**Ресурсы для подготовки к собеседованию**

Ссылки:

<https://jsehelper.blogspot.com/2016/01/html.html>

<https://ru.education-wiki.com/8929835-css-interview-questions>

<https://jsehelper.blogspot.com/2016/01/css.html>

<https://www.internet-technologies.ru/articles/40-vazhnyh-voprosov-i-otvetov-po-html5.html>

Так ладно. Разберу все 4 ссылки.

**Что такое HTML**

HTML(HyperТext Markup Language, HTML) - это язык разметки. Вы используете HTML для разметки текстового документа. Иными словами он отвечает за расположение в документе Ваших текстов, рисунков, таблиц предназначенных для жизни в сети Интернет.

**Что такое DOCTYPE и зачем он нужен?**

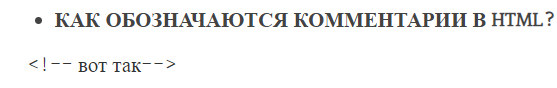
DOCTYPE - это определение типа документа (Document Type Definition (DTD)), правила, в соответствии с которыми осуществляется проверка конкретного документа (веб-страницы) XML или (X)HTML.

Благодаря этой записи, браузер определяет, какая в данном документе используется версия DTD. А нужен он для того, чтобы браузеры правильно отображали разметку документа. Если не указать DOCTYPE, то браузер будет добавлять "что то от себя", причём каждые браузеры будут добавлять свою "что то от себя". В результате, ни о какой кроссбраузерности не может идти и речи.

**Чем отличается div от span?**

**Div принято использовать как контейнер для стилизации и выравнивания элементов внутри этого контейнера. А span принято использовать именно для стилизации текста.**

div - это блочный контейнер, а span - линейный. Контейнеры - это то, во что может быть вложено что-нибудь еще. div используется для разметки блоков, а span - для текста. div формирует блок из того, что в нем с новой строки и после него элемент идет с новой строки (если не поменять поведение с помощью CSS), а span не переносит и "обтягивает".



**Как оставить ссылку на имейл?**

**<a href="mailto:testmail@mail.com">E-mail me</a>**

**Что пишут в теге HEAD?**

Тег title: Это очень важный тег, особенно с точки зрения SEO. Как он прописывается? С помощью тегов <title></title>.

Мета-теги. Мета теги, в основном, нужны для трёх вещей: прописать кодировку, description (описание) и keywords (ключевые слова).

Также внутри тегов head прописывается путь к таблице стилей, так связывается веб-страница и таблица стилей, или CSS пишется прямо на странице, внутри тегов style.

Внутри тега head подключаются всевозможные скрипты. Как они подключаются? Прописываются теги script с указанием типа языка и внутри них или пишется код скрипта, или к тегу script прописывается атрибут src, в котором указывается путь к скрипту.

**Зачем нужны теги <dl>,<dt>,<dd>?**

Теги <dl>,<dt>,<dd> предназначенны для создания списка определений. Каждый такой список начинается с контейнера <dl>, куда входит тег <dt> создающий термин и тег <dd> задающий определение этого термина. Закрывающий тег </dd> не обязателен, поскольку следующий тег сообщает о завершении предыдущего элемента. Тем не менее, хорошим стилем является закрывать все теги.

**В каком регистре лучше писать HTML-код?**

Раньше считалось, что это безразлично. Но с приходом XHTML на этот вопрос появился однозначный ответ - все теги, атрибуты и предопределенные значения пишем в нижнем регистре.

**Обязательно ли писать alt в <img>?**

Да. Этого требует стандарт, да и здравый смысл, т.к. некоторые пользователи все еще отключают картинки, а некоторые используют текстовые или даже голосовые браузеры. Если картинка не несет никакой смысловой нагрузки (элемент дизайна, "распорка" и т.п.), то можно поставить в тег <img> значение alt="".

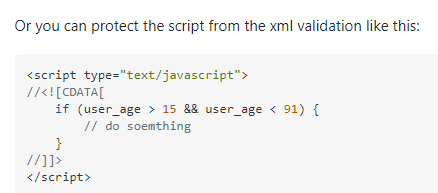
**Что такое entities?**

Entities - это комбинации знака & и буквенного или цифрового кода после нее, предназначенные для замещения символов, которые не могут встречаться в "чистом" виде в HTML-тексте, например, символа "<".

Думаю это в тему.





**CSS Интервью Вопросы**

Cascading Style Sheets(CSS) - каскадные таблицы стилей,

**Каковы преимущества CSS?**

Это помогает быстрее загружать страницы, так как нет необходимости каждый раз писать атрибуты HTML-тега. В CSS это может быть объявлено один раз, а cab используется много раз.

