY

Сайты, которые продают. Трафик, который покупает

Chief Technical Officer

Где я работаю?



MULTIKEY

Сайты, которые продают. Трафик, который покупает



Олег Бартунов

Chief Technical Officer

Чем ещё я занимаюсь?

Чем ещё я занимаюсь?

SAMARA

IT COMMUNITY

Чем ещё я занимаюсь?

SAMARA

IT COMMUNITY

VP, Business Development

GRINDCONF

THE LAST HOPE FOR HUMANITY... RESTS ON A HIGH-POWER MACHINE GUN!



PLANET TERROR

QUENTIN ^{and} ROBERT TARANTINO RODRIGUEZ

are back! But this time they're

BACK to BACK!

2 1/2 HOURS OF PURE DYNAMITE!

TOGETHER IN ONE SMASH EXPLOSIVE SHOW

JULY 7 2018

DIMENSION FILMS PRESENTS A TROUBLEMAKER INTERNATIONAL PICTURE
DIRICHED BY QUENTIN TARANTINO & ROBERT RODRIGUEZ PRODUCED BY JEFF COOPER & DAVID WEINSTEIN
WRITTEN BY QUENTIN TARANTINO & ROBERT RODRIGUEZ
PRODUCTION DESIGNER: RICHARD RODRIGUEZ
COSTUME DESIGNER: KAREN KELLY
PROPS: DAVID ELIZABETH WILSON
CINEMATOGRAPHY: JEFF COOPER & DAVID WEINSTEIN
DIRECTOR OF PHOTOGRAPHY: DAVID WEINSTEIN
EDITOR: ROBERT RODRIGUEZ & DAVID WEINSTEIN
MUSIC: DAVID WEINSTEIN
PRODUCTION DESIGNER: KAREN KELLY
COSTUME DESIGNER: KAREN KELLY
PROPS: DAVID ELIZABETH WILSON
CINEMATOGRAPHY: JEFF COOPER & DAVID WEINSTEIN
DIRECTOR OF PHOTOGRAPHY: DAVID WEINSTEIN
EDITOR: ROBERT RODRIGUEZ & DAVID WEINSTEIN
MUSIC: DAVID WEINSTEIN
TROUBLEMAKER STUDIOS

These 8 women are about to meet 1 diabolical man!



KURT RUSSELL is

'DEATH PROOF'

DIMENSION FILMS PRESENTS
DIRICHED BY ROBERT RODRIGUEZ'S 'DEATH PROOF'
PRODUCED BY DAVID WEINSTEIN & DAVID COOPER
WRITTEN BY ROBERT RODRIGUEZ
COSTUME DESIGNER: KAREN KELLY
PROPS: DAVID ELIZABETH WILSON
CINEMATOGRAPHY: JEFF COOPER & DAVID WEINSTEIN
DIRECTOR OF PHOTOGRAPHY: DAVID WEINSTEIN
EDITOR: ROBERT RODRIGUEZ & DAVID WEINSTEIN
MUSIC: DAVID WEINSTEIN
TROUBLEMAKER STUDIOS

DIMENSION FILMS

vk.com/samara_it_community

VP, Business Development

И так, про Vim

И так, про Vim

Vi Improved (VIM) – свободный текстовый редактор, созданный на основе более старого UNIX редактора - vi

И так, про Vim

Vi Improved (VIM) – свободный текстовый редактор, созданный на основе более старого UNIX редактора - vi

Первая версия vi была написана Биллом Джоем в 1976 году.



И так, про Vim

Vi Improved (VIM) – свободный текстовый редактор, созданный на основе более старого UNIX редактора - vi

Первая версия vi была написана Биллом Джоем в 1976 году.



И так, про Vim

Vi Improved (VIM) – свободный текстовый редактор, созданный на основе более старого UNIX редактора - vi

Первая версия vi была написана Биллом Джоем в 1976 году.



ВСЕ ЮЗАЛИ VI И РАДОВАЛИСЬ, ПОКА...

Пока в 1988 году некий Брам Моленар...

Пока в 1988 году некий Брам Моленар...



Пока в 1988 году некий Брам Моленар... не купил компьютер Amiga.



Пока в 1988 году некий Брам Моленар... не купил компьютер Amiga.

Он юзал "stevie" и у него жутко пригорало,
ему не нравилось, и он начал на писать Vim



Vim “всем зашёл”!

Его опубликовали на Fred Fish disk 591
(free ПО collection для Amiga), перенесли на
UNIX, MS-DOS, он начал обрастать
контрибьютерами и комьюнити.

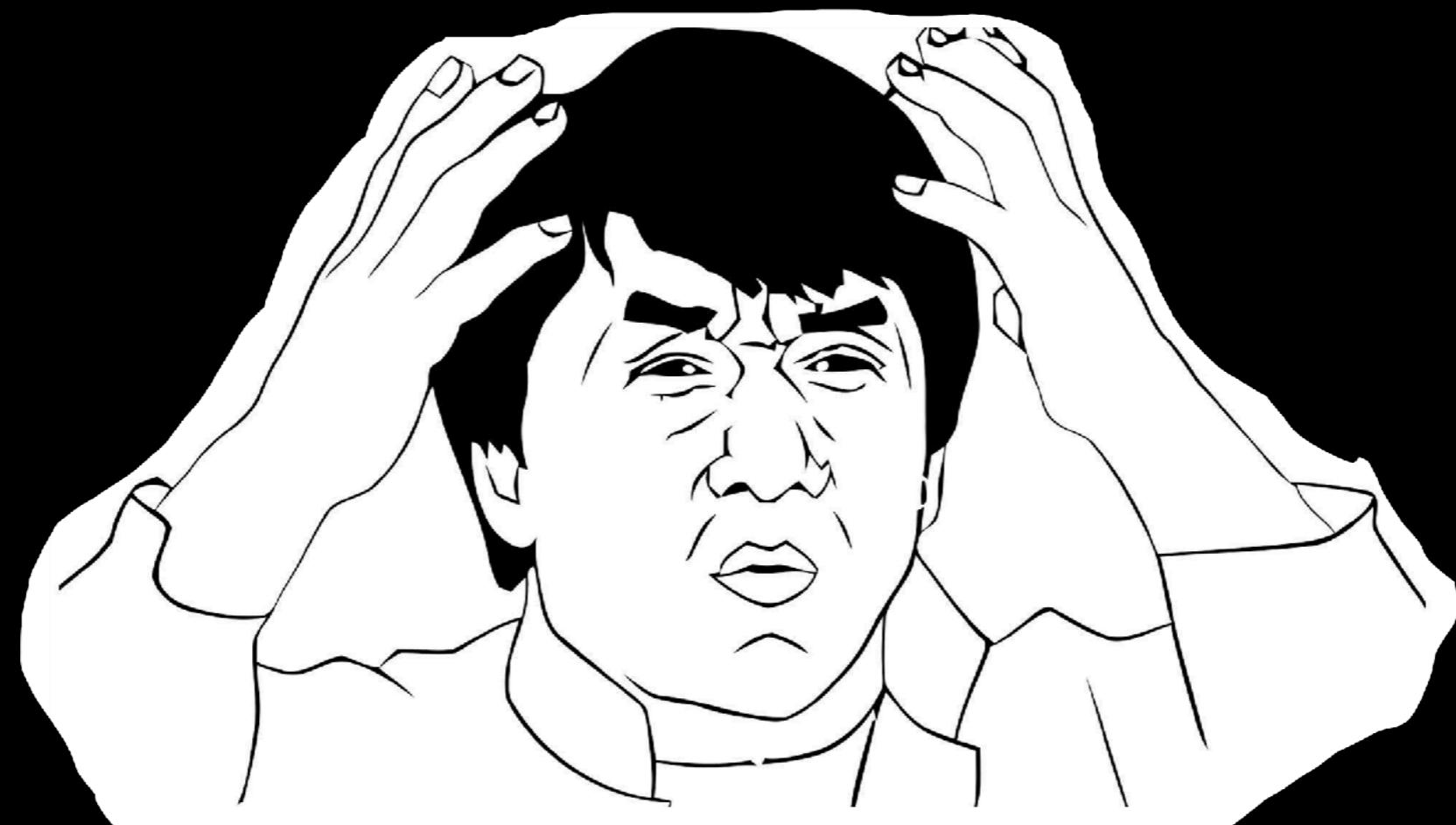
Vim “всем зашёл”!

Его опубликовали на Fred Fish disk 591
(free ПО collection для Amiga), перенесли на
UNIX, MS-DOS, он начал обрастать
контрибьютерами и комьюнити.

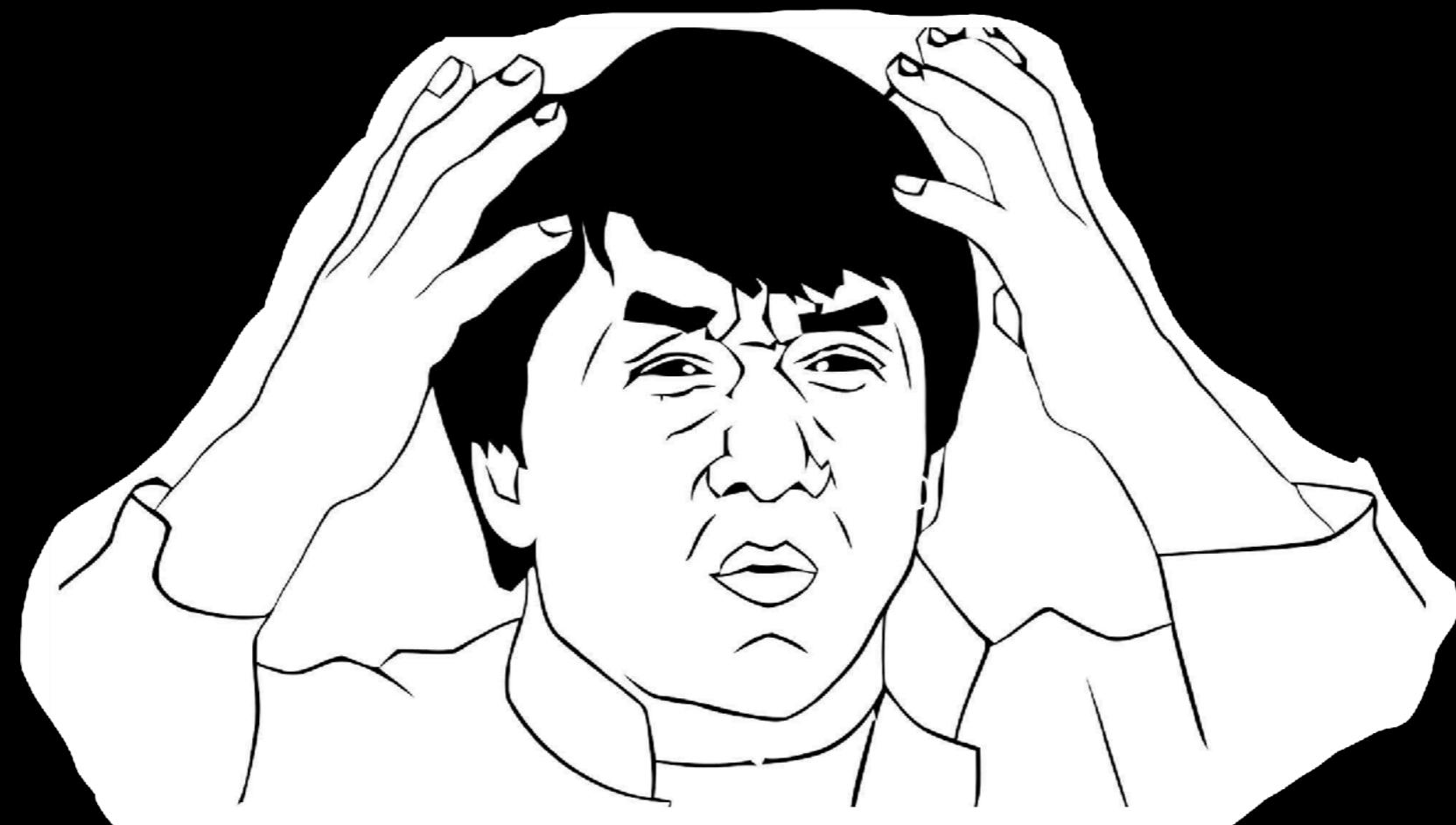
ВСЕ НАЧАЛИ ЮЗАТЬ VIM И РАДОВАТЬСЯ ПУЩЕ ПРЕЖНЕГО!

Но в чём “прекол”?

Но в чём “прекол”?



Но в чём “прекол”?



в 2018 году...

Если "из коробки" Vim выглядит так...

Если "из коробки" Vim выглядит так...

The screenshot shows a terminal window with a black background. At the top, there is a status bar containing several icons and text: a blue square icon, the word 'INSERT', the text '[No Name] [+]', '[unix]', '100%', a small icon, '4/41', another small icon, and a colon ':'. Below the status bar, the main area of the terminal is mostly empty, with only a single character '1' visible in the bottom-left corner. The rest of the screen is dark and devoid of any other text or graphical elements.

БЕНЕФИТЫ:

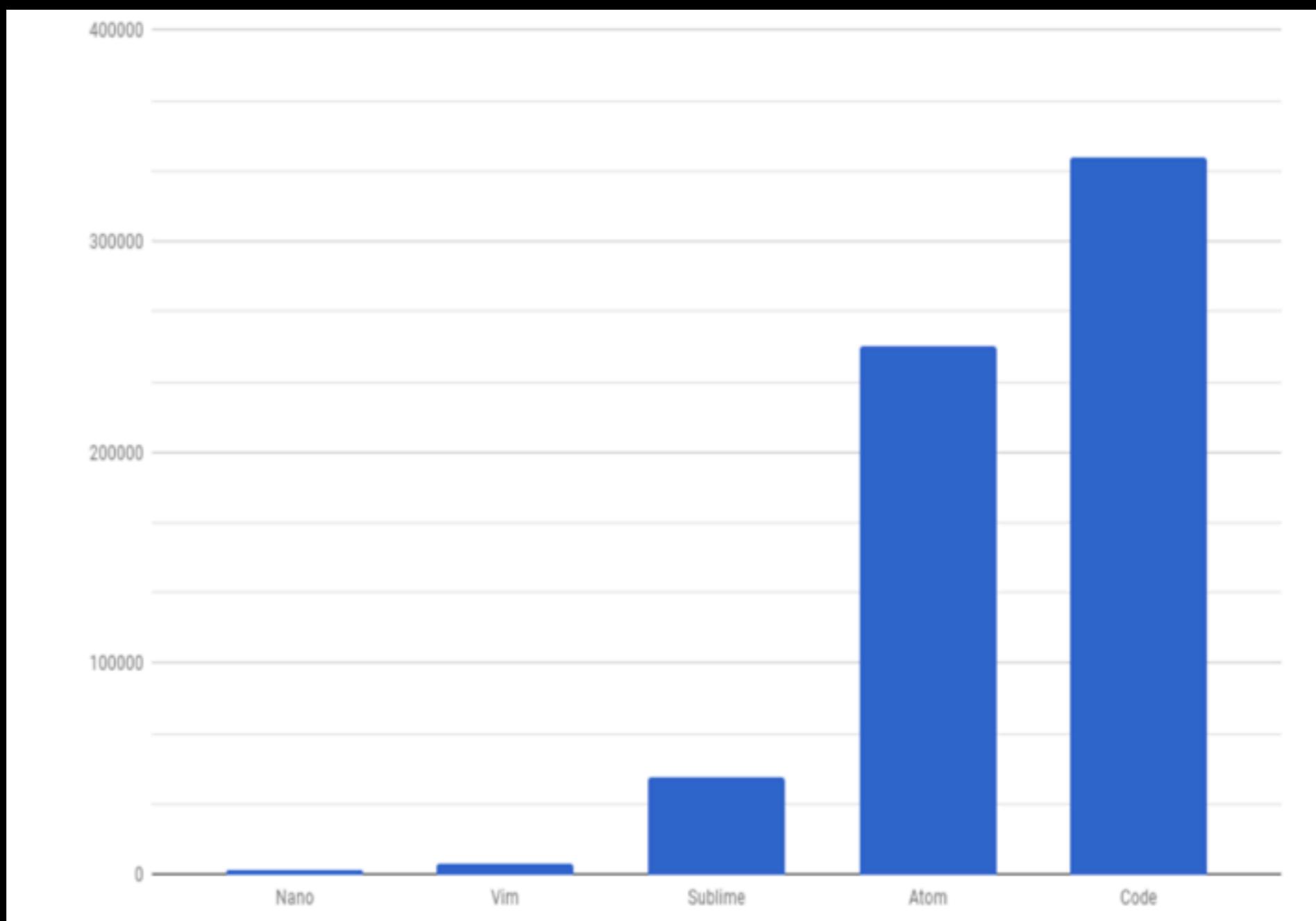
БЕНЕФИТЫ:

1) Терминал

БЕНЕФИТЫ:

- 1) Терминал
- 2) Терминал ;D (ресурсы и скорость)

Память



БЕНЕФИТЫ:

- 1) Терминал
- 2) Терминал ;D (ресурсы и скорость)
- 3) Кастомизация

```

File Edit View Scrollback Bookmarks Settings Help
" Press ? for help

.. (up a dir)
/var/www/
+advertpeople/
+apache2-default/
+cgi-bin/ -> /usr/lib/cgi-bin/
~drupal/
| ~includes/
| | -bootstrap.inc*
| | -cache.inc*
| | -common.inc*
| | -database.inc*
| | -database.mysql.inc*
| | -database.mysqli.inc*
| | -database.pgsql.inc*
| | -file.inc*
| | -form.inc*
| | -image.inc*
| | -install.inc*
| | -install.mysql.inc*
| | -install.mysqli.inc*
| | -install.pgsql.inc*
| | -locale.inc*
| | -menu.inc*
| | -module.inc*
| | -pager.inc*
| | -path.inc*
| | -session.inc*
| | -tablesort.inc*
| | -theme.inc*
| | -unicode.inc*
| | -xmlrpc.inc*
| | ` -xmlrpcs.inc*
+misc/
+modules/
+profiles/
mysql_query  sends a query to the currently active database on the server
that's associated with the specified link identifier. If {link_identifier}
isn't specified, the last opened link is assumed. If no link is open, the
function tries to establish a link as if mysql_connect was called with no
arguments, and use it. The result of the query is buffered.

mysql_real_escape_string()
string mysql_real_escape_string(string unescaped_string [, resource link_identifier])
{unescaped_string} The string to escape {link_identifier} (optional) The mysql
connection resource

mysql_result()
mixed mysql_result(resource result, int row [, mixed field])
mysql_result returns the contents of one cell from a MySQL result set. The
field argument can be the field's offset, or the field's name, or the field's
table dot field name (tablename.fieldname). If the column name has been
aliased ('select foo as bar from...'), use the alias instead of the column
php_manual.txt[Help][-][R0] 42 utf-8 0x 11650,51 54%
218     throw new LogicException("No sql query for prepare to execute :()");
219
220 }
221
222     if(func_num_args() < 2) {
223
224         throw new InvalidArgumentException("Two or more arguments required");
225
226 }
227
228     $args = func_;
229
230     $escaping_str = array_shift($args);
231
232     if(strlen($escaping_str) != sizeof($args)) {
233
<1[-] 97 utf-8 0x 30,8 Наверху CWPuDB.php[+] 95 utf-8 0x 228,21 38%
"php_manual.txt" [только для чтения] 21517L, 941824C
mobikit : vim

```

```
*-----*  
| It is easier to change the specification to fit the |  
| program than vice versa.  
|  
| - Alan Perlis  
*-----*  
| | | o  
| | | |o ^__^  
| | | |o (oo)\_____  
| | | | (__)\       )\/\| | | | |
| | | | | |----w |  
| | | | | |     ||  
  
[e] <empty buffer>  
  
MRU  
  
[0] ~/dev/magplan/app/controllers/articles_controller.rb  
[1] ~/dev/magplan/app/controllers/issues_controller.rb  
[2] ~/.vimrc  
[3] ~/.tmuxinator/vn.yml  
[4] ~/.wine/system.reg  
[5] ~/.gdbinit  
[6] /Applications/Google Chrome.app/Contents/MacOS/Google Chrome  
[7] /usr/local/etc/proxychains.conf  
[8] /usr/local/Cellar/metasploit-framework/4.16.4_1/share/metasploit-framework/  
[9] /usr/local/Cellar/metasploit-framework/4.16.4_1/share/metasploit-framework/  
  
MRU /Users/user  
  
[10] dev/magplan/app/controllers/articles_controller.rb  
[11] dev/magplan/app/controllers/issues_controller.rb  
[12] .vimrc  
[13] .tmuxinator/vn.yml  
startify [noperm]  
  
" Press ? for help  
.. (up a dir)  
/Users/user/  
▶ .bash_sessions/  
▶ .BurpSuite/  
▶ .cache/  
▶ .config/  
▶ .docker/  
▶ .dropbox/  
▶ .gem/  
▶ .lldb/  
▶ .local/  
▶ .msf4/  
▶ .npm/  
▶ .oh-my-zsh/  
▶ .oracle_jre_usage/  
▶ .recon-ng/  
▶ .ssh/  
▶ .tmuxinator/  
▶ .tor/  
▶ .Trash/  
▶ .vim/  
▶ .voltron/  
▶ .wine/  
▶ Applications/  
▶  
▶  
▶  
▶  
▶ NERD
```

