

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.

DocumentFragment

`DocumentFragment` является специальным DOM-узлом, который служит обёрткой для передачи списков узлов.

Мы можем добавить к нему другие узлы, но когда мы вставляем его куда-то, он «исчезает», вместо него вставляется его содержимое.

КРАТКО:

- Методы для создания узлов:
 - `document.createElement(tag)` – создаёт элемент с заданным тегом,
 - `document.createTextNode(value)` – создаёт текстовый узел (редко используется),
 - `elem.cloneNode(deep)` – копирует элемент, если `deep==true`, то со всеми дочерними элементами.
- Вставка и удаление:
 - `node.append(...nodes or strings)` – вставляет в `node` в конец,
 - `node.prepend(...nodes or strings)` – вставляет в `node` в начало,
 - `node.before(...nodes or strings)` – вставляет прямо перед `node`,
 - `node.after(...nodes or strings)` – вставляет сразу после `node`,
 - `node.replaceWith(...nodes or strings)` – заменяет `node`.
 - `node.remove()` – удаляет `node`.
- Устаревшие методы:
 - `parent.appendChild(node)`
 - `parent.insertBefore(node, nextSibling)`
 - `parent.removeChild(node)`
 - `parent.replaceChild(newElem, node)`
- Все эти методы возвращают `node`.
- Если нужно вставить фрагмент HTML, то `elem.insertAdjacentHTML(where, html)` вставляет в зависимости от `where`:
 - `"beforebegin"` – вставляет `html` прямо перед `elem`,
 - `"afterbegin"` – вставляет `html` в `elem` в начало,
 - `"beforeend"` – вставляет `html` в `elem` в конец,
 - `"afterend"` – вставляет `html` сразу после `elem`.
- Также существуют похожие методы `elem.insertAdjacentText` и `elem.insertAdjacentElement`, они вставляют текстовые строки и элементы, но они редко используются.
- Чтобы добавить HTML на страницу до завершения её загрузки:
 - `document.write(html)`