

WEB フロントエンド エンジニア 集会

2024 12

今月のフロントエンド

~ 2024 / 12 / 17

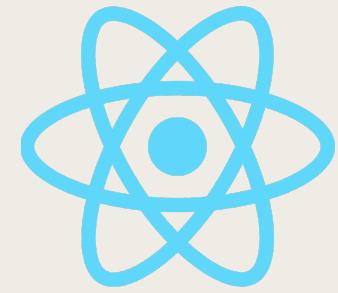
今月のフロントエンド

今月のフロントエンドのニュースを、以下のフォーマットで紹介します。

1. 何らかの出来事があった技術領域の名前(言語, F/W など)
2. 初見の人向けに、それが何者かを3行で解説
3. なにがあったかを解説

※ バージョンアップ情報は基本的には掲載しないことにしています。キリがない為。

**React and
its surroundings**

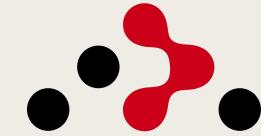


React 界隈に入さな進展

React 及び、それを取り巻く技術にて大きなアップデートがあった。
破壊的な要素を結構含むので、それらをまとめて紹介。

Remix React-Router

2024 / 11 / 22



React-Router とは？

- コンポーネント指向の web フロントエンドフレームワーク React にルーティング機能を追加するライブラリ
- ブラウザのページパスに応じてコンポーネントを切り替える
- 割とどのフレームワークにもあるルーティング系ライブラリの1つ

Remix とは？

- React をラップする“メタフレームワーク”
- ルーティング機能には React-Router を使用している
- ファイルルーティングなど、開発者体験高く React を利用できる

React-Router v7 がリリースされた・・が・・・

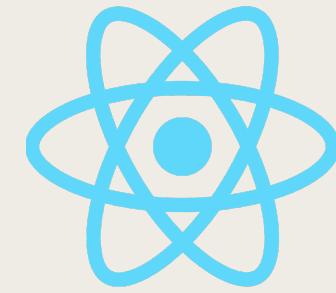
React-Router と Remix の機能差がほぼなくなったことから,
React-Router 側が Remix を吸収する形で v7 がリリースされた

* React-Router v7 は Remix v2 に当たる

React-Router v6, Remix v1 それぞれからのマイグレーションガイドラインがあるので参考に.

React

2024 / 12 / 5



React とは？

- コンポーネント指向の Web フロントエンドフレームワークの一つ
- 最も利用率と人気が高い
- meta 製 (facebook の頃に作られたので, organization は facebook になっている)

「なぜかぬつたか?」

React v19 がリリースされた
v18 から 2 年半ぶり

影響

メタフレームワーク系

- React-Router (Remix v2) .. 対応可
- Next.js .. 対応可

状態管理系

- React Redux .. 対応可
- recoil .. 不可
 - → 2023/4 頃からリリースがない
- jotai .. 対応可

その他

- Astro .. 対応可

React 関連の話は終わり.