Это также помогает сэкономить время, поскольку ту же таблицу стилей CSS можно использовать и на других страницах HTML.

Легко поддерживать, если есть какие-либо изменения в стиле, это можно сделать в одном месте, и нет необходимости вносить изменения в разных местах.

Это помогает улучшить внешний вид HTML-страницы по сравнению с HTML-атрибутами или тегами.

Он помогает использовать один и тот же HTML-документ на разных устройствах и в разных версиях веб-сайта.

Это помогает в работе в автономном режиме. Он хранит приложение в автономном кеше, что также помогает ускорить загрузку и повысить производительность.

Он также не зависит от платформы. Его также можно просматривать в последних браузерах.

**Объяснить компоненты CSS?**

Селектор - это HTML-тег, к которому будет применяться стиль. Теги похожи

и т.п.

Свойство - это тип атрибутов тега HTML, которые преобразуются в свойства CSS. Примеры этого - граница, цвет и т. Д.

Значения, которые присваиваются свойствам, например, определяют любое значение цвета или любое другое определенное свойство в CSS.

**Объясните различные виды селекторов и способы нацеливания на элементы?**

Различные виды селекторов - это теги, id и классы. Тег в CSS помогает ссылаться на тег HTML. Идентификатор помогает ссылаться на атрибут идентификатора тега HTML, а класс помогает ссылаться на атрибут класса в теге HTML.

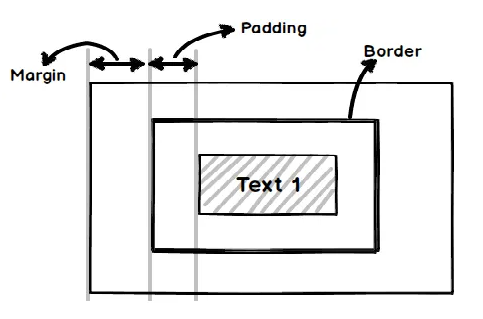
Различные способы нацеливания на элементы - это тег, класс и идентификатор. С помощью тега он может быть нацелен на то, чтобы тело и класс были указаны в HTML, это может быть цель с .classname. Идентификатор может быть ориентирован с помощью «#ID»,

**Объясните блочную модель CSS и ее элементы?**

Блочная модель CSS состоит из следующих элементов: Margin, Border, padding и content. Поле - это самый верхний слой, в котором описана общая структура. Граница находится между полем и отступом или может быть определена как заполнение и содержимое вокруг границы. Цвет фона влияет на границу. Отступы - это пространство между границей и содержимым. Это означает, что если элемент имеет цвет фона и этот цвет также будет заполнять отступы. Содержимое - это фактические данные, которые отображаются.

**В чем разница между margin и padding?**

margin - внешний отступ (снаружи от границ блока до остальных элементов страницы); padding - внутренний отступ (внутри от границ блока до контента). Допустим, у нас есть некий объект с рамкой вокруг него. Вот margin - это внешнее поле, а padding внутреннее. Или есть у нас абзац текста. Следовательно, margin - это отступ от других соседних объектов (левой, правой стороны экрана, соседних абзацев, разных окружающих картинок). А padding - это внутренние отступы. Если мы зададим padding: 50px для этого абзаца, то значит, сами буквы будут начинаться не от края абзаца, а с отступом в 50 пикселей.



**Объясните термин изящная деградация?**

В CSS, если какой-либо компонент выйдет из строя, постепенная деградация помогает компоненту работать должным образом. При разработке веб-страницы используется новейшее браузерное приложение. Если изображение не может быть просмотрено, его текст может быть показан.

**Ну то есть тот же атрибут alt у тега img.**

**Каковы определенные свойства измерения?**

Определенные свойства измерения: высота, минимальная высота, максимальная высота, ширина, минимальная ширина и максимальная ширина.

**Объясните различные типы носителей?**

Используются различные типы носителей: экранные, проекционные, звуковые, печатные и портативные. CSS можно использовать для разных экранов, таких как экран компьютера или ноутбука. Его можно использовать для портативного устройства, например мобильного телефона.

**Объясните, как обрабатывать различия браузеров в CSS?**

Последние браузеры обновляются в основном всеми функциями для работы с веб-сайтом или веб-приложениями. Между браузерами всегда есть разница, так как некоторые из них поддерживают стили CSS, или некоторые не поддерживают тот же стиль. Таким образом, всегда есть предел стиля, который можно использовать с использованием резервных стилей. Функция **@support** в CSS помогает определить определенную функцию, доступную в браузере или нет. Если это возможно, используйте другой альтернативный стиль.

