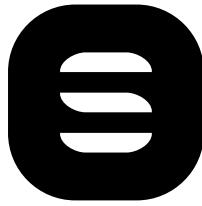


- [Web Components](#) * [Natif](#) * [StencilJS](#) * [LitElement](#)
- [Speakers](#)
- [Roadmap](#)
- [Instructions](#)
- [I want to build a Web app in 2019](#)
- [Choose Framework](#)
- [Choose Style](#)
- [Choose JavaScript Transpiler](#)
- [Development Not Easy](#)
- [Industry Moving too Fast](#)
- [Interoperability](#)
- [Reinventing](#)
- [Reinventing 2](#)
- [Solution](#)
- [Size matters](#)
- [Time matters](#)
- [Web Components Web Components](#)
- [History](#)
- [Components](#)
- [Custom Elements](#)
- [Shadow DOM](#)
- [HTML templates](#)
- [Natif code](#)
- [Browser support](#)
- [Polyfill support](#)
- [Framework Interoperability](#)
- [StencilJS StencilJS](#)
- [What](#)
- [Not a framework](#)
- [Compiler](#)
- [StencilJS is a set of great tools](#)
- [StencilJS works everywhere](#)
- [Stencil is concise](#)
- [Pour démarrer](#)
- [stencil code](#)
- [Lit-Elements LitElement](#)
- [What-Lit](#)
- [Templating](#)
- [lit-html](#)
- [lit-elements in JS](#)
- [lit-elements in TS](#)
- [lit-elements Support](#)
- [Workshop](#)
- [Exercises](#)
- [Conclusion](#)
- [Issues](#)
- [Alternatives modernes](#)
- [Fin](#)

