Back to Workshop Page 🕇

A few of our favourite tools



Marie-Hélène Burle & Alex Razoumov

training@westgrid.ca

March 16, 2022

Use left/right keys to change slide

lazygit

(Terminal UI for Git written in Go)

Status	Unstaged Changes	
r0:0 marie → m	modified: .emacs	
Files - Submodules		
→ ?? . Thome Imatic I parture S prog Imc Luit Imain. tf undo-treegz → ?? . Thome Imatic I parture S prog I memoteconfig1. I pasture undo-treegz → ?? . Thome Imatic I parture S prog I memoteconfig1. gitignore undo-treegz → ?? . Thome Imatic I parture S prog I trend to the treegz → ?? . Thome Imatic I parture S prog I trend to the treegz → ?? . Thome Imatic I parture S prog I trait I content I webinars I upscalling_data.org undo	<pre>[@ .emacs:146 @ mouse-3: menu for this date")</pre>	
	(company-lighter-base "")	
?? .!home!marie!systemd_log!mail_sync.log.~undo-tree~.gz ?? .!home!marie!systemd_log!monthly.log.~undo-tree~.gz	'(company-mirimum-prefix-length 10) '(company-uciktelp-color-background "brownd")	
?? .!home!marie!systemd_log!weekly.log.~undo-tree~.gz	(e.emacs:232 @ mouse-3: menu for this date")	
.zsh/custom ▼	'(emojify-download-emojis-p t)	
🛏 M alias.zsh	'(emojify-emoji-set "twemoji-v2-22")	
M .aspell.en.pws	'(emojify-emoji-styles '(unicode))	
M .emacs	'(emojify-inhibit-in-buffer-functions nil)	
M .gitconfig	'(erc-autojoin-channels-alist '(("localhost")))	
?? default_jupyter_kbd.json	(erc-edisable-ctop-teplies t) (erc-header-line-face-method t)	
Local Branches - Remotes - Tags	(el-'neauel'ille'nate*methou t) @ .emacs:2838 @ \$0`(yas-escape-text yas-selected-text)`"))	
	(require 'package) ; necessary	
	;; (setq package-archives '(("qnu" . "https://elpa.qnu.org/packages/")	
	;;; ("melpa". "https://melpa.org/packages/")	
	(add-to-list 'package-archives '("melpa" . "https://melpa.org/packages/") t)	
	(package-initialize)	
	@ .emacs:2849 @ \$0`(yas-escape-text yas-selected-text)`"))	
	(let ((inhibit-message t)) (message "curl: (6) Could not resolve host: emoji.slack-edge.com"))	
	(let ((inhibit-message t))	
	(message "curl: (6) Could not resolve host: secure.gravatar.com"))	
3da3fb17 MB alias; add comment	(let ((inhibit-message t))	
16b91ba9 MB Add xautolock on/off to ssof	(message 'curl' (6) Could not resolve host: avatars.slack-edge.com"))	
1e3dbe5e MB alias: update pdff		
1a226fd9 MB remove new line at end of .gitignore as that make it play well with		
744bf634 MB update gitignore	(message "Download Failed. STATUS: exit, EVENT: exited abnormally with code 6, URL: https://avatars.slack-edge.com/2021-01-11/1621414413302_7290da985afef09d5d	
9ce01fle MB alias: format + minor comment edit e4525347 MB alias: update path	61_32.jpg, NAME: /tmp/57ef60fe94e4862ca8a9f8b8402787b1.jpg, OUTPUT:"))	
e4525347 MB allas: update path b01dcc9b MB allas: better allas for llpp (pdf) kbd	<pre>(let ((inhibit-message t))</pre>	
46c934e3 MB add alias for ebook kbd	(uer (limituirmessage ()) (message "[error] [Compute Canada] REQUEST FAILED. URL: \"https://slack.com/api/rtm.connect\", PARAMS: ((\"mpim_aware\" . \"l\") (\"presence_sub\" . \"true\")	
eb38834b MB alias: big revamp of all clean functions	(mosting [compute conduct integers) integers) integers integers integers integers (comparison conduct integers (comparison conduct integers) integers (comparison conduct integers (comparison conduct integers) integers (comparison conduct integers) integers (comparison conduct integers (comparis	
bcf548c8 MB alias: add alex function to send large file over the net		
61071ef1 MB alias: fix wacom tablet code	;; ** change yes-no questions to y-n	
74fdabd2 MB ranger: make all fzf commands to cycle		
36c77004 MB xinitrc: better comments	(defalias 'yes-or-no-p' y-or-n-p)	
	@ .emacs:3332 @ displayed anywhere else." (bookmark-/ump-via bookmark (or display-func 'pop-to-buffer-same-window))	
	(DOOKmaTk-Jump-VLa DOOKmaTk (OC GISplay-Tunc pop-to-DUTTET-same-window)) (recenter-top-bottom))	
5: jump to panel, [/]: scroll left/right, ctrl+g: cancel, ctrl+k/ctrl+j: scroll	, q: quit, x: menu, ∢ ▶ ▲ ▼: navigate Donate Ask Question 0.33	

lazygit brings the ease of use of a visual Git tool to the command line

Unlike slow, buggy, & limited Git GUIs, it is nimble, portable, & super fast

In the following demo, as an additional visual goody, I will introduce **diff-sofancy** to display the outputs of <code>git diff</code>

Bat

("A cat clone with wings")

	File: /home/marie/parvus/prog/tdoc/djl/cjl/parallel/generic_loop/both.jl
1	<i>#####################################</i>
2 3	# sequential
4	/ # run it with:
5	# julia parallel.jl
6	<i>*************************************</i>
7	
8	using BenchmarkTools
9	
10	<pre>function slow(n::Int, digits::String)</pre>
	total = 0.
12	for i in 1:n
13	<pre>if !occursin(digits, string(i))</pre>
14	total += 1. / i
15	end
16	end
17	println("total = ", total)
18	end
19	
20	<pre>@btime slow(Int64(1e9),"9")</pre>
21	

bat is a great tool to display the content of files in your terminal

The standard Unix utility for this task is **cat**, but bat adds a lot to it

Syntax highlighting...

...in a huge number of languages. See the impressive list with:

bat -L

cat /home/marie/parvus/prog/slides/static/files/tools/julia.jl

Syntax highlighting...

...in a huge number of languages. See the impressive list with:

bat -L

cat /home/marie/parvus/prog/slides/static/files/tools/julia.jl

```
# sequential
# run it with:
# julia parallel.jl
using BenchmarkTools
function slow(n::Int, digits::String)
  total = 0.
     for i in 1:n
       if !occursin(digits, string(i))
          total += 1. / i
       end
    end
  println("total = ", total)
end
@btime slow(Int64(1e9),"9")
```

Syntax highlighting...

... in a huge number of languages. See the impressive list with:

bat -L

bat /home/marie/parvus/prog/slides/static/files/tools/julia.jl

Smart behaviour with pager

Uses less by default with options -RFX

Very convenient: quits the pager automatically if the entire content fits on one script

Integration with Git

File: /home/marie/parvus/prog/slides/static/files/tools/two_layer_net_tensor.py Size: 2.3 KB import torch from torchvision import datasets, transforms from matplotlib import pyplot as plt device = torch.device('cpu') dtype = torch.float epochs = 30 train = datasets.MNIST(train = False, download = False, transform = transforms.Compose([transforms.ToTensor(), transforms.Normalize((0.1307,), (0.3081,))])) 21 test = datasets.MNIST(23 ~ /~/projects/def-sponsor00/data', train = False, download = False, transform = transforms.Compose([transforms.ToTensor(), transforms.Normalize((0.1307,), (0.3081,))])) 30 N, D_in, H, D_out = 60000, 784, 100, 10 x = train.data.view(60000, 784).type(dtype) y = train.targets.unsqueeze(1).expand(60000, 10).type(dtype) w1 = torch.randn(D_in, H, device = device, dtype = dtype) w2 = torch.randn(H, D_out, device = device, dtype = dtype) 39 + | optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate) 40 + | for t in range(500): 42 + v pred = model(x) :

Git diff

bat -d

	<pre>File: /home/marie/parvus/prog/slides/static/files/tools/two_layer_net_tensor.py Size: 2.3 KB</pre>
	<pre>train = datasets.MNIST(# './data', '~/projects/def-sponsor00/data',</pre>
	<pre>train = False, download = False, transform = transforms.Compose([</pre>
	<pre>test = datasets.MNIST(# './data', '~/projects/def-sponsor00/data', train = False, download = False, </pre>
	transforms.ToTensor(), transforms.Normalize((0.1307,), (0.3081,))])) N, D_in, H, D_out = 60000, 784, 100, 10
	w2 = torch.randn(H, D_out, device = device, dtype = dtype)
	<pre>optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate) for t in range(500): # Forward pass: compute predicted y by passing x to the model. y_pred = model(x)</pre>
	<pre># Compute and print loss. loss = loss_fn(y_pred, y) print(t, loss.item())</pre>
	<pre>writer.add_scalar("training loss", loss, t) # Before the backward pass, use the optimizer object to zero all of the # gradients for the Tensors it will update (which are the learnable weights # of the model) optimizer.zero_grad()</pre>
53 + 54 + 55 + 56 +	# Backward pass: compute gradient of the loss with respect to model parameters loss.backward()

Man page syntax highlighting

Even colourizes your man pages if you set the MANPAGER environment variable (e.g. in your .bashrc):

export MANPAGER="sh -c 'col -bx | bat -l man -p'"

Useful flags

bat -n	# Remove frame
bat -p	# Remove frame & line numbers
I personally have an	alias for this: alias rat='bat -p'
bat -pp	# Remove frame & line numbers & don't use pager
bat -f	# Keep colour highlighting after piping

I made this the default behaviour for bat with the alias: alias bat='bat -f'

Fully customizable

The defaults are great, but if they don't suit you, you can change tab width, language, wrapping method...

bat --config-file # Location of config file (if it exists)
bat --generate-config-file # Create config file template
bat --list-themes # List themes with examples

ripgrep

(A fast recursive alternative to grep)

13:57 m ⊠ 1 ~/parvus/prog/slides rg tools content/git_branches/_index.md 415:Many GUI applications for Git as well as powerful text editors such as Emacs and Vim offer merge tools

content/r_gis/_index.md
103:`basemaps` is not on CRAN and needs to be installed from GitHub thanks to `devtools`:
106:install.packages("devtools")
107:devtools::install_github("16EAGLE/basemaps")

content/r_gis_rc/_index.md
98:`basemaps` is not on CRAN and needs to be installed from GitHub thanks to `devtools`:
101:install.packages("devtools")
102:devtools::install_github("16EAGLE/basemaps")

content/help/_index.md 94:Take the time to look for the official documentation & other high quality sources for the languages & tools you use.

content/cli_tools/_index.md
2:title = "cli_tools"
7:href = "https://westgrid-cli.netlify.app/webinars/cli_tools/"
24:
38:[[logo](/img/cli_tools/fzf.png)](https://github.com/junegunn/fzf)
510:[![logo](/img/cli_tools/andgr.png)](https://github.com/wting/autojump)
518:[![logo](/img/cli_tools/ranger.png)](https://github.com/ranger/ranger)
558:

263:{{}}

2:title = "tools" 7:href = "https://westgrid-cli.netlify.app/webinars/tools/" 14:## <div style="line-height: 5rem; font-size: 4.2rem; color: black; text-shadow: 0 0 0.7rem white;">A few of our favourite tools</div> 35:{{}} 44:{{}} 66:cat /home/marie/parvus/prog/slides/static/files/tools/julia.jl 68:{{}} 85:bat /home/marie/parvus/prog/slides/static/files/tools/julia.jl 87:{{}} 103:{{}} 114:{{}} 199:{{}} 207:{{}} 215:{{}} 223:{{}} 231:{{}} 239:{{}} 247:{{}} 255:{{}}

ripgrep provides the utility rg: an alternative to grep very fast for recursive searches

Syntax

The syntax is similar to that of grep

Because, by default, rg is recursive (grep isn't) & ignores some files, rg -uuu is equivalent to grep -r

Benchmarks

I will compare grep and rg to recursively search the word "Unix" in my directory "~/parvus/prog/tcl"

grep -r Unix ~/parvus/prog/tcl
rg Unix ~/parvus/prog/tcl

I will do a cold test (clearing my computer cache before each run) using **hyperfine**

hyperfine -i --prepare 'sync; echo 3 | sudo tee /proc/sys/vm/drop_caches' \
 'grep -r Unix ~/parvus/prog/tcl' \
 'rg Unix ~/parvus/prog/tcl'

Results

Benchmark 1: g	rep -r Uni	.x ~/parvu	is/prog/tc	21		
Time (mean ± σ): 1.	027 s ±	0.016 s	[User: 0.269 s,	System:	0.222 s]
Range (min … ma	ax): 1.	007 s	1.052 s	10 runs		
Benchmark 2: r	g Unix ~/p	oarvus/pro	og/tcl			
Time (mean ± σ	;): 76	6.0 ms ±	5.6 ms	[User: 21.5 ms,	System:	29.6 ms]
Range (min … ma	ax): 63	3.0 ms	82.4 ms	10 runs		
Summary						
'rg Unix ~/parvus/prog/tcl' ran						
13.51 ± 1.02 times faster than 'grep -r Unix ~/parvus/prog/tcl'						

What makes rg fast?

