

@kawaz

投稿日 2014年06月17日 更新日 2021年10月22日

# Mac新調したのでアプリや設定の個人的な棚卸し

Mac

- 「移行ツール」を使って旧Macから新Macにまるまる引っ越しした場合にやることは→こちらを確認。

## いろいろ入れるメモまとめ

とりあえず新しい環境でターミナル開いて↓これを上から順番に実行していけば大体整うんじゃないかな。

```
# Homebrew 本体
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)" brew doctor

# CLI系
brew install go # Go
brew install node # Node
brew install deno # 最近はNodeより使うかも
brew install php composer # PHP
brew install python # Python
brew install luajit # lua
brew install shellcheck # bashのlint的なやつ
brew install bash binutils coreutils findutils bash-completion # 基本コマンドGNU
brew install tmux reattach-to-user-namespace # ターミナル内のウィンドウ分割とか (20210506 wezterm使い始めてからtmux使わなくなってきた)
brew install starship # shellのプロンプトをパワフルにする。
```

```

brew install vim # Vimエディタ
#brew install homebrew/dupes/rsync # HFS+のNFD対応の文字コード変換付きrsync。macOS10.13以降のAPFSだとNFDじゃなくWin/Linuxと同様にNFCになったのでもう
brew install pstree ipcalc # 昔からの懐かしいツール
brew install peco jq # JSONのクエリツール
brew install itchyny/tap/gojq # goで再実装されたjqの別実装
brew install htmlq # HTMLをCSSセレクタでクエリするツール
brew install git git-extras git-lfs ghpub # git関係
brew install mercurial subversion # 時々使うCVS系
brew install imagemagick ffmpeg # 画像とか動画とか
brew install rar p7zip # アーカイブ系
brew install memcached redis # KVS系
brew install ansible # 構成ツール
brew install awscli # AWS
brew install asciinema # ターミナルキャプチャ共有サービス
brew install ctags dict cmigemo # 主にvim連携系で使うコマンドたち
brew install daemontools djbdns # djbdns系ツール
brew install poppler # pdfからテキストを抽出するコマンドpdftotextが入ってる
brew install pdfcpu # pdfに印影を埋め込んだりするのに使える
brew install direnv # .envrc でファイルを作っておくとディレクトリ毎に環境変数の自動設定が出来るようにするツール
brew install iproute2mac # Linuxのipコマンド風ラッパー、ipコマンドに慣れてるので入れておくと楽
brew install cuelang/tap/cue # 新しいデータ検証言語、データの型定義もできるし値を記述することも出来る
brew install act #GithubActionsをローカルで実行するコマンド
brew install node-sass # sassコンパイラ、VSCodeでsass書くときとかにランタイムとして要求される事があるので
brew install bat # ターミナルでシンタックスハイライト出来る cat みたいなやつ
brew install sk # pecoとかfzfみたいなやつ。ファジー検索で候補を絞り込むツール。

# Java関係
## jdkのパッケージは選択肢がいくつかあるが、特に思想信条が無ければ adoptopenjdkを入れとけば良い
## 色々入れたJDKのどれかをJAVA_HOMEとかに指定したい場合は `find /Library/Java/JavaVirtualMachines/*/Contents/Home -maxdepth 0` の出力から選択すれば良い
brew install --cask adoptopenjdk # OpenJDKのコミュニティ製バイナリ、パッケージ名に数字がついてないやつは最新が入る(20201014時点では15)
brew install --cask adoptopenjdk8 # java8が要求されるケースがあるので追加で入れとく...(Elastic系がOpenJDK14とか入っててもJava 1.8.0系を要求してくる@20200606)
#brew install --cask java # OpenJDKのソースからビルドするやつ、自分でビルドする意味は殆どないのでadoptopenjdkを使うのが無難
#brew install --cask oracle-jdk # Oracle JDK
#brew install --cask corretto # AWS Java
brew install elasticsearch kibana logstash # Elastic系

# GUI系アプリ
### これを先に設定しておくと面倒の一部が回避できる。cask系入れる前にセットしつつ普段のupgrade運用の時のことも考えるとbashrcとかに設定していくのが良い
export HOMEBREW_CASK_OPTS='--no-quarantine' # 入れたアプリが「このファイルはXXからダウンロードされました」みたいな確認ダイアログが出て困ったたりする奴(Gatekeeper)

