


# Lightning Web Componets について

 Slides are here

スライド:<https://powerninja.github.io/SSCLWC/ja/index.html>

リポジトリ:<https://github.com/powerninja/SSCLWC>

# はじめに

- どういう方に向けて話すかを記載しても良い
  - TODO:画像をもう少し増やしても良いかも
  - summer19 に登場し、約4年立ちました。
- 
- フロントの開発では、Aura や visualforce よりも選択される機会が増えてきていると思うので、少しずつ始めてみましょう! 😊👍

# トピックス

1. Lightning Web Components とAura の違いは？
2. なぜ Lightning Web Components を選択するのか
3. LWCの始め方
4. 使用した案件の紹介
5. おまけ

# Lightning Web Components と Aura の違いは？

- 共通箇所
  - salesforce上での見た目はほぼ同じ(LDSデザインで開発可能)
  - classic未対応
  - javascriptを用いた開発
  - 外部 JavaScript ライブラリの使用可能(静的リソースで読み込ませる)

# Lightning Web Components とAura の違いは？

- 相違点

- Auraは開発者コンソールで作成可能だが、LWCはvscodeが必要(サードパーティ製のエディタはあり)
- LWCで対応していない機能がまだある、その場合はAuraを作成する必要あり
- ただ、Auraは開発がアーカイブ化されている(サポートはしている)

Auraリポジトリ <https://github.com/forcedotcom/aura>

LWCリポジトリ <https://github.com/salesforce/lwc>

# なぜ Lightning Web Components を選択するのか

- (Auraと比べると)開発コミュニティが活発なため、新機能などに期待できる
- (Auraと比べると)標準的なjavascriptを使用することができるため、javascriptの開発経験がある方は開発しやすい
  - また、(Auraに比べると)直感的なため学習コストが低い
- (Auraと比べると)ファイルの数が少なく、初期段階の理解が早い(個人差あり)
- (Auraと比べると)パフォーマンスが良い

# なぜ Lightning Web Components を選択するのか

- LWCで作成されるファイル数

プロジェクト名(任意で設定可能)

| -HTML

| -javascript

| -xml

| -css(任意)

# おまけ

- こちらのスライドはgithub actionを使用して、vscodeでmarkdownを記載するだけでスライドを作成できるように。

または

- LWCのOSS版紹介



# Markdown 例

```
---  
marp: true  
paginate: true  
---  
  
<!-- _paginate: false -->  
  
# About [Marp CLI Action](https://github.com/KoharaKazuya/marp-cli-action) <!-- fit -->  
  
---  
  
## [Marp CLI Action](https://github.com/KoharaKazuya/marp-cli-action) is...
```

[このスライドのソース](#) を参照してください。

# "Marp CLI Action" は何を提供するの？

GitHub Action として [Marp CLI](#) を実行します。

リポジトリ内の Markdown ファイルから HTML、PDF、PowerPoint、画像を生成の自動化を素早く簡単にセットアップできます。

# 使い方

## 1. ワークフローに以下を追加します

```
- uses: KoharaKazuya/marp-cli-action@v1
```

または

1. [このテンプレート](#) を使用し、リポジトリを作成します
2. Marp の Markdown を書きます
3. コミットし、プッシュします

*Marp CLI Action は Markdown からファイルを生成するのみです。  
アップロード、リリース、公開などをする場合は他のアクションを*

# オプション

`action.yml` の `inputs` セクションを参照してください。

Marp CLI をカスタマイズするには `marp.config.json`、`marp.config.cjs`、`.marprc` [といった Marp CLI の設定ファイル](#) を使用してください。

設定ファイルはリポジトリルートに置くか、`config-file` input を指定します。

日本語ユーザーへ; Marp CLI オプションとして `lang: ja-JP` を指定してください。そうした場合、Marp CLI Action は CJK フォント問題を修正します。

楽しくスライドを書こう！ 🙌