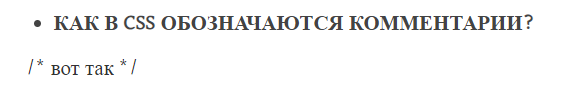
**Объясните Flexbox и CSS сетку?**

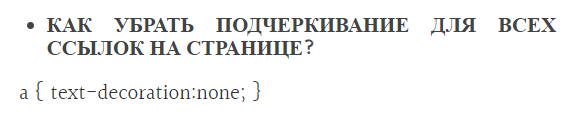
Flexbox - это инструмент верстки, который очень полезен в небольших областях сайта. Главной особенностью этого является выравнивание элементов по горизонтальной или вертикальной оси. Он используется для автоматического разметки элемента с другими параметрами макета и в определенном порядке для отображения элементов.

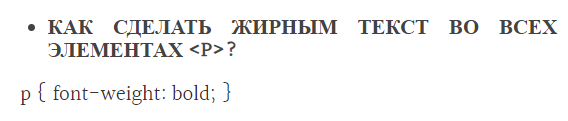
CSS-сетка - это почти инструмент для разметки одной страницы. Он используется для макетов как по горизонтальной, так и по вертикальной оси, в отличие от инструмента flexbox, который используется только для одной оси за раз.

**Какая разница между значениями 0 и auto в свойстве margin?**

В вертикальных полях, auto всегда означает 0. В горизонтальных полях, auto означает 0 только тогда, если свойство width также auto.

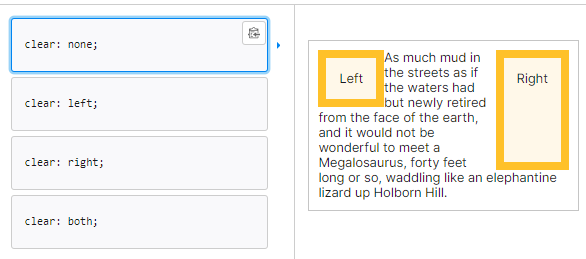


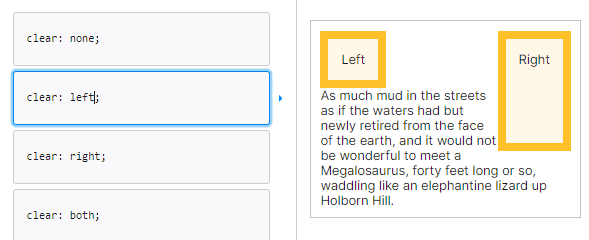


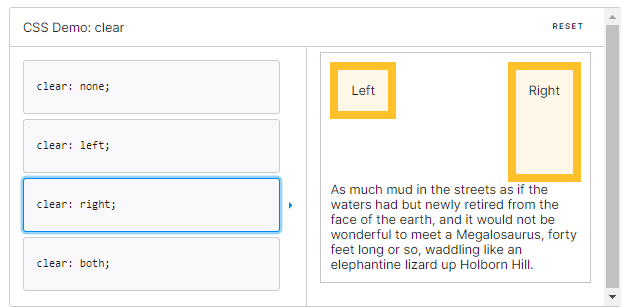


**Что делает свойство clear?**

Устанавливает, с какой стороны элемента запрещено его обтекание другими элементами. Если задано обтекание элемента с помощью свойства float, то clear отменяет его действие для указанных сторон.







**40 важных вопросов и ответов по HTML5**

Ой, много. Но ладно. Давай.

**Какая связь между SGML, HTML, XML и XHTML?**

SGML (Standard generalized markup language - стандартный обобщённый язык разметки) – это стандарт, который определяет разметку документа.

HTML – это язык разметки, который описывается с помощью SGML.

Итак, с помощью SGML было создано DTD (определение типа документа), на которое ссылается и которого должен придерживаться HTML. Поэтому вы всегда можете найти декларацию «DOCTYPE» в начале страницы HTML, которая определяет, какое DTD будет использовать браузер при разборе кода страницы.

Разбор кода по стандарту SGML был связан с определёнными трудностями, поэтому был создан XML, чтобы облегчить эту процедуру. XML использует SGML.

Например, в SGML вы должны использовать открывающие и закрывающие теги, тогда как в XML вы можете использовать самозакрывающиеся теги, которые закрываются автоматически (например, “”).

