

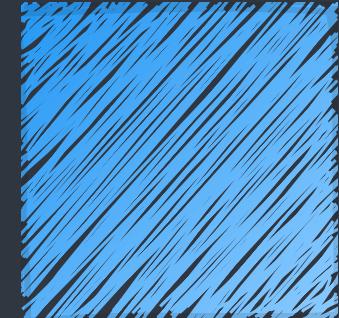
# Web Components:



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>>

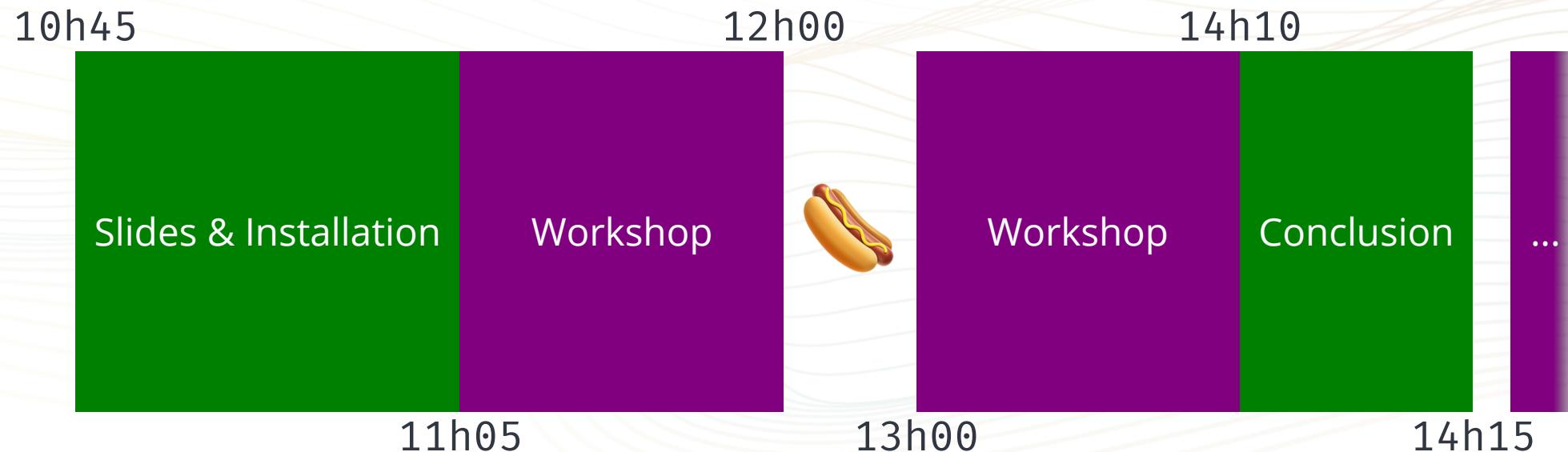
[igor@monkeypatch.io](mailto:igor@monkeypatch.io)

<<mailto:igor@monkeypatch.io>>





# Roadmap

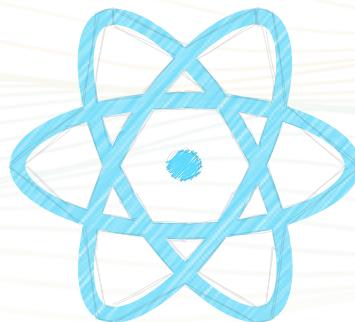


# Instructions

TODO: Installation instructions, Wifi

# 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

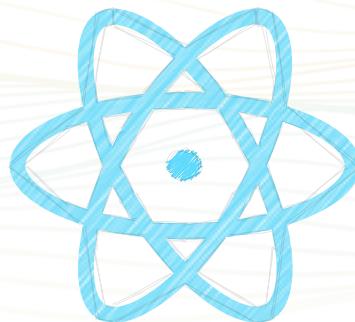


**Développer une application en JS n'est  
plus simple...**

# 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: 1,124 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 "1,755 people are talking about this" followed by a right-pointing arrow.

