



Занятие №16

Взаимодействие с HTML. BOM, DOM



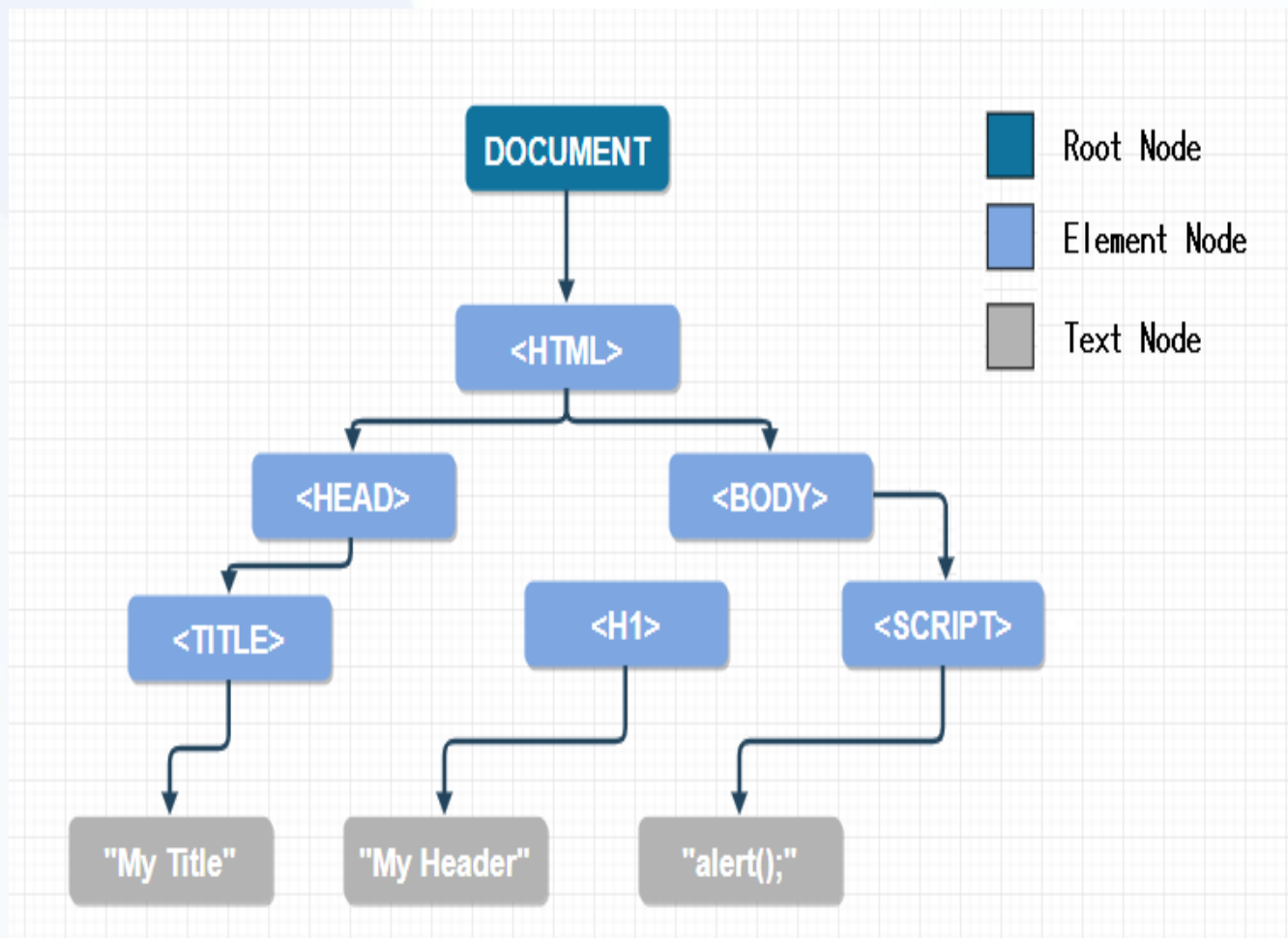


Как вы думаете, что из себя
представляет HTML страница?

DOM

DOM - Document Object Model - структура, хранящая все HTML элементы в браузере.

Образно представляется в виде дерева объектов.



Узлы DOM

В DOM есть узлы (**Nodes**) - отдельные составляющие дерева, которые в совокупности определяют внешний вид страницы.

Есть несколько типов узлов, вот основные:

1. Элемент - это тег
2. Текстовый узел - любой текст на странице
3. Комментарий - то, что написано в `<!-- -->`

Корневой узел

Корневым узлом называется объект `document`, внутри которого находятся все остальные узлы.

