

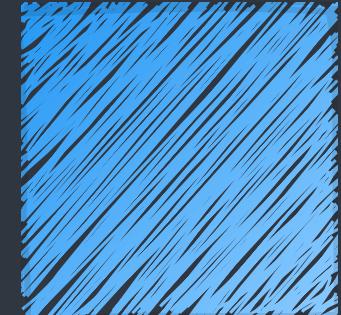
Web Components:



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>>

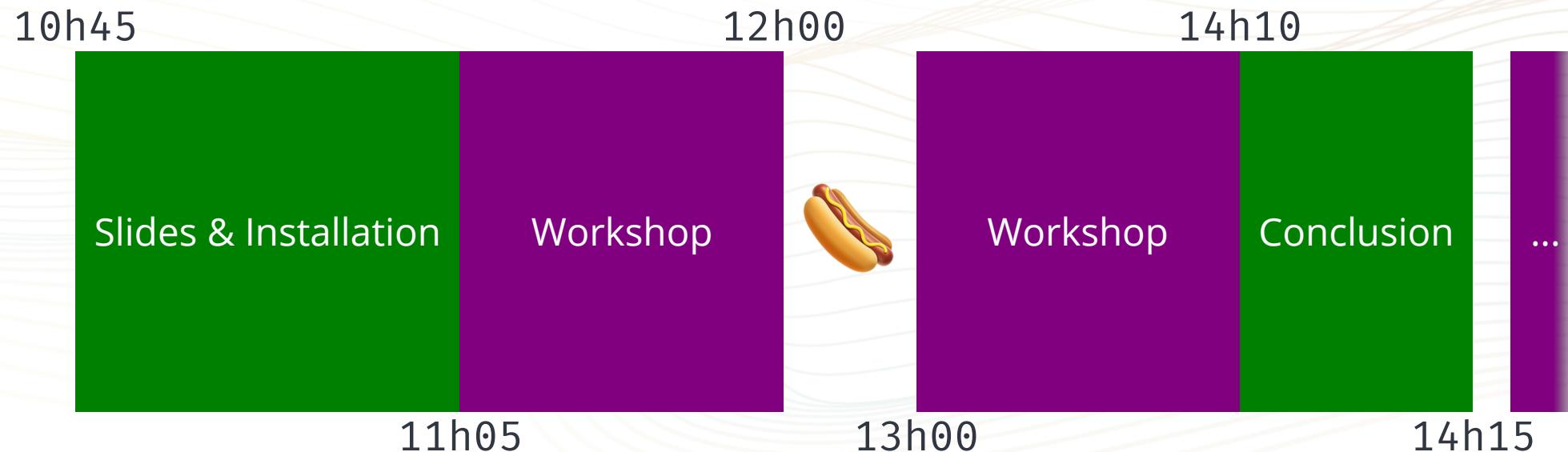
igor@monkeypatch.io

<<mailto:igor@monkeypatch.io>>





Roadmap

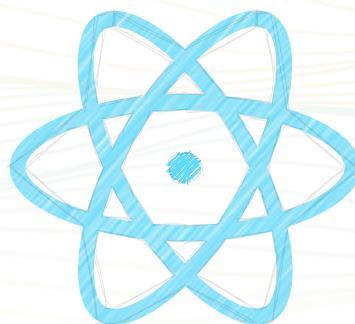


Instructions

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 ...



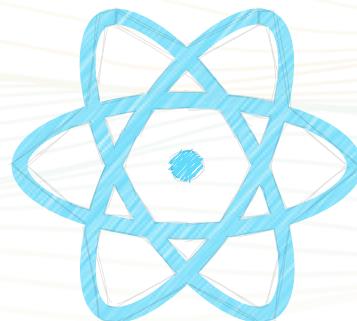
I Am Devloper
@jamdevloper

I think I've had milk last longer than some JavaScript frameworks.