XHML был создан из XML и использовался в HTML 4.0. Поэтому, например, в HTML, основанном на SGML, тег

недопустим, а в XHTML он допускается. Вы можете использовать XML определение документа, как показано в следующем примере:





**Вкратце, SGML стоит в основе всего. Старые версии HTML основаны на SGML, а HTML 4.0 использует XHTML, построенный на основе XML.**

**Что такое HTML 5?**

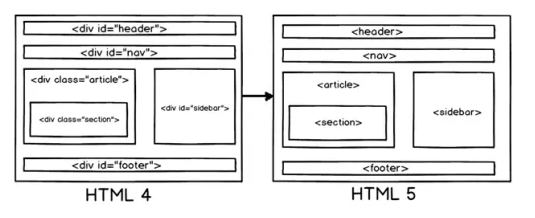
HTML 5 – это новый стандарт HTML, главной целью которого является предоставление любого контента без использования дополнительных плагинов, таких как Flash, Silverlight и т.д. Он содержит всё необходимое для отображения анимации, видео, богатого графического интерфейса и прочего.

HTML5 – это результат совместной работы World Wide Web Consortium (W3C) и Web Hypertext Application Technology Working Group (WHATWG).

HTML 5 не использует SGML или XHTML. Это полностью новая разработка, поэтому вам не нужно ссылаться на DTD. В HTML 5 вам нужно использовать следующую декларацию doctype, которая позволяет браузеру идентифицировать документ как HTML 5.



Появились новые теги.



<header>: Представляет блок заголовка страницы;

<footer>: Подвал страницы;

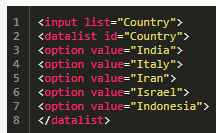
<nav>: Навигационные элементы страницы;

**<article>: Основной контент страницы (статья);**

<section>: Используется внутри статьи, чтобы разделить статью на секции;

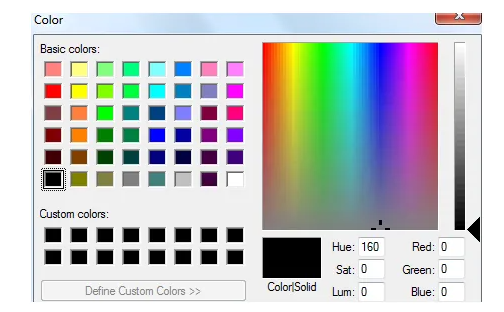
<aside>: Представляет боковой блок страницы.

**О новая структура, не знал о такой. Выглядит удобно.**

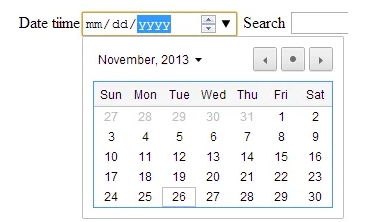
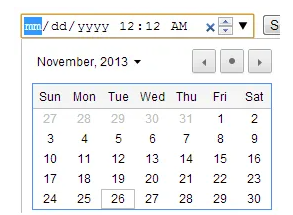


**Важные новые элементы формы:**

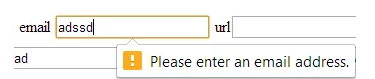
****

****

** **

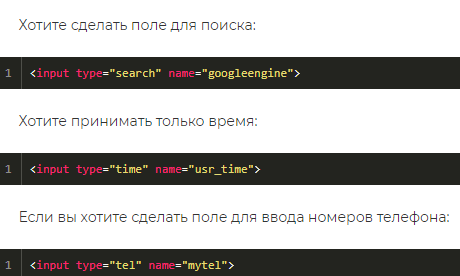
** **

** **

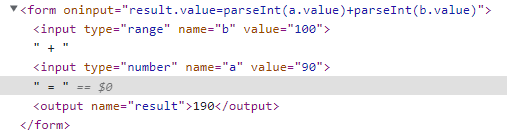
** **

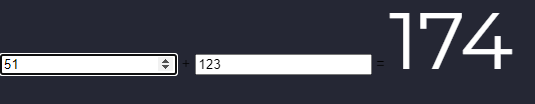
** **

** **

****

**Что такое элемент output в HTML 5?**



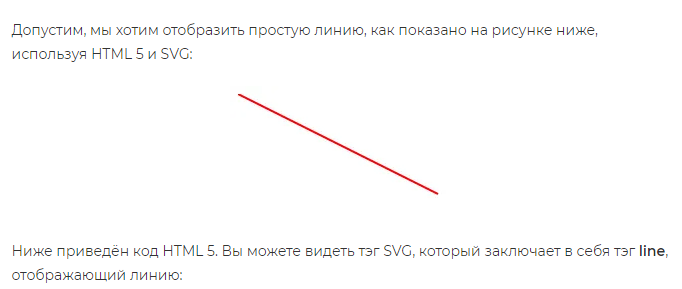


**Прикольно написано, что этот тег нужен для отображения суммы. Но отсюда я вижу много других интересных способов это реализовать.**

