

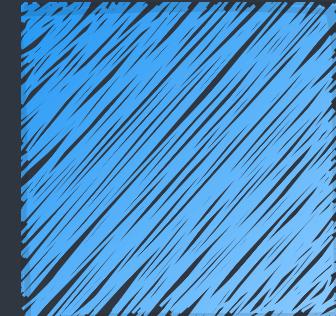
Web Components:



Natif



StencilJS



LitElement





Julien Renaux

GDE Web, Freelancer

@julienrenaux <<https://twitter.com/julienrenaux>>

<https://julienrenaux.fr/>

<<https://julienrenaux.fr/>>



Igor Laborie

Expert Web & Java

@ilaborie <<https://twitter.com/ilaborie>>

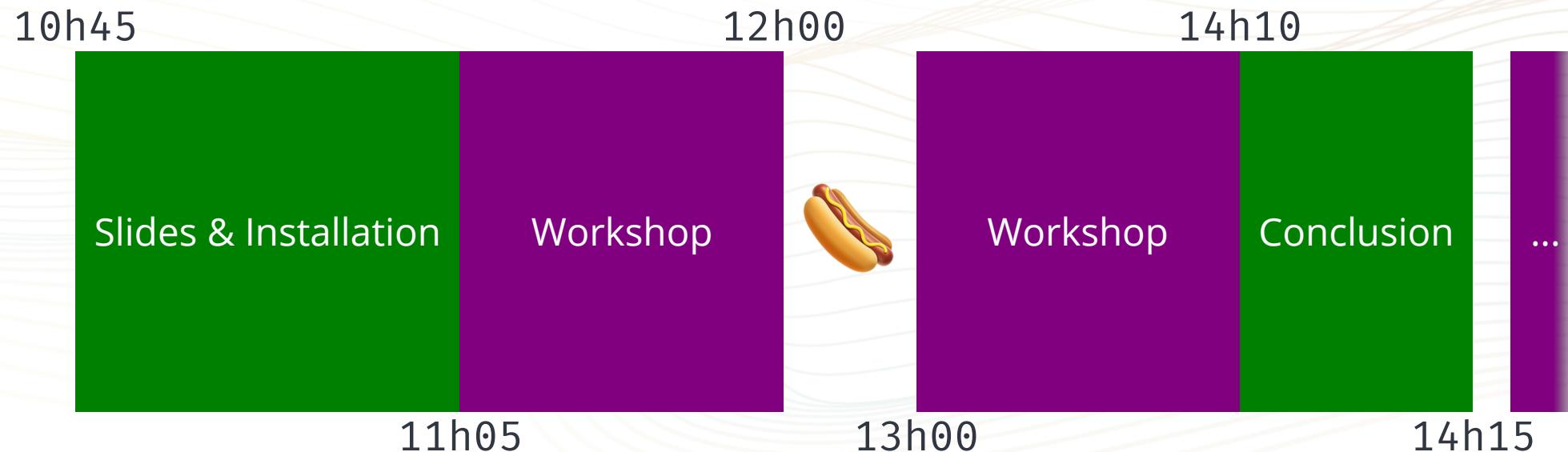
igor@monkeypatch.io

<<mailto:igor@monkeypatch.io>>





Roadmap



Instructions

<https://github.com/ilaborie/webcomponents-devoxx-19> <<https://github.com/ilaborie/webcomponents-devoxx-19>>

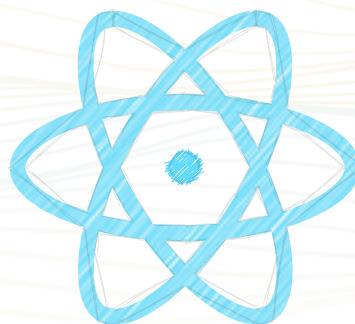
Puis dans les répertoires native, stencil,
lit-element, faire



```
$ npm ci
```

Développons une application Web en 2019

Commençons par choisir un Framework



React



Vue.js



Angular

Puis comment allons-nous écrire le style

- CSS
- Sass/Scss
- Less
- Stylus
- CSS-in-JS
- PostCSS
- NextCSS
- ...

