

Vim your Python

Python your Vim

Miroslav Šedivý

 eumiro

~	!	@	#	\$	%	^	&	*	(	)	-	+	←
1	2	3	4	5	6	7	8	9	0	-	=		
↩	Q	W	E	R	T	Y	U	I	O	P	{	}	
↩											[	]	\
⬆	A	S	D	F	G	H	J	K	L	:	"		↵
⬆										:	'		
⬆	Z	X	C	V	B	N	M	<	>	?		⬆	
								,	.	/			
Ctrl		Alt							Alt				Ctrl

# en, sk

~	!1 !+	@2 @'	#3 #š	\$4 \$č	%5 %ť	&6 &ž	*7 *ý	*8 *á	(9 (í	)0 )é	% %_	+ +'	←
↔	Q	W	E	R	T	Y y	U	I	O	P	/ [ú	( [ä	) [ň
↑	A	S	D	F	G	H	J	K	L	:" :"ô	!' !'š	↵	
↑ > <	Z z	X	C	V	B	N	M	? ,	> .	? / -	↑		
Ctrl		Alt							Alt			Ctrl	

á ä č ď é í ľ í ň ó ô ŕ š ť ú ý ž

# en, sk, de

° '^	!1 1+	@2 2'	#3 3š	\$4 4č	%5 5ť	&6 6ž	&7 7ý	*8 8á	(9 9í	)=0 0é	?1 1-	+2 2=	←
← →	Q	W	E	R	T	Y	U	I	O	P	{Ü [ú	*]ä ]ä	)ň
↑	A	S	D	F	G	H	J	K	L	Ö ô	'Ä 'š	' #	↵
↑ ↓	> <	Z	X	C	V	B	N	M	~ ,	~ .	? /-	↑	
Ctrl		Alt							Alt				Ctrl

á ä ç ď é í ľ ĺ ň ó ô ř š ť ú ý ž ö ü Œ

# en, sk, de, cs

° '^ ,	!1 1+	2 2ě	#3 3š	4 4č	5 5ř	'& ž	&/ ý	* á	(9) í	)= é	-? ß	+ =	←
← →	Q	W	E	R	T	Z	U	I	O	P	{Ü [ú	(* ]ě	) ň
↑	A	S	D	F	G	H	J	K	L	Ö ,û	'Ä 'š	' #	↵
↑ > <	> <	Z	X	C	V	B	N	M	;< ,	> .	? /-	↑	
Ctrl		Alt								Alt			Ctrl

á ä ç ď é í ľ ě Ń ó ô ř š ť ú ý ž ö ü ß ě ř ů

# en, sk, de, cs, fr

°	!1	2	#3	4	5	'&	&/	*(')	(9)	)=	:_	+'	←
^2 ,	1&	2è	3'	4ë	5(	6-	7è	8á	9ç	0à	-:)	=	
← →	A	VZ	E	R	T	Y	U	I	O	P	Ü	£	)
↑	Q	S	D	F	G	H	J	K	L	M	%	μ	↵
↑	> <	ZW	X	C	V	B	N	?	;	:/	§	↑	
Ctrl		Alt							Alt				Ctrl

á ä ç d' é í l' í ñ ó ô r' š t' ú ý ž ö ü ß ě ř ů à â æ ç è ë ê ï î œ ù û ŷ

# en, sk, de, cs, fr, es

!@	!1	2	#&	4	5	'&	&/	*(')	(9)	)=	:_	+;	←
^_	1&	2e	3'	4	5(	6-	7è	8á	9ç	0à	-:)	=;	
←	Q	W	E	R	T	Z	U	I	O	P	U	É	Ñ
↑	Q	S	D	F	G	H	J	K	L	M	Ç	µ	↵
↑	>	ZW	X	C	V	B	N	?	;	:/	§	↑	
Ctrl		Alt								Alt			Ctrl

á ä ç d' é í l' í ñ ó ô r' š t' ú ý ž ö ü ß ě ř ů à â æ ç è ë ê ï î œ ù û ÿ ñ ; i

# en, sk, de, cs, fr, es, pl

!@	!1	2	#&	4	5	'&	&/	*()	(9)	)=	0	+;	←
^_	1&	2e	3'	4c	5(	6-	7è	8á	9ç	0à	-:)	=í	
←	Q	W	E	R	T	Y	U	I	O	P	^	£	)
↑	Q	S	D	F	G	H	J	K	L	;	'	'	μ
↑	>	ZW	X	C	V	B	N	?	;	:/	;	!	↑
Ctrl		Alt							Alt				Ctrl

á ä ç ď é í ľ ě Ń ó ô ř š ť ú ý ž ö ü ß ě ř ů à â æ ç è ë ê ï î œ ù û ŷ ñ ĺ ; **ą ą ć ł ń ś ź ż**



# en, sk, de, cs, fr, es, pl, it

!@#	!1	@2	#3	\$4	%5	&6	&7	*8	(9)	)0	~	+^	←
!@#	!&	@e	#s	\$c	%i	&e	&e	*a	(c)	)a	-f)	=i	
←	A	VZ	E	R	T	Y	U	I	O	P	[É	]£	]'
←											[è	]\$	]ñ
↑	Q	S	D	F	G	H	J	K	L	;	'ç	'ç	's
↑											;	'ç	#ù
↑	>	ZW	X	C	V	B	N	?	;	:/	;	;	↑
↑	<							,	;	:/	;	;	↑
Ctrl		Alt								Alt			Ctrl

á ä ç ď é í ľ ě ň ó ô ř š ť ú ý ž ö ü ß ě ř ů à â æ ç è ë ê ï î œ ù û ŷ ñ ĺ ; ã ć ę ł ń ś ź ż ò

# en, sk, de, cs, fr, es, pl, it, sv

1/2	!@	~	#	\$	%	&	'	(	)	=	+	←
1/2	!@	~	#	\$	%	&	'	(	)	=	+	←
←	A	V	E	R	T	Z	U	I	O	P	U	U
←	A	V	E	R	T	Z	U	I	O	P	U	U
↑	Q	S	D	F	G	H	J	K	L	Ö	Ç	À
↑	Q	S	D	F	G	H	J	K	L	Ö	Ç	À
↑	>	Z	X	C	V	B	N	?	.	:/	§	↑
↑	>	Z	X	C	V	B	N	?	.	:/	§	↑
Ctrl		Alt								Alt		Ctrl