# **...Interopérabilité ne vient plus gratuitement...**



React

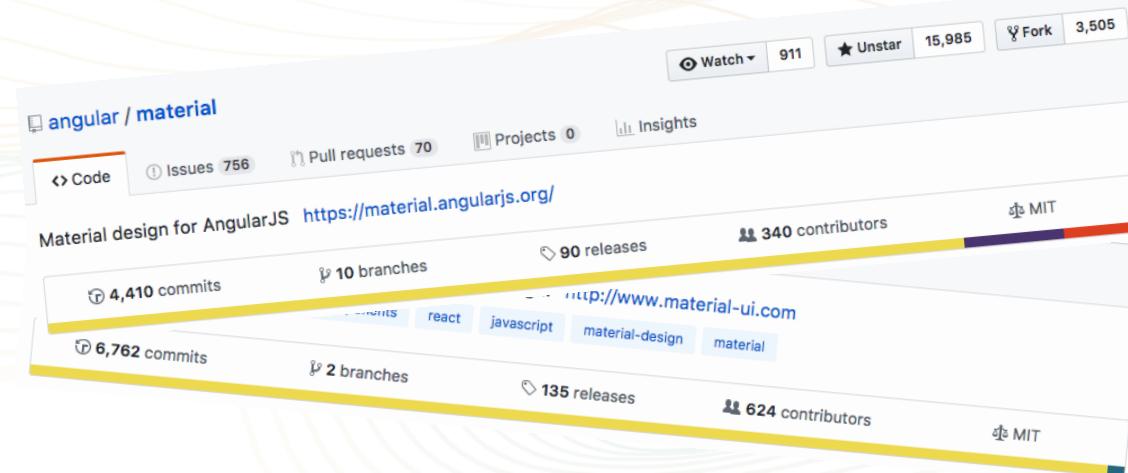


Vue.js



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

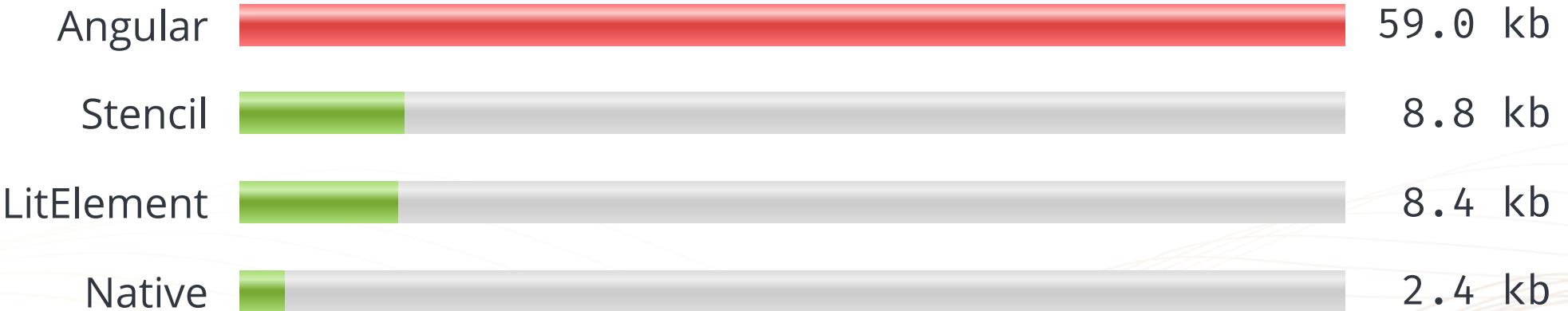
## Material Web Components

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

DO IT! DO IT! DO IT!