Файл Правка Вид Поиск Терминал Справка

```
endif

ifdef BOOST_INCLUDE_PATH
    CPPFLAGS += -I$(BOOST_INCLUDE_PATH)
endif

ifdef BOOST_LIBRARY_PATH
    EXTRALIBS += -L$(BOOST_LIBRARY_PATH)
endif

ifeq ($(CEXMC_USE_PERSISTENCY),yes)
    EXTRALIBS += -lboost_serialization
    CPPFLAGS += -DCEXMC_USE_PERSISTENCY
    ifeq ($(CEXMC_USE_CUSTOM_FILTER),yes)
        CPPFLAGS += -DCEXMC_USE_CUSTOM_FILTER
        ifeq ($(CEXMC_DEBUG_CUSTOM_FILTER),yes)
            CPPFLAGS += -DCEXMC_DEBUG_CF
        endif
    endif
endif

GNUmakefile
{
    break;
}
#endif CEXMC_USE_PERSISTENCY
if ( G4String( argv[ i ], 2 ) == "-w" )
{
    cmdLineData.wProject = argv[ i ] + 2;
    if ( cmdLineData.wProject == "" )
    {
        if ( ++i >= argc )
            throw CexmcException( CexmcCmdLineParseException );
        cmdLineData.wProject = argv[ i ];
    }
    break;
}
if ( G4String( argv[ i ], 2 ) == "-r" )
{
    cmdLineData.rProject = argv[ i ] + 2;
    if ( cmdLineData.rProject == "" )
183:    if ( G4String( argv[ i ], 2 ) == "-w" )
```

" Press <F1> for help

- ▼ __anon1* : namespace
 - [members]
 - CexmcVisManagerVerboseLevel
- ▼ CexmcCmdLineData : struct
 - [functions]
 - +CexmcCmdLineData()
 - [members]
 - +isInteractive
 - +startQtSession
 - +preinitMacro
 - +initMacro
 - +rProject
 - +wProject
 - +overrideExistingProject
 - +outputData
 - +customFilter
- ▼ functions
 - printUsage(void)
 - parseArgs(int argc, char ** argv, CexmcVisManagerVerboseLevel & verboseLevel)
 - main(int argc, char ** argv)

61% 62:0

cexmc.cc unix utf-8 < cpp 38% 193:13 Tagbar Tree

Applications Places System en Tue Feb 27, 17:58

avrsim.c (~/git/mcusim/avrsim) - GVIM

File Edit Tools Syntax Buffers Window Help

" Press ? for help

```

.. (up a dir)
/usr/home/dsl/
  Desktop/
  Documents/
  Downloads/
  games/
  git/
    dht-client/
    firesparks/
  mcusim/
    avrsim/
      include/
      man/
        avrsim.c
        decoder.c
        gdb_rsp.c
        getopt_long.c
        ihex.c
        peripheral_lua.c
        simcore.c
        simm2560.c
        simm328.c
        simm328p.c
        simm8a.c
        vcd_dump.c
      build/
      cmake/
      handbook/
      peripheral/
      rom/
      tests/
        CMakeLists.txt
        CONTRIBUTING.md
        LICENSE
        README.md
    xling/
  Pictures/
    caja.core
    dbus-launch.core
    ds_kdbx
    kicad.core
    marco.core
    mate-system-monitor.core
    octave-gui.core
    octave-workspace

  /usr/home/dsl

```

20 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 21 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR
 22 * ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 23 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 24 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 25 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 26 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
 27 * ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
 28 * OF SUCH DAMAGE.

29 */

30 #include <stdio.h>
 31 #include <stdint.h>
 32 #include <string.h>
 33 #include <stdlib.h>
 34 #include "mcusim/cli.h"
 35 #include "mcusim/getopt.h"
 36 #include "mcusim/avr/sim/sim.h"
 37 #include "mcusim/avr/sim/simcore.h"
 38 #include "mcusim/avr/sim/bootloader.h"
 39 #include "mcusim/avr/sim/peripheral_lua.h"
 40 #include "mcusim/avr/sim/gdb_rsp.h"
 41 #include "mcusim/avr/sim/interrupt.h"
 42 #include "mcusim/avr/sim/vcd_dump.h"
 43 /* Limits of statically allocated MCU memory */
 44 #define PMSZ 256*1024 /* Program memory limit */
 45 #define DMSZ 64*1024 /* Data memory limit */
 46 #define PM_PAGESZ 1024 /* PM page size */
 47
 48 #define GDB_RSP_PORT 12750 /* Default GDB RSP port */
 49 #define MAX_MEMOPS 16 /* Maximum MCU memory
 50 modifications during
 51 initialization */
 52
 53
 54 /* Command line options */
 55 #define CLI_OPTIONS ":p:U:r:P:f:"
 56 #define DUMP_REGS_OPT 7575
 57 #define VERSION_OPT 7576
 58 #define GDB_RSP_PORT_OPT 7577
 59 #define TRAP_AT_ISR_OPT 7578
 60
 61 static struct option longopts[] = {
 62 { "dump-reg", required_argument, NULL, DUMP_REGS_OPT },
 63 { "version", no_argument, NULL, VERSION_OPT },
 64 { "rsp-port", required_argument, NULL, GDB_RSP_PORT_OPT },
 65 { "trap-at-isr", no_argument, NULL, TRAP_AT_ISR_OPT }
 66 };
 67 /* END Command line options */
 68
 69 static struct MSIM AVR mcu; /* Central MCU descriptor */
 70 static struct MSIM_AVRBootloader bls; /* Bootloader section */
 71 static struct MSIM_AVRIInt intr; /* Interrupts and IRQs */
 72 static struct MSIM_VCDDetails vcdd; /* Details about registers to dump */
 73

" Press <F1>, ? for help

► macros

► prototypes

▼ variables

- bls
- dm
- intr
- longopts
- mcu
- memops
- memops_num
- mpm
- pm
- ppm
- print_regs
- vcdr
- vcdd**

▼ functions

- apply_memop(struct MSTM_AVR *m, struct main(int argc, char *argv[])
- parse_dump(char *cmd)
- parse_freq(const char *opt)
- parse_memop(char *opt)
- parse_rsp_port(const char *opt)
- print_config(const struct MSIM_AVR *)
- print_usage(void)
- set_fuse(struct MSIM_AVR *m, struct
- set_lock(struct MSIM_AVR *m, struct

73,0-1 4% [Name] avrsim.c

БЕНЕФИТЫ:

- 1) Терминал
- 2) Терминал ;D (ресурсы и скорость)
- 3) Кастомизация

БЕНЕФИТЫ:

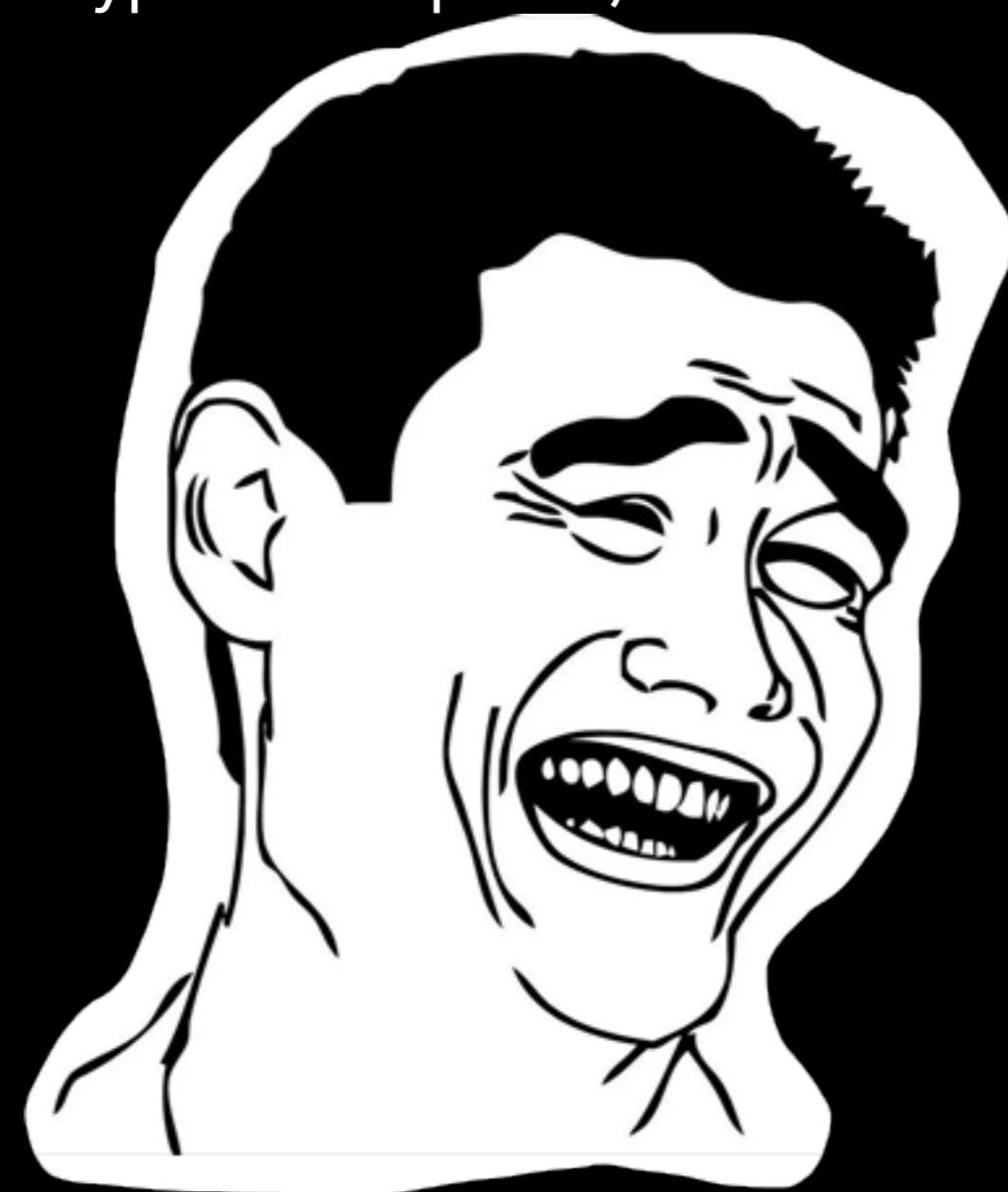
- 1) Терминал
- 2) Терминал ;D (ресурсы и скорость)
- 3) Кастомизация

И ВСЁ? СЕРЬЕЗНО?

БЕНЕФИТЫ:

- 1) Терминал
- 2) Терминал ;D (ресурсы и скорость)
- 3) Кастомизация

И ВСЁ? СЕРЬЕЗНО?



БЕНЕФИТЫ:

- 1) Терминал
- 2) Терминал ;D (ресурсы и скорость)
- 3) Кастомизация
- 4) Модальность (режимы)

Vim - это редактор с ручным переключением режимов ввода.
Переключение режимов раскрывает основной принцип Vim -
принцип "модальности".

Vim - это редактор с ручным переключением режимов ввода.
Переключение режимов раскрывает основной принцип Vim -
принцип "модальности".

Говоря просто: одни кнопки в разных режимах имеют разные функции.

Vim - это редактор с ручным переключением режимов ввода. Переключение режимов раскрывает основной принцип Vim - принцип "модальности".

Говоря просто: одни кнопки в разных режимах имеют разные функции.

МОДАЛЬНОСТЬ - ОСНОВА ФИЛОСОФИИ VIM!

но пока мы не начали “вотэтовотвсё”...

но пока мы не начали “вотэтовотвсё”...

НИМАНИЕ! АЖНО!

но пока мы не начали “вотэтовотвсё”...

НИМАНИЕ! АЖНО!

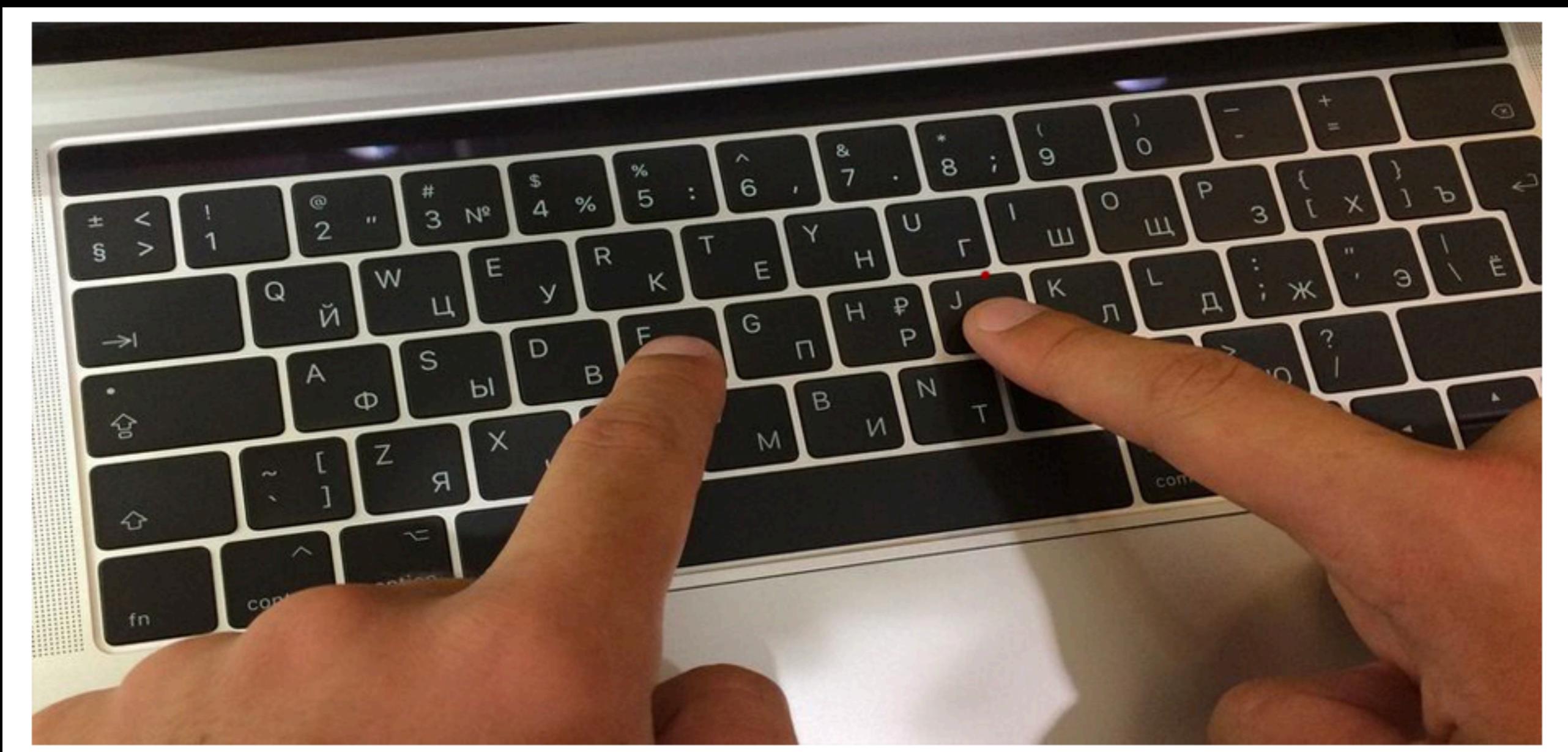
но пока мы не начали "вотэтовотвсё"...

НИМАНИЕ! АЖНО!

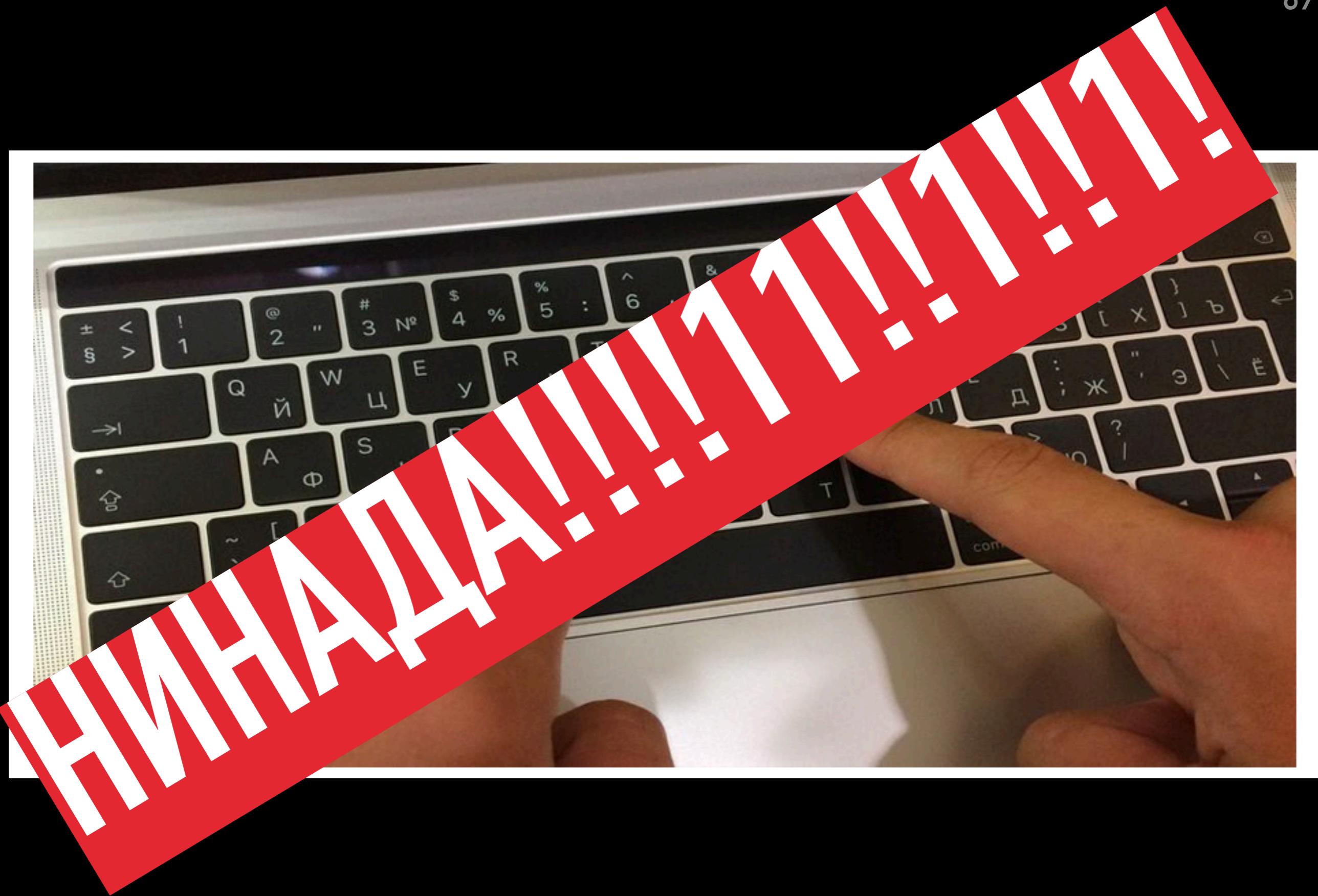
но пока мы не начали “вотэтовотвсё”...

НИМАНИЕ! АЖНО!

Хочешь Vim? Хоти слепую печать!







Режимы:

Режимы:

- "Нормальный". Он же "командный". Режим по умолчанию.

Тот самый режим, который не дает печатать!

Нажатия клавиш воспринимаются
как команды.



Режимы:

- "Нормальный". Он же "командный". Режим по умолчанию.

Тот самый режим, который не дает печатать!

Нажатия клавиш воспринимаются
как команды.



- "Вставка". Режим набора текста.

Нажатия клавиш воспринимаются как нажатие клавиш...

Режимы:

- "Нормальный". Он же "командный". Режим по умолчанию.

Тот самый режим, который не дает печатать!

Нажатия клавиш воспринимаются
как команды.



- "Вставка". Режим набора текста.

Нажатия клавиш воспринимаются как нажатие клавиш...