Puis construisons notre application

- Webpack
- ParcelJs
- RollupJs
- Bazel

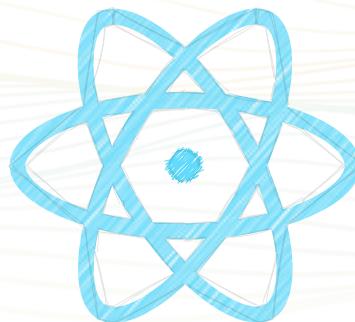


**Développer une application en JS n'est
plus simple...**

Ça va trop vite ...

A screenshot of a Twitter post. The profile picture is a small circular image of a person with short hair. The username is "I Am Devloper" and the handle is "@jamdevloper". The tweet text is: "I think I've had milk last longer than some JavaScript frameworks." Below the tweet are engagement metrics: 1,123 likes and 2:22 PM - Dec 4, 2014. To the right of the tweet is a blue Twitter icon. At the bottom of the card, there is a blue link that says "1,748 people are talking about this" followed by a right-pointing arrow.

...Interopérabilité ne vient plus gratuitement...



React

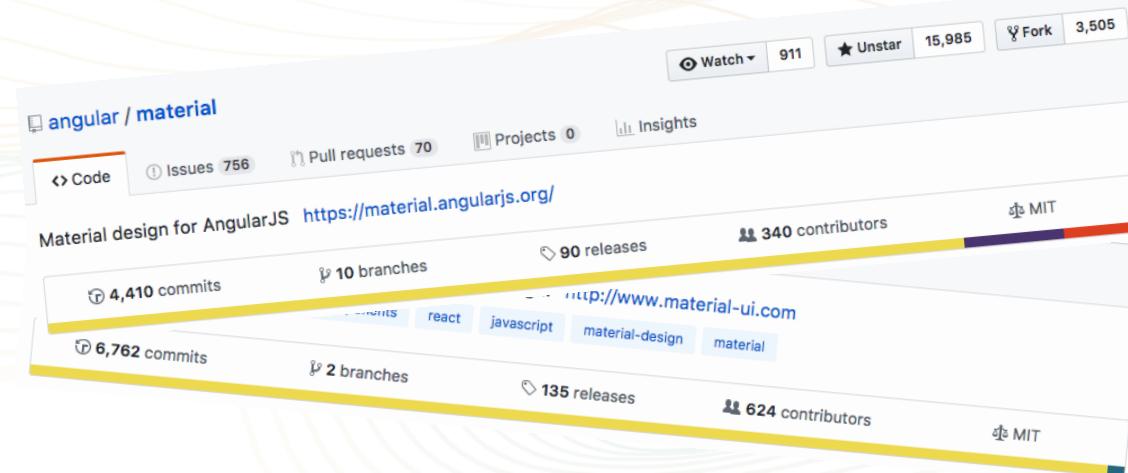


Vue.js



Angular

...et on réinvente sans arrêt la roue !





[material-components / material-components-web-components](#)

Watch 101 Star 786 Fork 137

Code Issues 75 Pull requests 28 Projects 1 Insights

Material Web Components - Material Design implemented as Web Components <https://material-components.github.io...>

290 commits 8 branches 13 releases 16 contributors Apache-2.0

Branch: master New pull request Create new file Upload files Find File Clone or download