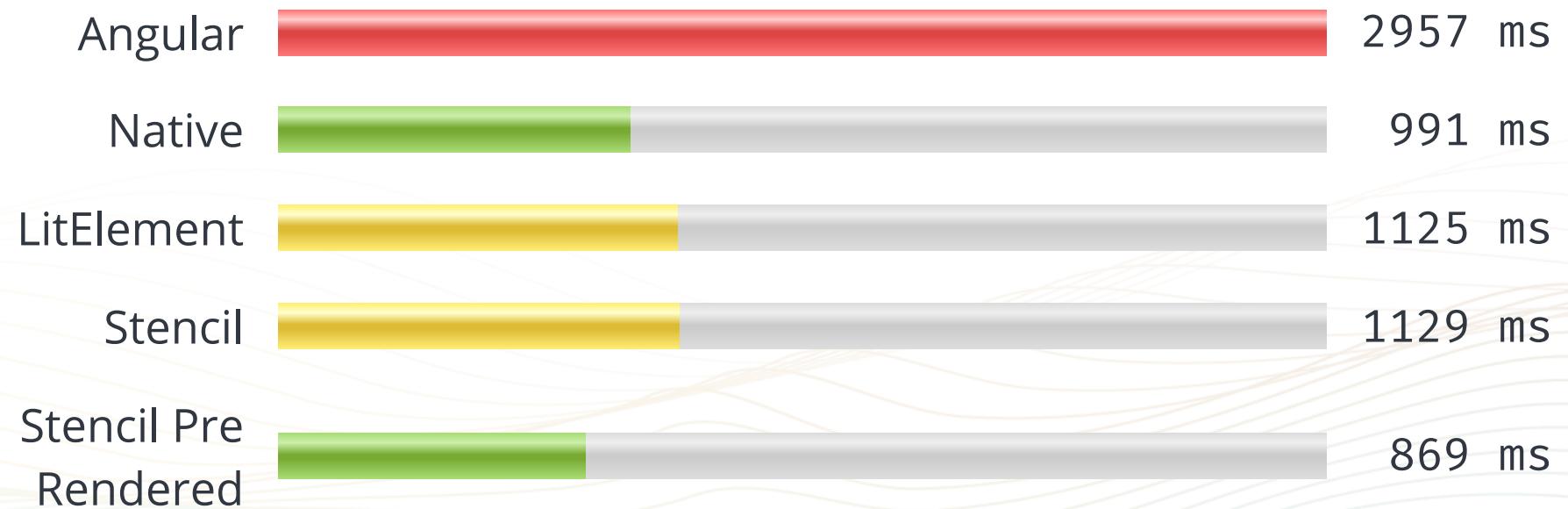


**Toute cette complexité doit  
s'arrêter avec les Web  
Components**



## Size matters (Gzipped)

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



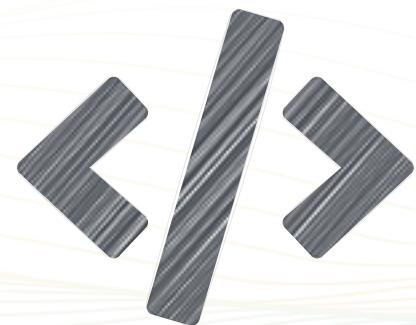
**Le temps ça compte (FMP 3G 📱 en ms)**

😢 Angular est 3 fois plus lent

# Web Components



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



Custom  
Elements



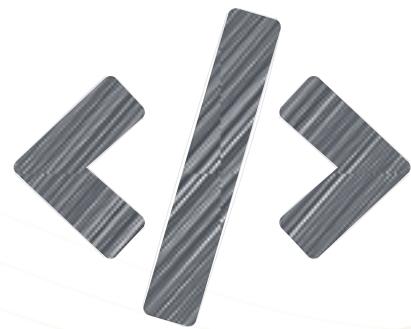
Shadow DOM



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);
```

# 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+



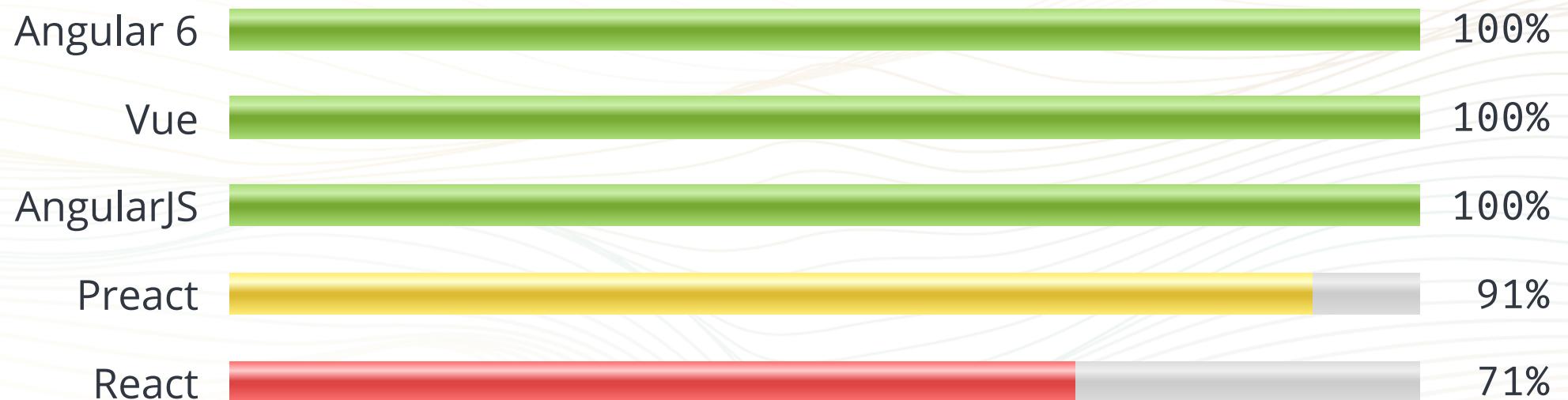
9+

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

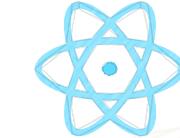
5.1k ⭐ sur github



**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     <style>
11       <li class="item" [isCompleted={checked}]>
12         <input type="checkbox" value="{{checked}}" checked="{{checked:change}}"/>
13         <label>{{text}}</label>
14         <button class="destroy">x</button>
15       </li>
16     </style>
17     <li class="item">
18       <input type="checkbox">
19       <label>{{text}}</label>
20       <button class="destroy">x</button>
21     </li>
22   }
23   static get observedAttributes() {
24     return ['text'];
25   }
26   attributeChangedCallback(name, oldValue, newValue) {
27     if (name === 'text') {
28       this._text = newValue;
29     }
30   }
31   set index(value) {
32     this._index = value;
33   }
34   get index() {
35     return this._index;
36   }
37   set checked(value) {
38     this._checked = Boolean(value);
39   }
40   get checked() {
41     return this.hasAttribute('checked');
42   }
43   _render() {
44     if (!this.$item) return;
45     this.$text.textContent = this._text;
46     if (this._checked) {
47       this.$item.classList.add('completed');
48       this.$checkbox.setAttribute('checked', '');
49     } else {
50       this.$item.classList.remove('completed');
51       this.$checkbox.removeAttribute('checked');
52     }
53   }
54 }
55 window.customElements.define('todo-item', TodoItem);
```

