

Config

Leo Aparisi de Lannoy

June 20, 2024

Contents

| | | |
|----------|--------------------------|----------|
| 1 | Compilation | 3 |
| 2 | Basic | 4 |
| 2.1 | ID | 4 |
| 2.2 | Async | 4 |
| 2.3 | Good defaults | 4 |
| 2.3.1 | Browser | 4 |
| 2.4 | Visual | 5 |
| 2.4.1 | Font | 5 |
| 2.4.2 | Theme | 5 |
| 2.4.3 | Default visual | 5 |
| 2.4.4 | Starting image | 5 |
| 2.5 | Marginalia | 5 |
| 2.6 | File Templates | 5 |
| 2.7 | Editor config | 6 |
| 3 | Org-Mode | 6 |
| 3.1 | Defaults | 6 |
| 3.2 | Babel | 6 |
| 3.2.1 | Completion | 7 |
| 3.3 | Visuals | 7 |
| 3.3.1 | Org-modern | 7 |

| | | |
|----------|--------------------------------|-----------|
| 3.3.2 | Org-Appear | 9 |
| 3.3.3 | General | 9 |
| 3.3.4 | Ligatures | 10 |
| 3.3.5 | Latex improvement | 10 |
| 3.4 | Bullets | 10 |
| 3.5 | Agenda | 10 |
| 3.5.1 | Visual | 10 |
| 3.6 | Roam | 11 |
| 3.6.1 | Defaults | 11 |
| 3.6.2 | Daily | 11 |
| 3.6.3 | Visuals | 11 |
| 3.7 | Ob-async | 12 |
| 3.8 | Pandoc import | 12 |
| 3.9 | Export | 13 |
| 3.9.1 | Preview | 13 |
| 3.10 | Zotero Integration | 14 |
| 3.11 | Org-Chef | 14 |
| 3.12 | Bibtex-Integration | 14 |
| 3.12.1 | Citar | 14 |
| 3.12.2 | Org-Roam integration | 15 |
| 3.13 | Latex templates | 16 |
| 3.13.1 | Preview | 16 |
| 3.13.2 | Article | 17 |
| 3.13.3 | Beamer | 19 |
| 3.13.4 | Export | 21 |
| 3.14 | Capture | 22 |
| 3.14.1 | Doct | 22 |
| 4 | Corfu | 22 |
| 5 | Jinx | 23 |
| 6 | LSP | 23 |
| 6.1 | Keybindings | 23 |
| 6.2 | LSP-Bridge | 23 |

| | | |
|------|------------------------|----|
| 6.3 | Digestif | 24 |
| 6.4 | LTeX | 24 |
| 7 | VLFI | 25 |
| 8 | PDF-Tools | 25 |
| 8.1 | Dark mode | 25 |
| 9 | Option key Fix | 25 |
| 10 | Centaur | 25 |
| 11 | RSS | 26 |
| 11.1 | Visual | 26 |
| 12 | Email | 29 |
| 12.1 | mu4e | 29 |
| 12.2 | Notification | 30 |
| 13 | Latex | 30 |
| 14 | Flycheck | 31 |
| 15 | TRAMP | 31 |

1 Compilation

```
;; brew tap railwaycat/emacsmacport
;; brew install emacs-mac --with-mac-metal --with-natural-title-bar
↩ --with-native-compilation --with-xwidget
```

2 Basic

2.1 ID

```
(setq user-full-name "Leo Aparisi de Lannoy"
      user-mail-address "leoaparisi@gmail.com")
```

2.2 Async

```
(setq dired-async-mode 1)
```

2.3 Good defaults

```
(setq-default
  delete-by-moving-to-trash t                ; Delete files to trash
  window-combination-resize t                ; take new window space from
  ↪ all other windows (not just current)
  x-stretch-cursor t)                       ; Stretch cursor to the glyph
  ↪ width
(add-to-list 'default-frame-alist '(fullscreen . maximized))

(setq undo-limit 80000000                    ; Raise undo-limit to 80Mb
      evil-want-fine-undo t                  ; By default while in insert
      ↪ all changes are one big blob. Be more granular
      auto-save-default t                    ; Nobody likes to loose work,
      ↪ I certainly don't
      truncate-string-ellipsis "...")
      scroll-margin 1)                       ; It's nice to maintain a
      ↪ little margin

(setq global-subword-mode 1)                  ; Iterate through
      ↪ CamelCase words
```

2.3.1 Browser

```
(setq browse-url-chrome-program "brave")
```

2.4 Visual

2.4.1 Font

```
(setq doom-font (font-spec :family "Iosevka" :size 14)
      doom-variable-pitch-font (font-spec :family "Lato")
      doom-unicode-font (font-spec :family "Iosevka")
      doom-big-font (font-spec :family "Iosevka" :size 24)
      doom-serif-font (font-spec :family "Iosevka Aile" :weight 'light))
```

2.4.2 Theme

1. Default theme

```
(load-theme 'catppuccin t t)
(setq doom-theme 'catppuccin)
(setq catppuccin-flavor 'mocha) ;; or 'latte, 'macchiato, or 'mocha
(catppuccin-reload)
```

2.4.3 Default visual

2.4.4 Starting image

```
;; (setq rancy-splash-image (expand-file-name "themes/doom-emacs-bw-light.svg"
↪ doom-user-dir))
```

2.5 Marginalia

```
(package! info-colors)
```

```
(after! info-colors
  :commands (info-colors-fontify-node))
(add-hook 'Info-selection-hook #'info-colors-fontify-node)
```