Uses parallelism

Uses Rust regex engine with finite automata, SIMD, & aggressive literal optimizations (regular expressions follow the syntax of grep -E; see this & this for details)

By default, ignores files in .gitignore & similar, hidden files/directories, & binary files (this behaviour can be changed with flags)

fd

(Fast & friendly alternative to find)

```
08:38 m ~/parvus/prog/tdoc fd -e jl
dcl/ccl/getting_help/script.jl
dcl/ccl/getting_help/script_fixed.jl
djl/cjl/covid/covid.jl
djl/cjl/covid/covid_old.jl
djl/cjl/dagger/benchmark.jl
djl/cjl/dagger/dagger oom.jl
djl/cjl/dagger/dagger_threads.jl
djl/cjl/flux/flux.jl
djl/cjl/intro/distributed.jl
djl/cjl/intro/intro.jl
djl/cjl/intro/live_code.jl
djl/cjl/intro/pi.jl
djl/cjl/intro/pi_parallel.jl
djl/cjl/intro/plots.jl
djl/cjl/intro/plots_bkup.jl
djl/cjl/intro/psort.jl
djl/cjl/intro/sort.jl
djl/cjl/intro/variables_types.jl
djl/cjl/jupyter/covid_jupyterdays.jl
djl/cjl/jupyter/hello.jl
djl/cjl/jupyter/literate_cleaned.jl
djl/cjl/jupyter/literate_example.jl
djl/cjl/jupyter/nbinclude.jl
```

fd has an intuitive syntax & is super fast

pass

(A command line password manager for Unix systems)



pass is a command line password manager using GnuPG for encryption

pass has a Git integration & several extensions

pass is easy to use in scripts or combined with other utilities

Main pass commands

pass init your-gpg-id

- pass insert somesite.ca/your@email
- pass somesite.ca/your@email
- pass -c somesite.ca/your@email
- pass edit somesite.ca/your@email
- pass generate othersite.com/userid 12
- pass generate -c othersite.com/userid 12 # Same, but copy to clipboard

pass rm othersite.com/userid

pass

- # Initialize password store
- # Enter password for a site
- # Print password
- # Copy password to clipboard
- # Edit password
- # Generate password of size 12
- pass generate -n othersite.com/userid 12 # Same, without special characters
 - # Delete password
 - # Print content of password store

TRAMP

("A remote file editing package for Emacs")

The **TRAMP** (Transparent Remote Access, Multiple Protocol) Emacs package allows to edit remote files as if they were local

You can thus use your init file & usual packages to write files on a remote machine or for another user (e.g. root)

You can also use the Emacs GUI to edit remote files without efficiency loss

It is a convenient alternative to using Emacs on a remote machine after you have sshed into it or to using sudo emacs to edit root files

Using TRAMP from within Emacs

Remote file

C-x C-f /ssh:user@server:/home/user/path/to/file RET

Root

C-x C-f /sudo::/path/to/file RET

Using TRAMP from the command line

Remote file

emacs /ssh:user@server:/home/user/path/to/file

Root

emacs /sudo::/path/to/file

Helm

("Incremental completion & selection narrowing framework")

C-j: Go to Line (keeping session) HO [.emacs]	<pre>'(vertical-border ((((type w32 tty)) (:inherit mode-line-inactive :foreground</pre>
18: '(TeX-auto- <mark>global</mark> "~/.emacs.d/auctex/")	"firebrick4"))))
316: '(global-aggressive-indent-mode t)	<pre>'(vimish-fold-mouse-face ((t (:background "gray15" :foreground "indianred"))))</pre>
317: '(global-emojify-mode t)	'(vimish-fold-overlay ((t (:background "MistyRose4" :foreground "black"))))
318: '(global-emojify-mode-line-mode t)	'(warning ((t (:foreground "red2" :weight bold))))
319: '(global-git-commit-mode t)	'(whitespace-empty ((t (:background "light green" :foreground "gray10"))))
320: '(global-undo-tree-mode t)	'(whitespace-hspace ((t (:foreground "orange red"))))
321: '(global-visual-line-mode nil)	'(whitespace-line ((t (:background "gray15"))))
692: '(org-cycle- <mark>global</mark> -at-bob t)	'(whitespace-space ((t (:foreground "olivedrab")))) '(whitespace-trailing ((t (:background "brown2"))))
2181: '(outshine-org-style-global-cycling-at-bob-p t)	
2260: '(smartparens-global-mode t)	'(window-divider ((t (:foreground "gray25")))) '(yas-field-highlight-face ((t (:underline t)))))
2261: '(smartparens-global-strict-mode nil)	(yas-rield-nightight-race ((t (:underline t)))))
2335: '(writeroom-global-effects	:: * unset env variable MANPAGER in Emacs only
2338: '(yas-global-mode t)	;; * UNSEL ENV VALUABLE MANFAGER (I EMACS ONLY ; MANPAGER is set in zsh to sh -c 'col -bx bat -l man -p'
2800:(global-set-key (kbd ";") 'ring-map)	
2804:;; (global-set-key (kbd "; SPC")	;; which provides bat colour highlighting in the man pages
2807:;; (global-set-key (kbd "; ; SPC") 'ignore)	;; but this messes up the help files in inferior-python-mode
2809:(global-set-key (kbd "; SPC") 'ignore)	(setenv "MANPAGER" "")
2811:(global-set-key (kbd "; '")	(setenv manfager)
2817:(global-set-key (kbd "C-z") 'ring-map-two)	;; * define new prefix keys
2877:;; useful only when global-emojify-mode is nil	;; * der the new prer tx keys
2934: (global-set-key (kbd " <kp-next>") #'fix-word-downcase) ; F+1 (l=lower case)</kp-next>	
2935:(global-set-key (kbd " <kp-left>") #'fix-word-upcase) ; F+u (u=upper case)</kp-left>	
2936: (global-set-key (kbd " <kp-down>") #'fix-word-capitalize) ; F+k (k=capitalize)</kp-down>	(define-prefix-command 'ring-map)
2940: (global-set-key (kbd "; h 1") "* ")	(global-set-key (kbd ";") 'ring-map)
2942:(global-set-key (kbd "; h 2") "** ")	(grobal-set-key (kbu ;) ring-map)
2944:(global-set-key (kbd "; h 3") "*** ")	;; *** make sure I can still type ;
2946:(global-set-key (kbd "; h 4") "**** ")	;; *** Make sure i can still Lype ;
2948:(global-set-key (kbd "; h 5") "***** ")	;; (<mark>global</mark> -set-key (kbd "; SPC")
2950:(<mark>global</mark> -set-key (kbd "; h 6") "****** ")	;; (lambda () (interactive) (insert ";")))
2952:(<mark>global</mark> -set-key (kbd "; h 7") "****** ")	,, (Tambua () (The active) (There t,)))
2954:(<mark>global</mark> -set-key (kbd "; h 8") "******* ")	;; (<mark>global</mark> -set-key (kbd "; ; SPC") 'ignore)
2963:(global-set-key (kbd " <f12> .") "•")</f12>	, (grobar set key (kba , , ste) ignore)
2964:(<mark>global</mark> -set-key (kbd " <f12> o") "∘")</f12>	(<mark>global</mark> -set-key (kbd "; SPC") 'ignore)
2965:(global-set-key (kbd " <f12> 0") "°")</f12>	(grown set key (kba , sie) ignore)
2969:(<mark>global</mark> -set-key (kbd " <f12> -") "-")</f12>	(<mark>global</mark> -set-key (kbd "; '")
2973:(global-set-key (kbd " <f12> _") "-")</f12>	(lambda () (interactive) (insert ":")))
2977:(<mark>global</mark> -set-key (kbd " <f12> d") "°")</f12>	
2981:(<mark>global</mark> -set-key (kbd " <f12> !") "i")</f12>	:: ** C-z
2982:(<mark>global</mark> -set-key (kbd " <f12> ?") "¿")</f12>	
2986:(<mark>global</mark> -set-key (kbd " <f12> ' o") "'")</f12>	(define-prefix-command 'ring-map-two)
2987:(<mark>global</mark> -set-key (kbd " <f12> ' c") "'")</f12>	(global-set-key (kbd "C-z") 'ring-map-two)
2991:(<mark>global</mark> -set-key (kbd " <f12> \" o") """)</f12>	
2992:(<mark>global</mark> -set-key (kbd " <f12> \" c") """)</f12>	;; * setup
2996:(<mark>global</mark> -set-key (kbd " <f12> \< \< o") "« ")</f12>	
2997:(<mark>global</mark> -set-key (kbd " <f12> \> \> c") "</f12>	(require 'org) ; necessary
3001:(<mark>global</mark> -set-key (kbd " <f12> \< o") "< ")</f12>	
3002:(<mark>global</mark> -set-key (kbd " <f12> \> c") " →")</f12>	:: ** set default load paths
3006:(<mark>global</mark> -set-key (kbd " <f12> a r") "→")</f12>	
3007:(<mark>global</mark> -set-key (kbd " <f12> a 1") "+")</f12>	h c:emacs 61% 2800 Emacs-Lisp s U: /home/marie/.emacs
<pre>*helm occur* L14 (HF) [331 Candidate(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C- wethere a statement of the statement</pre>	h c: - <mark>,emacs</mark> 61% 2800 Emacs-Lisp s U: /home/marie/.emacs
pattern: global	

The **Helm** package provides an incremental completion & narrowing selection API for search applications in Emacs

Virtually every Emacs search application has built-in integration with Helm

Grep equivalent

C-j: Go to Line (keeping session)	'(vertical-border ((((type w32 tty)) (:inherit mode-line-inactive :foreground
HO [.emacs]	"firebrick4"))))
18: '(TeX-auto- <mark>global</mark> "~/.emacs.d/auctex/")	'(vimish-fold-mouse-face ((t (:background "gray15" :foreground "indianred"))))
316: '(<mark>global</mark> -aggressive-indent-mode t)	'(vimish-fold-overlay ((t (:background "MistyRose4" :foreground "black"))))
317: '(<mark>global</mark> -emojify-mode t)	'(warning ((t (:foreground "red2" :weight bold))))
318: '(<mark>global</mark> -emojify-mode-line-mode t)	'(whitespace-empty ((t (:background "light green" :foreground "gray10"))))
319: '(<mark>global</mark> -git-commit-mode t)	'(whitespace-hspace ((t (:foreground "orange red"))))
320: '(global-undo-tree-mode t)	'(whitespace-line ((t (:background "gray15")))
321: (global-visual-line-mode nil)	'(whitespace-space ((t (:foreground "olivedrab"))))
692: '(org-cycle- <mark>global</mark> -at-bob t)	(whitespace-trailing ((t (:background "brown2"))))
2181: '(outshine-org-style-global-cycling-at-bob-p t)	(window-divider ((t (:foreground "gray25"))))
2260: '(smartparens- <mark>global</mark> -mode t)	'(yas-field-highlight-face ((t (:underline t)))))
2261: '(smartparens-global-strict-mode nil)	
2335: '(writeroom- <mark>global</mark> -effects 2338: '(yas- <mark>global</mark> -mode t)	;; * unset env variable MANPAGER in Emacs only
2338: (yas-global-mode t) 2800:(global-set-key (kbd ";") 'ring-map)	;; MANPAGER is set in zsh to sh -c 'col -bx bat -l man -p'
2800:(global-set-key (kbd ;) Fing-map) 2804:;; (global-set-key (kbd "; SPC")	;; which provides bat colour highlighting in the man pages
2807:;; (global-set-key (kbd "; SPC") 'ignore)	;; but this messes up the help files in inferior-python-mode
2809: (global-set-key (kbd "; SPC") 'ignore)	(setenv "MANPAGER" "")
2811:(global-set-key (kbd "; '")	(Setenv Main Addr)
2817:(global-set-key (kbd "C-z") 'ring-map-two)	;; * define new prefix keys
2877:;; useful only when global-emojify-mode is nil	,, where the men pretex keys
2934:(<pre>global</pre> -set-key (kbd " <kp-next>") #'fix-word-downcase) ; F+1 (1=lower case)</kp-next>	;; ** ;
2935:(global-set-key (kbd " <kp-left>") #'fix-word-upcase) ; F+u (u=upper case)</kp-left>	
2936:(global-set-key (kbd " <kp-down>") #'fix-word-capitalize) ; F+k (k=capitalize)</kp-down>	(define-prefix-command 'ring-map)
2940:(global-set-key (kbd "; h 1") "* ")	(global-set-key (kbd ";") 'ring-map)
2942:(<mark>global</mark> -set-key (kbd "; h 2") "** ")	
2944:(<mark>global</mark> -set-key (kbd "; h 3") "*** ")	;; *** make sure I can still type ;
2946:(<mark>global</mark> -set-key (kbd "; h 4") "**** ")	
2948:(<mark>global</mark> -set-key (kbd "; h 5") "***** ")	;; (<mark>global</mark> -set-key (kbd "; SPC")
2950:(global-set-key (kbd "; h 6") "****** ")	;; (lambda () (interactive) (insert ";")))
2952:(global-set-key (kbd "; h 7") "******* ")	
2954:(<mark>global</mark> -set-key (kbd "; h 8") "******** ") 2963:(<mark>global</mark> -set-key (kbd " <f12> .") ".")</f12>	;; (<mark>global</mark> -set-key (kbd "; ; SPC") 'ignore)
2963:(global-set-key (kbd <fl2> .)) 2964:(global-set-key (kbd "<fl2> o") "°")</fl2></fl2>	
2964:(global-set-key (kbd " <fl2> 0") "0") 2965:(global-set-key (kbd "<fl2> 0") "0")</fl2></fl2>	(<mark>global</mark> -set-key (kbd "; SPC") 'ignore)
2969:(global-set-key (kbd " <f12> 0") "0") 2969:(global-set-key (kbd "<f12> -") "-")</f12></f12>	
2909:(global-set-key (kbd <f12> -) -) 2973:(global-set-key (kbd "<f12> _") "-")</f12></f12>	(global-set-key (kbd "; '")
2977: (global-set-key (kbd " <f12> d") "°")</f12>	(lambda () (interactive) (insert ";")))
2981:(global-set-key (kbd " <f12> ") ")</f12>	
2982: (global-set-key (kbd " <f12> ?") "¿")</f12>	;; ** C-z
2986: (global-set-key (kbd " <f12> ' o") "'")</f12>	(define-prefix-command 'ring-map-two)
2987: (global-set-key (kbd " <fl2> ' c") ",")</fl2>	(global-set-key (kbd "C-z") 'ring-map-two)
2991:(<mark>global</mark> -set-key (kbd " <f12> \" o") """)</f12>	(grobal set key (kbd e 2) fing map (wb)
2992:(<mark>global</mark> -set-key (kbd " <f12> \" c") """)</f12>	;; * setup
2996:(<mark>global</mark> -set-key (kbd " <f12> \< \< o") "« ")</f12>	,, , , , , , , , , , , , , , , , , , ,
2997:(<mark>global</mark> -set-key (kbd " <f12> \> \> c") " »")</f12>	(require 'org) ; necessary
3001:(<mark>global</mark> -set-key (kbd " <f12> \< o") "< ")</f12>	, necessary
3002:(<mark>global</mark> -set-key (kbd " <f12> \> c") "</f12>	;; ** set default load paths
3006:(<mark>global</mark> -set-key (kbd " <f12> a r") "→")</f12>	
3007: (global-set-key (kbd " <f12> a l") "€")</f12>	lat ((default-directory merc_d(e)pa("))
<pre>*helm occur* L14 (HF) [331 Candidate(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C-h c: pattorn: glabal</pre>	emacs 61% 2800 Emacs-Lisp s U: /home/marie/.emacs
pattern: global	

