

Веб-программирование на стороне клиента

Кулаков Кирилл Александрович

Структура курса

- Продолжительность
 - 15 часов лекций
 - 15 часов практических занятий
- Контроль успеваемости
 - Выполнение практических заданий
 - Мини-проект
- Помощь
 - Сайт курса (<https://cs.petrSU.ru/~kulakov/courses/web-arch2/>)
 - График консультаций кафедры ИМО
 - Электронная почта (kulakov@cs.petrSU.ru)

Web как операционная система

- Операционная система = ядро + набор сервисов
- Интернет — неограниченный набор сервисов
- Идея: реализация веб-приложения, организующего платформу для работы с интернет-сервисами
 - Веб браузер реализует контейнер для выполнения программ
 - Веб браузер предоставляет графический интерфейс
 - Реализация концепции на базе внутренних возможностей браузера или с использованием сторонних технологий через плагины
 - Простой вариант: страница с html тегами iframe

Web как операционная система

- Примеры WebOS:
 - iCloud: виртуальный рабочий стол
 - AstraNOS: реализация веб версий стандартных приложений
 - Joli OS: интеграция популярных веб служб для просмотра контента
 - Silve OS: реализация веб версий стандартных приложений на базе silverlight
 - EyeOS: реализация рабочего стола на базе Adobe Flash

Web как операционная система



Web как операционная система

- Преимущества:
 - Единое хранилище приложений
 - Доступ из разных мест
 - Единое хранилище данных
- Недостатки:
 - Требуется интернет соединение
 - Работа на стороннем сервере
 - Единая точка отказа

Язык JavaScript

- Для работы в браузере необходим общепринятый метод выполнения операций
- Javascript — скриптовый (сценарный) интерпретируемый язык программирования
- JavaScript был изобретен Брендан Айх в 1995, и стал стандартом ECMA в 1997.
- ECMA-262 является официальным названием стандарта. ECMAScript является официальным названием языка.
- Интерпретатор встроен непосредственно в браузер
- Поддержка большинством современных браузеров
- Обработка скрипта (или другими словами — сценария) происходит на стороне клиента без обращения к серверной стороне

Язык JavaScript

- Версии
 - **JavaScript** был разработан для Netscape. Первый браузер для запуска JavaScript был Netscape 2 в 1996. После Netscape фонд Mozilla продолжил разработку JavaScript для браузера Firefox. Номера версий JavaScript запускаются с 1,0 до 1,8.
 - **ECMAScript** был разработан ECMA International после того, как Организация приняла JavaScript. Первое издание ECMAScript было выпущено в 1997. Номера версий ECMAScript запускаются от 1 до 7.
 - **JScript** был разработан Microsoft как совместимый язык JavaScript для Internet Explorer в 1996. Номера версий JScript выполняются с 1,0 по 9,0.

Язык JavaScript

Год	Имя	Описание
1997	ECMAScript 1	Первая версия
1998	ECMAScript 2	Небольшие правки
1999	ECMAScript 3	Регулярные выражения, try/catch
	ECMAScript 4	Не выпущена
2009	ECMAScript 5 (ES5)	«strict mode», JSON
2011	ECMAScript 5.1	Небольшие правки
2015	ECMAScript 6 (ECMAScript 2015, ES6)	Расширен функционал
2016	ECMAScript 7 (ECMAScript 2016)	экспоненциальный оператор (* *), массив, prototype, ...
2017	ECMAScript 8 (ECMAScript 2017)	Асинхронные функции, общая память, ...

Язык JavaScript

- Объекты JavaScript
 - Global — глобальные функции и свойства JavaScript
 - Operators — арифметические операторы
 - Statements — операторы JavaScript (break, for, while, ...)
 - Error — объект сведения об ошибке
 - Math — объект математических операций
 - Array — Объект массива
 - Boolean — Булевый объект
 - Date — объект для работы с датами и временем
 - JSON — объект работы с JSON
 - Number — объект числа
 - RegExp — объект регулярного выражения
 - String — объект строки

Язык JavaScript

- Где почитать?
 - <https://schoolsw3.com/js/index.php>
 - <https://html5css.ru/js/default.php>
 - <https://labs-org.ru/567-2/javascript/>
 - http://www.msiit.ru/x/eumkit/website___1.html
 - ...