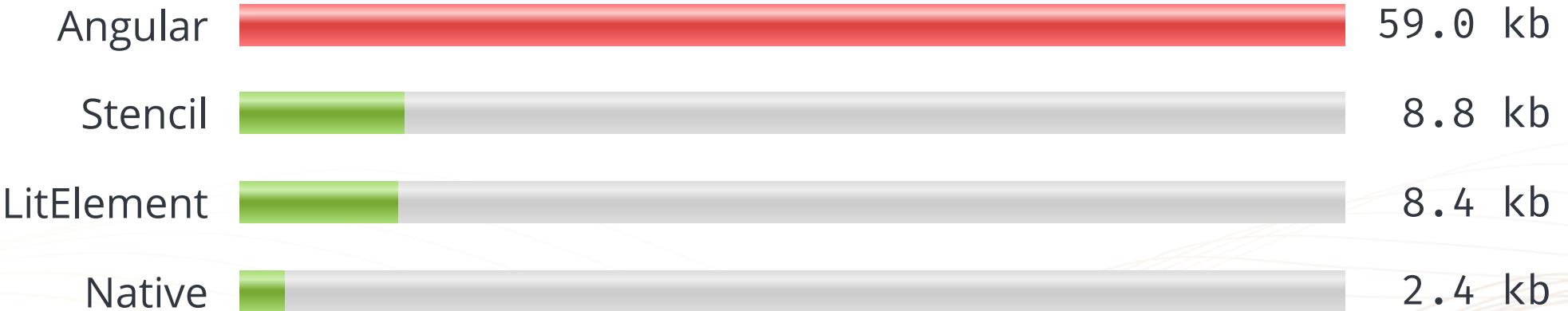
Material Web Components

► <https://github.com/material-components/material-components-web-components> <<https://github.com/material-components/material-components-web-components>>

DO IT! DO IT! DO IT!

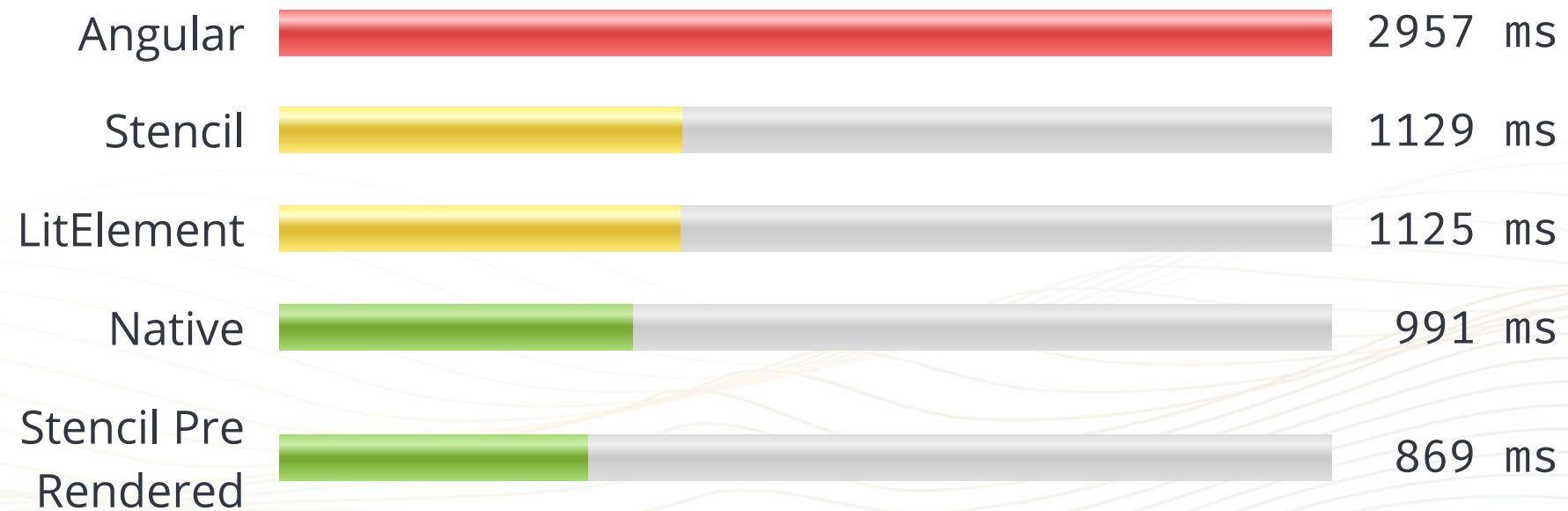


**Toute cette complexité doit
s'arrêter avec les Web
Components**



Size matters (Gzipped)