Navigate mark ring

C-j: Show this line (keeping session)	(with-eval-after-load 'helm
mark-ring	(define-key helm-map (kbd "; k")
3173: ;; ** helm kmacro 2788: ;; * unset env variable MANPAGER in Emacs only	without leaving helm (define-key helm-map (kbd "; s")
5781: ;; *** kill all slack buffers with ; q	<pre>(define-key helm-map (kbd "; s") 'helm-buffer-save-persistent) ; save buffer without leaving helm</pre>
2819: ;; * setup	(define-key helm-map (kbd "; n") 'helm-buffer-run-rename-buffer) ; will not rename
3060: ;; ** SMeX	the file
5449: (defface my-shortcode-face `((t (:foreground "dim gray"))) "Hugo shortcodes face")	(define-key helm-map (kbd "; h")
3185: ;; * goto	(define-key helm-map (kbd "; d")
5473: ;; **** global	(define-key helm-map (kbd "; a") 'helm-toggle-all-marks)
global-mark-ring	(define-key helm-map (kbd "; t")
3173:.emacs: ;; ** helm kmacro	helm kill ring which do not truncate by default
176:_index.md: Additionally, quoting the ripgrep README, rg "is built on top of Rus"	. (define-key helm-map (kbd "; =") 'helm-buffer-run-ediff) ; ediff current buffer with candidate. With two marked buffers, ediff those buffers
1:sketch.org: <empty_line> 1:alias.zsh: # For a full list of active aliases, run `alias`</empty_line>	(define-key helm-map (kbd "; m m") 'helm-buffer-run-query-replace) ; query replace in
. I.allas.zsn. # for a full list of active allases, full allas	marked buffers
	(define-key helm-map (kbd "; m r") 'helm-buffer-run-query-replace-regexp) ; query
	replace regexp in marked buffers (define-key helm-map (kbd "; r")
	(define-key helm-map (kbd "; i") 'helm-buffers-mark-similar-buffers))
	;; ** resume helm
	(global-set-key (kbd "; h SPC") 'helm-resume)
	;; ** helm kmacro
	(global-set-key (kbd "; h k") 'helm-execute-kmacro)
	;; ** helm-unicode
	(global-set-key (kbd "; h u") 'helm-unicode)
	;; * fzf
	(global-set-key (kbd "; f s") 'fzf)
	;; * goto
	;; ** navigate current buffer
	;; *** bm ;; bookmarks inside a buffer
	(global-set-key (kbd "; u") 'bm-toggle) (global-set-key (kbd "; e r") 'helm-bm)
Ahelm mark rings L1 (HF) [8 Candidate(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C- pattern:	- emacs 64% 3173 Emacs-Lisp s U: /home/marie/.emacs

Navigate kill ring

_C-j: DoNothing (keeping session) Kill Ring	<pre>'(window-divider ((t (:foreground "gray25")))) '(yas-field-highlight-face ((t (:underline t)))))</pre>	
go to repo /home/marie/parvus/ptmp/git_workshop_collab		
https://www.google.com/search?q=button+quail+chicks&tbm=isch&ved=2ahUKEwjvptG06Mj2AhWbBDQIU []	;; which provides bat colour highlighting in the man pages	
https://www.google.com/url?sa=i&url=https%3A%2F%2Ftwitter.com%2Fthegallowboob%2Fstatus%2F1	;; but this messes up the help files in inferior-python-mode	
(Except for the most degenerate	(setenv "MANPAGER" "")	
	;; * define new prefix keys	
<pre>def train(model, device, train_loader, optimizer, epoch): model.train()</pre>		
<pre>for batch_idx, (data, target) in enumerate(train_loader):</pre>	(define-prefix-command 'ring-map) (global-set-key (kbd ";") 'ring-map)	
<pre>output = model(data) loss = F.nll_loss(output, target)</pre>	;; *** make sure I can still type ;	
loss.backward() optimizer.step() if batch_idx % 10 == 0:	;; (global-set-key (kbd "; SPC") ;; (lambda () (interactive) (insert ";")))	
print('Tr []	;; (global-set-key (kbd "; ; SPC") 'ignore)	
import torch	(global-set-key (kbd "; SPC") 'ignore)	
import torch.nn as nn import torch.nn.functional as F import torch.optim as optim	<pre>(global-set-key (kbd "; '")</pre>	
from torch.optim.lr_scheduler import StepLR	;; ** C-z	
class Net(nn.Module):	(define-prefix-command 'ring-map-two) (global-set-key (kbd "C-z") 'ring-map-two)	
<pre>definit(self): super(Net, self)init()</pre>	;; * setup	
self.conv1 = nn.Conv2d(1, 32, 3, 1) self.conv2 = nn.Conv2d(32, 64, 3, 1)	(require 'org) ; necessary	
<pre>self.dropout1 = nn.Dropout2d(0.25)</pre>	;; ** set default load paths	
[]	<pre>(let ((default-directory "~/.emacs.d/elpa/"))</pre>	
	(setq load-path (append	
	<pre>(let ((load-path (copy-sequence load-path))) ;; Shadow (append</pre>	
import torch.nn as nn import torch.nn.functional as F from torchvision import datasets, transforms	<pre>(append (copy-sequence (normal-top-level-add-to-load-path '("."))) (normal-top-level-add-subdirs-to-load-path))) load-path)))</pre>	
<pre>device = torch.device('cpu')</pre>	;; ** package system	
<pre>#helm kill rings L1 [15 Candidate(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C-h c:C pattern:</pre>	emacs 61% 2788 Emacs-Lisp s U: /home/marie/.emacs	

Navigate open & recent files

skatch.org alids.zb, actigr.zh f): * unset nor variable MMMAGER in Eracs only actigr.zh NAMMAGER is set in at to sh - (° ob - bx ba - 1 man - p' i; which provides bat colour highlighting in the man pages i; which provides bat colour highlighting in the man pages i; which provides bat colour highlighting in the man pages i; which provides bat colour highlighting in the man pages i; which provides bat colour highlighting in the man pages i; which provides bat colour highlighting in the man pages i; * define new prefix keys mode weighting.com i; * i; define new prefix keys sittigrore alids.zbh contact.org	C-1: Show this buffer (keeping session) Buffers _index.md	<pre>'(window-divider ((t (:foreground "gray25")))) '(yas-field-highlight-face ((t (:underline t)))))</pre>
<pre>_remacs move weak year indusers.csw rifle.conf _index.md sketch.org sliggore dontation.gr sketch.org sliggore dontation.gr config z.zshc config y ** make sure I can still type ; y ** feedbal-set-key (kbd *; \$PC^*) 'ignore) (global-set-key (kbd *: \$PC^*) 'ignore) (global-</pre>	sketch.org alias.zsh script.sh	;; MANPAGER is set in zsh to sh -c 'col -bx bat -l man -p' ;; which provides bat colour highlighting in the man pages
move westpridusers.csv if if le.com introde.x.md setter, org setter, org setter, org contact.org conta	Recentf	(setenv "MANPAGER" "")
<pre>rifle.comf index.md sktch.org index.md ind</pre>		
Sitteprofe iffinger Sitteprofe (define-prefix-command 'ring-map) Sitteprofe (global-set-key (kbd "; ') 'ring-map) config ; *** make sure I can still type ; config ; *** make sure I (and still type ; config ; *** make sure I (and still type ; config ; *** make sure I (and still type ; config ; (global-set-key (kbd "; SPC") 'ignore) (insert *; *))) (global-set-key (kbd "; SPC") 'ignore) config ; ** C-z config ; ** C-z config ; ** C-z config ; ** setup (define-prefix-command 'ring-map-two) ; ** setup (global-set-key (kbd "c-c") 'ring-map-two) ; ** setup Kindows ; ** setup crass .eeass ; necessary st-286color, ranger;mage/logo/lwg ; ** set default load paths create buffer (cony-sequence (normal-top-level-add-t		;; * define new prefix keys
<pre>sktteh.org slitspore allas.zeh config config config pulla].z.zhc config pulla].z.z.zhc config pulla].z.z.zhc config pulla].z.z.z.z.z.z.z.z.z.z.z.z.z.z.z.z.z.z.z</pre>	_index.md	:: ** :
alia:sch contact.org contact.org contact.org script.sh bookarks upscaling_data.org covid_vaccination_info.o		
contact.org cproof bet by (Not -) - Hig map) contact.org cproof bet by (Not -) - Hig map) contact.org cproof bet by (Not -) - Hig map) contact.org cproof bet by (Not -) - Hig map) contact.org cproof bet by (Not -) - Hig map) contact.org cproof bet by (Not -) - Hig map) contact.org cproof bet by (Not -) - Hig map) contact.org cproof bet by (Not -) - Hig map) contact.org cproof bet by (Not -) - Hig map) contact.org cproof bet by (Not -) - Hig map) contact.org cproof bet by (Not -) - Hig map) contact.org cproof bet by (Not -) - Hig map) contact.org cproof bet by (Not -) - Hig map) contact.org (global-set-key (Mod *; strC) 'ignore) contact.org (global-set-key (Kbd *; strC) 'ignore) inux.org (global-set-key (Kbd *; strC) 'ignore) contact.org (global-set-key (Kbd *; strC) 'ignore) inux.org (global-set-key (Kbd *; strC) 'ignore) contact.org (global-set-key (Kbd *; strC) 'ignore) inux.org (global-set-key (Kbd *; strC) 'ignore) contact.org (global-set-key (Kbd *; strC) <t< th=""><th></th><th></th></t<>		
<pre>,rshr.d ,rshr.d ,</pre>		(global-set-key (kbd ";") "ring-map)
script:sh julia:j bokmarks upscaling.data.org covid.vaccination_info.corg covid.vaccination_info.cov corg.vaccination_info.co		;; *** make sure I can still type ;
<pre>julia.j1 bookmarks upscaling_data.org config.py linux.org random.org passphrase.ppg .xinitre fontlist-v330.json test.R monthly_script.sh weekly_script.sh X Windows st-256color.man rg st-256color.man rg st-256color.man rg create buffer hele_task_weith_11 [S BUFfer(s)]] C-h msHeip TAS:Act RET/f1/f2/f=nstthact C+1:Togs.suppend C-h c:com = means st-256color.man rg st-256color.man rg transection = the state of the st</pre>		
<pre>bookmarks proceding_data.org covid_vaccination_info.org covid_vaccinat</pre>		
<pre>covid_vaccination_info.org covid_vaccination_info.org covid_vaccination_info.org covid_vaccination_info.org covid_vaccination_info.org covid_vaccination_info.org covid_vaccination_info.org (global-set-key (kbd "; SPC") 'ignore) (lambda (interactive) (insert ";"))) (lambda (interactive) (insert ";"))) (interactive) (insert ";")) (interactive) (insert ";")) (interactive) (insert ";")) (interactive) (insert ";")) (interactive) (interactive) (insert ";")) (interactive) (inse</pre>	bookmarks	;; (lambda () (interactive) (insert ;)))
config.py (global-set-key (kbd "; SPC") 'ignore) linux.org (global-set-key (kbd "; SPC") 'ignore) random.org (global-set-key (kbd "; '') massphrase.gpg (global-set-key (kbd "; '') xinitrc (global-set-key (kbd "; '') fontlist-v330.json ;; ** C-Z test.R (define-prefix-command 'ring-map-two) (global-set-key (kbd "c-z") 'ring-map-two) ;; ** setup wonthly.script.sh ;; ** setup X Windows (require 'org) ; necessary st-256color. ranger:image/logo/lwg ;; ** set default load paths Enacsemacs .emacs st-256color. marie@prosoitos:~ (let ((default-directory "-/.emacs.d/elpa/")) st-256color. mar rg (defor-prefix-command 'cop-level-add-to-load-path '(","))) Create buffer (orgonal-top-level-add-subdirs-to-load-path)); ; Shadow (append (copy-sequence (normal-top-level-add-to-load-path))) (is the full '1' [S Buffer(S)] C-h m:Help TAB:Act RET/f1/f2/f=n:NthAct C=12Tog.suspend C=h c:Con = _emacs [St 2788 Emacs-Lisp s U: /home/marie/.emacs	upscaling_data.org	;; (global-set-key (kbd "; ; SPC") 'ignore)
<pre>linux.org passphrase.gpg .x.initrc forlist-v320.json test.R c.org julia-11-shared-arrays.md config monthly_script.sh Windows t v256color. ranger:image/logo/lwg Emacs .emacs t v256color. marie@prosoitos:- qutebrowser.0 top Inbox - marie.burle@westgrid.ca - Compute Canada Inc Mail st-256color. mar grep t v256color. mar grep t</pre>		
<pre>passphrase.gpg .xinitrc forlist-v330.json test.R mc.org julia-11-shared-arrays.md config monthly_script.sh weekly_script.sh X Windows st-256color. marie@prosoitos:~ st-256color. marie@prosoitos:~ st-256color. marie@prosoitos:~ st-256color. marig@prosoitos:~ st-256color. mar</pre>	linux.org	(global-set-key (kbd "; SPC") 'ignore)
<pre>passphrase.gpg .xinitrc fontlist-v330.json test.R mc.org julia-11-shared-arrays.md config monthly_script.sh weekly_script.sh X Windows st-256color. ranger:image/logo/lwg Emacs .emacs st-256color. marie@prosoitos:~ st-256color. marie@prosoitos:</pre>	random.org	(global-set-kev (kbd ": '")
fontlist-v330.json test.R mc.org julia-11-shared-arrays.md config monthly_script.sh weekly_script.sh weekly_script.sh X Windows \$t-256color. ranger:image/logo/lwg Emacs .emacs st-256color. marie@prosoitos:- qutebrowser.0 top Inbox - marie.burle@westgrid.ca - Compute Canada Inc Mail st-256color. man rg Create buffer Create buffer thelm-teskgwitch_L1 [5 Buffer(s)] C-h m:Help TAB:Act REI/f1/f2/f=n:NthAct C-1:tog.suspend C-h c:Con - Lemacs I (2788 Emacs-Lisp s U: /home/marie/.emacs	passphrase.gpg	
<pre>test.R mc.org julia-11-shared-arrays.md config monthly_script.sh weekly_script.sh X Windows st-256color. ranger:image/logo/lwg Emacsemacs st-256color. marie@prosoitos:~ qutebrowser. 0 top Inbox - marie.burle@westgrid.ca - Compute Canada Inc Mail st-256color. marie@prosoitos:~ qutebrowser. 0 top Inbox - marie.burle@westgrid.ca - Compute Canada Inc Mail st-256color. man rg Create buffer Create buffer Mthem-task_switch4_L1 [S Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-1:Tog.suspend C-h c:Con - temacs Mthem-task_switch4_L1 [S Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-1:Tog.suspend C-h c:Con - temacs Mthem-task_switch4_L1 [S Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-1:Tog.suspend C-h c:Con - temacs Mthem-task_switch4_L1 [S Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-1:Tog.suspend C-h c:Con - temacs Mthem-task_switch4_L1 [S Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-1:Tog.suspend C-h c:Con - temacs Mthem-task_switch4_L1 [S Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-1:Tog.suspend C-h c:Con - temacs Mthem-task_switch4_L1 [S Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-1:Tog.suspend C-h c:Con - temacs Mthem-task_switch4_L1 [S Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-1:Tog.suspend C-h c:Con - temacs Mthem-task_switch4_L1 [S Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-1:Tog.suspend C-h c:Con - temacs Mthem-task_switch4_L1 [S Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-1:Tog.suspend C-h c:Con - temacs Mthem-task_switch4_L1 [S Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-1:Tog.suspend C-h c:Con - temacs Mthem-task_switch4_L1 [S Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-1:Tog.suspend C-h c:Con - temacs Mthem-task_switch4_L1 [S Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-1:Tog.suspend C-h c:Con - temacs Mthem-task_switch4_L1 [S Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-1:Tog.suspend C-h c:Con - temacs Mthem-task_switch4_L1 [S Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-1:To</pre>		
julia-11-shared-arrays.md (definite prefix for the		;; ** L-Z
Julia-li-shared-arrays.md config monthly_script.sh weekly_script.sh X Windows st-256color. ranger:image/logo/lwg Emacs .emacs st-256color. marie@prosoitos:~ qutebrowser. 0 top Inbox - marie.burle@westgrid.ca - Compute Canada Inc Mail st-256color. marie@prosoitos:~ gtelfor. marie@prosoitos:~ gtelfor. marie@prosoitos:~ (let ((default-directory "~/.emacs.d/elpa/")) (setq load-path (append (let ((load-path (copy-sequence load-path))) ;; Shadow (append (copy-sequence (normal-top-level-add-to-load-path '("."))) (normal-top-level-add-to-load-path))) ;; ** package system *helm-taskgwitchs L1 [5 Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f=n:NthAct C-1:Tog.suspend C-h c:Con - emacs 61% 2788 Emacs-Lisp s U: /home/marie/.emacs	mc.org	(define-prefix-command 'ring-map-two)
<pre>monthly_script.sh weekly_script.sh weekly_script.sh weekly_script.sh weekly_script.sh it (if the integrate of the integr</pre>		
<pre>weekly_script.sh X Windows st-256color. ranger:image/logo/lwg Emacs .emacs .emacs st-256color. marie@prosoitos:- qutebrowser. 0 top Inbox - marie.burle@westgrid.ca - Compute Canada Inc Mail st-256color. man rg Create buffer Create buffer Ahelm-taskgwitchs L1 [5 Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f=n:NthAct C-1:Tog.suspend C-h c:Con -].emacs 61% 2788 Emacs-Lisp s U: /home/marie/.emacs</pre> (require 'org) ; necessary ;; * setUp (require 'org) ; necessary ;; ** setUp (require 'org) ; necessary ;; ** set default load paths (let ((default-directory "-/.emacs.d/elpa/")) (setq load-path (append (copy-sequence (normal-top-level-add-to-load-path '("."))) load-path))) ;; ** package system		
<pre>X Windows St-256color. ranger:image/logo/lwg Emacs .emacs st-256color. marie@prosoitos:- qutebrowser. 0 top Inbox - marie.burle@westgrid.ca - Compute Canada Inc Mail st-256color. marie@prosoitos:- st-256color. man grep st-256color. man grep st-256color. man rg Create buffer (let ((default-directory "-/.emacs.d/elpa/")) (setq load-path (append (let ((load-path (copy-sequence load-path)));; Shadow (append (copy-sequence load-path)));; Shadow (append (copy-sequence (normal-top-level-add-to-load-path '("."))) (normal-top-level-add-subdirs-to-load-path))) ;; ** package system *helm-taskawitchs L1 [5 Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C-h c:Con - emacs 61% 2788 Emacs-Lisp s U: /home/marie/.emacs</pre>	weekly_script.sh	;; * Setup
<pre>st-256color. ranger:image/logo/lwg Emacsemacs st-256color. marie@prosoitos:~ qutebrowser. 0 top Inbox - marie.burle@westgrid.ca - Compute Canada Inc Mail st-256color. man grep st-256color. man rg Create buffer</pre> (let ((default-directory "-/.emacs.d/elpa/")) (setq load-path (append (copy-sequence load-path))); Shadow (append (copy-sequence (normal-top-level-add-to-load-path '("."))) load-path))) ;; ** package system	Y Windows	(require 'org) ; necessary
<pre>st-256color. marie@prosoitos:- qutebrowser. 0 top Inbox - marie.burle@westgrid.ca - Compute Canada Inc Mail st-256color. marie@prosoitos:- st-256color. man grep st-256color. man grep st-256colo</pre>	st-256color. ranger:image/logo/lwg	;; ** set default load paths
<pre>quterowser. 0 top Inbox - marie.burieewestgrid.ca - Compute Canada Inc Mail st-256color. man grep st-256color. man grep st-256color. man rg Create buffer Create buffer Ahelm-taskawitchs L1 [5 Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C-h c:Con - emacs 61% 2788 Emacs-Lisp s U: /home/marie/.emacs</pre>	st-256color. marie@prosoitos:~	(let ((default-directory "~/ emacs d/elpa/"))
<pre>st-z56color. man grep st-z56color. man grep st-z56color. man grep create buffer (load-path (copy-sequence load-path)));; Shadow (append (copy-sequence (normal-top-level-add-to-load-path))); (normal-top-level-add-subdirs-to-load-path))) load-path))) ;; ** package system *helm-taskswitchs L1 [5 Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C-h c:Conemacs 61% 2788 Emacs-Lisp s U: /home/marie/.emacs</pre>	qutebrowser. 0 top Inbox - marie.burle@westgrid.ca - Compute Canada Inc Mail	
<pre>st-256color. man rg Create buffer Create buffer (let ((load-path (copy-sequence load-path))) ;; Shadow (append (copy-sequence (normal-top-level-add-to-load-path '("."))) (normal-top-level-add-subdirs-to-load-path))) load-path))) ;; ** package system *helm-taskswitchs L1 [5 Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C-h c:Conemacs 61% 2788 Emacs-Lisp s U: /home/marie/.emacs</pre>		(append
Create buffer (copy-sequence (normal-top-level-add-to-load-path '("."))) (normal-top-level-add-subdirs-to-load-path))) load-path))) ioad-path))) ioad-path))) *helm-taskswitchs L1 [5 Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C-h c:Conemacs 61% 2788 Emacs-Lisp s U: /home/marie/.emacs	st-256color. man grep	
<pre>Create buffer (normal-top-level-add-subdirs-to-load-path))) load-path))) ;; ** package system *helm-taskswitchs L1 [5 Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C-h c:Conemacs 61% 2788 Emacs-Lisp s U: /home/marie/.emacs</pre>		
;; ** package system *helm-taskswitch* L1 [5 Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C-h c:Conemacs 61% 2788 Emacs-Lisp s U: /home/marie/.emacs	Create buffer	<pre>(normal-top-level-add-subdirs-to-load-path)))</pre>
Anelm-taskswitchs L1 [5 Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C-h c:Con - emacs 61% 2788 Emacs-Lisp s U: /home/marie/.emacs		load-path)))
		;; ** package system
	<pre>*helm-taskswitch* L1 [5 Buffer(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C-h c</pre>	
	pattern:	

