

## DOM. Семантика HTML-документа.

Как Вы уже знаете, HTML-документ имеет древовидную структуру, корнем которой является тэг `<html>`. Для того, чтобы разработчики могли взаимодействовать с каждым узлом этого дерева посредством JS, во всех существующих браузерах поддерживается один из интерфейсов Web API – Document Object Model (DOM).

DOM представляет HTML-документ как виртуальное дерево, каждый узел которого (в том числе и корневой) является объектом со своими методами, событиями и свойствами. Корневым узлом является объект класса `Document`. Остальные узлы наследуют класс `Element`.

Благодаря тому, что браузеры поддерживают DOM через Web API, Вы можете осуществлять манипуляции с узлами HTML-документа напрямую из JavaScript. Например, создать элемент, воспользовавшись методом объекта `Document` – `createElement()`. Также DOM предоставляет методы для установки стилей и отслеживания событий.

Мы не будем сейчас приводить все доступные методы работы с DOM. Вместо этого предлагаем Вам ознакомиться со статьей [1] и её подразделами (особенно подраздел «Introduction to the DOM»).

В процессе создания HTML-документа Вы используете различные тэги. Каждый тэг имеет некоторое значение – это может быть как специфичное значение (элемент текста, формы и т. д.), так и неспецифичное. Во втором случае, тэги предназначены для объединения элементов с целью удобства компоновки элементов или для выделения части текста с целью изменения его стилей. Специфичные тэги также называют семантическими.

Само по себе слово «семантика» означает смысл чего-либо. Семантика очень важна в программировании в целом. Скорее всего Вы уже знаете, насколько важна семантика имён переменных и функций (процедур). Если функция складывает два числа, то следует её назвать «add», а не, например, «func\_a». Также некоторые наиболее популярные CSS-значения цветов имеют семантические имена. Например, красный цвет задается в шестнадцатеричной системе счисления как `#FF0000`, в формате RGB как `(255, 0, 0)`, но также он может быть задан семантическим словом «red».

Отдельно рассмотрим семантику в HTML. Она заключается в использовании семантических тэгов, то есть таких, которые однозначно указывают значение элемента как браузеру, так и разработчику.

Вообще все тэги делятся на семантические – h1-h6, p, section, nav и др., а несемантических всего два – это span и div – они не говорят ничего о том, что данный элемент из себя представляет или что содержит.

В процессе верстки некоторого функционала Вы должны задумываться о значении каждого элемента и применить к нему соответствующий тэг. Однако, если по смыслу для данного элемента нет соответствующего тэга, то лучше всего будет использовать div или span. Специально искать и придумывать смысл не нужно, это может только ухудшить читаемость кода.

Для чего нужна семантика? В первую очередь для самих разработчиков. Если нужно поправить документ или добавить новые элементы, большая вложенность div-ов друг в друга заберёт у разработчика большое количество времени на воссоздание в уме примерной структуры части документа. Чего не будет в случае хорошо размеченной страницы.

Во-вторых, семантические тэги позволяют людям с ограниченными возможностями использовать скринридеры – специальные программы, которые преобразуют текст и элементы интерфейса на экране в речь или шрифт Брайля. Тут становится очевидно, что div-ы не дадут никакой ценной информации пользователю и Ваше веб-приложение будет ему недоступно.

В-третьих, хорошо структурированный документ понятен для поисковых роботов, индексирующих веб-страницы в сети Интернет. Подробнее о SEO вы можете почитать тут [3] (если интересно).

## Вопросы и задания

1. Изучите страницу Базы знаний Вовлекай [2]. Напишите разметку этой страницы с использованием семантических и несемантических тэгов. Сделайте это внутри тэга <body>:

<body>...your structure there...</body>

Обратите внимание, что некоторые элементы должны быть рассмотрены в контексте всего приложения, а не только этой страницы.

2. Создайте простую веб-страницу с функционалом, представленным на видео [4]. При нажатии на кнопку «Add item» создаётся элемент списка с порядковым номером. Максимальное кол-во элементов ограничено и равно 10. При достижении максимального кол-ва элементов кнопка «Add item» переходит в состояние «disabled». Если кол-во элементов равно нулю, что кнопка «Remove item» переходит в состояние «disabled».

Параметры элементов списка: радиус 12px, цвет фона lightgreen, отступы между элементами 8px. Отступ между кнопками и списком равен 16px.

#### Ссылки

- [1]. [https://developer.mozilla.org/en-US/docs/Web/API/Document\\_Object\\_Model](https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model)
- [2]. <https://labmedia.vovlekay.online/>
- [3]. <https://seo.co/semantic-html/>
- [4]. <https://disk.yandex.ru/i/5loeWiazn4bjBQ>