## Polymer 2

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

## StencilJS

```
1 import { Component, Prop, Event, EventEmitter } from '@stencil/core';
2
3 @Component({
4   tag: 'todo-item',
5   styleUrl: 'todo-item.scss',
6   shadow: true,
7 })
8 export class TodoItem {
9   @Prop() checked: boolean;
10  @Prop() text: string;
11  @Prop() index: number;
12  @Event() onTodoItemChecked: EventEmitter;
13  @Event() onTodoItemRemove: EventEmitter;
14
15  handleOnRemove = () => this.onTodoItemRemove.emit(this.index);
16  handleOnChecked = () => this.onTodoItemChecked.emit(this.index);
17
18  render() {
19    return (
20      <li class="item">{{checked ? 'completed' : ''}}>
21        <input type="checkbox" checked={{checked}} onChange="this.handleOnChecked()" />
22        <label>{{text}}</label>
23        <button onClick="this.handleOnRemove()">>x</button>
24      </li>
25    );
26  }
27}
```

## SkateJS + Preact

```
1 // @sx h
2 import { props } from "skatejs/dist/esnext";
3 import { h } from "preact";
4 import { Component } from "./util";
5
6 export default class extends Component {
7   static events = ["check", "remove"];
8   static props = {
9     checked: props.boolean,
10    index: props.number
11  };
12
13  handleCheck = e => {
14    this.onCheck({ index: this.index, value: e.target.checked });
15  };
16  handleRemove = () => {
17    this.onRemove({ index: this.index });
18  };
19
20  render({ checked, handleCheck, handleRemove }) {
21    return (
22      <div>
23        <style>`<br>`</style>
24        <li class="${checked ? "completed" : ""}>
25          <input type="checkbox" checked={{checked}} onChange={handleCheck} />
26          <label>
27            <slot />
28          </label>
29          <button onClick={handleRemove}>x</button>
30        </li>
31      </div>
32    );
33  }
34}
```

## SkateJS + lit-html

```
1 import { props } from "skatejs/dist/esnext";
2 import { html } from "lit-html/lib/lit-extended";
3 import { Component } from "./util";
4
5 export default class extends Component {
6   static events = ["check", "remove"];
7   static props = {
8     checked: props.boolean,
9     index: props.number
10  };
11
12  handleCheck = e => {
13    this.onCheck({ index: this.index, value: e.target.checked });
14  };
15  handleRemove = () => {
16    this.onRemove({ index: this.index });
17  };
18
19  render({ checked, handleCheck, handleRemove }) {
20    return html`
21      <div>
22        <style>`<br>`</style>
23        <li class="${checked ? "completed" : ""}>
24          <input type="checkbox" checked={{checked}} on-change="${handleCheck}" />
25          <label>
26            <slot />
27          </label>
28          <button on-click="${handleRemove}">x</button>
29        </li>
30      </div>
31    `;
32  }
33}
```

