



Курс

«Программирование в компьютерных сетях»

Лекция 1

Приходько Татьяна Александровна
доцент кафедры
Вычислительных технологий КубГУ



Цели курса

Целью курса «Программирование в компьютерных сетях» является приобретение теоретических и практических знаний, умений и навыков, ориентированных на **эффективное профессиональное использование** современных web-технологий.



Цели курса

Научиться

- осуществлять проектирование web-приложения;
- осуществлять поэтапную разработку страниц приложения по отдельности, а также интегрировать их воедино в одно цельное приложение;
- Программировать на JavaScript;
- Встраивать в свои страницы работу с базами данных;
- Разрабатывать клиент-серверные приложения.



Цели курса

Иметь понимание:

- проблем и направлений развития Web-технологий;
- основные методы и средства проектирования программного обеспечения Web-сайтов;
- о возможностях использования дополнительных пакетов и библиотек при программировании;
- о современных объектно-ориентированных алгоритмических языках, их области применения и особенностях;
- о WEB-сервисах и их архитектуре.



Задачи курса

Освоить

- язык гипертекстовой разметки HTML;
- CSS – каскадные таблицы стилей;
- PHP;
- Java-script, JQuery;
- Форматы для передачи данных в клиент-серверных приложениях.

Выполнить

- Разработку сайта.



План лекции

Основные понятия

Основные виды браузеров и особенности их использования

Обзор инструментария

Технологии Web-программирования

Основные вехи WEB

Перечень лабораторных работ

«WEB-программирование»



Основные понятия

Основные виды браузеров и особенности их использования

Обзор инструментария

Технологии Web-программирования

Основные вехи WEB

Перечень лабораторных работ

«WEB-программирование»



➔ **Для чего нам нужен интернет и его инструменты?**

- *Для поиска и анализа информации*
- *Для презентации себя и своих достижений*
- *Для зарабатывания денег*
- *Для общения*



→ Какими инструментами мы пользуемся?

- **Для поиска и анализа информации**
 - ▢ Браузеры и поисковики
 - ▢ Соцсети, сервисы, средства аналитики
- **Для презентации себя и своих достижений**
 - ▢ WEB-страницы и сайты
- **Для зарабатывания денег**
 - ▢ Сайты, сервисы, средства аналитики
- **Для общения**
 - ▢ Соцсети, почта, сервисы

Основные понятия

- **WEB-Документ** – это документ, помещенный во Всемирной паутине для ознакомления с ним большого числа пользователей и содержащий текст, графику, анимацию.
 - **Web-сайт**
 - **Web-сервис**
 - **Web-приложение**
- Эти три понятия (Web-сайт, Web-сервис, Web-приложение) объединены скобкой и связаны линией с вопросом: Это одно и то же?

Основные понятия

- **Web-сайт** – набор документов информационного содержания, как правило, статический
- **Web-сервис** - сайт, созданный для выполнения каких-либо задач или предоставления услуг в рамках сети. У онлайн-сервиса есть уникальный функционал. Это может быть конструктор товара, фотохостинг, закрытая социальная сеть для корпоративного пользования, открытая социальная сеть, доска объявлений, почтовый сервис, файлообменный, облачное хранилище...
- **Web-приложение** – сервис может являться частью приложения.

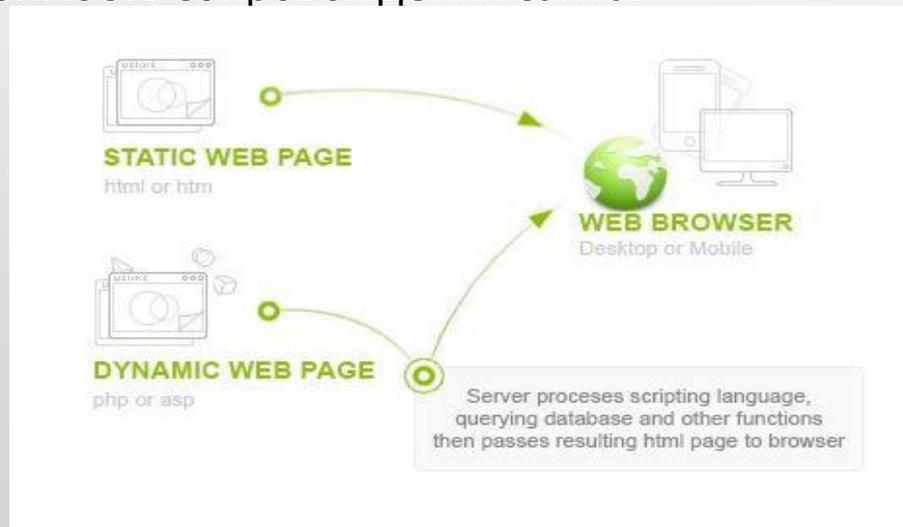
Основные понятия – сайт

- **Сайт** – в компьютерной сети объединенная под одним адресом (доменным именем или IP-адресом) совокупность документов частного лица или организации.