2.6 File Templates

```
(set-file-template! "\\\\.tex$" :trigger "__" :mode 'latex-mode)
(set-file-template! "\\\\.org$" :trigger "__" :mode 'org-mode)
```

2.7 Editor config

```
(setq display-line-numbers-type `relative)
(setq-default tab-width 4)
```

3 Org-Mode

3.1 Defaults

```
(setq org-directory "~/org/"
      org-agenda-files (list org-directory)           ; Seems like the
      ↪ obvious place.
      org-use-property-inheritance t                 ; It's convenient
      ↪ to have properties inherited.
      org-log-done 'time                             ; Having the time a
      ↪ item is done sounds convenient.
      org-list-allow-alphabetical t                   ; Have a. A. a) A)
      ↪ list bullets.
      org-catch-invisible-edits 'smart                ; Try not to
      ↪ accidentally do weird stuff in invisible regions.
      org-export-with-sub-superscripts '{}            ; Don't treat lone
      ↪ _ / ^ as sub/superscripts, require _{} / ^{}.
      org-export-allow-bind-keywords t                ; Bind keywords can
      ↪ be handy
      org-image-actual-width '(0.9))                 ; Make the
      ↪ in-buffer display closer to the exported result..#+end_src
```

3.2 Babel

```
(setq org-babel-default-header-args
      '(:session . "none")
        (:results . "replace")
        (:exports . "code")
        (:cache . "no")
        (:noweb . "no")
        (:hlines . "no")
        (:tangle . "no")
        (:comments . "link")))
```

3.2.1 Completion

```
(package! org-block-capf :recipe (:host github :repo "xenodium/org-block-capf"))
```

```
(use-package! org-block-capf
  :after org
  :hook (org-mode . org-block-capf-add-to-completion-at-point-functions))
```

3.3 Visuals

3.3.1 Org-modern

```
(package! org-modern)
```

```
(use-package! org-modern
:after org
:hook (org-mode . org-modern-mode)
:config
(setq org-modern-star '(" " " " " " " " " " " " " " " " " "))
      org-modern-table-vertical 1
      org-modern-table-horizontal 0.2
      org-modern-list '((43 . " ")
                        (45 . "-")
                        (42 . ".•"))

org-modern-todo-faces
'(("TODO" :inverse-video t :inherit org-todo)
 ("PROJ" :inverse-video t :inherit +org-todo-project)
 ("STRT" :inverse-video t :inherit +org-todo-active)
 ("[-]" :inverse-video t :inherit +org-todo-active)
 ("HOLD" :inverse-video t :inherit +org-todo-onhold)
 ("WAIT" :inverse-video t :inherit +org-todo-onhold)
 ("[?]" :inverse-video t :inherit +org-todo-onhold)
 ("KILL" :inverse-video t :inherit +org-todo-cancel)
 ("NO"   :inverse-video t :inherit +org-todo-cancel))

org-modern-footnote
(cons nil (cadr org-script-display))

org-modern-block-fringe nil

org-modern-block-name
'((t . t)
  ("src" ">>" "<<"))
```

```

("example" "»-" "-«")
("quote" " " " ")
("export" " " " ")
org-modern-progress nil
org-modern-priority nil
org-modern-horizontal-rule (make-string 36 ?)
org-modern-keyword
'((t . t)
  ("title" . " ")
  ("subtitle" . " ")
  ("author" . " ")
  ("email" . #(" " 0 1 (display (raise -0.14))))
  ("date" . " ")
  ("property" . " ")
  ("options" . " ")
  ("startup" . " ")
  ("macro" . " ")
  ("bind" . #(" " 0 1 (display (raise -0.1))))
  ("bibliography" . " ")
  ("print_bibliography" . #(" " 0 1 (display (raise -0.1))))
  ("cite_export" . " ")
  ("print_glossary" . #(" " 0 1 (display (raise -0.1))))
  ("glossary_sources" . #(" " 0 1 (display (raise -0.14))))
  ("include" . " ")
  ("setupfile" . " ")
  ("html_head" . " ")
  ("html" . " ")
  ("latex_class" . " ")
  ("latex_class_options" . #(" " 1 2 (display (raise -0.14))))
  ("latex_header" . " ")
  ("latex_header_extra" . " ")
  ("latex" . " ")
  ("beamer_theme" . " ")
  ("beamer_color_theme" . #(" " 1 2 (display (raise -0.12))))
  ("beamer_font_theme" . " ")
  ("beamer_header" . " ")
  ("beamer" . " ")
  ("attr_latex" . " ")
  ("attr_html" . " ")
  ("attr_org" . " ")

```



```

    ("call" . #(" " 0 1 (display (raise -0.15))))
    ("name" . " ")
    ("header" . ">")
    ("caption" . " ")
    ("results" . " "))
(custom-set-faces! '(org-modern-statistics :inherit
  ⇨ org-checkbox-statistics-todo)))

```

3.3.2 Org-Appear

```
(package! org-appear :recipe (:host github :repo "awth13/org-appear"))
```

```
(add-hook 'org-mode-hook #'org-appear-mode)
```

3.3.3 General

```

(setq org-src-fontify-natively t
      org-fontify-whole-heading-line t
      org-fontify-done-headline t
      org-fontify-quote-and-verse-blocks t
      org-startup-with-inline-images t
      org-startup-indented t

      ;; Org styling, hide markup etc.
      org-pretty-entities t
      )

(setq org-ellipsis "  "
      org-hide-leading-stars t
      org-priority-highest ?A
      org-priority-lowest ?E
      org-priority-faces
      '((?A . 'nerd-icons-red)
        (?B . 'nerd-icons-orange)
        (?C . 'nerd-icons-yellow)
        (?D . 'nerd-icons-green)
        (?E . 'nerd-icons-blue)))