- "Визуальный". Режим выделения текста.

Нажатия клавиш воспринимаются... ближе к командному, но есть отличия.

Режимы:

- "Нормальный". Он же "командный". Режим по умолчанию.

Тот самый режим, который не дает печатать!

Нажатия клавиш воспринимаются
как команды.



- "Вставка". Режим набора текста.

Нажатия клавиш воспринимаются как нажатие клавиш...

- "Визуальный". Режим выделения текста.

Нажатия клавиш воспринимаются... ближе к командному, но есть отличия.

- "Командная строка". Режим сложных команд.

Вызывается командная строка, где выполняются составные команды.

Посмотрим на Vim Cheat Sheet

Посмотрим на Vim Cheat Sheet

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

ESC	normal mode											
~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down	+ next line
^ goto mark	1 ²	2	3	4	5	6	7	8	9	0 "hard" bol	- prev line	= auto ³ format
Q ex mode	W next WORD	E end WORD	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	}	end parag.
q record macro	W next word	e end word	r replace char	t 'till	y yank ^{1,3}	u undo	i insert mode	o open below	p paste ¹ after	[misc]	• misc
A append at eol	S subst line	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd line	" reg. spec ¹	bol/ goto col	
a append	s subst char	d delete ^{1,3}	f find char	g extra ⁶ cmd	h ←	j ↓	k ↑	l →	; repeat t/T/f/F	' goto mk. bol	\ not used!	
Z quit ⁴	X back-space	C change to eol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un-indent ³	> indent ³	? find (rev.)			
Z extra ⁵ cmd	X delete char	C change ^{1,3}	V visual mode	b prev word	n next (find)	m set mark	,	,	/ find			

motion moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

Посмотрим на Vim Cheat Sheet

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

ESC	normal mode											
~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down	+ next line
^ goto mark	1 ²	2	3	4	5	6	7	8	9	0 "hard" bol	- prev line	= auto ³ format
Q ex mode	W next WORD	E end WORD	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	} end parag.	
q record macro	W next word	e end word	r replace char	t 'till	y yank ^{1,3}	u undo	i insert mode	o open below	p paste ¹ after	[misc] misc	
A append at eol	S subst line	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd	" reg. ¹ spec	bol/ goto col	
a append	s subst char	d delete ^{1,3}	f find char	g extra ⁶ cmd	h ←	j ↓	k ↑	l →	; repeat t/T/f/F	' goto mk. bol	\ not used!	
Z quit⁴	X back-space	C change to eol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un-³ indent	> indent³	? find (rev.)			
Z extra ⁵ cmd	X delete char	C change ^{1,3}	V visual mode	b prev word	n next (find)	m set mark	,	,	/ find			

motion moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

МНОГАБУКАФ!

Попробуем разобраться

Попробуем разобраться

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

ESC	normal mode											
~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down	+ next line
^ goto mark	1 ²	2	3	4	5	6	7	8	9	0 "hard" bol	- prev line	= auto ³ format
Q ex mode	W next WORD	E end WORD	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	}	end parag.
q record macro	W next word	e end word	r replace char	t 'till	y yank ^{1,3}	u undo	i insert mode	o open below	p paste ¹ after	[misc]	• misc
A append at eol	S subst line	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd line	" reg. ¹ spec	bol/ goto col	
a append	s subst char	d delete ^{1,3}	f find char	g extra ⁶ cmd	h ←	j ↓	k ↑	l →	; repeat t/T/f/F	' goto mk. bol	\ not used!	
Z quit ⁴	X back-space	C change to eol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un- ³ indent	> indent ³	? find (rev.)			
Z extra ⁵ cmd	X delete char	C change ^{1,3}	V visual mode	b prev word	n next (find)	m set mark	,	, reverse t/T/f/F	/ find			

motion moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

Попробуем разобраться

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

ESC	normal mode											
~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down	+ next line
^ goto mark	1 2	3 4	5 6	7 8	9 0	"hard" bol	- prev line	= auto format				
Q ex mode	W next WORD	E end WORD	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	}	
q record macro	W next word	e end word	r replace char	t 'till	y yank	u undo	i insert mode	o open below	p paste after	m misc	m misc	
A append at eol	S subst line	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd line	" reg. spec	bol/ goto col	
a append	s subst char	d delete	f find char	g extra cmd	h ←	j ↓	k ↑	l →	; repeat t/T/f/F	' goto mk. bol	\ not used!	
Z quit	X back-space	C change to eol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un indent	> indent	? find (rev.)			
Z extra cmd	X delete char	C change	V visual mode	b prev word	n next (find)	m set mark	,	,	/ find			

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

Попробуем разобраться

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

ESC	normal mode											
~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down	+ next line
^· goto mark	1 ²	2	3	4	5	6	7	8	9	0 "hard" bol	- prev line	= auto ³ format
Q ex mode	W next WORD	E end WORD	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	}	end parag.
q record macro	W next word	e end word	r replace char	t 'till	y yank ^{1,3}	u undo	i insert mode	o open below	p paste ¹ after	[misc]	misc
A append at eol	S subst line	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd line	'. reg. spec	bol/ goto col	
a append	s subst char	d delete ^{1,3}	f find char	g extra ⁶ cmd	h ←	j ↓	k ↑	l →	; repeat t/T/f/F	' goto mk. bol	\ not used!	
Z quit ⁴	X back-space	C change to eol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un- ³ indent	> indent ³	? find (rev.)			
Z extra ⁵ cmd	X delete char	C change ^{1,3}	V visual mode	b prev word	n next (find)	m set mark	,	,	/ find			

on moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

Попробуем разобраться

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

ESC	normal mode											
~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down	+ next line
^· goto mark	1 ²	2	3	4	5	6	7	8	9	0 "hard" bol	- prev line	= auto ³ format
Q ex mode	W next WORD	E end WORD	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	}	end parag.
q record macro	W next word	e end word	r replace char	t 'till	y yank ^{1,3}	u undo	i insert mode	o open below	p paste ¹ after	[misc]	misc
A append at eol	S subst line	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd line	'. reg. spec	bol/ goto col	
a append	s subst char	d delete ^{1,3}	f find char	g extra ⁶ cmd	h ←	j ↓	k ↑	l →	; repeat t/T/f/F	' goto mk. bol	\ not used!	
Z quit ⁴	X back-space	C change to eol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un- ³ indent	> indent ³	? find (rev.)			
Z extra ⁵ cmd	X delete char	C change ^{1,3}	V visual mode	b prev word	n next (find)	m set mark	,	,	/ find			

n moves the cursor, or defines the range for an operator

c command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

Попробуем разобраться

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

ESC	normal mode											
~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down	+ next line
^ goto mark	1 ²	2	3	4	5	6	7	8	9	0 "hard" bol	- prev line	= auto ³ format
Q ex mode	W next WORD	E end WORD	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	}	end parag.
q record macro	W next word	e end word	r replace char	t 'till	y yank ^{1,3}	u undo	i insert mode	o open below	p paste ¹ after	[misc]	• misc
A append at eol	S subst line	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd line	'. reg. spec	bol/ goto col	
a append	s subst char	d delete ^{1,3}	f find char	g extra ⁶ cmd	h ←	j ↓	k ↑	l →	; repeat t/T/f/F	' goto mk. bol	\ not used!	
Z quit ⁴	X back-space	C change to eol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un- ³ indent	> indent ³	? find (rev.)			
Z extra ⁵ cmd	X delete char	C change ^{1,3}	V visual mode	b prev word	n next (find)	m set mark	,	,	/ find			

motion moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

Первый вопрос...

Первый вопрос...

КАК ПИСАТЬ?????!!?!!?!!!11/11!?!//

Первый вопрос...

КАК ПИСАТЬ?????!!?!?!!!11/11!?!/



КОМАНДЫ

КОМАНДЫ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

ESC normal mode											
~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down
^ goto mark	1 2 3 4 5 6 7 8 9 0 "hard" bol										- prev line + next line
Q ex mode	W next WORD	E end WORD	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	
q record macro	W next word	e end word	r replace char	t 'till	y yank ^{1,3}	u undo	i insert mode	o open below	p paste after	}	end parag.
A append at eol	S subst line	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd line	" reg. spec	bol/ goto col
a append	s subst char	d delete ^{1,3}	f find char	g extra ⁶	h ←	j ↓	k ↑	l →	;" repeat t/T/f/F	' goto mk. bol	\ not used!
Z quit ⁴	X back-space	C change to eol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un- ³	> indent ³	? find (rev.)		
Z extra ⁵	X delete char	C change ^{1,3}	V visual mode	b prev word	n next (find)	m set mark	,	, reverse t/T/f/F	/ find		

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)

(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times

(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,

zb: bottom, zz: center
- (6) gg: top of file (vim only),

gf: open file under cursor (vim only)

moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDs: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

КОМАНДЫ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

ESC	normal mode													
~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down	_ prev line	+ next line	
~ goto mark	1 ²	2	3	4	5	6	7	8	9	0 "hard" bol	-	= auto ³ format		
O ex mode	W next WORD	E end WORD	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	}	end parag.		
Q record macro	W next word	E end word	R replace char	T 'till	Y yank ^{1,3}	U undo	I insert mode	O open below	P paste after	[misc]	[misc		
A append at eol	S subst line	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd line	" reg. spec ¹	bol/ goto col			
a append	s subst char	d delete ^{1,3}	f find char	g extra ⁶ cmd	h ←	j ↓	k ↑	l →	; repeat t/T/f/F	' goto mk. bol	\ not used!			
Z quit ⁴	X back-space	C change to eol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un- ³ indent	> indent ³	? find (rev.)					
Z extra ⁵ cmd	X delete char	C change ^{1,3}	V visual mode	b prev word	n next (find)	m set mark	,	, reverse t/T/f/F	/ find					

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)

(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times

(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,

zb: bottom, zz: center
- (6) gg: top of file (vim only),

gf: open file under cursor (vim only)

moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDs: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

КОМАНДЫ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

ESC normal mode																																																																																			
~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down																																																																								
^ goto mark	1 2 3 4 5 6 7 8 9 0 "hard" bol									- prev line	+ next line auto ³ format																																																																								
<table border="1"> <tr><td>Q ex mode</td><td>W next WORD</td><td>E end WORD</td><td>R replace mode</td><td>T back 'till</td><td>Y yank line</td><td>U undo line</td><td>I insert at bol</td><td>O open above</td><td>P paste before</td><td>{ begin parag.</td><td>}</td></tr> <tr><td>q record</td><td>W next word</td><td>e end word</td><td>r replace char</td><td>t 'till</td><td>y yank^{1,3}</td><td>u undo</td><td>i insert mode</td><td>o open below</td><td>p paste¹ after</td><td>m misc</td><td>m misc</td></tr> <tr><td>A append at eol</td><td>S subst line</td><td>D delete to eol</td><td>F "back" find ch</td><td>G eof/ goto ln</td><td>H screen top</td><td>J join lines</td><td>K help</td><td>L screen bottom</td><td>: ex cmd line</td><td>" reg. spec¹</td><td>bol/ goto col</td></tr> <tr><td>a append</td><td>s subst char</td><td>d delete^{1,3}</td><td>f find char</td><td>g extra⁶ cmd</td><td>h ←</td><td>j ↓</td><td>k ↑</td><td>l →</td><td>; repeat t/T/f/F</td><td>' goto mk. bol</td><td>\ not used!</td></tr> <tr><td>Z quit⁴</td><td>X back-space</td><td>C change to eol</td><td>V visual lines</td><td>B prev WORD</td><td>N prev (find)</td><td>M screen mid'l</td><td>< un-indent³</td><td>> indent³</td><td>? find (rev.)</td><td></td><td></td></tr> <tr><td>Z extra⁵</td><td>X delete char</td><td>C change^{1,3}</td><td>V visual mode</td><td>b prev word</td><td>n next (find)</td><td>m set mark</td><td>,</td><td>,</td><td>/ find</td><td></td><td></td></tr> </table>												Q ex mode	W next WORD	E end WORD	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	}	q record	W next word	e end word	r replace char	t 'till	y yank ^{1,3}	u undo	i insert mode	o open below	p paste ¹ after	m misc	m misc	A append at eol	S subst line	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd line	" reg. spec ¹	bol/ goto col	a append	s subst char	d delete ^{1,3}	f find char	g extra ⁶ cmd	h ←	j ↓	k ↑	l →	; repeat t/T/f/F	' goto mk. bol	\ not used!	Z quit ⁴	X back-space	C change to eol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un-indent ³	> indent ³	? find (rev.)			Z extra ⁵	X delete char	C change ^{1,3}	V visual mode	b prev word	n next (find)	m set mark	,	,	/ find		
Q ex mode	W next WORD	E end WORD	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	}																																																																								
q record	W next word	e end word	r replace char	t 'till	y yank ^{1,3}	u undo	i insert mode	o open below	p paste ¹ after	m misc	m misc																																																																								
A append at eol	S subst line	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd line	" reg. spec ¹	bol/ goto col																																																																								
a append	s subst char	d delete ^{1,3}	f find char	g extra ⁶ cmd	h ←	j ↓	k ↑	l →	; repeat t/T/f/F	' goto mk. bol	\ not used!																																																																								
Z quit ⁴	X back-space	C change to eol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un-indent ³	> indent ³	? find (rev.)																																																																										
Z extra ⁵	X delete char	C change ^{1,3}	V visual mode	b prev word	n next (find)	m set mark	,	,	/ find																																																																										
<table border="1"> <tr><td>motion</td><td colspan="11">is the cursor, or defines range for an operator</td></tr> <tr><td>command</td><td colspan="11">direct action command, if red, it enters insert mode</td></tr> <tr><td>operator</td><td colspan="11">requires a motion afterwards, operates between cursor & destination</td></tr> <tr><td>extra</td><td colspan="11">special functions, requires extra input</td></tr> <tr><td>Q.</td><td colspan="11">commands with a dot need a char argument afterwards</td></tr> </table>												motion	is the cursor, or defines range for an operator											command	direct action command, if red, it enters insert mode											operator	requires a motion afterwards, operates between cursor & destination											extra	special functions, requires extra input											Q.	commands with a dot need a char argument afterwards																						
motion	is the cursor, or defines range for an operator																																																																																		
command	direct action command, if red, it enters insert mode																																																																																		
operator	requires a motion afterwards, operates between cursor & destination																																																																																		
extra	special functions, requires extra input																																																																																		
Q.	commands with a dot need a char argument afterwards																																																																																		
<p>bol = beginning of line, eol = end of line, mk = mark, yank = copy words: <code>quux(foo, bar, baz);</code> WORDs: <code>quux (foo, bar, baz);</code></p>																																																																																			
<p>Main command line commands ('ex'): <code>:w</code> (save), <code>:q</code> (quit), <code>:q!</code> (quit w/o saving) <code>:e f</code> (open file f), <code>:%s/x/y/g</code> (replace 'x' by 'y' filewide), <code>:h</code> (help in vim), <code>:new</code> (new file in vim),</p> <p>Other important commands: <code>CTRL-R</code>: redo (vim), <code>CTRL-F/-B</code>: page up/down, <code>CTRL-E/-Y</code>: scroll line up/down, <code>CTRL-V</code>: block-visual mode (vim only)</p> <p>Visual mode: Move around and type operator to act on selected region (vim only)</p>																																																																																			
<p>Notes:</p> <ul style="list-style-type: none"> (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*) (e.g.: "ay\$ to copy rest of line to reg 'a') (2) type in a number before any action to repeat it that number of times (e.g.: 2p, d2w, 5i, d4j) (3) duplicate operator to act on current line (dd = delete line, >> = indent line) (4) ZZ to save & quit, ZQ to quit w/o saving (5) zt: scroll cursor to top, zb: bottom, zz: center (6) gg: top of file (vim only), gf: open file under cursor (vim only) 																																																																																			

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

КОМАНДЫ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

ESC	normal mode											
~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down	+ next line
^· goto mark	1 ²	2	3	4	5	6	7	8	9	0 "hard" bol	- prev line	= auto ³ format
Q ex mode	W next WORD	E end WORD	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	}	end parag.
q record	W next word	e end word	r replace char	t 'till	y yank ^{1,3}	u undo	i insert mode	o open below	p paste ¹ after	[misc]	misc
A append at eol	S subst line	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd	'. reg. ¹ spec	bol/ goto col	
a append	s subst char	d delete ^{1,3}	f find char	g extra ⁶ cmd	h ←	j ↓	k ↑	l →	; repeat t/T/f/F	' goto mk. bol	\ not used!	
Z quit ⁴	X back-space	C change to eol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un- ³ indent	> indent ³	? find (rev.)			
Z extra ⁵ cmd	X delete char	C change ^{1,3}	V visual mode	b prev word	n next (find)	m set mark	,	,	/ find			
motion	is the cursor, or defines range for an operator											
command	direct action command, if red, it enters insert mode											
operator	requires a motion afterwards, operates between cursor & destination											
extra	special functions, requires extra input											
Q·	commands with a dot need a char argument afterwards											
bol = beginning of line, eol = end of line, mk = mark, yank = copy												
words: <code>quux(foo, bar, baz);</code>												
WORDS: <code>quux (foo, bar, baz);</code>												

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

I	insert at bol
i	insert mode

A	append at eol
a	append

O	open above
O	open below

C	change to eol
---	------------------

S	subst line
S	subst char

R	replace mode
---	-----------------

I	insert at bol
i	insert mode

A	append at eol
a	append

O	open above
O	open below

C	change to eol
---	------------------

S	subst line
S	subst char

R	replace mode
---	-----------------

ПРИМЕР

КОМАНДЫ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

ESC normal mode											
~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down
^ goto mark	1 2 3 4 5 6 7 8 9 0 "hard" bol									- prev line	+ next line auto ³ format
Q ex mode	W next WORD	E end WORD	R replace	T back 'till	Y yank line	U undo line	I insert bol	C open above	P paste before	{ begin parag.	
q record	W next word	e end word	r replace char	t 'till	y yank ^{1,3}	u undo	i insert de	o open below	p paste ¹ after	}	end parag.
A append	S soft delete	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd	" reg. spec ¹	bol/ goto col
a append	s soft delete	d delete ^{1,3}	f find char	g extra ⁶ cmd	h ←	j ↓	k ↑	l →	; repeat	' goto mk. bol	\ not used!
Z quit ⁴	X back-space	C change ^{1,3}	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un-indent ³	> indent ³	? find (rev.)		
Z extra ⁵ cmd	X delete char	C change ^{1,3}	V visual mode	b prev word	n next (find)	m set mark	,	,	/ find		

motion moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

Написали, что дальше?