- 😢 Stencil et litElement sont 5 fois plus petits qu'Angular
- 😱 Native est 23 fois plus petit qu'Angular



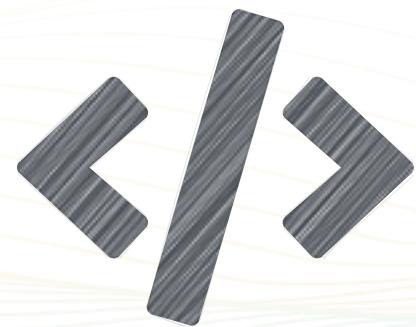
Le temps ça compte (FMP 3G 📱 en ms)

😢 Angular est 3 fois plus lent

Web Components



Spécifier par le **World Wide Web Consortium** (W3C)
Débuté en **2012**



Custom
Elements



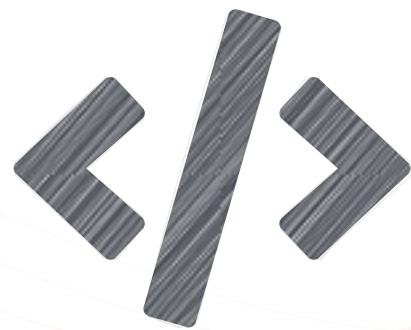
Shadow DOM



HTML templates



HTML imports



Custom Elements

“Les *Custom Elements* sont la capacité de créer un balise HTML avec ses propres attributs et méthodes



Shadow DOM

“Le Shadow DOM fournit l'encapsulation du DOM et du CSS”



HTML templates

“Définit un bloc d'HTML réutilisable au moment de l'exécution”

```
JS
class PopUpInfo extends HTMLElement {
  constructor() {
    super();
    // ...
  }
  // ...
}

customElements.define('popup-info', PopUpInfo);
```



```
<twitter-widget class="twitter-tweet twitter-tweet-rendered" id="twitter-widget-0" style="position: static; visibility: visible; display: block; transform: rotate(0deg); max-width: 100%; width: 500px; min-width: 220px; margin-top: 10px; margin-bottom: 10px;" data-tweet-id="540481335362875392">
...
  <#shadow-root (open) == $0>
    <style type="text/css">.SandboxRoot { display: none; max-height: 10000px; }</style>
    <div data-twitter-event-id="0" class="SandboxRoot env-bp-350" style="position: relative;">...
      <style type="text/css">...</style>
    </div>
</twitter-widget>
```

Support des navigateurs



11



18



65



72



12

	IE	Edge	Firefox	Chrome	Safari
Custom Elements (V1)	😢	😢	😊	😊	😊
Shadow DOM (V1)	😢	😢	😊	😊	😊
HTML templates	😢	😊	😊	😊	😊