```

```
(custom-set-faces!
  '(outline-1 :weight extra-bold :height 1.25)
  '(outline-2 :weight bold :height 1.15)
  '(outline-3 :weight bold :height 1.12)
  '(outline-4 :weight semi-bold :height 1.09)
  '(outline-5 :weight semi-bold :height 1.06)
  '(outline-6 :weight semi-bold :height 1.03)
  '(outline-8 :weight semi-bold)
  '(outline-9 :weight semi-bold))
(custom-set-faces!
  '(org-document-title :height 1.2))
```

3.3.4 Ligatures

3.3.5 Latex improvement

```
(setq org-highlight-latex-and-related '(native script entities))
```

```
(require 'org-src)
(add-to-list 'org-src-block-faces '("latex" (:inherit default :extend t)))
```

```
(package! org-fragtog)
```

```
;; (add-hook 'org-mode-hook #'org-latex-preview-auto-mode)
(add-hook 'org-mode-hook #'org-fragtog-mode)
```

3.4 Bullets

```
(setq org-list-demote-modify-bullet '(("+" . "-") ("- " . "+") ("*" . "+") ("1." .
↳ "a.")))
```

3.5 Agenda

3.5.1 Visual

```
(after! org-agenda
  (setq org-agenda-deadline-faces
    '((1.001 . error)
      (1.0 . org-warning)
      (0.5 . org-upcoming-deadline))
```

```
(0.0 . org-upcoming-distant-deadline))))
```

3.6 Roam

3.6.1 Defaults

```
;; (use-package! org-roam
;;   :after org
;;   :config
;;   (setq
;;     org-enable-roam-support t
;;     org-roam-directory (concat org-directory "/Roam")
;;     org-roam-v2-ack t))
```

3.6.2 Daily

```
;; (setq org-roam-dailies-directory "daily/")

;; (setq org-roam-dailies-capture-templates
;;   '(("d" "default" entry
;;     "* %?"
;;     :target (file+head "%<%Y-%m-%d>.org"
;;       "#+title: %<%Y-%m-%d>\n"))))
```

3.6.3 Visuals

1. UI and visualization

```
;; (package! org-roam-ui)
;; (package! websocket)
```

```
;; (defadvice! doom-modeline--buffer-file-name-roam-aware-a (orig-fun)
;;   :around #'doom-modeline-buffer-file-name ; takes no args
;;   (if (s-contains-p org-roam-directory (or buffer-file-name ""))
;;       (replace-regexp-in-string
;;         "\\\(?:^\\\\|\\.*/\\\\)\\\\\\([0-9]\\\\{4\\\\}\\\\)\\\\\\([0-9]\\\\{2\\\\}\\\\)\\\\\\([0-9]\\\\{2\\\\}\\\\)\\\\\\([0-9]*-"
```

```
;;      "\\1-\\2-\\3) "
;;      (subst-char-in-string ?_ ?  buffer-file-name))
;;      (funcall orig-fun)))
;; (use-package! websocket
;;   :after org-roam)
;; (use-package! org-roam-ui
;;   :after org-roam
;;   :commands org-roam-ui-open
;;   :hook (org-roam . org-roam-ui-mode)
;;   :config
;;   (setq org-roam-ui-sync-theme t
;;         org-roam-ui-follow t
;;         org-roam-ui-update-on-save t
;;         org-roam-ui-open-on-start t)
;;   (require 'org-roam) ; in case autoloaded
;;   (defun org-roam-ui-open ()
;;     "Ensure the server is active, then open the roam graph."
;;     (interactive)
;;     (unless org-roam-ui-mode (org-roam-ui-mode 1))
;;     (browse-url--browser (format "http://localhost:%d"
; ↵ org-roam-ui-port))))
```

3.7 Ob-async

3.8 Pandoc import

```
(package! org-pandoc-import
  :recipe (:host github
           :repo "tecosaur/org-pandoc-import"
           :files ("*.el" "filters" "preprocessors")))
```

```
(use-package! org-pandoc-import
  :after org)
```

3.9 Export

3.9.1 Preview

```
;; (map! :map org-mode-map

;;      :localleader
;;      :desc "View exported file" "v" #'org-view-output-file)

;; (defun org-view-output-file (&optional org-file-path)
;;   "Visit buffer open on the first output file (if any) found, using
  ↪ `org-view-output-file-extensions'"
;;   (interactive)
;;   (let* ((org-file-path (or org-file-path (buffer-file-name) ""))
;;          (dir (file-name-directory org-file-path))
;;          (basename (file-name-base org-file-path))
;;          (output-file nil))
;;     (dolist (ext org-view-output-file-extensions)
;;       (unless output-file
;;         (when (file-exists-p
;;               (concat dir basename "." ext))
;;           (setq output-file (concat dir basename "." ext))))))
;;     (if output-file
;;       (if (member (file-name-extension output-file)
;;                   org-view-external-file-extensions)
;;         (browse-url-xdg-open output-file)
;;         (pop-to-buffer (or (find-buffer-visiting output-file)
;;                           (find-file-noselect output-file))))
;;       (message "No exported file found"))))

;; (defvar org-view-output-file-extensions '("pdf" "md" "rst" "txt" "tex"
  ↪ "html")
;;   "Search for output files with these extensions, in order, viewing the first
  ↪ that matches")
;; (defvar org-view-external-file-extensions '("html")
;;   "File formats that should be opened externally.")
```