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

ESC	normal mode											
~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down	+ next line
^ goto mark	1 2	3 4	5 6	7 8	9 0 "hard" bol	- prev line	= auto format					
Q ex mode	W next WORD	E end WORD	R replace	T back 'till	Y yank line	U undo line	I insert bol	C open above	P paste before	{ begin parag.	}	
q record	W next word	e end word	r replace char	t 'till	y yank ^{1,3}	u undo	i insert de	o open below	p paste after	} end parag.		
A end bol	S last	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd	'. reg. ¹	bol/ goto col	
a append	s last	d delete ^{1,3}	f find char	g extra ⁶	h ←	j ↓	k ↑	l →	; repeat t/T/f/F	' goto mk. bol	\ not used!	
Z quit ⁴	X back-space	C ge ol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un- ³ indent	> indent ³	? find (rev.)			
Z extra ⁵	X delete char	C change ^{1,3}	V visual mode	b prev word	n next (find)	m set mark	, reverse t/T/f/F	· repeat cmd	/ find			

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	Visual mode											
~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down	+ next line
^ goto mark	1 2	3	4	5	6	7	8	9	0 "hard" bol	- prev line	= auto format	
Q ex mode	W next WORD	E end WORD	R replace	T back 'till	Y yank line	U undo line	I insert bol	C open above	P paste before	{ begin parag.	}	
q record	W next word	e end word	r replace char	t 'till	y yank ^{1,3}	u undo	i insert de	c open below	p paste after	misc	end parag.	
A end bol	S last	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd	" reg. ¹	bol/ goto col	
a append	s last	d delete ^{1,3}	f find char	g extra ⁶	h ←	j ↓	k ↑	l →	: line	' spec	not used!	
Z quit ⁴	X back-space	C ge bol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un- ³	> indent ³	? find (rev.)			
Z extra ⁵	X delete char	C change ^{1,3}	V visual mode	b prev word	n next (find)	m set mark	,	,	/ find			

motion moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	visual mode											
~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down	+ next line
^ goto mark	1 2	3 4	5 6	7 8	9 0	"hard" bol	- prev line	= auto format				
Q ex mode	W next WORD	E end WORD	R replace	T back 'till	Y yank line	U undo line	I insert bol	C open above	P paste before	{ begin parag.		
q record	W next word	e end word	r replace char	t 'till	y yank	u undo	i insert de	o open below	p paste after	}	end parag.	
A end bol	S last	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd	" reg. spec	bol/ goto col	
a append	s last	d delete 1,3	f find char	g extra 6	h ←	j ↓	k ↑	l →	; repeat t/T/f/F	' goto mk. bol	\ not used!	
Z quit 4	X back-space	C change 1,3	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un-indent 3	> indent 3	? find (rev.)			
Z extra 5	X delete char	C change 1,3	V visual mode	b prev word	n next (find)	m set mark	,	, reverse t/T/f/F	/ find			

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:
Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	vi / vim graphical cheat sheet													
normal mode														
motion	~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down	+ next line	
operator	~ goto mark	1 2	3 4	5 6	7 8	9 0	"hard" bol	- prev line	= auto format					
command	Q ex mode	W next WORD	E end WORD	R replace	T back 'till	Y yank line	U undo line	I insert bol	C open above	P paste before	{ begin parag.	}	end parag.	
extra	q record macro	W next word	e end word	r replace char	t 'till	y yank	u undo	i insert mode	o open below	p paste after	[misc]	[misc	
operator	A end bol	S last	D delete to eol	F "back" find ch	G eof/ go to ln	H screen ton	J join lines	K help	L screen bottom	: ex cmd	'. reg. spec	bol/ goto col		
extra	a append	s last	d delete 1,3	f find char	g extra cmds	h ←	j ↓	k ↑	l →	; repeat t/T/f/F	' goto mk. bol	\ not used!		
bol = beginning of line, eol = end of line, mk = mark, yank = copy	Z quit 4	X back-space	C geol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< unindent	> indent	? find (rev.)				
words: <code>quux(foo, bar, baz);</code>	Z extra 5	X delete char	C change 1,3	V visual mode	b prev word	n next (find)	m set mark	,	,	/ find				
WORDs: <code>quux (foo, bar, baz);</code>														

motion moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, **eol** = end of line, **mk** = mark, **yank** = copy

words: `quux(foo, bar, baz);`

WORDs: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ



ДВИЖЕНИЯ



ПРИМЕР

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz)`;

WORDS: `quux (foo, bar, baz)`;

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f), :%s/x/y/g (replace 'x' by 'y' filewide), :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

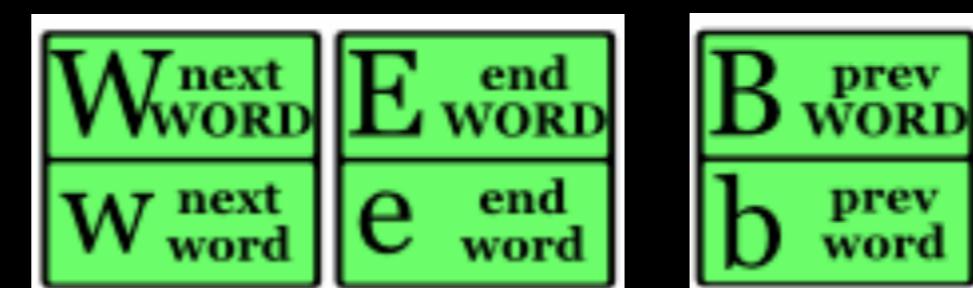
Move around and type operator to act on selected region (vim only)

Notes:

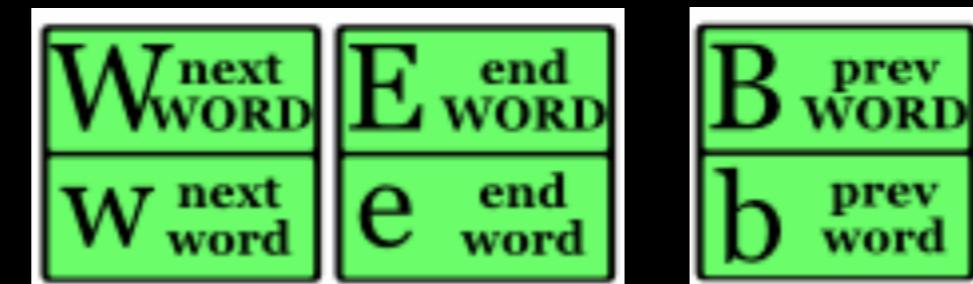
- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*); (e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times (e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top, zb: bottom, zz: center
- (6) gg: top of file (vim only), gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

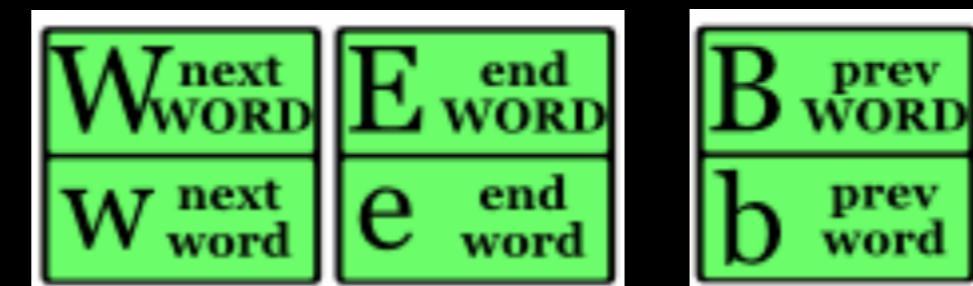


ДВИЖЕНИЯ



WORD? word? wat?

ДВИЖЕНИЯ



WORD? word? wat?

???

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	visual mode									
~ toggle case	! external filter	@ play macro								
^ goto mark	1 2 3 4 5 6 7 8 9 0	"soft" bol repeat :s next ident begin sentence end sentence "soft" bol down prev line next line auto format								
Q ex mode	W next WORD	E end WORD	R place	T back 'till	Y yank line	U undo line	I insert bol	C open above	P paste before	{ begin parag.
q record	W next word	e end word	r replace char	t 'till	y yank	u undo	i insert de	O open below	p paste after	} end parag.
A append	S soft delete	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd line	" reg. spec
a append	s soft	d delete	f find char	g extra cmnds	h	j	k	l	;	' goto mk. bol
Z quit	X back-space	C change	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un indent	> indent	? find (rev.)	\ not used!
Z extra cmd	X delete char	C change	V visual mode	b prev word	n next (find)	m set mark	,	,	/ find	

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
- (2) type in a number before any action to repeat it that number of times (e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top, zb: bottom, zz: center
- (6) gg: top of file (vim only), gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

```
words: quux(foo, bar, baz);  
WORDS: quux (foo, bar, baz);
```

ДВИЖЕНИЯ



words: quux(foo, bar, baz);
WORDS: quux (foo, bar, baz);

ДВИЖЕНИЯ



words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz) ;`

ПРИМЕР

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio"/>

motion moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz)`;

WORDS: `quux (foo, bar, baz)`;

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f), :%s/x/y/g (replace 'x' by 'y' filewide), :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*); (e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times (e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top, zb: bottom, zz: center
- (6) gg: top of file (vim only), gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDs: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

\$ eol

∧ "soft"
bol

0 "hard"
bol

| bol/
goto col

"soft" bol
— down

ДВИЖЕНИЯ

\$ eol	Λ "soft" bol	0 "hard" bol	bol/ goto col	"soft" bol — down
--------	-----------------	-----------------	------------------	----------------------

bol? eol? wat?

ДВИЖЕНИЯ

\$ eol

∧ "soft"
bol

0 "hard"
bol

| bol/
goto col

"soft" bol
— down

bol? eol? wat?

???

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz)`;

WORDS: `quux (foo, bar, baz)`;

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f), :%s/x/y/g (replace 'x' by 'y' filewide), :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*); (e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times (e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top, zb: bottom, zz: center
- (6) gg: top of file (vim only), gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	visual mode														
~	toggle case	!	external filter	@	play macro	#	prev ident	\$	eol	%	goto match	^	"soft" bol	&	repeat :s
'	goto mark	1	²	2		3		4		5		6		7	
Q	ex mode	V	next RD	E	end RD	R	place mode	T	back 'till	Y	yank line	U	undo line	I	insert bol
q	record macro	W	next word	e	end word	r	replace char	t	'till	y	yank ^{1,3}	u	undo	i	insert end
A	end bol	S	last bol	D	delete to eol	F	"back" find ch	G	eof/ goto ln	H	screen top	J	join lines	K	help
a	next word	s	last word	d	delete ^{1,3}	f	find char	g	extra cmds	h	help	j	join	k	screen bottom
Z	quit ⁴	X	back-space	C	ge bol	V	visual lines	B	next RD	N	prev (find)	M	screen mid'l	<	un indent ³
Z	extra cmd ⁵	X	delete char	C	change ^{1,3}	V	visual mode	b	next v	n	next (find)	m	set mark	,	reverse t/T/f/F
															?
															find (rev.)
															/
															find

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

special functions, requires extra input

commands with a dot need a char argument afterwards

bol = beginning of line, **eol** = end of line, **mk** = mark, **yank** = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

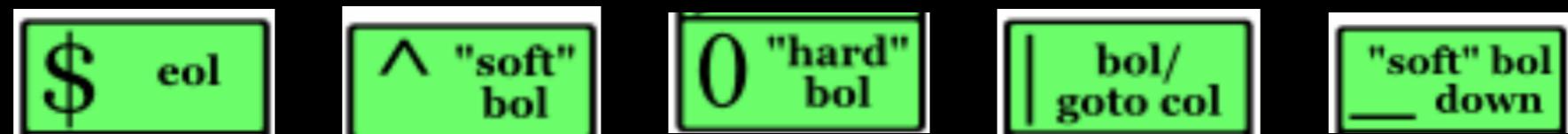
- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

bol = beginning of line, eol = end of line,

ДВИЖЕНИЯ



bol = beginning of line, **eol** = end of line,

ДВИЖЕНИЯ

§ eol ^ "soft" bol 0 "hard" bol | bol/
 goto col "soft" bol
 ____ down

bol = beginning of line, eol = end of line,

ПРИМЕР

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion	moves the cursor, or defines the range for an operator
command	direct action command, if red, it enters insert mode
operator	requires a motion afterwards, operates between cursor & destination
extra	special functions, requires extra input
Q·	commands with a dot need a char argument afterwards
bol = beginning of line, eol = end of line, mk = mark, yank = copy	
words: <code>quux(foo, bar, baz);</code>	
WORDS: <code>quux (foo, bar, baz);</code>	

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	visual mode											
~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ ft	& repeat :s	* next ident	(begin sentence) end sentence	" s eol	+ next line
^ goto mark	1 2	3 4	5	6	7	8	9	0 end	- prev line	= auto format		
Q ex mode	V next RD	E forward	R place	T back 'till	Y yank line	U undo line	I insert bol	C open above	P paste before	{ begin parag.	}	end parag.
q record	W next word	e end	r replace char	t 'till	y yank 1,3	u undo	i insert de	O open below	p paste after	[misc]	misc
A end	S last	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd	" reg. 1	! goto col	
a append	s last	d delete 1,3	f find char	g extra 6	h	j	k	l	: line	' spec	\ not used!	
Z quit 4	X back-space	C eol	V visual lines	B next RD	N prev (find)	M screen mid'l	< unindent 3	> indent 3	? find (rev.)			
Z extra 5	X delete char	C change 1,3	V visual mode	b yd	n next (find)	m set mark	,	, reverse t/T/f/F	/ find			

motion moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ



ДВИЖЕНИЯ

?	• find (rev.)
/	• find

N	prev (find)
n	next (find)

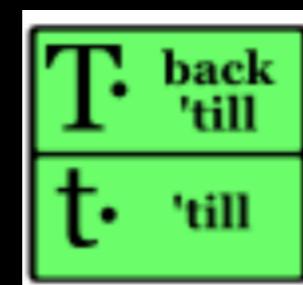
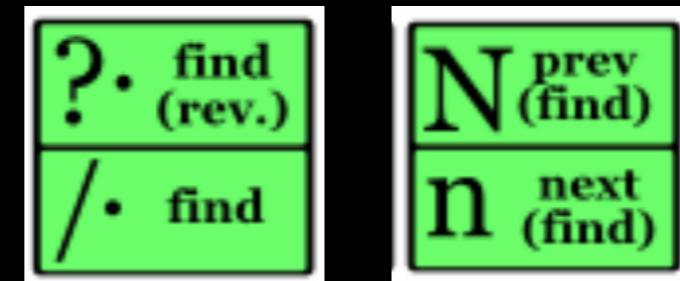
T	back 'till
t	• 'till

F	"back" find ch
f	• find char

,	reverse
	, t/T/f/F

•	repeat
;	, t/T/f/F

ДВИЖЕНИЯ

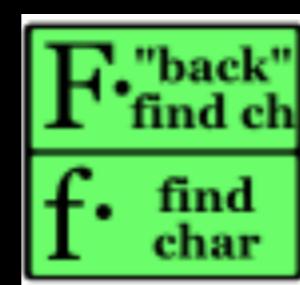
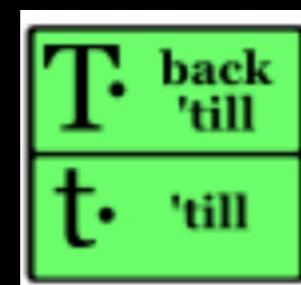
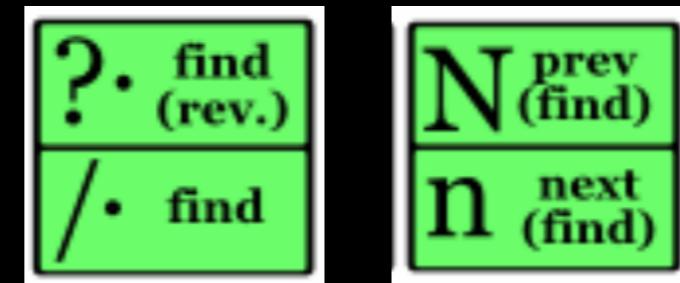


reverse
, t/T/f/F

• repeat
; t/T/f/F

dot? wat?

ДВИЖЕНИЯ



reverse
, t/T/f/F

• repeat
; t/T/f/F

dot? wat?

???

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion	moves the cursor, or defines the range for an operator
command	direct action command, if red, it enters insert mode
operator	requires a motion afterwards, operates between cursor & destination
extra	special functions, requires extra input
Q·	commands with a dot need a char argument afterwards
bol = beginning of line, eol = end of line, mk = mark, yank = copy	
words: <code>quux(foo, bar, baz);</code>	
WORDS: <code>quux (foo, bar, baz);</code>	

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	visual mode											
~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ ft	& repeat :s	* next ident	(begin sentence) end sentence	" s eol	+ next line
^ goto mark	1 2	3 4	5	6	7	8	9	0 end	- prev line	= auto format		
Q ex mode	V next RD	E forward	R place mode	T back 'till	Y yank line	U undo line	I insert bol	C open above	P paste before	{ begin parag.	}	end parag.
q record	W next word	e end	r replace char	t 'till	y yank ^{1,3}	u undo	i insert de	O open below	p paste after	[misc]	misc
A end	S last	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd	" reg. spec	goto col	
a append	s last	d delete ^{1,3}	f find char	g extra ⁶	h	j	k	l	:	! goto mk. bol	\ not used!	
Z quit ⁴	X back-space	C eol	V visual lines	B next RD	N prev (find)	M screen mid'l	< un-indent ³	> indent ³	? find (rev.)			
Z extra ⁵	X delete char	C change ^{1,3}	V visual mode	b yd	n next (find)	m set mark	,	,	/ find			

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDs: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

q. **commands with a dot need
a char argument afterwards**

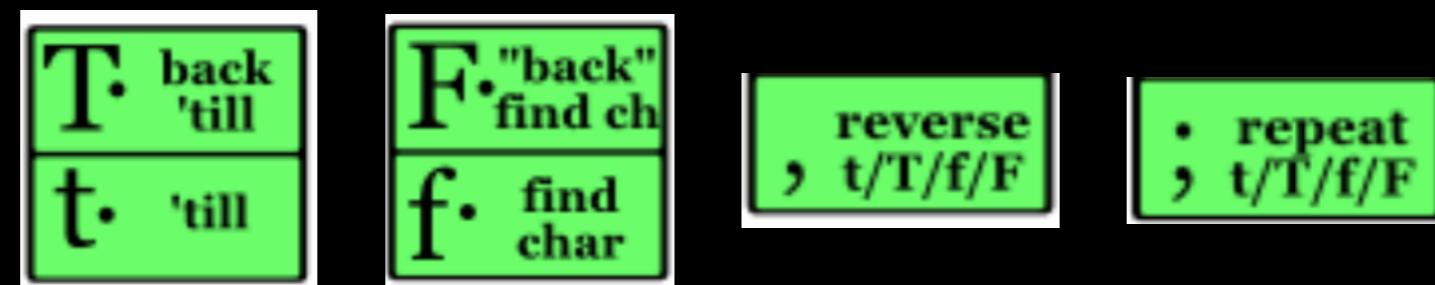
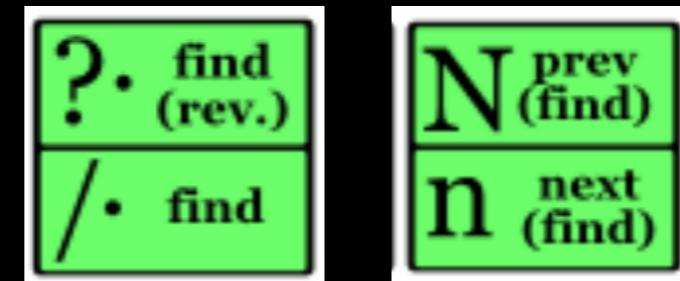
ДВИЖЕНИЯ

?. find (rev.)	N ^{prev} (find)
/• find	n ^{next} (find)

T• back 'till	F• "back" find ch	,	• repeat
t• 'till	f• find char	reverse	;

q• commands with a dot need
a char argument afterwards

ДВИЖЕНИЯ



q• commands with a dot need a char argument afterwards

ПРИМЕР

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	visual mode												
~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ ft	& repeat :s	* next ident	(begin sentence) end sentence	" s eol	- prev line	+ next line
^ goto mark	1 2	3 4	5	6	7	8	9	0 end	0 end	- prev line	= auto format		
Q ex mode	V next RD	E next RD	R place mode	T scroll	Y yank line	U undo line	I insert mode	C open above	P paste before	{ begin parag.	}	end parag.	
q record	W next word	w next word	e next word	r replace char	t scroll	y yank 1,3	u undo	o open below	p paste after	[misc]	misc	
A scroll	S scroll	D delete to eol	F scroll	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd	" reg. 1	! goto col		
a scroll	s scroll	d delete 1,3	f scroll	g extra 6 cmd	h scroll	j scroll	k scroll	l scroll	; line	' spec	\ not used!		
Z quit 4	X back-space	C eol	V visual lines	B next RD	N next RD	M screen mid'l	< unindent 3	> indent 3	? v.				
Z extra 5 cmd	X delete char	C change 1,3	V visual mode	b next w	n next w	m set mark	, use F	* repeat cmd	/ find				

motion moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion	command	operator	extra	visual mode									
F	! external filter	@ play macro	~ toggle case	^ prev ident	\$ eol	% goto match	^ repeat :s	*	next ident	(begin sentence) end sentence	" s eol	+ next line
~ goto mark	1 2	3 4	5	6	7	8	9	0	prev line	= auto format			
Q ex mode	V next RD	E end RD	R place	T scroll	Y yank line	U undo line	I insert	C open above	P paste before	{ begin parag.			
q record	W next word	e end word	r replace char	t scroll	y yank ^{1,3}	u undo	i insert	O open below	p paste after	}	end parag.		
A scroll	S scroll	D delete to eol	F scroll	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	:	• ex cmd	" reg. spec		
a scroll	s scroll	d delete ^{1,3}	f scroll	g extra ⁶	h scroll	j scroll	k scroll	l scroll	;	• line	! goto mk. bol		
Z quit ⁴	X back-space	C eol	V visual lines	B RD	N end	M screen mid'l	< un-indent ³	> indent ³	? scroll (v.)				
Z extra ⁵	X delete char	C change ^{1,3}	V visual mode	b RD	n end	m set mark	,	,	/ scroll				
Q·							,	,					

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

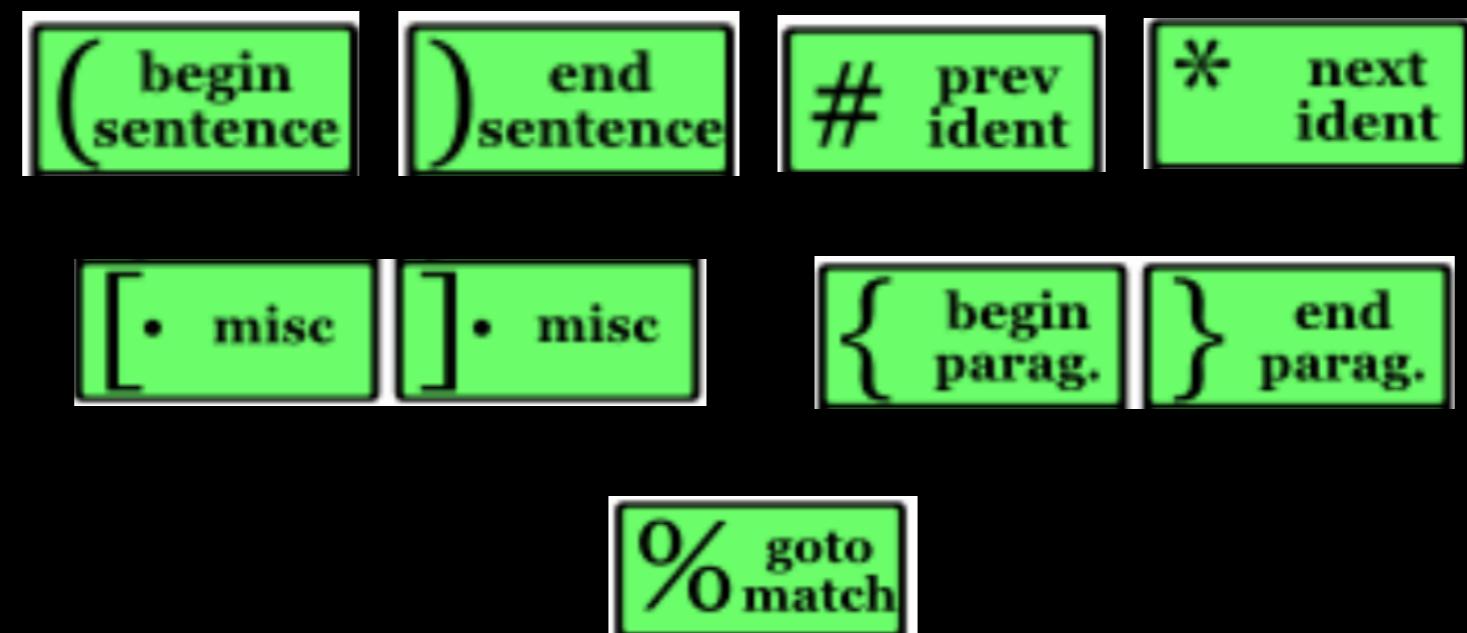
Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ



ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz)`;
WORDS: `quux (foo, bar, baz)`;

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDs: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

H^{screen}
top

M^{screen}
mid'l

L^{screen}
bottom

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDs: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:
Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

ˋ• goto
mark

G eof/
goto ln

!• goto
mk. bol

| bol/
goto col

- prev
line

+ next
line

ДВИЖЕНИЯ

Хочу на 20 строчек вниз!