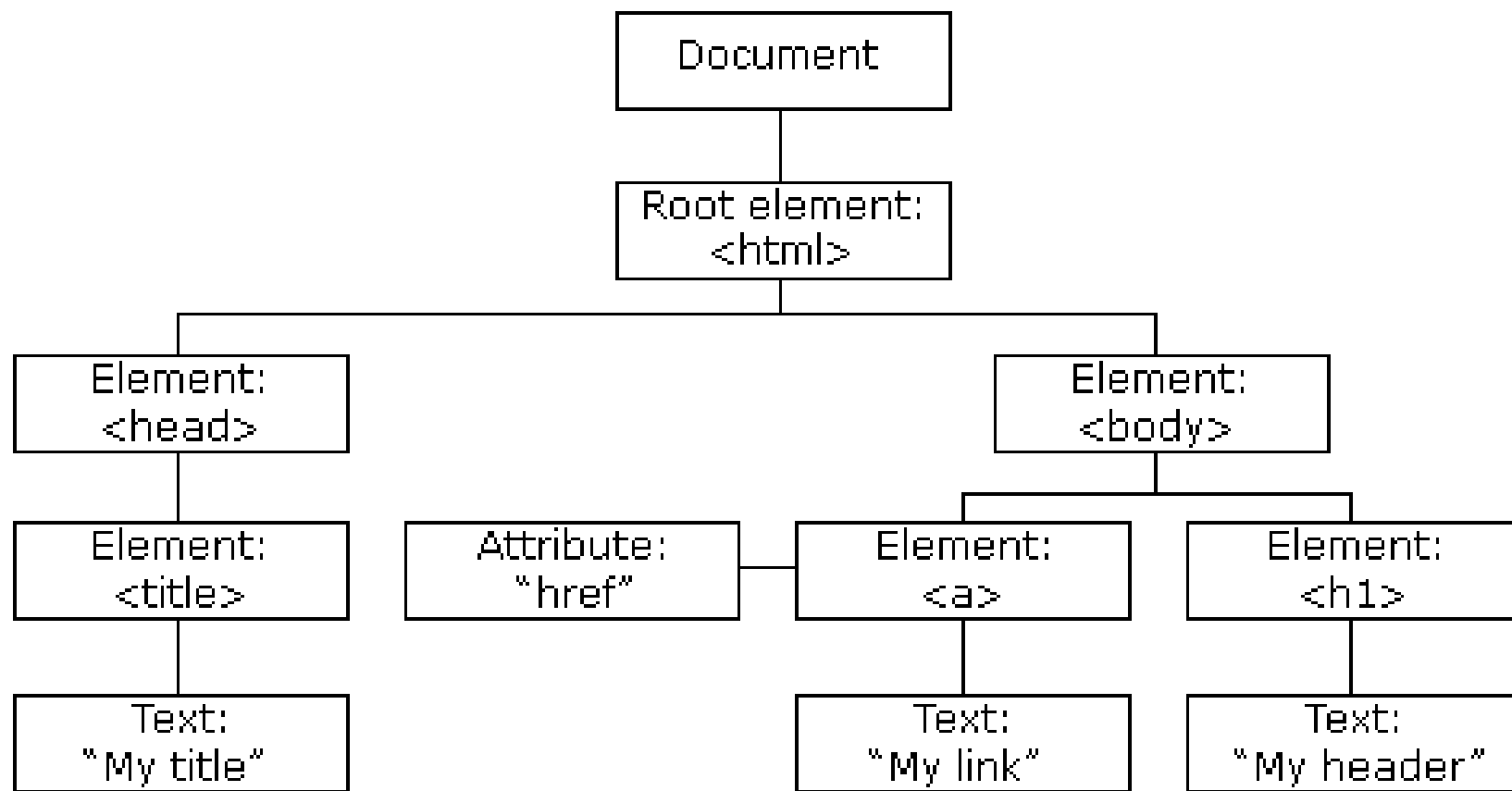
DOM

- Исходный код страницы → ??? → представление страницы на экране
=> нужен интерфейс для изменения содержания страницы
- Объектная модель документа W3C (DOM) - это не зависящий от платформы и языка интерфейс, который позволяет программам и сценариям динамически получать доступ и обновлять содержимое, структуру и стиль документа.
- DOM - это стандарт консорциума W3C (World Wide Web Консорциума)

