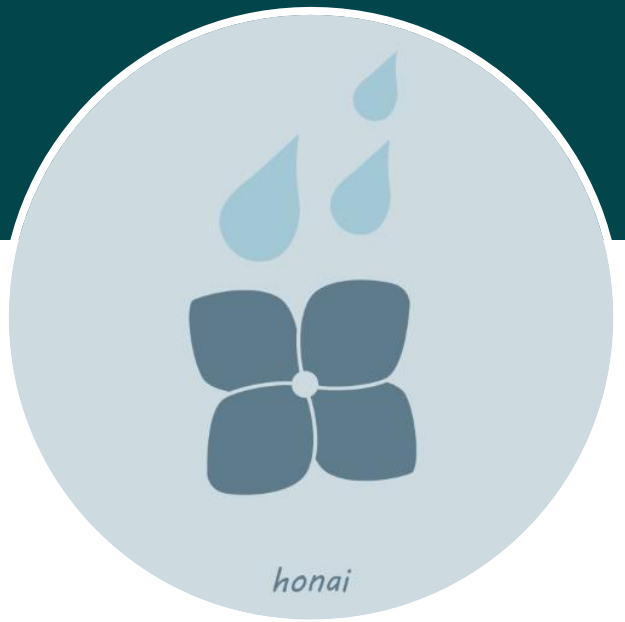
JSXでJSなしの静的サイトをつくる

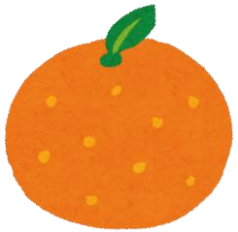
ほない (CAMPHOR-)



ほない

京都大学大学院 情報学研究科 修士1年
CAMPHOR- 2022年代表

好きな食べ物 : みかん



出 身 : 愛媛県

技 術 : Webフロントエンド、ネットワーク

好きな音楽 : サカナクション、ハニカムベアー

好きな漫画 : かぐや様は告らせたい

@honai

@_honai

honai.me

Reactは好きですか

JSXは好きですか

好きな静的サイトジェネレータは何ですか

Hugo, Gatsby, Jekyll etc...

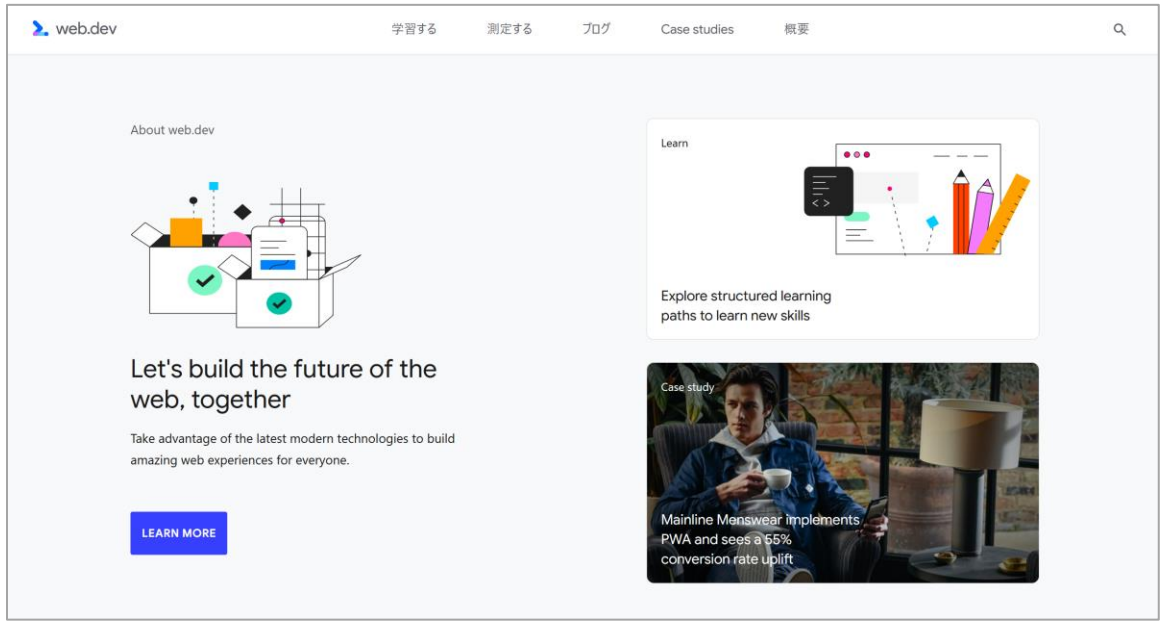
- JSXが好きな人向けの選択肢：Gatsby
- Gatsbyの特徴
  - Reactベース
  - SPAによるページ遷移で爆速
  - ReactなどのJavaScriptのバンドルが必要
  - GraphQLを使ってデータを定義する必要があり、少し難しい
- 今回は
  - クライアントのJavaScriptがないような静的サイトも、JSXで作りたい！を実現します

Eleventy (11ty), a simpler static site generator.

https://11ty.dev/

Eleventy (11ty), a simpler static site generator.

