

T7: Web Components

Javiera Inostroza, Elías Sabja, Samuel Zúñiga

¿Qué hicimos?

WebComponents



Custom Elements



Templates



Shadow DOM



Modules



CSS Scopes



Lit

DEMO

Web Components

SELL-ITEM



```
1  import { sellItemStyle } from "../../styles/sell-item.css.js";
2  const template = document.createElement('template');
3
4  template.innerHTML = `
5    <style>
6      ${sellItemStyle}
7    </style>
8
9    <head>
10     <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
11   </head>
12
13   <div class="card">
14     <div class="sell-item-discount"><h4></h4></div>
15     <div class="image-container">
16     </div>
17     <div class="sell-item-name-rating">
18       <h4 class="sell-item-name"></h4>
19       <h4 class="sell-item-rating"><i class="material-icons">star</i></h4>
20     </div>
21     <div class="sell-item-price-container">
22       <div> <h5 class="sell-item-discount-price"></h5> </div>
23       <div> <h6 class="sell-item-price"></h6> </div>
24     </div>
25   </div>
26 `;
```




```
1  export default class SellItem extends HTMLElement {
2    constructor() {
3      super();
4
5      this._shadowRoot = this.attachShadow({ mode: 'open' });
6      this._shadowRoot.appendChild(template.content.cloneNode(true));
7
8      this.$name = this._shadowRoot.querySelector('.sell-item-name');
9      this.$discount = this._shadowRoot.querySelector('.sell-item-discount');
10     this.$discountPrice = this._shadowRoot.querySelector('.sell-item-discount-price');
11     this.$price = this._shadowRoot.querySelector('.sell-item-price');
12     this.$imageContainer = this._shadowRoot.querySelector('.image-container');
13     this.$rating = this._shadowRoot.querySelector('.sell-item-rating');
14   }
15
16   appendTo(htmlElementId) {
17     let element = document.getElementById(htmlElementId)
18     element.appendChild(this);
19   }
```



```
1  connectedCallback() {
2    // Anade imagen
3    const imageElement = document.createElement('img');
4    imageElement.setAttribute('src', this.imageUrl)
5    this.$imageContainer.appendChild(imageElement);
6    // Anade el nombre y otros atributos de texto
7    this.$name.innerHTML = this.name;
8    this.$discountPrice.innerHTML = `>${this.discountPrice}`;
9    this.$price.innerHTML = `Precio normal: ${this.price}`;
10    this.$discount.innerHTML += `-${this.discount}%`;
11    this.$rating.innerHTML += `${this.rating}`;
12  }
```



```
1  get name() {
2      return this.getAttribute('name');
3  }
4
5  get imageUrl() {
6      return this.getAttribute('imageUrl');
7  }
8
9  get price() {
10     return this.getAttribute('price');
11 }
12
13 get discountPrice() {
14     return this.getAttribute('discountPrice');
15 }
16
17 get discount() {
18     return this.getAttribute('discount');
19 }
20
21 get rating() {
22     return this.getAttribute('rating');
23 }
24
25 static get observedAttributes() {
26     return ['name', 'imageUrl', 'price', 'discountPrice', 'discount', 'rating'];
27 }
```

LIT-SELL-ITEM

```
1  export default class LitSellItem extends LitElement {
2      static get styles() {
3          return [litSellItemStyle];
4      }
5
6      appendTo(htmlElementId) {
7          let element = document.getElementById(htmlElementId);
8          element.appendChild(this);
9      }
10
11     connectedCallback() {
12         super.connectedCallback();
13         this.name = this.getAttribute('name');
14         this.imageUrl = this.getAttribute('imageUrl');
15         this.price = this.getAttribute('price');
16         this.discount = this.getAttribute('discount');
17         this.discountPrice = this.getAttribute('discountPrice');
18         this.rating = this.getAttribute('rating');
19     }
20
21     render() {
22         return html`
23             <head>
24                 <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
25             </head>
26
27             <div class="card">
28                 <div class="sell-item-discount">-${this.discount}%</div>
29                 <div class="image-container"></img></div>
30                 <div class="sell-item-name-rating">
31                     <h4 class="sell-item-name">${this.name}</h4>
32                     <h4 class="sell-item-rating"><i class="material-icons">star</i>${this.rating}</h4>
33                 </div>
34                 <div class="sell-item-price-container">
35                     <div> <h5 class="sell-item-discount-price">${this.discountPrice}</h5> </div>
36                     <div> <h6 class="sell-item-price">Precio normal: ${this.price}</h6> </div>
37                 </div>
38             </div>
```