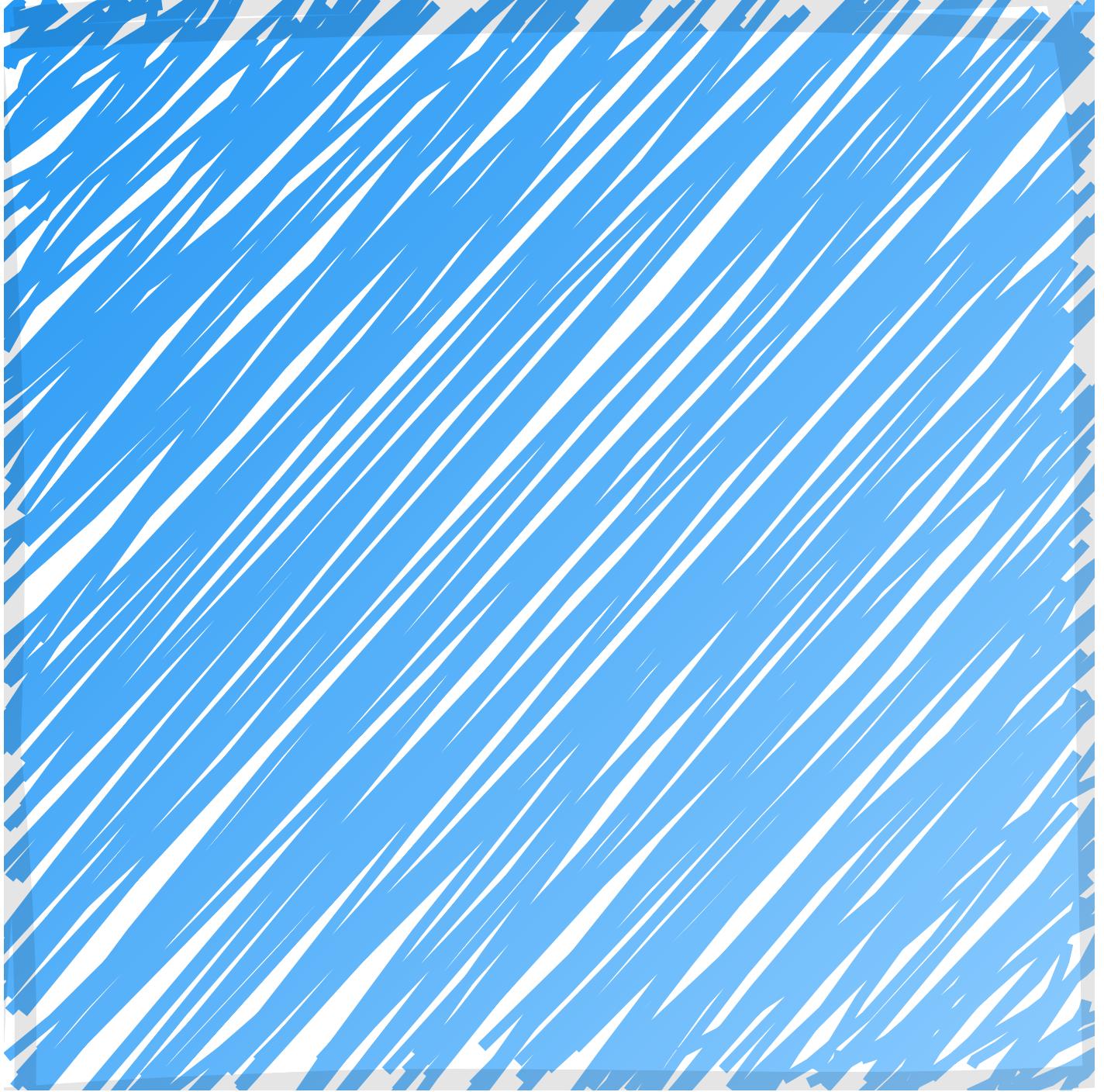
Web Components



Natif

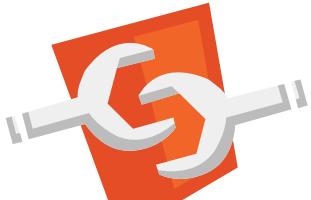


StencilJS

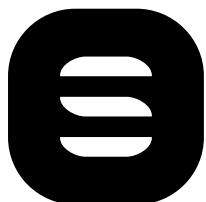


LitElement

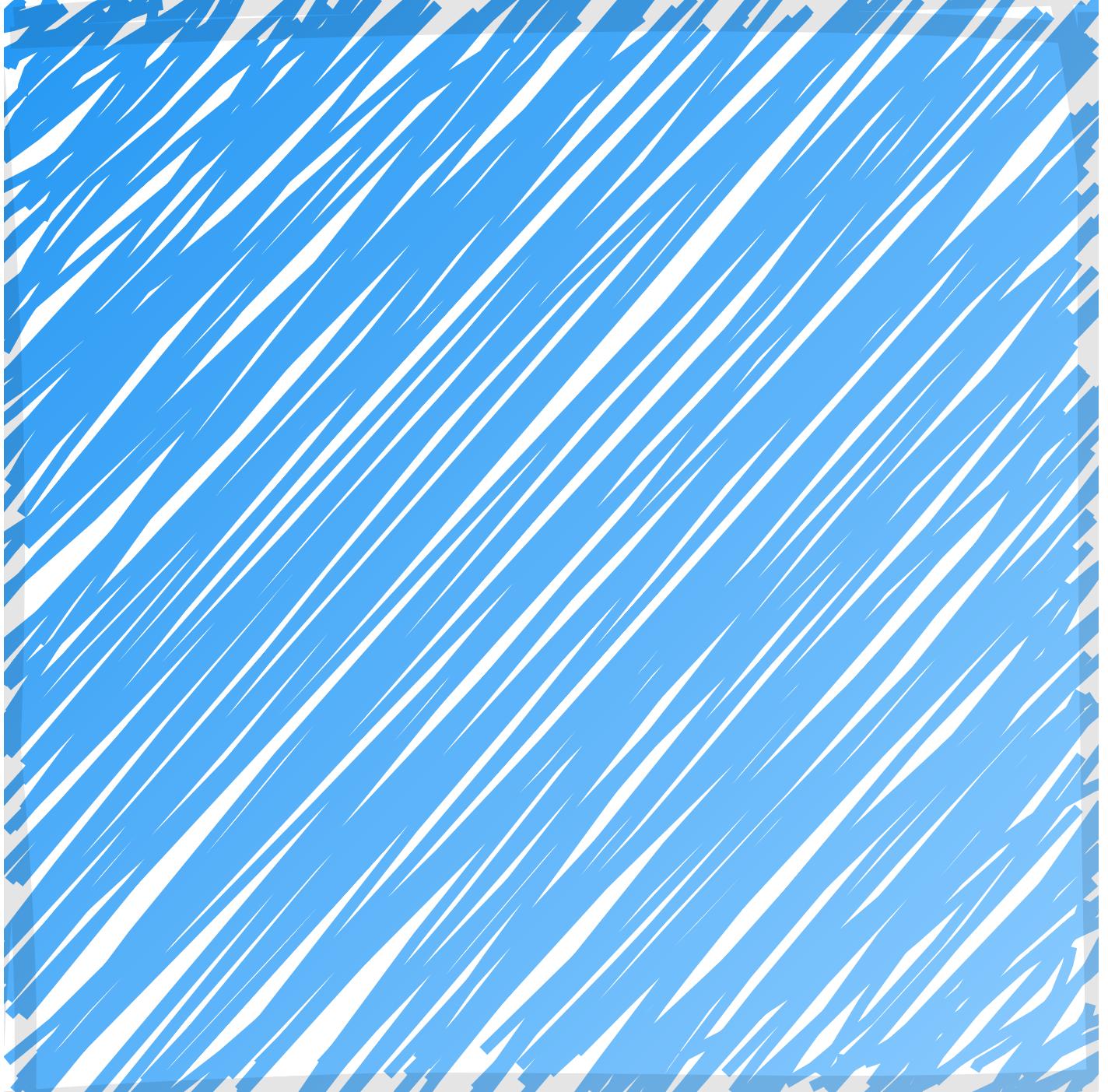
Web Components



Natif



StencilJS

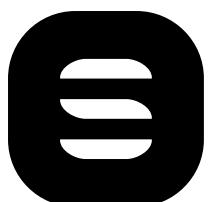


LitElement

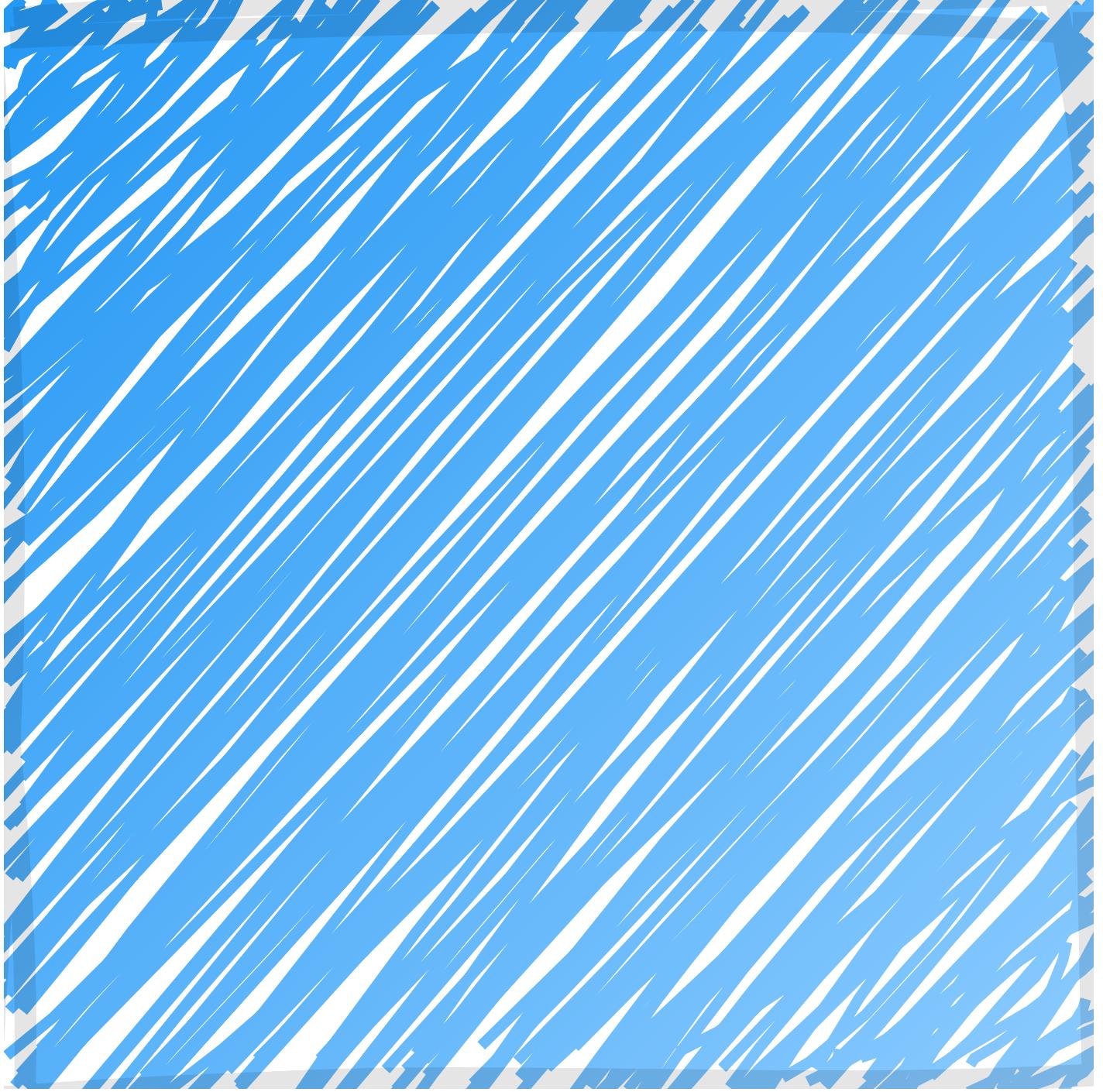
Web Components



Natif



StencilJS



LitElement

Speakers



Julien Renaux

GDE Web, Freelancer

[@julienrenaux <https://julienrenaux.fr/>](https://julienrenaux.fr/)



Igor Laborie

Expert Web & Java

@ilaborie igor@monkeypatch.io



Toptal

MonkeyPatch

Roadmap

- Slides & Installation
- Workshop
- 🍫
- Workshop
- Conclusion
- ...

Instructions

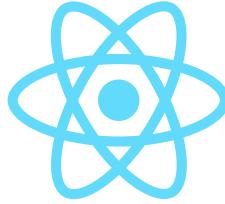
Installation instructions, Wifi

I want to build a Web app in 2019

Développons une application Web en 2019

Choose Framework

Commençons par choisir un Framework



• React



• Vue.js



• Angular

Choose Style

Puis comment allons-nous écrire le style

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

Choose JavaScript Transpiler

Puis construisons notre application

- Webpack
- Parcel.js
- Rollup.js
- Bazel

Development Not Easy

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

Industry Moving too Fast

Ça va trop vite ...



I Am Developer
@iamdeveloper



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

1,125 1:22 PM - Dec 4, 2014

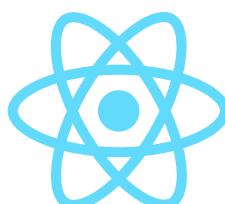


1,755 people are talking about this



Interoperability

...Interopérabilité ne vient plus gratuitement...