```

```
brew install --cask 1password # パスワードマネージャ
brew install --cask 1password-cli # opコマンド
brew install --cask atok # 日本語IMEの老舗、2021年頃Google日本語入力がうまく動かなくなった際に乗り換えた。install後に一度再起動しないと使えない点に注意。
brew install --cask appcleaner # アプリ掃除アプリ（アプリケーションフォルダからアプリを削除すると関連ファイルも消すかどうか聞いてくれるようになる）
# brew install --cask alacritty # 高速ターミナル（高速がウリだけどそれ以外の機能が物寂しいのでtmuxと併用することになるがtmuxが遅いせいで高速性にも難が...。あと日本語
brew tap wez/wezterm; brew install --cask wezterm-nightly --no-quarantine # 高速ターミナル、タブや画面分割(multiplex)など機能も豊富。バグフィックスとかがr
brew install --cask kitty # 高速ターミナル、タブや画面分割(multiplex)など機能も豊富。202107頃から利用中
brew install --cask alfred # opt+spaceで簡単起動アプリ
brew install --cask bettertouchtool # ジェスチャーアプリ、超重要
brew install alt-tab # アプリ単位じゃなくウィンドウ単位での切り替えが出来るようにする、デフォルトは opt+TABだがcmd+TABに上書きして使ってる。ドックにしまったウィンド
#brew install --cask cooViewer # 漫画ビューア、macOS 10.15 Catarina 以降使えなくなってしまったので削除... 20200202で64bit版として復活！でもbrewではまだ入れられ
# brew install --cask yacreader # 漫画ビューア、cooViewrの使い勝手に及ばないがとりあえず使ってる... cooViewer復活したしもうイラネw
brew install --cask docker # 最近はいろいろ使う。※cask無しにも同名のdockerがあるがバージョン古いので注意、caskの方を入れる
# brew install --cask kitematic # DockerのGUI？昔のDockerについてたけど今はいらねいぱい
brew install --cask dropbox # 便利ストレージサービス
brew install --cask visual-studio-code # VSCode、CLIからだとcodeで起動できる、最近流行りのエディタ／開発環境
brew install --cask eclipse-java # 開発用、最近VSCodeに乗り換えて不便しなくなってきたのそろそろ不要かも。JavaのRemoteDebugはeclipseの方がキビキビ動くからまだ時
#brew install --cask google-japanese-ime # Google日本語入力、管理者パスワードが求められる
brew install --cask google-chrome # Chromeブラウザ（これとは別にbetaやdevを本家から落として Google Chrome Dev.app 等にリネームしてアプリケーションフォルダにノ
brew install --cask firefox # 確認用に使う
brew install --cask vlc # 動画再生
brew install --cask skype # たまに仕事で使う
brew install --cask microsoft-office # Excel,Word,PowerPoint
brew install --cask parallels # Windows10を動かすのに使ってる
brew install --cask discord # チャットアプリ一部コミュで使ってる
brew install --cask ngrok # ローカルポートを外部に公開できるサービス（CLIツールだが何故かCaskで提供されてるが気にしない
#brew install --cask nox-app-player # Androidエミュレータ
#brew install --cask bluestacks # Androidエミュレータ
brew install --cask charles # Proxyツール、通信内用の盗聴や改ざん等何でもござれの神ツール。SSLもMITMできる。
brew install --cask steam # ゲームプラットフォーム
#brew install --cask hammerspoon # Macの自動操作やキーバインディングのカスタマイズなど、設定記述言語はLua。入れたもののいじる時間が取れず結局使ってないのでコメント
brew install --cask commander-one # FTP/SFTP/S3/GoogleDrive/OneDrive/WebDAV/Dropbox/Boxクライアント
brew install --cask thunderbird # MUA=メールー、メール環境の試験用に使う
#brew install --cask google-drive-file-stream # GSuiteアカウントのDriveをPCにマウントするやつ (deprecated: google-drive にリネームされた)
brew install --cask google-drive # GSuiteアカウントのDriveをPCにマウントするやつ
brew install --cask messenger # Facebook Messenger

brew install --cask xquartz # X11+バー
```

```

brew install --cask xquartz # ハイドゲート
brew install --cask karabiner-elements # キーボードカスタマイズ
brew install --cask raspberry-pi-imager # ラズパイのOSイメージインストーラ（昔はN00BSでツールが定番だったが2020移行はこっち推奨らしい。ラズパイ使わない人は不要）
brew install --cask makemkv # DVD/Blu-rayから mkv ファイルを作成するリッピングツール。ライセンス購入はメニューのHelp>Purchase/Registerから行う
brew install --cask zoom # リモートミーティング用
brew install --cask google-cloud-sdk #gcloudコマンドなど

# Finderのクイックルックプラグイン (Space押すと出るやつ)
brew install --cask qlstephen # 拡張子がないファイルもプレビュー
brew install --cask qlmarkdown # Markdownビューア
brew install --cask quicklook-json # JSONビューア
brew install --cask quicklook-csv # CSVをテーブル表示でプレビュー
brew install --cask provisionql # ipaファイルをプレビュー
#brew install --cask betterzipql # zipアーカイブ (無くなった)
brew install --cask qlcolorcode # htmlとかいろんなソースのハイライト

# AppStore系
brew install mas # AppStore系アプリのCLI管理ツール、リストは mas list で取れる
mas install 497799835 # XCode
mas install 425424353 # The Unarchiver
mas install 803453959 # Slack
mas install 539883307 # LINE
mas install 405399194 # Kindle
mas install 409201541 # Pages
mas install 409183694 # Keynote
mas install 409203825 # Numbers

# フォント (主にコンソール用で、見易さ重視&Powerline対応なやつ)
#curl -sL http://bit.ly/install-Ricty | bash # Rictyフォント → 毎回ビルド面倒なうえにbrewの引数がコロコロ変わるので白源に乗り換えた
#curl -sL http://bit.ly/install-HackGen | bash # 白源フォント、Rictyインスペクトな後発フォント。Powerline対応でバイナリ配布もあり導入が楽で良い
brew install --cask homebrew/cask-fonts/font-hackgen # 白源フォント、Rictyインスペクトな後発フォント。Powerline対応でバイナリ配布もあり導入が楽で良い。v1.3.0]
brew install --cask homebrew/cask-fonts/font-hackgen-nerd # HackGen フォントに Nerd を足したやつ
# sudoでTouchIDが使えるようにする ( https://qiita.com/kawaz/items/0593163c1c5538a34f6f )
curl -sL https://gist.github.com/kawaz/d95fb3b547351e01f0f3f99783180b9f/raw/install-pam_tid-and-pam_reattach.sh | bash

# ログインシェルを変更する
new_shell=/usr/local/bin/bash
if type "$new_shell"; then
  grep -q "^\$new_shell" /etc/shells || sudo tee -a /etc/shells <<< "$new_shell"
  chsh -s "$new_shell"