SELL-ITEM

V/S

LIT-SELL-ITEM


```

1 import { sellItemStyle } from "../../styles/sell-item.css.js";
2 const template = document.createElement('template');
3
4 template.innerHTML = `
5   <style>
6     ${sellItemStyle}
7   </style>
8
9   <head>
10    <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
11  </head>
12
13  <div class="card">
14    <div class="sell-item-discount"><h4></h4></div>
15    <div class="image-container">
16      </div>
17    <div class="sell-item-name-rating">
18      <h4 class="sell-item-name"></h4>
19      <h4 class="sell-item-rating"><i class="material-icons">star</i></h4>
20    </div>
21    <div class="sell-item-price-container">
22      <div> <h5 class="sell-item-discount-price"></h5> </div>
23      <div> <h6 class="sell-item-price"></h6> </div>
24    </div>
25  </div>
26 `;
27
28 export default class SellItem extends HTMLElement {
29   constructor() {
30     super();
31
32     this._shadowRoot = this.attachShadow({ mode: 'open' });
33     this._shadowRoot.appendChild(template.content.cloneNode(true));
34
35     this.$name = this._shadowRoot.querySelector('.sell-item-name');
36     this.$discount = this._shadowRoot.querySelector('.sell-item-discount');
37     this.$discountPrice = this._shadowRoot.querySelector('.sell-item-discount-price');
38     this.$price = this._shadowRoot.querySelector('.sell-item-price');
39     this.$imageContainer = this._shadowRoot.querySelector('.image-container');
40     this.$rating = this._shadowRoot.querySelector('.sell-item-rating');
41   }
42
43   appendTo(htmlElementId) {
44     let element = document.getElementById(htmlElementId);
45     element.appendChild(this);
46   }
47
48   get name() {
49     return this.getAttribute('name');
50   }
51
52   get imageUrl() {
53     return this.getAttribute('imageUrl');
54   }
55
56   get price() {
57     return this.getAttribute('price');
58   }
59
60   get discountPrice() {
61     return this.getAttribute('discountPrice');
62   }
63
64   get discount() {
65     return this.getAttribute('discount');
66   }
67
68   get rating() {
69     return this.getAttribute('rating');
70   }
71
72   static get observedAttributes() {
73     return ['name', 'imageUrl', 'price', 'discountPrice', 'discount', 'rating'];
74   }
75
76   connectedCallback() {
77     // Anade imagen
78     const imageElement = document.createElement('img');
79     imageElement.setAttribute('src', this.imageUrl);
80     this.$imageContainer.appendChild(imageElement);
81     // Anade el nombre y otros atributos de texto
82     this.$name.innerHTML = this.name;
83     this.$discountPrice.innerHTML = `${this.discountPrice}`;
84     this.$price.innerHTML = `Precio normal: ${this.price}`;
85     this.$discount.innerHTML += `-${this.discount}%`;
86     this.$rating.innerHTML += `${this.rating}`;
87   }
88 }