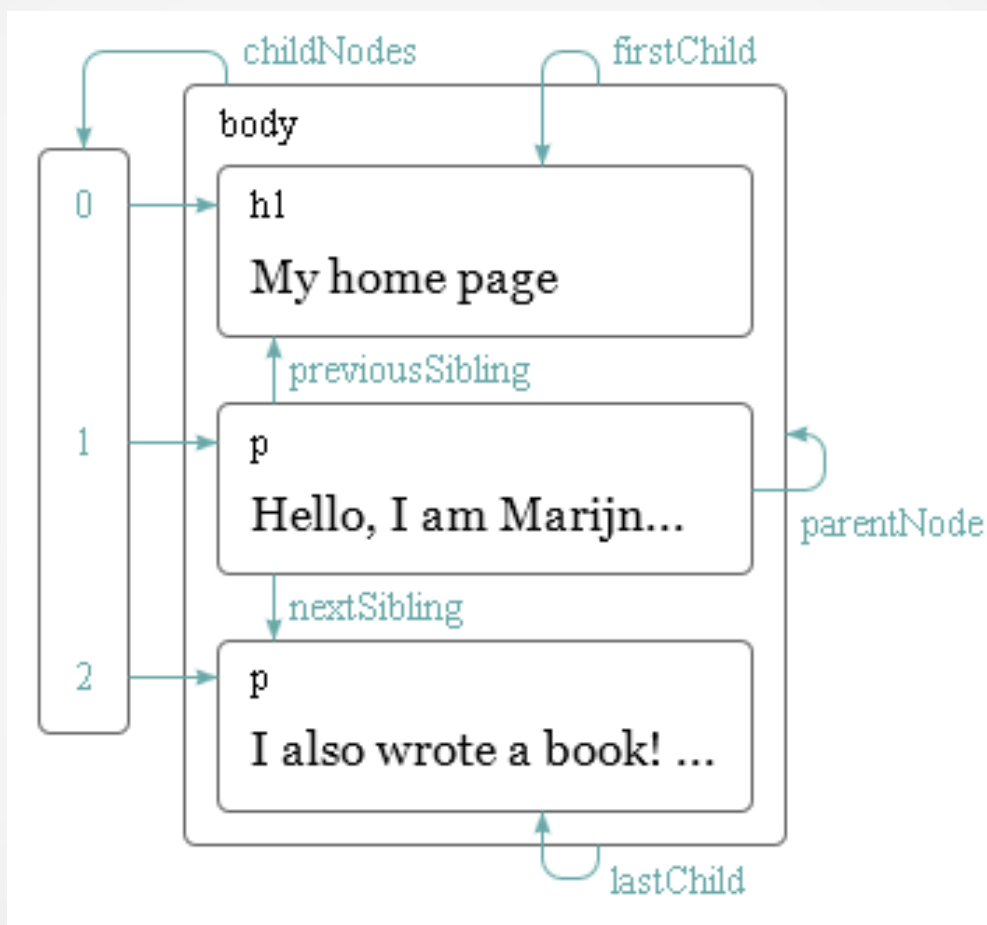
DOM

- Части DOM
 - Core DOM - стандартная модель для всех типов документов
 - XML DOM - стандартная модель для XML документов
 - HTML DOM - стандартная модель для HTML документов
- HTML DOM:
 - HTML элементы как объекты
 - В свойствах для всех элементов HTML
 - Методы, чтобы получить доступ ко всем HTML элементам
 - В событиях для всех элементов HTML

DOM



DOM



DOM

- Получение элементов DOM
 - **getElementsByTagName()** - возвращает коллекцию объектов `HTMLElement` по заданному имени тега;
 - **getElementsByClassName()** - возвращает коллекцию объектов `HTMLElement` по заданному имени CSS-класса;
 - **getElementById()** - возвращает HTML-элемент с заданным идентификатором, если на странице присутствуют два тега с одним и тем же идентификатором - будет возвращен первый найденный элемент;
 - **querySelector()** - возвращает первый элемент, удовлетворяющий набору произвольных CSS-селекторов.

DOM

- Изменение элементов DOM
 - Каждый узел DOM — объект, содержащий набор методов и свойств
 - Изменение элемента = изменение свойства
- Удаление элементов из DOM
 - **removeChild()** - вызывается применительно к элементу, у которого необходимо удалить прямой дочерний элемент, аргументом принимает удаляемый HTML-тег и возвращает ссылку на него;
 - **remove()** - вызывается применительно к элементу, который необходимо удалить из DOM.

DOM

- Создание элементов DOM
 - Для создания нового HTML-тега в JavaScript используется метод **document.createElement()**, который единственным параметром принимает наименование тега и возвращает объект типа `HTMLElement`.
 - **appendChild()** - принимает элемент, добавляет его прямым последним дочерним элементом к элементу, применительно к которому был вызван метод и возвращает ссылку на привязанный HTML-тег;
 - **insertBefore()** - принимает в качестве аргументов два элемента: который необходимо вставить в DOM и после которого необходимо вставить первый, а сам метод вызывается применительно прямого родителя переданных в качестве параметров элементов;
 - **after()** - принимает в качестве аргумента элемент, который необходимо вставить после элемента, относительно которого вызывается сам метод;

DOM

- Атрибуты элементов
 - **getAttribute()** - принимает наименование атрибута и возвращает его значение;
 - **setAttribute()** - принимает наименование и значение атрибута и устанавливает его для элемента, относительно которого вызывается, если атрибут уже существует, то просто перезаписывает его значение;
 - **hasAttribute()** - принимает наименование атрибута и возвращает значение логического типа: true - указанный атрибут задан у элемента, false - не задан;
 - **removeAttribute()** - принимает наименование атрибута и удаляет его у элемента, относительно которого вызывается.