**Что такое SVG?**

SVG означает scalable vector graphics (масштабируемая векторная графика). Это текстовой язык графики, с помощью которого можно рисовать изображения в виде текста, линий, точек и т.п., что позволяет создавать лёгкие и быстро обрабатываемые браузером изображения.

**Можно ли сделать простое SVG изображение, используя HTML 5?**

****

****

Этот код у меня не заработал.

**Что такое канва в HTML 5?**

Канва – это область HTML документа, в которой вы можете рисовать.

Итак, как же нарисовать простую линию при помощи канвы?

Определите область канвы;



Получите доступ к контексту канвы;



Нарисуйте изображение.



Вот это работает. Может с этим можно даже какую-то змейку нарисовать.

**В чём разница между канвой и SVG?**

****

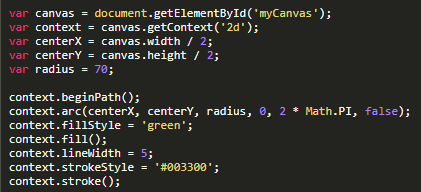
**Прямоугольник**

****

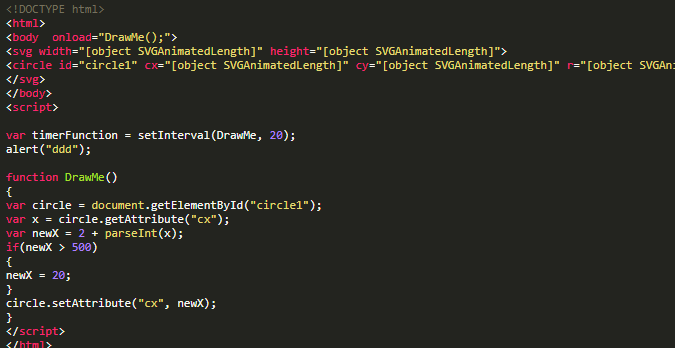
** **

**Круг**

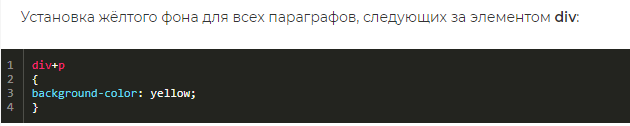
****

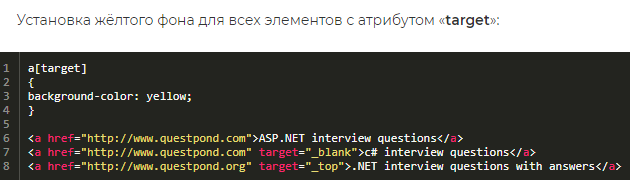
****

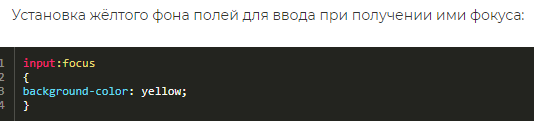
**Вот пример кода манипуляции над svg.**