3.10 Zotero Integration

```
;; (package! zotxt)
```

```
;; (use-package! zotxt  
;;   :after org)
```

3.11 Org-Chef

```
(package! org-chef)
```

```
(use-package! org-chef  
  :after org  
  :commands (org-chef-insert-recipe org-chef-get-recipe-from-url))
```

3.12 Bibtex-Integration

3.12.1 Citar

```
(package! org-cite-csl-activate :recipe (:host github :repo  
  ↪ "andras-simonyi/org-cite-csl-activate"))
```

```
(use-package! citar  
  :defer t  
  :no-require  
  :custom  
  (org-cite-global-bibliography '("~/org/Lecture_Notes/MyLibrary.bib"))  
  (citar-bibliography org-cite-global-bibliography)  
  (citar-symbols  
    '(note ,(nerd-icons-octicon "nf-oct-note" :face 'nerd-icons-blue :v-adjust  
      ↪ -0.3) . " ")  
    '(link ,(nerd-icons-octicon "nf-oct-link" :face 'nerd-icons-orange  
      ↪ :v-adjust 0.01) . " "))  
  :hook  
  (org-mode . citar-capf-setup))
```

```
(use-package! citar-embark  
  :after citar embark  
  :no-require)
```

```
:config (citar-embark-mode))
```

```
(use-package! oc-csl
  :after oc
  :config
  (setq org-cite-csl-styles-dir "~/Zotero/styles/"))
(after! oc
  (setq org-cite-export-processors '((t csl))))
```

```
(use-package! oc-csl-activate
  :after org
  :config
  (setq org-cite-activate-processor 'csl-activate)
  (setq org-cite-csl-activate-use-document-style t)
  (setq org-cite-csl-activate-use-document-locale t)
  (add-hook! 'org-mode-hook
    (cursor-sensor-mode 1)
    (org-cite-csl-activate-render-all)))
```

3.12.2 Org-Roam integration

```
;; (use-package! citar-org-roam
;;   :after citar org-roam
;;   :config (citar-org-roam-mode))
;; (setq org-roam-capture-templates
;;   '(("d" "default" plain
;;     "%?"
;;     :target
;;     (file+head
;;       "%<%Y%m%d%H%M%S>-${slug}.org"
;;       "#+title: ${title}\n")
;;     :unnarrowed t)
;;     ("n" "literature note" plain
;;       "%?"
;;       :target
;;       (file+head
;;         "%(expand-file-name \"literature\"
; org-roam-directory)/${citekey}.org"
```

```
;;          "#+title: ${citekey}. ${title}.\n#+created: %U\n#+last_modified:
↪ %U\n\n")
;;          :unnarrowed t)))
;; (setq citar-org-roam-capture-template-key "n")
```

3.13 Latex templates

3.13.1 Preview

1. PNG

```
(after! org
  ;; ORG LATEX PREVIEW
  (setq org-format-latex-options
    (plist-put org-format-latex-options :background "Transparent"))
  (setq org-format-latex-options
    (plist-put org-format-latex-options :scale 1))
  (setq org-preview-latex-default-process 'dvisvgm)
  (setq org-preview-latex-image-directory "~/.cache/ltximg/")
)
```

2. Header

```
(setq org-format-latex-header "\\documentclass[12pt]
{article}
\\usepackage[usenames]{xcolor}
\\usepackage{booktabs}
\\pagestyle{empty}           % do not remove
% The settings below are copied from fullpage.sty
\\setlength{\\textwidth}{\\paperwidth}
\\addtolength{\\textwidth}{-3cm}
\\setlength{\\oddsidemargin}{1.5cm}
\\addtolength{\\oddsidemargin}{-2.54cm}
\\setlength{\\evensidemargin}{\\oddsidemargin}
\\setlength{\\textheight}{\\paperheight}
\\addtolength{\\textheight}{-\\headheight}
\\addtolength{\\textheight}{-\\headsep}
\\addtolength{\\textheight}{-\\footskip}
\\addtolength{\\textheight}{-3cm}
\\setlength{\\topmargin}{1.5cm}
```



```

\\addtolength{\\topmargin}{-2.54cm}
% my custom stuff
\\usepackage{xfrac}
\\usepackage{siunitx}
\\usepackage{diffcoeff}
\\usepackage{nicematrix}
\\usepackage[varbb]{newpxmath}
\\DeclareMathOperator{\\Var}{Var}
\\DeclareMathOperator{\\cov}{Cov}
\\DeclareMathOperator{\\E}{\\mathbb{E}}
\\DeclareMathOperator*{\\argmax}{arg\\,max}
\\DeclareMathOperator*{\\argmin}{arg\\,min}
")

```

3.13.2 Article

```

(with-eval-after-load 'ox-latex
(add-to-list 'org-latex-classes
  ("article"
    "\\documentclass[12pt]{article}
\\usepackage[american]{babel}
\\usepackage[margin=1.25in]{geometry}
\\usepackage{parskip}
\\usepackage{booktabs}
\\usepackage{float}
\\usepackage{microtype}
\\usepackage{graphicx}
\\usepackage{mathtools}
\\usepackage{amsthm}
\\usepackage{amssymb}
\\usepackage{bm}
\\usepackage[no-math]{newpxtext}
\\usepackage[varbb]{newpxmath}
\\usepackage{xfrac}
\\usepackage{siunitx}
\\usepackage{caption}
\\captionsetup{labelfont=bf,font={small,singlespacing}}
\\usepackage{subcaption}
\\usepackage{cancel}
\\usepackage{setspace}

