

2019年11月19日

株式会社インプレスR&D

<https://nextpublishing.jp/>

標準技術を使って Web フロントエンド開発！
『やさしく学べる Web Components』発行
技術の泉シリーズ、11月の新刊

インプレスグループで電子出版事業を手がける株式会社インプレス R&D は、『やさしく学べる Web Components』(著者:大木 尊紀、水野 廉也、アサオカ アツコ、aggre)を発行いたします。

最新の知見を発信する『技術の泉シリーズ』は、「技術書典」や「技術書同人誌博覧会」をはじめとした各種即売会や、勉強会・LT 会などで頒布された技術同人誌を底本とした商業書籍を刊行し、技術同人誌の普及と発展に貢献することを目指します。

『やさしく学べるWeb Components』

<https://nextpublishing.jp/isbn/9784844378419>



著者:大木 尊紀、水野 廉也、アサオカ アツコ、aggre
小売希望価格:電子書籍版 1600円(税別)／印刷書籍版 1800円(税別)
電子書籍版フォーマット:EPUB3／Kindle Format8
印刷書籍版仕様:B5判／カラー／本文 90 ページ
ISBN:978-4-8443-7841-9
発行:インプレス R&D

<<発行主旨・内容紹介>>

近年、Web フロントエンドでは「コンポーネント指向」のフレームワークやライブラリが多く開発され、コンポーネント単位での開発が主流になってきました。Web Components は、特定のフレームワークやライブラリに依存せず、Web の標準技術を使用してコンポーネント単位での開発を可能にする考え方です。

本書では、Web Components とは何か、どうやって使うのかを解説します。Web Components をベースに Web アプリケーションを構築する JavaScript ライブラリである Polymer を使った Web サイト構築についても解説します。

(本書は、次世代出版メソッド「NextPublishing」を使用し、出版されています。)

Web Components の概要と未来についてまず解説

Web Componentsを利用できるライブラリーが多数出ており、手軽に利用することができます。本書3章では、Polymerを利用したWebアプリケーション開発手法を紹介します。

・webcomponents.js

- <https://github.com/webcomponents/webcomponentsjs>
- 未対応のブラウザにもWeb Componentsを提供するためのPolyfill

・Polymer

- <https://www.polymer-project.org/>
- Googleが中心となって開発した、宣言的にWeb Componentsを作成する機能セットを提供するライブラリー

・Slimjs

- <http://slimjs.com/>
- ES2015+のネイティブクラス継承を利用して、データバインディング等の拡張機能を提供するライブラリー

1.2 Web Componentsで変わる開発

巻には既にUIのコンポーネント化を提供するフレームワークが多くありますが、Web Componentsを利用するメリットとはなんでしょうか？

1.2.1 コンポーネント指向とAtomic Design

Webアプリケーション開発の現場で「コンポーネント指向（設計）」という言葉が使われて久しくなりました。画面上の機能をコンポーネント毎に定義・管理することで、生産性と保守性が向上するメリットがあります。近年ではコンポーネント指向を用いたフレームワークも多く登場しています。また、UIデザインの観点でもコンポーネント指向が取り入れられつつあります。

デザイン手法では、「Atomic Design」（アトミック・デザイン）がコンポーネント指向の道れを跟んでいると言えます。以下のような5つのレベルでコンポーネントをデザインします。

1. Atoms - 原子

機能的にこれ以上分割できないもの。ラベルやインプット要素に加えて、カラーパレットやアニメーションなどもここに含まれる。例) ラベル・インプット要素・ボタン等

2. Molecules - 分子

Atomsを組み合わせてできる要素。Moleculesは目的を持った最小のコンポーネント単位と言える。例) 入力フォーム = ラベル + インプット要素 + ボタン

3. Organisms - 有機体

AtomsやMoleculesを組み合わせて、具体的なUIを作成する要素群の単位。再利用可能であることが望ましい。例) ヘッダー、フッター、リストアイテム等

4. Templates - テンプレート

AtomsからOrganismsまでを組み合わせて、具体的な内容を持たないページの構成を表す。一般的にワイヤフレームと呼ばれるものに相当する

5. Pages - ページ

Templatesに具体的な内容（画像やテキストなど）が入ったもの。一般的にデザインカンパと呼ばれるものに相当する

Molecules以降のレベルはそれ以前のレベルの要素を元に作られ、レベルが低いほど抽象度が上がります。詳しい内容は提唱者であるBrad Frost氏のページを参照してください。（参照：<http://bradfrost.com/blog/post/atomic-web-design/>）