```

```

1 export default class LitSellItem extends LitElement {
2   static get styles() {
3     return [litSellItemStyle];
4   }
5
6   appendTo(htmlElementId) {
7     let element = document.getElementById(htmlElementId);
8     element.appendChild(this);
9   }
10
11   connectedCallback() {
12     super.connectedCallback();
13     this.name = this.getAttribute('name');
14     this.imageUrl = this.getAttribute('imageUrl');
15     this.price = this.getAttribute('price');
16     this.discount = this.getAttribute('discount');
17     this.discountPrice = this.getAttribute('discountPrice');
18     this.rating = this.getAttribute('rating');
19   }
20
21   render() {
22     return html`
23       <head>
24         <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
25       </head>
26
27       <div class="card">
28         <div class="sell-item-discount">-${this.discount}%</div>
29         <div class="image-container"></img></div>
30         <div class="sell-item-name-rating">
31           <h4 class="sell-item-name">${this.name}</h4>
32           <h4 class="sell-item-rating"><i class="material-icons">star</i>${this.rating}</h4>
33         </div>
34         <div class="sell-item-price-container">
35           <div> <h5 class="sell-item-discount-price">${this.discountPrice}</h5> </div>
36           <div> <h6 class="sell-item-price">Precio normal: ${this.price}</h6> </div>
37         </div>
38       </div>
39     `;
40   }
41 }

```

ToDo-List



```
1  const template = document.createElement('template');
2  template.innerHTML = `
3      <style>
4          :host {
5              display: block;
6              font-family: sans-serif;
7          }
8          button {
9              border: none;
10             cursor: pointer;
11         }
12     </style>
13     <li class="item">
14         <input type="checkbox">
15         <label></label>
16         <button>-</button>
17     </li>
18 `;
```




```
1  import './todo-item.js';
2
3  const template = document.createElement('template');
4  template.innerHTML = `
5      <style>
6          :host {
7              display: block;
8              font-family: sans-serif;
9              text-align: center;
10         }
11         button {
12             border: none;
13             cursor: pointer;
14         }
15         ul {
16             list-style: none;
17             padding: 0;
18         }
19
20     </style>
21
22     <h1 class='titulo'></h1>
23     <form id="todo-input">
24         <input class='prompt' type="text"></input>
25         <button>+</button>
26     </form>
27
28     <ul id="todos" class="ul"></ul>
29 `;
```



```
1  _renderTodoList() {
2      this.$todoList.innerHTML = '';
3
4      let _todos = this._todos ? this._todos : this.$initialTodos
5
6      _todos.forEach((todo, index) => {
7          let $todoItem = document.createElement('todo-item');
8          $todoItem.setAttribute('text', todo.text);
9
10         if(todo.checked) {
11             $todoItem.setAttribute('checked', '');
12         }
13
14         $todoItem.setAttribute('index', index);
15         $todoItem.addEventListener('onRemove', this._removeTodo.bind(this));
16         $todoItem.addEventListener('onToggle', this._toggleTodo.bind(this));
17
18         this.$todoList.appendChild($todoItem);
19     });
20 }
```



```
1  connectedCallback() {  
2      this.$item1 = this.item1;  
3      this.$item2 = this.item2;  
4      this.$item3 = this.item3;  
5  
6      this.$initialTodos = [  
7          { text: this.$item1, checked: false },  
8          { text: this.$item2, checked: false },  
9          { text: this.$item3, checked: false }  
10     ]  
11  
12     this._todos = this.$initialTodos  
13     this._renderTodoList();  
14  
15     this.$prompt.placeholder = this.prompt;  
16     this.$titulo.innerHTML = this.titulo;  
17 }
```