1,124 2:22 PM - Dec 4, 2014

1,755 people are talking about this >

...Interopérabilité ne vient plus gratuitement...



React

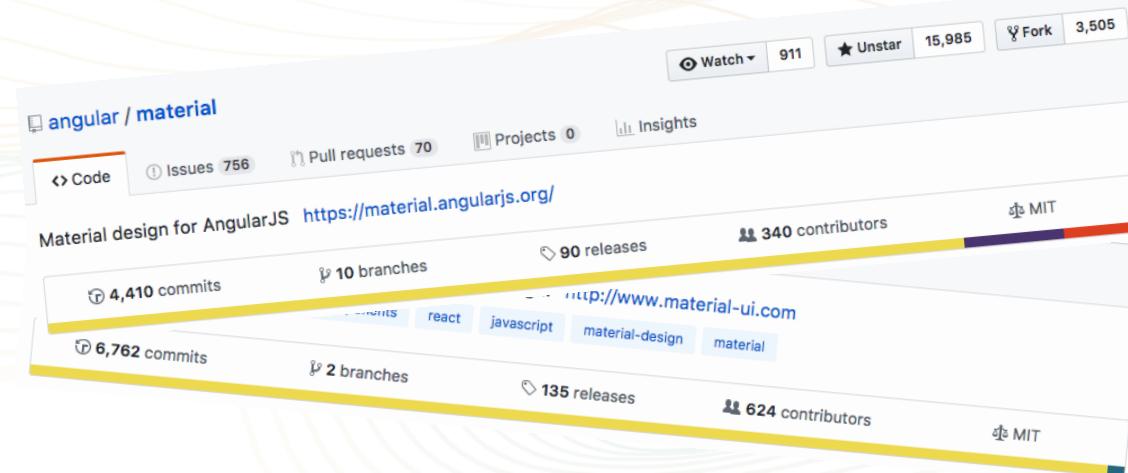


Vue.js



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

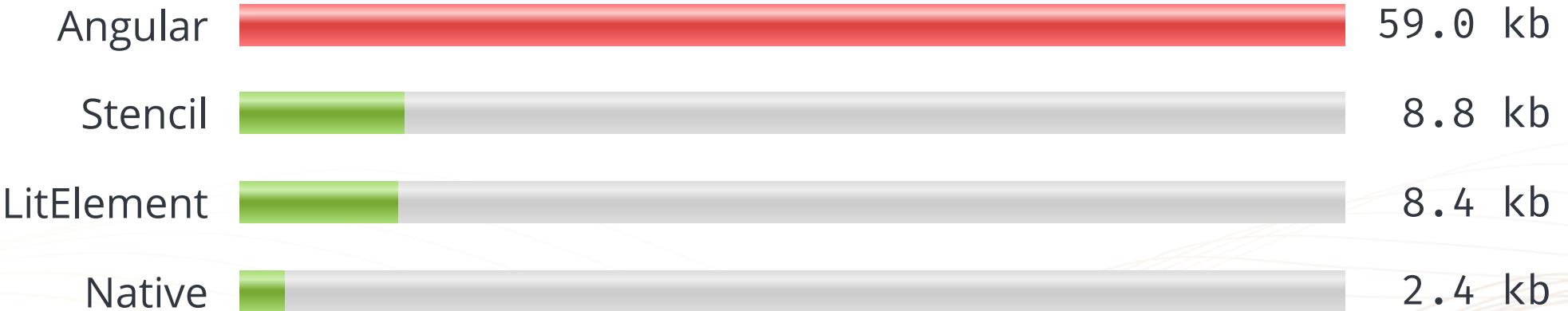
Material Web Components

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

DO IT! DO IT! DO IT!