```

```
chenh - 3 $new_shrc
fi

# 自分用環境設定
git clone git@github.com:kawaz/dotfiles.git ~/.dotfiles
# Macの初期状態だと ~/.bashrc とか ~/.bash_profile が無いのでなければ作る
grep -Eq '^#[^#]*/.bashrc' ~/.bash_profile 2>/dev/null || echo ". ~/.bashrc" >> ~/.bash_profile
grep -q dotfiles ~/.bashrc 2>/dev/null || echo ". ~/.dotfiles/config/bash/bashrc" >> ~/.bashrc
# ~/.config や ~/.local を $XDG_CONFIG_HOME とかに置き換える
ln -s ~/.dotfiles/config/ ~/.config
ln -s ~/.dotfiles/local/ ~/.local
# カーソル移動を加速する (Preference>キーボードで設定できる限界の突破)
defaults write -g InitialKeyRepeat -int 12 # normal minimum is 15 (225 ms)
defaults write -g KeyRepeat -int 1 # normal minimum is 2 (30 ms)

# テキスト系コンポーネントでのキーカスタマイズ (Home/End/PageUp/PageDownなどの挙動問題)
curl -sL https://bit.ly/kawaz-install-DefaultKeyBinding | bash

# sshd (共有>リモートアクセス) のパスワード認証の無効化
if ! grep -q "^PasswordAuthentication no$" /etc/ssh/sshd_config; then sudo perl -i -pe's/^s*PasswordAuthentication\s.*/#$/' /etc/ssh/sshd_config
if ! grep -q "^ChallengeResponseAuthentication no$" /etc/ssh/sshd_config; then sudo perl -i -pe's/^s*ChallengeResponseAuthentication\s.*/#$/'

# スクリーンショットの保存場所の変更 (デフォルトは多分デスクトップ)
defaults write com.apple.screencapture location '~/Documents/'

# Finderタイトルにフルパスを表示する
defaults write com.apple.finder _FXShowPosixPathInTitle -boolean true
killall Finder # Finderを再起動で反映する (Finderはkillしっぱなしでも勝手に復活される)
```

## アプリ

入れた順でメモメモ

## Google Chrome

メインブラウザ。拡張やブックマークは同期から全部引っ張ってきてくれるので非常に楽。  
あと以下の設定をしている。

- Chromeの検索バーを強化する設定

## Google日本語入力

IME。

ステータスバーのアイコンからキーボード環境設定を開いて、ことえりは削除する。ついでにカタカナとか全角英数とかの入力ソースも使わないので削除する。

- Google 日本語入力 環境設定>一般タブ>¥キーで入力する文字> バックスラッシュ(\)を選択 **超重要**

## Firefox

2つめのブラウザ。テスト用とかで使う。

## Alfred

ランチャとして優秀。 Alt+Space から何でも検索&起動。

ドキュメントの検索や計算機としてもよく使う。

- Preferences > Advanced > Fource Keyboard → Alphanumeric ... Alfredを起動した時にIMEを強制OFFにする設定

- Preferences > Features > Dictionary > Define a word → d スペース で辞書検索出来るようにする

- 追加Workflow
  - Kill Process 最近Dockがよく固まるので kill Dock で簡単再起動が出来るようにした。

## AppCleaner

---

アプリを関連ファイル含めて削除するツール。

## The Unarchiver

---

AppStoreから

アーカイブファイルの解凍ソフト。

## 1Password

---

ずっとサーバ上においていたテキストファイルでパスワードメモしてたんだけど、昔からよく紹介されてた1Passwordを入れてみた。iPhone版もあって両方買ってる。そして最近はクラウド保存をファミリータイプで課金して使ってる。これがあるとサイト毎にパスワードを64文字のランダム文字列で設定するとかいう運用も超簡単で悩みが一つ減る。

AppStoreにもあるが Chrome 拡張連携がうまく動かなかったのでサイトからDLしたパッケージで入れる。

- 設定>一般>キーボードショートカット>現在のウェブページにログインを入力する→空にする
  - デフォルト割り当ての [⌘] は、Chromeの進むと被っててChrome側でよく使うのでとりあえず外してる。別のショートカットにするかは今後考える。