```

```

\\usepackage{xcolor}
\\usepackage{diffcoeff}
\\usepackage{nicematrix}
\\usepackage{braket}
\\usepackage{enumitem}
\\usepackage{acronym}
\\usepackage{footmisc}
\\usepackage[authoryear,longnamesfirst]{natbib}
\\usepackage{xurl}
\\onehalfspacing{}
\\bibliographystyle{ecta}
\\DeclareMathOperator{\\Var}{Var}
\\DeclareMathOperator{\\Cov}{Cov}
\\DeclareMathOperator{\\E}{\\mathbb{E}}
\\DeclareMathOperator*{\\argmax}{arg\\,max}
\\DeclareMathOperator*{\\argmin}{arg\\,min}
\\newcommand{\\Et}[2]{\\E_{#2} \\left[#1\\right]}
\\newcommand{\\Covt}[3]{\\cov_{#3}\\left(#1, #2\\right)}
\\newcommand{\\Vart}[2]{\\Var_{#2} \\left[#1\\right]}
\\DeclarePairedDelimiter\\abs{\\lvert}{\\rvert}
\\DeclarePairedDelimiter\\norm{\\lVert}{\\rVert}
\\DeclarePairedDelimiterX\\innerp[2]{\\langle}{\\rangle}{#1,#2}
\\theoremstyle{plain}% default
\\newtheorem{thm}{Theorem}
\\newtheorem{lem}[thm]{Lemma}
\\newtheorem{prop}[thm]{Proposition}
\\newtheorem*{cor}{Corollary}
\\theoremstyle{definition}
\\newtheorem{defn}{Definition}
\\newtheorem{exmp}{Example}
\\providecommand*{\\defnautorefname}{Definition}
\\theoremstyle{remark}
\\newtheorem*{rem}{Remark}
\\newtheorem*{note}{Note}
\\newtheorem{case}{Case}
\\renewcommand{\\leq}{\\leqslant}
\\renewcommand{\\geq}{\\geqslant}
\\usepackage{hyperref}
\\usepackage[] {cleveref}
[NO-DEFAULT-PACKAGES]

```

```
[PACKAGES]
[EXTRA] "
    ("\\section{%s}" . "\\section*{%s}")
    ("\\subsection{%s}" . "\\subsection*{%s}")
    ("\\subsubsection{%s}" . "\\subsubsection*{%s}")
    ("\\paragraph{%s}" . "\\paragraph*{%s}"))))
```

3.13.3 Beamer

```
(setq org-beamer-frame-level 2)
```

```
(setq org-beamer-theme "[progressbar=frametitle, titleformat=smallcaps,
↪ numbering=fraction]metropolis")
```

Define Beamer class:

```
(with-eval-after-load 'ox-latex
(add-to-list 'org-latex-classes
  ("beamer"
   "\\documentclass[c]{beamer}
\\usepackage[american]{babel}
\\settheme[progressbar=frametitle, titleformat=smallcaps,
↪ numbering=fraction]{metropolis}
\\usepackage{parskip}
\\usepackage{booktabs}
\\usepackage{float}
\\usepackage{microtype}
\\usepackage{graphicx}
\\usepackage{mathtools}
\\usepackage{amsthm}
\\usepackage{amssymb}
\\usepackage{bm}
\\usepackage[no-math]{newpxtext}
\\usepackage[varbb]{newpxmath}
\\usepackage{xfrac}
\\usepackage{siunitx}
\\usepackage{caption}
\\captionsetup{labelfont=bf,font={small,singlespacing}}
\\usepackage{subcaption}
```

```

\\usepackage{cancel}
\\usepackage{setspace}
\\usepackage{xcolor}
\\usepackage[ISO]{diffcoeff}
\\usepackage{nicematrix}
\\usepackage{braket}
\\usepackage{enumitem}
\\usepackage{acronym}
\\usepackage{footmisc}
\\usepackage[authoryear,longnamesfirst]{natbib}
\\usepackage{xurl}
\\usepackage{appendixnumberbeamer}
\\usepackage{dirtytalk}
\\DeclareMathOperator{\\Var}{Var}
\\DeclareMathOperator{\\Cov}{Cov}
\\DeclareMathOperator{\\E}{\\mathbb{E}}
\\DeclareMathOperator*{\\argmax}{arg\\,max}
\\DeclareMathOperator*{\\argmin}{arg\\,min}
\\newcommand{\\Et}[2]{\\E_{#2} \\left[#1\\right]}
\\newcommand{\\Covt}[3]{\\cov_{#3}\\left(#1, #2\\right)}
\\newcommand{\\Vart}[2]{\\Var_{#2} \\left[#1\\right]}
\\DeclarePairedDelimiter\\abs{\\lvert}{\\rvert}
\\DeclarePairedDelimiter\\norm{\\lVert}{\\rVert}
\\DeclarePairedDelimiterX\\innerp[2]{\\langle}{\\rangle}{#1,#2}
\\theoremstyle{plain}% default
\\newtheorem{thm}{Theorem}
\\newtheorem{lem}[thm]{Lemma}
\\newtheorem{prop}[thm]{Proposition}
\\newtheorem*{cor}{Corollary}
\\theoremstyle{definition}
\\newtheorem{defn}{Definition}
\\newtheorem{exmp}{Example}
\\providecommand*{\\defnautorefname}{Definition}
\\theoremstyle{remark}
\\newtheorem*{rem}{Remark}
\\newtheorem*{note}{Note}
\\newtheorem{case}{Case}
\\renewcommand{\\leq}{\\leqslant}
\\renewcommand{\\geq}{\\geqslant}
\\definecolor{textcolor}{HTML}{2E3440}