Supported

Partial Support

Not Supported

<https://caniuse.com> <<https://caniuse.com>>

Support avec le polyfill

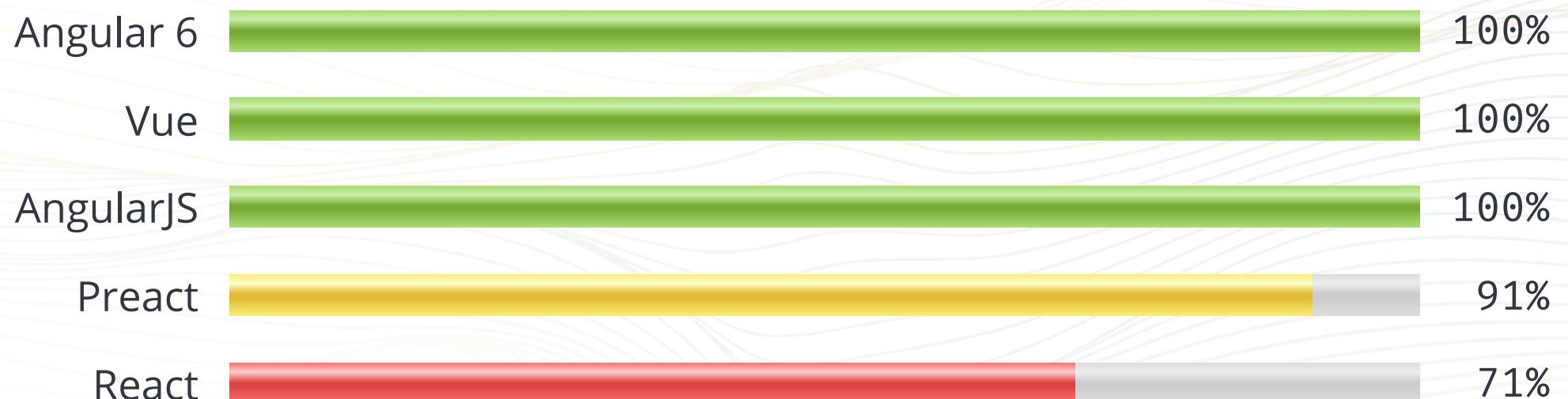


	11+	Edge	Firefox	Chrome	Safari
Custom Elements (V1)	✓	✓	✓	✓	✓
Shadow DOM (V1)	✓	✓	✓	✓	✓
HTML templates	✓	✓	✓	✓	✓

<https://github.com/webcomponents/webcomponentsjs>

<<https://github.com/webcomponents/webcomponentsjs>>

Interopérabilité des Web Component



<https://custom-elements-everywhere.com/> <<https://custom-elements-everywhere.com/>>

StencilJS



<https://stenciljs.com/> <<https://stenciljs.com/>>



Projet **Open Source**, ➡ MIT License <[https://github.com/ionic-](https://github.com/ionic-team/stencil/blob/master/LICENSE)

team/stencil/blob/master/LICENSE>

Créé par l'équipe d'**Ionic** en 2017

5.2k ⭐ sur github

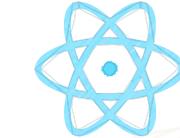


**StencilJS n'est pas un autre
framework**



**StencilJS c'est un compilateur qui
génère des web components**

StencilJS c'est un ensemble de bons outils

				
JSX / Virtual DOM	✗	✗	✓	✓
TypeScript	✗	✓	✗	✓
Decorators	✗	✓	✗	✓
Prerendering SSR	✗	✗	✗	✓

StencilJS marche partout

Il charge les polyfills à la demande

La syntaxe de Stencil est concise

Web component

```
1 class TodoItem extends HTMLElement {
2     constructor() {
3         super();
4         this._root = this.attachShadow({ 'mode': 'open' });
5         this._checked = false;
6         this._text = '';
7     }
8     connectedCallback() {
9         this._root.innerHTML =
10         `
```

Polymer 2

```
1 <link rel="import" href="../../bower_components/polymer/polymer.html">
2 <dom-module id="todo-item">
3   <template>
4     <style>
5       <style>
6         :item[[isCompleted({checked})]]>
7           <input type="checkbox" value="{{(checked)}}> checked="{{(checked) ? 'checked' : ''}}>
8           <label>{{text}}</label>
9           <button class="destroy" on-click="handleOnRemove">x</button>
10      </style>
11    </template>
12    <script>
13      class TodoItem extends Polymer.Element {
14        static get is() { return 'todo-item'; }
15        static get properties() {
16          return {
17            checked: { type: Boolean, value: false },
18            index: { type: Number, },
19            text: { type: String, value: '' }
20          };
21        }
22        handleOnRemove(e) {
23          this.dispatchEvent(new CustomEvent('remove', { detail: e }));
24        }
25        handleOnChecked(e) {
26          this.dispatchEvent(new CustomEvent('toggle', { detail: e }));
27        }
28        isCompleted(completed) {
29          return completed ? 'completed' : '';
30        }
31      }
32      window.customElements.define(TodoItem.is, TodoItem);
33    </script>
34  </dom-module>
```

