

Zsh

ゴール

- 2時間の研修を通じて、Zsh の必要性を体で感じる事ができた状態になっている
- CLIの環境を整えることで、CLIでの操作が研修前よりも身近で扱いやすいものになっている

内容

- どんなにときに、なぜ？
- Zsh
- 実践
 - 環境を整える
 - CLIに触れてみる
 - 便利さを感じる

sfz での具体的な案件・場面

- 日常的に CLI (Zsh) を使うようになる

なぜ Zsh を利用するか

- GUI アプリケーションにはない便利なツール（コマンド）があるため
- ↑ さらに、GUIアプリケーションよりCLIの方が効率的な作業ができるケースがあるため
- CLIを起点として、開発に必要なになる技術／ツールを利用できるため
- 手順などをテキストとして表すことできるため
- ↑ 言い換えれば、CLI 環境の自動化は GUI の自動化よりも簡単のため
- ドキュメントに残せることで、その作業についての属人性をなくし、常に作業を共有／引継ぎできるようになるため
- → GitHub の [README.md](#) などのドキュメントにしやすい
 - e.g. <https://github.com/sforzando/marziale/blob/master/README.md>

Zsh って？

- [Z Shell - Wikipedia](#)
- CLI (Command Line Interface) で扱えるシェルのひとつ
- macOS 10.15 Catalinaから、デフォルトのシェルになった

Zsh 実践

環境を整えよう

iTerm2

- まず設定しておく良さそうなところ
 - Keys > Hotkey (→ すぐに CLI を開けるようにしたい)
 - Profiles > Terminal (→ 操作ログは遡れるようにしたい)
 - Scrollback Buffer: Unlimited scrollback
- 他、各自が使いやすいようにしていくと良いです
 - cf. [Documentation - iTerm2 - macOS Terminal Replacement](#)
- 文字カラーなどはThemeを見つけてきて使うのも良いかも？
 - cf. [【iTerm2】おすすめのカラースキームとカラースキームを適用する方法 | Blog](#)

Prezto

- Zsh を便利するためのフレームワーク
 - プロンプトの見た目や、コマンドショートカットなど、用意されているものを利用するだけで、必要十分な恩恵が得られると思う
- インストールしよう
 - cf. [sorin-ionescu/prezto: The configuration framework for Zsh](#)

- ターミナルエディタを変えてみよう
 - `$ git commit` など自動的に開かれるエディタを設定しておく
 - ターミナルエディタとしては、GNU nano、Vim、Emacsなどがある
 - Preztoの設定ファイルを編集し、適応させる
 - `$ open ~/.zprofile` ← テキストエディタで開くと思う
 - 2箇所を編集して保存
 - `export EDITOR='nano'`
 - `export VISUAL='nano'`
 - `$ source ~/.zprofile` ← 編集した設定を適応させる

- コマンドプロンプトを変えてみよう
 - どんなプロンプトが用意されているか一覧でプレビューしてみる
 - `$ prompt -p`
 - 変更してみる
 - `$ prompt -s <THEME NAME>`

- モジュールを有効にしよう
 - gitコマンドやdockerコマンドのように頻繁に使うコマンドのショートカット（エイリアス）があらかじめ登録されていたり、拡張モジュールが用意されている
 - どんなモジュールがあるか？
 - `$ ls ~/.zprezto/modules` もしくは、`$ open`
`https://github.com/sorin-ionescu/prezto/tree/master/modules`
 - 何ができるようになるかはREADME.mdを読めば良い
 - 設定ファイルを編集してモジュールを有効にする
 - 編集するファイルは、`~/.zpreztorc`
 - 30行目付近の `zstyle ':prezto:load' pmodule` に続けて、有効にしたいモジュールを追記する
 - `$ source ~/.zpreztorc` ← 編集した設定を適応させる

peco, ghq + peco

- コマンド履歴を一覧し、そこから実行する
 - cf. [peco](#) で [zsh](#) のコマンド履歴検索を超快適にする！ - Qiita
- gitのリポジトリを素早く開く
 - cf. [ghq](#), [peco](#), [hub](#) で快適 Git ライフを手に入れよう！ - Qiita

CLI で操作してみよう

- [zsh: The Z Shell Manual](#)
- [Mac ターミナルの基本的な使い方・操作方法（１） / Web Design Leaves](#)
 - ターミナル操作についてはこちらがわかりやすい（← ただし、bash）

- 変数、変数（パラメータ）展開、配列、ブレース展開

- `$ var=some_variable`

- `$ echo ${var}`

- `$ array=(a b c)`

- [zsh の配列操作の基本から応用まで - Qiita](#)

- `$ brace_arr={0..99}`

- コマンドの連携（パイプ）
 - | パイプ
 - コマンドの実行結果を使って、次のコマンドを実行する
 - cf. [リダイレクトとパイプ](#)
 - cf. [Linux コマンドを連続して使うには - Qiita](#)
- 参考: コマンドまとめ
 - [これから学ぶ macOS ターミナル・コマンドまとめ](#)

よく使うコマンドをピックアップ

- [ripgrep](#)
 - 高機能 grep (grep + find なイメージ)
 - grep - 文字列検索
 - find - ファイル検索
 - 使い方
 - [【搜索生活向上 ①】ファイル検索ツール「ripgrep」を利用して、俊足かつ効率的に見つけ出す。 - Qiita](#)
 - [rg の使い方メモ - Qiita](#)

- curl, wget
 - ネットワークアクセス、ファイルダウンロード
 - cf. [curl と Wget の比較 | POSTD](#)
 - cf. [wget と curl の使い分けについて | Red Full Moon](#)
- jq
 - JSON 編集ツール
 - 使い方
 - e.g. `$ curl https://www.stopcovid19.jp/data/covid19japan-fast.json | jq`

- open
 - ファイルをデフォルトのエディタで開いたり、URL をデフォルトのブラウザで開いたり、
- pbcopy
 - クリップボードにコピー

- FFmpeg

- 動画編集

- case. GitHubのPRに動画を貼りたいときに、.mov から.gifの生成に使用

- e.g. `$ ffmpeg -i for_PR.mov -filter_complex "[0:v] fps=10,scale=1280:-1,split [a][b];[a] palettegen [p];[b][p] paletteuse=dither=none" for_PR.gif`

やってみよう

- 次の条件を満たすフォルダ、ファイルを用意できる
 - 100個のフォルダを用意し、それぞれ AAA , AAB ,, ABA , ABB ,, のフォルダ名が付けられている
 - それぞれのフォルダの中に100個のファイルが入っている
 - AAA のフォルダには AAA-00000.txt ,,, AAA-00099.txt という名前のファイルが入っている