

Введение в Javascript. Введение в DOM

Frontend. Урок 21

Вопросы на повторение

1. Что такое класс и объект?
2. Что такое инкапсуляция?
3. Для чего нужны геттеры и сеттеры?

Постановка проблемы

Как вы думаете, можно ли управлять HTML-разметкой из JavaScript-кода?

DOM

DOM (Document Object Model) — это специальная древовидная структура, которая позволяет управлять HTML-разметкой из JavaScript-кода.

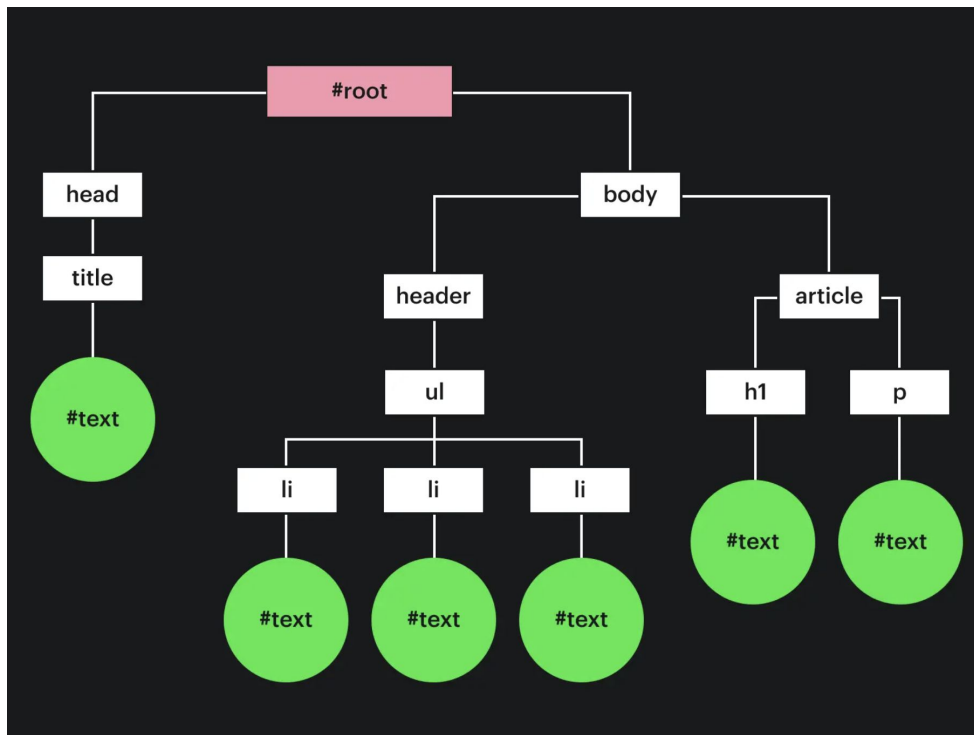
Управление обычно состоит из добавления и удаления элементов, изменения их стилей и содержимого.

Браузер создает DOM при загрузке страницы, складывает его в переменную `document` и сообщает, что DOM создан, с помощью события `DOMContentLoaded`. С переменной `document` начинается любая работа с HTML-разметкой в JavaScript.

Посмотрим на основные свойства

- `title`
- `body`
- `head`

DOM-дерево



DOM-дерево

Дерево состоит из обычных и текстовых узлов. Обычные узлы — это HTML-теги, а текстовые узлы — текст внутри тегов.

Обычный узел называется Element, и он содержит в себе описание тега, атрибутов тега и обработчиков. Если изменить описание — изменится и HTML-код этого элемента (возможно что-то даже изменится на экране. Например, если поменять цвет шрифта).

У любого узла есть один родительский узел и дочерние. Родительский узел — элемент, в который вложен текущий узел, он может быть только один.

Дочерние — узлы, которые вложены в текущий узел.

Это правило не работает только в двух случаях:

корневой узел — у такого узла нет родителя;

текстовый узел — у таких узлов нет дочерних узлов, только родитель.

Последний уровень любого DOM-дерева состоит из текстовых узлов.

Методы DOM дерева:

DOM позволяет нам делать что угодно с элементами и их содержимым, но для начала нужно получить соответствующий DOM-объект.

- `getElementById` — поиск элемента по идентификатору;
Если у элемента есть атрибут `id`, то мы можем получить его вызовом `document.getElementById(id)`, где бы он ни находился.
- `getElementsByClassName` — поиск элементов по названию класса;
Здесь ситуация меняется, если `id` значение каждый раз уникальное, тоо класс, как правило применяется сразу к нескольким элементам, поэтому работая с таким методом, мы будем работать с массивом
- `querySelector` — поиск первого элемента, подходящего под CSS-селектор;
- `querySelectorAll` — поиск всех элементов подходящих под CSS-селектор.

Свойства и методы, связанные с DOM

Пример: <https://doka.guide/js/element/demos/dom/>

- children — список дочерних элементов;
- parentElement — получить родительский элемент;
- nextElementSibling и previousElementSibling — получить следующий/предыдущий узел-сосед

Свойства с информацией о содержимом

- `innerHTML` — это свойство возвращает HTML-код всего, что вложено в текущий элемент. При записи в это свойство, предыдущее содержимое будет затерто.
- `textContent` — свойство, возвращает текст всех вложенных узлов без HTML-тегов.

Если нужно добавить текст в элемент, то всегда используйте свойство `textContent`. Другие свойства обрабатывают HTML, это может привести к дырам в безопасности.

Для работы с классами элемента есть удобные методы, доступные через свойство `classList`:

добавить класс — метод `add()`.

удалить класс — метод `remove()`.

Дополнительная теория к практике

.addEventListener()

Учим приложение реагировать на действия пользователя.

Найдем кнопку на странице и будем выводить сообщение в консоль, когда произошёл клик по этой кнопке.

```
const element = document.querySelector('button')
```

```
element.addEventListener('click', function (event) {
```

```
  console.log('Произошло событие', event.type)
```

event.type — строка, содержащая название события. Наиболее популярные события 'click', 'change', 'submit', 'keydown', 'keyup', 'mousemove', 'mouseenter', 'mouseleave'.

handler — функция, которая будет вызвана, когда событие произойдёт.

Дополнительная теория к практике

Стрелочные функции

Существует еще один очень простой и лаконичный синтаксис для создания функций, который часто лучше, чем Function Expression.

Он называется «функции-стрелки» или «стрелочные функции» (arrow functions), т.к. выглядит следующим образом:

```
let func = (arg1, arg2, ...argN) => expression;
```

Практика А

Создайте кнопку, при клике на которую добавляется 1 красный квадрат на экран (нужно, чтобы кнопка умела добавлять много квадратов - 1 клик = 1 квадрат)

Практика В

Измените кнопку, чтобы она задавала случайный цвет квадрату. Для каждого квадрата добавьте удаление при клике на него самого.

Практика С

Добавьте возможность отрисовки сообщений в наш макет мессенждера с помощью скрипта:

- Создайте массив сообщений
- Изучите работу метода `forEach` у массивов
- Отрисуйте сообщения в зависимости от того, кто его прислал