## LimeChat

---

IRCクライアント。特別使いやすいとも思わないけどWindows時代からずっと使ってる。

## CheatSheet

---

Cmdキー押しっぱでそのアプリのキーボードショートカットの一覧が表示できるという便利ソフト。

- 設定：起動後適当なアプリで⌘押しっぱでCheatSheetを表示する→⌘押したまま右下の⚙(ギアアイコン)をクリック→Launch at loginをチェック

## XCode

---

AppStore or β版はAppleDevelopperCenterから。

アプリ開発に使う。

## Eclipse

---

Java書くのに使う。

## Dash

---

各種ドキュメント検索に超便利なやつ。

## BetterTouchTool

---

キーボードショートカットやジェスチャーを設定できるやつ。

設定変更と登録してるジェスチャは↓こんな感じ。タブ操作が主。

- Settings>Launch BettetTouchTool on startup→チェックON
- Action Settings>Window Snapping>Window Snapping Enabled→OFF
  - この辺りのカスタマイズはBetterSnapToolを使うのでBTT側の関連機能は無効にしておく
- 高度な設定>一般>Install Process Priority Helper Tool
  - BTTのプライオリティが低いとフック処理に遅延や操作漏れが起きることがあるので有効にしておくと良いらしい。
- Gestures
  - Global (Trackpads)

Trackpad Gesture	Assigned Action / Shortcut	Notes	Modifiers
Four Finger Swipe Right	⌘F1	同一アプリの前のウィンドウ	
Four Finger Swipe Left	⌘⇧F1	同一アプリの次のウィンドウ	

- ブラウザとか同一アプリの複数ウィンドウが開くタイプのアプリで、ウィンドウ切り替えを行うジェスチャを追加してます。
- Finder (Trackpads / Keyboard)

Trackpad Gesture	Assigned Action / Shortcut	Notes	Modifiers
Three Finger Swipe Right	△↖→	左のタブへ移動	
Three Finger Swipe Left	↖→△	右のタブへ移動	
Three Finger Swipe Down	⌘W	タブを閉じる	
Rotate Left	⌘[	戻る	
Rotate Right	⌘]	進む	

- タブ移動系はよく使うので重要。Finder/Chrome/etcで共通操作になるようBTTで吸収している。

Shortcut	Assigned Action	Enabled
⌘P	No Action	<input checked="" type="checkbox"/>
⇧⌘N	⌘⇧N	<input checked="" type="checkbox"/>
⌘⌥N	⌘⌥N	<input checked="" type="checkbox"/>

- ⌘P の即時印刷機能とか事故にしかならないので無効化する。
- 新規フォルダのショートカットはよく使うけどキーの組み合わせを忘れるので適当に押しても行けるようにしてます。
- Chrome (Trackpads / Keyboard)

Trackpad Gesture	Assigned Action / Shortcut	Notes	Modifiers
Rotate Left	⌘[	戻る	
Rotate Right	⌘]	進む	
Three Finger Swipe Up	⌘⇧T	最後に閉じたタブを開く	
Three Finger Swipe Down	⌘W	タブを閉じる	
Three Finger Swipe Left	⌘⇧]	右のタブに移動	
Three Finger Swipe Right	⌘⇧[	左のタブに移動	
Four Finger Tap	⌘⇧B	ブックマークバーの出し入れ	
Three Finger Double-Tap	⌘R	リロード	

- タブ移動系はよく使うので重要。Finder/Chrome/etcで共通操作になるようBTTで吸収している。
- 閉じたタブの復活や、ブックマークバーの出し入れとか、よく使う操作もジェスチャ化する。

Shortcut	Assigned Action	Enabled
⌘⇧J	⌘⌥J	<input checked="" type="checkbox"/>
⌘J	⌘⇧J	<input checked="" type="checkbox"/>
^J	⌘⇧J	<input checked="" type="checkbox"/>
⌘^N	⌘⌥N	<input checked="" type="checkbox"/>
⇧⌘N	⌘⌥N	<input checked="" type="checkbox"/>

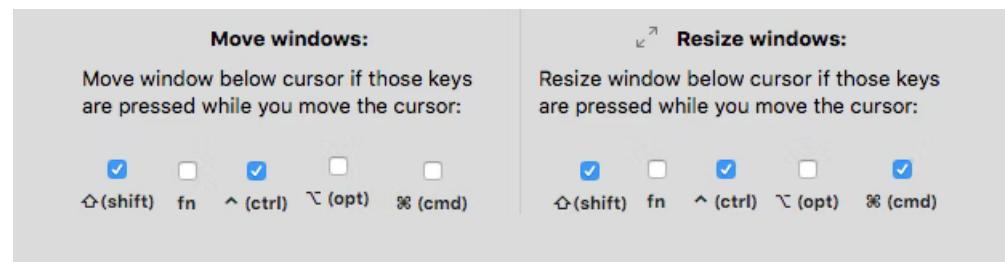
- シークレットウィンドウやJavascriptコンソールはよく使うけどキーの組み合わせを忘れるので適当に押しても行けるようにしてます。
- iTerm (Trackpads)

Trackpad Gesture	Assigned Action / Shortcut	Notes	Modifiers
Three Finger Swipe Left	⌘△]	右のタブに移動	
Three Finger Swipe Right	⌘△[	左のタブに移動	

- タブ移動系は共通操作になるようBTTで吸収しているが、iTermの場合閉じる操作は不要なのでタブ移動だけに留める。
  - VLC (Keyboard)