```

```

\\definecolor{titlecolor}{HTML}{a3be8c}
\\definecolor{alertcolor}{HTML}{BF616A}
\\definecolor{bgcolor}{HTML}{ECEFF4}
\\definecolor{barcolor}{HTML}{88C0D0}
\\definecolor{bgbarcolor}{HTML}{D8DEE9}
\\setbeamercolor{progress bar}{fg=barcolor,bg=bgbarcolor}
\\setbeamercolor{frametitle}{fg=titlecolor,bg=bgcolor}
\\setbeamercolor{normal text}{fg=textcolor,bg=bgcolor}
\\setbeamercolor{alerted text}{fg=alertcolor,bg=bgcolor}
\\setbeamercolor{example text}{fg=examplecolor}
\\setbeamercovered{dynamic}
\\usecolortheme{rose}
[NO-DEFAULT-PACKAGES]
[PACKAGES]
[EXTRA] "
    ("\\section{%s}" . "\\section*{%s}")
    ("\\subsection{%s}" . "\\subsection*{%s}")
    ("\\subsubsection{%s}" . "\\subsubsection*{%s}")
    ("\\paragraph{%s}" . "\\paragraph*{%s}")
    ("\\subparagraph{%s}" . "\\subparagraph*{%s}"))))

```

3.13.4 Export

```

(setq org-latex-pdf-process '("LC_ALL=en_US.UTF-8 latexmk -f -pdf -%latex
↪ -shell-escape -interaction=nonstopmode -output-directory=%o %f"))

```

```

(setq org-latex-tables-booktabs t
      org-latex-hyperref-template "\\providecolor{url}{HTML}{81a1c1}
\\providecolor{link}{HTML}{d08770}
\\providecolor{cite}{HTML}{d08770}
\\hypersetup{
pdfauthor={%a},
pdftitle={%t},
pdfkeywords={%k},
pdfsubject={%d},
pdfcreator={%c},
pdflang={%L},
breaklinks=true,
colorlinks=true,
linkcolor=link,

```

```

urlcolor=url,
citecolor=cite
}
"

org-latex-reference-command "\\cref{%s}")

```

1. Preview

```

;; Use pdf-tools to open PDF files
(setq TeX-view-program-selection '((output-pdf "PDF Tools"))
TeX-source-correlate-start-server t)

```

```

;; Update PDF buffers after successful LaTeX runs
(add-hook 'TeX-after-compilation-finished-functions
  ↪ #'TeX-revert-document-buffer)

```

2. Code blocks

```

(package! engrave-faces)

```

```

(use-package! engrave-faces-latex
  :after ox-latex)
(setq org-latex-listings 'engraved)
(setq org-latex-engraved-theme 'doom-nord)

```

3.14 Capture

3.14.1 Doct

```

(package! doct)

```

4 Corfu

```

(setq corfu-popupinfo-delay 0)

```

5 Jinx

```
(package! jinx)
```

```
(use-package! jinx
  :defer t
  :init
  (add-hook 'doom-init-ui-hook #'global-jinx-mode)
  :config
  ;; Extra face(s) to ignore
  (push 'org-inline-src-block
    (alist-get 'org-mode jinx-exclude-faces))
  ;; Take over the relevant bindings.
  (after! evil-commands
    (global-set-key [remap ispell-word] #'jinx-correct))
  (after! evil-commands
    (global-set-key [remap evil-next-flyspell-error] #'jinx-next)
    (global-set-key [remap evil-prev-flyspell-error] #'jinx-previous)))
```

6 LSP

6.1 Keybindings

```
;; (map! :after lsp-mode
;;       :map evil-normal-state-map
;;       "K" #'lsp-ui-doc-show)
```

6.2 LSP-Bridge

```
;; (when (package! lsp-bridge
;;       :recipe (:host github
;;                 :repo "manateelazycat/lsp-bridge"
;;                 :branch "master"
;;                 :files ("*.el" "*.py" "acm" "core" "langserver"
;;                        "multiserver" "resources")
;;                 ;; do not perform byte compilation or native compilation for
;;                 lsp-bridge
;;                 :build (:not compile)))
```

```
;; (package! markdown-mode)
;; (package! yasnipet))
```

```
;; (use-package! lsp-bridge
;;   :config
;;   (setq lsp-bridge-enable-log nil)
;;   (global-lsp-bridge-mode))
```

6.3 Digestif

```
;; (after! lsp-mode
;;   (setq lsp-tex-server 'digestif))
```

6.4 LTeX

```
(package! lsp-ltex)
```

```
(use-package! lsp-ltex
  :ensure t
  :defer t
  :init
  (setq lsp-ltex-version "16.0.0")) ; make sure you have set this, see below

;; (use-package! eglot-ltex
;;   :init
;;   (setq eglot-ltex-server-path "/opt/homebrew/"
;;         eglot-ltex-communication-channel 'tcp)) ; 'stdio or 'tcp
```

```
;; (after! eglot
;;   (add-to-list 'eglot-server-programs
;;     `((latex-mode :language-id "latex")
;;       . ,(eglot-alternatives '("texlab")
;;                             ("ltex-ls" "--server-type"
;;                               "TcpSocket" "--port" :autoport)))))) ;
```


7 VLFI

```
(package! vlfi)
```

```
(use-package! vlf-setup  
  :defer-incrementally vlf-tune vlf-base vlf-write vlf-search vlf-occur  
  ↪ vlf-follow vlf-ediff vlf)
```