á ä ç ď é í ľ í ň ó ô ř š ť ú ý ž ö ü ß ě ř ů à â æ ç è ë ê ï î œ ù û ŷ ñ ç ; ã ć ę ł ń ś ź ż ò å

# en, sk, de, cs, fr, es, pl, it, sv, hu

!@#	!'	@	#&	\$%	%	'&/	&/=	*()	(9)	)ö	ü	+ó	←
!@#	!&	@e	#'	\$	%('	'&	&e	*á	(ç	)à	-+ü	=i	
←	A	VZ	E	R	T	Z	U	I	O	P	Ü Ö	Ú	Ń
↑	Q	S	D	F	G	H	J	K	L	Ń Ę	Á	Ű	↵
↑	> í	ZW	X	C	V	B	N	Ń	;	:/	Ű	↑	
Ctrl		Alt							Alt				Ctrl

á ä ç ď é í ľ ĺ Ń ó ô ř š ť ú ý ž ö ü ß ě ř ů à â æ ç è ë ê ï î œ ù û ŷ ñ ħ ; ; a c e ł n s z z ò å Ń ú

en, sk, de, cs, fr, es, pl, it, sv, hu, eo

!@# \$%^&* ( ) ð ö ù ú ó ←	!@# \$%^&* ( ) ð ö ù ú ó ←	!@# \$%^&* ( ) ð ö ù ú ó ←	!@# \$%^&* ( ) ð ö ù ú ó ←	!@# \$%^&* ( ) ð ö ù ú ó ←	!@# \$%^&* ( ) ð ö ù ú ó ←	!@# \$%^&* ( ) ð ö ù ú ó ←	!@# \$%^&* ( ) ð ö ù ú ó ←	!@# \$%^&* ( ) ð ö ù ú ó ←	!@# \$%^&* ( ) ð ö ù ú ó ←	!@# \$%^&* ( ) ð ö ù ú ó ←	!@# \$%^&* ( ) ð ö ù ú ó ←	!@# \$%^&* ( ) ð ö ù ú ó ←
← →	A V E R T Z U I O P	← →	A V E R T Z U I O P	← →	A V E R T Z U I O P	← →	A V E R T Z U I O P	← →	A V E R T Z U I O P	← →	A V E R T Z U I O P	← →
↑	Q S D F G H J K L	↑	Q S D F G H J K L	↑	Q S D F G H J K L	↑	Q S D F G H J K L	↑	Q S D F G H J K L	↑	Q S D F G H J K L	↑
↑	> <	Z W X C V B N	↑	> <	Z W X C V B N	↑	> <	Z W X C V B N	↑	> <	Z W X C V B N	↑
Ctrl		Alt							Alt			Ctrl

á ä ç ď é í ľ ĺ ň ó ô ř š ť ú ý ž ö ü ß ě ř ů à â æ ç è ë ê ï î œ ù û ŷ ñ ĺ ; ą ć ę ł ń ś ź ż ò ą ń ó ů ĉ ĝ ĥ ĵ ŝ ŭ

# en, sk, de, cs, fr, es, pl, it, sv, hu, eo, tr

!@#	~	^	&	*	( )	ö	ü	ó	←				
!@#	~	^	&	*	( )	ö	ü	ó	←				
←	A	V	E	R	T	Z	U	I	O	P	Ü	Û	;
↑	Q	S	D	F	G	H	J	K	L	;	'	ü	←
↑	>	Z	X	C	V	B	N	,	;	;	;	↑	
Ctrl		Alt							Alt			Ctrl	

á ä ç d' é í l' l' ñ ó ô r' s' t' ú ý ž ö ü ß ë ř ů à â æ ç è ë ê ì î œ ù û ÿ ñ ç ; ã c ç ł n' s' z' z' ò å ö ú ç ĝ h j s ů ĝ i i  
 §



# Character maps?

File View Search Go Help

Noto Sans **Bold** *Italic* 12 - +

Script	Character Table												Character Details
Kannada	Ŝ	ŝ	Ş	ş	Š	š	Ŧ	ŧ	Ř	ř	Ƒ	ƒ	Ũ
Katakana	ŭ	Ū	ū	Ŭ	ŵ	Ű	ű	Ů	ů	Ʋ	Ƴ	Ŵ	ŵ
Kayah Li	Ŷ	ŷ	Ÿ	Ž	ž	Ż	ż	Ž	ž	ƀ	Ɓ	Ƃ	ƃ
Kharoshthi	Ბ	Გ	Დ	Ე	Ვ	Ზ	Თ	Ი	Კ	Ლ	Მ	Ნ	Ო
Khmer	Ბ	Გ	Დ	Ე	Ვ	Ზ	Თ	Ი	Კ	Ლ	Მ	Ნ	Ო
Khojki	Ბ	Გ	Დ	Ე	Ვ	Ზ	Თ	Ი	Კ	Ლ	Მ	Ნ	Ო

Text to copy:  Copy

U+0160 LATIN CAPITAL LETTER S WITH CARON

# Compose Key





# Compose Key Sequences

⌘ " a ä	⌘ a e æ	⌘ . . ...
⌘ " o ö	⌘ s s ß	⌘ ? ? ¿
⌘ ' e é	⌘ t h þ	⌘ ! ! ¡
⌘ ` e è	⌘ / l ł	⌘ + - ±
⌘ ^ u û	⌘ o c ©	⌘ % o ‰
⌘ = o ö	⌘ o r ®	⌘ / = ≠
⌘ u u ů	⌘ / o ø	⌘ 1 2 ½
⌘ ~ n ñ	⌘ = e €	⌘ m u μ
⌘ c s š	⌘ - L £	⌘ o s §
⌘ , c ç	⌘   c Ć	⌘ ^ 2 ²
⌘ . z ž	⌘ m u μ	⌘ : ) 😊

