

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»

Новоуральский технологический институт –

филиал федерального государственного автономного образовательного учреждения
высшего образования «Национальный исследовательский ядерный университет «МИФИ»

(НТИ НИЯУ МИФИ)

Колледж НТИ

Цикловая методическая комиссия информационных технологий

КОНТРОЛЬНАЯ РАБОТА

ПО ТЕМЕ

«КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ»

ПМ.03 «Участие в интеграции программных модулей»

МДК 03.01 «Документирование и сертификация»

Специальность СПО 09.02.03

«Программирование в компьютерных системах»

очная форма обучения

на базе основного общего образования

Выполнил

студент группы КПр–47 Д

Иванов И.И.

дата

подпись

Проверил

преподаватель

Горницкая И.И.

дата

подпись

Новоуральск 2020

Цель разработки HTML5

Для разработчиков html5 помогает писать понятный семантический код. Позволяет управлять многими процессами на странице своими стандартными методами, без использования javascript или сторонних плагинов и сервисов. Это означает, что решаются некоторые проблемы кроссбраузерности, поскольку браузеры одинаково реализуют новые возможности.

концепция HTML5

В HTML5 контент страницы можно разделить на смысловые группы, которые характеризуют содержимое веб-страницы. Эти группы получили название контентных моделей (content model). Каждая модель описывает тип содержимого элемента. Содержимым элемента является текст и дочерние элементы. Элемент может принадлежать как к одной, так и к нескольким категориям.

понятие логотипа HTML5

Логотип является ключевым элементом дизайна сайта и важен для бизнеса в целом и о том, что такое эффективный логотип, а также как его создать, мы сегодня и поговорим. ... Конечно же, для этого лого должен быть эффективным. И о том, что такое эффективный логотип, а также как его создать, мы сегодня и поговорим! Немного теории: понятие эффективности логотипа

Особенности HTML5

Теги (исключено, введено)

HTML5 активно взаимодействует с CSS, поэтому запрет на многие атрибуты, начатый в HTML4 в пользу стилей, только усилился. Включение новых элементов вроде <video> и <audio> также сводит на нет теги, на которые возлагалась функция мультимедиа. По этой причине некоторые теги перестали поддерживаться или были заменены:

<applet> — добавляет Java-апплет в документ. Вместо него следует использовать <embed> или <object>.

<acronym> — этот тег вызывал постоянные вопросы, что такое акроним и чем он отличается от аббревиатуры. Для упрощения остался единственный тег <abbr>.

<bgsound> — определяет музыкальный файл, который будет проигрываться на веб-странице при её открытии. Для воспроизведения музыки используйте новый элемент <audio>.

<dir> — создает список, содержащий названия директорий, вместо него используйте .

<frame>, <frameset>, <noframe> — фреймы более не поддерживаются. Если они вам требуются, используйте другую версию HTML или <iframe> совместно со стилями.

<isindex> — предназначен для поискового индекса в текущем документе. Комбинация <form> и <input> лучше справляется с этой задачей.

<listing>, <xmp> — для вывода листинга программы предназначены <pre> и <code>.

<nextid> — этот тег не предназначен для людей и указывает идентификатор следующего документа для автоматических редакторов HTML. Полностью исключён.

<noembed> — предназначен для отображения информации на веб-странице, если браузер не поддерживает работу с плагинами. В качестве альтернативы используйте <object>.

<plaintext> — отображает содержимое контейнера «как есть», любые теги выводятся как текст. Вместо тега используйте MIME-тип text/plain.

<rb> — определяет базовый текст внутри <ruby>. Этот тег полностью исключён.

<strike> — для зачёркнутого текста применяется <s>, а для указания редакторской правки .

<basefont>, <big>, <blink>, <center>, , <marquee>, <multicol>, <nobr>, <spacer>, <tt>, <u> — вместо этих тегов управляющих видом текста применяются стили.