Angular Elements

```
1 import { Component, EventEmitter, Input, Output, ViewEncapsulation } from '@angular/core';
2
3 @Component({
4   selector: 'todo-item',
5   template: `
6     <li class="item" [class.completed]="checked">
7       <input type="checkbox" [checked]="checked" (change)="handleOnChecked($event)">
8       <label>{{text}}</label>
9       <button class="destroy" (click)="handleOnRemove($event)">x</button>
10    </li>
11  `,
12  styles: [
13    ''
14  ],
15  encapsulation: ViewEncapsulation.Native
16 })
17 export class TodoItem {
18   @Input() checked: boolean;
19   @Input() text: string;
20   @Input() index: number;
21   @Output() onTodoItemChecked = new EventEmitter<number>();
22   @Output() onTodoItemRemove = new EventEmitter<number>();
23
24   handleOnRemove = () => this.onTodoItemRemove.emit(this.index);
25   handleOnChecked = () => this.onTodoItemChecked.emit(this.index);
26 }
27
28
29
30
31 // @jsx h
32 import { props } from "skatejs/dist/esnext";
33 import { h } from "preact";
34 import { Component } from "@util";
35
36 export default class extends Component {
37   static events = ["check", "remove"];
38   static props = {
39     checked: props.boolean,
40     index: props.number
41   };
42
43   handleCheck = e => {
44     this.onCheck({ index: this.index, value: e.target.checked });
45   };
46   handleRemove = () => {
47     this.onRemove({ index: this.index });
48   };
49
50   render({ checked, handleCheck, handleRemove }) {
51     return (
52       <div>
53         <style>`-`</style>
54         <li class="item" checked="checked" style="background-color: ${checked ? "#ccc" : "#fff"}">
55           <input type="checkbox" checked="checked" onChange={handleCheck}>
56           <label>${this.props.text}</label>
57           <slot />
58           <label><button onClick={handleRemove}>x</button></label>
59         </li>
60       </div>
61     );
62   }
63 }
```

StencilJS

```
import { Component, Prop, Event, EventEmitter } from '@stencil/core';

@Component({
  tag: 'todo-item',
  styleUrl: 'todo-item.scss',
  shadow: true,
})
export class TodoItem {
  @Prop() checked: boolean;
  @Prop() text: string;
  @Prop() index: number;
  @Event() onTodoItemChecked: EventEmitter;
  @Event() onTodoItemRemove: EventEmitter;

  handleOnRemove = () => this.onTodoItemRemove.emit(this.index);
  handleChecked = () => this.onTodoItemChecked.emit(this.index);

  render() {
    return (
      <li class={this.checked ? 'completed' : ''}>
        <input type="checkbox" checked={this.checked} onChange={this.handleOnChecked}/>
        <label>{this.text}</label>
        <button onClick={this.handleOnRemove}>x</button>
      </li>
    );
  }
}

SkateJS + lit-html

import { props } from "skatejs/dist/esnext";
import { html } from "lit-html/lib/lit-extended";
import { Component } from "./util";

export default class extends Component {
  static events = ["check", "remove"];
  static props = {
    checked: props.boolean,
    index: props.number
  };
  handleCheck = e => {
    this.onCheck({ index: this.index, value: e.target.checked });
  };
  handleRemove = () => {
    this.onRemove({ index: this.index });
  };
  render({ checked, handleCheck, handleRemove }) {
    return html`
      <li class="${checked ? 'completed' : ''}">
        <input type="checkbox" checked="${checked}" on-change="${handleCheck}" />
        <label>
          <slot></slot>
        </label>
        <button on-click="${handleRemove}">x</button>
      </li>
    `;
  }
}
```

SkateJS + Preact

SkateJS + lit-html

```
import {Component, Prop} from '@stencil/core';

@Component({
  tag: 'my-first-component',
  styleUrl: 'my-first-component.scss'
})
export class MyComponent {
  // Indicate that name should be
  // a public property on the component
  @Prop() name: string;

  render() {
    // JSX
    return (<p>My name is {this.name}</p>);
  }
}
```