3章ではAtomic Design手法を用いたWebアプリケーション開発手法を紹介しています。

1.2.2 グローバルスコープとの戦い

生産性・保守性の向上が見込めるコンポーネント指向ですが、大きく立ちかかるとはグローバルスコープの壁です。HTMLやCSSは基本的にグローバルスコープなので、フレームワークやライブラリーを使用してコンポーネント化しても、そのDOMやCSSは外部から操作できます。アプリケーション側からコンポーネント側へ影響を与える可能性があり、コンポーネント単体で完結しているとは言えません。

Web Componentsはこの問題を、Shadow DOMを用いたコンポーネントのカプセル化で解決します。こちらも既に触れている通り、JavaScriptだけではなくHTML/CSSを含め要素に関連するものをまるごとカプセル化するため、コンポーネントのHTML/CSS/JavaScriptはグローバルスコープになりません。また、コンポーネント側でDOM構造からスタイル・挙動まで全て提供できるため、利用者は周辺への影響を気にすることなくコンポーネントの提供する機能を利用することができます。

1.3 Web Componentsの未来

1.3.1 HTML Importsの廃止とES Modules

Web Componentsを構成する4仕様のうちの4つ目で紹介したHTML Importsですが、前述の通り既に開発は中止されています。その代わりとしてECMAScriptのES Modulesの採用が「HTML Modules」という仕様が検討されています。

ES ModulesとはECMAScript 2015にて策定されたモジュールを扱うための仕様です。ES Modules（とくにimport構文）は策定に時間が掛かり、普及が遅れていました。これにはいくつか理由があるのですが、その一つがモジュールの読み込みに関する理由です。ES Modulesでは、再帰的にモジュールを読み込むため、読み込むモジュール数が増える問題がありました。HTTP/2は1リクエスト1コネクションのため、多くのモジュールを読み込むためには並列リクエスト数を増やさなければならず、そうするとブラウザやサーバーのリソース（メモリやポート）を激しく消費してしまいます。直列にリクエストするとリソースの消費を抑えられますが、読み込み速度が遅くなります。

Web Components を使ったウェブサイト制作について解説

第3章 Polymerでウェブサイト構築

この章では実際にWeb Componentsを使って、ウェブサイトを作っていきます。そして、静的ホスティングサービスのNetlifyを使って実際にウェブサイトを公開してみよう。

この章で作成したウェブサイトは下記URLで公開されています。

<https://inu-polymer-sample.netlify.com>

また、GitHubにソースコードを公開しています。コードをすぐ動かしてみたい人はチェックアウトしてみてください。

<https://github.com/inutetraplus/polymer-sample>

3.1 Polymer 3.0

早速ウェブサイトを作っていきますが、ここで1つ注意しなければいけない点があります。それは、全てのブラウザでWeb Componentsの全ての機能が使えるわけではないということです。第1章でも述べましたが、ブラウザの実装に差異があるためネイティブのWeb Componentsのみでウェブサイトを作ることはまだできません。2018年10月現在、最新のブラウザの対応状況は下記表のとおりです。¹

	Chrome	Opera	Safari	Firefox	Edge
HTML Templates	○	○	○	○	○
Custom Elements	○	○	○	△	×
Shadow DOM	○	○	○	△	×
ES Modules	○	○	○	○	○

○: STABLE, △: DEVELOPING, △×: ON HOLD, ×: CONSIDERING

上の表を見ると、Google Chromeでは全ての機能を使用できますがFirefoxやMicrosoft EdgeではTemplates以外の機能を使用できないことがわかります。そのため、全てのモダンブラウザで閲覧可能なウェブサイトを作るには、Web ComponentsのPolyfillを使用する必要があります。

本書では、PolyfillであるwebcomponentsjsをバンドルしたJavaScriptフレームワークであるPolymer 3.0を使用してウェブサイト構築していきます。

