



Столыпинский
вестник

Научная статья
Original article
УДК 004

ПРОЦЕССЫ ПРОИСХОДЯЩИЕ В БРАУЗЕРЕ ПРИ ЗАГРУЗКЕ ВЕБ САЙТА

**THE PROCESSES THAT TAKE PLACE IN THE BROWSER WHEN LOADING
A WEB SITE**

**Ярвая Екатерина Владимировна, Магистранта Гродненского
государственного университета им Я.Купалы, Беларусь, г.Гродно**

**Yaravaya Katsiaryna Vladimiravna, Master's student Yanka Kupala State
University of Grodno, Belarus, Grodno**

Аннотация

В данной статье будет рассмотрена работа веб-браузера и как происходит отображение страницы на клиентской части приложения. Затронем как строится DOM дерево, CSS дерево, дальнейшее их преобразование и размещение на странице, взаимодействие JavaScript во время разбора HTML разметки (парсинг), какие проблемы могут возникнуть во время этого процесса и способы их решения. Рассмотрим процесс обновления расположения элемента на странице, природа возникновения некорректного и заторможенного отображения.

Annotation

This tutorial will explain how a web browser works and how a page is rendered in a Cuneiform part of an application. This article takes a look at how a CSS DOM tree is built, how CSS is further transformed and positioned on the page, how JavaScript interacts during HTML parsing, what problems might arise during this process and how to deal with them. Let's consider the process of updating the layout of the elements on the page, the nature of the occurrence of incorrect and hindered display.

Ключевые слова: HTML, DOM дерево, CSS дерево, веб-браузер

Keywords: HTML, DOM tree, CSS tree, web browser

Давайте погрузимся в достаточно увлекательный и интересный процесс, узнать, что происходит после того, как вы ввели в адресную строку браузера адрес вашего любимого сайта. И разберем что именно скрыта от наших глаз. Какие внутренние механизмы заставляют загрузить веб страницу и отобразить ее пользователю.

Первый в мире браузер был создан не так давно, а всего лишь в 1990 году Тимом Бернерсом-Ли. Всем известны Internet Explorer в 1995 году занимал всего лишь 5% от рынка, а абсолютный фаворитом был браузер под названием Netscape Navigator, но уже к 2001 году ситуация изменилась полностью наоборот и Internet Explorer занимал 96% рынка. 1995 год ознаменовался выпуском браузер Opera который распространялся по модели условно бесплатного программного обеспечения из-за чего имел низкую популярность в мире, но очень понравился пользователям из СНГ. В 2008 году был выпущен на базе Chromium появился Chrome который в то время имел некоторые новшество, например режим “инкогнита”. В 2012 году в Яндекс создали Яндекс.Браузер. В 2015 году Microsoft представила новый браузер Microsoft Edge. А с 15 июня 2022 компания Microsoft прекращает поддержку браузера Internet Explorer из-за невозможности внедрения новых возможностей.

Большинство современных браузеров умеет определять, что мы вводим в адресную строку, либо мы хотим что-то найти через поисковую систему, либо нам нужно перейти на определенный сайт. И первое что должен сделать браузер, это определить, что это поисковый запрос или URL после того, как браузер определил, что это, URL инициирует соответствующие протоколы, DNS (Domain Name System “система доменных имён”) и TLS соединения в данной статье мы не будем рассматривать. После того, как браузер получил ответ, он определяет с помощью поля Content-type, какой ответ мы получили, если это HTML, то управление передает рендер процессу, или, к примеру, это zip файл, то начинает выполняться процесс загрузки и в действие вступает менеджер загрузок. Мы также не будем сильно углубляться и опустим пару шагов и перейдем к моменту, когда браузер на вашем персональном компьютере получил HTML страницу.