React



Vue.js



Angular

Reinventing

...et on réinvente sans arrêt la roue !

[google / material-design-lite](#) Watch ▾ 1,381 Unstar 29,385 Fork 5,095

Code Issues 297 Pull requests 0 Wiki Insights

Material Design Components in HTML/CSS/JS <http://getmdl.io>

[material-design](#) [material-design-lite](#) [mdl](#) [material-components](#) [material](#)

2,872 commits 73 branches 26 releases 174 contributors Apache-2.0

Material 1

[posva / vue-mdc](#) Watch ▾ 40 Star 1,132 Fork 102

Code Issues 11 Pull requests 7 Projects 0 Wiki Insights

Material web components for Vue.js

[mdl](#) [vue](#) [material](#) [design](#) [lite](#) [google](#) [mdc](#) [components](#) [web](#)

213 commits 19 branches 67 releases 4 contributors MIT

Material 2

[mlaursen / react-md](#) Watch ▾ 60 Star 1,543 Fork 191

Code Issues 118 Pull requests 0 Projects 0 Wiki Insights

React material design <https://react-md.mlaursen.com>

[react](#) [material-design](#) [react-md](#)

2,415 commits 13 branches 71 releases 41 contributors MIT

Material 3

[zdhxiong / mdui](#) Watch ▾ 84 Star 1,810 Fork 177

Code Issues 19 Pull requests 0 Projects 0 Wiki Insights

MDUI 是一个基于 Material Design 的前端框架。 <https://www.mdui.org>

[css](#) [javascript](#) [material](#) [google](#)

442 commits 1 branch 6 releases 6 contributors MIT

Material 4

[Code](#)[Issues 48](#)[Pull requests 1](#)[Projects 0](#)[Wiki](#)[Insights](#)

The first responsive front-end framework based on Angular & Google Material Design specifications
<https://ui.lumapps.com>

1,358 commits

27 branches

273 releases

34 contributors

MIT

Material 5

[Code](#)[Issues 243](#)[Pull requests 38](#)[Projects 0](#)[Insights](#)

Modular and customizable Material Design UI components for the web <https://material.io/components/>

[material-components](#) [material-design](#) [web](#) [css](#) [javascript](#)

3,919 commits

88 branches

547 releases

265 contributors

Apache-2.0

Material 6

[Code](#)[Issues 11](#)[Pull requests 7](#)[Projects 0](#)[Wiki](#)[Insights](#)

Material web components for Vue.js

[mdl](#) [vue](#) [material](#) [design](#) [lite](#) [google](#) [mdc](#) [components](#) [web](#)

213 commits

19 branches

67 releases

4 contributors

MIT

Material 7

[Code](#)[Issues 271](#)[Pull requests 119](#)[Projects 1](#)[Wiki](#)[Insights](#)

Materialize, a CSS Framework based on Material Design <http://materializetcss.com>

[design](#) [css](#) [framework](#) [javascript](#) [material-design](#) [material](#)

3,617 commits

21 branches

39 releases

238 contributors

MIT

Material 8

[Code](#)[Issues 138](#)[Pull requests 8](#)[Wiki](#)[Insights](#)

React components that implement Google's Material Design. <http://www.material-ui.com>

[material-ui](#) [google-material](#) [react-components](#) [react](#) [javascript](#) [material-design](#) [material](#)

6,762 commits

2 branches

135 releases

624 contributors

MIT

Material 9

[Code](#)[Issues 756](#)[Pull requests 70](#)[Projects 0](#)[Insights](#)

Material design for AngularJS <https://material.angularjs.org/>

4,410 commits

10 branches

90 releases

340 contributors

MIT

- Material 10

Reinventing 2

Meme Luci Do It: add another one in WebComponent LitElement...

Solution

Toute cette complexité doit s'arrêter avec les Web Components

Size matters

Angular	
59.0 kb	
Stencil	
8.8 kb	
LitElement	
8.4 kb	
Native	
2.4 kb	
Size matters (Gzipped)	

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

Time matters

Angular	
3000 ms	
Stencil	
1070 ms	
Native	
1030 ms	
Stencil Pre Rendered	
980 ms	

Le temps ça compte (FMP 3G 📱 en ms)