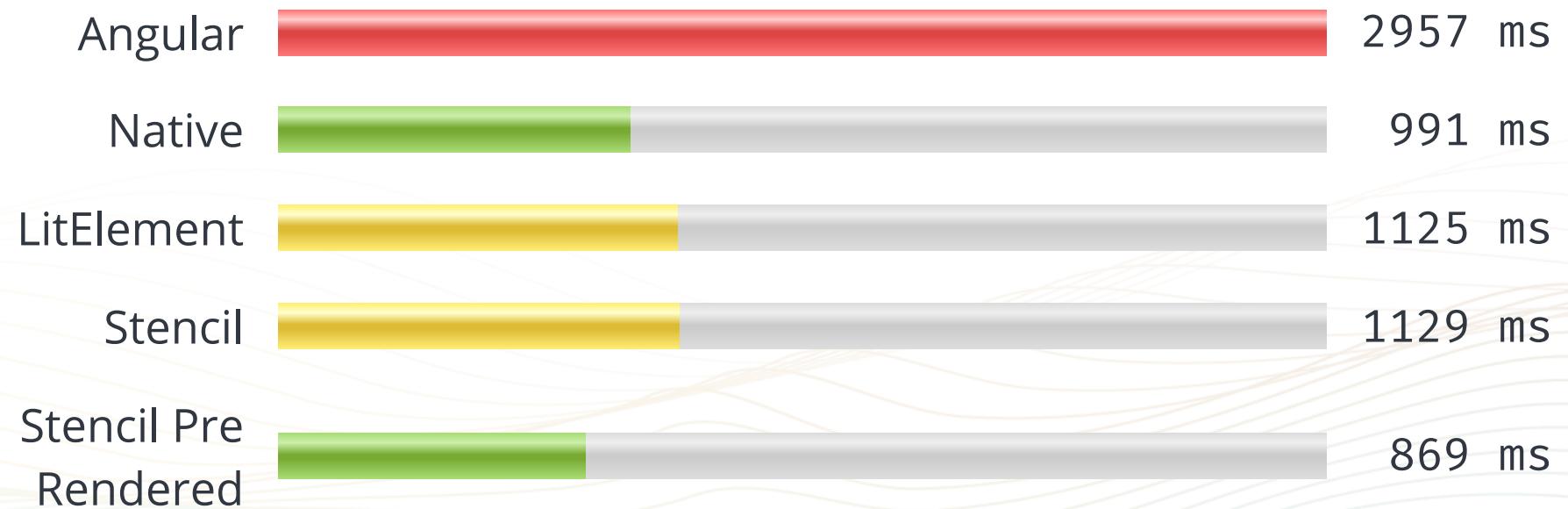


**Toute cette complexité doit
s'arrêter avec les Web
Components**



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

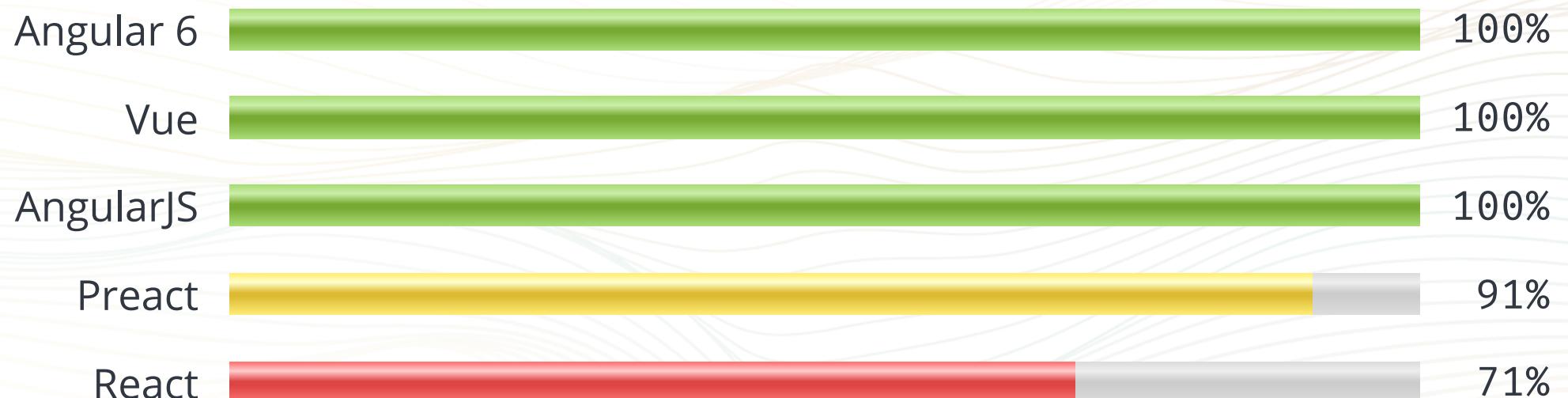


	Internet Explorer 11+	Edge	Firefox	Chrome	Opera 9+
Custom Elements (V1)	✓	✓	✓	✓	✓
Shadow DOM (V1)	✓	✓	✓	✓	✓
HTML templates	✓	✓	✓	✓	✓

<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             `<input type="checkbox">` +
12             `<label>${this._text}</label>` +
13             `<button class="destroy"><x></button>` +
14         `</style>` +
15         `<li class="item">` +
16             `<input type="checkbox" checked="" ${this._checked ? 'checked' : ''}>` +
17             `<label>${this._text}</label>` +
18             `<button class="destroy"><x></button>` +
19         `</li>`;
20         this._site = this._root.querySelector('.item');
21         this._removeButton = this._root.querySelector('.destroy');
22         this._text = this._root.querySelector('.text');
23         this._checkbox = this._root.querySelector('input');
24         this._removeButton.addEventListener('click', (e) => {
25             e.preventDefault();
26             this.dispatchEvent(new CustomEvent('onRemove', { index: this._index }));
27         });
28         this._checkbox.addEventListener('click', (e) => {
29             e.preventDefault();
30             this.dispatchEvent(new CustomEvent('onToggle', { index: this._index }));
31         });
32         this._render();
33     }
34     disconnectedCallback() []
35     static get observedAttributes() {
36         return ['text'];
37     }
38     attributeChangedCallback(name, oldValue, newValue) {
39         if (name === 'text') {
40             this._text = newValue;
41         }
42         setIndex(value) {
43             this._index = value;
44         }
45         getIndex() {
46             return this._index;
47         }
48         set checked(value) {
49             this._checked = Boolean(value);
50         }
51         get checked() {