Web Almanac

http↗
archive

Web Almanac とは？

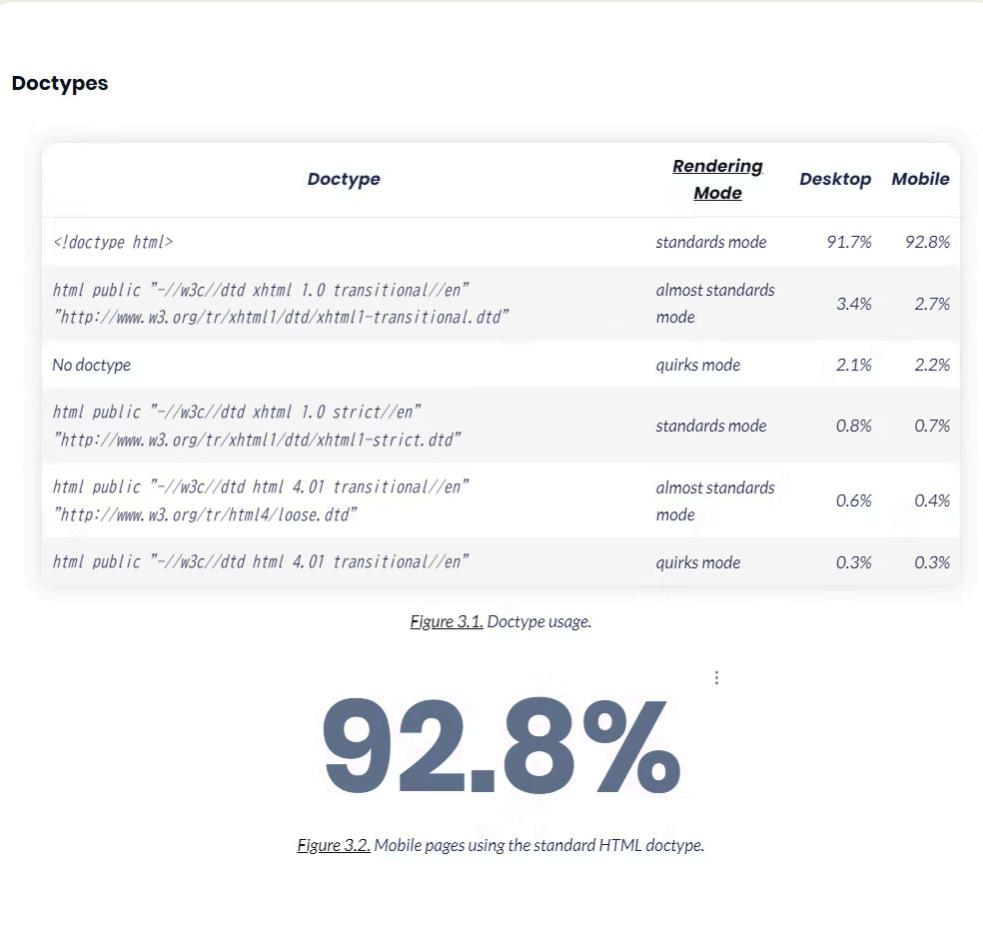
- Almanac .. 年鑑
- HTTP Archive が集計した年ごとのまとめ
- 2019 から集計している. ※ 2023 はなぜか欠番.

HTTP Archive とは？

- ウェブがどのように構築されているかを追跡することを使命として, 2010年に開始されました
- ウェブ上のトップサイトを定期的にクロールし, その分析結果をまとめている組織 / 及びそのサイト
- 統計結果を BigQuery で集計可能にしている

