

# stencil

The magical, reusable web component compiler



**Web Component ?**  
**Plaît-il ?**



# Web Component

A set of web platform APIs that allow you to create new custom, reusable, encapsulated HTML tags to use in web pages and web apps...

...will work across modern browsers, and can be used with any JavaScript library or framework that works with HTML.

<whoop-whoop></whoop-whoop>



# Concepts

- Custom Element API
- Shadow DOM
- HTML imports
- HTML Template

```
<template>
  <p>Hello <strong></strong></p>
</template>

<script>
(function(window, document, undefined) {

  var thatDoc = document;
  var thisDoc = (thatDoc._currentScript || thatDoc.currentScript).ownerDocument;

  var template = thisDoc.querySelector('template').content;

  var MyElementProto = Object.create(HTMLElement.prototype);
  MyElementProto.who = 'world';

  MyElementProto.createdCallback = function() {
    var shadowRoot = this.createShadowRoot();
    var clone = thatDoc.importNode(template, true);
    shadowRoot.appendChild(clone);
    this.strong = shadowRoot.querySelector('strong');
    if (this.hasAttribute('who')) {
      var who = this.getAttribute('who');
      this.setWho(who);
    }
    else {
      this.setWho(this.who);
    }
  };
  MyElementProto.attributeChangedCallback = function(attr, oldVal, newVal) {
    if (attr === 'who') {
      this.setWho(newVal);
    }
  };
  MyElementProto.setWho = function(val) {
    this.who = val;
    this.strong.textContent = this.who;
  };

  window.MyElement = thatDoc.registerElement('hello-world', {
    prototype: MyElementProto
  });
})(window, document);
</script>
```

```

<template>
  <p>Hello <strong></strong></p>
</template>

<script>
(function(window, document, undefined) {

  var thatDoc = document;
  var thisDoc = (thatDoc._currentScript || thatDoc.currentScript).ownerDocument;

  var template = thisDoc.querySelector('template').content;

  var MyElementProto = Object.create(HTMLElement.prototype); // Custom Element API
  MyElementProto.who = 'world';

  MyElementProto.createdCallback = function() { // Custom Element API
    var shadowRoot = this.createShadowRoot();
    var clone = thatDoc.importNode(template, true);
    shadowRoot.appendChild(clone);
    this.strong = shadowRoot.querySelector('strong');
    if (this.hasAttribute('who')) {
      var who = this.getAttribute('who');
      this.setWho(who);
    }
    else {
      this.setWho(this.who);
    }
  };
  MyElementProto.attributeChangedCallback = function(attr, oldVal, newVal) { // Custom Element API
    if (attr === 'who') {
      this.setWho(newVal);
    }
  };
  MyElementProto.setWho = function(val) {
    this.who = val;
    this.strong.textContent = this.who;
  };

  window.MyElement = thatDoc.registerElement('hello-world', { // Custom Element API
    prototype: MyElementProto
  });
})(window, document);
</script>

```

```

<template>
  <p>Hello <strong></strong></p>
</template>

<script>
(function(window, document, undefined) {

  var thatDoc = document;
  var thisDoc = (thatDoc._currentScript || thatDoc.currentScript).ownerDocument;

  var template = thisDoc.querySelector('template').content;

  var MyElementProto = Object.create(HTMLElement.prototype);
  MyElementProto.who = 'world';

  MyElementProto.createdCallback = function() {
    var shadowRoot = this.createShadowRoot(); // Shadow DOM
    var clone = thatDoc.importNode(template, true);
    shadowRoot.appendChild(clone);
    this.strong = shadowRoot.querySelector('strong');
    if (this.hasAttribute('who')) {
      var who = this.getAttribute('who');
      this.setWho(who);
    }
    else {
      this.setWho(this.who);
    }
  };
  MyElementProto.attributeChangedCallback = function(attr, oldVal, newVal) {
    if (attr === 'who') {
      this.setWho(newVal);
    }
  };
  MyElementProto.setWho = function(val) {
    this.who = val;
    this.strong.textContent = this.who;
  };

  window.MyElement = thatDoc.registerElement('hello-world', {
    prototype: MyElementProto
  });
})(window, document);
</script>

