

# JavaScript DOM

DOM简介

获取元素

修改元素

添加和移除元素

事件处理

## DOM简介

JavaScript DOM 是指 JavaScript 中的文档对象模型（Document Object Model）；它允许 JavaScript 与 HTML 页面交互，使开发者可以通过编程方式动态地修改网页内容和样式，以及响应用户的交互。

## 获取元素

获取元素是使用 JavaScript DOM 最常见的操作，可以使用以下方法获取元素：

- `document.getElementById(id)` 通过元素的 ID 获取元素
- `document.getElementsByClassName(className)` 通过类名获取元素
- `document.getElementsByTagName(tagName)` 通过标签名获取元素
- `document.querySelector(selector)` 通过选择器获取第一个匹配的元素
- `document.querySelectorAll(selector)` 通过选择器获取所有匹配的元素

例如，以下代码将获取元素并将其存储在变量 `myElement` 中：

```
1  var myElement = document.getElementById("myId");
```

## 修改元素

一旦获取了元素，就可以修改它的内容、样式或属性。以下是一些基础的元素修改方法：

- `element.innerHTML = html` 修改元素的 HTML 内容
- `element.innerText = text` 修改元素的文本内容

- `element.setAttribute(name, value)` 设置元素的属性
- `element.style.property = value` 修改元素的样式

例如，以下代码将修改元素的背景颜色：

```
▼ JavaScript |
1 myElement.style.backgroundColor = "red";
```

## 添加和移除元素

可以使用以下方法来添加或移除元素：

- `document.createElement(tagName)` 创建新的元素
- `parentElement.appendChild(newElement)` 将一个新元素添加到现有元素中
- `parentElement.removeChild(element)` 从现有元素中删除元素

例如，以下代码将创建一个新的 `div` 元素并将其添加到 `body` 元素中：

```
▼ JavaScript |
1 var newDiv = document.createElement("div");
2 document.body.appendChild(newDiv);
```

## 事件处理

通过 JavaScript DOM，可以添加事件处理程序来响应用户的交互。以下是一些添加事件处理程序的方法：

- `element.addEventListener(event, function)` 向元素添加事件监听器
- `element.removeEventListener(event, function)` 从元素中删除事件监听器

例如，以下代码将向按钮添加点击事件监听器：

```
▼ JavaScript |
1 myButton.addEventListener("click", function() {
2     alert("Button clicked!");
3 });
4
```