

Web components



Grupo 3

Martín Ocqueteau

Nicolás Fraga

Agustín Ríos



1

NFTs y To-do List

Demostración

2

Custom Components

sell-item

```
/* index.html */  
<script defer type="module" src="src/components/SellItem.js">  
</script>  
  
<sell-item  
  title="Astronaut Monkey"  
  price="13900"  
  discount="10"  
  rating="3.7"  
  imgSrc="https://dmn-dallas-news-prod.cdn.arcp...">  
</sell-item>
```



Astronaut Monkey

\$12,510

\$13900

-10%



3.7

```
/* src/components/SellItem.js */
```

```
class SellItem extends HTMLElement {
```

```
  constructor() {
```

```
    super();
```

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

```
    this.shadowRoot.appendChild(template.content.cloneNode(true));
```

```
  }
```

```
/* src/components/SellItem.js */
```

```
const template = document.createElement('template');
```

```
template.innerHTML = /* html */`
```

```
  <div class="container">  // Comment tagged templates
```

```
    <h4 class=title>
```

```
      <slot id="title" name="title">Mundo</slot>
```

```
    </h4>
```

```
/* src/components/SellItem.js */
connectedCallback () {
  if ((this.imgSrc = this.getAttribute('imgSrc'))) {
    this.shadowRoot.getElementById('product-image').setAttribute('src', this.imgSrc);
  }
  if ((this.price = this.getAttribute('price'))) {
    const discount = this.getAttribute('discount') || 0;
    this.shadowRoot.getElementById('sale-price').innerHTML =
      (this.price * (100 - discount) / 100).toLocaleString();
    this.shadowRoot.getElementById('normal-price').innerHTML =
this.price.toLocaleString();
    this.shadowRoot.getElementById('discount-tag').innerHTML = discount;
  }
}
```


todo-list

```
/* index.html */  
<script defer type="module"  
src="src/components/todo-list.js"  
>  
</script>  
  
<todo-list  
  title="ToDo"  
  item1="entrega t7"  
  item2="informes varios"  
  item3="proyecto fluidos"  
  prompt="mas carga">  
</todo-list>
```

ToDo

entrega t7

Delete

informes varios

Delete

proyecto fluidos

Delete

**mas
carga:**

Add

```
/* src/components/todo-list.js */
```

```
class TodoList extends HTMLElement {
```

```
  constructor() {
```

```
    super();
```

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

```
    this.shadowRoot.appendChild(todo_list_template.content.cloneNode(true));
```

```
    this.title;
```

```
    this.item1;
```

```
    this.item2;
```

```
    this.item3;
```

```
    this.prompt;
```

```
    this.item_count = 4;
```

```
  }
```

```
/* src/components/todo-list.js */
connectedCallback() {
  ...
  // give delete button delete function
  this.shadowRoot.querySelectorAll('.item-button').forEach(item => {
    item.addEventListener('click', () => {
      this.delete(item.id);
    });
  });

  // add item button
  this.shadowRoot.getElementById('add-item').addEventListener('click', () => {
    this.add_item(this.shadowRoot.querySelector('#new-item').value);
    this.shadowRoot.querySelector('#new-item').value = '';
  });
}
```

```
/* src/components/todo-list.js */  
add_item(text) {  
  // div item container  
  this.item = document.createElement('div');  
  this.item.id = `${this.item_count}-todo`;   
  // div p item  
  this.p = document.createElement('p');  
  this.p.innerHTML = text;  
  // button deleter  
  this.button = document.createElement('button');  
  this.button.innerHTML = 'Delete';  
  this.button.addEventListener('click', () => {...});  
  // append nodes  
  this.item.appendChild(this.p);  
  this.item.appendChild(this.button);  
  this.shadowRoot.querySelector('.items').appendChild(this.item);  
  // increase item count  
  this.item_count++;  
}
```

3

Lit Elements 

lit-sell-item

```
/* src/components/lit/SellItem.js */
import { LitElement, html, css } from 'lit';
class LitSellItem extends LitElement {
  static get properties() {
    return {
      title: { type: String },
      imgSrc: { type: String },
      price: { type: Number },
      discount: { type: Number },
      rating: { type: Number },
    };
  }
  constructor() {
    super();
    this.title = '';
    this.imgSrc = '';
    this.price = 0;
    this.discount = 0;
    this.rating = 0;
  }
}
```

```
/* src/components/lit/SellItem.js */
render() {
  return html`
    <div class="container">
      <div class="img-container">
        <img id="product-image" src=${this.imgSrc} alt="No Image" />
      </div>
      <h4 class="title">
        ${this.title}
      </h4>
    {...}
  static get styles() {
    return css`
      .container {
        background-color: white;
        border: 1px solid lightgray;
        border-radius: 8px;
        {...}
      }
    `;
  }
}
```