# /usr/share/X11/locale/en\_US.UTF-8/Compose

```
<Multi_key> <period> <period> : "..." ellipsis # HORIZONTAL ELLIPSIS
<Multi_key> <plus> <minus> : "±" plusminus # PLUS-MINUS SIGN
<Multi_key> <apostrophe> <a> : "á" aacute # LATIN SMALL LETTER A WITH ACUTE
<Multi_key> <s> <s> : "ß" ssharp # LATIN SMALL LETTER SHARP S
<Multi_key> <comma> <c> : "ç" ccedilla # LATIN SMALL LETTER C WITH CEDILLA
<Multi_key> <c> <S> : "Š" U0160 # LATIN CAPITAL LETTER S WITH CARON
```

+6k more lines

# /usr/share/X11/locale/en\_US.UTF-8/Compose

```
<Multi_key> <period> <period> : "..." ellipsis # HORIZONTAL ELLIPSIS
<Multi_key> <plus> <minus> : "±" plusminus # PLUS-MINUS SIGN
<Multi_key> <apostrophe> <a> : "á" aacute # LATIN SMALL LETTER A WITH ACUTE
<Multi_key> <s> <s> : "ß" ssharp # LATIN SMALL LETTER SHARP S
<Multi_key> <comma> <c> : "ç" ccedilla # LATIN SMALL LETTER C WITH CEDILLA
<Multi_key> <c> <S> : "Š" U0160 # LATIN CAPITAL LETTER S WITH CARON
```

+6k more lines

~/.XCompose

# /usr/share/X11/xkb/rules/base

```
compose:ralt      = +compose(ralt)
compose:lwin      = +compose(lwin)
compose:lwin-altgr = +compose(lwin-altgr)
compose:rwin      = +compose(rwin)
compose:rwin-altgr = +compose(rwin-altgr)
compose:menu      = +compose(menu)
compose:menu-altgr = +compose(menu-altgr)
compose:lctrl     = +compose(lctrl)
compose:lctrl-altgr = +compose(lctrl-altgr)
compose:rctrl     = +compose(rctrl)
compose:rctrl-altgr = +compose(rctrl-altgr)
compose:caps      = +compose(caps)
compose:caps-altgr = +compose(caps-altgr)
compose:102       = +compose(102)
compose:102-altgr = +compose(102-altgr)
compose:paus      = +compose(paus)
compose:prsc      = +compose(prsc)
compose:sclk      = +compose(sclk)
```

# /usr/share/X11/xkb/rules/base

```
compose:ralt      = +compose(ralt)
compose:lwin      = +compose(lwin)
compose:lwin-altgr = +compose(lwin-altgr)
compose:rwin      = +compose(rwin)
compose:rwin-altgr = +compose(rwin-altgr)
compose:menu      = +compose(menu)
compose:menu-altgr = +compose(menu-altgr)
compose:lctrl     = +compose(lctrl)
compose:lctrl-altgr = +compose(lctrl-altgr)
compose:rctrl     = +compose(rctrl)
compose:rctrl-altgr = +compose(rctrl-altgr)
compose:caps      = +compose(caps)
compose:caps-altgr = +compose(caps-altgr)
compose:102       = +compose(102)
compose:102-altgr = +compose(102-altgr)
compose:paus      = +compose(paus)
compose:prsc      = +compose(prsc)
compose:sclk      = +compose(sclk)
```

```
setxkbmap us -option 'compose:menu'
```

19:23:22 <C00LGUY> HELLO EVERYONE! WHAT'S GOING ON?  
19:23:58 <me> Please hit that large key to the left from 'A'.  
19:24:10 <C00LGUY> why?  
19:25:11 <C00LGUY> COOL, I DON'T HAVE TO HOLD THE SHIFT KEY ANY MORE!

19:23:22 <C00LGUY> HELLO EVERYONE! WHAT'S GOING ON?  
19:23:58 <me> Please hit that large key to the left from 'A'.  
19:24:10 <C00LGUY> why?  
19:25:11 <C00LGUY> COOL, I DON'T HAVE TO HOLD THE SHIFT KEY ANY MORE!

**yEAH! WHO NEEDS "cAPS IOCK" ANYWAY?**

# /usr/share/X11/xkb/rules/base

```
caps:ctrl_modifier = +capslock(ctrl_modifier)
```



# /usr/share/X11/xkb/rules/base

```
caps:ctrl_modifier = +capslock(ctrl_modifier)
```

```
setxkbmap us -option 'compose:menu,caps:ctrl_modifier'
```

# Why just another plain Ctrl key?

if you can get two keys for the price of one

CapsLock + [X] ⇒ Ctrl + [X]

CapsLock ⇒ Esc

# Why just another plain Ctrl key?

if you can get two keys for the price of one

CapsLock + [X] ⇒ Ctrl + [X]