Navigate recent files

C-j: Show this file (keeping session)	(global-set-key (kbd "; b s") 'bookmark-save)
c-j: Show this file (keeping session) Recentf	(global set key (kbd ; b s) bookilark-save)
<pre>//emacs</pre>	;; **** jump to bookmark and recenter
<pre>~/parvus/ptmp/dl/move-westgrid-users.csv</pre>	,, where Jump to bookhark and recenter
~/.config/ranger/rifle.conf	(defun my-bookmark-jump (bookmark &optional display-func)
<pre>~/parvus/prog/slides/content/tools/_index.md</pre>	(interactive
~/parvus/adm/notes/sketch.org	(list (bookmark-completing-read "Jump to bookmark"
~/.gitignore	bookmark-current-bookmark)))
~/.zsh/custom/alias.zsh	(unless bookmark
~/parvus/adm/notes/contact.org	(error "No bookmark specified"))
~/.config/bat/config	(bookmark-maybe-historicize-string bookmark)
~/.zshrc	(bookmarkjump-via bookmark (or display-func 'pop-to-buffer-same-window))
<pre>~/parvus/prog/slides/static/files/tools/script.sh</pre>	(recenter-top-bottom))
~/parvus/prog/slides/static/files/tools/julia.jl ~/.emacs.d/bookmarks	
<pre>~/.emacs.c/bookmarks ~/parvus/prog/tml/content/webinars/upscaling_data.org</pre>	(global-set-key (kbd "; e k") 'my-bookmark-jump)
<pre>~/parvus/adm/medical/vaccination/covid/covid_vaccination_info.org</pre>	an anti-directory of average buffer
<pre>~/.config/qutebrowser/config.py</pre>	;; ** goto directory of current buffer
<pre>~/parvus/adm/notes/linux.org</pre>	(fset 'my-go-to-current-directory
//parvus/adm/notes/random.org	[?\M-x ?i ?d ?o ? return return])
<pre>~/.password-store/passphrase.gpg</pre>	
~/.xinitrc	(global-set-key (kbd "; C-l") 'my-go-to-current-directory)
~/.cache/matplotlib/fontlist-v330.json	
~/parvus/prog/progtest/test.R	:: ** goto github
~/parvus/adm/notes/mc.org	· · · · · · · · · · · · · · · · · · ·
~/parvus/prog/courses/content/julia202202/julia-11-shared-arrays.md	(global-set-key (kbd "; h v f") 'helm-open-github-from-file) ; v = version (for version
~/.config/i3/config	control)
~/hbin/btimers/monthly_script.sh	
~/hbin/btimers/weekly_script.sh	(global-set-key (kbd "; h v c") 'helm-open-github-from-commit)
	;; ** search regexp in content
	;; **** within current buffer
	(defun my-helm-occur ()
	(interactive)
	(helmadvice-push-mark)
	(helm-occur))
	(global-set-key (kbd "; j") 'my-helm-occur)
	;; *** within open buffers (ag)
	;; search within content of buffers
	;; **** all open buffers
	(global-set-key (kbd "; h a b") 'helm-do-ag-buffers) ; helm buffers
	;; **** open org buffers
	(global cot how (kbd " h a a") their and nifle)
<pre>#helm recentf# L1 [27 File(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C-h c:Conf pattern:</pre>	emacs 66% 3313 Emacs-Lisp s U: /home/marie/.emacs

Find file

C-j: Hit1 Expand Candidate, Hit2 or (C-u) Find file (keeping session)	(helmadvice-push-mark)
Read File Name (C-1: Go up one level)	(helm-occur))
/home/marie/. /home/marie/	
.Rprofile	(global-set-key (kbd "; j") 'my-helm-occur)
Xauthority	the site of the second buffers (sec)
Xresources	;; *** within open buffers (ag)
.abcde.conf	;; search within content of buffers
android	;; **** all open buffers
aspell.en.prepl	;; AAAA att open burrers
.aspell.en.pws	(global-set-key (kbd "; h a b") 'helm-do-ag-buffers) ; helm buffers
.aspell.fr.prepl	(global set key (kbu , if a b) helm do ag burrers), helm burrers
.aspell.fr.pws	:: **** open org buffers
.authinfo.gpg	,, and open of g barrers
.bash_profile	(global-set-key (kbd "; h a o") 'helm-org-rifle)
.bashrc	
.borg	;; *** within files (ag)
. cache	;; search within content of files (no need to be open)
.cargo	
.cht.sh	:: **** in directory
.config	
, emacs	(global-set-key (kbd "; h a d") 'helm-do-ag)
.emacs-bmk-bmenu-state.el	;; defaults to current directory, but path can be edited
.emacs.d	
. fasd	;; **** org files in directory
.git	
.gitconfig .gitignore	(global-set-key (kbd "; h a r") 'helm-org-rifle-directories)
.gnome	;; with prefix: recursively
.gnupg	
.googleearth	;; **** in current project
gtkrc-2.0	
.gtkrc-2.0.mine	(global-set-key (kbd "; h a p") 'helm-do-ag-project-root) ; helm project
icons	
.inputrc	;; ** search file names
.ipython	
.julia	;; *** current directory
.jupyter	;; search files in current directory
.less	;; directory can be changed
.lesshst	(global-set-key (kbd "; f f") 'helm-find-files)
.lesskey	
.llpp_my_kbd.ml	;; *** open buffers
.llpp_original_kbd.ml	;; search among buffer names (not content)
local	, search among sarrer names (not concent)
mbsyncrc	(global-set-key (kbd "; i") 'helm-taskswitch)
.msmtp.log	
.msmtprc	(require 'helm-taskswitch) ; not necessary to load the package, but necessary
.npm .nv	for the function below not to break it
.nv .nvidia-settings-rc	
.password-store	
.password-score *Helm file completions* L1 [92 Candidate(s)] C-h m:Help C/M-RET:Empty TAB:Act RET/f1/f2/f-n:NthAct C-	emacs 66% 3347 Emacs-Lisp s U: /home/marie/.emacs
Search in file(s): /home/marie/	

Locate

C-j: helm-ff-kill-or-find-buffer-fname (keeping session)	;; *** current directory
	;; search files in current directory
	;; directory can be changed
/etc/abcde.conf	
/etc/conf.d	(global-set-key (kbd "; f f") 'helm-find-files)
/etc/daxctl.conf.d	
/etc/dconf	;; *** open buffers
/etc/dhcpcd.conf	;; search among buffer names (not content)
/etc/e2scrub.conf	, search among burrer names (not content)
/etc/fuse.conf	(global-set-key (kbd "; i") 'helm-taskswitch)
/etc/gai.conf	(giobal-sel-key (kbu), i) heim-taskswitch)
/etc/grc.conf	(require their techemitch)
/etc/healthd.conf	<pre>(require 'helm-taskswitch) ; not necessary to load the package, but necessary for the function below not to break it</pre>
/etc/host.conf	for the function below not to break it
/etc/isstatus.conf	
/etc/krb5.conf	(defun helm-taskswitch ()
/etc/ld.so.conf	"Use helm to switch between tasks (X11 windows, buffers or recentf)."
/etc/ld.so.conf.d	(interactive)
/etc/libao.comf	(run-hooks 'helm-taskswitch-open-hooks)
/etc/libadit.conf	(select-frame-set-input-focus (window-frame (selected-window)))
/etc/libva.conf	(unless helm-source-buffers-list
/etc/locale.conf	(setq helm-source-buffers-list
/etc/makepkg.conf	<pre>(helm-make-source "Buffers" 'helm-source-buffers)))</pre>
/etc/man_db.conf	(helm :sources '(helm-source-buffers-list
/etc/mdadm.conf	helm-source-recentf
/etc/mke2fs.conf	helm-taskswitch-source-x-windows
	helm-source-buffer-not-found)
/etc/ndctl.conf.d	<pre>buffer "*helm-taskswitch*"</pre>
/etc/netconfig	:truncate-lines t))
/etc/netconf	
/etc/nsswitch.conf	;; *** current git project
/etc/nts.conf	
/etc/pacman.conf	(global-set-key (kbd "; h p") 'helm-cmd-t)
/etc/resolv.conf	;; *** git of files in current git project
/etc/resolv.conf.bak	;; using helm-ls-git package
/etc/resolvconf.conf	
/etc/sensors3.conf	(global-set-key (kbd "; h g") 'helm-browse-project)
/etc/sudo_logsrvd.conf	;; *** all files in computer
/etc/ts.conf	;; equivalent to: locate -i -e -Aregex
/etc/updatedb.conf	
/etc/xattr.conf	
/etc/R/Makeconf	(global-set-key (kbd "; h l") 'helm-locate)
Inter (D) linear and	
/etc/UPower/UPower.conf	;; ** search email
/etc/X11/xorg.conf.d	
/etc/audit/audisp-remote.conf	(global-set-key (kbd "; m s") 'helm-mu) ; mu search
/etc/audit/auditd.conf	
/etc/audit/zos-remote.conf	(global-set-key (kbd "; m c") 'helm-mu-contacts)
/etc/avahi/avahi-daemon.conf *helm locate* L1	emacs 66% 3383 Emacs-Lisp s U: /home/marie/.emacs
	- Mindes 00% 3563 Emacs-Elsp 5.0. /Mome/marie/.emacs
pattern: conf	