ДВИЖЕНИЯ

Хочу на 20 строчек вниз!

И что мне? 20 раз жать кнопку???

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz)`;
WORDS: `quux (foo, bar, baz)`;

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz)`;
WORDS: `quux (foo, bar, baz)`;

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:
Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	visual mode
~	toggle case
!	external filter
@	play macro
#	v
\$	eol
0	to start
^	ft"
&	repeat :s
*	next
(start
)	end
"	search
-	diff
+	select
=	auto format
Q	ex mode
q	record
V	next word
W	end word
E	word
R	place
T	fall
y	yank line
U	undo line
I	insert
C	open
P	paste before
p	paste after
{	{edit ag.
}	}edit ag.
A	end
a	append
S	last
s	start
D	delete to eol
d	delete 1,3
F	fall
f	find
G	ln
g	extra cmd
H	open
h	help
J	join lines
j	join
K	help
k	help
L	open
l	open
:	ex cmd
;	line
"	reg. spec
!	!cmd
\	not used!
Z	quit 4
Z	extra 5
X	back-space
X	delete char
C	geol
C	change 1,3
V	visual lines
V	visual mode
B	next word
N	word
M	end'l
m	set mark
<	unindent 3
,	,use F
>	indent 3
?	?d(v.)
/	/r(d)

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDs: `quux (foo, bar, baz);`

Main command line commands ('ex')

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
- (e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times (e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top, zb: bottom, zz: center
- (6) gg: top of file (vim only), gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ДВИЖЕНИЯ

(2) type in a number before any action
to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)

И что, 20 раз жать кнопку?

4

W^{next}
WORD

И что, 20 раз жать кнопку?



И что, 20 раз жать кнопку?

4

W_{next}
WORD

7

k ↑

5

5

5

j ↓

И что, 20 раз жать кнопку?

4

W_{next}
WORD

7

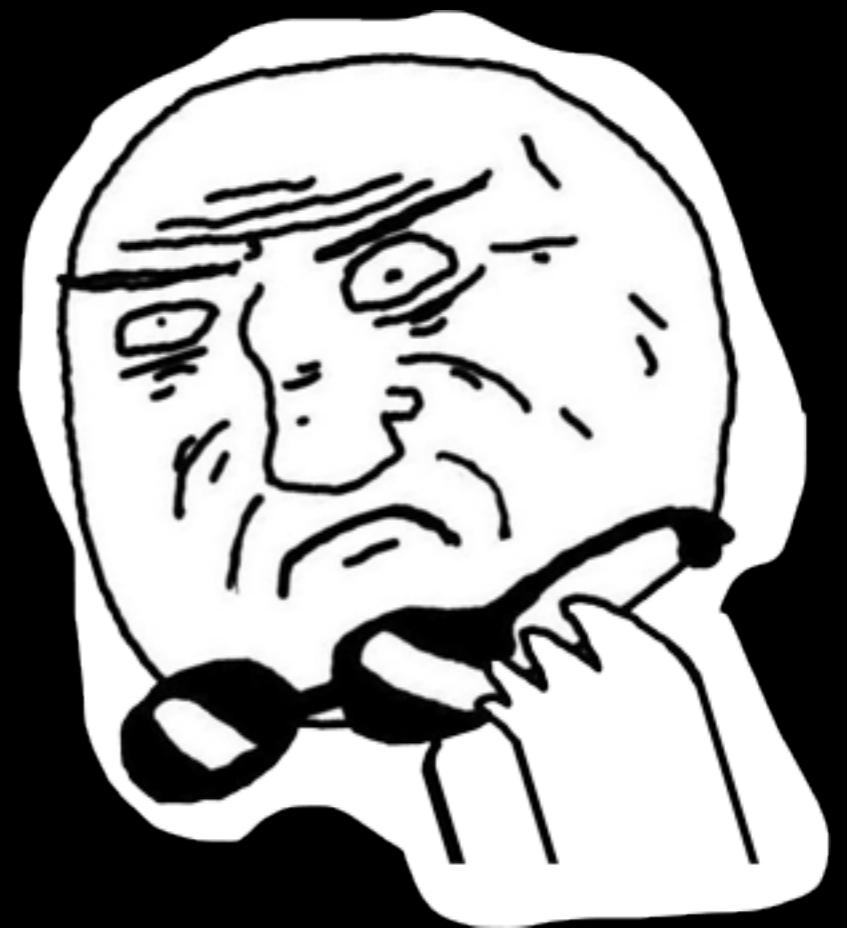
k ↑

5

5

5

j ↓



И что, 20 раз жать кнопку?

4

W_{next}
WORD

7

k ↑

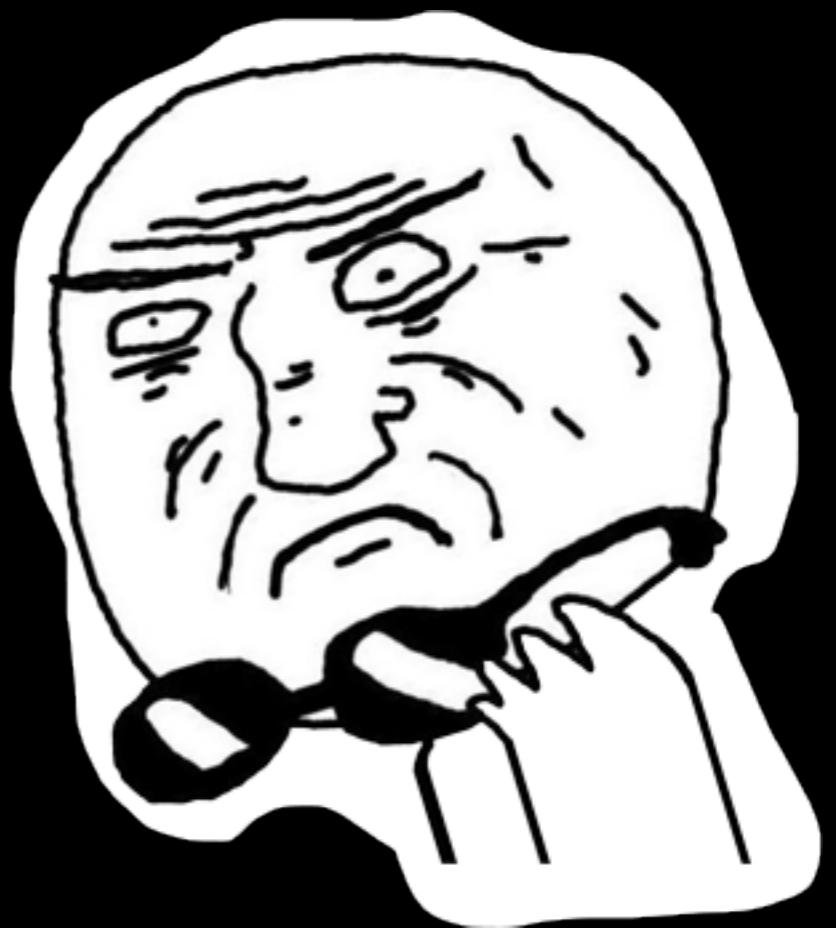
5

5

5

j ↓

ПРИМЕР



version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion	moves the cursor, or defines the range for an operator
command	direct action command, if red , it enters insert mode
operator	requires a motion afterwards, operates between cursor & destination
extra	special functions, requires extra input
Q·	commands with a dot need a char argument afterwards
bol	= beginning of line, eol = end of line, mk = mark, yank = copy
words:	<code>quux(foo, bar, baz);</code>
WORDS:	<code>quux (foo, bar, baz);</code>

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

КОМАНДЫ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	visual mode
~	toggle case
!	external filter
@	play macro
#	print
\$	end of line
0	beginning of line
^	beginning of line
&	repeat :s
*	repeat
(start of line
)	end of line
"	copy
-	delete
"	insert
+	insert
=	auto format
Q	ex mode
q	record
V	next word
W	word
E	end word
R	replace
T	block
Y	yank line
U	undo line
I	insert
C	open
P	paste before
p	paste after
{	begin tag
}	end tag
A	end block
a	end word
S	last
s	last
D	delete to eol
d	delete 1,3
F	block
f	block
G	ln
g	extra 6
H	open
h	extra 1,3
J	join lines
j	extra 1,3
K	help
k	extra 1,3
L	open
l	extra 1,3
:	ex cmd
;	line
"	reg. spec 1
!	! tool
\	not used!
Z	quit 4
Z	extra 5
X	back-space
X	delete char
C	geol
C	change 1,3
V	visual lines
V	visual mode
B	next word
b	prev word
N	next word
n	prev word
M	men' d'l
m	set mark
<	un-3 indent
,	use F
>	indent 3
?	repeat cmd
/	find

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	visual mode
~	toggle case
!	external filter
@	play macro
#	print
\$	end of line
0	beginning of line
^	beginning of line
&	repeat :s
*	repeat :%
(start selection
)	end selection
"	copy
-	delete
"	insert
+	insert
=	auto format
O	ex mode
Q	record macro
V	next word
W	next word
E	end word
R	replace
T	block
y	yank line
Y	yank
U	undo line
u	undo
I	insert
i	insert
C	open file
c	open file
P	paste before
p	paste after
{	{edit ag.
}	{edit ag.
A	append
a	append
S	soft delete
s	soft delete
D	delete to eol
d	delete 1,3
F	block search
f	block search
G	go to ln
g	extra cmd 6
H	open
h	help
J	join lines
j	join lines
K	help
k	help
L	open
l	open
:	ex cmd
;	line
"	reg. spec 1
!	!cmd
\	not used!
Z	quit 4
Z	extra cmd 5
X	back-space
X	delete char
C	copy
C	change 1,3
V	visual lines
V	visual mode
B	next word
b	next word
N	next word
n	next word
M	men' d'l
m	set mark
<	un-3 indent
,	use F
>	indent 3
?	repeat cmd
?	/find

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	visual mode													
~ toggle case	! external filter	@ play macro	# ¹ v	\$ ¹ el	0 ² to match	^ ¹ ft"	& repeat :s	* next	(search) search	" save	' ¹ ed	+ ¹ pt	= auto ³ format
~ w	1 ²	2 [✓]	3 [✓]	4 [✓]	5 [✓]	6 [✓]	7 [✓]	8 [✓]	9 [✓]	0 [✓]	- [✓]	+	=	
Q ex mode	V next word	E word	R place	T scroll	Y yank line	U undo line	I insert	O open	P paste before	{ begin tag	}	end tag		
q record	w next word	e word	r replace char	t scroll	y yank ^{1,3}	u undo	i insert	c delete	p paste after	o open]	[
A end	S last	D delete to eol	F back ch	G ln	H open	J join lines	K help	L open	: ex cmd	" line	! reg. spec	g col		
a append	s last	d delete ^{1,3}	f forward	g extra ⁶	h open	j join	k help	l open	;	!	!	\ not used!		
Z quit ⁴	X back-space	C ¹ go ol	V visual lines	B next	N ¹ d	M ¹ en d'l	< un ³ indent	> indent ³	? find					
Z extra ⁵	X delete char	C change ^{1,3}	V visual mode	b forward	n next	m set mark	,	, use F	repeat cmd	/ find				

motion moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	visual mode
~	toggle case
!	external filter
@	play macro
#	!v
\$	eol
0	to start
^	ft"
&	repeat :s
*	next
(start
)	end
"	service
-	diff
"	diff
+	select
=	auto format
Q	ex mode
q	record
V	next word
W	word
E	end word
R	replace
T	block
Y	yank line
U	undo line
I	insert
C	open
P	paste before
p	paste after
{	{edit ag.
}	}edit ag.
A	end
S	last
D	delete to eol
f	block
G	ln
H	open
J	join lines
K	help
L	open
:	ex cmd
"	line
!"	reg. spec
g	col
Z	quit 4
Z	extra cmd
X	back-space
X	delete char
C	geol
C	change 1,3
V	visual lines
V	visual mode
B	next word
N	prev word
M	men' d'l
<	un-indent 3
>	indent 3
?	repeat cmd
/	find
motion	moves the cursor, or defines the range for an operator
command	direct action command, if red, it enters insert mode
operator	requires a motion afterwards, operates between cursor & destination
extra	special functions, requires extra input
Q·	commands with a dot need a char argument afterwards
bol	= beginning of line, eol = end of line,
mk	= mark, yank = copy
words:	quux(foo, bar, baz);
WORDS:	quux (foo, bar, baz);
Main command line commands ('ex'):	
:w (save), :q (quit), :q! (quit w/o saving)	
:e f (open file f),	
:%s/x/y/g (replace 'x' by 'y' filewide),	
:h (help in vim), :new (new file in vim),	
Other important commands:	
CTRL-R: redo (vim),	
CTRL-F/-B: page up/down,	
CTRL-E/-Y: scroll line up/down,	
CTRL-V: block-visual mode (vim only)	
Visual mode:	
Move around and type operator to act on selected region (vim only)	
Notes:	
(1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)	
(e.g.: "ay\$ to copy rest of line to reg 'a')	
(2) type in a number before any action to repeat it that number of times	
(e.g.: 2p, d2w, 5i, d4j)	
(3) duplicate operator to act on current line (dd = delete line, >> = indent line)	
(4) ZZ to save & quit, ZQ to quit w/o saving	
(5) zt: scroll cursor to top, zb: bottom, zz: center	
(6) gg: top of file (vim only), gf: open file under cursor (vim only)	

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

КОМАНДЫ

r replace
char

D delete
to eol

Y yank
line

P paste
before
p paste¹
after

X back-
space
X delete
char

U undo
line
U undo

КОМАНДЫ

r replace
char

D delete
to eol

Y yank
line

P paste
before
p paste¹
after

X back-
space
X delete
char

U undo
line
U undo

ПРИМЕР

КОМАНДЫ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

КОМАНДЫ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDs: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

КОМАНДЫ

J join
lines

K help

~ toggle
case

• repeat
cmd

Q ex
mode

КОМАНДЫ

m•^{set}
mark

КОМАНДЫ

m• set
mark

• goto
mark !• goto
mk. bol

КОМАНДЫ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	visual mode
~ e	file
! external filter	
@ play macro	
# revert	
\$ eol	
0 go to start	
^ go to end	
& repeat :s	
* repeat :t	
(search	
) search	
" save	
- scroll	
" scroll	
+ scroll	
= auto format	
Q record	q. record
V next word	v. next word
E previous word	e. previous word
R replace	r. replace
T scroll	t. scroll
Y yank	y. yank
U undo	u. undo
I insert	i. insert
C delete	c. delete
P paste	p. paste
{ search	{ search
} search	} search
A append	a. append
S soft delete	s. soft delete
D delete	d. delete
F search	f. search
G go to ln	g. extra cmd
H open	h. extra cmd
J join lines	j. extra cmd
K jump	k. extra cmd
L open com	l. extra cmd
: ex cmd	: line
" reg. spec	" reg. spec
! get col	\ not used!
Z quit	z. quit
X backspace	x. backspace
C change	c. change
V visual lines	v. visual lines
B next word	b. visual mode
N previous word	n. visual mode
M scroll l	m. scroll r
< unindent	, unindent
> indent	> indent
? search	? search
Z extra cmd	z. extra cmd
X delete	x. delete
C change	c. change
V visual mode	v. visual mode
B scroll w	b. scroll w
N scroll b	n. scroll b
M scroll c	m. scroll c
< unindent	, unindent
> indent	> indent
? search	? search

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

КОМАНДЫ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	visual mode
~ e	file
! external filter	
@ play macro	
# revert	
\$ eol	end of line
0 go to start	
^ go to first char	
& repeat :s	substitute
* repeat :t	
(search forward	
) search backward	
" save	
- visual	
+ select	
= auto format	
Q record	quit
V next word	visual word
E previous word	
R replace	replace
T scroll	scroll
Y yank	yank
U undo	undo
I insert	insert
C delete	delete
P paste	paste
{ open block	
} close block	
A append	append
S substitute	substitute
D delete	delete
F find	find
G go to ln	go to line
H help	help
J join lines	join lines
K kill	kill
L open	open
: ex cmd	execute command
'' reg. spec	register specification
\ not used!	not used!
Z quit ⁴	quit
X backspace	backspace
C change	change
V visual lines	visual lines
B next word	next word
N previous word	previous word
M mark	mark
< unindent ³	unindent
> indent ³	indent
? what(.	what(
Z extra cmd ⁵	extra command
X delete	delete
C change ^{1,3}	change
V visual mode	visual mode
b block	block
n next block	next block
m mark block	mark block
,	use F
, repeat	repeat
? what(what(
/ find	find

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

КОМАНДЫ

Q• record
macro @• play
macro

КОМАНДЫ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	visual mode
!e	external filter
dd	delete
1o	1 o ²
2o	2 o
3o	3 o
4o	4 o
5o	5 o
6o	6 o
7o	7 o
8o	8 o
9o	9 o
0o	0 o
-o	- o
+o	+ o
auto	auto ³ format
Q	quit
W	next word
E	end word
R	replace
T	block
Y	yank ^{1,3}
U	undo
I	insert
C	open
P	paste
{	{ selection
}	} selection
A	append
S	last
D	del ^{1,3}
F	block search
G	g/ln
H	open
J	lines
K	open
L	open
:	ex cmd
"	line
reg	reg. ¹
get	get
a	and
s	st
d	delete ^{1,3}
f	find
g.	extra ⁶
h	extra
j	extra
k	extra
l	extra
;	line
!	! not/F
;	! not
\	not used!
Z	quit ⁴
X	cancel
C	change
V	visual lines
B	next word
N	prev word
M	mark
<	un-indent ³
>	indent ³
?	? (v.)
Z	extra ⁵
X	cancel
C	change ^{1,3}
V	visual mode
b	next word
n	prev word
m	mark
,	, use F
,	, use L
/	/ find

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

Почти закончили!

Почти закончили!

Остались только операторы и...

Почти закончили!