- 😳 Native et Stencil sont 3 fois plus rapides qu'Angular

Web Components



Web Components

Web Components

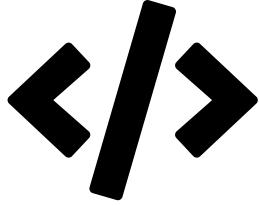


Web Components

History

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

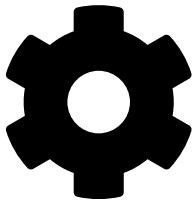
Components



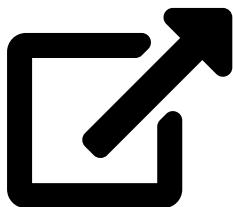
Custom Elements



Shadow DOM

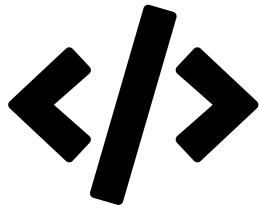


HTML templates



HTML imports

Custom Elements



Custom Elements

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

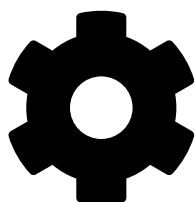
Shadow DOM



Shadow DOM

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

HTML templates



HTML templates

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

Natif code

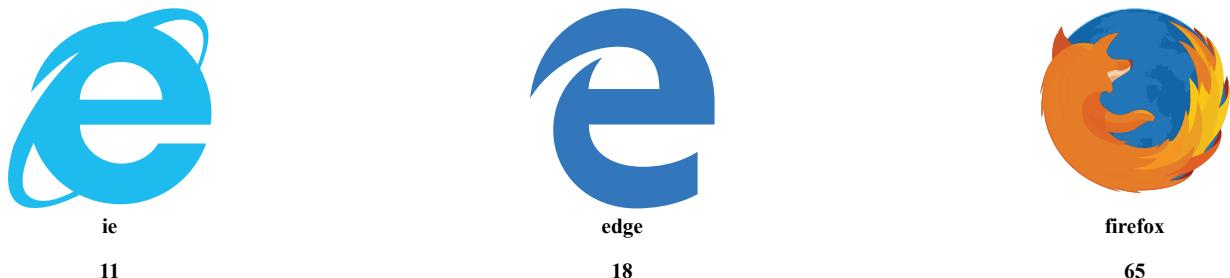
```
class PopUpInfo extends HTMLElement {  
    constructor() {  
        super();  
    }  
}
```

```
// ...
}

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

Browser support

Support des navigateurs



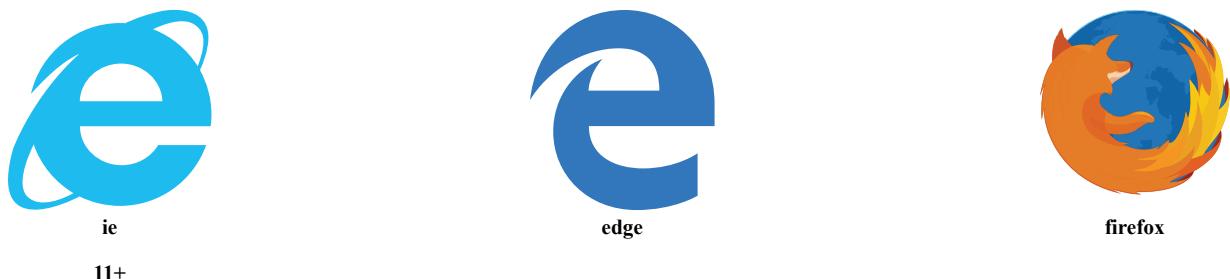
Custom Elements :(
(V1)
Shadow DOM :(
(V1)
HTML templates :(

- Supported
- Partial Support
- Not Supported

<https://caniuse.com>

Polyfill support

Support avec le polyfill

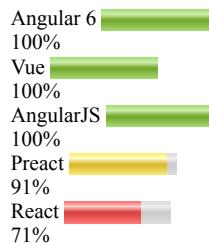


Custom Elements ✓
(V1)
Shadow DOM ✓
(V1)
HTML templates ✓

- Supported
- Partial Support
- Not Supported

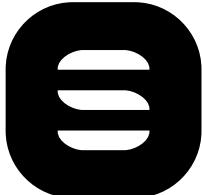
<https://github.com/webcomponents/webcomponentsjs>

Framework Interoperability



Interopérabilité des Web Component
<https://custom-elements-everywhere.com/>

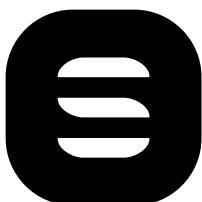
StencilJS



StencilJS

<https://stenciljs.com/>

StencilJS



StencilJS

<https://stenciljs.com/>

What

- Projet **Open Source**, [MIT License](#)
- Crée par l'équipe d'**Ionic** en 2017
- 5.1k ⭐ sur github

Not a framework

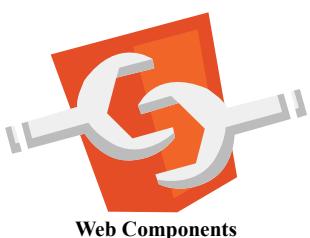
StencilJS n'est pas un autre framework

Compiler

StencilJS c'est un compilateur qui génère des web components

StencilJS is a set of great tools

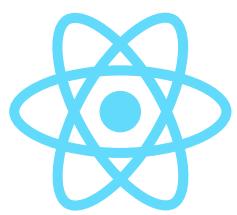
StencilJS c'est un **ensemble de bons outils**



Web Components



Angular



React

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

✗	✓
✓	✗
✓	✗
✗	✗

StencilJS works everywhere

- StencilJS marche partout
- Il charge les polyfills à la demande

Stencil is concise

Web component

Polymer 2

Angular

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

La syntaxe de Stencil est concise

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
```

