

3. Работа с DOM из консоли

Исследовать и изменять DOM можно с помощью инструментов разработки, встроенных в браузер.

Отображение DOM во вкладке Elements не совсем соответствует реальному. В частности, там не отображаются пробельные узлы. Это сделано для удобства просмотра. Но мы-то знаем, что они есть.

С другой стороны, в Elements можно увидеть CSS-псевдоэлементы, такие как `::before`, `::after`. Это также сделано для удобства, в DOM их не существует.

Последний элемент, выбранный во вкладке Elements, доступен в консоли как `$0`, предыдущий – `$1` и так далее.

Есть и обратная дорожка. Любой элемент из JS-переменной можно посмотреть во вкладке Elements, для этого:

1. Выведите эту переменную в консоли, например при помощи `console.log`.
2. Кликните на выводе в консоли правой кнопкой мыши.
3. Выберите соответствующий пункт меню.

Таким образом, можно легко перемещаться из Elements в консоль и обратно.

Ещё методы консоли

Для поиска элементов в консоли есть два специальных метода:

- `$$("div.my")` – ищет все элементы в DOM по данному CSS-селектору.
- `$("div.my")` – ищет первый элемент в DOM по данному CSS-селектору.

4. Навигация по DOM-элементам

DOM позволяет делать что угодно с HTML-элементом и его содержимым, но для этого нужно сначала нужный элемент получить. Доступ к DOM начинается с объекта `document`. Из него можно добраться до любых узлов.

Сверху `documentElement` и `body`

Самые верхние элементы дерева доступны напрямую из `document`.

`<HTML> = document.documentElement`

Первая точка входа – `document.documentElement`. Это свойство ссылается на DOM-объект для тега `<html>`.

`<BODY> = document.body`

Вторая точка входа – `document.body`, который соответствует тегу `<body>`.

Дети: `childNodes`, `firstChild`, `lastChild`

Здесь и далее мы будем использовать два принципиально разных термина.

- **Дочерние элементы (или дети)** – элементы, которые лежат *непосредственно* внутри данного. Например, внутри `<HTML>` обычно лежат `<HEAD>` и `<BODY>`.
- **Потомки** – все элементы, которые лежат внутри данного, вместе с их детьми, детьми их детей и так далее. То есть, всё поддерево DOM.

Псевдо-массив `childNodes` хранит все дочерние элементы, включая текстовые.

Коллекции – не массивы

DOM-коллекции, такие как `childNodes` и другие, которые мы увидим далее, не являются JavaScript-массивами.

В них нет методов массивов, таких как `forEach`, `map`, `push`, `pop` и других.

Навигация только по элементам

Навигационные ссылки, описанные выше, равно касаются всех узлов в документе. В частности, в `childNodes` сосуществуют и текстовые узлы и узлы-элементы и узлы-комментарии, если есть.

Эти ссылки похожи на те, что раньше, только в ряде мест стоит слово `Element`:

- `children` – только дочерние узлы-элементы, то есть соответствующие тегам.
- `firstElementChild`, `lastElementChild` – соответственно, первый и последний дети-элементы.
- `previousElementSibling`, `nextElementSibling` – соседи-элементы.
- `parentElement` – родитель-элемент.

В DOM доступна навигация по соседним узлам через ссылки:

- По любым узлам.
- Только по элементам.