Остались только операторы и... ещё два режима!

Почти закончили!

Остались только операторы и... ещё два режима!

ЕЩЁ ДВА РЕЖИМА!!!??!?!?!!?/111!//1/!??!?!??

Почти закончили!

Остались только операторы и... ещё два режима!

ЕЩЁ ДВА РЕЖИМА!!!??!?!?!!??!!?/111



ОПЕРАТОРЫ

ОПЕРАТОРЫ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	visual mode
~	file
!	external filter
a	copy
#	prev
\$	sel
0	to start
^	last
&	repeat :s
*	ext
(start
)	end
"	sel
-	sel
+	sel
auto	format
Q	quit
V	next word
E	end word
R	replace
T	roll
Y	yank
U	undo
I	insert
C	open
P	paste
{	{ open
}	} open
A	end
S	last
D	del
F	find
G	ln
H	open
J	lines
K	open
L	open
:	ex cmd
"	line
!	reg. spec
\	get col
Z	quit ⁴
X	cancel
C	change
V	visual lines
B	next word
N	prev word
M	open d'l
<	un-indent ³
>	indent ³
?	?
Z	extra cmd ⁵
X	cancel
C	change ^{1,3}
V	visual mode
b	next word
n	prev word
m	mark
,	use F
,	use L
;	not used!
/	find

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDs: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*). (e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times (e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top, zb: bottom, zz: center
- (6) gg: top of file (vim only), gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ОПЕРАТОРЫ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	visual mode
~e	file
! external filter	
@y	macro
#n	current
\$el	select
0	to start
^ft"	select
&repeat	:s
*	select
(select
)	select
"s	select
ol	select
+	select
=auto	format
Q	quit
V	next word
E	end word
R	replace
T	kill
Y	yank
U	undo
I	insert
C	change
P	paste
{	open brace
}	close brace
A	append
S	soft delete
D	delete
F	find
G	go to ln
H	open p
J	join lines
K	join p
L	open l
:	ex cmd
"	reg. spec
g	get col
a	append
S	soft d
d	delete
f	find
g.	extra cmds
h	help
j	join
k	join p
l	open l
;	line
!	!cmd
;	reg. spec
\	not used!
Z	quit
X	cancel
C	change
V	visual lines
B	next word
N	prev word
M	mark
<	un indent
>	indent
?	help
Z	extra cmd
X	cancel
C	change
V	visual mode
b	next w
n	prev w
m	mark
,	use F
,	cancel
;	at l
/	find

operator moves the cursor, or defines the range for an operator
operator direct action command, if red, it enters insert mode
operator requires a motion afterwards, operates between cursor & destination
extra special functions, requires extra input
Q commands with a dot need a char argument afterwards
bol = beginning of line, eol = end of line, mk = mark, yank = copy
words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):
:w (save), :q (quit), :q! (quit w/o saving)
:e f (open file f),
:%s/x/y/g (replace 'x' by 'y' filewide),
:h (help in vim), :new (new file in vim),

Other important commands:
CTRL-R: redo (vim),
CTRL-F/-B: page up/down,
CTRL-E/-Y: scroll line up/down,
CTRL-V: block-visual mode (vim only)

Visual mode:
Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ОПЕРАТОРЫ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

operator	action	example
normal mode		
! external filter		
copy	m	ay
macro	a	qanew
repeat	#	5ay
file	\$	\$ay
to	%	%ay
left	^	^ay
right	&	&ay
repeat:	^	^ay
ext	*	*ay
select	((ay
select))ay
line	"	"ay
column	"	"ay
auto format	=	=ay
undo	Q	Qay
redo	V	Vay
word	E	Eay
WORD	W	Way
end word	Q	Qay
end WORD	V	Vay
end word	E	Eay
end WORD	W	Way
place	R	Ray
roll	T	Tay
yank	Y	Yay
do	U	Uay
start	I	Iay
end	C	Cay
move	P	Pay
open	{	{ay
close	}	}ay
append	A	Aay
insert	S	Say
delete	D	Day
extra cmd	F	Fay
ln	G	Gay
open	H	Hay
lines	J	Jay
join	K	Kay
open	L	Lay
com	:	:ay
line	''	''ay
reg. spec	!"	!"ay
get col	g	gay
not used!	\	\ay
quit	Z	Zay
mark	X	Xay
define	C	Cay
visual lines	V	Vay
visual mode	B	Bay
visual mode	N	Nay
block	M	May
un indent	<	<ay
indent	>	>ay
?	?	?ay
extra cmd	Z	Zay
exit	X	Xay
change	C	Cay
visual mode	b	bay
insert	n	nay
block	m	may
use F	,	,ay
use L	,	,ay
not used!	/	/ay

operator moves the cursor, or defines the range for an operator
command direct action command, if red, it enters insert mode
operator requires a motion afterwards, operates between cursor & destination
extra special functions, requires extra input
Q commands with a dot need a char argument afterwards
 bol = beginning of line, eol = end of line, mk = mark, yank = copy
 words: `quux(foo, bar, baz);`
 WORDs: `quux (foo, bar, baz);`

Main command line commands ('ex'):
`:w` (save), `:q` (quit), `:q!` (quit w/o saving)
`:e f` (open file f),
`:%s/x/y/g` (replace 'x' by 'y' filewide),
`:h` (help in vim), `:new` (new file in vim),

Other important commands:
 CTRL-R: redo (vim),
 CTRL-F/-B: page up/down,
 CTRL-E/-Y: scroll line up/down,
 CTRL-V: block-visual mode (vim only)

Visual mode:
 Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ОПЕРАТОРЫ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	visual mode
~	file
!	external filter
a	copy
#	prev
\$	sel
0	to start
^	gt
&	repeat :s
*	ext
(start
)	end
"	sel
-	sel
+	sel
=	auto format
Q	quit
V	next word
E	end word
R	replace
T	roll
Y	yank
U	undo
I	insert
C	open
P	paste
{	{ open
}	} open
A	end
S	last
D	del
F	find
G	ln
H	open
J	lines
K	open
L	open
:	ex cmd
"	line
!	reg. spec
\	get
Z	quit ⁴
X	cancel
C	change
V	visual lines
B	next word
N	prev word
M	open &l
<	un-indent ³
>	indent ³
?	? (v.)
Z	extra cmd ⁵
X	cancel
C	change ^{1,3}
V	visual mode
b	prev word
n	next word
m	mark
,	, use F
,	use L
;	repeat
/	find

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q. commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ОПЕРАТОРЫ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	visual mode
~	file
!	external filter
a	copy
#	prev
\$	sel
0	to start
^	last
&	repeat :s
*	ext
(select
)	select
"	sel
-	sel
"	sel
+	sel
=	auto format
Q	end
V	next word
E	end word
R	replace
T	block
Y	yank
U	undo
I	insert
C	open
P	move
{	open {
}	open }
A	end
S	last
D	del
F	block
G	ln
H	open
J	lines
K	end
L	open
:	ex cmd
"	line
!	reg. spec
\	get
Z	quit
X	cancel
C	change
V	visual lines
B	next word
N	prev word
M	open &l
<	un indent
>	indent
?	?
Z	extra cmd
X	cancel
C	change
V	visual mode
b	next word
n	prev word
m	mark
,	use F
,	use L
;	at
/	find

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q. commands with a dot need a char argument afterwards

bol = beginning of line, **eol** = end of line, **mk** = mark, **yank** = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ОПЕРАТОРЫ

c change^{1,3}

d delete^{1,3}

y yank^{1,3}

ОПЕРАТОРЫ

Операторы позволяют делать “составные команды”.

ОПЕРАТОРЫ

Операторы позволяют делать “составные команды”.

Нам не обязательно привязываться к действию, мы можем составлять цепочки из операторов, количества повторений и команд.

ОПЕРАТОРЫ

Операторы позволяют делать “составные команды”.

Нам не обязательно привязываться к действию, мы можем составлять цепочки из операторов, количества повторений и команд.



ОПЕРАТОРЫ

Операторы позволяют делать “составные команды”.

Нам не обязательно привязываться к действию, мы можем составлять цепочки из операторов, количества повторений и команд.

d delete
1,3

5

j ↓

y yank
1,3

4

W next WORD

ОПЕРАТОРЫ

Операторы позволяют делать “составные команды”.

Нам не обязательно привязываться к действию, мы можем составлять цепочки из операторов, количества повторений и команд.

d delete^{1,3}

5

j ↓

y yank^{1,3}

4

W next WORD

c change^{1,3}

9

l →

ОПЕРАТОРЫ

Операторы позволяют делать “составные команды”.

Нам не обязательно привязываться к действию, мы можем составлять цепочки из операторов, количества повторений и команд.

d delete
1,3

5

j ↓

y yank
1,3

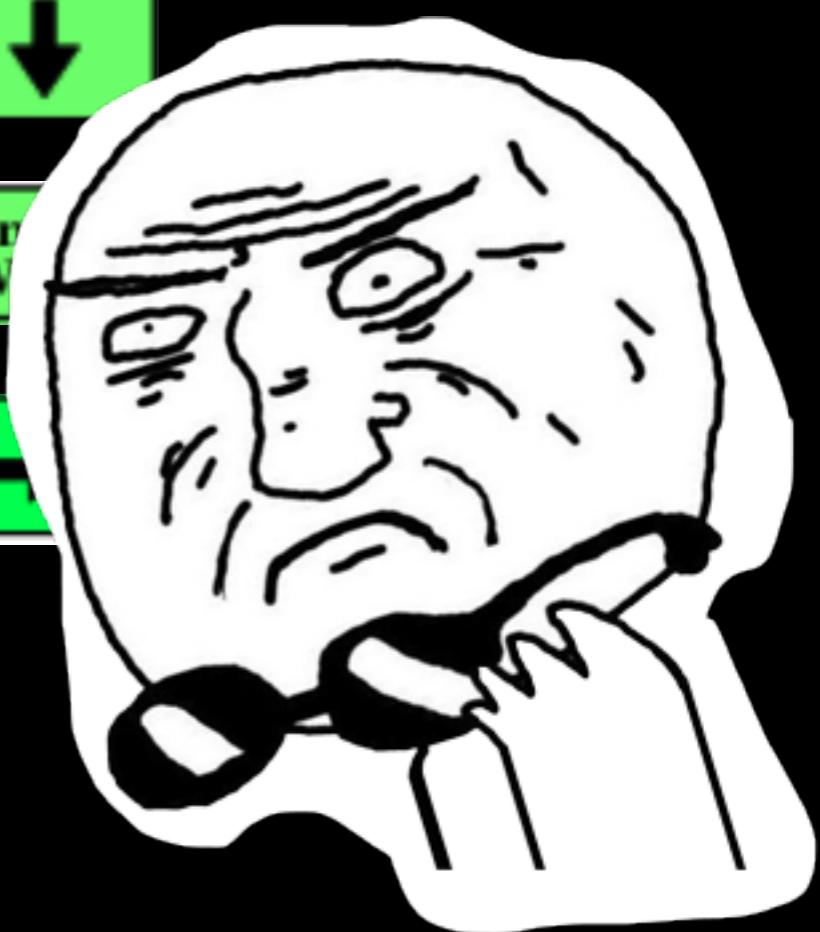
4

Ww

c change
1,3

9

l -



ОПЕРАТОРЫ

Операторы позволяют делать “составные команды”.

Нам не обязательно привязываться к действию, мы можем составлять цепочки из операторов, количества повторений и команд.

d delete
1,3

5

j ↓

y yank
1,3

4

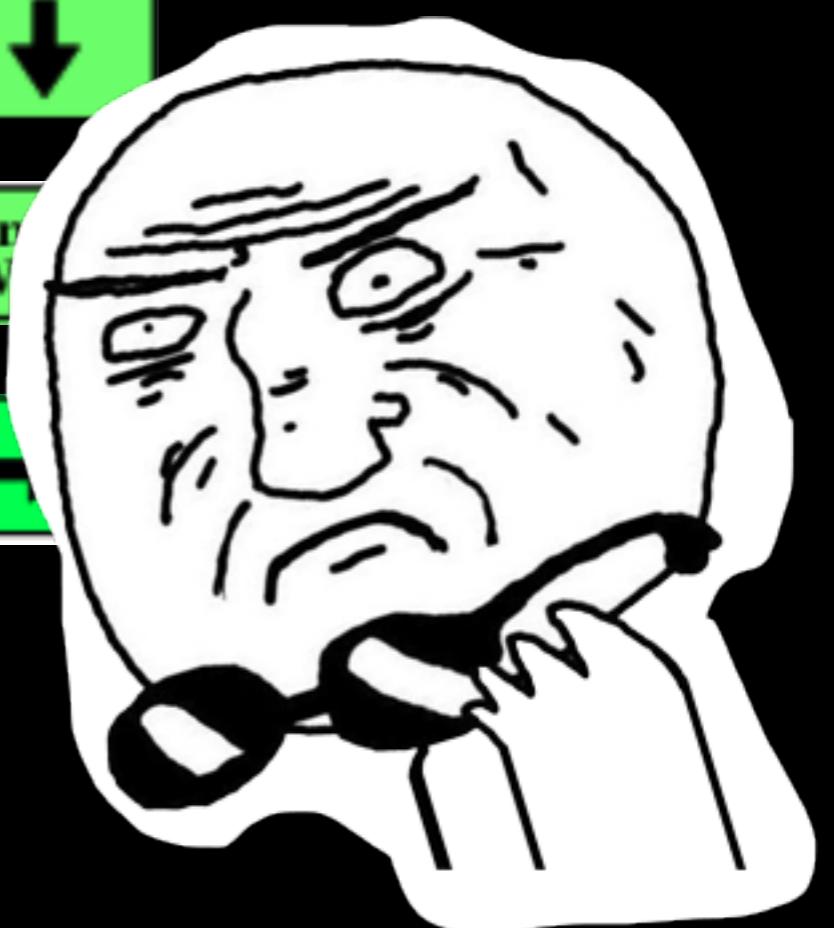
Ww

c change
1,3

9

l -

ПРИМЕР



ОПЕРАТОРЫ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	visual mode
~	file
!	external filter
a	copy
#	prev
\$	sel
0	to start
^	gt
&	repeat :s
*	ext
(start
)	end
"	sel
s	sel
+	gt
=	auto format
Q	end
V	next word
E	end word
R	replace
T	gt
Y	gt
U	undo
I	insert
C	open
P	move
{	open {
}	open }
A	end
S	last
D	gt
F	gt
G	ln
H	open
J	lines
K	gt
L	open
:	ex cmd
"	line
!	reg. spec
\	get
Z	quit
X	cancel
C	gt
V	visual lines
B	next word
N	gt
M	gt
<	unindent
>	indent
?	gt
Z	extra cmd
X	cancel
C	gt
V	visual mode
b	gt
n	gt
m	gt
,	use F
,	gt
;	gt
/	gt

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q. commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ОПЕРАТОРЫ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	visual mode
~	e
!	external filter
@	copy
#	prev
\$	sel
0	to start
^	gt
&	repeat :s
*	ext
(start of line
)	end of line
"	sel
-	sel
=	auto format
Q	quit
V	next word
E	end word
R	replace
T	roll
Y	yank
U	undo
I	insert
C	open
P	paste
{	{ open
}	} open
A	end col
S	last col
D	del
F	find
G	ln
H	open
J	lines
K	open
L	open
:	ex cmd
"	line
!	reg. spec
\	get col
Z	quit ⁴
X	cancel
C	copy
V	visual lines
B	next word
N	prev word
M	open & l
<	un indent ³
>	indent ³
?	?
Z	extra cmd ⁵
X	cancel
C	copy ³
V	visual mode
b	prev word
n	next word
m	mark
,	use F
.	repeat
/	find

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q. commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

ОПЕРАТОРЫ

! external filter < un-indent³ > indent³ = auto³ format

ОПЕРАТОРЫ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

Category	Description
motion	moves the cursor, or defines the range for an operator
command	direct action command, if red , it enters insert mode
operator	requires a motion afterwards, operates between cursor & destination
extra	special functions, requires extra input
Q.	commands with a dot need a char argument afterwards
bol	beginning of line, eol = end of line,
mk	mark, yank = copy
words:	quux(foo, bar, baz);
WORDS:	quux (foo, bar, baz);

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

Что осталось?

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDs: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q. commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDs: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- L-P: redo (vim),
- F/-B: page up/down,
- E/-Y: scroll line up/down,
- C-V: block-visual mode (vim only)

Visual mode:
Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q. commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`

WORDs: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Important commands:

- CMD-Z: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

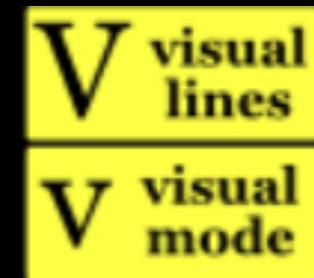
Visual mode:
Move around and type operator to act on selected region (vim only)

Notes:

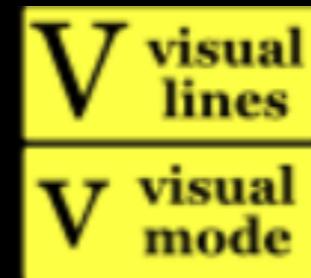
- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

РЕЖИМ "ВЫДЕЛЕНИЕ" И "СТРОКОВОЕ ВЫДЕЛЕНИЕ" И "БЛОЧНОЕ ВЫДЕЛЕНИЕ"

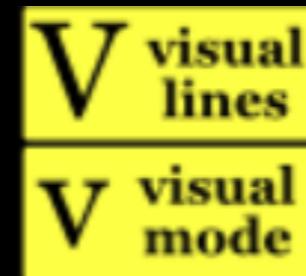


РЕЖИМ "ВЫДЕЛЕНИЕ" И "СТРОКОВОЕ ВЫДЕЛЕНИЕ" И "БЛОЧНОЕ ВЫДЕЛЕНИЕ"



Visual mode:
**Move around and type operator to act
on selected region (vim only)**

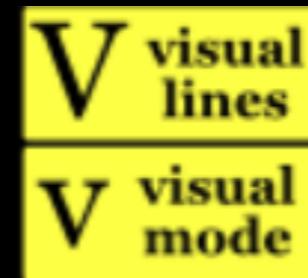
РЕЖИМ “ВЫДЕЛЕНИЕ” И “СТРОКОВОЕ ВЫДЕЛЕНИЕ” И “БЛОЧНОЕ ВЫДЕЛЕНИЕ”



Visual mode:
**Move around and type operator to act
on selected region (vim only)**

CTRL-V: block-visual mode (vim only)

РЕЖИМ "ВЫДЕЛЕНИЕ" И "СТРОКОВОЕ ВЫДЕЛЕНИЕ" И "БЛОЧНОЕ ВЫДЕЛЕНИЕ"



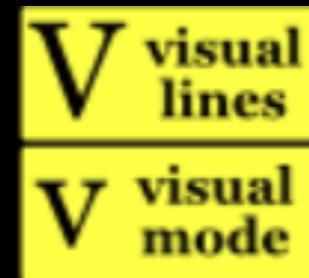
Visual mode:

**Move around and type operator to act
on selected region (vim only)**

CTRL-V: block-visual mode (vim only)

Служат для выделения текста и дальнейшего его изменения.

РЕЖИМ "ВЫДЕЛЕНИЕ" И "СТРОКОВОЕ ВЫДЕЛЕНИЕ" И "БЛОЧНОЕ ВЫДЕЛЕНИЕ"



Visual mode:

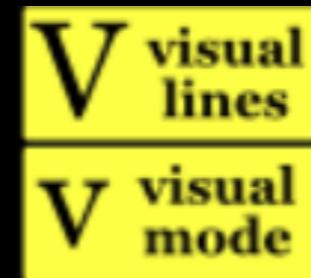
**Move around and type operator to act
on selected region (vim only)**

CTRL-V: block-visual mode (vim only)

Служат для выделения текста и дальнейшего его изменения.

Строковое и блочное выделение - мощнейшие инструменты Vim

РЕЖИМ "ВЫДЕЛЕНИЕ" И "СТРОКОВОЕ ВЫДЕЛЕНИЕ" И "БЛОЧНОЕ ВЫДЕЛЕНИЕ"



Visual mode:

**Move around and type operator to act
on selected region (vim only)**

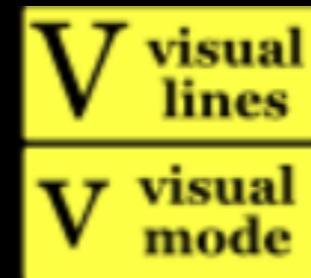
CTRL-V: block-visual mode (vim only)

Служат для выделения текста и дальнейшего его изменения.

Строковое и блочное выделение - мощнейшие инструменты Vim

Слишком обширная тема...