****

**Важные селекторы.**

****

****

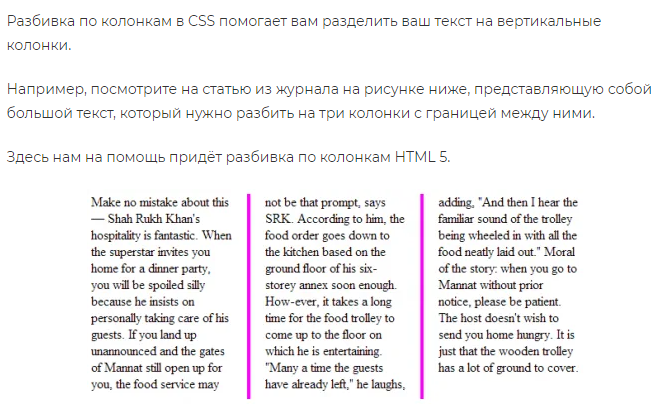
****

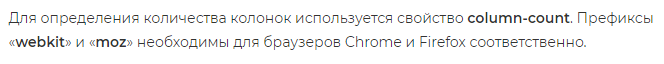
**Active:** пока мышь зажата

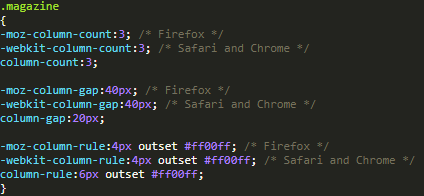
**Hover:** наведение

**Focus:** клик на инпут например, и пока не кликнуть куда-то еще будет работать.

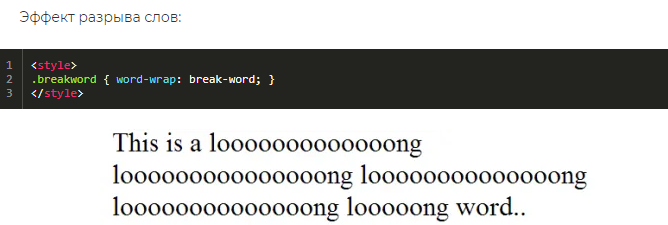
**Как использовать разбивку контента по колонкам в CSS?**

****

****

****

****

****

**Что такое web workers и зачем они нужны?**

Допустим есть тяжелый цикл, условно. Который отрабатывает миллион раз.

function SomeHeavyFunction() {

for (i = 0; i < 1000000000; i++) {

console.log(i)

}

}

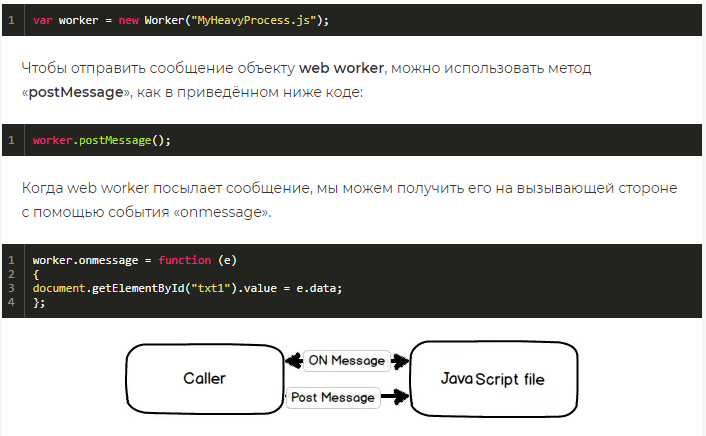
Так, как он выполняется синхронно, то сайт будет ожидать его выполнения. И может вообще зависнуть.

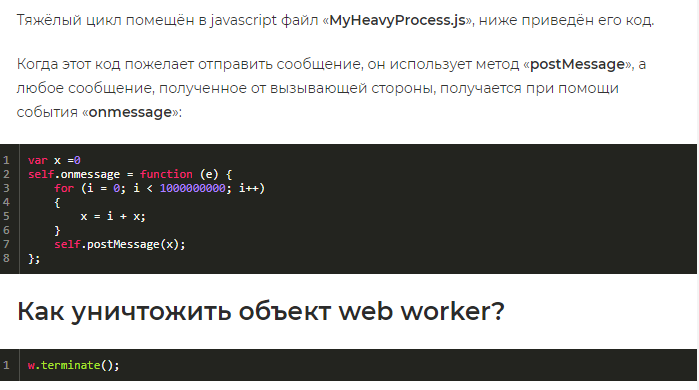
Поэтому если бы мы могли поместить этот тяжёлый цикл в файл JavaScript и запустить его асинхронно, тогда браузеру не пришлось бы ждать его завершения, и мы бы получили более отзывчивое приложение. Для этого и предназначена технология web workers, позволяющая исполнять файлы JavaScript асинхронно.

**Какие ограничения накладываются на поток Web Worker?**

Потоки web worker не могут изменять HTML элементы, глобальные переменные и некоторые свойства окон, такие как window.location. Вы можете использовать типы данных javascript, вызовы XMLHttpRequest и прочее.

Чтобы создать поток web worker, мы должны передать имя файла JavaScript в конструктор Worker.





**В локальных файлах у меня не получилось. Но это точно нужно попробовать. Так что тут еще не все.**

**Зачем в HTML 5 введены события server-sent?**

Одной из наиболее частых потребностей в сетевом мире является получение обновлений с сервера. Например, в приложении, отображающем биржевые сводки, браузер должен регулярно получать последние котировки с сервера.