Search objects in working environment

C-j: Show this entry (keeping session)	'(window-divider ((t (:foreground "gray25"))))
Imenu	'(yas-field-highlight-face ((t (:underline t))))
Types / my-shortcode-face	
Types / my-nested-org-block-face	;; * unset env variable MANPAGER in Emacs only
Variables / aw-dispatch-alist	;; MANPAGER is set in zsh to sh -c 'col -bx bat -l man -p'
Function / custom set variables	;; which provides bat colour highlighting in the man pages
Function / custom set faces	;; but this messes up the help files in inferior-python-mode
Function / unset env variable MANPAGER in Emacs only	
Function / define new prefix keys	
Function /;	
Function / make sure I can still type ;	;; * define new prefix keys
Function / C-z	
Function / setup	
Function / set default load paths	
Function / package system	(define-prefix-command 'ring-map)
Function / prevent series of error messages in minibuffer due to slack	(global-set-key (kbd ";") 'ring-map)
Function / change yes-no questions to y-n	
Function / allow a selected area to be overwritten	;; *** make sure I can still type ;
Function / start emacs server	
Function / emojify emacs lisp mode	;; (global-set-key (kbd "; SPC")
Function / font size	;; (lambda () (interactive) (insert ";")))
Function / frame	
Function / create new frames without scrollbars and in fullscreen Function / mv/disable-scroll-bars	;; (global-set-key (kbd "; ; SPC") 'ignore)
Function / my/disable-scroll-bars	
Function / disable menubar Function / set frame title to buffer name	(global-set-key (kbd "; SPC") 'ignore)
Function / kill buffer when frame is killed	(global-set-key (kbd "; '")
Function / my-kill-buffer-with-frame	<pre>(lambda () (interactive) (insert ";")))</pre>
Function / explicitly load helm-mode early to prevent new error message Function / fix word	
Function / headings	;; ** C-z
Function / inicode	
Function / dot	(define-prefix-command 'ring-map-two)
Function / m-dash	(global-set-key (kbd "C-z") 'ring-map-two)
Function / n-dash	
Function / degree	;; * setup
Function / Spanish punctuation	
Function / single quotes	(require 'org) ; necessary
Function / double quotes	
Function / French quotes	;; ** set default load paths
Function / French single quotes	
Function / French Single quotes	(let ((default-directory "~/.emacs.d/elpa/"))
Function / mathematical signs	
Function / beginning of text on line	Cappend
Function / reload buffer after file modification without confirmation	<pre>(let ((load-path (copy-sequence load-path)));; Shadow</pre>
Function / revert-buffer-no-confirm	(append
Function / ido and smex	<pre>(copy-sequence (normal-top-level-add-to-load-path '(".")))</pre>
Function / ido completing read+ and ido ubiquitous	<pre>(normal-top-level-add-subdirs-to-load-path)))</pre>
Function / ido completing for some functions	load-path)))
Function / ido vertical mode	
Function / flx ido	;; ** package system
<pre>*helm imenu* L6 (HF) [100 Candidate(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C-h c:</pre>	emacs 61% 2788 Emacs-Lisp s U: /home/marie/.emacs
pattern:	

Search in Git repo

C-j: Diff (keeping session)	:: *** current directory
Git status (m)	;; search files in current directory
M .aspell.en.pws	; directory can be changed
M .config/llpp.conf	
M .config/qutebrowser/config.py	(global-set-key (kbd "; f f") 'helm-find-files)
M .config/qutebrowser/quickmarks	
M .config/ranger/rc.conf	:: *** open buffers
M .config/ranger/rifle.conf	;; search among buffer names (not content)
M .emacs	
M .zsh/custom/alias.zsh	(global-set-key (kbd "; i") 'helm-taskswitch)
?? .config/bat/	
<pre>?? .config/ranger/initial_rifle.conf</pre>	(require 'helm-taskswitch); not necessary to load the package, but necessary
	for the function below not to break it
Git branches	
* m: a4141ef ranger: important commit: massive changes to riffle.conf: the file setting op	
Buffers in git project (m)	
_index.md	
sketch.org	(select-frame-set-input-focus (window-frame (selected-window)))
alias.zsh	(unless helm-source-buffers-list
script.sh	(setq helm-source-buffers-list
.emacs	(helm-make-source "Buffers" 'helm-source-buffers))
	(helm : sources '(helm-source-buffers-list
Git files (m)	helm-source-recentf
.config/autokey/data/phrase/1.txt	
.config/autokey/data/phrase/archive/smileys/1.txt .config/autokey/data/phrase/archive/smileys/2.txt	helm-source-buffer-not-found)
.config/autokey/data/phrase/archive/smileys/2.txt	:buffer "*helm-taskswitch*"
.config/autokey/data/phrase/archive/smileys/3.txt	:truncate-lines t))
.config/autokey/data/phrase/archive/smileys/5.txt	
.config/autokey/data/phrase/archive/smileys/D.txt	;; *** current git project
.config/autokey/data/phrase/archive/smileys/b.txt	
.config/autokey/data/phrase/archive/smileys/o.txt	(global-set-key (kbd "; h p") 'helm-cmd-t)
.config/autokey/data/phrase/archive/animal/pp.txt	and sit of files is succeed ait seriest
.config/autokey/data/phrase/archive/animal/sc.txt	;; *** git of files in current git project ;; using helm-ls-git package
.config/autokey/data/phrase/archive/animal/st.txt	;; using heim-is-git package
.config/autokev/data/phrase/archive/animal/ts.txt	(global-set-key (kbd "; h g") 'helm-browse-project)
.config/autokey/data/phrase/craigslist/cl.txt	(grobal-set-key (kbd , if g) iterm-blowse-ploject)
.config/autokey/data/phrase/date_and_time/day.py	;; *** all files in computer
.config/autokey/data/phrase/expression/eg.txt	;; equivalent to: locate -i -e -Aregex
.config/autokey/data/phrase/expression/er.txt	;; regexp
.config/autokey/data/phrase/acronym/hpc.txt	
.config/autokey/data/phrase/archive/animal/vta.txt	(global-set-key (kbd "; h l") 'helm-locate)
.config/autokey/data/phrase/archive/animal/vtu.txt	
.config/autokey/data/phrase/archive/names/dov.txt	;; ** search email
.config/autokey/data/phrase/archive/names/mag.txt	
.config/autokey/data/phrase/email_address/sfu.txt	(global-set-key (kbd "; m s") 'helm-mu) ; mu search
.config/autokey/data/phrase/emoji_for_emacs_slack_reactions/boo.txt	
.config/autokey/data/phrase/expression/bcp.txt	(global-set-key (kbd "; m c") 'helm-mu-contacts)
.config/autokey/data/phrase/expression/btw.txt	
.config/autokey/data/phrase/acronym/DHSI.txt	se coach halp
<pre>*helm lsgit* L1 [10 Candidate(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C-h c:Conf pattern:</pre>	emacs 66% 3383 Emacs-Lisp s U: /home/marie/.emacs