Pour démarrer

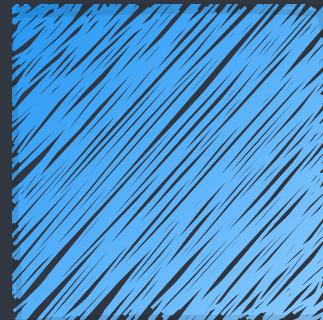


```
$ npm init stencil
```



```
? Pick a starter > - Use arrow-keys. Return to submit.  
> ionic-pwa      Everything you need to build fast, production ready PWAs  
    app           Minimal starter for building a Stencil app or website  
    component     Collection of web components that can be used anywhere
```

Lit-Elements



<https://lit-element.polymer-project.org/> <[https://lit-](https://lit-element.polymer-project.org/)
[element.polymer-project.org/](https://lit-element.polymer-project.org/)>



Projet **Open Source**, ➔ BSD 3-Clause License

<<https://github.com/Polymer/lit-element/blob/master/LICENSE>>

Créer par l'équipe **Polymer Team** en 2017

2.0k ⭐ sur github

Utilise la bibliothèque de template **lit-html**

<https://lit-html.polymer-project.org/> <<https://lit-html.polymer-project.org/>>

4.4k ⭐ sur github

Basé sur les **templates HTML**

Avec les ➔ *Template literals* <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Template_literals> de ES2015

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

// A lit-html template uses
// the `html` template tag:
const sayHello = (name) =>
  html`<h1>Hello ${name}</h1>`;

// It's rendered with the `render()` function:
render(sayHello('World'), document.body);

// And re-renders only update the
// data that changed, without VDOM diffing!
render(sayHello('Everyone'), document.body);
```



```
import { LitElement, html, css } from 'https://unpkg.com/lit-element/lit-element';

class MyElement extends LitElement {
    static get properties() {
        return {
            mood: { type: String }
        }
    }
    static get styles() {
        return css` .mood { color: green; } `;
    }
    render() {
        return html` Web Components are
            <span class="mood">${this.mood}</span>! `;
    }
}

customElements.define('my-element', MyElement);
```

```
import { /* ... */ } from 'lit-element';
import { html, css } from 'lit';

@customElement('my-element')
class MyElement extends LitElement {

    @property({ type: String }) mood;