```



```

<template> <!-- HTML Template -->
  <p>Hello <strong></strong></p>
</template>

<script>
(function(window, document, undefined) {

  var thatDoc = document;
  var thisDoc = (thatDoc._currentScript || thatDoc.currentScript).ownerDocument;

  var template = thisDoc.querySelector('template').content; // HTML Template

  var MyElementProto = Object.create(HTMLElement.prototype);
  MyElementProto.who = 'world';

  MyElementProto.createdCallback = function() {
    var shadowRoot = this.createShadowRoot();
    var clone = thatDoc.importNode(template, true); // HTML Template
    shadowRoot.appendChild(clone); // HTML Template
    this.strong = shadowRoot.querySelector('strong'); // HTML Template
    if (this.hasAttribute('who')) {
      var who = this.getAttribute('who');
      this.setWho(who);
    }
    else {
      this.setWho(this.who);
    }
  };
  MyElementProto.attributeChangedCallback = function(attr, oldVal, newVal) {
    if (attr === 'who') {
      this.setWho(newVal);
    }
  };
  MyElementProto.setWho = function(val) {
    this.who = val;
    this.strong.textContent = this.who; // HTML Template
  };

  window.MyElement = thatDoc.registerElement('hello-world', {
    prototype: MyElementProto
  });
})(window, document);
</script>

```

**et stencil ?**

# The magical, reusable web component *compiler*

<https://stenciljs.com>

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

@Component({
  tag: 'hello-world',
  styleUrls: 'hello-world.scss'
})
export class HelloWorldComponent {

  @Prop() who: string;

  render() {
    return (
      <p>
        Hello {this.who}
      </p>
    );
  }
}
```

# Quels intérêts ?

# Quels intérêts ?

- Syntaxe légère
- Polyfills chargés dynamiquement
- Prerendering
- Server Side Rendering
- Service Workers (Progressive Web App)
- Facilités de développement (livereload, test unitaire, distribution ...)

# Getting started

```
c:\dev
λ npm init stencil
npx: installed 1 in 1.702s
👋 Welcome to Stencil Create App!

What kind of project do you want to create?

? Pick a starter » - Use arrow-keys. Return to submit.
  💎 components (Collection of web components that can be used anywhere)
  💎 app        (Minimal starter for building an stencil app or website)
  💎 ionic-pwa  (Everything you need to build fast, production ready PWAs)
  Other (specify)
```



***Grrr***



***Waouf***



# Questions ?

# StencilJS

```

1
2 import { Component, Prop, Event, EventEmitter } from
  '@stencil/core';
3
4 @Component({
5   tag: 'todo-item',
6   styleUrl: 'todo-item.scss',
7   shadow: true,
8 })
9 export class TodoItem {
10   @Prop() checked: boolean;
11   @Prop() text: string;
12   @Prop() index: number;
13   @Event() onTodoItemChecked: EventEmitter;
14   @Event() onTodoItemRemove: EventEmitter;
15
16   handleOnRemove = () => this.onTodoItemRemove.emit(this.index);
17   handleOnChecked = () => this.onTodoItemChecked.emit(this.index);
18
19   render() {
20     return (
21       <div class={this.checked ? 'completed' : ''}>
22         <input type="checkbox" checked={this.checked} onChange=
23           {this.handleOnChecked} />
24         <label>{this.text}</label>
25         <button onClick={this.handleOnRemove}>x</button>
26       </div>
27     );
28   }
29 }

```

# 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  handleClick = 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, handleClick, handleRemove }) {
20    return html`
21      <style>
22        </style>
23      <li class=${checked ? "completed" : ""}>
24        <input type="checkbox" checked=${checked} on-change=${handleClick}>
25        <label>
26          <slot>
27        </label>
28        <button on-click=${handleRemove}>x</button>
29      </li>
30    `;
31  }
32 }

```