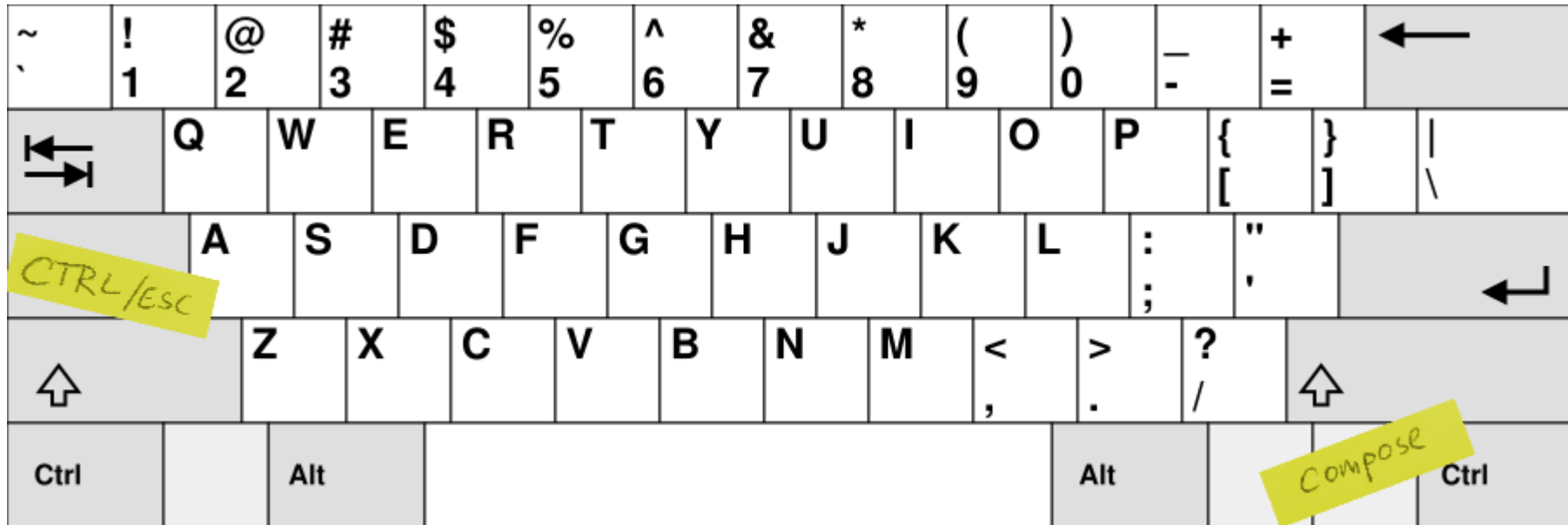
CapsLock ⇒ Esc

[github.com/alols/xcape](https://github.com/alols/xcape)

“xcape allows you to use a modifier key as another key when pressed and released on its own. [...] If you don't understand why anybody would want this, I'm guessing that Vim is not your favourite text editor ;)”

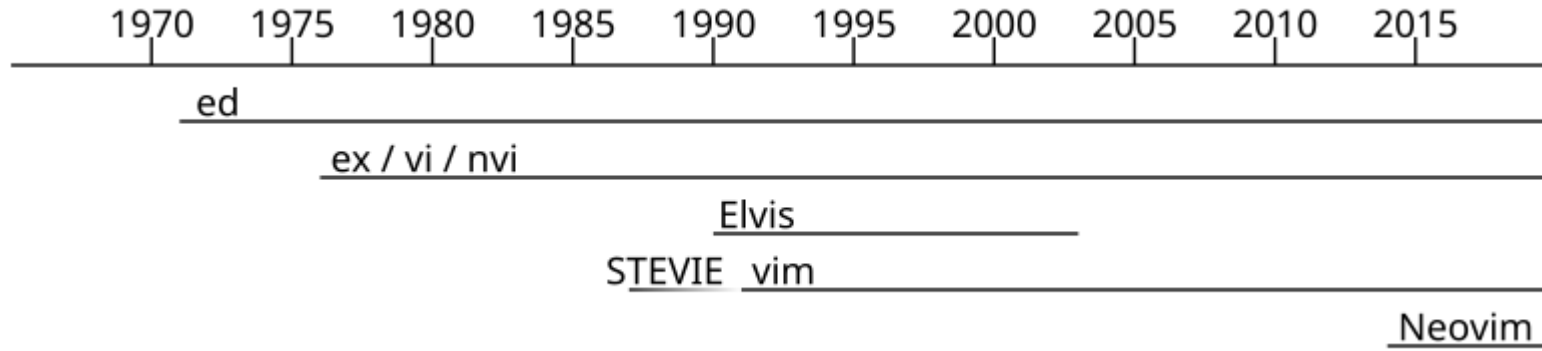
```
xcape -e "Caps_Lock=Escape"
```

# READY TO ROCK ^H^H^H^HVIM!



```
setxkbmap us -option 'compose:menu,caps:ctrl_modifier'  
xcapex -e "Caps_Lock=Escape"
```