- GitHubスター：11.6k
  - 参考: hugo 57.6k, Gatsby 52.5k
- 有名どころだと web.dev にはEleventyが使われてます



Eleventyが “simpler” たるゆえん

```
_data/
└── profile.json
_includes/
└── post-page-template.ejs
posts/
├── hello-world.md
└── second-post.md
index.ejs
```

- フォルダ構造がとてもシンプル
- ←デフォルト設定だとこんな構造

  `` `npx @11ty/eleventy` ``
- この例の出力
  - /index.html
  - /posts/hello-world/index.html
  - /posts/second-post/index.html

Eleventyが “simpler” たるゆえん

- Frontmatterをベースとした設定

```
_data/
└── profile.json
_includes/
└── post-page-template.ejs
posts/
├── hello-world.md
└── second-post.md
index.ejs
```

index.ejs

```ejs
---
title: ポートフォリオ
---

<!DOCTYPE html>
<html lang="en">
<head>
  <title><%= title %></title>
</head>
<body>
  <h1><%= title %></h1>
  <p><%= profile.introduceText %></p>
</body>
</html>
```

Eleventyが “simpler” たるゆえん

- Frontmatterをベース基本とした設定

```
_data/
└── profile.json
_includes/
└── post-page-template.ejs
posts/
├── hello-world.md
└── second-post.md
index.ejs
```

posts/hello-world.md

```
---
layout: post-page-template.ejs
postTitle: はじめての投稿
---

# こんにちは
```

Eleventyが “simpler” たるゆえん

- Frontmatterをベース基本とした設定

```
_data/
└── profile.json
_includes/
└── post-page-template.ejs
posts/
├── hello-world.md
└── second-post.md
index.ejs
```

_includes/post-page-template.ejs

```ejs
<!DOCTYPE html>
<html lang="en">
<head>
  <title><%= postTitle %></title>
</head>
<body>
  <%- content %>
</body>
</html>
```

v1.0.0 が1月10日に公開

Releases | Tags

Find a release

10 Jan 2022

zachleat

v1.0.0

c8741c0

Compare

Eleventy v1.0.0: First! Latest

Eleventy v1.0.0 🎈🐁⚡

- Full docs for v1.0.0 on our web site

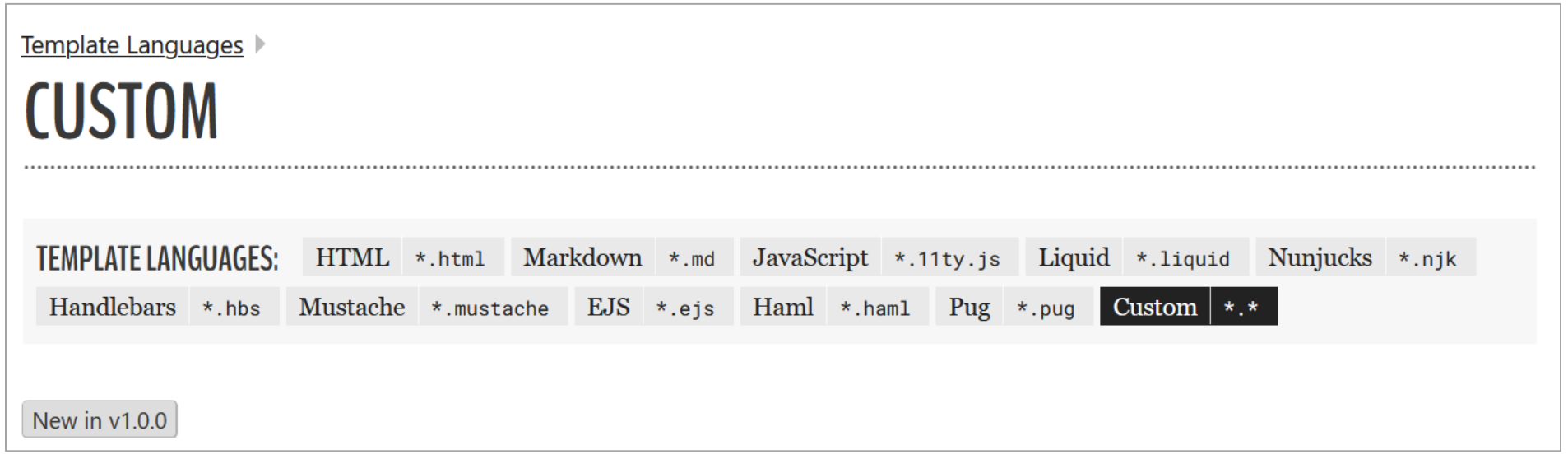
This project would not be possible without our lovely community. Thank you to everyone that built something with Eleventy (×476 authors on our web site!), wrote a blog post about Eleventy, contributed code to core or plugins, documentation, asked questions, answered questions, braved The Leaderboards, participated on Discord, filed issues, attended (or organized!) a meetup, said a kind word on Twitter ❤️.