РЕЖИМ "ВЫДЕЛЕНИЕ" И "СТРОКОВОЕ ВЫДЕЛЕНИЕ" И "БЛОЧНОЕ ВЫДЕЛЕНИЕ"



Visual mode:
**Move around and type operator to act
on selected region (vim only)**

CTRL-V: block-visual mode (vim only)

Служат для выделения текста и дальнейшего его изменения.

Строковое и блочное выделение - мощнейшие инструменты Vim

Слишком обширная тема...

ПРИМЕР

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

Legend:

- motion**: moves the cursor, or defines the range for an operator
- command**: direct action command, if red, it enters insert mode
- operator**: requires a motion afterwards, operates between cursor & destination
- extra**: special functions, requires extra input
- Q.**: commands with a dot need a char argument afterwards

bol = beginning of line, **eol** = end of line, **mk** = mark, **yank** = copy

words: `quux(foo, bar, baz);`
WORDs: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- `:w` (save), `:q` (quit), `:q!` (quit w/o saving)
- `:e f` (open file f),
- `:%s/x/y/g` (replace 'x' by 'y' filewide),
- `:h` (help in vim), `:new` (new file in vim),

Other important commands:

- `CTRL-R`: redo (vim),
- `CTRL-F/-B`: page up/down,
- `CTRL-E/-Y`: scroll line up/down,
- `CTRL-V`: block-visual mode (vim only)

Visual mode:
Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') ($x=a..z,*$)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: `2p`, `d2w`, `5i`, `d4j`)
- (3) duplicate operator to act on current line
(`dd` = delete line, `>>` = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) `zt`: scroll cursor to top,
`zb`: bottom, `zz`: center
- (6) `gg`: top of file (vim only),
`gf`: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

Legend:

- motion**: moves the cursor, or defines the range for an operator
- command**: direct action command, if **red**, it enters insert mode
- operator**: requires a motion afterwards, operates between cursor & destination
- extra**: special functions, requires extra input
- Q.**: commands with a dot need a char argument afterwards

bol = beginning of line, **eol** = end of line, **mk** = mark, **yank** = copy

words: `quux(foo, bar, baz);`
WORDs: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- `:w` (save), `:q` (quit), `:q!` (quit w/o saving)
- `:e f` (open file f),
- `:%s/x/y/g` (replace 'x' by 'y' filewide),
- `:h` (help in vim), `:new` (new file in vim),

Other important commands:

- CTRL-R**: redo (vim),
- CTRL-F/-B**: page up/down,
- CTRL-E/-Y**: scroll line up/down,
- CTRL-V**: block-visual mode (vim only)

Visual mode:
Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) **zt**: scroll cursor to top,
zb: bottom, **zz**: center
- (6) **gg**: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

- ex cmd
- line

РЕЖИМ КОМАНДНОЙ СТРОКИ

РЕЖИМ КОМАНДНОЙ СТРОКИ

Предназначен для выполнения сложных составных команд.

Например: поиск по регулярному выражению с заменой.

РЕЖИМ КОМАНДНОЙ СТРОКИ

Предназначен для выполнения сложных составных команд.

Например: поиск по регулярному выражению с заменой.

ПРИМЕР

РЕЖИМ КОМАНДНОЙ СТРОКИ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	Visual mode													
!e	!w	!y	!d	#n	\$c	0x	^ft	&repeat	*x	(s)s	"s	+x	
o	o	o	o	o	o	o	o	:s	:s	(s)s	"s	+x	
1	2	3	4	5	6	7	8	9	0	-	-	=	=	
Q	V	E	R	T	Y	U	I	C	P	{	}	{	}	
q	w	w	w	t	y	u	i	c	p	
A	S	D	F	G	H	J	K	L	:	;"	".	g	not used!	
a	s	d	f	g	h	j	k	l	;	;"	".	g	not used!	
Z quit ⁴	X ack-ee	C ome	V isual	B RD	N ew	M open	< repeat ³	> repeat ³	? what(v.)					
Z extra cmd ⁵	X erase	C on	V isual	b RD	n ext	m ove	, use F	* what	/ find					

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q. commands with a dot need a char argument afterwards

bol = beginning of line, **eol** = end of line, **mk** = mark, **yank** = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- `:w` (save), `:q` (quit), `:q!` (quit w/o saving)
- `:e f` (open file f),
- `:%s/x/y/g` (replace 'x' by 'y' filewide),
- `:h` (help in vim), `:new` (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

РЕЖИМ КОМАНДНОЙ СТРОКИ

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	Visual mode													
!e	!w	!r	!y	#n	\$c	0x	^ft	&repeat	*x	(s)s	"s	+x	
o	o	o	o	o	o	o	o	;	x	s	s	w	=x	
Q	W	E	R	T	Y	U	I	C	P	{	}	p]	
q	q	q	q	t	y	u	i	c	p	o	o	p]	
A	S	D	F	G	H	J	K	L	:	;"	;"	;"	\	
a	s	d	f	g	h	j	k	l	;	!	!	!	not used!	
Z quit ⁴	X ack-ee	C ome	V issal	B RD	N ed	M open	< repeat ³	> repeat ³	? what(v.)					
Z extra cmd ⁵	X erate	C on	V isal	b	n	m	,	,	/ read					

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q. commands with a dot need a char argument afterwards

bol = beginning of line, **eol** = end of line, **mk** = mark, **yank** = copy

words: `quux(foo, bar, baz);`
WORDS: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- `:w` (save), `:q` (quit), `:q!` (quit w/o saving)
- `:e f` (open file f),
- `:%s/x/y/g` (replace 'x' by 'y' filewide),
- `:h` (help in vim), `:new` (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:
Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

РЕЖИМ КОМАНДНОЙ СТРОКИ

&^{repeat}
:s

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

Symbol	Command	Description
green circle with checkmark	normal mode	Enters normal mode.
green circle with checkmark	!e[dit] file	Edits the current file.
orange circle with checkmark	!w[rite]	Saves the current file.
green circle with checkmark	@[macro]	Runs a macro.
green circle with checkmark	#[n]erv	Switches to the nth previous buffer.
green circle with checkmark	\$[e]nd	Moves the cursor to the end of the line.
green circle with checkmark	0[ro]t[e] to [char]	Rotates the screen to the specified character.
green circle with checkmark	^[f]it"	Zooms in on the current line.
green circle with checkmark	&[reat]	Creates a new window.
green circle with checkmark	*[reat]	Creates a new buffer.
green circle with checkmark	([se)n[e]ce	Starts a new session.
green circle with checkmark)[se)n[e]ce	Ends a session.
green circle with checkmark	"[s]ub[stitut[e]	Substitutes text.
green circle with checkmark	+[re]p[ate]	Repeats the last command.
green circle with checkmark	~[o]r[n]	Switches to the previous buffer.
green circle with checkmark	1[0]	Switches to the first buffer.
green circle with checkmark	2[1]	Switches to the second buffer.
green circle with checkmark	3[2]	Switches to the third buffer.
green circle with checkmark	4[3]	Switches to the fourth buffer.
green circle with checkmark	5[4]	Switches to the fifth buffer.
green circle with checkmark	6[5]	Switches to the sixth buffer.
green circle with checkmark	7[6]	Switches to the seventh buffer.
green circle with checkmark	8[7]	Switches to the eighth buffer.
green circle with checkmark	9[8]	Switches to the ninth buffer.
green circle with checkmark	0[0]	Switches to the tenth buffer.
green circle with checkmark	-[w]	Closes the current buffer.
green circle with checkmark	=[o]r[n]	Switches to the next buffer.
green circle with checkmark	Q[uit]	Quits the editor.
green circle with checkmark	V[isit] RD	Visits a remote directory.
green circle with checkmark	E[dit] RD	Edits a remote directory.
green circle with checkmark	R[eset] RD	Resets a remote directory.
green circle with checkmark	T[ime] RD	Shows the time in a remote directory.
yellow circle with checkmark	Y[ank]	Yanks the current line.
yellow circle with checkmark	U[ndo]	Undoes the last command.
yellow circle with checkmark	I[nsert]	Enters insert mode.
yellow circle with checkmark	C[on]tinue	Continues the editor.
yellow circle with checkmark	P[aste]	Pastes the clipboard.
yellow circle with checkmark	{[c]l]ip[g.]	Shows the clipboard.
yellow circle with checkmark	Q[uote]	Shows the quote register.
yellow circle with checkmark	W[ord]	Searches for a word.
yellow circle with checkmark	e[sc]ape	Exits the editor.
yellow circle with checkmark	Q[uote]	Shows the quote register.
yellow circle with checkmark	W[ord]	Searches for a word.
yellow circle with checkmark	e[sc]ape	Exits the editor.
yellow circle with checkmark	Q[uote]	Shows the quote register.
yellow circle with checkmark	W[ord]	Searches for a word.
yellow circle with checkmark	e[sc]ape	Exits the editor.
yellow circle with checkmark	Q[uote]	Shows the quote register.
yellow circle with checkmark	W[ord]	Searches for a word.
yellow circle with checkmark	e[sc]ape	Exits the editor.
yellow circle with checkmark	A[dd]	Adds text.
yellow circle with checkmark	S[ub]st	Substitutes text.
yellow circle with checkmark	D[elete]	Deletes text.
yellow circle with checkmark	F[ind]	Finds text.
yellow circle with checkmark	G[o] to ln	Go to line number.
yellow circle with checkmark	H[elp]	Shows help.
yellow circle with checkmark	J[oin] lines	Joins lines.
yellow circle with checkmark	K[ill]	Kills text.
yellow circle with checkmark	L[ist]	Lists files.
yellow circle with checkmark	:[v]isual	Enters visual mode.
yellow circle with checkmark	":[v]isual	Enters visual mode.
yellow circle with checkmark	":[v]isual	Enters visual mode.
yellow circle with checkmark	":[v]isual	Enters visual mode.
yellow circle with checkmark	":[v]isual	Enters visual mode.
yellow circle with checkmark	":[v]isual	Enters visual mode.
yellow circle with checkmark	":[v]isual	Enters visual mode.
yellow circle with checkmark	":[v]isual	Enters visual mode.
yellow circle with checkmark	":[v]isual	Enters visual mode.
yellow circle with checkmark	Z[quit] 4	Quits the editor.
yellow circle with checkmark	X[ack]ee	Shows ack command.
yellow circle with checkmark	C[on]tinue	Continues the editor.
yellow circle with checkmark	V[isual] es	Enters visual mode.
yellow circle with checkmark	B[lock] RD	Blocks a range.
yellow circle with checkmark	N[ew]	Creates a new file.
yellow circle with checkmark	M[ark]	Marks the current line.
orange circle with checkmark	<[w]it	Waits for input.
orange circle with checkmark	>[w]it	Waits for input.
orange circle with checkmark	?[w]it	Waits for input.
green circle with checkmark	Z[quit] 5	Quits the editor.
green circle with checkmark	X[ack]te	Shows ack command.
green circle with checkmark	C[on]tinue	Continues the editor.
green circle with checkmark	V[isual] le	Enters visual mode.
green circle with checkmark	b[lock]	Blocks a range.
green circle with checkmark	n[ext]	Next line.
green circle with checkmark	m[ark]	Marks the current line.
green circle with checkmark	,	Comma separator.
green circle with checkmark	,[w]it F	Waits for input.
green circle with checkmark	,[w]it	Waits for input.
green circle with checkmark	*,[w]it	Waits for input.
green circle with checkmark	?,[w]it	Waits for input.

motion moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q. commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz);`
WORDs: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- duplicate operator to act on current line (dd = delete line, >> = indent line)
- ZZ to save & quit, ZQ to quit w/o saving
- zt: scroll cursor to top,
zb: bottom, zz: center
- gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

Специальные команды

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

Legend:

- motion**: moves the cursor, or defines the range for an operator
- command**: direct action command, if **red**, it enters insert mode
- operator**: requires a motion afterwards, operates between cursor & destination
- extra**: special functions, requires extra input
- Q.**: commands with a dot need a char argument afterwards

bol = beginning of line, **eol** = end of line, **mk** = mark, **yank** = copy

words: `quux(foo, bar, baz);`
WORDs: `quux (foo, bar, baz);`

Main command line commands ('ex'):

- `:w` (save), `:q` (quit), `:q!` (quit w/o saving)
- `:e f` (open file f),
- `:%s/x/y/g` (replace 'x' by 'y' filewide),
- `:h` (help in vim), `:new` (new file in vim),

Other important commands:

- CTRL-R**: redo (vim),
- CTRL-F/-B**: page up/down,
- CTRL-E/-Y**: scroll line up/down,
- CTRL-V**: block-visual mode (vim only)

Visual mode:
Move around and type operator to act on selected region (vim only)

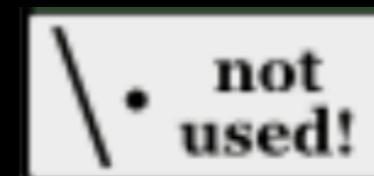
Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) **zt**: scroll cursor to top,
zb: bottom, **zz**: center
- (6) **gg**: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

Z• quit ⁴	!!• reg. ¹ spec
Z• extra ⁵ cmds	g• extra ⁶ cmds

* not
used!



NOT USED!!!

\ • not
used!

NOT USED!!!



version 1.1
April 1st, 06

vi / vim graphical cheat sheet

F	Visual mode
	m ^{otion}
	c ^{ommand}
	o ^{perator}
	e ^{xtra}
Q.	commands with a dot need a char argument afterwards
bol = beginning of line, eol = end of line, mk = mark, yank = copy	
words: <code>quux(foo, bar, baz);</code>	
WORDs: <code>quux (foo, bar, baz);</code>	

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-F/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

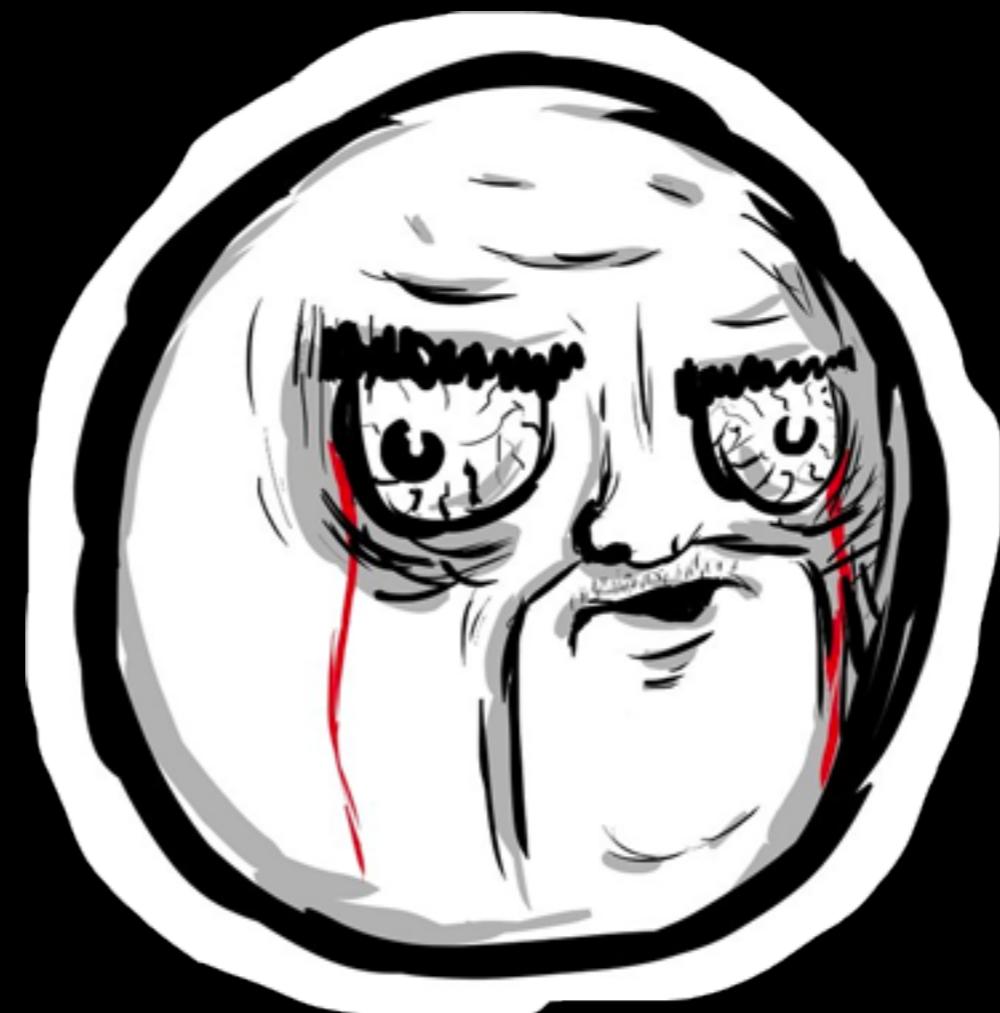
Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*)
(e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times
(e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line
(dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top,
zb: bottom, zz: center
- (6) gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

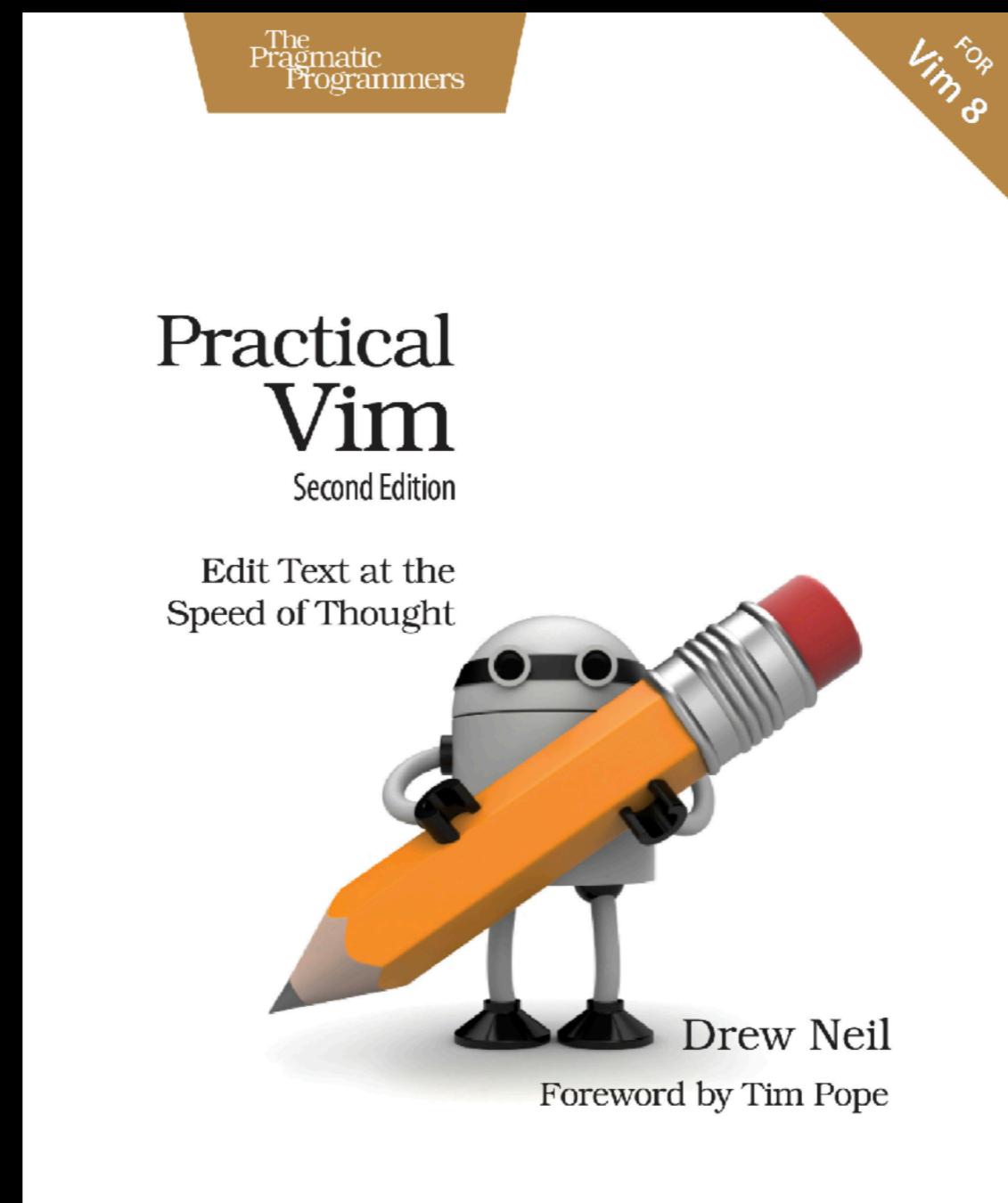
КТО ХОТЬ ЧТО-ТО ЗАПОМНИЛ?