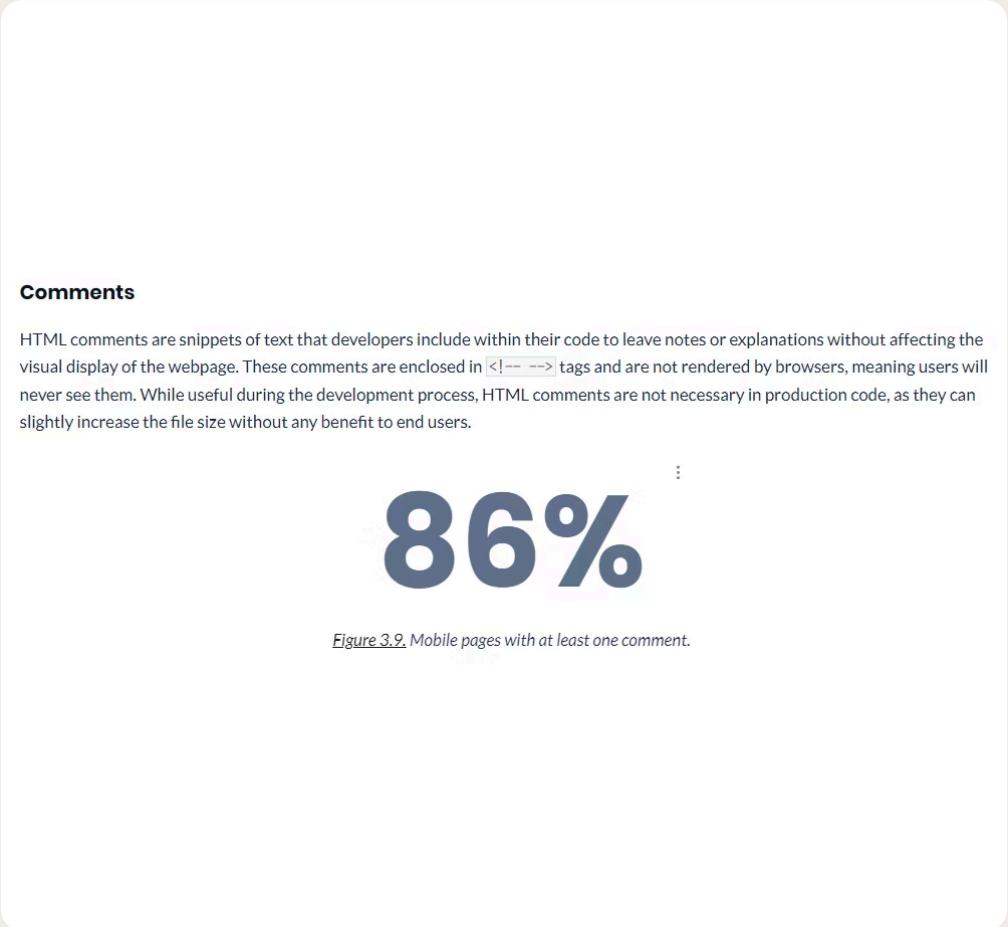
Web Almanac 2024 を軽く眺めてみる

Doctypes と レンダリングモード



- 91.7 % が <!doctype html> を書いてる
- ごくわずかに quirks mode が残っている
- standards mode は 2022 年に比べて 3 % ほど高い

コメント



- 86% のサイトが何らかのコメントを含む HTML
- コメントには IE かどうかで分岐する構文もある。
- モバイル向けサイト全体の 26% にまだ残っている。

特殊コメント ↓ こういうやつ

```
<!--[if IE]> <link rel='stylesheet' href='ie-only-  
styles.css'> <![endif]-->
```

人気のエレメント

Top elements

The following elements are used most frequently:

2021	2022	2023	2024
<code>div</code>	<code>div</code>	<code>div</code>	<code>div</code>
<code>a</code>	<code>a</code>	<code>a</code>	<code>a</code>
<code>span</code>	<code>span</code>	<code>span</code>	<code>span</code>
<code>li</code>	<code>li</code>	<code>li</code>	<code>li</code>
<code>img</code>	<code>img</code>	<code>script</code>	<code>script</code>
<code>script</code>	<code>script</code>	<code>img</code>	<code>img</code>
<code>p</code>	<code>p</code>	<code>p</code>	<code>p</code>
<code>link</code>	<code>link</code>	<code>link</code>	<code>link</code>
<code>meta</code>	<code>i</code>	<code>meta</code>	<code>path</code>
<code>i</code>	<code>meta</code>	<code>path</code>	<code>meta</code>

Figure 3.12. Most used elements.

The list remains largely consistent with previous years, but some shifts have occurred.

29%

Figure 3.13. Percentage of elements which are `div` elements.

`<div>` remains by far the most dominant element. So “divitis” is still a thing, and it doesn’t look like it’s going to change in the next few years.

- 圧倒的 `div`
- 全体の 29% が `div`
- `divitis` に終わりは来ないようだ

`divitis` .. あらゆるものを `div` で作ってしまったがために発生する技術的負債

(類似語 .. `classitis`)

おわり