GRUPO 10

WEB COMPONENTS

DISEÑO AVANZADO DE APLICACIONES WEB

RAFAELA KARA - CRISTOBAL MUÑOZ - FLAVIO TARSETTI

DEMO

CARACTERÍSTICAS IMPLEMENTADAS

COMPONENTE SENCILLA

- Elemento rating
- Elemento descuento

COMPONENTE COMPLEJA

- Componente vista de árbol

Lit

HTML

COMPONENTE SENCILLA

ELEMENTO DE RATING

```
1  import { LitElement, html, css } from 'https://cdn.skypack.dev/lit';
2
3  class RatingStars extends LitElement {
4    static styles = css`
5      ...
6    `;
7
8    static properties = {
9      rating: { type: Number }
10   };
11
12   constructor() {
13     super();
14     this.rating = null;
15   }
16
17   getRatingText() {
18     return this.rating ? `${this.rating}/5 stars` : 'No rating'
19   }
20
21   handleClick(rating) {
22     this.rating = rating;
23   }
24   ...
25 }
26
```

Definición CSS propio

Definición de atributos

Inicialización del componente

FUNCIONES AUXILIARES

- Obtener texto tooltip
- Modificar rating

```
1  import { LitElement, html, css } from 'https://cdn.skypack.dev/lit';
2
3  class RatingStars extends LitElement {
4    ...
5    render() {
6      const stars = [];
7
8      for (let i = 1; i ⩽ 5; i++) {
9        stars.push(html`
10         <span
11           class="star ${i ⩽ this.rating ? 'checked' : ''}"
12           @click=${() ⇒ this.handleClick(i)}
13         >
14           &#9733;
15         </span>
16       `);
17     }
18
19     return html`
20       <div class="tooltip">
21         ${stars}
22         <span class="tooltiptext">${this.getRatingText()}</span>
23       </div>`;
24   }
25 }
26
27 customElements.define('rating-stars', RatingStars);
```

Definición de estrellas según rating

Definición HTML

Definición de etiqueta

COMPONENTE SENCILLA

ELEMENTO DE DESCUENTO

```
class DiscountElement extends LitElement {
  static styles = css`
    ...
  `;

  static properties = {
    discount: { type: Number },
    originalPrice: { type: Number },
    imageUrl: { type: String },
    rating: { type: Number }
  };

  constructor() {
    super()
    this.imageUrl = 'https://www.proclinic-products.com/build/static/default-product.30484205.png'
    this.discount = 0;
  }

  render() {
    const discountedPrice = parseInt(this.originalPrice - (this.originalPrice * (this.discount / 100)), 10);

    return html`
      <div class="discount-info">
        <span class="discount">${this.discount > 0 ? `${this.discount}% OFF` : null}</span>
        <img class="product-image" src="${this.imageUrl}" alt="Product Image"></img>
        <slot></slot>
        <span class="discounted-price">$${discountedPrice}</span>
        <span class="original-price">${this.discount > 0 ? `$${this.originalPrice}` : null}</span>
        <rating-stars rating="${this.rating}"></rating-stars>
      </div>
    `;
  }
}

customElements.define('discount-element', DiscountElement);
```

1. Definición CSS propio
2. Definición de atributos
3. Inicialización del componente y valores por defecto
4. Cálculo del nuevo precio
5. Defínción del HTML, incluyendo componente de rating
6. Definición de etiqueta

COMPONENTE COMPLEJA

TREE-ITEM

1. Inicialización del componente, variables y Shadow DOM
2. Al agregarse tree-item al DOM, se define un EventListener
3. Función que modifica variable isOpen, al hacer click
4. Definición de HTML para elementos anidados
5. Define el ícono que se encuentra al inicio del tree-item dependiendo si esta abierto o cerrado.

```
1  class TreeItem extends HTMLElement {
2      constructor() {
3          super();
4          this.attachShadow({ mode: 'open' });
5          this.isOpen = false;
6      }
7
8      connectedCallback() {
9          this.render();
10         this.addEventListener('click', this.toggleMenu.bind(this));
11     }
12
13     toggleMenu(event) {
14         event.stopPropagation();
15         if (this.childNodes.length > 1) {
16             this.isOpen = !this.isOpen;
17         }
18         this.render();
19     }
20
21     getNestedTreeItems() {
22         const treeItems = Array.from(this.children)
23         return treeItems.map(elem => elem.outerHTML).join('')
24     }
25
26     getDetailsIcon() {
27         if (this.isOpen) {
28             return `'▼'`
29         }
30         return `'▶'`
31     }
32     ...
33 }
```