Поэтому, чтобы реализовать этот тип запросов, разработчики обычно пишут некоторый pull-код, который отправляет запросы на сервер и получает оттуда данные через определённые интервалы времени. Это хорошее решение, но оно забивает сеть множеством запросов, а также добавляет нагрузку на сервер.

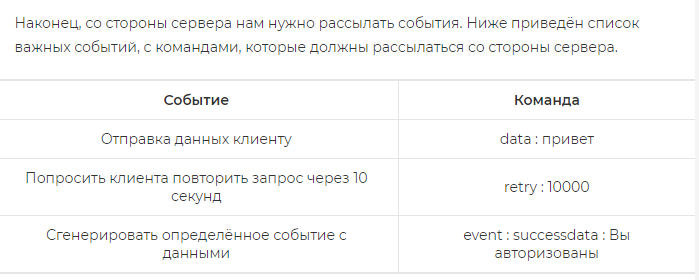
Поэтому вместо pull-запросов было бы здорово иметь какое-нибудь push-решение. Проще говоря, когда сервер получает обновление, он рассылает его браузерам клиентов. Это может быть достигнуто с помощью **«SERVER-SENT СОБЫТИЙ».**

Сначала браузер должен связаться с источником на сервере, который будет посылать обновления. Допустим, у нас есть страница «stock.aspx», которая рассылает обновления котировок.

Чтобы связаться со страницей, мы должны использовать привязку к объекту источника события, как показано в коде ниже:

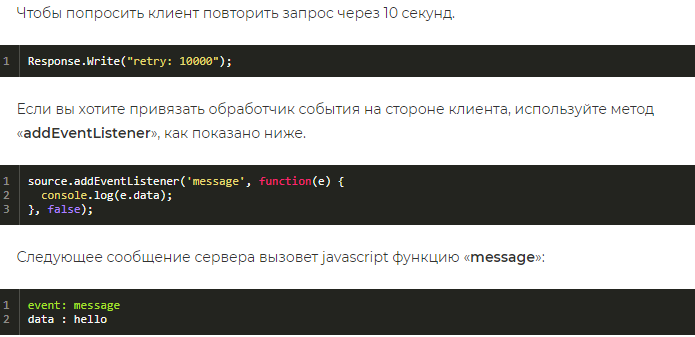






Например, если мы хотим отправить данные, ниже приведён код ASP.NET, чтобы сделать это. Пожалуйста, обратите внимание, что заголовок «ContentType» установлен в «text/event-stream





**Расскажите о концепции локального хранилища в HTML 5**

Часто нам нужно хранить информацию о пользователе на его локальном компьютере. Допустим, пользователь наполовину заполнил большую форму, и неожиданно пропадает интернет соединение.

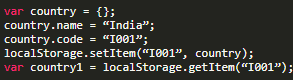
Поэтому хорошо было бы хранить эту информацию локально у пользователя, а когда интернет соединение восстановится, пользователь использовал бы её и отправил на сервер.

Современные браузеры имеют хранилище, называемое «Локальным хранилищем» («Local storage»), где вы можете хранить эту информацию.

Данные добавляются и удаляются из локального хранилища с помощью пары «ключ-значение». Ниже приведён пример кода добавления страны «India» с ключом «Key001»:







**Если вы хотите хранить данные в формате JSON, вы можете использовать функцию «JSON.stringify»:**

****

Локальное хранилище не имеет срока жизни. Данные будут храниться до тех пор, пока пользователь не удалит их из своего браузера, либо вы не удалите их, используя код JavaScript.

**В чём разница между локальным хранилищем и cookies?**

****

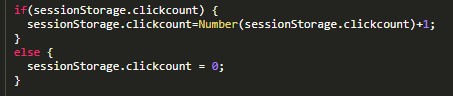
**Что такое хранилище сессии и как его создать?**

Хранилище сессии похоже на локальное хранилище, но данные действительны на протяжении сессии. Проще говоря, данные удаляются, как только вы закрываете браузер.

Чтобы создать хранилище сессии, вам нужно использовать конструкцию «sessionStorage.имяпеременной». В следующем коде мы создаём переменную «clickcount».