52             return this.hasAttribute('checked');
53         }
54         _render() {
55             if (!this._item) return;
56             this._text.textContent = this._text;
57             if (!this._checked) {
58                 this._item.classList.add('completed');
59                 this._checkbox.setAttribute('checked', '');
60             } else {
61                 this._item.classList.remove('completed');
62                 this._checkbox.removeAttribute('checked');
63             }
64         }
65     }
66 }
67 
```

Polymer 2

```
1 <link rel="import" href="../../bower_components/polymer/polymer.html">
2 <dom-module id="todo-item">
3   <template>
4     <style>
5       <!-->
6       <!-->
7       <li class="{$item [isCompleted (checked)]}">
8         <input type="checkbox" value="{$(checked)}" checked="{$(checked)}" />
9         <label>{$text}</label>
10        <button class="destroy" on-click="handleOnRemove">x</button>
11      </li>
12    </template>
13    <script>
14      class TodoItem extends Polymer.Element {
15        static get is() { return 'todo-item'; }
16        static get properties() {
17          return {
18            checked: { type: Boolean, value: false },
19            index: { type: Number, },
20            text: { type: String, value: '' }
21          };
22        }
23        handleOnRemove(e) {
24          this.dispatchEvent(new CustomEvent('remove', { detail: e }));
25        }
26        handleOnChecked(e) {
27          this.dispatchEvent(new CustomEvent('toggle', { detail: e }));
28        }
29        isCompleted(completed) {
30          return completed ? 'completed' : '';
31        }
32      }
33      window.customElements.define(TodoItem.is, TodoItem);
34    </script>
35  </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',
  styleUrls: ['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);
  handleOnChecked = () => 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`
```

SkateJS + Preact

SkateJS + lit-html

Pour démarrer