I really wish I had time to list everyone, but I do want to mention a few folks that have made tremendous contributions:

- 🏆 A super special thanks to Peter DeHaan and Binyamin Green for their tireless contributions on the Eleventy Issue tracker.
- All of our supporters on Open Collective ❤️
  - Gold Sponsors: **Sanity.io**, **Nordhealth**, **Screen recorder for Mac**
  - Silver Sponsors: **Piccalilli**, **ESLint**, **Unabridged Software**, **PQINA**, **The Coders Guild**, **Bejamas**
  - A full list of Backers can be found below!
- Contribute on Open Collective
- How else can you contribute to Eleventy?

Install or Upgrade

v1.0.0 の新機能の1つ: Custom Template Language



- JSで好きなようにテンプレート言語を作れる
  - JSから呼び出せるならJSで書かれている必要もない（はず）

わかりやすい利用例: SCSSに対応させる

FILENAME .eleventy.js

```
// Don't forget to `npm install sass`!
const sass = require("sass");

module.exports = function(eleventyConfig) {
  eleventyConfig.addTemplateFormats("scss");

  // Creates the extension for use
  eleventyConfig.addExtension("scss", {
    outputFileExtension: "css", // optional, default: "html"

    // `compile` is called once per .scss file in the input directory
    compile: async function(inputContent) {
      let result = sass.compileString(inputContent);

      // This is the render function, `data` is the full data cascade
      return async (data) => {
        return result.css;
      };
    }
  });
};
```

https://www.11ty.dev/docs/languages/custom/

JSXに対応してみよう

.eleventy.js

```
...
  eleventyConfig.addExtension("jsx", require('./jsxConfig'));
...
```

jsxConfig.js

```
const { createElement } = require('react')
const { renderToStaticMarkup } = require('react-dom/server’)
```

JSXに対応してみよう

```
const { createElement } = require('react')
const { renderToStaticMarkup } = require('react-dom/server')

module.exports = {
  outputFileExtension: "html",
  init: () => {
    require("@babel/register")({
      extensions: [".jsx"],
    })
  },
  compile: (_, inputPath) => {
    const ExportComponent = require(inputPath).default;
    return function (data) {
      const Component = createElement(ExportComponent, data, null);
      return renderToStaticMarkup(Component);
    };
  },
  getData(inputPath) {
    return require(inputPath).data
  },
  read: false
};
```

Babelのregister機能で
requireしたときにJSX/ESM
をCommonJSに変換

データをpropsに渡して
HTMLにレンダリング

frontMatterの代わりに
export const data = {}

スタイルもCSS in JSで書きたい！

- 多くのCSS in JSライブラリは、ランタイム/Node.jsサーバーでのスタイル生成を前提としていて、今回の構成には合わせづらい
- Emotionというライブラリなら（詳しくは省くが）簡単に組み込める
- ただし、CSSファイルとして吐かれるのではなく、<style>タグとしてHTMLに埋め込まれます
  - classベースでスタイルが適用されるので、競合などの問題はないです
  - 良い方法がないか検討中

Emotion

https://emotion.sh/docs/introduction

EmotionでCSS in JS

Babel.config.json

```
{
  "presets": [
    [
      "@babel/preset-react",
      { "runtime": "automatic", "importSource": "@emotion/react" }
    ],
    [
      "@babel/preset-env",
      {
        "targets": {
          "node": "current"
        }
      }
    ]
  ],
  "plugins": ["@emotion/babel-plugin"]
}
```

こんなふうにページが書けるようになります

こんなふうにページが書けるようになります

```
import { css } from "@emotion/react";
import styled from '@emotion/styled'

export const data = {
  title: "タイトル"
}

export default (props) => (
  <div>
    <h1 css={
      css`
        color: red;
      `}
    >{props.title}</h1>

    <div dangerouslySetInnerHTML={{ __html: props.content }} />

    <Footer>ポートフォリオ</Footer>
  </div>
);

const Footer = styled.footer`
  color: blue;
`
```

FrontMatterの代わり

インラインでCSS-in-JS

マークダウン等の中身は
ここに

styled-components風の
書き方もOK

まとめ

- JSXでクライアントのJavaScriptなしの静的サイトをつくった
- Eleventyはいいぞ
- Babelのregisterでトランスパイルも簡単に
- EmotionでCSS in JS

JSXを楽しもう！

Thank you for listening!