Lanka Nippon Biztech Institute

Web Component

CMP120230 – Web Programming

Name : Gayan Dhananjaya

Index Number : UGC0122020

Batch : SE - 01

OUTLINE

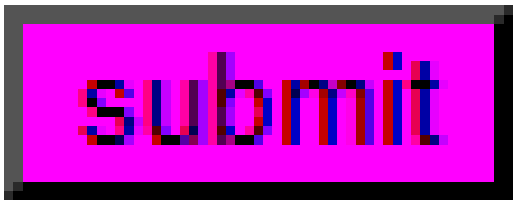
INTRODUCTION

Colorful button web component

“Colorful button” web component is a tool and ready-made solution that helps you add a button color changing feature to your website or application. Allows developers to customize the appearance of the language selector through the associated CSS styles, providing flexibility in adaption the component to match the overall design of the web application.

The <custom-button> element is defined as a custom web componenet using the customElements.define() method. It takes two arguments: the name of the custom element (‘custom-button’) and the class that defines its behavior (customButton).

LAYOUT



This is the code part that I implemented.

HTML

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    .custom-button {
      display: inline-block;
      padding: 10px 20px;
      font-size: 16px;
      cursor: pointer;
    }
  </style>
</head>
<body>
  <custom-button text="submit" color="#ff00ff"></custom-button>

  <script src="componant.js"></script>

</body>
</html>
```

JavaScript

```
document.addEventListener('DOMContentLoaded', function () {
  class CustomButton extends HTMLElement {
    constructor() {
      super();

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

      const template = document.createElement('template');
      template.innerHTML = `
        <button class="custom-button"></button>
      `;

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

      this.buttonElement = this.shadowRoot.querySelector('.custom-button');
      this.buttonElement.addEventListener('click', this.handleButtonClick.bind(this));
    }

    connectedCallback() {
      this.updateButton();
    }

    static get observedAttributes() {
      return ['text', 'color'];
    }

    attributeChangedCallback(name, oldValue, newValue) {
      if (oldValue !== newValue) {
        this.updateButton();
      }
    }
```

Ln 12, Col 9 Spaces: 2 UTF-8 CRLF {} JavaScri

```
33
34      updateButton() {
35        const buttonText = this.getAttribute('text') || 'Click me';
36        const buttonColor = this.getAttribute('color') || '#3498db';
37
38        this.buttonElement.textContent = buttonText;
39        this.buttonElement.style.backgroundColor = buttonColor;
40      }
41
42      handleButtonClick() {
43        alert('Button clicked!');
44        // You can perform additional actions here when the button is clicked.
45      }
46    }
47
48    customElements.define('custom-button', CustomButton);
49  });
50
```

USAGE

In the HTML body, a <custom-button>element is used with the attributes text=”customText” and color=”#e74c3c”. This creates an instance of the custom button with the specified text and color.

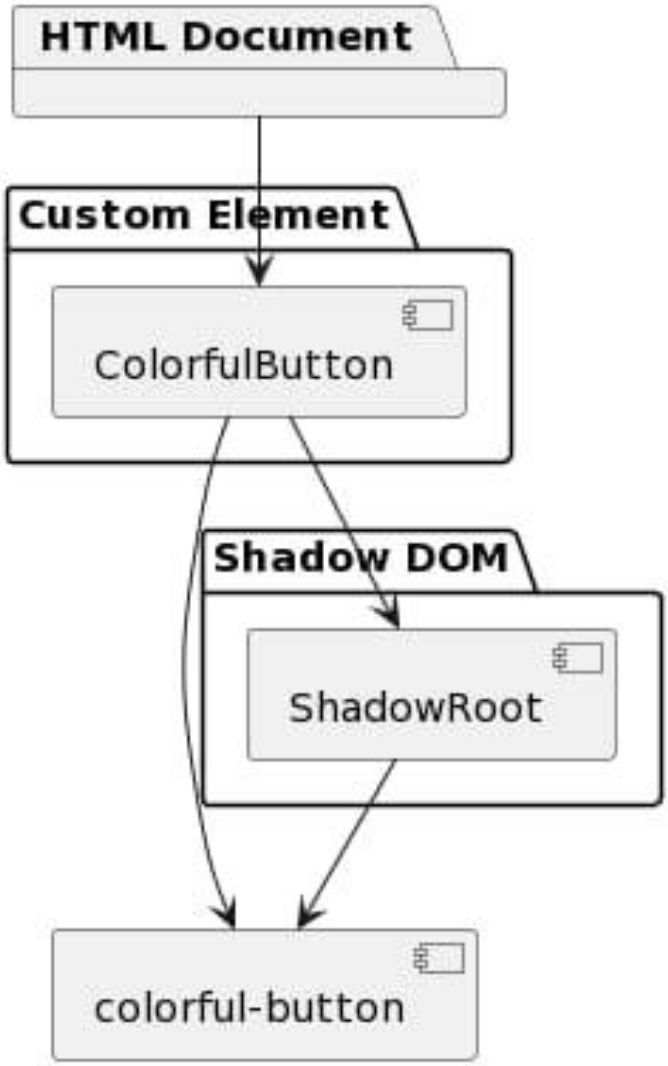
- Customizable Styling

  Allows developers to customize the appearance of the language selector through the associated CSS styles, providing flexibility in adapting the component to match the overall design of the web application.

- Encapsulation

  The button's internal structure and styles are encapsulated within the Shadow DOM, preventing external styles from affecting its appearance and ensuring better isolation from the rest of the document.

Component Diagram



REFERENCES

- ✓ https://developer.mozilla.org/en-US/docs/Web/API/Web_components
- ✓ https://www.webcomponents.org/