

Ejemplo 1: Shadow DOM con Custom Element y JS Interno

Todo el código está incluido dentro del HTML.

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4    <meta charset="UTF-8">
5    <title>Shadow DOM Interno</title>
6  </head>
7  <body>
8
9    <!-- Uso del Custom Element -->
10   <mi-tarjeta></mi-tarjeta>
11
12   <script>
13     // Definición del Custom Element
14     class MiTarjeta extends HTMLElement {
15       constructor() {
16         super();
17
18         // Crear el Shadow DOM en modo abierto
19         const shadow = this.attachShadow({ mode: 'open' });
20
21         // Estructura interna del componente
22         shadow.innerHTML = `
23           <style>
24             .tarjeta {
25               border: 2px solid #000;
26               border-radius: 8px;
27               padding: 10px;
28               width: 200px;
29               text-align: center;
30               background-color: #f9f9f9;
31             }
32             h3 { color: #007BFF; }
33           </style>
34           <div class="tarjeta">
35             <h3>Hola desde Shadow DOM</h3>
36             <p>Este es un componente personalizado.</p>
37           </div>
38         `;
39       }
40     }
41
42     // Registrar el Custom Element
43     customElements.define('mi-tarjeta', MiTarjeta);
44   </script>
45
46 </body>
47 </html>
```

Explicación:

- El **Shadow DOM** encapsula estilos y estructura.
- Los estilos dentro del shadow **no afectan** al resto de la página.
- El modo "open" permite acceder desde el navegador usando `document.querySelector('mi-tarjeta').shadowRoot`.

Ejemplo 2: Shadow DOM con Custom Element y JS Externo

Separa el JavaScript en un archivo externo para mejor organización.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Shadow DOM Externo</title>
</head>
<body>

  <h2>Lista de Productos</h2>

  <!-- Uso del Custom Element -->
  <producto-card nombre="Café" precio="2.50"></producto-card>
  <producto-card nombre="Té Verde" precio="1.80"></producto-card>

  <!-- Importa el archivo JS externo -->
  <script type="module" src="producto-card.js"></script>

</body>
</html>
```

```
// Clase para el Custom Element
class ProductoCard extends HTMLElement {
  constructor() {
    super();

    // Crear el Shadow DOM
    const shadow = this.attachShadow({ mode: 'open' });

    // Obtener atributos del HTML
    const nombre = this.getAttribute('nombre') || 'Producto';
    const precio = this.getAttribute('precio') || '0.00';

    // Estructura interna del componente
    shadow.innerHTML = `
      <style>
        .producto {
          border: 1px solid #ddd;
          border-radius: 5px;
          padding: 15px;
          width: 150px;
          text-align: center;
          background-color: #f5f5f5;
        }
        .producto h4 { color: #e91e63; margin: 5px 0; }
        .producto p { font-weight: bold; }
      </style>
      <div class="producto">
        <h4>${nombre}</h4>
        <p>Precio: ${precio}</p>
      </div>
    `;
  }
}

// Registrar el Custom Element
customElements.define('producto-card', ProductoCard);
```

Explicación:

- El **HTML** usa el componente `<producto-card>` con atributos personalizados (`nombre` y `precio`).
 - El **JS externo** gestiona la lógica del componente.
 - El **Shadow DOM** encapsula los estilos y la estructura.
 - Los estilos dentro del componente **no afectan** al resto de la página.
-



Diferencia Entre JS Interno y Externo:

- **Interno:** Más simple para pequeños proyectos o pruebas rápidas.
- **Externo:** Mejor organización y escalabilidad para proyectos grandes.