DOM

DOM (Document Object Model) в JavaScript представляет собой структуру документа в виде дерева объектов, где каждый узел представляет собой часть документа (например, элемент, атрибут, текст и т.д.). JavaScript может взаимодействовать с DOM, изменяя его, обновляя, добавляя или удаляя элементы. Одним из основных методов доступа к элементам DOM является getElementById.

Пример использования getElementById:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>DOM Example</title>

<style>

.highlight {

color: red;

font-weight: bold;

}

</style>

</head>

<body>

<h1 id="mainTitle">Hello, DOM!</h1>

<p>This is a simple example of the Document Object Model (DOM).</p>

<script>

// Получение элемента по его идентификатору (id)

var titleElement = document.getElementById('mainTitle');

// Изменение содержимого элемента

titleElement.innerHTML = 'New Title';

// Изменение стиля элемента

titleElement.style.color = 'blue';

// Добавление класса к элементу

titleElement.classList.add('highlight');

</script>

</body>

</html>

В этом примере:

* <h1 id="mainTitle">Hello, DOM!</h1> - у нас есть элемент <h1>, который имеет уникальный идентификатор (id) "mainTitle".
* var titleElement = document.getElementById('mainTitle'); - мы используем getElementById для получения ссылки на этот элемент по его идентификатору.
* titleElement.innerHTML = 'New Title'; - мы изменяем содержимое элемента.
* titleElement.style.color = 'blue'; - мы изменяем стиль элемента.
* titleElement.classList.add('highlight'); - мы добавляем класс "highlight" к элементу.

Таким образом, использование getElementById позволяет нам получать доступ к элементам по их уникальному идентификатору и взаимодействовать с ними с помощью JavaScript.

HTML DOM — это также и объектная модель HTML. Он определяет:

* HTML-элементы как объекты
* Свойства для всех элементов HTML.
* Методы для всех элементов HTML.
* События для всех элементов HTML

HTML DOM — это API (интерфейс программирования, то есть определенный общий набор методов работы) для JavaScript:

* JavaScript может добавлять/изменять/удалять элементы HTML.
* JavaScript может добавлять/изменять/удалять атрибуты HTML.
* JavaScript может добавлять/изменять/удалять стили CSS.
* JavaScript может реагировать на события HTML.
* JavaScript может добавлять/изменять/удалять события HTML.

Когда веб-страница загружается, браузер создает объектную модель документа страницы.

Модель HTML DOM построена как дерево объектов.

Поиск HTML-элементов

Если вы хотите получить доступ к HTML-элементам с помощью JavaScript, вам нужно сначала найти эти элементы.

Есть несколько способов сделать это:

* Поиск HTML-элементов по идентификатору
* Поиск HTML-элементов по имени тега.
* Поиск HTML-элементов по имени класса.
* Поиск элементов HTML с помощью селекторов CSS.
* Поиск элементов HTML по коллекциям объектов HTML.

Поиск HTML-элемента по идентификатору

Самый простой способ найти элемент HTML в DOM — использовать идентификатор элемента.

В этом примере выполняется поиск элемента с id="intro":

var myElement = document.getElementById("intro");