Shortcut	Assigned Action	Enabled
←	⌘←	✓
→	⌘→	✓
↑	⌘↑	✓
↓	⌘↓	✓

- 片手で、戻る/進む、ボリュームUP/DOWN、が出来るようにする。
- Advanced Settings
  - Window Moving & Resizing Settings



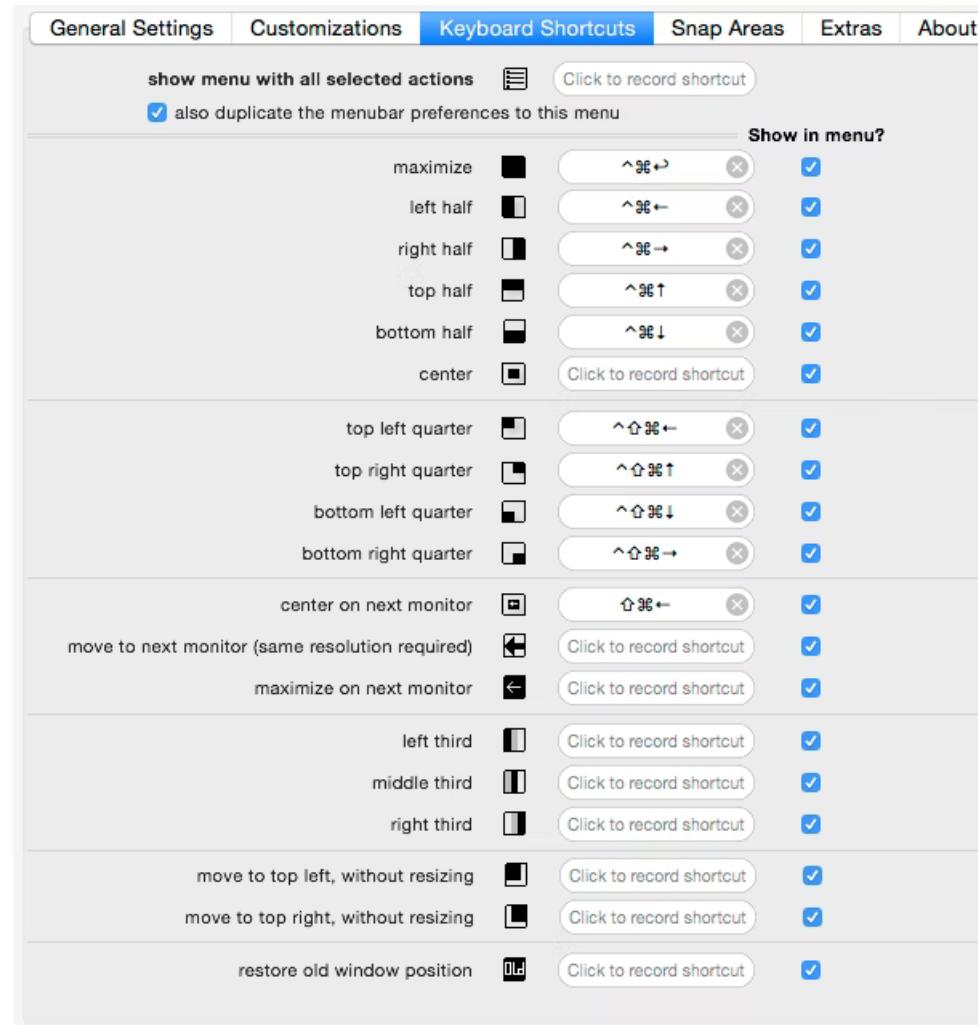
- Move windows に ⌈ (shift) と ⌉ (ctrl) を割り当てる(ウィンドウの左上を引っ張るイメージなので縦配置の組み合わせ)
- Resize windows に ⌈ (shift) と ⌉ (ctrl) と ⌘ (cmd) を割り当てる (右下を引っ張るイメージでMoveの組み合わせに右下を足す)

## BetterSnapTool

ウィンドウを画面端にフィットさせたり、キーボードでウィンドウ移動したりとかするやつ。

BetterTouchToolに同種の機能があるけどこっちのほうが細かいカスタマイズが効くのでBTT側のスナップ機能は無効にしてこっち使ってる。

- General Settings>Start BetterSnapTool everytime your Mac starts up→チェックON
- Customizations>Appeareance of the preview overlay>Animated→チェックOFF
- Customizations>Appeareance of the preview overlay>Use rounded corners→チェックOFF
- Keyboard Shortcuts ↓デフォルトは全部空だがこんな感じに設定してる



## Dropbox

言わずと知れたクラウドストレージ

## Skype

---

ビデオチャット。仕事で使う。

## Docker

---

開発で使う。

## cooViewer

---

漫画ビューア

## IntelliJ IDEA

---

Android開発に使う。

## iTerm2

---

ターミナルアプリ。

以下の設定変更を行う。特にbell設定がうるさいので速攻消す。あと閉じるときの確認が邪魔なので出ないようにする。

- General>Quit when all windows are closed→チェックON
- General>Confirm closing multiple sessions→チェックOFF
- General>Confirm "Quit iTerm2 (⌘Q)" command→チェックOFF
- General>Characters considered part of word for selection→@と:を足す