Manage major/minor modes

minimica ; ** scarch enail trick is scarch enail trick (global:set-kay (kbd *; n s') *helm-ma); nu search trick (global:set-kay (kbd *; n s') *helm-ma); nu search trick (global:set-kay (kbd *; n s') *helm-ma); nu search trick (global:set-kay (kbd *; n s') *helm-ma); nu search trick (global:set-kay (kbd *; n s') *helm-ma); nu search trick (global:set-kay (kbd *; n s') *helm-ma); nu search trick (global:set-kay (kbd *; n s') *helm-ma); nu search trick (global:set-kay (kbd *; n n s') *helm-ma); nu search trick (global:set-kay (kbd *; n n s') *helm-ma); nu search trick (global:set-kay (kbd *; n n s') *helm-ma); nu search trick (global:set-kay (kbd *; n n s') *helm-ma); nu search trick (global:set-kay (kbd *; n n s') *helm-mapropos); helm help info trick (global:set-kay (kbd *; n n s') *helm-mapropos); helm help apropos trick (global:set-kay (kbd *; n n s') *helm-mapropos); helm help apropos trick (global:set-kay (kbd *; n n s') *helm-mapropos); helm help apropos trick (global:set-kay (kbd *; n n s') *helm-mapropos); helm help apropos trick (global:set-kay (kbd *; n n s') *helm-mapropos); helm help apropos trick (global:set-kay (kbd *; n n s') *helm-mapropos); helm help apropos trick <t< th=""><th>C-j: Describe major mode (keeping session)</th><th></th></t<>	C-j: Describe major mode (keeping session)	
Addition Search enail City minor desc [cibal-set-key (kid '; m s') 'helm-mu-contacts) Where our package mode [cibal-set-key (kid '; m s') 'helm-mu-contacts) Where our package mode ;; ** search help Where our package mode ;; ** search Where our package mode ;; ** sear	Major mode	(global-set-key (kbd "; h l") 'helm-locate)
<pre>status mior modes here-mode dubits - was profile mode dubits - was profile mode dubits - was profile mode dubits - was profile mode with - compressions - compressions mode with - compressions - co</pre>	emacs-lisp-mode	
bbbevended (global-set-key (kdd *; n s*) 'helm-mu); mu search ggressive-inden-mode (global-set-key (kdd *; n s*) 'helm-mu-contacts) syn-bytecom-preckage-mode ;; ** search syn-bytecom-preckage-mode ;; ** search ubcompany-mode ;; ** info ubcompany-mode (global-set-key (kdd *; h h i*) 'helm-info); helm help info ubcompany-mode ;; ** info ubcompany-mode (global-set-key (kdd *; h h a*) 'helm-apropos); helm help apropos ubcompany-mode ;; ** info ubcompany-mode ;; *** info ubcompany-mode ;; *** info ubcompany-mode ;; *** info ubcompany-mode ;; *** info ubco-dinint-mode ;; *** info		;; ** search email
digitive wrapperfixed (global set.wg (bd '; m c') 'helm-mucontacts) sync bytecompackage mode (global set.wg (bd '; m c') 'helm-mucontacts) sync bytecompackage mode ;; ** search help sompacy-mode ;; are spot the code of the function at point sompacy-mode ;; are standmode sompacy-mode ;; are standset sompacy-mode<		
spire store (global-set-key (kbd *; n c*) 'helm-mu-contacts) spire bytecomparison mode ; ** sect help uuto-compasion mode </td <th></th> <td>(global-set-key (kbd "; m s") 'helm-mu) ; mu search</td>		(global-set-key (kbd "; m s") 'helm-mu) ; mu search
<pre>isjnc-tytecomp-package mode uuto-composition-mode uuto-composition-mode uuto-composition-mode uuto-composition-mode uuto-composition-mode uuto-composition-mode uuto-composition-mode uuto-composition-mode uuto-composition-mode uuto-composition-mode uuto-composition-mode uuto-composition-mode uuto-composition-mode uuto-composition-mode uppared uppared later selection-mode lister selection-mode (global-set-key (kbd *; h h 1°) 'helm-info); helm help info ;; *** function description (global-set-key (kbd *; h h a*) 'helm-apropos); helm help apropos (global-set-key (kbd *; h h a*) 'helm-apropos); helm help apropos (global-set-key (kbd *; h h a*) 'helm-apropos); helm help apropos (global-set-key (kbd *; cf)> m) 'describe-mode) (global-set-key (kbd *; cf)> m) 'helm-describe-mode) (global-set-key (kbd *; f) m 1 'helm-describe-mode) (global-set-key (kbd *; f) m 1 'helm-describe-mode) (global-set-key (kbd *; f) m 1 'helm-describe-mode) (global-set-key (kbd *; h m s*) 'helm-describe-mode) (global-set-key (kbd *; h m s*) 'helm-describe-mode) (kbd *cf)> m' 'helm-describe-mode) (global-set-key (kbd *; h m s*) 'helm-desclinds) (global-set-key (kbd *; h m s*) 'helm-descl</pre>		
undo - composition mode ;; * * search help undo - composition mode ;; ** of to the code of the function at point undo - composition mode ;; already defined H company undo the phoch - mode ;; *** info company undo - defined H		(global-set-key (kbd "; m c") 'helm-mu-contacts)
unto-compression-mode ;; *** status http: unto-compression-mode ;; *** spots the code of the function at point sink-cursor-mode ;; arredy defined M singpary Takedo ;; *** spots the truth company mode ;; *** spots the truth idet = selection-mode ;; *** spots idet = selection-mode ;; *** spots <t< td=""><th></th><td></td></t<>		
unit or encryption mode ;; *** splot the code of the function at point init accursor mode ;; already defined H*. ispace of the code of the function at point ;; already defined H*. ispace of the code of the function at point ;; already defined H*. ispace of the code of the function at point ;; already defined H*. ispace of the code of the function at point ;; already defined H*. ispace of the code of the function at point ;; already defined H*. ispace of the code of the function at point ;; already defined H*. ispace of the code of the function at point ;; already defined H*. ispace of the code of the function at point ;; already defined H*. ispace of the code of the function at point ;; already defined H*. ispace of the code of the function at point ;; already defined H*. ispace of the code of the function at point ;; already defined H*. ispace of the code of the function at point ;; already defined H*. ispace of the code of the function at point ;; already defined H*. ispace of the code of the function at point ;; already defined H*. ispace of the code of the function at point ;; already defined H*. ispace of the code of the code of the function at point		;; ** search help
link-cursor-mode ;; arredy defined H company nuclekhelp-local-mode ;; arredy defined H company nuclekhelp-node ;; arredy defined H is posk with N+. ;; arredy defined H mojify mode ;; arredy defined H mojify mode ;; arredy defined H mojify mode ;; arredy defined H is posk with N+. ;; arredy defined H is posk with N ;; arredy defined H is posk with N+. ;; arredy defined H is posk with N ;; arredy defined H is posk with N ;; arredy defined H is posk with N ;; arredy defined H <		
company-mode company-quickhelp-local mode company-quickhelp-local mode company-quickhelp-local mode company-quickhelp-local mode company-quickhelp-local mode company-quickhelp-local mode company-quickhelp-local mode company-quickhelp-local mode ; #** info iter = set info ; #** function description iter = set info ; #** function description iter = set info ; #** function description mode [] ; #** node iter = set info ; #** describe node and helm describe modes iter = set info ; #** describe node and helm describe modes iter = set info ; #** describe node and helm describe modes iter = set info ; #*** describe node and helm describe modes iter = set info ; #*** describe node and helm describe modes iter = set info ; #*** describe node and helm describe modes iter = set info ; #*** describe node and helm describe modes iter = set info ; #*** describe node manger iter = set info ; #*** helm node manager iter = mode ; #*** kelpindig iter = set info ; #*** kelpindig iter = set info		;; *** goto the code of the function at point
company-quickhelp-mode ;; po back with H*, company-quickhelp-mode ;; *** info company-quickhelp-mode ;; *** info (global-set-key (kbd *; h h i') 'helm-info); helm help info ;; *** function description wild beginster in the set of the s		
company-quickNelp-load ;; *** info ig boost Num, ', ', ', ', ', ', ', ', ', ', ', ', ',		;; already defined M
compary quickbalp-mode ;; *** if o ide bescale ction-mode (global-set-key (kbd "; h h i') 'helm-info); helm help info ide ction-mode ;; *** function description moilf y-mode (global-set-key (kbd "; h h a') 'helm-apropos); helm help apropos int-indo-mode ;; *** mode int-indo-mode ;; *** mode int-indo-mode ;; *** mode int-indo-mode ;; *** mode int-indo-mode (global-set-key (kbd "; i of) of 'secribe-mode) int-indo-mode (global-set-key (kbd "; i of) of 'secribe-mode) int-indo-mode (global-set-key (kbd "; i of) of 'secribe-mode) int-indo-mode (global-set-key (kbd "; i of) of 'secribe-mode) int-indo-mode (global-set-key (kbd "(i of) of 'secribe-mode) indo-mode (global-set-key (kbd "(i of) of 'secribe-mode) indo-mode (global-set-key (kbd "(i of) of 'secribe-mode) indo-mode (global-set-key (kbd "(i of) of 'secribe-mode) indo-vertical-mode (global-set-key (kbd "(i of) of) of isecribe-mode) indo-vertical-mode (global-set-key (kbd "(i of) of) of isecribe-mode) indo-vertical-mode (global-set-key (kbd "; h m d') 'helm-describe-mode) indo-vertical-mode (g		;; go back with M-,
<pre>rmc.ustom-mode relation-mode relations mode relations relatio</pre>		
ide test selection-mode (global-set-key (kbd "; h h i") 'helm-info); helm help info ildoc-mode ;; *** function description mojify-mode_line-mode (global-set-key (kbd "; h h i") 'helm-apropos); helm help apropos mojify-mode_line-mode (global-set-key (kbd "; h h a") 'helm-apropos); helm help apropos 'lp-nade-shadevmode ;; *** node 'lp-nade-shadevmode ;; *** describe mode and heln describe modes (lobal-compary-mode (global-set-key (kbd "; <fi>m") 'describe-mode) (lobal-compary-mode (global-set-key (kbd "; <fi>m") 'helm-describe-mode) (lobal-compary-mode (global-set-key (kbd "; <fi>m") 'helm-describe-mode) (lobal-compary-mode (global-set-key (kbd "; fi>m") 'helm-describe-mode) (lobal-readify-mode line-mode (global-set-key (kbd "; fi>m") 'helm-describe-mode) (lobal-readify-mode line-mode (global-set-key (kbd "; fi m s") 'helm-describe-mode) (lobal-rot-lock-mode (global-set-key (kbd "; fi m s") 'helm-describe-mode) (lobal-not-lock-mode (global-set-key (kbd "; h m s") 'helm-describe-mode) (lobal-not-lock-mode (global-set-key (kbd "; h m s") 'helm-describe-mode) (lobal-set-key (kbd "; h m s") 'helm-describe-mode) (global-set-key (kbd "; h m s") 'helm-deschide) (lobal-set-key (kbd "; h m s") 'helm-deschide) <t< td=""><th></th><td>;; *** info</td></t<></fi></fi></fi>		;; *** info
ldoc-mode (global-set-key (kdo *, in n f) inelminity), inelminity mojify-mode ;; *** function description mojify-mode (global-set-key (kbd *; h h a*) 'helm-apropos); helm help apropos lile-mane-shadow-mode (global-set-key (kbd *; h h a*) 'helm-apropos); helm help apropos lip-in-mode (global-set-key (kbd *; cfl> m*) 'describe-mode) (lobal-mode/segive-indet-mode (global-set-key (kbd *; cfl> m*) 'describe-mode) (lobal-mode/fip-mode (global-set-key (kbd *; cfl> m*) 'helm-describe-modes) (lobal-mode/fip-mode (global-set-key (kbd *; cfl> m*) 'helm-describe-mode) (lobal-mode/fip-mode (global-set-key (kbd *; h m s*) 'helm-describe-mode) <th></th> <td></td>		
electric-indent-mode ;; *** function description empifyr-mode-line-mode (global-set-key (kbd "; h h a") 'helm-apropos); helm help apropos iber amper shadow-mode ;; *** mode 'ly-spell-mode ;; **** describe mode and help describe-modes 'lobal-ager seive-indent-mode (global-set-key (kbd "; <f1> m") 'describe-mode) 'lobal-ager seive-indent-mode (global-set-key (kbd "; <f1> m") 'describe-mode) 'lobal-company-mode (global-set-key (kbd "; <f1> m") 'describe-mode) 'lobal-company-mode (global-set-key (kbd "; <f1> m") 'describe-mode) 'lobal-company-mode (global-set-key (kbd "; <f1> m") 'describe-mode) 'lobal-enojify-mode (global-set-key (kbd "; <f1> m") 'helm-describe-modes) 'lobal-mode' (global-set-key (kbd "; >m") 'helm-describe-modes) 'lobal-mode' (global-set-key (kbd "; >m") 'helm-describe-modes) 'lobal-rot-inde (global-set-key (kbd "; >m") 'helm-describe-modes) 'lobal-mode' (global-set-key (kbd "; >m m") 'helm-describe-mode) 'lobal-rot-inde (global-set-key (kbd "; >m m") 'helm-describe-mode) 'lobal-word-worde (global-set-key (kbd "; >m m") 'helm-describe-mode) 'lobal-word-worde (global-set-key (kbd "; >m m") 'helm-describe-mode) 'lobal-set-key (kbd "; >m m") 'helm-desc</f1></f1></f1></f1></f1></f1>		(global-set-key (kbd "; h h i") 'helm-info) ; helm help info
<pre>mojify-mode "; *** function description "; *** function "; *** function description "; *** function "; **** functin "; **** function ";</pre>		
mojify:mode-line-mode (global-set-key (kbd "; h h a") 'helm-apropos); helm help apropos ly:spell-mode j; *** mode lyspell-mode j; *** mode lyspell-mode (global-set-key (kbd "; <f)>m') 'describe-mode) (lobal-compary-mode (global-set-key (kbd "; <f)>m') 'describe-mode) (lobal-compary-mode (global-set-key (kbd "; <f)>m') 'describe-mode) (lobal-enojify-mode-line-mode (global-set-key (kbd "; <f)>m') 'describe-mode) (lobal-enojify-mode-line-mode (global-set-key (kbd "; <f)>m') 'describe-mode) (lobal-enojify-mode-line-mode (global-set-key (kbd "sf)>m') 'helm-describe-modes) (do-verytike-mode (global-set-key (kbd "sf)>m) 'helm-describe-modes) (do-verytike-mode (global-set-key (kbd "sf)>m m s') 'helm-switch-major-mode) vortil:ne-mode (global-set-key (kbd "s h m s') 'helm-describe-mides) vortil:ne-mode (global-set-key (kbd "s h m s') 'helm-describe-mode) vortil:ne-mode (global-set-key (kbd "s h m s') 'helm-describe-mode) vorti:ne-mode (global-set-key (k</f)></f)></f)></f)></f)>		;; *** function description
Tile-name-shadow-mode \global-set-key (kdd ', i'' i'' a') 'helm-deproposity', inclin help aproposity', inclin help aproposity		
Ils-ido-mode ;; *** mode Ont-lock-mode ;; *** mode Ont-lock-mode ;; **** describe mode and heln describe modes Jobal-aggressive-indet-mode (global-set-key (kbd "; <f)> m") 'describe-mode) Jobal-emojify-mode (global-set-key (kbd "; <f)> m") 'describe-mode) Jobal-emojify-mode (global-set-key (kbd "; <f)> m") 'describe-mode) Jobal-emojify-mode (global-set-key (kbd "; <f)> m") 'describe-modes) Jobal-emojify-mode (global-set-key (kbd "; <f)> m") 'helm-describe-modes) Jobal-emojify-mode (global-set-key (kbd "; + m") 'helm-describe-modes) Jobal-emojify-mode (global-set-key (kbd "; hm s") 'helm-describe-modes) Jobal-indo-tree-mode (global-set-key (kbd "; hm s") 'helm-switch-major-mode) Jobal-indo-tree-mode (global-set-key (kbd "; hm s") 'helm-switch-major-mode) Jobal-indo-mode (global-set-key (kbd "; hm e") 'helm-enable-minor-mode) Jobal-set-key (kbd "; hm e") 'helm-descbinds) :; *** keybinding Taihbow-delimiters-mode (global-set-key (kbd "; hm e") 'helm-descbinds) Jobal-set-key (kbd "; hm e") 'helm-descbinds) :; *** package Taihbow-delimiters-mode ;; *** package Signartiperes-mode ;: wheu sead and the function below is loaded (otherwise it is loaded only i</f)></f)></f)></f)></f)>		(global-set-key (kbd "; h h a") 'helm-apropos) ; helm help apropos
TypeII-mode i; *** House TypeII-mode i; *** House Stabla-aggressive-indent-mode i; **** describe mode and helm describe modes Stabla-aggressive-indent-mode (global-set-key (kbd "; <fl>m") 'describe-mode) Stabla-aggressive-indent-mode (global-set-key (kbd "; <fl>m") 'describe-mode) Stabla-emojify-mode (global-set-key (kbd "; <fl>m") 'helm-describe-modes) Stabla-emojify-mode (global-set-key (kbd "fl>m") 'helm-describe-modes) Stabla-font-lock-mode (global-set-key (kbd "fl>m") 'helm-describe-modes) Stabla-emode (global-set-key (kbd "fl>m s") 'helm-describe-modes) Stabla-emode (global-set-key (kbd "fl>m s") 'helm-describe-mode) Stabla-emode (global-set-key (kbd "fl>m s") 'helm-describe-mode)</fl></fl></fl>		
iont-lock-mode ;; **** describe mode and helm describe modes iobal-scompany-mode (global-set-key (kbd "; <fi>m") 'describe-mode) ilobal-eldoc-mode (global-set-key (kbd "; <fi>m") 'describe-mode) ilobal-emojify-mode (global-set-key (kbd "; <fi>m") 'helm-describe-mode) ilobal-mojify-mode (global-set-key (kbd "<fi>m") 'helm-describe-modes) ilobal-emojify-mode (global-set-key (kbd "<fi>m") 'helm-describe-modes) ilobal-undo-tree-mode (global-set-key (kbd "<fi>m") 'helm-describe-modes) ido-everywhere (global-set-key (kbd ", i m s") 'helm-describe-modes) ido-everywhere (global-set-key (kbd ", i m s") 'helm-switch-major-mode) uotse-wheel-mode (global-set-key (kbd ", i m s") 'helm-switch-major-mode) uotse-wheel-mode (global-set-key (kbd ", i m s") 'helm-describe-mide) uotse-wheel-mode (global-set-key (kbd ", i m s") 'helm-switch-major-mode) uotse-wheel-mode (global-set-key (kbd ", i m s") 'helm-describe-minor-mode) uotse-wheel-mode (global-set-key (kbd ", i m s") 'helm-describe-minor-mode) uotse-wheel-mode (global-set-key (kbd ", i m d") 'helm-describe-minor-mode) uotse-wheel-mode (global-set-key (kbd ", i h b") 'helm-describe-minor-mode) uotse-wheel-mode (global-set-key (kbd ", i h b") 'helm-describe-mino</fi></fi></fi></fi></fi></fi>		;; *** mode
Iobal-aggressive-indent-mode ;; **** describe Mode and net M describe Modes Iobal-aggressive-indent-mode (global-set-key (kbd "; <fi>m") 'describe-mode) Iobal-company-mode (global-set-key (kbd "; <fi>m") 'helm-describe-mode) Iobal-and if y-mode line-mode (global-set-key (kbd "; <fi>m") 'helm-describe-mode) Iobal-fort-lock-mode (global-set-key (kbd ", fi) m") 'helm-describe-modes) Iobal-tortere-mode (define-key dired-mode-map (kbd "; fi] m") 'helm-describe-modes) Ido-ubiguitous-mode (global-set-key (kbd ", fi] m") 'helm-describe-mode) (ido-ubiguitous-mode (global-set-key (kbd ", fi] m") 'helm-describe-mode) (ido-ubiguitous-mode (global-set-key (kbd ", fi] m s") 'helm-describe-mode) (ido-ubiguitous-mode (global-set-key (kbd ", fi m s") 'helm-switch-major-mode) (ubiguitous-mode (global-set-key (kbd ", fi m s") 'helm-enable-minor-mode) (use-wheel-mode (global-set-key (kbd ", fi m s") 'helm-enable-minor-mode) uithine-mode (global-set-key (kbd ", fi m s") 'helm-enable-minor-mode) uithine-mode (global-set-key (kbd ", fi b ") 'helm-descbinds) ainbow-delinitfiers-mode ;; *** keybinding e-eentf-mode (global-set-key (kbd ", fi b b") 'helm-descbinds) smartparens-mode ;; *** package</fi></fi></fi>		