8 PDF-Tools

8.1 Dark mode

```
;; (add-hook 'pdf-tools-enabled-hook 'pdf-view-midnight-minor-mode)
```

9 Option key Fix

```
(defun switch-left-and-right-option-keys ()  
  "Switch left and right option keys.  
   On some external keyboards the left and right option keys are swapped,  
   this command switches the keys so that they work as expected."  
  (interactive)  
  (let ((current-left mac-option-modifier)  
        (current-right mac-right-option-modifier))  
    (setq mac-option-modifier current-right  
          mac-right-option-modifier current-left)))
```

10 Centaur

```
;; (after! centaur-tabs  
;;   (centaur-tabs-mode -1)  
;;   (setq centaur-tabs-height 36  
;;         centaur-tabs-set-icons t  
;;         centaur-tabs-modified-marker "o"  
;;         centaur-tabs-close-button "×"  
;;         centaur-tabs-set-bar 'above  
;;         centaur-tabs-gray-out-icons 'buffer)  
;;   (centaur-tabs-change-fonts "P22 Underground Book" 160))
```

```
;; (setq x-underline-at-descent-line t)
```

11 RSS

```
;; (add-hook! 'elfeed-search-mode-hook #'elfeed-update) ;  
;; (after! elfeed  
;;   (setq elfeed-goodies/entry-pane-position 'bottom)  
;;   (setq rmh-elfeed-org-files '("~/org/elfeed.org")))
```

11.1 Visual

```
;; (after! elfeed  
;;   (setq elfeed-search-filter "@1-week-ago +unread"  
;;         elfeed-search-print-entry-function '+rss/elfeed-search-print-entry  
;;         elfeed-search-title-min-width 80  
;;         elfeed-show-entry-switch #'pop-to-buffer  
;;         elfeed-show-entry-delete #'elfeed-kill-buffer  
;;         elfeed-show-refresh-function #' +rss/elfeed-show-refresh--better-style  
;;         shr-max-image-proportion 0.6)  
  
;;   (add-hook! 'elfeed-show-mode-hook (hide-mode-line-mode 1))  
;;   (add-hook! 'elfeed-search-update-hook #'hide-mode-line-mode)  
  
;;   (defface elfeed-show-title-face '((t (:weight ultrabold :slant italic  
↩ :height 1.5)))  
;;     "title face in elfeed show buffer"  
;;     :group 'elfeed)  
;;   (defface elfeed-show-author-face `((t (:weight light)))  
;;     "title face in elfeed show buffer"  
;;     :group 'elfeed)  
;;   (set-face-attribute 'elfeed-search-title-face nil  
;;                       :foreground 'nil  
;;                       :weight 'light)  
  
;;   (defadvice! +rss-elfeed-wrap-h-nicer ()  
;;     "Enhances an elfeed entry's readability by wrapping it to a width of  
;; `fill-column' and centering it with `visual-fill-column-mode'."  
;;     :override #' +rss-elfeed-wrap-h  
;;     (setq-local truncate-lines nil
```

```

;;          shr-width 140
;;          visual-fill-column-center-text t
;;          default-text-properties '(line-height 1.2))
;;      (let ((inhibit-read-only t)
;;            (inhibit-modification-hooks t))
;;          (setq-local shr-current-font '(:family "Lato" :height 1.2))
;;          (set-buffer-modified-p nil)))

;;      (defun +rss/elfeed-search-print-entry (entry)
;;        "Print ENTRY to the buffer."
;;        (let* ((elfeed-goodies/tag-column-width 40)
;;               (elfeed-goodies/feed-source-column-width 30)
;;               (title (or (elfeed-meta entry :title) (elfeed-entry-title entry)
;;                           ↪ ""))
;;               (title-faces (elfeed-search--faces (elfeed-entry-tags entry)))
;;               (feed (elfeed-entry-feed entry))
;;               (feed-title
;;                (when feed
;;                  (or (elfeed-meta feed :title) (elfeed-feed-title feed))))
;;               (tags (mapcar #'symbol-name (elfeed-entry-tags entry)))
;;               (tags-str (concat (mapconcat 'identity tags ",")))
;;               (title-width (- (window-width)
;;                               ↪ elfeed-goodies/feed-source-column-width
;;                               elfeed-goodies/tag-column-width 4))
;;               (tag-column (elfeed-format-column
;;                            tags-str (elfeed-clamp (length tags-str)
;;                                                    ↪ elfeed-goodies/tag-column-width)
;;                            :left))
;;               (feed-column (elfeed-format-column
;;                              feed-title (elfeed-clamp
;;                                           ↪ elfeed-goodies/feed-source-column-width
;;                                           elfeed-goodies/feed-source-column-width)
;;                              :left)))