- Profile>Terminal>Scrollbar Lines→10000に増やす
- Profile>Terminal>Report Terminal Type→xterm-256colorに変更
- Profile>Terminal>Silence bell→チェックON

## Parallels Desktop

Mac上でWindowsアプリを使うのに使う。

Windowsデスクトップ非表示でアプリウィンドウだけを表示するCoherenceモードが便利。

## VLC

動画再生。

## ステイツキーズ

デスクトップ付箋アプリ。

デフォアプリなんだけど何故かLaunchpadには表示されないのアプリケーションフォルダから探して起動する。一度起動後はDockから「ログイン時に開く」をチェックしておく。

## sudoでTouchIDが使えるようにする

/etc/pam.d/sudo の最初に以下を1行追加する。

```
auth      sufficient      pam_tid.so
```

## ログインシェルを変更する

例えば `/usr/local/bin/bash` にする場合、`/etc/shells` に載ってないので chsh で変えようとする怒られる。のでまず追加してから chsh する。

```
grep -q ^/usr/local/bin/bash /etc/shells || sudo tee -a /etc/shells <<< /usr/local/bin/bash
chsh -s /usr/local/bin/bash
```

## Homebrew

主にCUIコマンド類を色々探して入れるツールと認識してる。

- brew本体を入れる

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)" brew doctor
```

- ls, rm, find, xargs とかの基本的なコマンドをGNU版にする。

```
brew install xz binutils coreutils findutils
( echo '# coreutils'
  echo 'export PATH="$(brew --prefix coreutils)/libexec/gnubin:$PATH"'
  echo '# findutils'
  echo 'alias find=gfind'
  echo 'alias xargs= gxargs'
) >> ~/.bashrc
```

- bash-completionを入れる

```
brew install bash-completion
# bash-completion
( echo '# bash-completion'
  echo 'if [ -f $(brew --prefix)/share/bash-completion/bash_completion ]; then'
  echo '  . $(brew --prefix)/share/bash-completion/bash_completion'
  echo 'fi'
) >> ~/.bashrc
```

- tmuxを入れる

```
brew install tmux reattach-to-user-namespace
```

- Ricty フォント (Powerlineパッチ版) を入れる

```
(  
set -e  
brew install sanemat/font/ricty --with-powerline  
/bin/cp /usr/local/opt/ricty/share/fonts/Ricty*Powerline.ttf ~/Library/Fonts  
fc-cache -vf  
brew uninstall ricty # フォントさえ手に入れば不要  
)
```

- 白源フォント (Rictyの後発フォント。ファミリ名は HackGen Console for Powerline。バイナリ配布で導入簡単でPowerlineも対応してる)

```
curl -sL https://gist.github.com/kawaz/655a491963366bf452e8dfdf62d448f2/raw/install-fonts-HackGen.sh | bash
```

## smcFanControl

本体温度とファンの回転数をステータスバーに表示するアプリ。強制的にファンの速度を上げて冷却を頑張らせることも可能。夏になると入れて秋ごろ削除する。

## 設定関連

---

デフォルトから変更しているところを列挙

### システム環境設定>一般

---

- 書類を閉じるときに変更内容を保持するかどうかを確認→チェックON

### システム環境設定>Dock

---

- Dockを自動的に隠す／表示→チェックON

### システム環境設定>Mission Control

---

- Dashboardを操作スペースとして表示→チェックOFF
  - ダッシュボード使わないから要らない！
- キーボードとマウスのショートカット→全部無効にする
  - ジェスチャー使うのでキーボードショートカットは要らない！

### システム環境設定>Spotlight

---

- 検索結果>Spotlightメニューのキーボードショートカット→チェックOFF
- 検索結果>Spotlightウィンドウのキーボードショートカット→チェックOFF
  - ^スペース とか超有用なキーボードをSpotlightなんかに盗られると迷惑なので！
- 検索結果>カテゴリリストのチェックボックス→全てOFF
  - 更にターミナルで `sudo mdutil -a -i off` を実行。
  - 要は Spotlight 検索とか使わないので普段からリソース食って邪魔だから止める！

## システム環境設定>ディスプレイ

---

- 解像度を最大に変更。綺麗さより広さ！

## システム環境設定>省エネルギー

---

- 電源アダプタ>ディスプレイが切のときはコンピュータを自動でスリープさせない→チェックON
  - 放置タスクが動いてることも多いし電源ささってるときは勝手にスリープされたら困るので！

## システム環境設定>キーボード

---

- キーボード>キーのリピート→最速から2番目にする
- キーボード>リピート入力認識までの時間→最短にする
- キーボード>F1、F2などの全てのキーを標準のファンクションキーとしてしよう→チェックON
  - 使用頻度からして輝度や音量変更の方が使わないので
- キーボード>修飾キー...>CapsLock→`^ Control` に変更
  - CapsLockは邪魔でしかないので！
- ショートカット>アクセシビリティ>VoiceOverの入／切→チェックOFF
  - ⌘F5とかIDFで使ったりするのでデフォのショートカットは邪魔なので切る

- ショートカット>フルキーボードアクセス→全てのコントロール
  - システムダイアログのボタンフォーカスをTabで移動できるようにするなど
- 入力ソース>書類ごとに入力ソースを自動的に切り替える→チェックON
  - Chromeで日本語入力した後とかにiTermに戻ってきたときに勝手に日本語入力になってる問題から開放される

## システム環境設定>トラックパッド

---

- ポイントとクリック>タップでクリック→チェックON
- その他のジェスチャ>ページ間をスワイプ→チェックOFF
  - この設定の動作は戻る進むの感覚が逆なので...、ただ戻る進むはあると便利なのでBetterTouchToolで好みの設定を作る
- その他のジェスチャ>アプリケーションExpose(3本指で下にスワイプ)→チェックOFF
- その他のジェスチャ>Lynchpad(親指と3本指でピンチ)→チェックOFF
  - Alfred導入したら不要になったので

## システム環境設定>サウンド

---

- サウンドエフェクト>ユーザインターフェースのサウンドエフェクトを再生→チェックOFF
  - Finderでファイル操作したりする度に音がなるのがうざすぎるので

## システム環境設定>ユーザとグループ

---

- 顔写真を設定する

## システム環境設定>日付と時刻

---

- 時計>秒を表示→チェックON
- 時計>曜日を表示→チェックON
- 時計>日付を表示→チェックON

## Finder

- 環境設定>一般>新規Finderウィンドウで次を表示→ダウンロードに変更
  - デフォルトのマイファイルは邪魔なので...
- 環境設定>詳細>すべてのファイル名拡張子を表示→チェックON
- メニュー>表示>パスバーを表示→チェックON
  - 上位パスがわからないと不便なので
- ターミナルで以下を実行（タブバーにフルパスが表示されるようにする）
  - `defaults write com.apple.finder _FXShowPosixPathInTitle -boolean true; killall Finder`
  - パスバーもいいけど上にもフルパス表示したいので

## テキスト系コンポーネントでのキーカスタマイズ

Macだと PageUp/Down や Home/End の挙動が、表示エリアだけスクロールしてカーソルがついてこないので元の場所に戻ってしまってムキーってなることが多い。何の嫌がらせかと思うがその挙動を以下ファイルで変更することができるWindows風にしておくことにする。あとバックスラッシュを入力したいのに ¥ になってしまう問題とかもIMEの種類によらず解決できたりする。

デフォルトだとファイルもディレクトリも無いのでどちらも新規にする。このファイルを更新してから各アプリを再起動するとキーボ動作が変わっている筈だ。