iobal-company-mode (global-set-key (kbd "; <fi>m") 'describe-mode) iobal-endify-mode (global-set-key (kbd ";, fi>m") 'describe-mode) iobal-endify-mode (global-set-key (kbd ";, fi>m") 'describe-mode) idbal-undo-tree-mode (global-set-key (kbd "<fi>m") 'helm-describe-modes) ido-ubiquitous-mode (global-set-key (kbd "<fi>m ns") 'helm-describe-mode) ido-ubiquitous-mode (global-set-key (kbd "<fi>m ns") 'helm-describe-mode) ido-ubiquitous-mode (global-set-key (kbd "; h m s") 'helm-describe-mode) uptishine-mode (global-set-key (kbd "; h m d") 'helm-descbinds) inhow-delimiters-mode (global-set-key (kbd "; h m d") 'helm-descbinds) inhow-delimiters-mode (global-set-key (kbd "; h b") 'helm-descbinds) is inhow-identifiers-mode (global-set-key (kbd "; h b") 'helm-descbinds)</fi></fi></fi></fi></fi></fi></fi></fi></fi></fi>		;; **** describe mode and helm describe modes
[bbal-eldoc-mode (gtobal-set-key (kbd "; f1> m") 'describe-mode) [bbal-emojify-mode (gtobal-set-key (kbd "; f1> m") 'describe-mode) [bbal-fort-lock-mode (gtobal-set-key (kbd "; f1> m") 'helm-describe-modes) [bbal-ort-lock-mode (gtobal-set-key (kbd "; f1> m") 'helm-describe-mode) [bbal-uot-tree-mode (gtobal-set-key (kbd "; f1> m") 'helm-describe-modes) (do-vertical-mode (gtobal-set-key (kbd "; h m s") 'helm-describe-mode) (gtobal-set-key (kbd "; h m s") 'helm-describe-mode) (gtobal-set-key (kbd "; h m s") 'helm-describe-mode) outline-minor-mode (gtobal-set-key (kbd "; h m s") 'helm-describe-mode) outline-minor-mode (gtobal-set-key (kbd "; h m s") 'helm-describe-mode) outline-minor-mode (gtobal-set-key (kbd "; h m s") 'helm-otable-minor-mode) outline-minor-mode (gtobal-set-key (kbd "; h m d") 'helm-descbinds) ainbow-identifiers-mode ;; **** keybinding shell-dirtrack-mode (gtobal-set-key (kbd "; h b") 'helm-descbinds) simartparens-mode ;; **** package telega-atrons-mode ;; necessary to load it before the function below is loaded (otherwise it is loaded only telega-atrons-mode ;; when used and the function below is replaced by the default function)		
Iobal-emojify-mode (define-Rey dired-mode-map (kbd "; fr1") ") describe-mode) Iobal-emojify-mode (global-set-key (kbd "cf1>m") 'helm-describe-modes) Iobal-undo-tree-mode (define-key dired-mode-map (kbd "cf1>m") 'helm-describe-modes) Ido-everywhere (define-key dired-mode-map (kbd "cf1>m") 'helm-describe-modes) Ido-everywhere (global-set-key (kbd "cf1>m") 'helm-describe-modes) Ido-everywhere (global-set-key (kbd "cf1>m") 'helm-switch-major-mode) Ido-everywhere (global-set-key (kbd "; h m s") 'helm-switch-major-mode) outse-wheel-mode (global-set-key (kbd "; h m e") 'helm-enable-minor-mode) outsine-mode (global-set-key (kbd "; h m d") 'helm-enable-minor-mode) outsine-mode (global-set-key (kbd "; h m d") 'helm-descbinds) reacher mode (global-set-key (kbd "; h b") 'helm-descbinds) reacher mode (global-set-key (kbd "; h b") 'helm-descbinds) shell-dirtrack-mode (global-set-key (kbd "; h b") 'helm-descbinds) shell-dirtrack-mode (global-set-key (bd "; h b") 'helm-descbinds) shell-dirtrack-mode (global-set-key (kbd "; h b") 'helm-descbinds) shell-dirtrack-mode (global-set-key (kbd "; h b") 'helm-descbinds) shell-dirtrack-mode (global-set-key (kbd "; h b") 'helm-descbinds) st		(global-set-key (kbd "; <f1> m") 'describe-mode)</f1>
global-semojity_mode-line-mode (global-set-key (kbd " <fi>m") 'helm-describe-modes) global-font-lock-mode (define-key dired-mode-map (kbd "<fi>m") 'helm-describe-modes) ido-utipuitous-mode ;; **** helm mode manager ido-verywhere (global-set-key (kbd "; h m s") 'helm-switch-major-mode) wouse-wheel-mode (global-set-key (kbd "; h m s") 'helm-switch-major-mode) wouse-wheel-mode (global-set-key (kbd "; h m e") 'helm-switch-major-mode) wouse-wheel-mode (global-set-key (kbd "; h m d") 'helm-descbinds) wouse-wheel-mode (global-set-key (kbd "; h b") 'helm-descbinds) wouse-show-paren-mode (global-set-key (kbd "; h b") 'helm-descbinds)</fi></fi>		(define-key dired-mode-map (kbd "; <f1> m") 'describe-mode)</f1>
Clobal-font-lock-mode (global-set-key (kbd) helm-describe-modes) Clobal-undo-tree-mode (define-key dired-mode-map (kbd (fi>m) 'helm-describe-modes) Clobal-undo-tree-mode ;; **** helm mode manager (global-set-key (kbd "; h m s") 'helm-describe-modes) Clobal-set-key (kbd "; h m s") 'helm-describe-modes) (global-set-key (kbd "; h m s") 'helm-describe-modes) Nousse-wheel-mode (global-set-key (kbd "; h m s") 'helm-enable-minor-mode) (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) Systems-mode (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) systems-mode (global-set-key (kbd "; h m d") 'helm-descbinds) ;; **** keybinding recentf-mode ;; *** keybinding ;; **** package ;; *** package smartparens-global-mode ;; necessary to load it before the function below is loaded (otherwise it is loaded only ;; when used and the function below is replaced by the default function) Wellen Describe Modes Lil [] C-h m:Help TAB:Act RET/fl/f2/f-n:NthAct C-!:Tog.suspend C- 67% 3422 Emacs-Lisp s U: /home/marie/.emacs		
Iobal-undo-tree-mode (define-key dired-mode-map (kbd (kf) m) 'helm-describe-modes) ido-verywhere ;; **** helm mode manager ido-vertical-mode (global-set-key (kbd "; h m s") 'helm-switch-major-mode) nouse-wheel-mode (global-set-key (kbd "; h m e") 'helm-enable-minor-mode) outline-minor-mode (global-set-key (kbd "; h m d") 'helm-enable-minor-mode) outshine-mode (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) oyvenv-mode (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) rainbow-idelimiters-mode (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) rainbow-idelimiters-mode (global-set-key (kbd "; h b") 'helm-descbinds) rainbow-idelimiters-mode ;; *** keybinding shell-dirtrack-mode (global-set-key (kbd "; h b") 'helm-descbinds) smartparens-global-mode ;; *** package smartparens-mode ;; necessary to load it before the function below is loaded (otherwise it is loaded only stelega-rot-suto-fill-mode ;; when used and the function below is replaced by the default function) wielm Describe Modes Lil II Candidate(s)] C-mmether MaskAtt RET/f1/f2/f-m:NthAct C-1:Tog.suspend C-1 - Lemacs (Lisp s U): /home/marie/.emacs		
do-everywhere ;; **** helm mode manager ido-ubiquitous-mode (global-set-key (kbd "; h m s") 'helm-switch-major-mode) nouse-wheel-mode (global-set-key (kbd "; h m s") 'helm-enable-minor-mode) vutline-minor-mode (global-set-key (kbd "; h m d") 'helm-enable-minor-mode) vutline-minor-mode (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) vytenv-mode (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) ainbow-idelimiters-mode ;; *** keybinding ecentf-mode (global-set-key (kbd "; h b") 'helm-disable-minor-mode) shell-dirtrack-mode ;; *** keybinding shell-dirtrack-mode (global-set-key (kbd "; h b") 'helm-descbinds) smartparens-global-mode ;; *** package smartparens-mode ;; when used and the function below is loaded (otherwise it is loaded only selega-artors-mode ;; when used and the function below is replaced by the default function) wielem bescribe Modes Lil II Candidate(s)] C-h m:Help TAB:Act RET/f1/2/f-n:NtAct C=!:Tog.suspend C 67% 3422 Emacs-Lisp s		(define-key dired-mode-map (kbd " <f1> m") 'helm-describe-modes)</f1>
<pre>ido-ubiquitous-mode iido-vertical-mode mouse-wheel-mode poutshine-minor-mode outshine-minor-mode outshine-mode yoyeen-mode ainbow-delimiters-mode ainbow-identifiers-mode recentf-mode shell-dirtrack-mod</pre>		
ido-vertical-mode (global-set-key (kbd "; h m s") 'helm-switch-major-mode) nouse-wheel-mode (global-set-key (kbd "; h m s") 'helm-switch-major-mode) outshine-mode (global-set-key (kbd "; h m s") 'helm-enable-minor-mode) outshine-mode (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) outshine-mode (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) rainbow-delimiters-mode (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) rainbow-delimiters-mode (global-set-key (kbd "; h b") 'helm-disable-minor-mode) rainbow-delimiters-mode (global-set-key (kbd "; h b") 'helm-disable-minor-mode) rainbow-delimiters-mode (global-set-key (kbd "; h b") 'helm-disable-minor-mode) shell-dirtrack-mode (global-set-key (kbd "; h b") 'helm-disable-minor-mode) smartparens-global-mode (global-set-key (kbd "; h b") 'helm-disable-minor-mode) smartparens-global-mode (global-set-key (kbd "; h b") 'helm-descbinds) smartparens-global-mode ;; *** package smartparens-mode ;; mecessary to load it before the function below is loaded (otherwise it is loaded only stelega-rot-auto-fill-mode [fill Candidate(s)] C-h m:Help TAB:Act RET/f1/2/f-n:NthAct C-1:Tog.suspend C-1 [film desc-tisp s U: /home/marie/.emacs		;; **** helm mode manager
wheel-mode (global-set-key (kbd "; h m s") 'helm-switch-major-mode) wutline-minor-mode (global-set-key (kbd "; h m s") 'helm-enable-minor-mode) wutsine-mode (global-set-key (kbd "; h m s") 'helm-enable-minor-mode) wytenv-mode (global-set-key (kbd "; h m s") 'helm-enable-minor-mode) ainbow-ideInitifiers-mode (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) ainbow-ideInitifiers-mode (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) eccentf-mode (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) shell-dirtrack-mode (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) show-paren-mode (global-set-key (kbd "; h b") 'helm-disable-minor-mode) smartparens-global-mode (global-set-key (kbd "; h b") 'helm-descbinds) smartparens-global-mode (global-set-key (kbd "; h b") 'helm-descbinds) elega-atrons-mode (global-set-key (bd "; h b") 'helm-descbinds) elega-atrons-mode (global-set-key (bd "; h b") 'helm-descbinds) elega-atrons-mode (global-set-key (bd "; h b") 'helm-disable-minor-mode) elega-atrons-mode (global-set-key (bd "; h b") 'helm-descbinds) elega-atrons-mode (global-set-key (bd i i before the function below is loaded (otherwise it is loaded only elega-atrons-mode (global-set-		
but line-minor-mode (global-set-key (kbd "; h m e") 'helm-enable-minor-mode) but shine-mode (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) rainbow-delimiters-mode (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) rainbow-identifiers-mode ;; *** keybinding recentf-mode (global-set-key (kbd "; h b") 'helm-disable-minor-mode) shell-dirtrack-mode (global-set-key (kbd "; h b") 'helm-disable-minor-mode) smart parens-global-mode (global-set-key (kbd "; h b") 'helm-descbinds) smart parens-global-mode ;; *** package selega-atrons-mode ;; necessary to load it before the function below is loaded (otherwise it is loaded only is replaced by the default function) selega-atrons-mode ;; when used and the function below is replaced by the default function) selega-tibe Modes Ll1 [11 Candidate(s)] C-h m:Help TAB:Act RET/f1/2/f=n:NthAct C=1:tog.suspend C		(global-set-key (kbd "; h m s") 'helm-switch-major-mode)
butshine-mode (global-set-key (kbd "; h m d") 'helm-enable-minor-mode) byverv-mode (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) rainbow-delimiters-mode (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) rainbow-delimiters-mode ;; *** keybinding recentf-mode (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) shell-dirtrack-mode (global-set-key (kbd "; h b") 'helm-disable-minor-mode) show-pare-mode ;; *** keybinding smartparens-global-mode (global-set-key (kbd "; h b") 'helm-descbinds) smartparens-global-mode ;; *** package telega-active-locations-mode ;; mecessary to load it before the function below is loaded (otherwise it is loaded only telega-root-auto-fill-mode ;; when used and the function below is replaced by the default function) whelm Describe Modes LU IL Candidate(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-1:Tog.suspend C-1 amacs Lisp s U: /home/marie/.emacs		
byvenv-mode (global-set-key (kbd "; h m d") 'helm-disable-minor-mode) rainbw-delimiters-mode ;; *** keybinding recentf-mode ;; *** keybinding shell-dirtrack-mode (global-set-key (kbd "; h b") 'helm-disable-minor-mode) shell-dirtrack-mode (global-set-key (kbd "; h b") 'helm-disable-minor-mode) shell-dirtrack-mode (global-set-key (kbd "; h b") 'helm-descbinds) smartparens-global-mode ;; *** package smartparens-mode ;; necessary to load it before the function below is loaded (otherwise it is loaded only selega-artors-mode ;; when used and the function below is replaced by the default function) Helm Describe Modes Ll1 Il Candidate(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-1:Tog.suspend C-1		(global-set-key (kbd "; h m e") 'helm-enable-minor-mode)
ainbow-delimiters-mode (global-set-key (kbd '; n m d') 'helm-disable-minor-mode) rainbow-identifiers-mode ;; *** keybinding shell-dirtrack-mode (global-set-key (kbd '; h b') 'helm-disable-minor-mode) shew-paren-mode (global-set-key (kbd '; h b') 'helm-disable-minor-mode) shew-paren-global-mode (global-set-key (kbd '; h b') 'helm-disable-minor-mode) smartparens-global-mode (global-set-key (kbd '; h b') 'helm-discbinds) smartparens-mode ;; *** package telega-ative-locations-mode ;; necessary to load it before the function below is loaded (otherwise it is loaded only telega-rot-auto-fill-mode ;; when used and the function below is replaced by the default function) Helm Descripte Modes Lil [1 Candidate(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-1:Tog.suspend C-i		
<pre>rainbow-identifiers-mode recentf-mode shell-dirtrack-mode shell-dirtrack-mode show-paren-mode smartparens-global-mode smartparens-global-mode smartparens-global-mode selega-active-locations-mode telega-patrons-mode telega-root-auto-fill-mode Hellm Describe Modes LLI C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend Cmacs 67% 3422 Emacs-Lisp s.U: /home/marie/.emacs</pre>		(global-set-key (kbd "; h m d") 'helm-disable-minor-mode)
<pre>recentf-mode ;; *** keybinding ;; *** keybinding ;; *** keybinding ;; *** keybinding (global-set-key (kbd "; h b") 'helm-descbinds) smartparens-mode smartparens-mode :elega-ative-locations-mode :elega-patrons-mode :elega-patrons-mode :elega-root-auto-fill-mode Helm Describe Modes Ll C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C 67% 3422 Emacs-Lisp s U: /home/marie/.emacs</pre>		
shell-dirtrack-mode (global-set-key (kbd "; h b") 'helm-descbinds) smartparens-mode (global-set-key (kbd "; h b") 'helm-descbinds) smartparens-mode ;; *** package stelega-active-locations-mode ;; necessary to load it before the function below is loaded (otherwise it is loaded only stelega-root-auto-fill-mode ;; when used and the function below is replaced by the default function) Helm Describe Modes LI [I Candidate(s)] C-h m:Help TAB:Act RET/f1/f2/f=n:NthAct C-1:Tog.suspend C-1 - Image III 67% 3422 Emacs-Lisp s U: /home/marie/.emacs	recentf-mode	;; *** keybinding
show-paren-mode (global-set-key (kbd "; h b") "helm-descbinds) smartparens-global-mode ;; *** package smartparens-mode ;; *** package selega-active-locations-mode ;; mecessary to load it before the function below is loaded (otherwise it is loaded only selega-root-auto-fill-mode ;; when used and the function below is replaced by the default function) wHelm Describe Modes LI [I Candidate(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C-l- 67% 3422 Emacs-Lisp s U: /home/marie/.emacs	shell-dirtrack-mode	
<pre>smartparens-global-mode smartparens-mode selega-patrons-mode selega-patrons-mode selega-patrons-mode selega-patrons-fill-mode Helm Describe Modes Ll [1 Candidate(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C Immed G7% 3422 Emacs-Lisp s U: /home/marie/.emacs</pre>	show-paren-mode	(global-set-key (kbd "; h b") 'helm-descbinds)
smartparens-mode telega-active-locations-mode telega-patrons-mode telega-root-auto-fill-mode #Helm Describe Modes* L1 [1 Candidate(s)] C-h m:Help TAB:Act RET/f1/f2/f=n:NthAct C-!:Tog.suspend Cemacs 67% 3422 Emacs-Lisp s U: /home/marie/.emacs	smartparens-global-mode	
telega-active-locations-mode telega-patrons-mode telega-root-auto-fill-mode Helm Describe Modess L1 [1 Candidate(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C- macs 67% 3422 Emacs-Lisp s U: /home/marie/.emacs		;; *** package
<pre>telega-patrons-mode ;; necessary to load it before the function below is loaded (otherwise it is loaded only ;elega-root-auto-fill-mode Helm Descripte Modes+ L1 [1 Candidate(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C-! - Immacs 67% 3422 Emacs-Lisp s U: /home/marie/.emacs</pre>	telega-active-locations-mode	
telega-root-auto-fill-mode ;; when used and the function below is replaced by the default function) *Helm Describe Modes* L1 [1 Candidate(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C- emacs contact conta	telega-patrons-mode	
Helm Describe Modes L1 [1 Candidate(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend Cemacs 67% 3422 Emacs-Lisp s U: /home/marie/.emacs	telega-root-auto-fill-mode	;; when used and the function below is replaced by the default function)
battern:		- emacs 67% 3422 Emacs-Lisp s U: /home/marie/.emacs
	pattern:	

