{javascript}

Desvendando os Web Components

Willian de Mattos Silva

Introdução aos Web Components

- O que são Web Components?
  - Web Components são um conjunto de tecnologias da web que permitem a criação de elementos HTML personalizados e reutilizáveis.
  - Eles encapsulam HTML, CSS e JavaScript para criar componentes independentes e interoperáveis.

Tecnologias dos Web Components

HTML Templates:

- Permite a definição de modelos de elementos HTML que podem ser clonados e inseridos no documento quando necessário.

Custom Elements:

- Permite a definição de novos elementos HTML personalizados, com comportamento e estilo específicos.

Shadow DOM (DOM Sombreado):

- Permite encapsular o estilo e o comportamento de um componente, isolando-o do restante da página.

HTML Imports (Importações HTML) (opcional):

- Permite importar e reutilizar componentes HTML em outros documentos HTML.

HTML Templates

- Os HTML Templates são uma forma de definir modelos de elementos HTML dentro do documento que podem ser clonados e inseridos no DOM quando necessário.

- Eles são úteis para definir estruturas complexas de elementos que podem ser reutilizadas várias vezes.

- Os templates são inertes por padrão, o que significa que seu conteúdo não é renderizado até ser ativado via JavaScript.

HTML Templates

```html
<template id="myTemplate">
  <p>Este é um template HTML.</p>
</template>

<script>
  const template =
document.getElementById('myTemplate');
  const clone =
template.content.cloneNode(true);
  document.body.appendChild(clone);
</script>
```

Custom Elements

- Os Custom Elements permitem a definição de novos elementos HTML personalizados, com comportamento e estilo específicos.
- Eles são criados usando JavaScript e podem encapsular funcionalidades complexas em um único componente.
- Os Custom Elements são registrados no navegador usando **customElements.define()**.

Custom Elements

```
<script>
class MyCustomElement extends HTMLElement {
  constructor() {
    super();
    this.textContent = 'Este é um elemento
HTML personalizado.';
  }
}

customElements.define('my-custom-element',
MyCustomElement);
</script>

<my-custom-element></my-custom-element>
```

Shadow DOM (DOM Sombreado)

- O Shadow DOM permite encapsular o estilo e o comportamento de um componente, isolando-o do restante da página.
- Cada elemento com Shadow DOM tem sua própria árvore de DOM, estilo e eventos encapsulados.
- Isso evita que estilos externos afetem o componente e vice-versa, proporcionando uma maior modularidade e reutilização.

Propriedades do Shadow DOM

- **Mode**: Define se o Shadow DOM é aberto (open) ou fechado (closed). No modo aberto, o Shadow DOM pode ser acessado fora do elemento que o contém.

- **Encapsulation**: O Shadow DOM encapsula estilos e comportamentos do componente, garantindo que não sejam afetados por estilos externos e que não afetem outros elementos na página.

- **Scoped CSS**: O CSS dentro do Shadow DOM é aplicado apenas aos elementos dentro do componente, evitando que afete outros elementos na página.

Shadow DOM (DOM Sombreado)

```
<script>
class MyCustomElementWithShadowDOM extends
HTMLElement {
  constructor() {
    super();
    const shadow = this.attachShadow({ mode:
'open' });
    const paragraph =
document.createElement('p');
    paragraph.textContent = 'Este é um
elemento com Shadow DOM.';
    shadow.appendChild(paragraph);
  }
}

customElements.define('my-custom-element-
shadow', MyCustomElementWithShadowDOM);
</script>

<my-custom-element-shadow></my-custom-
element-shadow>
```

HTML Imports (Importações HTML)

- HTML Imports é uma forma de importar e reutilizar componentes HTML em outros documentos HTML.
- Eles permitem dividir o código em módulos reutilizáveis, tornando o desenvolvimento mais organizado e modular.
- No entanto, HTML Imports não é amplamente suportado em todos os navegadores e está sendo substituído por outros métodos de importação, como módulos JavaScript.

HTML Imports (Importações HTML)

```html
<!-- my-component.html -->
<template id="myComponentTemplate">
  <p>Este é um componente HTML importado.
</p>
</template>
<script>
  const template =
document.getElementById('myComponentTemplate
');
  const clone =
template.content.cloneNode(true);
  document.body.appendChild(clone);
</script>
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
  <title>HTML Imports Example</title>
  <link rel="import" href="my-
component.html">
</head>
<body>
  <script>
    const link =
document.querySelector('link[rel="import"]')
;
    const content =
link.import.querySelector('template').conten
t;

 document.body.appendChild(content.cloneNode
(true));
  </script>
</body>
</html>
```

Vantagens dos Web Components

- Reutilização de código: Os componentes podem ser facilmente reutilizados em diferentes projetos.
- Encapsulamento: O Shadow DOM permite encapsular estilos e comportamentos, evitando interferências externas.
- Interoperabilidade: Os Web Components funcionam em todos os navegadores modernos, sem a necessidade de bibliotecas externas.

Conclusão

- Os Web Components são uma poderosa ferramenta para o desenvolvimento de aplicações web modulares e escaláveis.

- Combinando HTML Templates, Custom Elements, Shadow DOM e possivelmente HTML Imports, podemos criar componentes personalizados e encapsulados.

Perguntas ?

{Obrigado}