COMPONENTE COMPLEJA

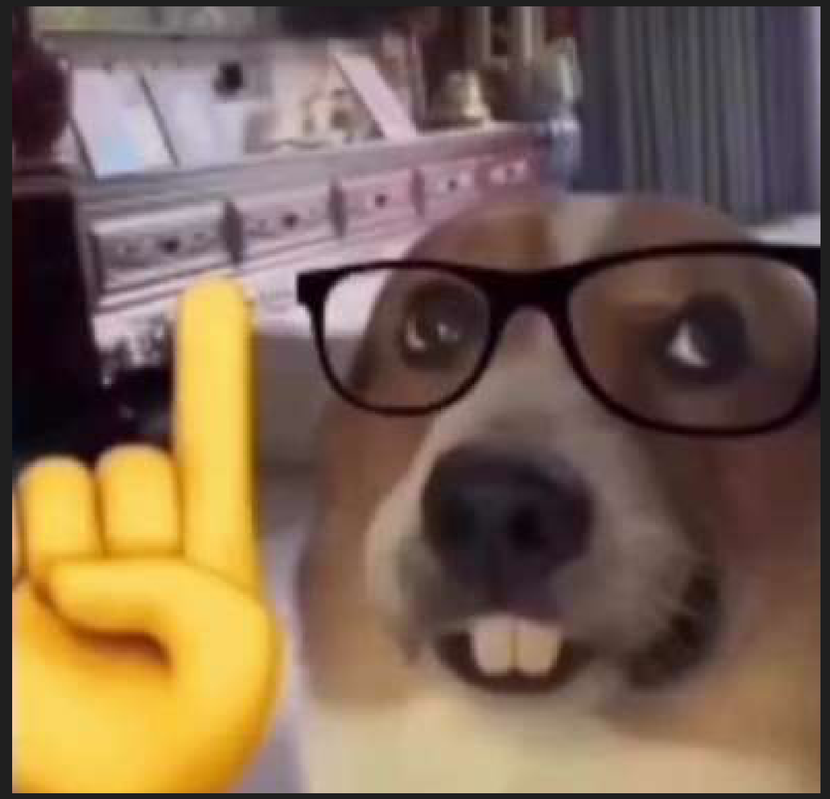
TREE-ITEM

```
class TreeItem extends HTMLElement {
    ...

    render() {
        const template = document.createElement('template');
        template.innerHTML = `
        <style>
            ...
        </style>

        ${this.childNodes.length > 1 ?
        `
        <details class="tree-item" open>
            <summary>
                ${this.childNodes[0].textContent}
            </summary>
            <div class="submenu">
                ${this.getNestedTreeItems()}
            </div>
        </details>
        `
        :
        `
        <div class="tree-item">
            <slot></slot>
        </div>
        `
        }
        `;
        this.shadowRoot.innerHTML = '';
        this.shadowRoot.appendChild(template.content.cloneNode(true));
    }
}

customElements.define('tree-item', TreeItem);
```

6 7 8 9 10

6. Definición CSS propio
7. Definición HTML de nodos padre
8. Definición HTML de nodos hoja
9. Definimos contenido del Shadow DOM
10. Definición de etiqueta

CONCLUSIONES



- No hay gran diferencia entre LitElement y HTMLElement (por ahora)
- No es tan complejo crear web-components.
- Puede ser muy útil para reutilizar código.

MUCHAS GRACIAS