# vi, vim, neovim



# Ritchie, Thompson, PDP-11 and ASR-33 (1970)

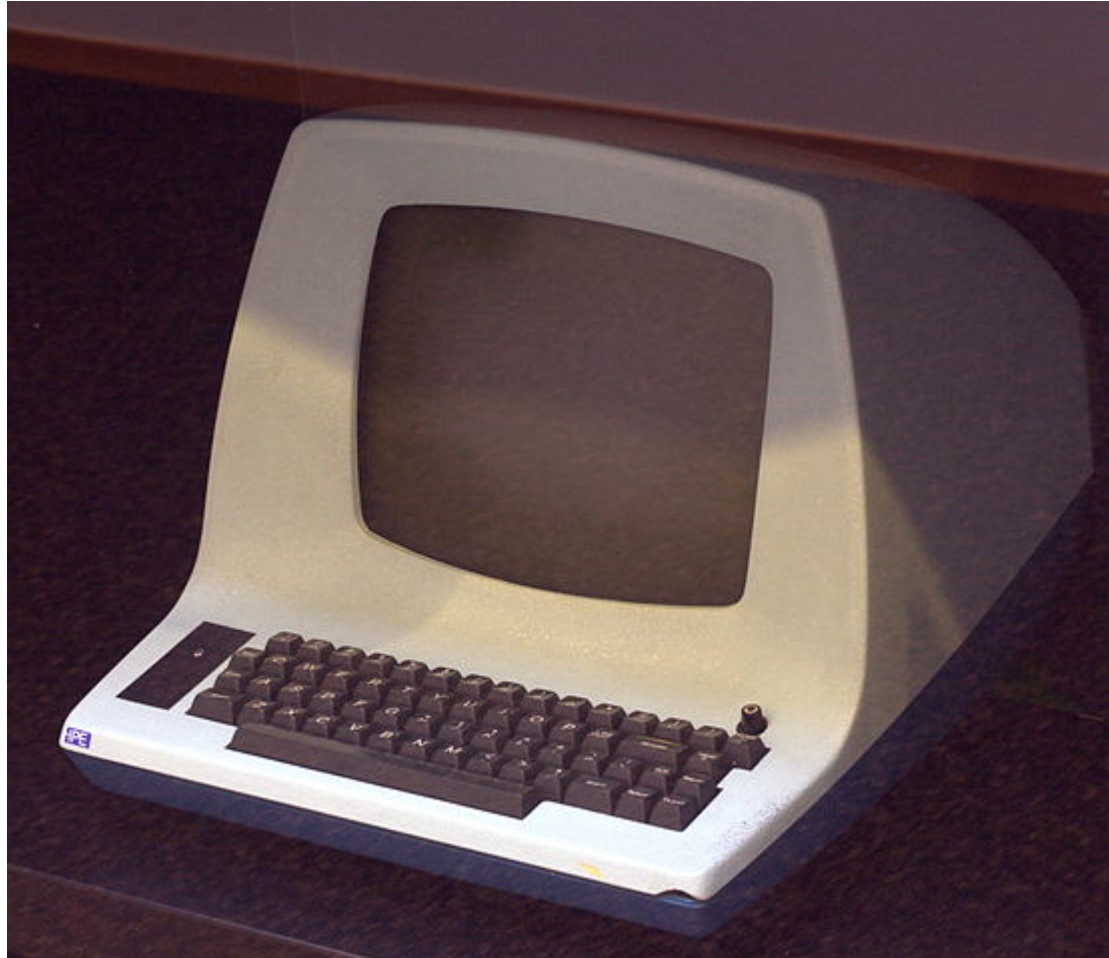


Source: Peter Hamer / Wikipedia, CC BY-SA 2.0

# ed(1)

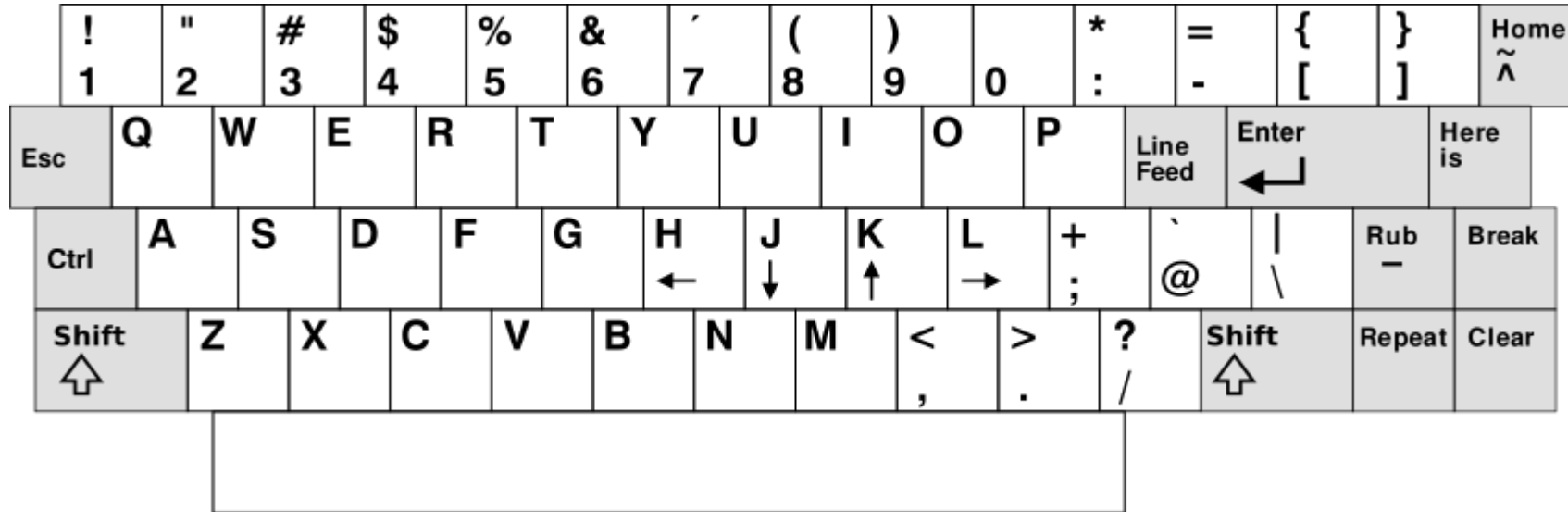
```
$ ed
a
Hello world
and everyone at PyParis
.
2p
and everyone at PyParis
a
How are you today?
.
%n
1 Hello world
2 and everyone at PyParis
3 How are you today?
1s/world/PyParis/
2d
%n
1 Hello PyParis
2 How are you today?
w test.txt
45
q
$
```

# ADM-3A Terminal





# ADM-3A Terminal



Source: StuartBrady / Wikipedia, CC BY-SA 3.0

# Where do h/j/k/l “arrows” come from?

0	^@	NUL	Null character	16	^P	DLE	Data link escape
1	^A	SOH	Start of Header	17	^Q	DCL	Xon (transmit on)
2	^B	STX	Start of Text	18	^R	DC2	Device control 2
3	^C	ETX	End of Text	19	^S	DC3	Xoff (transmit off)
4	^D	EOT	End of Transmission	20	^T	DC4	Device control 4
5	^E	ENQ	Enquiry	21	^U	NAK	Negative acknowledge
6	^F	ACK	Acknowledge	22	^V	SYN	Synchronous idle
7	^G	BEL	Bell	23	^W	ETB	End of transmission
8	^H	BS	Backspace	24	^X	CAN	Cancel
9	^I	HT	Horizontal tab	25	^Y	EM	End of medium
10	^J	LF	Line feed	26	^Z	SUB	Substitute
11	^K	VT	Vertical tab	27	^[	ESC	Escape
12	^L	FF	Form feed	28	^\ ^]	FS	File separator
13	^M	CR	Carriage return	29	^]	GS	Group separator
14	^N	SO	Shift out	30	^^	RS	Record separator
15	^O	SI	Shift in	31	^_ ^_	US	Unit separator

“Using a mouse is like pointing and grunting to get your point across.  
Vim lets you speak in full sentences.”

@Dean, [superuser.com/questions/599150/why-arrow-keys-are-not-recommended-in-vim](https://superuser.com/questions/599150/why-arrow-keys-are-not-recommended-in-vim)

# Vim Keyboard

~ 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.	} end 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	[ misc	] misc	
ESC normal mode	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 cmds	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 cmds	X delete char	C change	V visual mode	b prev word	n next (find)	m set mark	, reverse t/T/f/F	. repeat cmd	/ find		