¹ <https://www.webcomponents.org/>
² <https://www.webcomponents.org/>

3.1.1 Polymerの特徴

PolymerはGoogleのChromeチームが開発を主導しているOSSライブラリーで、Web Componentsの機能を全て使用することができます。また、Polymerは下記の5つの機能を備えており、Web Componentsをより簡単に作成することができます。

- ・Custom Elementsの登録
- ・ライフサイクルコールバックの使用
- ・プロパティの監視
- ・テンプレートでのShadow DOMの作成
- ・データバインディングの使用

おまけ

第1章でも紹介しましたが、先日公開されたYouTubeの新UIはこのPolymerを使用して作られています。ChromeのDeveloperToolなどを使えば、カスタムタグなどが使われているのが確認できます。

<https://www.youtube.com>

3.2 環境構築

まずはPolymerの開発環境を構築しましょう。

3.2.1 前提条件

次の環境を前提としています。

- ・Node.js (8.12.0)
- ・npm v6以降

1. polymer-cliをインストール

まずPolymer CLIをインストールします。

```
npm install -g polymer-cli
```

2. 新しいフォルダを作成

```
mkdir polymer-sample  
cd polymer-sample
```

最後に応用編として lit-html の紹介を掲載

4.1.3 lit-htmlにできること

lit-htmlにできるすべての機能をご紹介します。

1. テンプレートを作る
2. テンプレートに動的な値を挿入する
3. テンプレートから DOM に値を渡す
4. テンプレートにイベントをバインドする
5. HTMLを差分箇所だけ書き換える
6. テンプレートを操作する独自の関数を作る

lit-htmlはこの6つの機能だけを提供します。基本的には**テンプレートの生成**と、そのテンプレートによる**HTMLの差分更新**を担うのがlit-htmlの役割と言えます。

最後のテンプレートを操作する独自の関数を作るという機能にはテンプレートの生成を補助する目的がありますが、うまく応用すればlit-htmlだけでアプリケーションを構築するのにも役立ちます。

4.1.4 インストール方法

自分のプロジェクトにlit-htmlを導入する方法を書いておきます。

```
npm install lit-html
```

使うときは、lit-htmlがexportしているモジュールをimportします。lit-htmlにはexport defaultがないので、必ず名前付きでimportします。例えばこんなふうに。

```
import { html, render } from 'lit-html'
```

4.2 lit-html APIs

ここからはlit-htmlのAPIをひとつずつ見ていきます。といっても、主なAPIはhtml.render.directiveの3つだけです。

4.2.1 html - テンプレートを作る

lit-htmlの基礎を成すAPIです。Tagged templatesによって、テンプレートをJavaScript標準の記法で定義できます。基本的な使い方は次のように、htmlに続いてテンプレートリテラルを書きます。(この仕様こそがTagged templatesです)

```
import { html } from 'lit-html'

export const hello = html`
```

```
<p>Hello</p>
```

これでhelloテンプレートが作られました。

戻り値はTemplateResultという型のlit-htmlのテンプレート用オブジェクトです。このオブジェクトを後の章にあるrenderに渡すことで、テンプレートがHTMLとしてブラウザに描画されます。lit-htmlの機能のうちほとんどが、このhtmlによって実現されています。先に挙げたリストのうち、次の4つはhtmlの役割です。

1. テンプレートを作る
2. テンプレートに動的な値を挿入する
3. テンプレートからDOMに値を渡す
4. テンプレートにイベントバインドする

1は前述の部分で説明したので、ここからは2以降についてそれぞれ簡単にしていきたいと思

います。

テンプレートに動的な値を挿入する

JavaScriptでHTMLを操作するならテンプレートの一部を変数で書き換えたいと考えるはずす。例えば、テキストノードの一部を書き換える場合の例を以下に示します。

```
import { html } from 'lit-html'

export const hello = (name: string) =>
  html`
  <p>Hello ${name}</p>
  `
```

このテンプレート操作はTagged templatesの仕様を利用しているにすぎません。つまりlit-htmlはライブラリ独自のコンパイラを不取にし、そのままブラウザで実行可能なコードとしてテンプレートを定義しています。標準仕様なので、IDEやエディタの環境構築をしなくてもすぐ開発に取りかかれるという利点もあります。

