

# Emacs Settings

Cole Turner

November 1, 2019

## Contents

<b>1</b>	<b>about this file</b>	<b>2</b>
<b>2</b>	<b>appearance</b>	<b>3</b>
2.1	theme . . . . .	3
2.2	mode line . . . . .	4
2.2.1	time and date . . . . .	4
2.2.2	column numbers . . . . .	4
<b>3</b>	<b>autocomplete</b>	<b>4</b>
3.1	spell checking . . . . .	4
<b>4</b>	<b>backups</b>	<b>4</b>
<b>5</b>	<b>bookmarks</b>	<b>4</b>
<b>6</b>	<b>personal dictionary</b>	<b>4</b>
<b>7</b>	<b>helm</b>	<b>4</b>
<b>8</b>	<b>initial buffer</b>	<b>5</b>
<b>9</b>	<b>movement keys</b>	<b>5</b>
<b>10</b>	<b>my commands</b>	<b>5</b>
10.1	transparency . . . . .	5
10.2	comment/uncomment . . . . .	5
10.3	org time heading . . . . .	6
10.4	pdict-add . . . . .	7
10.5	minimap . . . . .	7

<b>11 org</b>	<b>7</b>
11.1 org-mode . . . . .	7
11.1.1 indentation . . . . .	7
11.1.2 capture . . . . .	7
11.1.3 agenda . . . . .	8
11.1.4 emphasized text . . . . .	8
11.1.5 ellipsis . . . . .	8
11.2 org-bullets . . . . .	8
11.3 publish HTML . . . . .	8
11.4 org-journal . . . . .	9
11.5 mind mapping . . . . .	9
<b>12 package archives</b>	<b>9</b>
<b>13 smartparens</b>	<b>9</b>
13.1 pairs . . . . .	10
<b>14 word wrap</b>	<b>10</b>
14.1 visual fill column . . . . .	10
<b>15 shell buffer</b>	<b>10</b>
<b>16 python</b>	<b>10</b>
16.1 devel . . . . .	10
This is my Emacs settings file.	

## 1 about this file

This Dropbox setup should support both Emacs 25 and 26. Upon fresh installation, modify (on Linux) the .desktop file located in /usr/share/applications. This can be found with:

```
ls /usr/share/applications | grep emacs
```

Then edit the file as superuser and change the "Exec" line so that running Emacs will execute the command:

```
emacs -q --load "~/Dropbox/.emacs_config/init.el"
```

OR - just put the previous line as the first line in the .emacs file.

This loads the Dropbox init.el file before establishing the default value of user-emacs-directory.

The following additional packages are required and can be downloaded using "M-x package-install":

- soothe-theme
- company
- helm
- org
- org-bullets
- org-journal
- org-mind-map
- smartparens
- visual-fill-column

## 2 appearance

```
(tool-bar-mode -1)
(menu-bar-mode -1)
(scroll-bar-mode -1)
(set-face-attribute 'default nil :height 150) ;; 130 = 13pt font
```

### 2.1 theme

```
(setq custom-safe-themes
  (quote ("a77ced882e25028e994d168a612c763a4feb8c4ab67c5ff48688654d0264370c" de
(load-theme 'soothe)
(custom-set-faces
 '(org-hide ((t (:background "#110F13" :foreground "#110F13"))))
 ;'(font-lock-comment-face ((t (:background "#110F13" :foreground "#7868B5" :slant ita
)
```

## 2.2 mode line

### 2.2.1 time and date

```
(setq display-time-day-and-date t)
(display-time-mode t)
```

### 2.2.2 column numbers

```
(column-number-mode t)
```

## 3 autocomplete

```
(require 'company)
(global-company-mode t)
(require 'ido)
(ido-mode t)
```

### 3.1 spell checking

```
(setq ispell-program-name "/usr/bin/aspell")
(add-hook 'org-mode-hook 'flyspell-mode)
```

## 4 backups

```
(setq backup-directory-alist '(("." . "~/emacs.d/backup")))
;;(setq backup-directory-alist '(("." . (concat user-emacs-directory "backup"))))
```

## 5 bookmarks

```
(global-set-key (kbd "C-c b") 'bookmark-jump)
```

## 6 personal dictionary

```
(setq ispell-personal-dictionary (expand-file-name ".ispell_pdict" (file-name-directory
```

## 7 helm

```
(require 'helm)
(helm-mode t)
```

```
(global-set-key (kbd "M-x") 'helm-M-x)
(global-set-key (kbd "C-x C-f") 'helm-find-files)
(global-set-key (kbd "C-x C-b") 'helm-buffers-list)
```