```
~/Library/KeyBindings/DefaultKeyBinding.dict
```

```
// https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/EventOverview/TextDefaultsBindings/TextDefaultsBindings.html
// ^ for Control
// $ for Shift
```

```
// ~ for Option
// @ for Command
// # for numeric keypad

{
    // backslash
    "\$" = ("insertText:", "\\\");

    // home/end
    "\UF729" = "moveToBeginningOfLine:";
    "\UF72B" = "moveToEndOfLine:";
    "$\UF729" = "moveToBeginningOfLineAndModifySelection:";
    "$\UF72B" = "moveToEndOfLineAndModifySelection:";
    "\^UF729" = "moveToBeginningOfDocument:";
    "\^UF72B" = "moveToEndOfDocument:";
    "$\^UF72B" = "moveToEndOfDocumentAndModifySelection:";
    "$\^UF729" = "moveToBeginningOfDocumentAndModifySelection:");

    // page up/down
    "\UF72C" = "pageUp:";
    "\UF72D" = "pageDown:";
}
```

## シェル環境構築

### 自分設定

自分設定をgithubに纏めてるからそれ取ってきてポン。

```
git clone https://github.com/kawaz/dotfiles.git ~/.dotfiles
~/.dotfiles/bin/setup.sh
```

```
#上に書いたHomebrew系のも自分用なら以下でOK  
bash -x ~/.dotfiles/bin/build-tmux.sh  
bash -x ~/.dotfiles/bin/build-mac-coreutils.sh  
bash -x ~/.dotfiles/bin/build-bash-completion.sh
```

## 削除したもの

---

デフォアプリや昔気まぐれで買ってしまったアプリで、使わないどころか間違って起動すると終了するのが面倒くさかったりで邪魔にしかなったことないもの一覧。

アプリを消すときは `Applications/` 以外に `Library/` とか `Application Support/` とか色々なとこに設定やデータファイルがあるときがあるので、手動削除する前に `TrashMe` でチェックして消した上で `TrashMe` で消せなかったものを主導削除すると良い。

## iPhoto

---

`iPhone`を`iTunes`に繋げたいのに何故かこいつに処理を奪われて指すたびに起動してきたりと何かと邪魔なので削除。

1.5GBくらい空く。

## iMovie

---

一度も使ったこと無いうえに結構容量も食ってるので要らない。

2.9GBくらい空く、...と思ったらOSXに必要とされているため削除できないって言われたンゴ。

## Numbers

---

エクセルで開きたいのに間違って起動しちゃった上にiCloud保存とかのダイアログが⌘Qですぐに殺せなくて邪魔にしかなったこと無いので削除。

350MBくらい空く。

## Pages

---

使ったこと無いので消す。

470MBくらい空く。

## Photo Booth

---

使ったこと無いし、なんだか目障りなので消す。

20MB空く、...と思ったらOSXに必要とされているため消せない...。

## GarageBand

---

使わないので消す。

本体と関連ファイルで2.3GBくらい空く。