    static get styles() {
        return css` .mood { color: green; } `;
    }

    render() {
        return html` Web Components are
            <span class="mood">${this.mood}</span>! `;
    }
}
```

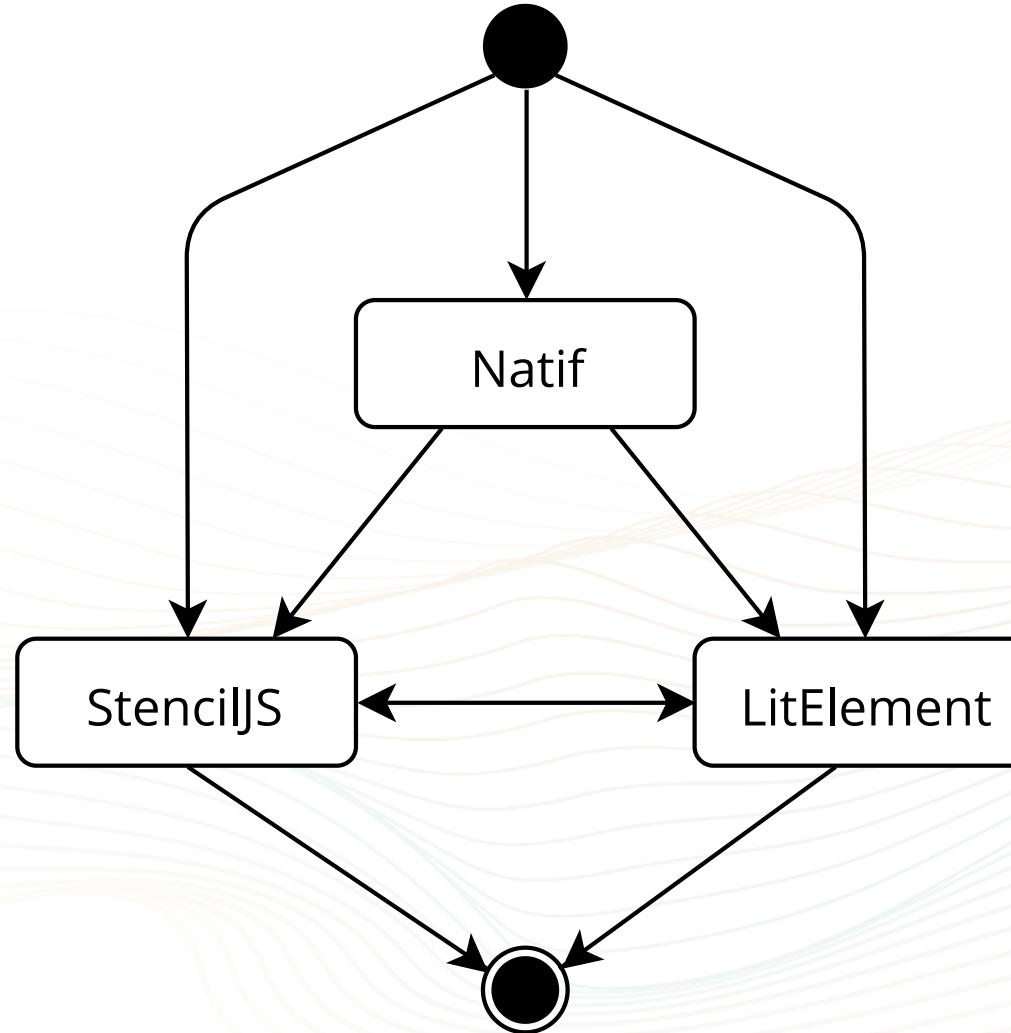
✓ Chrome, Safari, Opera, Firefox

 polyfills <<https://github.com/webcomponents/webcomponentsjs>> pour Edge et
IE 11

Workshop

@julienrenaux @ilaborie #DevoxxFR #webcomponents

DEVOXX
France



<http://bit.ly/devoxx-webc> <<http://bit.ly/devoxx-webc>>

Conclusion

@julienrenaux @ilaborie #DevoxxFR #webcomponents

DEVOXX
France

Les limites

Support des navigateurs

Thème

Ce n'est pas un framework

➤ Beyond the polyfills: how Web Components affect

us today? <<https://dev.to/webpadawan/beyond-the-polyfills-how-web-components-affect-us-today-3j0a>>

et ➤ The journey of Web Components: wrong ways, lacking parts and promising paths

<<https://dev.to/webpadawan/the-journey-of-web-components-wrong-ways-lacking-parts-and-promising-paths-1d5a>>

Les alternatives modernes

➡ SkateJS <<https://skatejs.netlify.com/>>

➡ Svelte <<https://svelte.technology/>>

➡ Slim.js <<http://slimjs.com>>

➡ Atomico <<https://github.com/atomicojs/core>>

...

Le future



Pseudo elements avec ➡ `::part` and `::theme`, an
`::explainer` <<https://meowni.ca/posts/part-theme-explainer/>>, ➡ CSS Shadow
Parts <<https://www.w3.org/TR/css-shadow-parts-1/>>

➡ Scoped Custom Element Registries

<<https://github.com/w3c/webcomponents/issues/716>>

SSR pour lit-html et litElement

...

Des liens

➤ Web Components sur MDN <https://developer.mozilla.org/en-US/docs/Web/Web_Components>

➤ Série d'articles sur *css-tricks* <<https://css-tricks.com/an-introduction-to-web-components/>>

➤ Web Components Todo <<https://wc-todo.firebaseio.com/>>

➤ Web Components specifications

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

➤ A curated list of awesome Web Components resources. <<https://github.com/mateusortiz/webcomponents-the-right-way>>

➤ A curated list of awesome lit-html resources.

<<https://github.com/web-padawan/awesome-lit-html>>

Fin

Merci

Pensez à nous faire des retours (votez !)