

DOOM Config

Shadorain

July 22, 2020

Contents

1	DOOM Config.el	2
1.1	Initialization Settings	2
1.1.1	Lexical binding	2
1.1.2	Emacs Package Installers	2
1.1.3	Optimizations	2
1.1.4	Tabs to Spaces	3
1.1.5	Username + Email	3
1.2	Theming	3
1.2.1	Theme & Font	3
1.2.2	Extra Colors	3
1.3	Package config	3
1.3.1	Annotate	4
1.3.2	Avy	4
1.3.3	Company	4
1.3.4	ERC	4
1.3.5	Escreen	4
1.3.6	Evil Snipe	4
1.3.7	Gpastel	5
1.3.8	Hlinum	5
1.3.9	Loads	5
1.3.10	Pdf-tools	5
1.3.11	Rainbow-mode	6
1.3.12	Yasnippet-snippets	6
1.3.13	Vimish Folds	6
1.4	Latex	7
1.4.1	Auto-Compile + Refresh (from org)	7
1.5	Org-Mode	7

1.5.1	General org-mode	7
1.5.2	org-startup-folded	8
1.5.3	Custom Links	8
1.5.4	Org Fancy Priorities	8
1.5.5	Org Super Agenda *	8
1.5.6	Org-Superstar	9
1.6	Custom Set Variables	9
1.7	Compile Org -> Elisp	9

1 DOOM Config.el

1.1 Initialization Settings

1.1.1 Lexical binding

```
;;; config.el --- -*- lexical-binding: t -*-
```

1.1.2 Emacs Package Installers

1. MELPA/ELPA

```
(require 'package)
(add-to-list 'package-archives '("melpa" . "https://melpa.org/packages/") t)
```

2. GNU

```
(add-to-list 'package-archives
  '("gnu" . "http://elpa.gnu.org/packages/"))
(package-initialize)
(unless (package-installed-p 'use-package)
  (package-refresh-contents)
  (package-install 'use-package))
(setq package-check-signature nil)
(eval-when-compile
  (require 'use-package))
```

1.1.3 Optimizations

```
(setq auto-window-vscroll nil)
(setq scroll-conservatively 101)
;; (add-to-list 'default-frame-alist '(inhibit-double-buffering . t))
```

```
(setq inhibit-compacting-font-caches t) (setq linum-relative-backend 'display-  
line-numbers-mode) (pixel-scroll-mode)
```

1.1.4 Tabs to Spaces

```
(setq-default indent-tabs-mode nil)
```

1.1.5 Username + Email

```
(setq user-full-name "Shadorain"  
      user-mail-address "shadorain7517@gmail.com")
```

1.2 Theming

1.2.1 Theme & Font

```
(setq  
  doom-font (font-spec :family "Mononoki Nerd Font" :size 16)  
  doom-variable-pitch-font (font-spec :family "Mononoki Nerd Font" :size 16)  
  doom-theme 'shado  
  projectile-project-search-path '("~/xmonad/" "~/Documents/Programming/" "~/dwm/"))  
  
doom-font (font-spec :family "Agave" :size 16) doom-theme 'doom-challenger-  
deep doom-theme 'doom-one data-format-on-save t)
```

1.2.2 Extra Colors

```
(set-face-foreground 'line-number "#a1a1dd") ; Line number colors  
(set-face-foreground 'line-number-current-line "#de286e") ; Current line number color  
(add-to-list 'default-frame-alist '(background-color . "#0f0f17")) ; Background color
```

1.3 Package config

1. Help Message Here are some additional functions/macros that could help you configure Doom: - 'load!' for loading external *.el files relative to this one - 'use-package!' for configuring packages - 'after!' for running code after a package has loaded - 'add-load-path!' for adding directories to the 'load-path', relative to this file. Emacs searches the 'load-path' when you load packages with 'require' or 'use-package'. - 'map!' for binding new keys