Stencil code

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

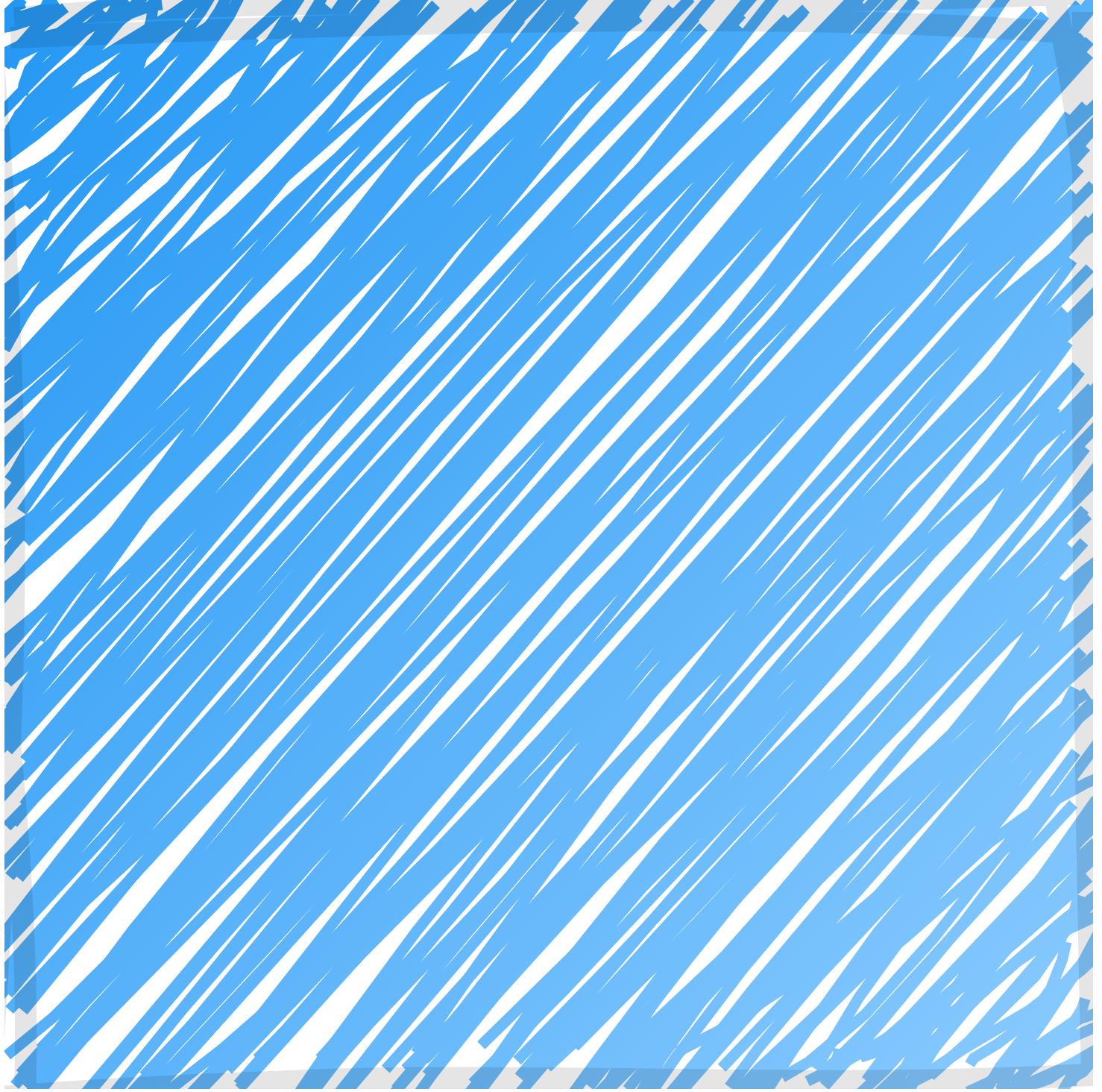
@Component({
  tag: 'my-first-component',
  styleUrls: 'my-first-component.scss'
})
export class MyComponent {
  // Indicate that name should be
  // a public property on the component
  @Prop() name: string;

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

Lit-Elements

```
1 <link rel="import" href="../../bower_components/polymer/polymer-element.html">
2
3 <dom-module id="todo-item">
4   <template>
5     <style>-
6     </style>
7     <li class="item [[isCompleted(checked)]]">
8       <input type="checkbox" value="{{checked}} checked="{{checked::change}}">
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: this.index }));
25      }
26      handleOnChecked(e) {
27        this.dispatchEvent(new CustomEvent('toggle', { detail: this.index }));
28      }
29      isCompleted(completed) {
30        return completed ? 'completed' : '';
31      }
32    }
33    window.customElements.define(TodoItem.is, TodoItem);
34  </script>
35 </dom-module>
```

