EMACS Configurations

Configurations for Doom Emacs

Arc

Contents

EMACS CONFIGURATIONS:

1 DOOM

1.1 Installation

To install Doom Emacs you can use this command:

```
$ git clone --depth 1 https://github.com/hlissner/doom-emacs ~/.emacs.d
$ ~/.emacs.d/bin/doom install
```

You might want to create an alias like this:

```
alias doom='~/.emacs.d/bin/doom'
```

1.2 Restore Configurations

This is to restore Doom Emacs configurations:

```
DOOM=$HOME/Project/NixOS/
ln -sf $DOOM/doom/ ~/.doom.d
```

2 NixOS

Lifehacks for NixOS options and stuffs.

```
;;(add-to-list 'company-backends 'company-nixos-options)
```

3 Bindings

This bind your keys to yourself your emacs configs.

```
(map! :leader
      :desc "Tangle current file" "c t"   #'org-babel-tangle

      :desc "Tangle current file" "t t"   #'org-babel-tangle
      :desc "Highlight"    "t h"   #'hl-todo-mode
      :desc "Time"    "t T"   #'display-time

      :desc "Dired" "o d"   #'dired-jump
```

```
9
10        :desc "Toggle command log mode" "t c m" #'command-log-mode
11        :desc "Show command log buffer" "t c l" #'clm/toggle-command-log-
              buffer)
12
13  (map! :leader
14        (:prefix-map ("T" . "Text")
15
16          :desc "Figlet Border" "f b" #'figlet-border
17          :desc "Figlet Future" "f f" #'figlet-future
18          :desc "Figlet Pagga"  "f p" #'figlet-pagga
19          :desc "Figlet Small"  "f s" #'figlet-small
20          :desc "Figlet Future Border" "f F" #'figlet-future-border
21
22          :desc "List - Lorem Ipsum" "l" #'lorem-ipsum-insert-list
23          :desc "Sentences - Lorem Ipsum" "s" #'lorem-ipsum-insert-
                sentences
24          :desc "Paragraphs - Lorem Ipsum" "p" #'lorem-ipsum-insert-
                paragraphs))
25
26  (map! :leader
27        (:prefix-map ("N" . "NixOS")
28          :desc "NixOS Options" "h" #'helm-nixos-options))
29
30  (evil-better-visual-line-on)
31  (define-key evil-normal-state-map (kbd "j") 'evil-next-visual-line)
32  (define-key evil-normal-state-map (kbd "k") 'evil-previous-visual-line)
33  (define-key evil-normal-state-map (kbd "J") 'evil-scroll-down)
34  (define-key evil-normal-state-map (kbd "K") 'evil-scroll-up)
```

3.1 MODE

```
1  (map! :leader
2        (:prefix-map ("M" . "mode")
3
4          :desc "Shell" "s" #'shell-mode
5       ;; :desc " " "f f" #'
6       ;; :desc " " "f p" #'
7       ;; :desc " " "f s" #'
8          ))
```

4 User Interface

All eye-candy configurations from yours truly.

4.1 Fonts

```
(set-window-margins nil 2)
(setq doom-font (font-spec :family "Iosevka" :size 15)
      doom-variable-pitch-font (font-spec :family "Iosevka" :size 15)
      doom-big-font (font-spec :family "Iosevka" :size 24))
(after! doom-themes
  (setq doom-themes-enable-bold t
        doom-themes-enable-italic t))
(custom-set-faces!
  '(font-lock-comment-face :slant italic)
  '(font-lock-keyword-face :slant italic))

(when (file-exists-p "~/.doom.d/banner")
  (setq +doom-dashboard-banner-padding '(2 . 2)
        +doom-dashboard-banner-file "arkiv.png"
        +doom-dashboard-banner-dir "~/.doom.d/banner"))
```

4.2 Modeline

```
(setq doom-modeline-enable-word-count nil)
;;(set-face-attribute 'header-line nil  :height 36)
(setq-default header-line-format " Archive-Code // 01-Ark // %M")
(setq display-time-mode 1)
```

4.3 Centaur Tabs

```
(setq centaur-tabs-set-bar 'over
      centaur-tabs-set-icons t
      centaur-tabs-gray-out-icons 'buffer
      centaur-tabs-height 18
      centaur-tabs-set-modified-marker t
      centaur-tabs-style "bar"
      centaur-tabs-modified-marker " ")
(map! :leader
      :desc "Toggle tabs on/off"
      "t c" #'centaur-tabs-local-mode)
(evil-define-key 'normal centaur-tabs-mode-map (kbd "g <right>") '
   centaur-tabs-forward        ; default Doom binding is 'g t'
                                           (kbd "g <left>")  '
                                              centaur-tabs-backward
                                                     ; default Doom
                                               binding is 'g T'
                                           (kbd "g <down>")  '
                                              centaur-tabs-forward-
                                              group
```

```
14                                                    (kbd "g <up>")    '
                                                         centaur-tabs-backward
                                                         -group)
```

5 Command Line Interface

6 Emacs Repair

Some utilities to repair this doom that has been lingering since..