КТО ХОТЬ ЧТО-ТО ЗАПОМНИЛ?



Думать больше не нужно!

Думать больше не нужно!

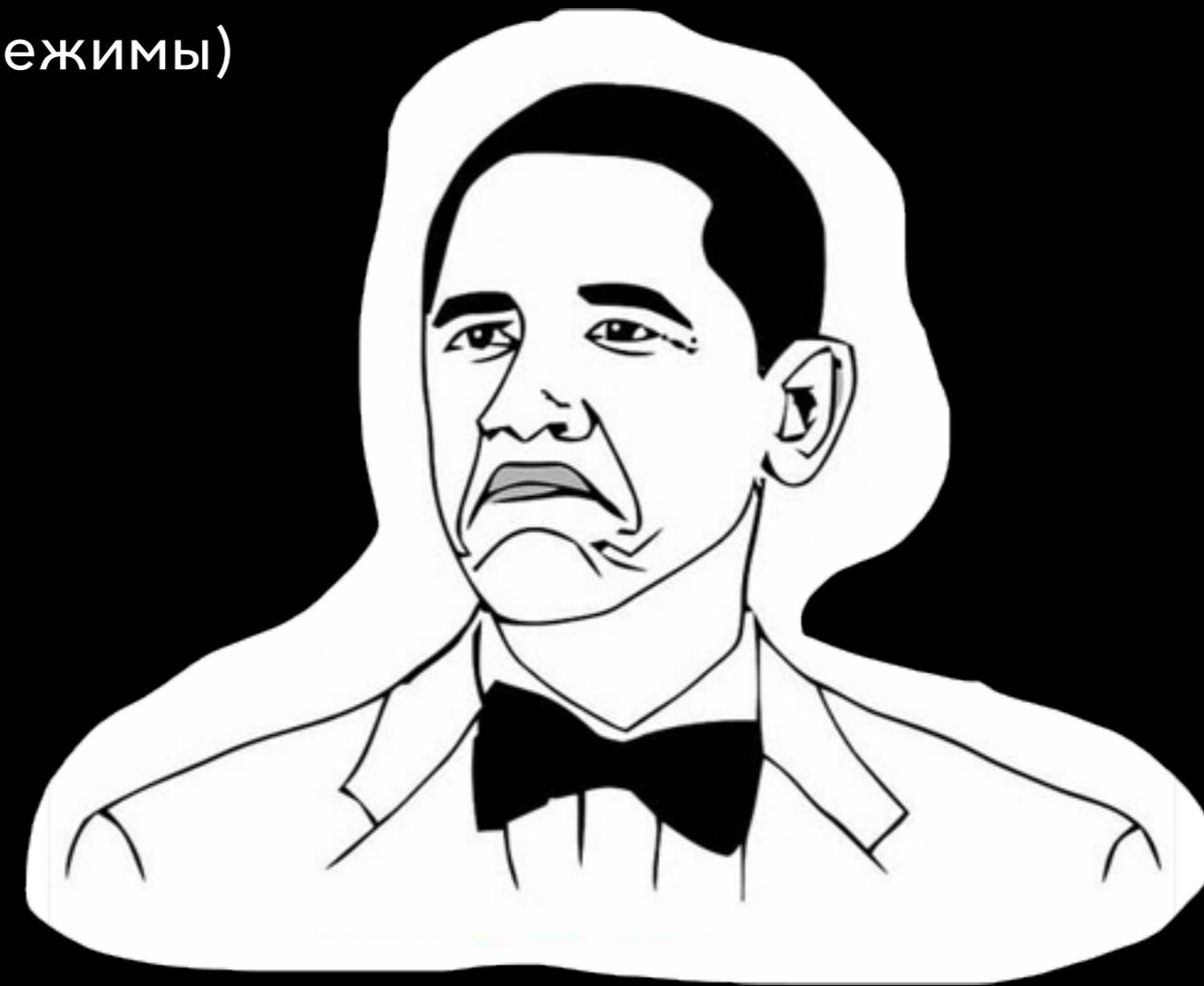


БЕНЕФИТЫ:

- 1) Терминал
- 2) Терминал ;D (ресурсы и скорость)
- 3) Кастомизация
- 3) Модальность (режимы)

БЕНЕФИТЫ:

- 1) Терминал
- 2) Терминал ;D (ресурсы и скорость)
- 3) Кастомизация
- 3) Модальность (режимы)



Но зачем мне Vim?

Но зачем мне Vim?

Ведь у меня есть \${ваша_любимая_идея} и она умеет...

БЕНЕФИТЫ:

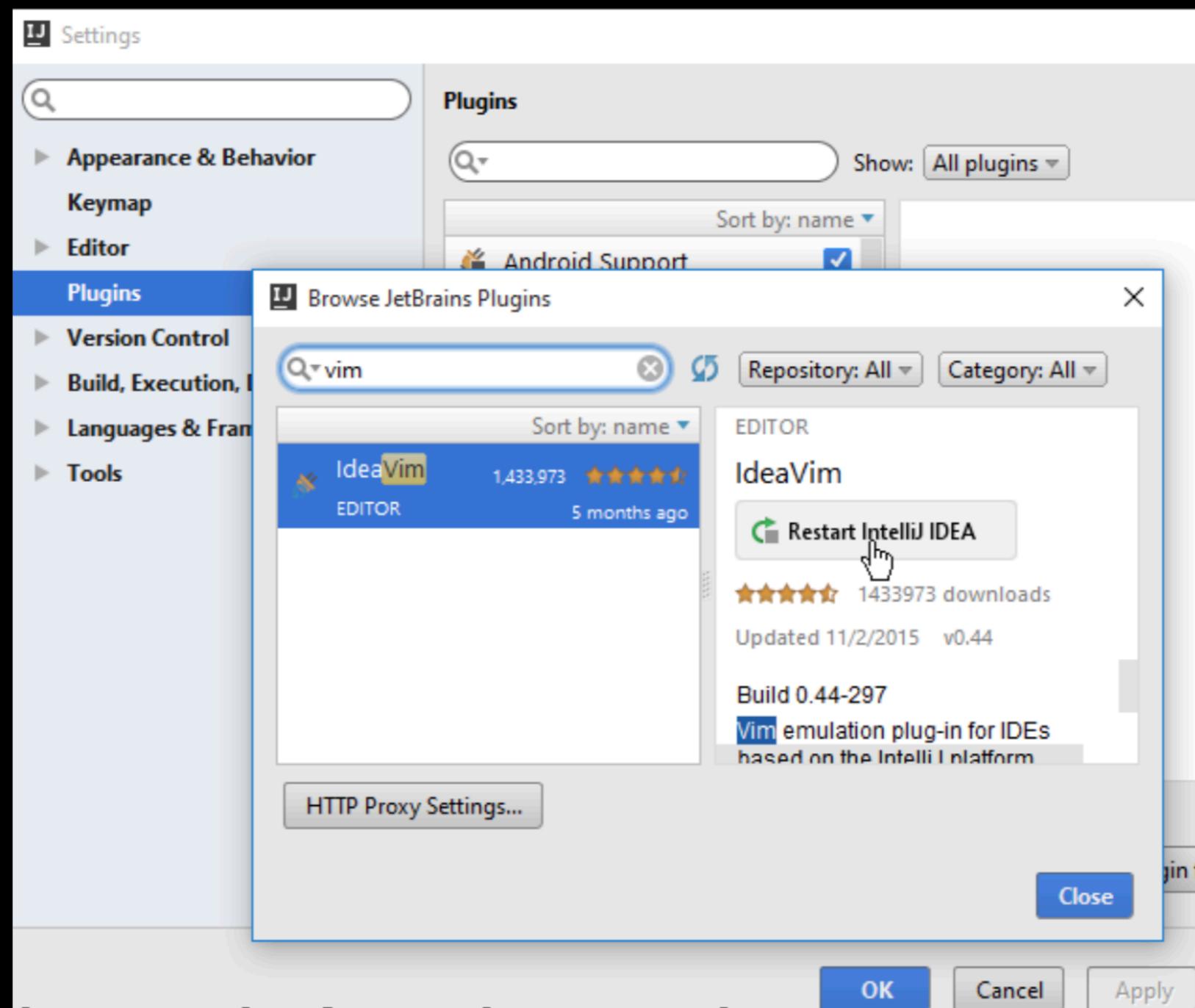
- 1) Терминал
- 2) Терминал ;D (ресурсы и скорость)
- 3) Кастомизация
- 4) Модальность (режимы)

БЕНЕФИТЫ:

Модальность (режимы)

БЕНЕФИТЫ:

Модальность (режимы) + \${ваша_любимая_идея}



 **Vim** `vscodevim.vim`

vscodevim |  2,553,390 | ★★★★★ | [Repository](#) | [License](#)

Vim emulation for Visual Studio Code

[Disable ▾](#) [Uninstall](#)

[Details](#) [Contributions](#) [Changelog](#) [Dependencies](#)

А есть туториалы?

А есть туториалы?



А есть туториалы?



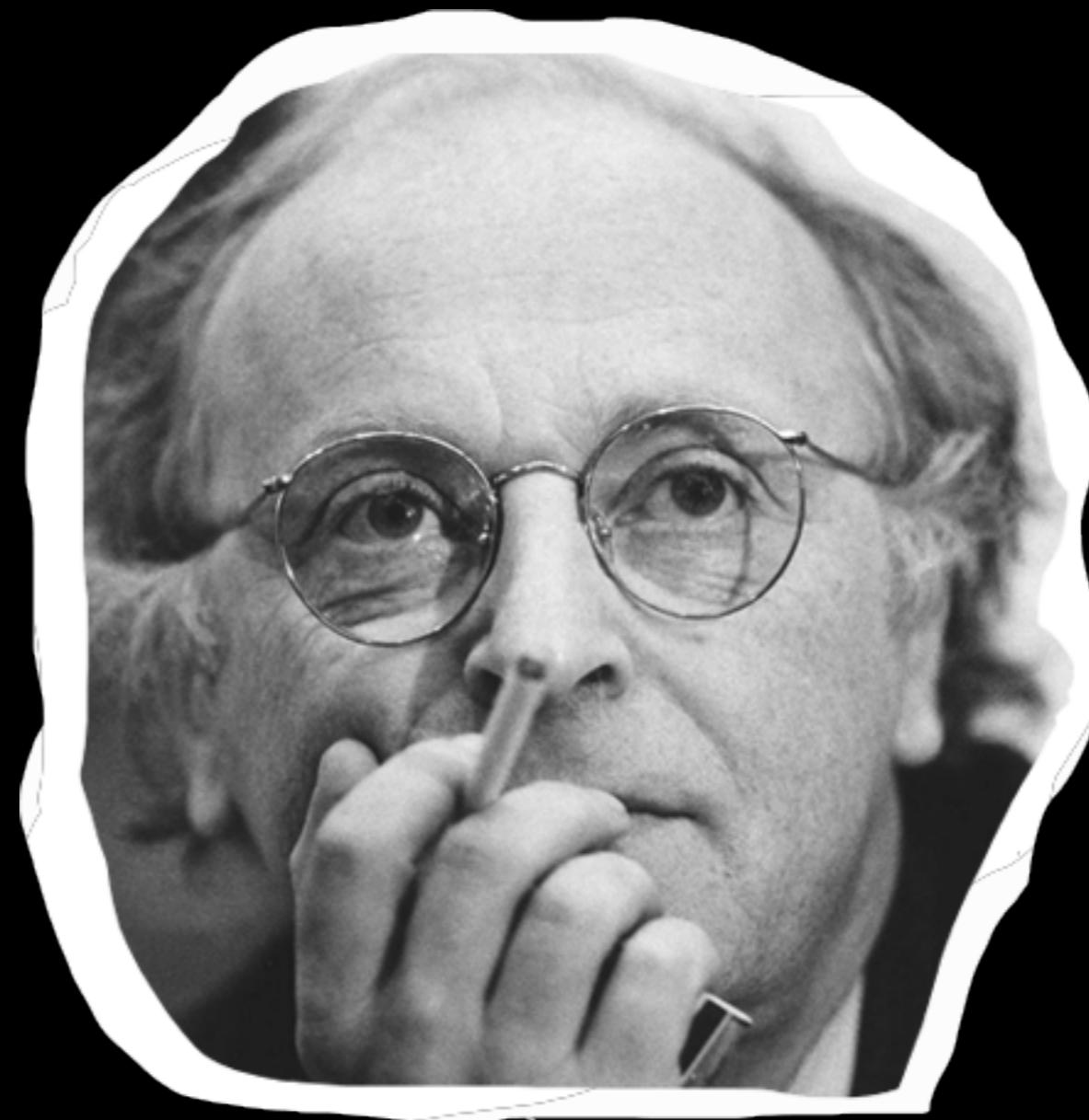
А есть туториалы?

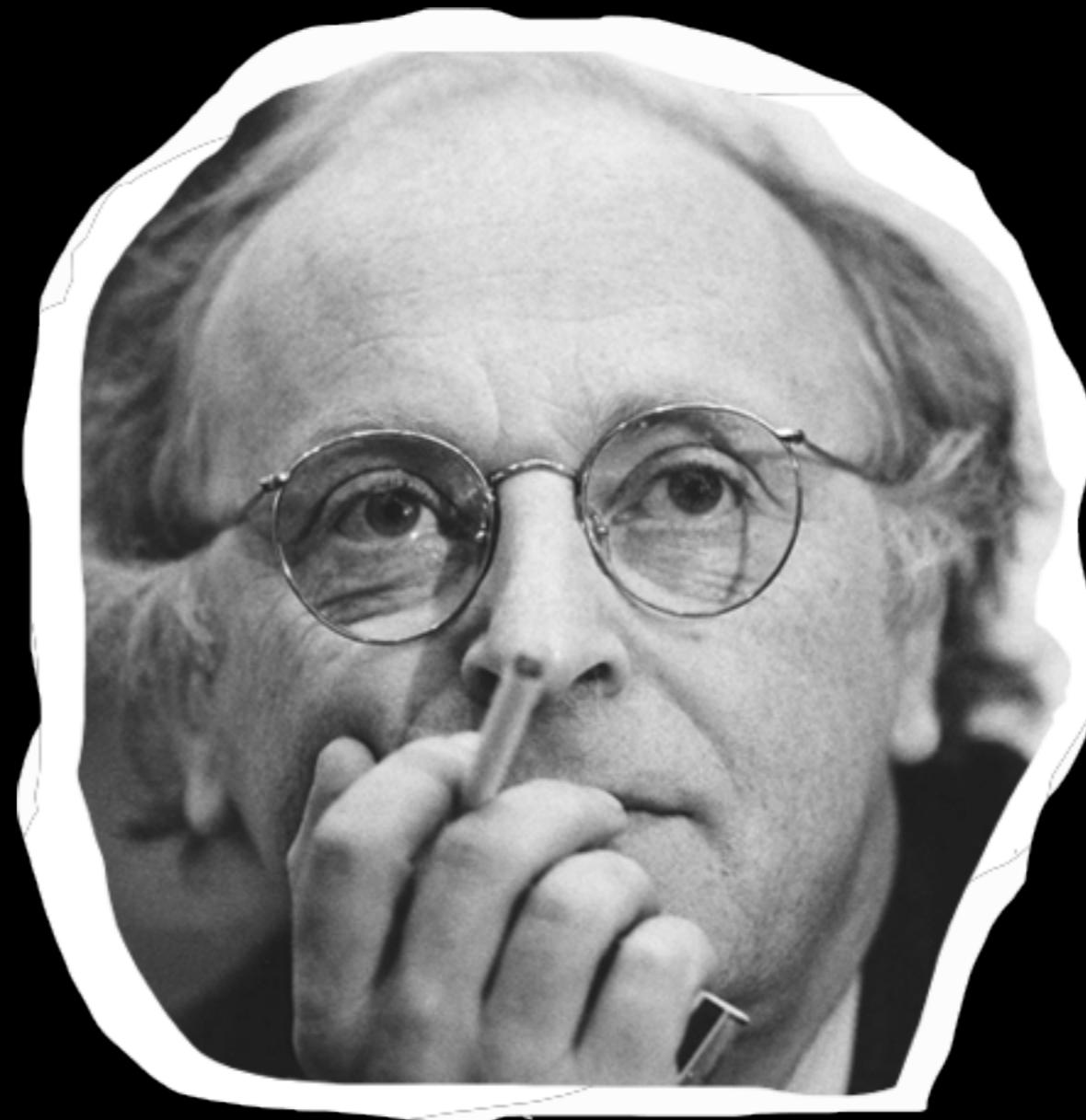


БЕНЕФИТЫ:

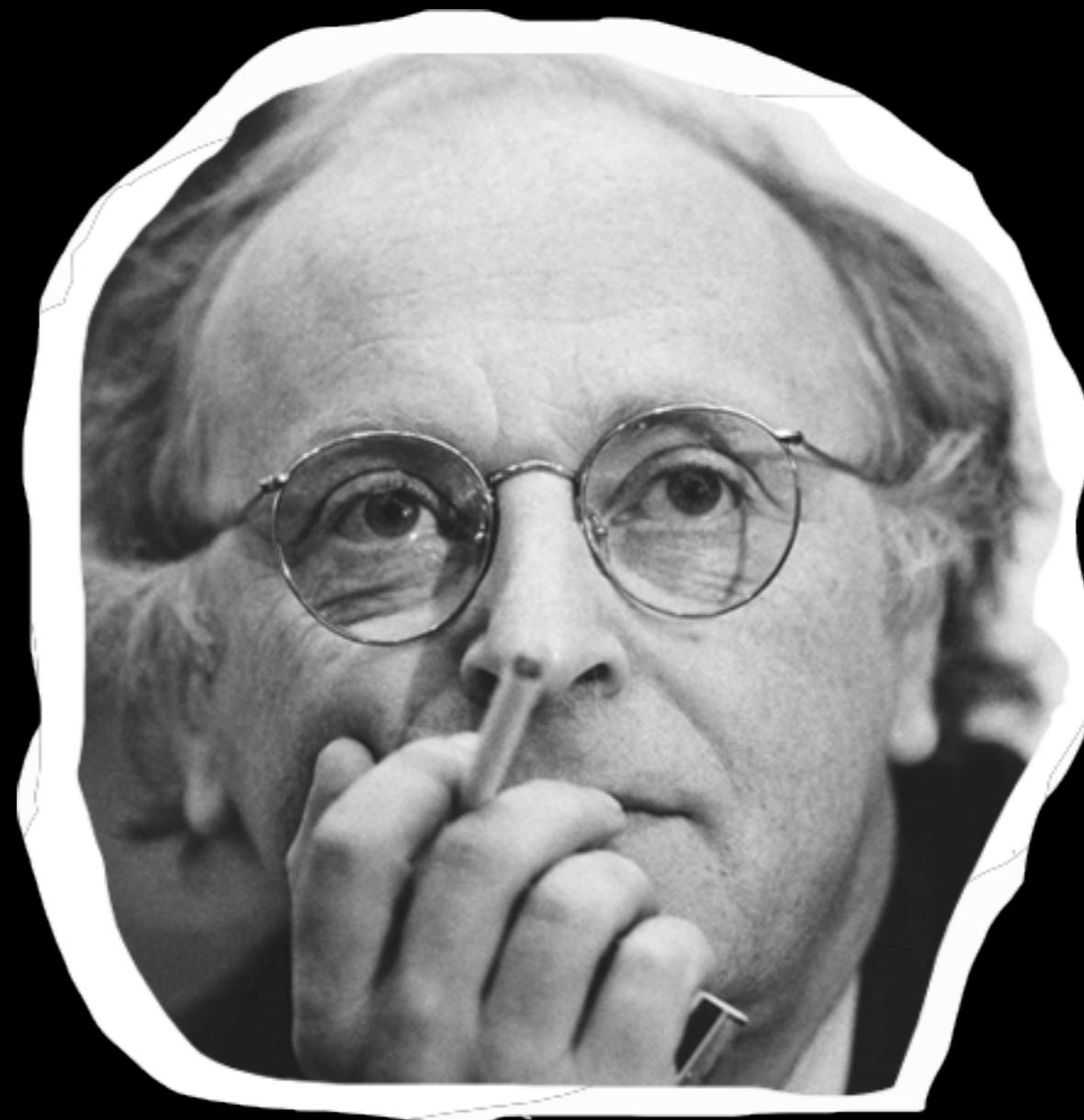
Модальность (режимы) + \${ваша_любимая_идея}

Vim везде!





НЕ ВЫХОДИ ИЗ VIM



НЕ ВЫХОДИ ИЗ VIM

НЕ СОВЕРШАЙ ОШИБКУ

КОНЕЦ





Вопросы / ответы

slak@samaradom.ru

nikolasmelui - соцсети

cto@multikey.studio