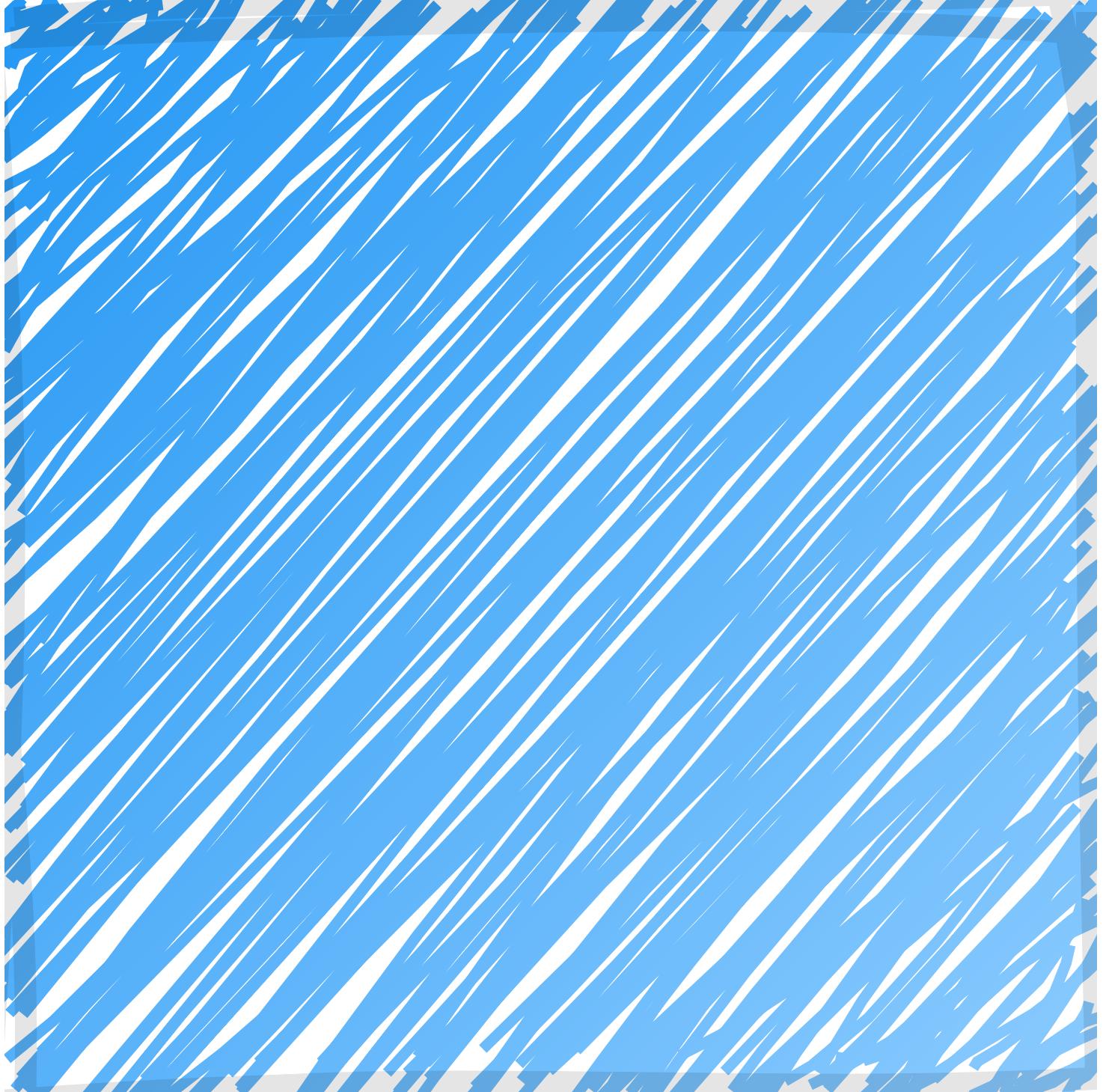
```
1 import { Component, Event } from '@angular/core';
2
3 @Component({
4   selector: 'todo-item',
5   template: `
6     <li class="item" [[isCompleted(checked)]]>
7       <input type="checkbox" value="{{checked}} checked="{{checked::change}}"/>
8       <label>{{text}}</label>
9       <button class="destroy" on-click="handleOnRemove">x</button>
10    </li>
11  `,
12  styles: [`-
13    `],
14  encapsulation: ViewEncapsulation.None
15 })
16 export class TodoItem {
17   @Input() checked: boolean;
18   @Input() text: string;
19   @Input() index: number;
20   @Output() onRemove: () => void;
21   @Output() onTodoItemToggled: () => void;
22
23   handleOnRemove = () => {
24     this.onRemove();
25   }
26   handleOnChecked = () => {
27     this.onTodoItemToggled();
28   }
29 }
30
31 // @jsx h
32 import { props } from 'stencil';
33 import { h } from 'preact';
34 import { Component } from 'stencil';
35
36 export default class TodoItem extends Component {
37   static events = ['check'];
38   static props = {
39     checked: props.boolean,
40     index: props.number
41   };
42
43   handleCheck = e => {
44     this.onCheck({ index: e.target.getAttribute('index') });
45   };
46   handleRemove = () => {
47     this.onRemove({ index: this.index });
48   };
49
50   render({ checked, hand }) {
51     return (
52       <div>
53         <style>`-
54           `</style>
55         <li class={checked ? 'completed' : ''}>
56           <input type="checkbox" checked={checked} value="checked"/>
57           <label>{{text}}</label>
58           <slot />
59           <button on-click="handleRemove">x</button>
60         </li>
61       </div>
62     );
63   }
64 }
```