Source: ViEmu.com, Esc key moved by MŠ



# vim-like software

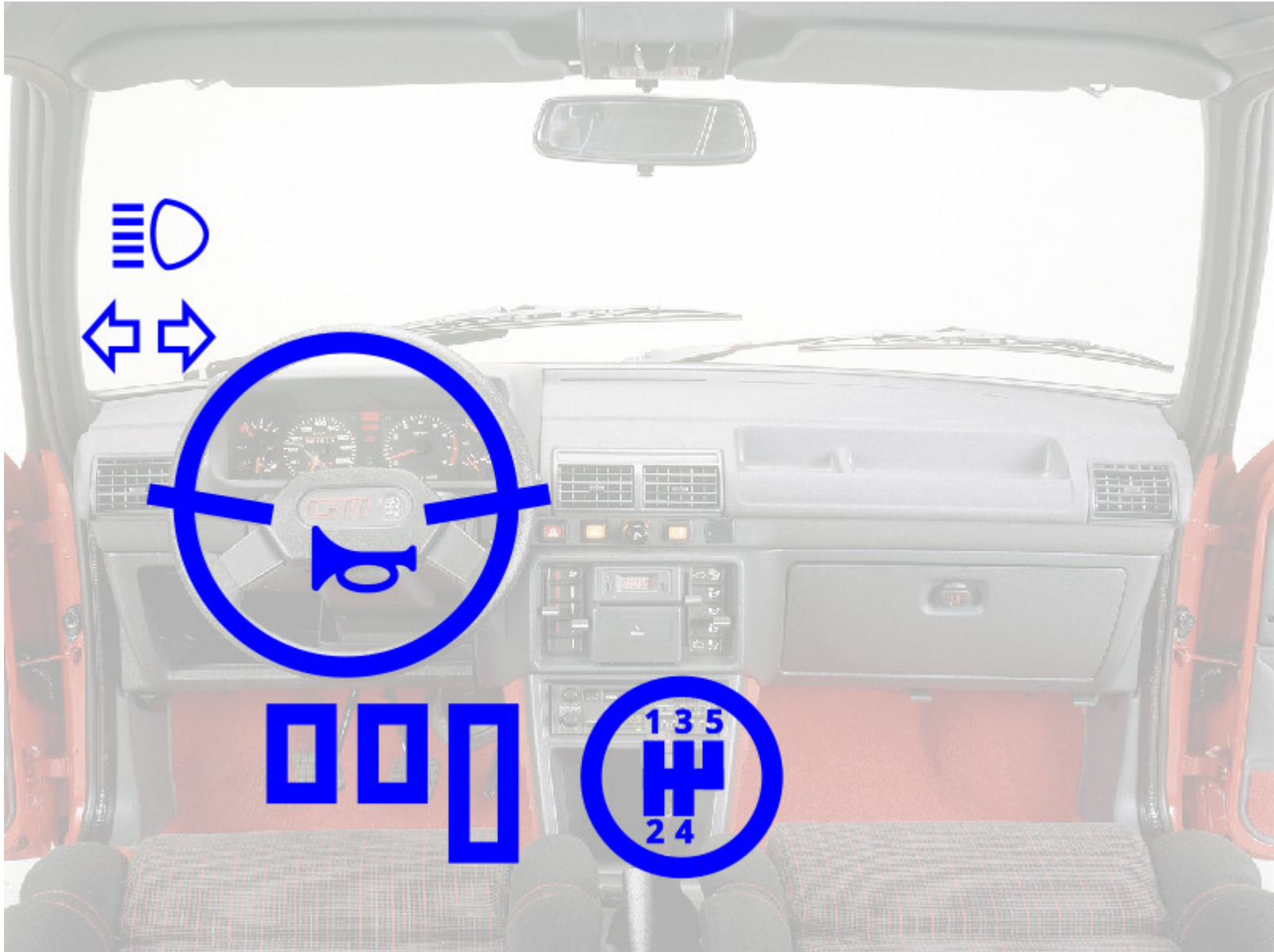
“vim-like” = hjkl movement, /? search, modes

- bash, zsh
- i3\*, tmux
- qutebrowser, vimperator, vimium
- mutt, newsbeuter
- less, vifm, zathura
- feh, ViPaint
- and more: [xaizek.github.io/2016-08-13/big-list-of-vim-like-software/](https://xaizek.github.io/2016-08-13/big-list-of-vim-like-software/)
- plenty of text editors and IDEs in “vi-mode”\*

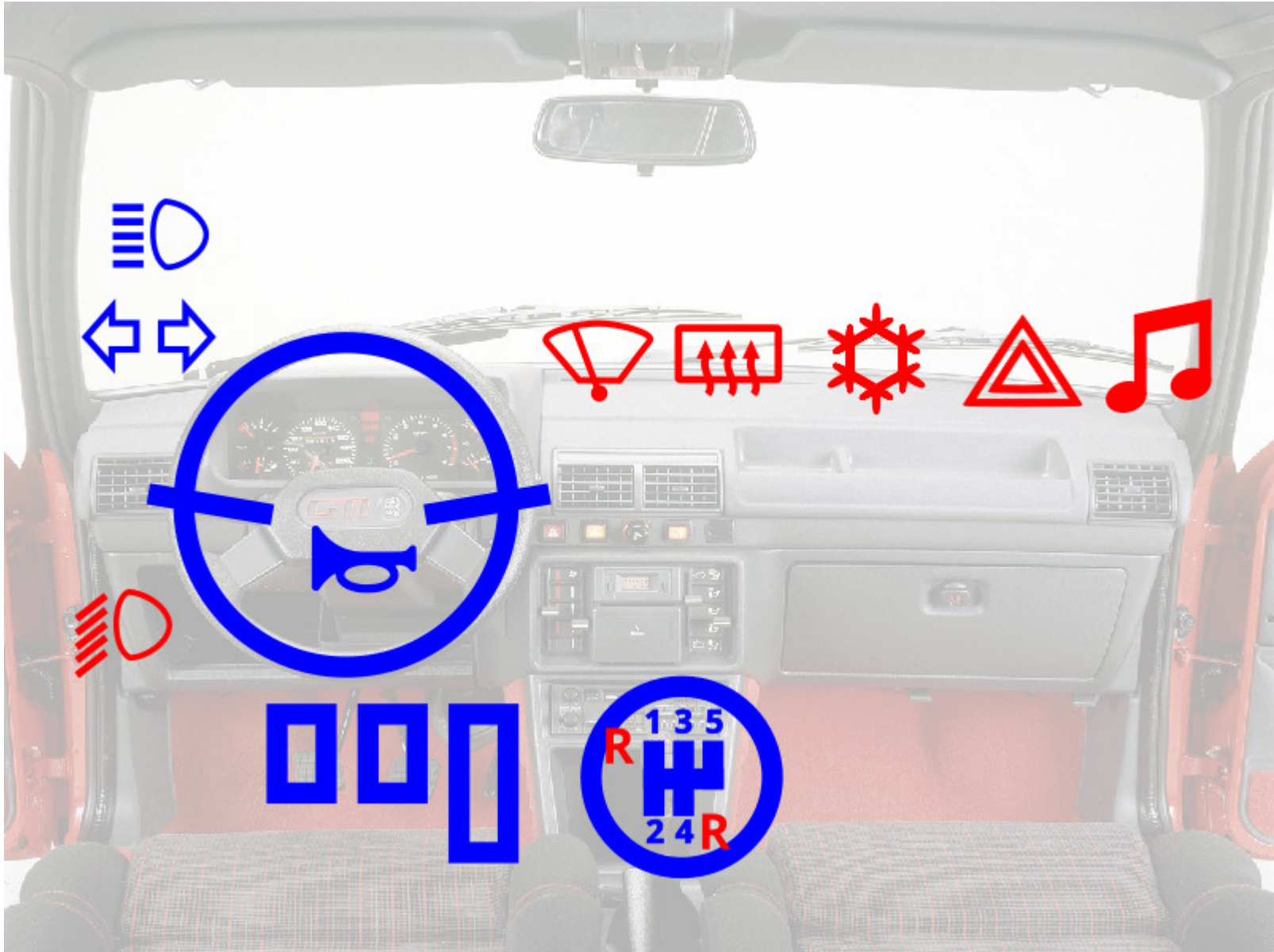
# Is it worth the time?