■ 属性値を書き換える

要素の属性を書き換えたいときは次のように書きます。

```
import { html } from 'lit-html'

export const photo = (url: string) =>
  html`
  <img src=${url} />
  `
```

<<目次>>

第1章 Web Components とは

1.1 基本概念

1.2 Web Components で変わる開発

1.3 Web Components の未来

1.4 1章まとめ

第2章 Web Components を使ってみよう

2.1 Custom Elements

2.2 Shadow DOM

2.3 <slot>要素

2.4 HTML Template

2.5 ESModules

2.6 2章のまとめ

第3章 Polymer でウェブサイト構築

3.1 Polymer 3.0

3.2 環境構築

3.3 Custom Element を作る

3.4 子要素を使う

3.5 配布されてる Custom Element を使う

3.6 ページを作る

3.7 サイトを公開する

3.8 発展: Redux

3.9 Polymer の今後

3.10 まとめ

3.11 参考文献

第4章 lit-html(応用編)

4.1 lit-html の基本

4.2 lit-html APIs

4.3 Simple But Powerful

<<著者紹介>>

大木 尊紀(おおき たかのり)

都内で働くフロントエンドエンジニア。猫、温泉、ゲーム、ロボットアニメ、自転車、フォントが好き。HTML/CSS/JavaScript とたまに WordPress。最近では React や Vue をよく書いている。

Twitter : @takanoripe

GitHub : @takanorip

水野 廉也(みずの れんや)

名古屋で働いているフロントとかバックエンドとかやってるひと。睡眠欲の塊。「めろたん」と呼ばれている。

Twitter : @renyamizuno_

Github : @renyamizuno

アサオカアツコ

猫派の甘党、胃が弱い系WEB屋。フリーランスのWebデザイナー/フロントエンドエンジニアとして日々をぼんやり過ごしている。基本出不精だが、ライブ参戦の為に台湾まで出向く信仰心も持つ。

Twitter : @atsuco

aggre(あぐり)

スタートアップでデザイナーしてたらSPA/PWA/AMP 書いたりサーバーレスやってた。FRAME00 CTO。TypeScript と lit-html とアニメが大好き。

Twitter : @aggre_

<<販売ストア>>

電子書籍:

Amazon Kindle ストア、楽天 kobo イーブックストア、Apple Books、紀伊國屋書店 Kinoppy、Google Play Store、honto 電子書籍ストア、Sony Reader Store、BookLive!、BOOK☆WALKER

印刷書籍:

Amazon.co.jp、三省堂書店オンデマンド、honto ネットストア、楽天ブックス

※ 各ストアでの販売は準備が整いしだい開始されます。

※ 全国の一般書店からもご注文いただけます。

【インプレス R&D】 <https://nextpublishing.jp/>

株式会社インプレスR&D(本社:東京都千代田区、代表取締役社長:井芹昌信)は、デジタルファーストの次世代型電子出版プラットフォーム「NextPublishing」を運営する企業です。また自らも、NextPublishing を使った「インターネット白書」の出版など IT 関連メディア事業を展開しています。

※NextPublishing は、インプレス R&D が開発した電子出版プラットフォーム(またはメソッド)の名称です。電子書籍と印刷書籍の同時制作、プリント・オンデマンド(POD)による品切れ解消などの伝統的出版の課題を解決しています。これにより、伝統的出版では経済的に困難な多品種少部数の出版を可能にし、優秀な個人や組織が持つ多様な知識の流通を目指しています。

【インプレスグループ】 <https://www.impressholdings.com/>

株式会社インプレスホールディングス(本社:東京都千代田区、代表取締役:唐島夏生、証券コード:東証1部9479)を持株会社とするメディアグループ。「IT」「音楽」「デザイン」「山岳・自然」「旅・鉄道」「学術・理工学」を主要テーマに専門性の高いメディア&サービスおよびソリューション事業を展開しています。さらに、コンテンツビジネスのプラットフォーム開発・運営も手がけています。

【お問い合わせ先】

株式会社インプレス R&D NextPublishing センター

TEL 03-6837-4820

電子メール: np-info@impress.co.jp