1.3.1 Annotate

```
;; (annotate-mode)
```

1.3.2 Avy

```
;; (avy-setup-default)
(global-set-key (kbd "C-c C-j") 'avy-resume)
```

1.3.3 Company

```
(require 'company)
(setq company-idle-delay 0.01
      company-minimum-prefix-length 2)
```

1.3.4 ERC

```
(defun start-erc ()
  "Log into freenode with less keystrokes"
  (interactive)
  (let
    ((password-cache nil))
    (erc
     :server "irc.freenode.net"
     :port "6667"
     :nick erc-nick
     :password erc-password))))
```

1.3.5 Escreen

```
(load "escreen")
(escreen-install)
(global-set-key escreen-prefix-char 'escreen-prefix)
```

1.3.6 Evil Snipe

```
(require 'evil-snipe)
(evil-snipe-mode +1)
(evil-snipe-override-mode +1)
(add-hook 'magit-mode-hook 'turn-off-evil-snipe-override-mode)
```

1.3.7 Gpastel

```
(add-to-list 'load-path "~/.emacs.d/lib/gpastel/")
```

1.3.8 Hlinum

```
(require 'hlinum)
(hlinum-activate)
(setq display-line-numbers-type 'relative) ; Relative line numbers with current line sl
```

1.3.9 Loads

```
(load! "binds")
(load "~/.erc-auth")
```

1.3.10 Pdf-tools

```
(require 'ox-latex)
(require 'pdf-tools)
;; (setq Tex-auto-save t)
;; (setq Tex-parse-self t)
;; (setq-default Tex-master nil)
;; (add-hook 'LaTeX-mode-hook 'visual-line-mode)
;; (add-hook 'LaTeX-mode-hook 'flyspell-mode)
;; (add-hook 'LaTeX-mode-hook 'LaTeX-math-mode)
;; (add-hook 'LaTeX-mode-hook 'turn-on-reftex)
;; (setq reftex-plug-into-AUCTeX t)
;; (setq TeX-PDF-mode t)
;; (pdf-tools-install)
(with-eval-after-load 'pdf-tools
  (require 'pdf-sync))
(use-package pdf-tools
  :init (pdf-tools-install)
  :bind (:map pdf-view-mode-map
    ("T" . pdf-annot-add-text-annotation)
    ("D" . pdf-annot-delete)
    ("t" . pdf-annot-add-highlight-markup-annotation)
    ("j" . image-next-line)
    ("k" . image-previous-line)
    ("l" . image-forward-hscroll)
    ("h" . image-backward-hscroll))
```

```

      ("G" . pdf-view-last-page)
      ("g" . nil)
      ("gg" . pdf-view-first-page)
      ("C-c C-c" . image-toggle-display)
      ("C-s" . isearch-forward))
:config
(setq-default pdf-view-display-size 'fit-page)
:custom
(yas-minor-mode nil)
(pdf-cache-image-limit 32)
(pdf-view-max-image-width 2048)
(pdf-view-resize-factor 1.8)
(pdf-isearch-batch-mode t)
(pdf-annot-activate-created-annotations t))
(setq pdf-sync-backward-display-action t)
(setq pdf-sync-forward-display-action t)

```

1.3.11 Rainbow-mode

```
; (use-package! rainbow-mode)
```

1.3.12 Yasnippet-snippets

```

(use-package yasnippet
  :config
  (yas-global-mode 1)
  (yas-load-directory "~/.doom.d/snippets"))

(yas-load-directory "/usr/share/yasnippet-snippets"))

```

1.3.13 Vimish Folds

```

(setq vimish-fold-find-marks-on-open t)
(turn-off-evil-vimish-fold-mode)
(vimish-fold-from-marks)

(setq vimish-fold-global-mode 1)

