Intro to Web Components

How each API is used to create Web Components.

Iván Gómez Alonso - 05-10-2020

Index

- What is a Web Component?
- APIs:
    - Custom Elements
    - Shadow DOM
    - ES Modules
    - HTML Template
- lit-html
- LitElement
- Other mentions
- Questions

Web Components

What is a Web Component?

A Web Component is a set of APIs available in the browser which allow us to create custom, reusable, encapsulated components without any library like Vue, React, Angular…

And the best of all this is that we can use these components inside those libraries mentioned earlier.

Web APIs used

Custom Elements

- They are written using ES 2015 class syntax.
- It gives us lifecycle callbacks:
  - connectedCallback
  - disconnectedCallback
  - adoptedCallback
  - attributeChangedCallback with the observedAttributes method
- It provides us with a way to register custom elements with the CustomElementRegistry.

ES 2015 class syntax

```
class MyCustomElement extends HTMLElement {
    constructor() {
        super();
        ...
    }
    ...
}
```

CustomElementRegistry

- The names must have a hyphen.
- You cannot register two elements with the same name.

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

customElements.get('my-custom-element');

customElements.whenDefined('my-custom-element');

customElements.upgrade(element);
```

Shadow DOM

- It allows us to encapsulate our styles and templates.

```
this.attachShadow({mode: 'open'});
```

- Slots

```
<slot name="my-slot-name"></slot>

<p slot="my-slot-name"></p>
```

ES Modules

- With them we can load and reuse JS documents.

```
export class MyCustomElement extends HTMLElement {}

import { MyCustomElement } from './path/to/the/file';
```

Note: previously HTML imports were used in the version 0 of the standard

HTML Template

- We declare fragments of our markup so we can instantiated it later (insert it in a document using a script).

```
<template id="my-custom-element">
        <style>
            .my-class {
                background: black;
                color: white;
                min-height: 70px;
                border-radius: 10px;
                text-align: center;
                font-family: 'Roboto', sans-serif;
            }
        </style>
        <p class="my-class"
>This is my web component using template</p>
    </template>
```

lit-html

lit-html

- We can create HTML Templates in JavaScript!!!!

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

// A lit-html template uses the `html` template tag:
let 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);
```

Bindings

```
const myTemplate = (subtotal, tax) => html`<div>Total: ${subtotal + tax}</div>`;

// set the class attribute
const myTemplate2 = (data) => html`<div class=${data.cssClass}>Stylish text.</div>`;

// boolean attributes
const myTemplate3 = (data) => html`<div ?disabled=${!data.active}>Stylish text.</div>`;

// properties
const myTemplate4 = (data) => html`<input .value=${data.value}></input>`;

// events
const myTemplate5 = () => html`<button @click=${clickHandler}>Click Me!</button>`;
```

LitElement

LitELement

- A base class to create our Web Components

```
import { LitElement, html, css } from 'lit-element';

export class LitElementPlayground extends LitElement {
    constructor() {
        super();
        this.description = '';
    }

    static get properties() {
        return {
            description: {
                type: String,
            },
        };
    }

    static get styles() {
        // It supports [css]
        return css``;
    }

    render() {
        // It supports [html]
        return html``;
    }
}
```

Other mentions

Other mentions

- Open WC
- Stencil
- Lion Web Components
- Polymer
- Slim.js
- Shoelace

Questions?