

10. Методы `contains` и `compareDocumentPosition`

Метод `contains` для проверки на вложенность

Синтаксис:

```
var result = parent.contains(child);
```

Возвращает `true`, если `parent` содержит `child` или `parent == child`.

Метод `compareDocumentPosition` для порядка узлов

Бывает, что у нас есть два элемента, к примеру, `` в списке, и нужно понять, какой из них выше другого.

Метод `compareDocumentPosition` – более мощный, чем `contains`, он предоставляет одновременно информацию и о содержании и об относительном порядке элементов.

Синтаксис:

```
var result = nodeA.compareDocumentPosition(nodeB);
```

Возвращаемое значение – битовая маска.

- Для проверки, является ли один узел предком другого, достаточно метода `nodeA.contains(nodeB)`.
- Для расширенной проверки на предшествование есть метод `compareDocumentPosition`.
- Для IE8 нужен полифил для `compareDocumentPosition`.
-

11. Изменение документа

Создание элемента

DOM-узел можно создать двумя методами:

`document.createElement(tag)`

Создаёт новый элемент с заданным тегом:

```
let div = document.createElement('div');
```

`document.createTextNode(text)`

Создаёт новый текстовый узел с заданным текстом:

```
let textNode = document.createTextNode('А вот и я');
```

Создание сообщения

В нашем случае сообщение – это `div` с классом `alert` и HTML в нём:

```
let div = document.createElement('div');
```

```
div.className = "alert";
```

```
div.innerHTML = "<strong>Всем привет!</strong> Вы прочитали важное сообщение.»;
```

Методы вставки

Чтобы наш `div` появился, нам нужно вставить его где-нибудь в `document`. Например, в `document.body`. Для этого есть метод `append`, в нашем случае: `document.body.append(div)`.

Вот методы для различных вариантов вставки:

- `node.append(...nodes or strings)` – добавляет узлы или строки в конец `node`,
- `node.prepend(...nodes or strings)` – вставляет узлы или строки в начало `node`,
- `node.before(...nodes or strings)` – вставляет узлы или строки до `node`,
- `node.after(...nodes or strings)` – вставляет узлы или строки после `node`,
- `node.replaceWith(...nodes or strings)` – заменяет `node` заданными узлами или строками.

Удаление узлов

Для удаления узла есть методы `node.remove()`.

Клонирование узлов: `cloneNode`

Как вставить ещё одно подобное сообщение?

Мы могли бы создать функцию и поместить код туда. Альтернатива – клонировать существующий `div` и изменить текст внутри него (при необходимости).

Иногда, когда у нас есть большой элемент, это может быть быстрее и проще.

- Вызов `elem.cloneNode(true)` создаёт «глубокий» клон элемента – со всеми атрибутами и дочерними элементами. Если мы вызовем `elem.cloneNode(false)`, тогда клон будет без дочерних элементов.