6.1 EAF

```
1  ;;(add-to-list 'load-path "~/.emacs.d/.local/straight/build-27.1/eaf")
```

6.2 Projectile

```
1  ;; (setq projectile-mode-line "Projectile")
2  ;; (setq projectile-track-known-projects-automatically 'nil)
```

6.3 TODO Fast Scroll

```
1  (require 'fast-scroll)
2  (fast-scroll-config)
3  (fast-scroll-mode 1)
4  (setq fast-but-imprecise-scrolling 't)
5  (setq jit-lock-defer-time 0)
6  (setq display-line-numbers-type nil)
7  (setq doom-theme 'doom-tomorrow-night)
8  ;;(setq doom-theme 'doom-nord-light)
9  (setq doom-modeline-enable-word-count nil)
```

7 Garbage Collection

Sometimes it is useful to collect your garbage.

```
1  (after! gcmh
2    (setq gcmh-high-cons-threshold 33554432))
```

8 Provide Config File

Provide this config file to be used in `init.el`

```
(provide 'config)
```

9 Extra Packages

```
;; NIXOS
;;(package! nixos-options)
;;(package! helm-nixos-options)
;;(package! company-nixos-options)

;; WINDOW MANAGER
;;(package! exwm)

;; OTHERS
;;(package! origami)

(package! fast-scroll)
(package! lorem-ipsum)
(package! org-bullets)
(package! command-log-mode)
(package! figlet)

(package! doom-themes)
(package! evil-better-visual-line)
(provide 'packages)
```

9.1 Company install

```
(setq company-idle-delay 0)
(setq company-tooltip-limit 6)
(setq company-dabbrev-downcase nil)
(setq company-minimum-prefix-length 1)
(setq company-dabbrev-ignore-case nil)
(setq company-selection-wrap-around t)
(setq company-selection-default 0)
```

9.2 Figlet

```
(defun figlet-border (&optional b e)
  (interactive "r")
```

```
3     (shell-command-on-region b e "toilet -w 200 -f term -F border" (
          current-buffer) t))
4
5   (defun figlet-future (&optional b e)
6     (interactive "r")
7     (shell-command-on-region b e "toilet -w 200 -f future" (current-
          buffer) t))
8
9   (defun figlet-future-border (&optional b e)
10    (interactive "r")
11    (shell-command-on-region b e "toilet -w 200 -f future -F border" (
          current-buffer) t))
12
13  (defun figlet-pagga (&optional b e)
14    (interactive "r")
15    (shell-command-on-region b e "toilet -w 200 -f pagga -F border" (
          current-buffer) t))
16
17  (defun figlet-small (&optional b e)
18    (interactive "r")
19    (shell-command-on-region b e "figlet -f small" (current-buffer) t))
```

10 ORG MODE

```
1   (setq org-hide-emphasis-markers t)
2   (setq org-ellipsis " ")
3
4   ;; Company mode
5   (defun trigger-org-company-complete () (interactive)
6     (if (string-equa(setq org-hide-emphasis-markers t))))
7
8   (setq org-ellipsis " ")
9
10  ;; Company mode
11  (defun trigger-org-company-complete ()
12    (interactive)
13    (if (string-equal "#" (string (preceding-char)))
14      (progn
15        (insert "+")
16        (company-complete))
17      (insert "+")))
18
19  (eval-after-load 'org '(define-key org-mode-map
20                 (kbd "+") 'trigger-org-company-complete))
21
22  (custom-set-faces
23    '(org-level-1 ((t (:inherit outline-1 :height 1.2))))
24    '(org-level-2 ((t (:inherit outline-2 :height 1.1)))))
```

```
  '(org-level-3 ((t (:inherit outline-3 :height 1.0))))
  '(org-level-4 ((t (:inherit outline-4 :height 1.0))))
  '(org-level-5 ((t (:inherit outline-5 :height 1.0)))))
```

10.1 Prettify ORG MODE

```
;; Hook in HOOKS FOREVER
(prettify-symbols-mode t)
(global-prettify-symbols-mode t)
```

11 Globals

International variables on emacs.

```
(global-hl-todo-mode 1)
(global-hi-lock-mode 1)
```

12 Hooks Forever

Add-Hook to your emacs

12.1 INIT Hooks

```
(add-hook 'after-init-hook 'global-company-mode)
(add-hook 'after-init-hook 'display-time-mode)
```

12.2 ORG Hooks

```
(add-hook 'org-mode-hook (lambda () (hl-todo-mode 1)))
(add-hook 'org-mode-hook (lambda () (org-bullets-mode 1)))
(add-hook 'org-mode-hook
          '(lambda ()
             (add-hook 'write-contents-functions
                       'delete-trailing-whitespace)))
```

12.3 OTHER Hooks

```
;;(add-hook 'fast-scroll-start-hook (lambda () (flycheck-mode -1)))
;;(add-hook 'fast-scroll-end-hook (lambda () (flycheck-mode 1)))
```

13 Defaults

```
(setq figlet-default-font "Future")
```