



HTML Modules

2017.11.15 @サイボウズ 東京オフィス
Takayoshi Kochi / 河内 隆仁
twitter: @tkochi



自己紹介

- 河内 隆仁 (こうち たかよし)
- BlinkでWeb Componentsまわりをやっています
- Real Web Developerではありません



おさらい: HTML Imports / 使用法

使い方: HTML内で

```
<link rel="import" href="my-button.html">
```

API:

```
HTMLLinkElement.import (readonly)
```

=> ロードされたDocumentを返す

Import内のエレメントの取り方

```
var link = document.querySelector("link[rel=import]");  
var el = link.import.querySelector("#my-element");
```

超Simple!!



おさらい: HTML Imports / 利用例

index.html:

```
<link rel="import" href="my-button.html">
<link rel="import" href="my-input.html">
<my-input></my-input>
<my-button>OK</my-button>
```

my-button.html:

```
<link rel="import" href="common.html">
<script>
class MyButton extends HTMLElement { ... }
window.customElements.define('my-button', MyButton);
</script>
```

my-input.html:

```
<link rel="import" href="common.html">
...
```



おさらい: HTML Imports / 動作

<script>の実行:

my-button.html:

```
<link rel="import" href="common.html">
```

my-input.html:

```
<link rel="import" href="common.html">
```

common.html:

```
<script> <= 二度目は実行されない
```

```
...
```

```
</script>
```

一つの依存ツリー内で<script>の実行順序は決定的
(ただし、<link async> <script async>など例外あり)



おさらい: HTML Imports / うれしい点など

- `<script>` が実行される
 - Custom Elementの定義などに便利
- 依存関係を解決しながらロードしてくれる
 - ロード→HTMLパース→script実行が(うまくいけば)きれいなパイプラインに乗る
 - 重複も解決される
- テンプレート、スタイル、スクリプトを一つのファイルにかけてわかりやすい、うれしい
- Vulcanizeがcatだけでできる

おさらい: HTML Imports / 問題点など

- Chrome以外の実装なし
- Documentのツリーができてしまう
 - しかしHTML ImportsのDocumentは表示されない
 - よくある勘違い: client side include
 - <template>, <script> 以外はあまり意味を持たない
 - documentって何を指している?
 - `let doc = document.currentScript.ownerDocument;`
 - base URL問題 (imageの投機的ロードなど)
 - Quirks modeなどの諸々のDocumentが持つ問題を引きずる
- <style>がメインドキュメントに適用される
 - Chrome M65で廃止予定
 - /deep/などでテーマ指定していた過去の名残
- ES Modulesと相互運用できない
 - 依存解決の方法は似ているが一つのツリーに混ざることはない
 - import/exportできない



HTML Modules

<https://github.com/w3c/webcomponents/issues/645>

- 長大なスレッド(100+)
- みどころ
 - HTML moduleとはそもそも何か？
 - Frameworkの人たちの興味は？
 - HTML-in-JS vs JS-in-HTML (JSX, lit-html, ...)
 - Appleのアイデア (import ... from ... as Foo)
 - AshleyScirraさん ([Scirra Construct](#)というゲームエディタの作者)
 - Symbol.importer
- 問題: 論点が多すぎてまとまらない



HTML Modules / 具体例

```
<script type="module">  
  
import tmpl from 'my-template.html';  
  
document.querySelector('#id').appendChild(  
    document.importNode(tmpl.content));  
  
</script>
```



HTML Modules / 具体例

```
<script type="module">  
import * from 'lgm-button.html';  
</script>  
<lgm-button></lgm-button>
```



HTML Modules / 具体例

```
<template>
```

```
  <style>...</style>
```

```
  <button>LGTM</button>
```

```
</template>
```

```
<script>
```

```
class LGTMButton extends HTMLElement { ... }
```

```
window.customElements.define('lgtm-button', LGTMButton);
```

```
</script>
```



HTML Modules / 具体例 (import/export)

```
<template>
  <style>...</style>
  <button>LGTM</button>
</template>

<script>
import MyButton from './my-button.html';

export class LGTMButton extends MyButton { ... }

window.customElements.define('lgtm-button', LGTMButton);

</script>
```



HTML Modules / 議論

- なにがうれしいか
 - <template>を読み込みたい!
 - lit-html / template partsは希望の光
 - innerHTML / コンパイルされた多数の createElement() じゃないDOM生成手段
 - JSで文字列からDOMを生成 (innerHTMLとか)よりメモリ/CPU的に効率的
 - ネイティブHTMLパーサーを使うと性能が出る
 - HTMLパーズングは重いといえどロード時間に占める割合は低い
- 言ってはいけない
 - Webpack使えば?



HTML Modules / 議論

- TPAC 2017で出たポイント
 - JS moduleベースのシステム
 - Inert なHTMLがロードできる
 - single-file componentが定義できる
 - `<script>` が実行できる?
 - DOMの一部を exportできる?
 - 他のモジュールを importできる?
 - Document, or DocumentFragment?
 - サードパーティのモジュールのための isolatedな空間
 - 具体的なユースケースで話をしようぜ