Именно поэтому `document.write` дописывает код HTML

Поиск по дереву DOM

Есть несколько методов объекта `document`, позволяющие находить элементы внутри HTML страницы.

ВАЖНО! Поиск будет происходить в момент выполнения строки. Если на этот момент дерево DOM будет составлено не полностью, тогда поиск выполняться не будет. Поэтому первое время мы будем писать подключения тега `script` в конце `body`, чтобы на момент выполнения скрипта все узлы были готовы к работе.

Метод getElementById

```
document.getElementById("id")
```

Выполняет поиск по id элемента. id задается в скобках в виде строкового значения, например:

```
let el = document.getElementById("my-id");
```

Результат работы метода (возвращаемое значение) - один элемент, экземпляр класса Element, или null, если элемент с заданным id не найден.

Метод `getElementsByClassName`

`document.getElementsByClassName(class)`

Выполняет поиск по заданному классу. Класс задаётся в виде строкового значения, например:

```
let els = document.getElementsByClassName("my-class");
```

Результат работы метода (возвращаемое значение) - коллекция элементов, экземпляр класса [HTMLCollection](#).

То есть возвращаемое значение - это всегда набор (псевдомассив) элементов, даже если элементов с заданным классом нет. В данном случае псевдомассив будет пустой.

Метод `getElementsByTagName`

```
document.getElementsByTagName(tag)
```

Выполняет поиск по заданному имени тега. Имя тега задаётся в виде строкового значения, например:

```
let els = document.getElementsByTagName("div");
```

Результат работы метода (возвращаемое значение) - коллекция элементов, экземпляр класса `HTMLCollection`.

То есть возвращаемое значение - это всегда набор (псевдомассив) элементов, даже если элементов с заданным классом нет. В данном случае псевдомассив будет пустой.

Метод `querySelector`

`document.querySelector(selector)`

Выполняет поиск по заданному css-селектору. Селектор задаётся в виде строкового значения, например:

```
let els = document.querySelector(".my-class");
```

Результат работы метода (возвращаемое значение) - один элемент, экземпляр класса `Element`, или `null`, если элемент, соответствующий селектору, не был найден.

Переданный селектор должен быть правильным css-селектором, иначе будет возникать ошибка.

Метод `querySelectorAll`

```
document.querySelectorAll(selector)
```

Выполняет поиск по заданному css-селектору. Селектор задаётся в виде строкового значения, например:

```
let els = document.querySelectorAll(".my-class");
```

Результат работы метода (возвращаемое значение) - коллекция элементов, экземпляр класса `NodeList`.

В отличии от `querySelector`, `querySelectorAll` - это всегда коллекция, даже если элементов 1 или 0.

Изменение элементов

Изменение может производиться различными способами, мы рассмотрим сегодня только 3: изменение текстового узла, изменение внутреннего кода HTML, изменение стилей CSS.

Изменение текстового узла производится через свойство `textContent`

```
el.textContent = "Новый текст"
```

ВНИМАНИЕ: изменение текстового узла приведет к удалению вложенных тегов, если они есть.

Также, если задать в `textContent` код HTML, то он в итоге будет как набор символов, а не как элементы HTML

Изменение внутреннего кода HTML

Для изменения внутреннего кода HTML используется свойство `innerHTML`.

```
el.innerHTML = '<a href="https://ya.ru">YA.RU</a>';
```

Использовать `innerHTML` нужно аккуратно, иначе это может привести к беде.

Изменение стиля

Изменение стиля элемента производится следующим образом: вы прописываете свойство `style`, а у этого свойства вызываете свойство с соответствующим CSS-свойством.

```
el.style.color = "red";
```

Если название свойство записывается через дефис (как `background-color`), то его запись трансформируется в верблюжий стиль: убирается дефис, а следующее слово пишется с большой буквы.

```
el.style.backgroundColor = "green";
```

Задания

1. Напишите функцию, которая изменит выбранную карточку. Создать три карточки товаров (красивые). Обращаться по id. Передать несколько значений и изменить их.
2. Создать аналог магазина, где есть кнопка выбрать всё. Напишите функцию для этой кнопки, которая для всех товаров меняет задний фон. Обращаться нужно по классу.
3. Напишите свой слайдер для фотографии языков программирования (изображение и две кнопки для смены фотографий). Попробовать два варианта (тег `img` и `background-image`) Реализовать функцию обращаясь к самому тегу.
4. Напишите форму, которая позволяет добавлять несколько товаров. Аналог чека. Вывести данные объекта после отправки.



Давайте подведем итоги!
Чему мы научились?
Что мы использовали?