HOW LONG CAN YOU WORK ON MAKING A ROUTINE TASK MORE EFFICIENT BEFORE YOU'RE SPENDING MORE TIME THAN YOU SAVE?  
(ACROSS FIVE YEARS)

		HOW OFTEN YOU DO THE TASK					
		50/DAY	5/DAY	DAILY	WEEKLY	MONTHLY	YEARLY
HOW MUCH TIME YOU SHAVE OFF	1 SECOND	1 DAY	2 HOURS	30 MINUTES	4 MINUTES	1 MINUTE	5 SECONDS
	5 SECONDS	5 DAYS	12 HOURS	2 HOURS	21 MINUTES	5 MINUTES	25 SECONDS
	30 SECONDS	4 WEEKS	3 DAYS	12 HOURS	2 HOURS	30 MINUTES	2 MINUTES
	1 MINUTE	8 WEEKS	6 DAYS	1 DAY	4 HOURS	1 HOUR	5 MINUTES
	5 MINUTES	9 MONTHS	4 WEEKS	6 DAYS	21 HOURS	5 HOURS	25 MINUTES
	30 MINUTES		6 MONTHS	5 WEEKS	5 DAYS	1 DAY	2 HOURS
	1 HOUR		10 MONTHS	2 MONTHS	10 DAYS	2 DAYS	5 HOURS
	6 HOURS				2 MONTHS	2 WEEKS	1 DAY
1 DAY					8 WEEKS	5 DAYS	



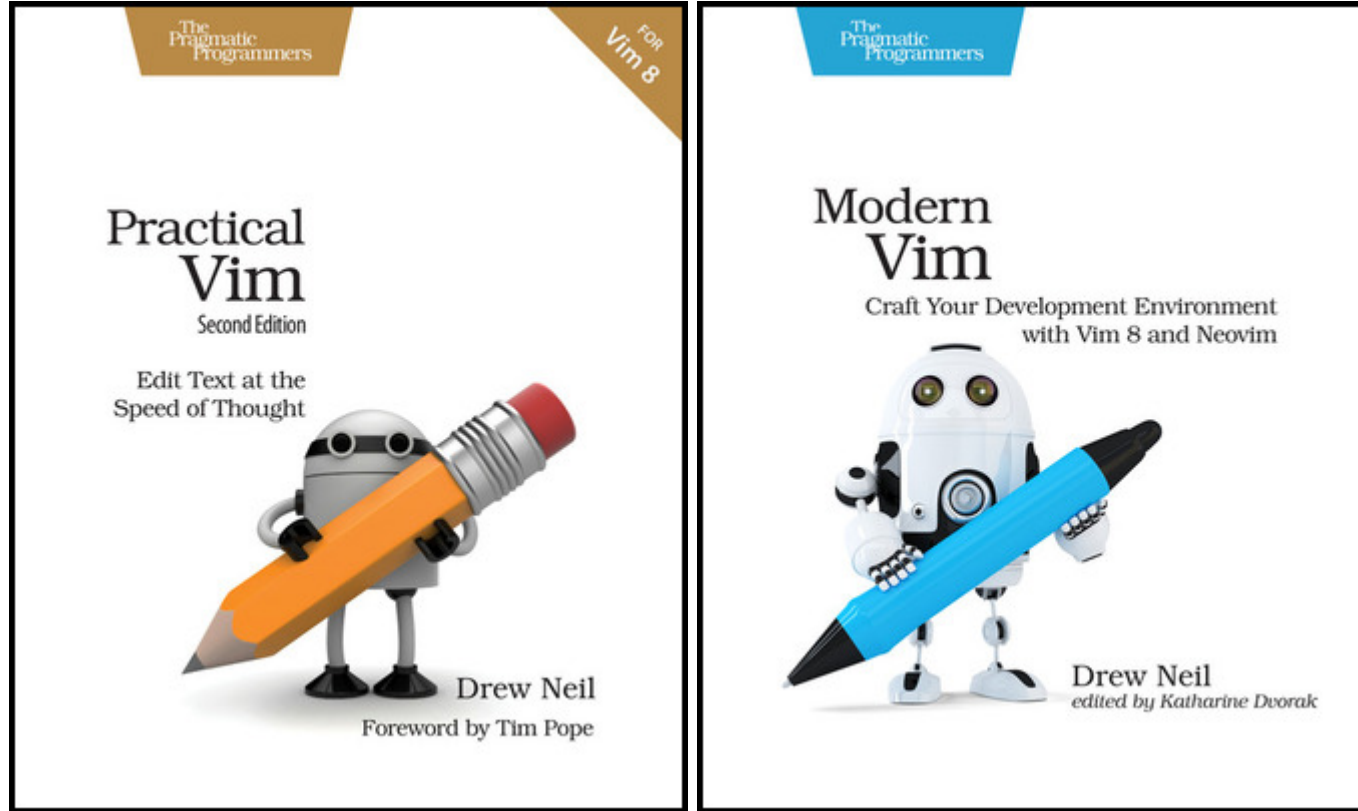




**“If you use a (software) tool more than 1 hour a day:  
read at least one book on it.”**

[stevenvanbael.com/read-about-the-software-you-use](https://stevenvanbael.com/read-about-the-software-you-use)

