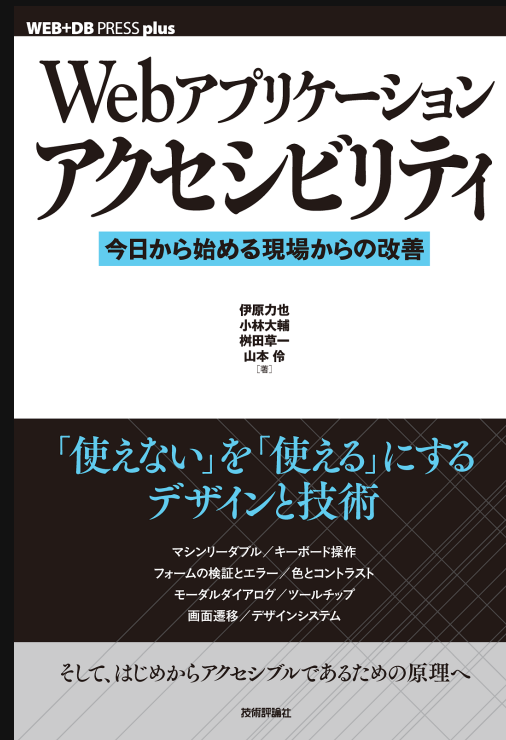


受託開発においてアクセシビリティ対応を当たり前にするにはどうしたら良いのか考える

「Webアプリケーションアクセシビリティ」という本を読んだ

- アクセシビリティの基礎が学べる
- 実践的な内容が多いのですぐ活用できる
- Webに関わる人みんなに読んでほしい



アクセシビリティとは？🤔

より多くの人々が、環境や状態に左右されることなく利用できること

✗ 障害者や高齢者だけのためのもの

✓ 誰もが一時的に障害を持つことはあるため、全ての人に対して考える

アクセシビリティ対応することのメリット(ビジネス視点)

- アクセスできるユーザーが増える
- 会社のイメージ向上
- 競合サービスとの差別化

自社開発の場合これらのメリットを感じやすい

でも受託開発だと

- 納品したら終わりだからユーザー数とかどうでも(ry
 - 納期が決まっていて対応してる余裕がない
 - 頑張っただけでも気づかれにくい上に工数だけかかる
- 対応するメリットが少ないので軽視されがち

😞 (アクセシビリティが重要なことは分かる。でも・・・)

じゃあどうしたら「当たり前」になるのか？

法が解決してくれる

- 障害者差別解消法が施行されたら対応せざるを得なくなる
- 海外では既に訴訟が起こっている

とはいえ法律違反だから仕方なく対応するというのはどうなの・・・

一人ひとり、ひいては会社全体がアクセシビリティに対する意識を高める

アクセシビリティは特別なことではなく当たり前のこと

アクセシビリティは難しい

実際の利用例を想定して考える

目が不自由な人など

スクリーンリーダーを使ってサイト上のテキストを読み上げます。

- 画像は読み上げられないので、altを設定する
- 読み上げ不要な画像はaltを空にする



DELETE

SEND



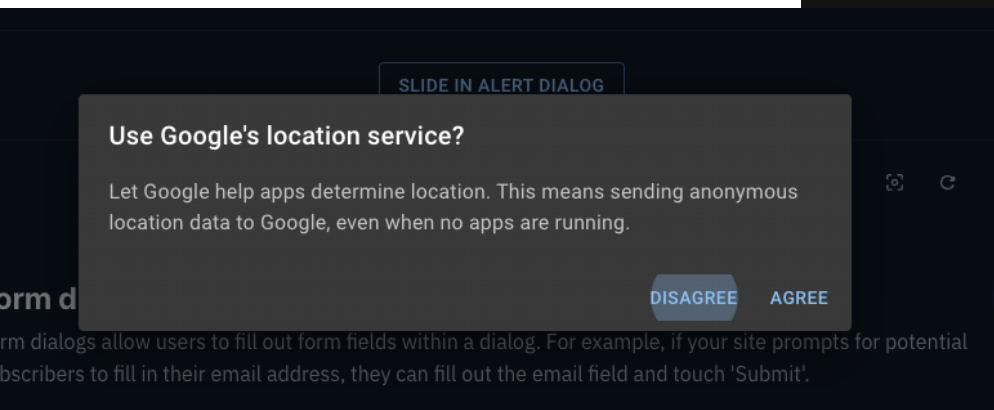
マウスが使えない人など

キーボードだけを使って操作します。

- Tabキーを押したときにどこにフォーカスされているか分かるようにする
- モーダルやメニューはキーボードだけで操作できるか



デザインをカスタマイズしたときにアクセシビリティが損なわれやすい



フォーカスが裏側に行ってしまうのか・Escで閉じるか

エンジニアが出来ること

- HTMLセマンティクスに沿ったマークアップをする
- ライブラリがアクセシビリティに対応しているか確認する
- ESLintやaxeなどのツールで機械的にチェックし、それをCIで実行する
- よく使う複雑UI(カルーセル、メニュー、ツールチップなど)は社内で知見を共有する

エンジニア以外が出来ること

- 画面設計やデザインの段階で考慮する
 - この複雑なUIは必要か？文章量が多すぎないか？配色は適切か？
- QAにアクセシビリティチェックを含める
 - スクリーンリーダーで読み上げられるか
 - Tabキーでフォーカス移動できるか

現状の課題と今後

アクセシビリティ対応がエンジニアやデザイナー個人の裁量に委ねられている

会社全体でアクセシビリティに対する意識を高めていきたい！