И вот наш браузер держит у себя HTML, и он должен его вывести пользователю, тут мы остановимся и разберемся немного подробнее. Из полученного HTML, браузер должен построить так называемый DOM (Document Object Model), этот процесс весьма затратный по ряду причин, одна из них то, что в сети есть много кода, мягко говоря сомнительного качества, и все ошибки, которые были допущены во время написания HTML разметки, браузер должен определить и устранить, но даже если у вас идеально написан HTML, все равно ресурсы на его проверку будут потрачены. Мы говорим про HTML, а что будет, если сайт не просто состоит из HTML, но где, то есть и код JavaScript, в данном случае парсинг остановится, загрузчик разберется и выполнится, парсинг HTML продолжится, может возникнуть резонный вопрос, а зачем? С помощью JavaScript можно изменить DOM дерево. Для того, чтобы процесс построения DOM дерева (парсинга) не останавливался, скрипт добавляют в конец документа, но есть и более современные и производительные решения, можно добавить атрибуты defer или async в тег <script> (тег <script> оборачивает код написанный на JavaScript).

Для того, чтобы наш сайт был красивым, и дизайнеры могли кушать свой хлеб, существует CSS (Cascading Style Sheets), он также преобразуется в CSS дерево, и он также блокирует парсинг HTML. С помощью этих двух деревьев будет сформировано третье, под названием рендер дерево (render tree). Элементы, которые не имеют своего видимого представления будут проигнорированы. Следующий этап, который должен выполнить браузер, высчитать где, какой элемент должен позиционироваться на экране пользователя, этот процесс называется Layout. Layout — это достаточно непростая задача, и обязательная, это как рисовать по описанию, есть красный треугольник и черный квадрат. Layout — это этап, на котором рендер дерево занимает важные позиции на веб сайте. И завершающий этап — это отрисовка, продолжая аналогию с картиной, если у нас есть две фигуры, нам же надо знать какая из них будет стоять впереди, будут ли одна накрывать другую, вот это задача и решается в момент отрисовки.

Еще хотелось бы рассказать о природе возникновения, дрожания “тормозов” или “лагов”. Мы все осознаем, если происходит сдвиг или изменение на странице веб сайтов, то браузер должен заново позиционировать все элементы. Допустим у нас на странице есть какой то анимированный интерактивный элемент, вторая вводная - экран с частотой обновления 60 раз в секунду, и если браузер отвечает каждый так называемый фрейм (frame), то анимация выглядит плавно, но если один или несколько фреймов выпадает, то страница выглядит дрожащей. Еще одна причина может быть в том, что все операции выполнялись в одном потоке, и запуск JavaScript будет выполняться в том же потоке, что может вызвать задержку и выпадение фреймов. Если есть большой и трудоемкий скрипт, его можно разбить на так называемые чанки (это контейнеры для HTML/CSS/JS-кода), и планировать выполнение каждой из них между обновлением фреймов.

Когда все готово, все вычислено и может быть отрисовано в виде пикселей на экране у пользователя, запускается процесс растеризации (rasterizing). Но

зачем растеризировать сразу все, если пользователь может видеть только часть, по такому принципу и работали первые версии браузера chrome за что он и заслужил звание самого быстрого браузера во время первых своих шагов. Он просто показывал пользователю сначала то, что он видит, а дальше отрисовывал остальные элементы страницы, включая все свои внутренние процессы. Сейчас же современные браузеры запускают куда более сложные процессы.

Литература

1. В. Эстель, HTML5. Разработка приложений для мобильных устройств (2015)
2. Дэвид Сойер Мак-Фарленд и Макфарланд Дэвид, Новая большая книга CSS (2018)
3. С. Пьюривал, Основы разработки веб-приложений (2015)
4. Prateek Mehta, Creating Google Chrome Extensions (2016)
5. Шкляр Л., Архитектура веб-приложений (2010)

Literature

1. V. Estel, HTML5. Developing Applications for Mobile Devices (2015)
2. David Sawyer McFarland and McFarland David, The New Big Book of CSS (2018)
3. S. Purewal, Fundamentals of Web Application Development (2015)
4. Prateek Mehta, Creating Google Chrome Extensions (2016)
5. Shklar, L., Web Application Architecture (2010)

© Яровая Е.В., 2022 Научный сетевой журнал «Столыпинский вестник» №5/2022

Для цитирования: Яровая Е.В. ПРОЦЕССЫ ПРОИСХОДЯЩИЕ В БРАУЗЕРЕ ПРИ ЗАГРУЗКЕ ВЕБ САЙТА// Научный сетевой журнал «Столыпинский вестник» №5/2022