Быль также введены в HTML5 несколько новых тегов: <article>, <aside>, <footer>, <header>, <nav>, которые заменяют в некоторых случаях привычный <div>. Хотя кажется, что особой разницы между тегами <div class="header"> и <header> нет, между ними лежит огромная пропасть. Теги ориентированы не на людей, которым нет смысла заглядывать в исходный код страницы, а на машины, интерпретирующие код. Машины или роботы не понимают <div class="header">, для них это типовой тег разметки — замени его на <div class="abracadabra"> и смысл не поменяется. Другое дело <header>, робот, обнаружив этот тег, воспринимает его именно как шапку сайта или раздела.

Интерактивность

В HTML 5 появились несколько различных интерактивных элементов, которые позволяют изменять страницу без её перезагрузки, что очень полезно для создания динамических сайтов.

Вот список интерактивных элементов появившихся в HTML 5:

1. menu.
2. command.
3. datagrid.
4. details.

Мультимедиа возможности

В HTML 5 имеются два элемента для работы с мультимедиа:

1. Audio.
2. Video.

Первый элемент, как следует из названия, служит для добавления аудио-файлов. Простейший пример использования элемента audio в HTML 5.

II

поддержка технологии WebSockets

WebSocket — это самое кардинальное расширение протокола HTTP с его появления. Это не финтифлюшки, это сдвиг парадигмы HTTP. Изначально синхронный протокол, построенный по модели «запрос — ответ», становится полностью асинхронным и симметричным. Теперь уже нет клиента и сервера с фиксированными ролями, а есть два равноправных участника обмена данными. Каждый работает сам по себе, и когда надо отправляет данные другому. Отправил — и пошел дальше, ничего ждать не надо. Вторая сторона ответит, когда захочет — может не сразу, а может и вообще не ответит. Протокол дает полную свободу в обмене данными, вам решать как это использовать.

Передача данных между доменами

В стандарте HTML5 предусмотрена отсылка javascript-сообщения из одного окна в другое при помощи специального вызова window.postMessage.

При этом окна могут быть с любых доменов, не важно. Например, один iframe с yandex.ru , а другой - с vaysarupkin.info. Чтобы получать кросс-доменные сообщения, принимающее окно регистрирует специальный обработчик onmessage, в котором, кроме всего прочего, может проверить, с какого домена пришло сообщение.

Хранение с данными на стороне клиента

Главное отличие различных типов хранилищ – время хранения данных и их доступность. sessionStorage хранит данные в рамках одной сессии (посещения, т.е. до закрытия пользователем окна браузера).

localStorage позволяет хранить данные и после прекращения сеанса.

С точки зрения программирования различие в использовании сеансового и локального типов хранилищ сводится к различию имен объектов, посредством которых осуществляется доступ к ним: `sessionStorage` и `localStorage` соответственно.

Возможности web-форм

формы являются элементами управления, которые применяются для сбора информации от посетителей веб-сайта.

Веб-формы состоят из набора текстовых полей, кнопок, списков и других элементов управления, которые активизируются щелчком мыши. Технически формы передают данные от пользователя удаленному серверу.

Для получения и обработки данных форм используются языки веб-программирования, такие как PHP, Perl.

До появления HTML5 веб-формы представляли собой набор нескольких элементов `<input type="text">`, `<input type="password">`, завершающихся кнопкой `<input type="submit">`. Для стилизации форм в разных браузерах приходилось прилагать немало усилий. Кроме того, формы требовали применения JavaScript для проверки введенных данных, а также были лишены специфических типов полей ввода для указания повседневной информации типа дат, адресов электронной почты и URL-адресов.

Конфиденциальность

С появлением оффлайн-веб-приложений границы безопасности изменились. В веб-приложениях до HTML5 все решения, связанные с контролем доступа к данным и функциям, делались на стороне сервера. С появлением оффлайн-веб-приложений часть проверок прав доступа переместилась на сторону ПА. Для оффлайн-веб-приложений, таким образом, не достаточно реализации защитных мер исключительно на сервере. Цель атаки на веб-приложение больше не ограничена сервером, становится возможным атаковать и клиентскую часть приложения.

Это главным образом нарушает требование "Защита ПА". Однако, при нарушении данного требования неявной угрозе подвергаются все остальные требования безопасности. Например, если имеется возможность нарушить требование «Безопасное кэширование», атакующий сможет внедрить в кэш оффлайн-веб-приложения любое содержимое и использовать его для нарушения других требований безопасности.