Lit -ToDo-List



```
1  import { LitElement, html } from 'https://cdn.jsdelivr.net/gh/lit/dist@2/core/lit-core.min.js';
2  import { litTodoItemStyle } from "../../styles/lit-todo-item.css.js";
3
4  export class LitToDoItem extends LitElement {
5      static get properties() {
6          return {
7              item: {type: String},
8              deleteItem: {type: Function},
9          };
10     }
11
12     static get styles() {
13         return [litTodoItemStyle];
14     }
15
16     render() {
17         return html`
18             <li class="item ToDoItem">
19                 <input type="checkbox">
20                 <label class="ToDoItem-Text">${this.item}</label>
21                 <button class="ToDoItem-Delete" @click=${this.deleteItem}>-</button>
22             </li>
23         `;
24     }
25 }
26
27 customElements.define('lit-todo-item', LitToDoItem)
28
```



```
1  import { LitElement, html } from 'https://cdn.jsdelivr.net/gh/lit/dist@2/core/lit-core.min.js';
2  import { litToDoListStyle } from "../../styles/lit-todo-list.css.js";
3  import './lit-todo-item.js';
4
5  export default class LitToDoList extends LitElement {
6      static get properties() {
7          return {
8              list: {type: Array},
9              todo: {type: String},
10             item1: {type: String},
11             item2: {type: String},
12             item3: {type: String},
13             prompt: {type: String},
14             titulo: {type: String}
15         };
16     }
17
18     connectedCallback() {
19         super.connectedCallback()
20
21         this.list = [
22             this.todoItem(this.item1),
23             this.todoItem(this.item2),
24             this.todoItem(this.item3)
25         ];
26         this.todo = '';
27     }
28
29     static get styles() {
30         return [litToDoListStyle];
31     }
32 }
```



```
1  render() {
2    return html`
3      <h1 class='titulo'>${this.titulo}</h1>
4      <form id="todo-input">
5        <input
6          class='prompt'
7          type="text"
8          .value=${this.todo}
9          @input=${this.handleInput}
10         @keypress=${this.handleKeyPress}
11         placeholder=${this.prompt}
12       ></input>
13       <button @click=${this.createNewToDoItem}>+</button>
14     </form>
15
16     <ul id="todos" class="ul">
17       ${this.list.map((item, key) => {
18         return html`
19           <lit-todo-item
20             item=${item.todo}
21             .deleteItem=${this.deleteItem.bind(this, key)}
22           ></lit-todo-item>
23         `;
24       })}
25     </ul>
26   `;
27 }
28 }
```



```
1  import Button from "../components/web-components/my-button.js";
2  import Catalog from "../components/web-components/catalog.js";
3  import SellItem from "../components/web-components/sell-item.js";
4  import MyElement from "../components/lit-element/lit-test.js";
5  import LitSellItem from "../components/lit-element/lit-sell-item.js";
6  import TodoList from "../components/web-components/todo-list.js";
7  import LitToDoList from "../components/lit-element/lit-todo-list.js";
8
9  window.customElements.define('my-button', Button);
10 window.customElements.define('my-catalog', Catalog);
11 window.customElements.define('sell-item', SellItem);
12 window.customElements.define('my-element', MyElement);
13 window.customElements.define('lit-sell-item', LitSellItem);
14 window.customElements.define('todo-list', TodoList);
15 window.customElements.define('lit-todo-list', LitToDoList);
```




```
1 <html>
2   <head>
3     <script src="src/index.js" type="module"></script>
4     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
5   </head>
6   <body>
7
8     <todo-list
9       titulo="Titulo desde html ...."
10      prompt="Agrega algo a la lista :D"
11      item1="item111"
12      item2="item222"
13      item3="item333"
14    ></todo-list>
15
16    <lit-todo-list
17      titulo="Titulo 2 desde html lit"
18      prompt="Agrega algo a la lista jeje xd lit"
19      item1="item1sdfsd1 lit"
20      item2="item22 ... lit"
21      item3="item3 lit 3"
22    ></lit-todo-list>
23
24  </body>
25  <script></script>
26 </html>
```

DIFICULTADES Y CONCLUSIONES

Requiere un manejo manual del DOM con Javascript

Web components son complejos en relación a los componentes de frameworks

Mayor flexibilidad para modificar los componentes (por separado y según necesidad)

Una forma interesante de crear componentes transversales al framework

Se permite encapsulamiento de estilos y comportamiento con el shadowDOM

T7: Web Components

Javiera Inostroza, Elías Sabja, Samuel Zúñiga