# Drew Neil



- [vimcasts.org](http://vimcasts.org)

# Vim 8 / Neovim

- Neovim is a fork of Vim 7.4
- asynchronous (lint, tests)
- `:terminal`
- modern defaults, runs on modern systems
- XDG directories: `~/.config/nvim/init.vim` instead of `~/.vimrc`
- `python-neovim`, `npm: neovim`

# Vim your Python

- syntax highlighting
- indenting
- auto-completion
- asynchronous linting (ale)
- git (live diff with gitgutter, manipulation with fugitive)
- searching ag/fzf
- snippets (neosnippet)
- folding
- interaction with terminal
- spellchecking

# \* your (Neo)Vim?

- C#
- C++
- Clojure
- Common Lisp
- Elixir
- Go
- Haskell
- Java
- Julia
- Lua
- Node.js
- Perl
- Python
- R
- Racket
- Ruby
- Rust
- Swift

# Python your (Neo)vim

```
$ pacman -S neovim python-neovim
```

```
$ conda install -c conda-forge neovim
```

# Python your (Neo)vim

```
$ pacman -S neovim python-neovim
```

```
$ conda install -c conda-forge neovim
```

```
from neovim import attach
```



# Python your (Neo)vim

```
$ pacman -S neovim python-neovim
```

```
$ conda install -c conda-forge neovim
```

```
from neovim import attach
```

```
NVIM_LISTEN_ADDRESS=/tmp/nvim nvim
```

```
nvim = attach('socket', path='/tmp/nvim')
```

```
nvim = attach('child', argv=["/bin/env", "nvim", "--embed"])
```

# Python your (Neo)vim

```
$ pacman -S neovim python-neovim
```

```
$ conda install -c conda-forge neovim
```

```
from neovim import attach
```

```
NVIM_LISTEN_ADDRESS=/tmp/nvim nvim
```

```
nvim = attach('socket', path='/tmp/nvim')
```

```
nvim = attach('child', argv=["/bin/env", "nvim", "--embed"])
```

```
nvim.current.buffer[0] # first line  
nvim.command('vsplit') # :vsplit
```

# ~/.config/nvim/rplugin/python3/hello.py

```
import email
from email.parser import Parser
import neovim

@neovim.plugin
class HelloPlugin:
    def __init__(self, nvim):
        self.nvim = nvim

    @neovim.autocmd('BufEnter', pattern='mutt-pad-*', sync=True)
    def on_bufenter(self):
        msg = Parser().parsestr('\n'.join(self.nvim.current.buffer))
        addresses = email.utils.getaddresses(msg.get_all('To'))
        hello_line = black_magic(addresses)
        first_empty = next((i for i, line in enumerate(self.nvim.current.buffer)
                           if not line), -1)
        self.nvim.current.buffer.append(['', hello_line], first_empty)
```

:UpdateRemotePlugins

# Customized (Neo)vim?

- master Vim without customization

# Customized (Neo)vim?

- master Vim without customization
- customize your own Vim
  - deactivate arrow keys
  - remap keys
  - shorten commands
  - appearance
  - behaviour
  - plugins

# Customized (Neo)vim?

- master Vim without customization
- customize your own Vim
  - deactivate arrow keys
  - remap keys
  - shorten commands
  - appearance
  - behaviour
  - plugins

**Don't put any lines in your vimrc that you don't understand.**

**.dotfiles are your tech CV**

# .dotfiles are your tech CV

```
alias config='/usr/bin/git --git-dir=$HOME/.dotfiles/ --work-tree=$HOME'
```



# .dotfiles are your tech CV

```
alias config='/usr/bin/git --git-dir=$HOME/.dotfiles/ --work-tree=$HOME'
```

```
git init --bare $HOME/.dotfiles  
config config --local status.showUntrackedFiles no  
  
# git remote add X Y (github, gitlab, uberspace, ...)
```

# .dotfiles are your tech CV

```
alias config='/usr/bin/git --git-dir=$HOME/.dotfiles/ --work-tree=$HOME'
```

```
git init --bare $HOME/.dotfiles  
config config --local status.showUntrackedFiles no  
  
# git remote add X Y (github, gitlab, uberspace, ...)
```

```
config add .config/nvim/init.vim  
  
# commit, branch, ...
```

# One Brain, One Keyboard, One Editor?

# One Brain, One Keyboard, One Editor?

- use your muscle memory

# One Brain, One Keyboard, One Editor?

- use your muscle memory
- one keyboard layout (US QWERTY with Compose)

# One Brain, One Keyboard, One Editor?

- use your muscle memory
- one keyboard layout (US QWERTY with Compose)
- one editor (the powerful one)

# One Brain, One Keyboard, One Editor?

- use your muscle memory
- one keyboard layout (US QWERTY with Compose)
- one editor (the powerful one)
- your programming language of choice

# One Brain, One Keyboard, One Editor?

- use your muscle memory
- one keyboard layout (US QWERTY with Compose)
- one editor (the powerful one)
- your programming language of choice
- take your dotfiles with you



# One Brain, One Keyboard, One Editor?

- use your muscle memory
- one keyboard layout (US QWERTY with Compose)
- one editor (the powerful one)
- your programming language of choice
- take your dotfiles with you
- don't put any lines in your dotfiles that you don't understand

# One Brain, One Keyboard, One Editor?

- use your muscle memory
- one keyboard layout (US QWERTY with Compose)
- one editor (the powerful one)
- your programming language of choice
- take your dotfiles with you
- don't put any lines in your dotfiles that you don't understand
- back to the roots

*“The enjoyment of one's tools  
is an essential ingredient of successful work.”*

*Donald E. Knuth*

Miroslav Šedivý

[ˈmɪrɔslav ˈʃɛɟɪviː]

 eumiro  eumiro  in šedivý