Contents

1 Tutorial: how to use emacs

1.1 Frequency used shortcut key and very useful stuff

Manually Defined:

- open recent files

C-x C-r [M-x recentf-open-files]: open recent files

- How to input code block quickly "<s + TAB"

Input "<s" and then type <TAB>

1.2 Mode and Buffer Introduction

- major mode: only one
- minor mode: 0-n; for Example: tool-bar, scroll bar

  C-h m: look up all the enabled minor mode

- help buffer

1.3 control command

- c-g: quit
- g: refresh the dired view of the current directory to see change

1.4 Getting Help

M-x find-fuction M-x find-variable C-h k: describe the function a key runs C-h f: describe a function C-h m: get mode-specific information, and this cna list the models currently used

```
(global-set-key (kbd "C-h C-f") 'find-function)
(global-set-key (kbd "C-h C-v") 'find-variable)
(global-set-key (kbd "C-h C-k") 'find-function-on-key
```

1.5 File and Buffer commands

- c-x c-f: open a file
- c-x c-k: kill the buffer reprsenting a file .... not deleteing a file
- C-x s: saves all fiels with a prompt
- C-x C-s: saves file without a prompt
- C-s C-w: saves the file with a different name. Askss you for the name
- c-x b: switch to a different buffer in a window, asks you which buffer to swich to
- C-x C-b: Switches buffers, but shows you the list of buffers in a new window
- m-<: begining of the buffer
- m->: end of the file
- M-x recover-file: recovers the auto-saved file
- M-x write-file: write the buffer to a different file

1.6 Mouse move command

1.6.1 word

- c-f: move forward one character
- c-b: move back one character
- M-f: move forward one world
- M-b: move back one world

1.6.2 line

- c-p: previous line/up
- c-n: next line/down
- c-a: move to the beginning of the a line
- c-e: move to the end of the line

1.6.3 page

- c-v: gage down
- m-v: page up

1.6.4 screen

- c-l: center the screen

1.7 Edit command

- c-d: delete a character
- m-d: delete a word
- c-_: undo
- c-/: undo
- C-g c-/: Redo
- c-w: cut
- c-y: yandk/paste

- m-u: upper case
- m-l: lower case
- m-c: capitalize

1.8 Multiple Windos

- C-M-v: scroll other window
- c-x 2: split top/down
- c-x 3: split left/right
- c-x o: other window

1.9 search

- c-s text: search
- c-s TEXT: case sensitive search
- m-x query-replace <—-> m-%
- m-x replace-string
- M-C-s: search a regexp
- M-s o: searches and shows alll the occurances in an **Occur** buffer. You can click on the lines to jump to those lines.
- m-x grep <enter>

1.10 mark

- c-space: start/toggle marking a region

1.11 check

- m-$: spell check word
- m-x flyspell-mode
- m-x ispell-region: check a small region
- m-x ispell-buffer: check all of the buffer

1.12 shell

- m-x shell: start a bash command line

1.13 Customize variavle, group, mode, function

2 Configuration for project and IDE

2.1 Package Management

- Introduction

MELPA: Milkypostman's Emacs Lisp Package Archives

- M-x package-list-packages

d: delete i: install x: execute

- melpa.org

- Add melpa package

```
  ;;;initialize package
  (require 'package)
  (setq package-archives '(
  ;  ("gnu" . "https://elpa.gnu.org/packages/")
;     ("melpa" . "https://melpa.org/packages/")
   ("melpa-stable" . "https:://stable.melpa.org/packages/"))
  )
  (package-initialize)
```

- Auto install package configuration

```

     (when (>= emacs-major-version 24)
       (require 'package)
       (package-initialize)
       (add-to-list 'package-archives  '(
         ("melpa" . "https://melpa.org/packages/"))))
     (require 'cl)
     ;;add whatever package you want here
     (defvar yubao/packages '(
      company
```

```
 )
  "Default packages")
(defun yubao/packages-installed-p ()
  (loop for pkg in yubao/packages
when (not (package-installed-p pkg)) do (return nil)
finally (return t)))

(unless (yubao/packages-installed-p)
  (message "%s" "Refreshing package database .... ")
  (package-refresh-contents)
  (dolist (pkg yubao/packages)
    (when (not (package-installed-p pkg))
      (package-install pkg))))
```

2.2 linum Mode

(global-linum-mode t) (linum-mode t)

2.3 Company Mode

- company-mode
- What's Company Mode?

Company => company anything

- How to enable company mode?

  (company-mode t);work on current buffer (global-company-mode t);work on all the opened buffer
  Use M-n or M-p to select candidate item

2.4 Speedbar

m-x speedbar <enter> or m-x speed <tab> <enter> :list project files

2.5 Compile

m-x compile

2.6 Debug

c-x ' : jump to the next error. That ' is a back quote on the top left of the keyboard

2.7 Format

- Auto Update the Sequence Number

Example:

1. first
2. second
3. third
4. fourth

Then I want to insert one item: Example:

1. first
2. second
3. Inserted new item
4. third
5. fourth

Therefore, think a question: how to auto sort the list?
Method:
Move the curser to the end, and press 'M' (meta), and then press Return key.
Sorted items:

1. first
2. second
3. Inserted new item
4. third
5. fourth
6. Indent M-x indent-gegion: indents the region

2.8 Show match parents "()"

[menu]=>[Options]=>[Highlight Matching Parentheses]

```
(add-hook 'emacs-lisp-mode-hook 'show-paren-mode)
```