## 8 initial buffer

```
(setq initial-buffer-choice t)
```

## 9 movement keys

```
(global-set-key (kbd "s-r") 'beginning-of-line)
;;(global-set-key (kbd "C-f") 'forward-word)
;;(global-set-key (kbd "C-b") 'backward-word)
;;(global-set-key (kbd "M-f") 'sp-forward-sexp)
;;(global-set-key (kbd "M-b") 'sp-backward-sexp)
;;(global-set-key (kbd "M-p") 'sp-up-sexp)
;;(global-set-key (kbd "M-n") 'sp-down-sexp)
;;(global-set-key (kbd "M-P") 'sp-backward-up-sexp)
;;(global-set-key (kbd "M-N") 'sp-backward-down-sexp)
```

## 10 my commands

```
(defun my-commands () (interactive)
  (setq unread-command-events (listify-key-sequence "my-"))
  (call-interactively 'helm-M-x))
(global-set-key (kbd "M-X") 'my-commands)
```

### 10.1 transparency

```
(defun my-transparency () (interactive)
  (let ((b (- (read-key "Set transparency value [0-9]") 48)))
    (if (< b 10) (set-frame-parameter (selected-frame) 'alpha (- 100 (* b 4)))))
  (set-frame-parameter (selected-frame) 'alpha 100)
  (global-set-key (kbd "C-c t") 'my-transparency))
```

### 10.2 comment/uncomment

```
(defun my-comment-or-uncomment () (interactive)
  (if (region-active-p)
```

```

    (call-interactively 'comment-or-uncomment-region)
    (comment-or-uncomment-region (point-at-bol) (point-at-eol))))
(global-set-key (kbd "C-c ;") 'my-comment-or-uncomment)

```

### 10.3 org time heading

```

(require 'org-journal)
(define-key org-journal-mode-map (kbd "C-c s") 'org-edit-headline)
(define-key org-journal-mode-map (kbd "C-c C-<return>") 'insert-journal-entry)
(define-key org-journal-mode-map (kbd "C-c RET") 'insert-timestamp)

(defun insert-journal-entry () "Insert timestamp with new journal entry headline" (interactive)
  (org-return)
  (set-mark (point))
  (sp-backward-whitespace)
  (org-delete-backward-char 1)
  (org-return)
  (org-return)
  (insert "** \n")
  (insert-timestamp))

(defun insert-timestamp () "Insert current date and time inactive timestamp" (interactive)
  (org-time-stamp-inactive '(16))
  (org-return))

(require 'org-journal)
(defun insert-time () "Insert the current 24h time in HH:MM format" (interactive)
  (let ((arr (split-string (current-time-string) ":")))
    (insert (concat (substring (first arr) -2) ":" (nth 1 arr)))))
(defun insert-time-heading () "Insert time as new org heading" (interactive)
  (org-insert-heading-respect-content)
  (insert-time)
  (org-beginning-of-line)
  (set-mark (point))
  (sp-backward-whitespace)
  (org-delete-backward-char 1)
  (org-return)
  (org-return)
  (org-end-of-line)
  (org-return))

```

```
(define-key org-mode-map (kbd "C-c RET") 'insert-time-heading)
```

## 10.4 pdict-add

```
(defun pdict-add () "Add current word to dictionary" (interactive)
  (setq unread-command-events (listify-key-sequence "iy"))
  (if (not (flyspell-check-previous-highlighted-word))
      (setq unread-command-events (listify-key-sequence ""))))
(define-key org-mode-map (kbd "C-c a") 'pdict-add)
```

## 10.5 minimap

```
(global-set-key (kbd "C-c m") 'minimap-mode)
```

# 11 org

## 11.1 org-mode

```
(require 'org)
```

```
(setq org-return-follows-link t)
(global-set-key (kbd "C-c l") 'org-store-link)
```

### 11.1.1 indentation

```
(define-key org-journal-mode-map (kbd "C-c <right>") 'insert-time-heading-demote)
(define-key org-journal-mode-map (kbd "C-c <left>") 'insert-time-heading-promote)
```

```
(defun insert-time-heading-demote () "" (interactive)
  (insert-time-heading)
  (org-demote-subtree))
```

```
(defun insert-time-heading-promote () "" (interactive)
  (insert-time-heading)
  (org-promote-subtree))
```