todo-list-lit

```
/* index.html */  
<todo-list-lit  
  title="ToDo"  
  item1="entrega t7"  
  item2="informes varios"  
  item3="proyecto fluidos"  
  prompt="mas carga">  
</todo-list-lit>
```



```
/* src/components/lit/todo-list-lit.js */  
class TodoListLit extends LitElement {  
  
  static get properties() {  
    return {  
      title: { type: String },  
      item1: { type: String },  
      item2: { type: String },  
      item3: { type: String },  
      prompt: { type: String },  
    };  
  }  
  
  constructor() {  
    super();  
    this.title = 'Todo List';  
    this.prompt = 'Add Task';  
    this.todo_list = [];  
    this.input = "";  
  }  
}
```

```
/* src/components/lit/todo-list-lit.js */  
  
firstUpdated() {  
  this.todo_list = [this.item1, this.item2, this.item3];  
  this.update(this.todo_list);  
}  
  
remove_item(idx) {  
  this.todo_list.splice(idx, 1);  
  this.update(this.todo_list);  
}  
  
add_item() {  
  if (this.input !== '') {  
    this.todo_list.push(this.input);  
    this.input = "";  
    this.update(this.input);  
  } else { alert("No trates de agregar un item vacio" ); }  
}
```

```
/* src/components/lit/todo-list-lit.js */  
  
<div class="items">  
  ${repeat(this.todo_list, (item) => item, (item, index) => html`  
    <div class="item">  
      <div class="prompt-item">  
        ${item}  
      </div>  
      <div class="item-button">  
        <button @click="${() => this.remove_item(index)}">Delete</button>  
      </div>  
    </div>  
  `)}  
</div>  
  
. . .  
<input type="text" value="${this.input}" placeholder="new-item" @change=${e =>  
this.input = e.target.value}>  
<button @click="${this.add_item}" type="button">Add</button>
```

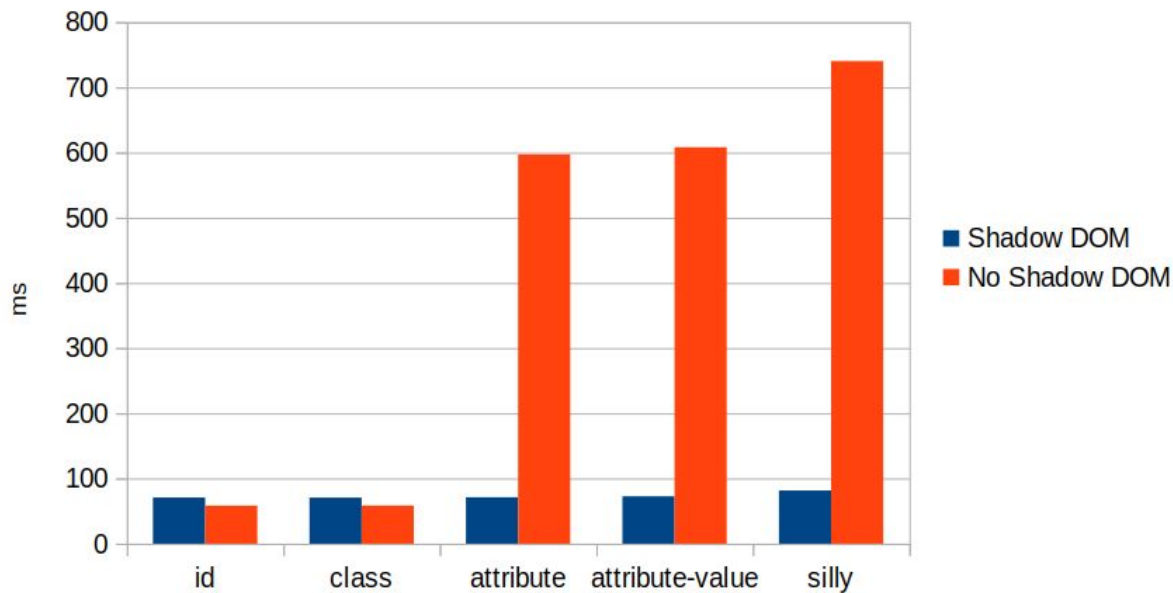


4

Shadow DOM

Rendimiento

Rendering time (lower is better)



<https://nolanlawson.com/2021/08/15/does-shadow-dom-improve-style-performance/>

5

Comentarios 