## チェス

---

要らないので消す。

9MB、...と思ったらOSXに必要とされているため消せない...。

## Growl

---



デスクトップ通知ソフト。何かとこれに対応してるアプリやスクリプトがあるので入れておいたんだけど、1年位入れてたけど活用したことがなかったので削除。



200

あとアップデートの通知がウザい、AppStore経由で入れたのに非AppStore版のアップデート通知出してくるとか意味分からん。AppStoreのアプリアップデートしろよって話なのにな。

f

...

## AirServer

---

AirPlayサーバになるアプリ。iPhoneの画面を Mac 上に表示したりとか。

iOS8.1+Yosemite(OSX10.10)から、iPhoneを Mac に有線で繋げばQuicktimePlayerの新規ムービー収録から、iPhone画面を Mac 上に出したりそのまま録画したり出来るようになったので、AirServer要らなくなった。

## LightPaper

---

Markdownエディタ。

最近は MacDown 使ってる。

## TrashMe

---

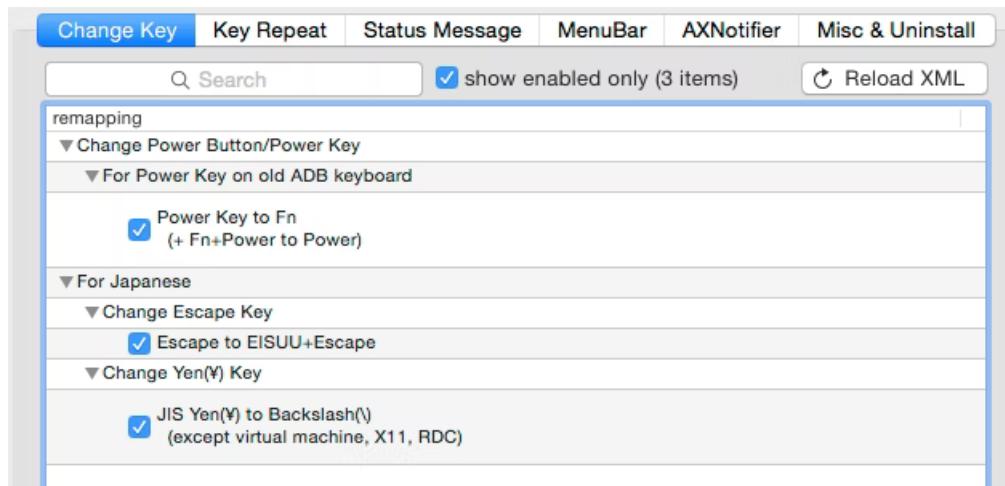
AppCleanerの方が使いやすいので乗り換えた。

## Karabiner (旧KeyRemap4MacBook)

---

キーボードのカスタマイズするやつ。macOS Sierraで動かなくなったので使わなくなったんだがなくてもそんなに困らないのでいいかな...。

- 今使ってる設定は以下のとおり。



- 
- Vimmerにとって Escape→EISUU+Escape のマッピングは最重要。

## XtraFinder

---

Finderを強化する。 OSXが上がるたびに不具合出るので面倒になった。

あと `⌘X`, `⌘V` でカット・アンド・ペーストが出来るのが良かったんだが、最近は `⌘C`, `⌘⌥V` の操作に慣れたので要らんくなった。

## MacDown

---

Markdownエディタ→vimでいいや

## WebStorm

---

HTMLやJavascript書くのに使う→最近使ってない

## Bartender

ステータスバーの増えすぎたメニューインをスッキリ整理してくれるアプリ。

→ macOS Sierra から ⌘ を押しながらドラッグすることでアイコンの位置入れ替えが出来るようになったのでそれで結構満足したので要らなくなつた。

↑これが、↓こうなる。

- General > Launch Bartender at Login → チェックON
- General > Autohides → チェックON
- Menu Items > 不要なアプリを片っ端から設定変更
  - Menu Bar Item → Show in Bartender Bar, not in Menu Bar に変更。
  - Item Updates → チェックON



Yoshiaki Kawazu

@kawaz

フォロー



## コメント

この記事にコメントはありません。

うそです  
n ^ \_ ^ n  
[(\*-\*')]  
Y Y

**投稿する**

編集 プレビュー ? ☺

テキストを入力

画像を選択 0B / 100MB 投稿

How developers code is here.

© 2011-2022 Qiita Inc.

ガイドとヘルプ	コンテンツ	SNS	Qiita 関連サービス	運営
About	リリースノート	Twitter Qiita (キータ) 公式	Qiita Team	運営会社
利用規約	イベント	Twitter Qiita マイルストーン	Qiita Jobs	採用情報
プライバシーポリシー	アドベントカレンダー	Twitter Qiita 人気の投稿	Qiita Zine	Qiita Blog
ガイドライン	Qiita 表彰プログラム	Facebook Qiita (キータ) 公式	Qiita 公式ショップ	
デザインガイドライン	API			
ご意見	Qiitadon (β)			
ヘルプ				
広告掲載				