### 11.1.2 capture

```
(global-set-key (kbd "C-c c") 'org-capture)
(setq org-directory "~/Dropbox/beorg")
```

### 11.1.3 agenda

```
(global-set-key (kbd "C-c a") 'org-agenda)
```

### 11.1.4 emphasized text

```
(setq org-hide-emphasis-markers t)
```

### 11.1.5 ellipsis

```
(setq org-ellipsis "")
```

## 11.2 org-bullets

```
(require 'org-bullets)
(add-hook 'org-mode-hook 'org-bullets-mode)
(setq org-bullets-bullet-list (quote ("→")))
```

## 11.3 publish HTML

```
;;(global-set-key (kbd "C-c p") 'org-publish-project)
;;(global-set-key (kbd "C-c P") 'org-publish-all)
(setq org-publish-project-alist
      '(
("org-html"
 :base-directory "~/Documents/web/src"
 :base-extension "org"
 :publishing-directory "~/Documents/web/html"
 :recursive nil
 :publishing-function org-html-publish-to-html
 :headline-levels 4 ; Just the default for this project.
 :auto-preamble t
 :auto-sitemap t
 :sitemap-sort-folders ignore)
("org-resources"
 :base-directory "~/Documents/web/src"
 :base-extension "css\\|js\\|png\\|jpg\\|gif\\|pdf\\|mp3\\|ogg\\|swf"
 :publishing-directory "~/Documents/web/html"
 :recursive t
 :publishing-function org-publish-attachment
 )
)
```



```
("org-journal"
 :base-directory "~/Documents/journal/"
 :base-extension "org"
 :publishing-directory "~/Documents/web/html/journal"
 :recursive nil
 :publishing-function org-publish-attachment
 )))
```

## 11.4 org-journal

For Emacs 24, download org-journal version 1.13.1 from <https://github.com/bastibe/org-journal/releases>

```
(require 'org-journal)
(global-set-key (kbd "C-c j") 'org-journal-new-entry)
(global-set-key (kbd "C-c J") (lambda () "" (interactive) (org-journal-new-entry t) (r
(setq org-journal-dir "~/Dropbox/journal") ;; needed in customize
(setq org-journal-file-format "%Y%m%d.org"))
```

## 11.5 mind mapping

```
(require 'ox-org)
(require 'cl) ;; this fixed export problem, not sure why
(load (concat user-emacs-directory "elpa/org-mind-map-20180826.2340/org-mind-map.el"))
(org-mind-map-make-node-fn default "Makes a plain box node without double borders" nil

(setq org-mind-map-node-formats (quote (("default" . org-mind-map-default-node))))
```

## 12 package archives

```
(setq package-archives
 (quote
  (("gnu" . "http://elpa.gnu.org/packages/")
   ("melpa" . "http://melpa.milkbox.net/packages/"))))
```

## 13 smartparens

```
(smartparens-global-mode t)
(show-smartparens-global-mode t)
(define-key smartparens-mode-map (kbd "C-S-<left>") 'sp-forward-barf-sexp)
```

```
(define-key smartparens-mode-map (kbd "C-S-<right>") 'sp-forward-slurp-sexp)
(setq sp-use-subword t)
```

### 13.1 pairs

```
(sp-local-pair '(emacs-lisp-mode lisp-interaction-mode) "" nil :actions nil)
(sp-local-pair '(org-mode tex-mode) "$" "$")
(sp-local-pair '(org-mode tex-mode) "\\textit{" "}" :insert "C-S-i")
(sp-local-pair '(org-mode tex-mode) "\\textbf{" "}" :insert "C-S-b")
;;(sp-local-pair '(org-mode tex-mode) "\\left(" "\\right)" :insert "C-c l")
;;(sp-local-pair '(org-mode tex-mode) "\\left[" "\\right]" :insert "C-c L")
(sp-local-pair '(java-mode) "System.out.println(" ")" :trigger "syso")
```

## 14 word wrap

```
(toggle-word-wrap)
(add-hook 'org-mode-hook 'visual-line-mode)
```

### 14.1 visual fill column

```
(require 'visual-fill-column)
(add-hook 'visual-line-mode-hook 'visual-fill-column-mode)
```

## 15 shell buffer

```
(global-set-key (kbd "C-!") 'shell)
```

## 16 python

```
(elpy-enable)
```

### 16.1 devel

```
(when (require 'flycheck nil t)
  (setq elpy-modules (delq 'elpy-module-flymake elpy-modules))
  (add-hook 'elpy-mode-hook 'flycheck-mode))

(require 'py-autopep8)
(add-hook 'elpy-mode-hook 'py-autopep8-enable-on-save)
```