```
$ npm init stencil
```



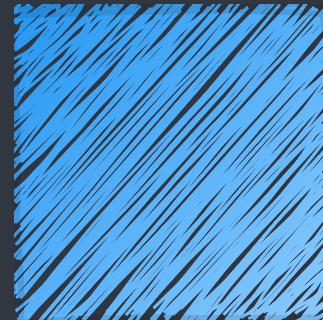
```
? Pick a starter > - Use arrow-keys. Return to submit.  
> ionic-pwa      Everything you need to build fast, production ready PWAs  
    app           Minimal starter for building a Stencil app or website  
    component     Collection of web components that can be used anywhere
```

```
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> 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>! `;
    }
}
```

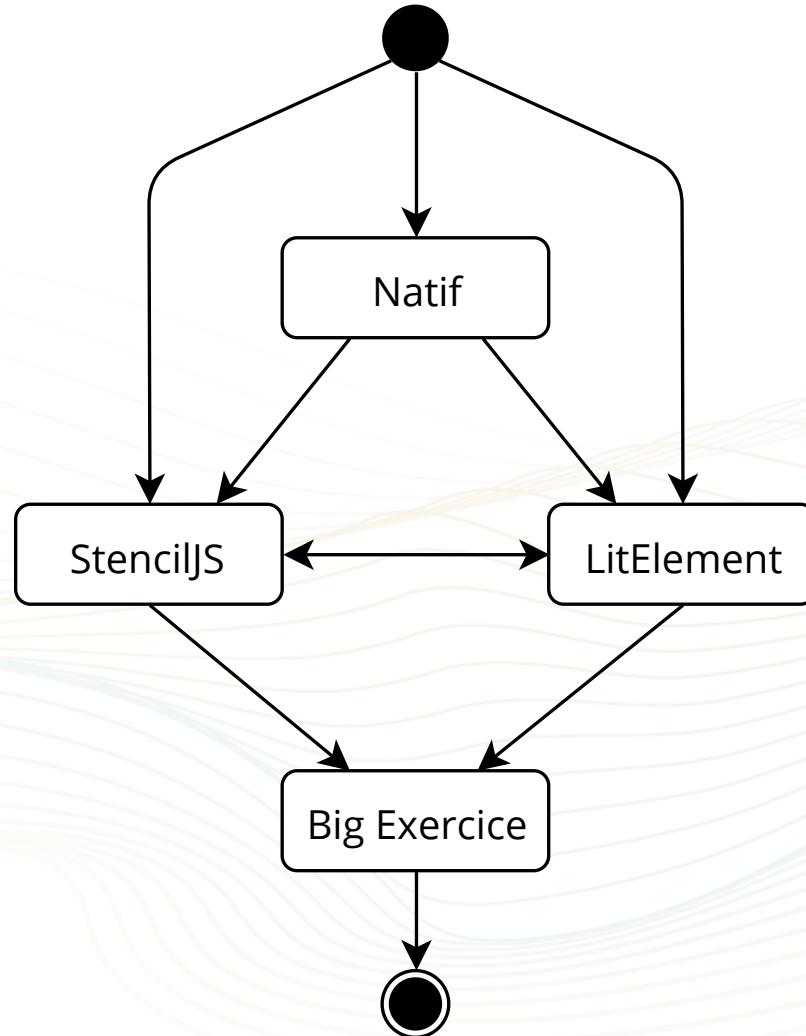
✓ Chrome, Safari, Opera, Firefox

 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 !)