Search help files

C-j: Toggle describe command (keeping session)	
Commands	(global-set-key (kbd "; h l") 'helm-locate)
helm-ff-RET	
helm-ff-TAB	:: ** search email
helm-ag-mode	
helm-epa-mode	(global-set-key (kbd "; m s") 'helm-mu) ; mu search
helm-mark-all	
helm-locate <u>C-x c l</u>	(global-set-key (kbd "; m c") 'helm-mu-contacts)
<mark>helm</mark> -info-bfd	
<mark>helm</mark> -info-cpp	;; ** search help
<mark>helm</mark> -info-ksba	
helm-info-gzip	;; *** goto the code of the function at point
helm-info-wget	
helm-info-libc	;; already defined M
helm-info-gperf	;; go back with M-,
helm-info-gpgme	
<mark>helm</mark> -info-gpgrt <mark>helm</mark> -info-sieve	;; *** info
helm-info-sieve	
helm-world-time	(global-set-key (kbd "; h h i") 'helm-info) ; helm help info
helm-info-nettle	
helm-info-eshell	;; *** function description
helm-info-gccint	
helm-info-R-lang	(global-set-key (kbd "; h h a") 'helm-apropos) ; helm help apropos
helm-follow-mode	:: *** mode
helm-sexp-eval-1	
helm-info-dired-x	;; **** describe mode and helm describe modes
<mark>helm</mark> -info-gettext	,, AAAA describe mode and neth describe modes
<mark>helm</mark> -info-libidn2	(global-set-key (kbd "; <f1> m") 'describe-mode)</f1>
<mark>helm</mark> -ag-mode-jump	(define-key dired-mode-map (kbd "; <f1> m") 'describe-mode)</f1>
<mark>helm</mark> -ag-this-file	
helm-swap-windows	(global-set-key (kbd " <f1> m") 'helm-describe-modes)</f1>
helm-ff-icon-mode	(define-key dired-mode-map (kbd " <f1> m") 'helm-describe-modes)</f1>
<mark>helm</mark> -info-gsl-ref	
helm-narrow-window	;; **** helm mode manager
helm-info-readline	
helm-info-autoconf	(global-set-key (kbd "; h m s") 'helm-switch-major-mode)
<mark>helm</mark> -keyboard-quit <mark>helm</mark> -info-libtasn1	
helm-adaptive-mode	(global-set-key (kbd "; h m e") 'helm-enable-minor-mode)
helm-manage-advice	
helm-select-xfont C-x c F	(global-set-key (kbd "; h m d") 'helm-disable-minor-mode)
helm-grep-mode-jump	
figures grand grand here the second	;; *** keybinding
helm info-inetutils	
helm-cmd-t-run-grep	(global-set-key (kbd "; h b") 'helm-descbinds)
helm-info-nxml-mode	
helm-enlarge-window	;; *** package
helm-ff-run-grep-ag	, proceeding to load it before the function below is loaded (athenning it is loaded or the
<mark>helm</mark> -ff-run-pdfgrep	;; necessary to load it before the function below is loaded (otherwise it is loaded only
helm-buffers- <u>list C-x c C-x C-b</u>	;; when used and the function below is replaced by the default function)
<pre>*helm apropos* L1 [100 Candidate(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C-h c:Con</pre>	emacs 67% 3422 Emacs-Lisp s U: /home/marie/.emacs
pattern: helm	

Search key bindings

C-j: helm-deschinds-action:describe (keeping session)			
Major Mode Bindi			
C-x	Prefix Command		
ESC	Prefix Command		
DEL	backward-delete-char-untabify		
C-x C-a	Prefix Command		
C-M-q	indent-pp-sexp		
	eval-defun		
	debug-go-mode		
	debug-where		
	debug-set-initial-mode		
	debug-next-mode		
C X C 4 C 5			
`tracking-mode'	Minor Mode Bindings:		
	Prefix Command		
	tracking-next-buffer		
	tracking next-buffer		
`flyspell-mode'	Minor Mode Bindings:		
	Prefix Command		
	flyspell-goto-next-error		
	flyspell-auto-correct-word		
	flyspell-auto-correct-previous-word		
	flyspell-correct-word-before-point		
	Minor Mode Bindings:		
TAB	outshine-kbd-TAB		
ESC	Prefix Command		
<m-s-down></m-s-down>	outshine-kbd-M-S- <down></down>		
<m-s-left></m-s-left>	outshine-kbd-M-S- <left></left>		
<m-s-right></m-s-right>	outshine-kbd-M-S- <right></right>		
<m-s-up></m-s-up>	outshine-kbd-M-S- <up></up>		
<m-down></m-down>	outline-next-visible-heading		
<m-left></m-left>	outshine-kbd-M- <left></left>		
	outshine-kbd-M- <right></right>		
	utline-previous-visible-heading		
	outshine-kbd- <backtab></backtab>		
	Prefix Command		
	utshine-cycle-buffer		
	utshine Kod-MRET		
	sert-command> outshine-self-insert-command		
`outline-minor-r	ode' Minor Mode Bindings:		
	Prefix Command		
	Prefix Command		
	Prefix Command		
	org-cycle		
	of source of the second s		
	outline new outside heading		
	outshine-yevele-buffer		
helm-desc <u>b</u> inds			
pattern:			

Search in emails

C-J: neim-	mu-persistent-action (keeping session)	
Search em	ail with mu	
22-02-21	Mountain Conditions Re mhburle@gmail.com	[MCR] Asulkan Valley - Sapphire Col to Dome
22-02-06	Mountain Conditions Re mhburle@gmail.com	[MCR] Avalanche Conditions Summary - Central Selkirk Ranges - Observation from this
22-02-04	Mountain Conditions Re mhburle@gmail.com	[MCR] Slocan Ranges - Snowpack and ski conditions
22-01-28	Mountain Conditions Re mhburle@gmail.com	[MCR] Avalanche Crest - Rogers Pass - Avalanche Crest #3 - Something to keep in mi
21-05-06	Mountain Conditions Re mhburle@gmail.com	[MCR Regional Summary] ACMG Mountain Conditions Report for the Rockies and Columbia
20-11-12	Mountain Conditions Re mhburle@gmail.com	[MCR] Balu Pass, Glacier NP - Crusts galore at Rogers Pass
20-11-12	Mountain Conditions Re mhburle@gmail.com	[MCR] Balu Pass, Glacier NP - Crusts galore at Rogers Pass

Select colour

C-j: Kill entry in RGB format. (keeping session)			
Colors			(global-set-key (kbd "; h m s") 'helm-switch-major-mode)
black	black	#000000	
dim gray	dim gray,dim grey,DimGray,DimGrey	#696969	(global-set-key (kbd "; h m e") 'helm-enable-minor-mode)
dark gray	dark gray,dark grey,DarkGray,DarkGrey	#a9a9a9	
gray	gray,grey	#bebebe	(global-set-key (kbd "; h m d") 'helm-disable-minor-mode)
light gray	light gray,light grey,LightGray,LightGrey	#d3d3d3	
gainsboro	gainsboro	#dcdcdc	;; *** keybinding
white smoke	white smoke,WhiteSmoke	#f5f5f5	
white	white	#ffffff	(global-set-key (kbd "; h b") 'helm-descbinds)
red	red	#ff0000	
orange red dark aranga	orange red,OrangeRed dark orange,DarkOrange	#ff4500 #ff8c00	;; *** package
dark orange orango	orange	#ffa500	
orange gold	gold	#ffd700	;; necessary to load it before the function below is loaded (otherwise it is loaded only
vellow	yellow	#ffff00	;; when used and the function below is replaced by the default function)
chartreuse	chartreuse	#7fff00	
lawn green	lawn green,LawnGreen	#7cfc00	(defun helm-system-packages-pacman-uninstall (_)
green	green	#00ff00	"Uninstall marked candidates."
spring green	spring green,SpringGreen	#00ff7f	(helm-system-packages-run-as-root-over-installed
medium spring green	medium spring green, MediumSpringGreen	#00fa9a	"pacman" "-Rsn"
cyan	cyan	#00ffff	(when helm-current-prefix-arg "recursive")
deep sky blue	deep sky blue,DeepSkyBlue	#00bfff	(unless helm-system-packages-pacman-confirm-p "noconfirm")))
blue		#0000ff	
medium blue		#0000cd	🕼 global-set-key (kbd "; h h p") 'helm-system-packages)
dark violet	dark violet,DarkViolet	#9400d3	
dark magenta	dark magenta,DarkMagenta	#8b008b	:: ** color
magenta	magenta	#ff00ff	
dark red	dark red,DarkRed	#8b0000	(global-set-key (kbd "; h c") 'helm-colors)
brown	brown	#a52a2a	
firebrick	firebrick	#b22222	;; ** search web engines with surfraw
indian red	indian red, IndianRed	#cd5c5c	
light coral	light coral,LightCoral salmon	#f08080 #fa8072	(global-set-key (kbd "; h w") 'helm-surfraw) ; helm web
salmon light salmon	light salmon,LightSalmon	#ffa07a	
tomato	tomato	#ff6347	;; * helm M-x
coral	coral	#ff7f50	
dark salmon	dark salmon,DarkSalmon	#e9967a	(global-set-key (kbd "; M-x") 'helm-M-x)
rosy brown	rosy brown,RosyBrown	#bc8f8f	v s unde /sede
sienna	sienna	#a0522d	;; * undo/redo
saddle brown	saddle brown,SaddleBrown	#8b4513	(global-set-key (kbd "C-1") 'undo)
chocolate	chocolate	#d2691e	
peru	peru	#cd853f	;; ** undo tree
sandy brown	sandy brown,SandyBrown	#f4a460	,, and the tree
burlywood	burlywood	#deb887	(global-set-key (kbd "s-t") 'undo-tree-visualize)
tan	tan	#d2b48c	
navajo white	navajo white,NavajoWhite	#ffdead	;; go to selected undo state
wheat	wheat	#f5deb3	(define-key undo-tree-visualizer-mode-map (kbd " <return>") 'undo-tree-visualizer-quit)</return>
dark goldenrod	dark goldenrod,DarkGoldenrod	#b8860b	
goldenrod	goldenrod	#daa520	;; cancel (return to state before calling undo-tree-visualize)
Light goldenrod	light goldenrod,LightGoldenrod	#eedd82	(define-key unde-tree-vigualizer-mode-man (khd "a") unde-tree-vigualizer-abert)
helm colors L1 [100	Candidate(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspe	na c-n c:Conf	emacs 67% 3456 Emacs-Lisp s U: /home/marie/.emacs

pattern:

Select emoji

C-j: Insert Character (keeping session)	'(window-divider ((t (:foreground "gray25"))))
unicode-characters FACE PALM 🔒	'(yas-field-highlight-face ((t (:underline t))))
	: * unset env variable MANPAGER in Emacs only
FACE WITH MONOCLE @	;; MANPAGER is set in zsh to sh -c 'col -bx bat -l man -p'
FACE WITHOUT MOUTH	;; which provides bat colour highlighting in the man pages
FACE THROWING A KISS 😚	;; but this messes up the help files in inferior-python-mode
FACE WITH COLD SWEAT	
FACE WITH COWBOY HAT 😇	(setenv "MANPAGER" "")
FACE WITH OK GESTURE 😔	
FACE WITH OPEN MOUTH 😢	;; * define new prefix keys
FACE WITH THERMOMETER 🥹	
FACE SCREAMING IN FEAR W	
FACE WITH HEAD-BANDAGE 🛜 FACE WITH MEDICAL MASK 🎦	
FACE WITH MEDICAL MASK T	(define-prefix-command 'ring-map)
FACE WITH ROLLING EVES	(global-set-key (kbd ";") 'ring-map)
FACE WITH PLEAKING EYES (8)	;; *** make sure I can still type ;
FACE WITH LOOK OF TRIUMPH 6	,, the nake sale i can seere type ;
ACE WITH NO GOOD GESTURE	;; (global-set-key (kbd "; SPC")
FACE WITH STUCK-OUT TONGUE 😔	;; (lambda () (interactive) (insert ";")))
FACE WITH ONE EYEBROW RAISED 🐵	
FACE SAVOURING DELICIOUS FOOD	;; (global-set-key (kbd "; ; SPC") 'ignore)
FACE WITH OPEN MOUTH VOMITING 🕾	
FACE WITH PARTY HORN AND PARTY HAT 🧭	(global-set-key (kbd "; SPC") 'ignore)
FACE WITH OPEN MOUTH AND COLD SWEAT 😟	(global-set-key (kbd "; '")
FACE WITH UNEVEN EYES AND WAVY MOUTH 🛞	(lambda () (interactive) (insert ";")))
FACE WITH FINGER COVERING CLOSED LIPS 🛞	(Imbad () (Interdetive) (Interdetive)
FACE WITH STUCK-OUT TONGUE AND WINKING EYE 😔	;; ** C-z
FACE WITH STUCK-OUT TONGUE AND TIGHTLY-CLOSED EYES 😁	
CAT FACE 😼	(define-prefix-command 'ring-map-two)
COW FACE 🐺	(global-set-key (kbd "C-z") 'ring-map-two)
DOG FACE	
	;; * setup
BEAR FACE 😇 FROG FACE 🛗	(require 'org) ; necessary
ANGRY FACE	(require 'org) ; necessary
CLOWN FACE	;; ** set default load paths
HORSE FACE	<pre>(let ((default-directory "~/.emacs.d/elpa/"))</pre>
CRYING FACE 😧	
DRAGON FACE 🧐	(append
HUSHED FACE	<pre>(let ((load-path (copy-sequence load-path)));; Shadow</pre>
FEARFUL FACE	(append
	<pre>(copy-sequence (normal-top-level-add-to-load-path '("."))) (copy-sequence (normal-top-level-add-to-load-path '(".")))</pre>
GIRAFFE FACE	<pre>(normal-top-level-add-subdirs-to-load-path))) load-path)))</pre>
	Ioau pach)))
HUGGING FACE	;; ** package system
<pre>#helm-unicode-search* L1 [100 Candidate(s)] C-h m:Help TAB:Act RET/f1/f2/f-n:NthAct C-!:Tog.suspend C- pattorn: face</pre>	emacs 61% 2788 Emacs-Lisp s U: /home/marie/.emacs
pattern: face	