LitElement

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

Lit-Elements



LitElement

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

What-Lit

- Projet Open Source, [BSD 3-Clause License](#)
- Créer par l'équipe **Polymer Team** en 2017
- 1.7k ★ sur github

Templating

Utilise la bibliothèque de template lit-html

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

- Basé sur les **templates HTML**
- Avec les [*Template literals*](#) de ES2015

lit-html

```
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);
```

lit-elements in JS

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

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);
```

lit-elements in TS

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

@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>! `;
    }
}
```

lit-elements Support

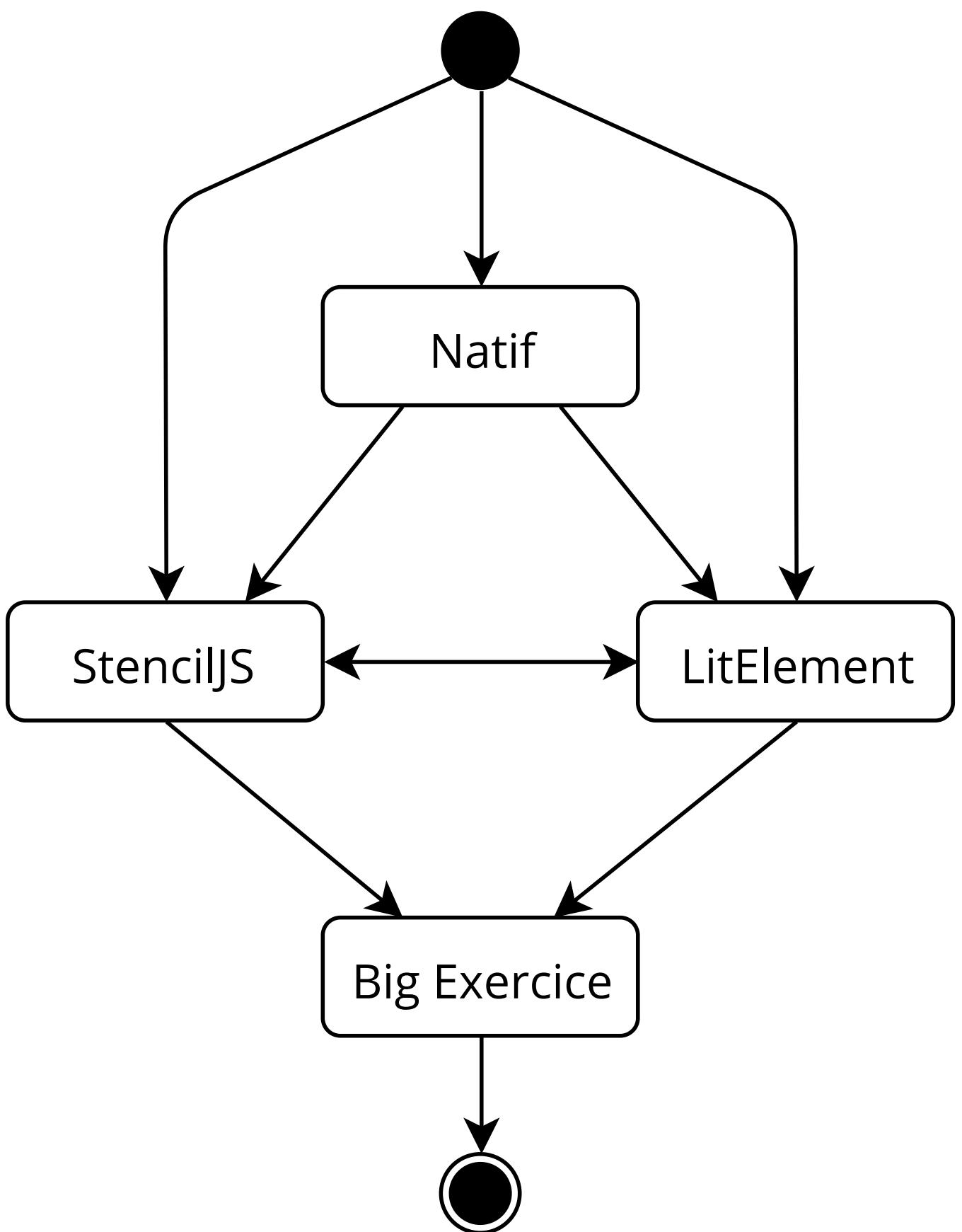
Chrome, Safari, Opera, Firefox

[polyfills](#) pour Edge et IE 11

Workshop

Workshop

Exercises



Exercises

Conclusion

Conclusion

Issues

- Construire une application
- Utiliser un gestionnaire d'état externe
- Style avec un thème
- Utilisez les *custom properties CSS*

- Support des navigateurs
- Utiliser le polyfill ou Electron

...

Alternatives modernes

- [SkateJS](#)
- [Svelte](#)
- [Slim.js](#)
- ...

Fin

Merci

Pensez à nous faire des retours (votez !)