2.9 Highlight current line

```
(global-hl-line-mode t)
```

2.10 Disable backup file (*.~)

```
;;disable backup file (*.~)
(setq make-backup-files nil)
```

2.11 Enable Recent Files

```
(require 'recentf)
(recentf-mode t)
(setq recentf-max-menu-items 25)
;;uncomment this statement if u want to use shortcut key
(global-set-key "\C-x\ \C-r" 'recentf-open-files)
```

2.12 Delete Selection Mode

```
;;add delete selection mode
(delete-selection-mode t)
```

2.13 Install Hungary Delete mode

```
;;config hungry-delete mode
(require 'hungry-delete)
(global-hungry-delete-mode)
```

2.14 Install a Theme

```
(load-theme 'monokai t)
```

```
(require 'smex) ; Not needed if you use package.el
(global-set-key (kbd "M-x") 'smex)
(global-set-key (kbd "M-X") 'smex-major-mode-commands)
;; This is your old M-x.
(global-set-key (kbd "C-c C-c M-x") 'execute-extended-command)
```

2.15 Install swiper and counsel

- swiper
- configuration

```
  (ivy-mode 1)
(setq ivy-use-virtual-buffers t)
(setq enable-recursive-minibuffers t)
(global-set-key "\C-s" 'swiper)
(global-set-key (kbd "C-c C-r") 'ivy-resume)
(global-set-key (kbd "<f6>") 'ivy-resume)
(global-set-key (kbd "M-x") 'counsel-M-x)
(global-set-key (kbd "C-x C-f") 'counsel-find-file)
(global-set-key (kbd "<f1> f") 'counsel-describe-function)
(global-set-key (kbd "<f1> v") 'counsel-describe-variable)
(global-set-key (kbd "<f1> l") 'counsel-find-library)
(global-set-key (kbd "<f2> i") 'counsel-info-lookup-symbol)
(global-set-key (kbd "<f2> u") 'counsel-unicode-char)
(global-set-key (kbd "C-c g") 'counsel-git)
(global-set-key (kbd "C-c j") 'counsel-git-grep)
(global-set-key (kbd "C-c k") 'counsel-ag)
(global-set-key (kbd "C-x l") 'counsel-locate)
(global-set-key (kbd "C-S-o") 'counsel-rhythmbox)
(define-key read-expression-map (kbd "C-r") 'counsel-expression-history)
```

2.16 Install and Configure Smartparens mode

- Install samartparents
- Configure

```
(require 'smartparens-config)
(add-hook 'emacs-lisp-mode-hook 'smartparens-mode)
```

2.17 Configure Javascript IDE

- install js2-mode in Emacs
- Configuration js2-mode

Thde default mode is "javascript mode", use this to change to Javascript IDE:

```
;;configure for js2-mode
(setq auto-mode-alist
      (append
       '(("\\.js\\'" . js2-mode))
auto-mode-alist))
```

- Install nodejs in OS
- Install nodejs-repl
- Configure nodejs-repl set nodejs-repl-command to "nodejs" in ubuntu system

```
;Type M-x nodejs-repl to run Node.js REPL. See also comint-mode to check key bindings.
;You can define key bindings to send JavaScript codes to REPL like below:

(add-hook 'js-mode-hook
  (lambda ()
    (define-key js-mode-map (kbd "C-x C-e") 'nodejs-repl-send-last-sexp)
    (define-key js-mode-map (kbd "C-c C-r") 'nodejs-repl-send-region)
    (define-key js-mode-map (kbd "C-c C-l") 'nodejs-repl-load-file)
    (define-key js-mode-map (kbd "C-c C-z") 'nodejs-repl-switch-to-repl)))
```

3 org-mode basics

3.1 Introduction and Common Configuration

3.1.1 How to enter source code edit mode

- C-c ' (C-c and single quote) to enter into the source code edit mode, and then use it to turn back
- C-c C-k to abort
- Example:

```
;;press "C-c ' " to edit source code
  (message "Emacs lisp")
```

3.2 Schedule and Calenda and Todo

- TODO creating todo and donw items

  shift-RightArrow or C-c C-t: togle TODO state

- C-c C-s: to schedule time
- C-c C-d: to set deadline of time
- C-c a: lookup the schedual

3.2.1 TODO todo

3.2.2 DONE done

3.3 Links

- baidu : www.baidu.com C-c C-l: edit the link

4 Emacs Lisp

4.1 Study Resources

- learnxinyminutes

4.2 Command

- M-: -> :to go to the evaluate buffer where you can evaluate a lisp statement.

For example, "setq" sets a variable to a value: (setq your\$_{\text{var}}$ '123)

- M-x ielm : ELISP, describe-mode for help
- C-x C-e: runs the command eval-last-sexp (found in global-map)
- M-x eval-buffer :run commands on the current buffer

4.3 Elisp Grammer

4.3.1 Example

```
;;set a variable
(setq my-name "yubao")

;;show the variable's value
(message my-name)

;;define a func to show my name
(defun showMyName ()
(interactive);M-x call
(message "Hello, %s" my-name)
)
```

```
;;call "showMyName" fuction
(showMyName)

;;how to bind the key
(global-set-key (kbd "<C-f2>") 'showMyName)
```

4.3.2 Variable

4.3.3 Function

(+ 2 2) p

5 Reference

- MasterEmacsIn21Days
- learnxinyminutes