```

1.4 Latex

1.4.1 Auto-Compile + Refresh (from org)

These two functions first time running will open the pdf in a split, concurring saves will update the pdf!

```
(setq
  split-width-threshold 0
  split-height-threshold nil)

(global-auto-revert-mode t)
(defun org-export-pdf-then-refresh()
  (org-latex-export-to-pdf))

(defun org-auto-export-on()
  (interactive)
  (cond
    ((string-match "LATEX_CMD: latex" (buffer-string))
     (add-hook 'after-save-hook 'org-export-pdf-then-refresh t t)
     (split-window-right)
     (windmove-right)
     (org-open-file (org-latex-export-to-pdf))
     (windmove-left))))
```

1.5 Org-Mode

1.5.1 General org-mode

```
(after! org
  (setq org-highlight-latex-and-related nil)
  (setq org-hide-emphasis-markers t) ; Hides ugly markup symbols
  (setq org-agenda-skip-scheduled-if-done t
        org-agenda-files (list "~/org/Tech/Emacs/")
        org-ellipsis " ... "
        org-fontify-emphasized-text t
        ;; org-ellipsis ""
        ; org-add-link-type "yt" #'make-yt-link
        org-todo-keywords '((sequence "TODO(t)" "INPROGRESS(i)" "WAITING(w)" "|" "DONE")
        org-todo-keyword-faces
        '(("TODO" :foreground "#943d84" :weight bold :underline t)
          ("WAITING" :foreground "#9f7efe" :weight normal :underline t)
```

```

        ("INPROGRESS" :foreground "#0098dd" :weight normal :underline t)
        ("DONE" :strike-through t :foreground "#2F2F4A" :background nil :weight light)
        ("CANCELLED" :strike-through t :foreground "#de286e" :background nil :weight bold)
;; org-priority-faces '((?A :foreground "#de286e")
;;                      (?B :foreground "#0098dd")
;;                      (?C :foreground "#a1a1dd"))
; org-tags-column -80
))
(setq org-directory "~/org/")

```

1.5.2 org-startup-folded

```
(setq org-startup-folded t)
```

1.5.3 Custom Links

```

; (defun make-yt-link (youtube_id)
;   (browse-url (concat "https://www.youtube.com/embed/" youtube_id)))

```

1.5.4 Org Fancy Priorities

```

(use-package! org-fancy-priorities
  :hook (org-mode . org-fancy-priorities-mode)
  :config
  (setq org-fancy-priorities-list '("" "" "")))

```

1.5.5 Org Super Agenda *

```

(use-package! org-super-agenda
  :after org-agenda
  :init
  (setq org-super-agenda-groups '((:name "Today"
                                   :time-grid t
                                   :scheduled today)
                                   (:name "Due today"
                                   :deadline today)
                                   (:name "Important"
                                   :priority "A")
                                   (:name "Overdue"
                                   :deadline past)
                                   (:name "Due soon"

```



```

                                :deadline future)
                                (:name "Big Outcomes"
                                   :tag "bo"))
:config
(org-super-agenda-mode))

```

```
;; (require 'org-superstar)
;; (setq org-bullets-face-name (quote org-bullet-face))
;; (add-hook 'org-mode-hook (lambda () (org-bullets-mode 1)))
;; (setq
;;   org-superstar-headline-bullets-list '(" " " " " " " " ")
;;   org-superstar-cycle-headline-bullets t
;;   org-superstar-item-bullet-alist t
;;   org-superstar-prettify-item-bullets t)
```

```
(custom-set-variables
;; custom-set-variables was added by Custom.
;; If you edit it by hand, you could mess it up, so be careful.
;; Your init file should contain only one such instance.
;; If there is more than one, they won't work right.
'(package-selected-packages
  (quote
    (vimish-fold rainbow-mode ivy-avy forge evil-snipe evil-multiedit evil-avy))))
(custom-set-faces
;; custom-set-faces was added by Custom.
;; If you edit it by hand, you could mess it up, so be careful.
;; Your init file should contain only one such instance.
;; If there is more than one, they won't work right.
)
```

```
;; (org-babel-load-file (expand-file-name "~/doom.d/config.org"))
```