# Pour démarrer



```
$ npm init stencil
```



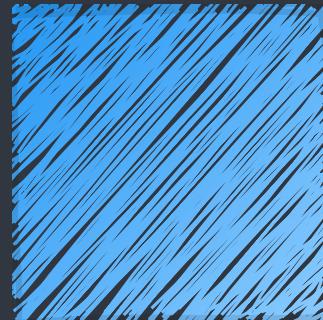
```
? Pick a starter > - Use arrow-keys. Return to submit.  
> ionic-pwa      Everything you need to build fast, production ready PWAs  
    app           Minimal starter for building a Stencil app or website  
    component     Collection of web components that can be used anywhere
```

```
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>);
  }
}
```

# 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

1.8k ⭐ sur github

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

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

Basé sur les **templates HTML**

Avec les ➡ *Template literals* <[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/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>! `;
    }
}
```

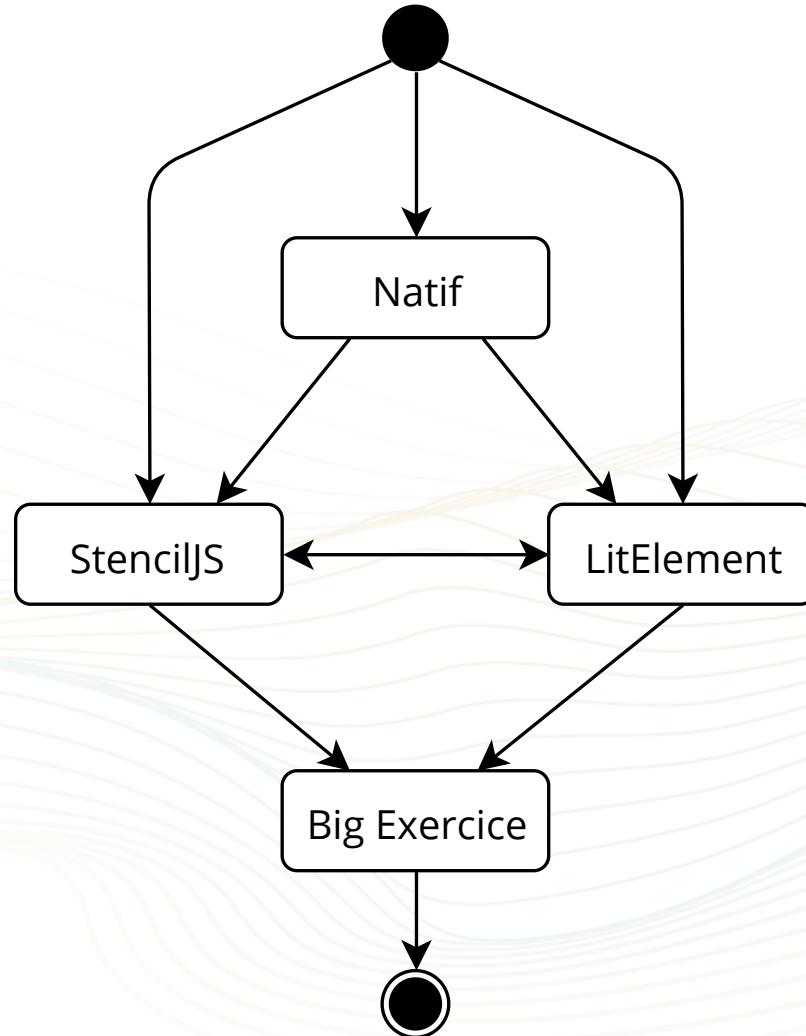
✓ Chrome, Safari, Opera, Firefox

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

# Workshop

@julienrenaux @ilaborie #DevoxxFR #webcomponents

DEVOXX  
France



# Conclusion

@julienrenaux @ilaborie #DevoxxFR #webcomponents

DEVOXX  
France

Construire une application

Style avec un thème

Support des navigateurs

Utiliser un gestionnaire d'état externe

Utilisez les *custom properties* CSS

Utiliser le polyfill ou Electron

TODO: ...

# Alternatives modernes

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

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

➤ Slim.js <<http://slimjs.com>>

...

# Fin

**Merci**

Pensez à nous faire des retours (votez !)