```

```

;;      (insert (propertize feed-column 'face 'elfeed-search-feed-face) " ")
;;      (insert (propertize tag-column 'face 'elfeed-search-tag-face) " ")
;;      (insert (propertize title 'face title-faces 'kbd-help title))
;;      (setq-local line-spacing 0.2)))

;; (defun +rss/elfeed-show-refresh--better-style ()
;;   "Update the buffer to match the selected entry, using a mail-style."
;;   (interactive)
;;   (let* ((inhibit-read-only t)
;;          (title (elfeed-entry-title elfeed-show-entry))
;;          (date (seconds-to-time (elfeed-entry-date elfeed-show-entry)))
;;          (author (elfeed-meta elfeed-show-entry :author))
;;          (link (elfeed-entry-link elfeed-show-entry))
;;          (tags (elfeed-entry-tags elfeed-show-entry))
;;          (tagsstr (mapconcat #'symbol-name tags ", "))
;;          (nicedate (format-time-string "%a, %e %b %Y %T %Z" date))
;;          (content (elfeed-deref (elfeed-entry-content elfeed-show-entry)))
;;          (type (elfeed-entry-content-type elfeed-show-entry))
;;          (feed (elfeed-entry-feed elfeed-show-entry))
;;          (feed-title (elfeed-feed-title feed))
;;          (base (and feed (elfeed-compute-base (elfeed-feed-url feed)))))
;;     (erase-buffer)
;;     (insert "\n")
;;     (insert (format "%s\n\n" (propertize title 'face
⇐ 'elfeed-show-title-face)))
;;     (insert (format "%s\t" (propertize feed-title 'face
⇐ 'elfeed-search-feed-face)))
;;     (when (and author elfeed-show-entry-author)
;;       (insert (format "%s\n" (propertize author 'face
⇐ 'elfeed-show-author-face))))
;;     (insert (format "%s\n\n" (propertize nicedate 'face
⇐ 'elfeed-log-date-face)))
;;     (when tags
;;       (insert (format "%s\n"
;;                        (propertize tagsstr 'face 'elfeed-search-tag-face))))
;;     ;; (insert (propertize "Link: " 'face 'message-header-name))
;;     ;; (elfeed-insert-link link link)
;;     ;; (insert "\n")
;;     (cl-loop for enclosure in (elfeed-entry-enclosures elfeed-show-entry)

```

```
;;          do (insert (propertize "Enclosure: " 'face
↪ 'message-header-name))
;;          do (elfeed-insert-link (car enclosure))
;;          do (insert "\n"))
;;      (insert "\n")
;;      (if content
;;      (if (eq type 'html)
;;      (elfeed-insert-html content base)
;;      (insert content))
;;      (insert (propertize "(empty)\n" 'face 'italic)))
;;      (goto-char (point-min))))
;; )
```

12 Email

12.1 mu4e

```
;; add to $DOOMDIR/config.el
(after! mu4e
  (setq sendmail-program (executable-find "msmtp")
        send-mail-function #'smtpmail-send-it
        message-sendmail-f-is-evil t
        message-sendmail-extra-arguments '("--read-envelope-from")
        message-send-mail-function #'message-send-mail-with-sendmail)
  ;; how often to call it in seconds:
  (setq mu4e-sent-messages-behavior 'sent ;; Save sent messages
        mu4e-headers-auto-update t      ; avoid to type `g' to
        ↪ update
        mu4e-compose-signature-auto-include nil ; I don't want a message
        ↪ signature
        mu4e-use-fancy-chars t          ; allow fancy icons for mail
        ↪ threads
        mu4e-context-policy 'pick-first ;; Start with the first context
        mu4e-compose-context-policy 'ask) ;; Always ask which context to use
        ↪ when composing a new mail
  (setq mu4e-update-interval (* 1 60))
  (setq mu4e-attachment-dir "~/Downloads")
  (set-email-account! "gmail")
```

```

      '(mu4e-sent-folder      . "/gmail/[Gmail]/Sent Mail")
      (mu4e-drafts-folder    . "/gmail/[Gmail]/Drafts")
      (mu4e-trash-folder     . "/gmail/[Gmail]/Trash")
      (mu4e-refile-folder    . "/gmail/[Gmail]/All Mail")
      (smtpmail-smtp-user    . "leoaparis@gmail.com")
      (mu4e-compose-signature . "---\nLeo Aparisi de Lannoy"))
    t)
(set-email-account! "U Chicago"
  '(mu4e-sent-folder      . "/UChicago/Sent Mail")
  (mu4e-drafts-folder    . "/UChicago/Drafts")
  (mu4e-trash-folder     . "/UChicago/Trash")
  (mu4e-refile-folder    . "/UChicago/All Mail")
  (smtpmail-smtp-user    .
    ↪ "laparisidelannoy@uchicago.edu")
  (mu4e-compose-signature . "---\nLeo Aparisi de Lannoy"))
  t)
(setq +mu4e-gmail-accounts '(("leoaparis@gmail.com" . "/gmail/[Gmail]")))
(setq mu4e-compose-dont-reply-to-self t)
;; Add a unified inbox shortcut
(add-to-list
 'mu4e-bookmarks
 '(:name "Unified inbox" :query "maildir:/*inbox/" :key ?i) t)
)

```

12.2 Notification

```

;;(mu4e-alert-set-default-style 'notifier)
;;(add-hook 'after-init-hook #'mu4e-alert-enable-notifications)

```

13 Latex

```

(setq +latex-viewers '(pdf-tools))
(defun compile-save()
  "Test of save hook"
  (when (eq major-mode 'latex-mode)
    (+latex/compile)))
(add-hook 'after-save-hook #'compile-save)
(setq TeX-save-query nil
  TeX-show-compilation nil

```

```
TeX-command-extra-options "-shell-escape")
(after! latex
  (add-to-list 'TeX-command-list '("XeLaTeX" "%`xelatex%(mode)%" %t" TeX-run-TeX
    ↪ nil t)))
```

14 Flycheck

```
;; (setq flycheck-eglot-exclusive nil)
(map! :map evil-normal-state-map
      "SPC c b" #'consult-flycheck)
```

15 TRAMP

```
(after! tramp
  (setenv "SHELL" "/bin/bash")
  (setq tramp-shell-prompt-pattern "\\(?:~|\\n|\\x0d|\\) [^]#>\\n]*#?[]#> ]
    ↪ *\\(\\e\\[[0-9;]*[a-zA-Z] *\\)*)" ) ;; default +
  (setq vc-ignore-dir-regexp
        (format "\\(%s\\)\\|\\(%s\\)"
                  vc-ignore-dir-regexp
                  tramp-file-name-regexp))
```