

Contents

```
;;; $DOOMDIR/config.el -- lexical-binding: t; --
;; Place your private configuration here! Remember, you do not need to
run 'doom ;; sync' after modifying this file!
;; Some functionality uses this to identify you, e.g. GPG configuration,
email ;; clients, file templates and snippets. (setq user-full-name "Zohar
Cochavi" user-mail-address "cochavi.zohar@protonmail.com")
;; Doom exposes five (optional) variables for controlling fonts in Doom.
Here ;; are the three important ones: ;; + 'doom-font' ;; + 'doom-variable-
pitch-font' ;; + 'doom-big-font' – used for 'doom-big-font-mode'; use this
for ;; presentations or streaming. ;; They all accept either a font-spec,
font string ("Input Mono-12"), or xlfed ;; font string. You generally only
need these two: ;; (setq doom-font (font-spec :family "monospace" :size
12 :weight 'semi-light) ;; doom-variable-pitch-font (font-spec :family "sans"
:size (setq doom-font (font-spec :family "Iosevka Extended" :size 14) doom-
variable-pitch-font(font-spec :family "Iosevka Aile" :size 14 :weight 'light)
doom-big-font (font-spec :family "Fira Mono" :size 24))
(custom-set-faces! '(font-lock-comment-face :slant italic) '(font-lock-keyword-
face :slant italic))
;; There are two ways to load a theme. Both assume the theme is installed
and ;; available. You can either set 'doom-theme' or manually load a theme
with the ;; 'load-theme' function. This is the default:
(setq doom-theme 'doom-xresources) (setq doom-themes-padded-modeline
t)
;; If you use 'org' and don't want your org files in the default location
below, ;; change 'org-directory'. It must be set before org loads! (setq org-
directory "~/Documents/.org/")
;; This determines the style of line numbers in effect. If set to 'nil', line ;;
numbers are disabled. For relative line numbers, set this to 'relative'. (setq
display-line-numbers-type 'relative)
;; Here are some additional functions/macros that could help you con-
figure 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 ;; To get information about any of these functions/macros, move the
cursor over ;; the highlighted symbol at press 'K' (non-evil users must press
'C-c c k'). ;; This will open documentation for it, including demos of how
```