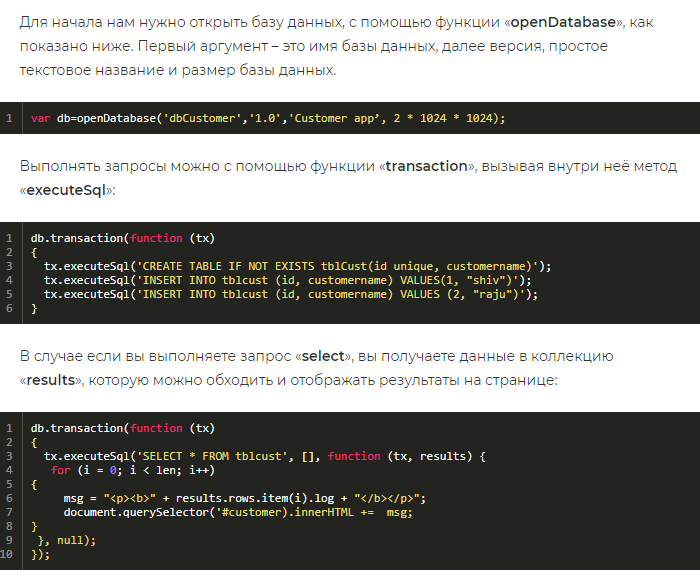
Если вы обновите страницу в браузере, счётчик увеличится. Но если вы закроете браузер и откроете его снова, переменная «clickcount» снова установится в ноль:

Черт, какая удобная штука для отладки. Обновляешь страницу, а она работает.



**Что такое WebSQL?**

WebSQL – это реляционная база данных, работающая на стороне браузера клиента. Внутри браузера есть реляционная СУБД, в которой вы можете выполнять SQL запросы. WebSQL не является частью спецификации HTML 5. Спецификация основана на SQLite.

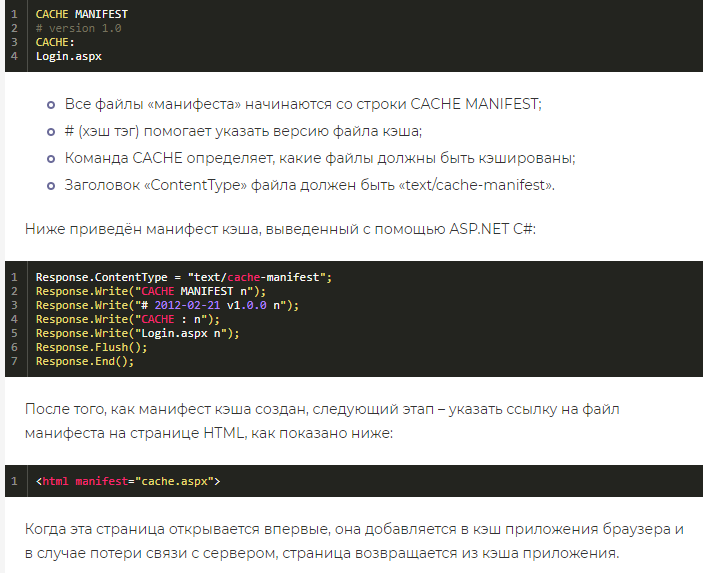


**Что такое кэш приложения в HTML5?**

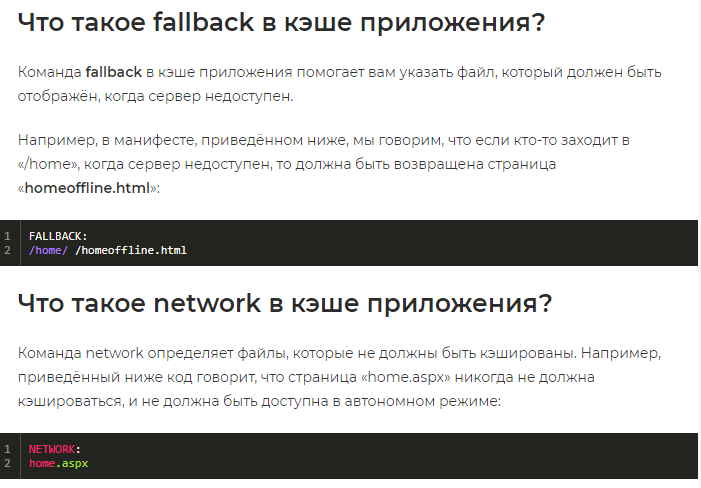
Одной из наиболее востребованных функций для конечного пользователя является работа в автономном режиме. Другими словами, если соединение с интернетом недоступно, страница должна возвращаться из кэша браузера, и кэш приложения поможет вам в этом.

Кэш приложения помогает вам определить, какие файлы должны быть кэшированы, а какие нет.

Нам необходимо создать файл «манифеста». Файл «манифеста» помогает вам определить, как должно работать кэширование. Ниже приведена структура файла «манифеста»:



Кэш приложения обновляется при смене номера версии, указанного после тэга «#», как показано в следующем коде:

****