Изначально веб-сайты представляли собой совокупности *статических* документов. В настоящее время большинству из них свойственна *динамичность и интерактивность*. В большинстве случаев в Интернете одному веб-сайту соответствует одно *доменное имя*. Именно по доменным именам сайты идентифицируются в глобальной сети.

Основные понятия

- Статические сайты представляют из себя набор статических страниц, размещенных на сервере. Основным недостатком статических сайтов является сложность их модификации в виду того, что для изменения содержимого администратору необходимо вручную вносить изменения в код страниц.
- Динамические сайты, как правило, построены на основе какой-либо системы управления содержимым (CMS), что позволяет вносить изменения и публиковать страницы, новости, изображения и другие документы непосредственно из панели управления сайтом, что существенно уменьшает стоимость сопровождения сайта.



Основные понятия

- Подавляющее большинство веб-сайтов в Интернете являются гибридами. Например, интернет-магазин - это гибрид, который с одной стороны содержит каталог продукции и предоставляет информацию о каждом отдельно взятом продукте, с другой стороны позволяет выполнять покупки и оплачивать их с помощью банковских карт или электронных систем расчетов, например, WebMoney и Яндекс-Деньги.

Основные понятия – Доменное имя

- **Доменное имя** — символьное имя домена. Должно быть уникальным в рамках одного домена. Полное имя домена состоит из имён всех доменов, в которые он входит, разделённых точками. Например, полное имя `ru.wikipedia.org` обозначает домен третьего уровня `ru`, который входит в домен второго уровня `wikipedia`, который входит в домен `org`, входящий, в свою очередь, в корневой домен.

Доменное имя служит для адресации узлов сети Интернет и расположенных на них сетевых ресурсов (веб-сайтов, серверов электронной почты, других служб) в удобной для человека форме.

Основные понятия - DNS

- Для преобразования IP-адресов серверов в доменные имена (и наоборот) служит DNS, представляющая собой распределённую систему серверов. DNS хранит ресурсные записи, каждая из которых имеет имя, тип записи и поле данных, содержание которого зависит от типа записи. Как правило, поле данных содержит некоторый IP-адрес или имя другого хоста.
- Каждый домен второго уровня имеет минимум две NS-записи, хранящие доменные имена или IP-адреса дублирующих друг друга DNS-серверов, которые обслуживают данный домен. Эти серверы, как правило, расположены у какого-либо хостинг-провайдера или интернет-провайдера и содержат все прочие типы записей для домена, а также все DNS-записи для поддоменов данного домена.

Хостинг

Услуга сдачи в аренду веб- и DNS-серверов для размещения веб-сайтов называется *хостингом* и оказывается хостинг-провайдерами, имеющими необходимые лицензии, специально оборудованные помещения (дата-центры), каналы связи, техническую поддержку. Мелкие хостинг-провайдеры, так называемые реселлеры, не имеют своего дата-центра и арендуют необходимые ресурсы.

Распространены следующие виды хостинга веб-сайтов:

- Colocation;
- Dedicated;
- Virtual dedicated server (VDS);
- Virtual private server (VPS);
- Shared-хостинг;
- Cloud computing.

Хостинг

Виды хостинга веб-сайтов:

- **Colocation:** сервер и его программное обеспечение принадлежит заказчику и размещается в дата-центре провайдера в стойке с другими серверами;
- **Dedicated:** владельцем сервера является хостинг-провайдер, который может оказывать техническую поддержку операционной системы и ПО;
- **Virtual dedicated server (VDS):** на мощном сервере запущено несколько виртуальных серверов с помощью технологии виртуализации, один из которых сдается в аренду конкретному клиенту. Каждая виртуальная машина имеет свой IP-адрес, может содержать любую ОС и ПО, имеет гарантированные минимальные ресурсы (память, процессорное время или количество ядер CPU);

Хостинг

- **Virtual private server (VPS):** от предыдущего варианта отличается использованием технологий виртуализации с общим ядром и одинаковой версией всех гостевых ОС на сервере.
 - Минусы – ограничения, связанные с меньшим уровнем изолированности и разделения ресурсов виртуальных машин.
 - Плюсы – накладные расходы ресурсов серверов на обслуживание виртуализации меньше, а следовательно, и ниже цена хостинга.

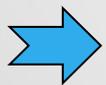
Хостинг

- **Shared-хостинг:** используется многопользовательская ОС, клиенту предоставляется аккаунт одного пользователя с ограниченными правами, у каждого пользователя своя область дисковой памяти. Общие для всех пользователей ресурсы сервера и, как правило, IP-адрес. Несколько сотен сайтов могут быть размещены на одном физическом сервере с одним IP-адресом (виртуальный хостинг по имени). **Самый массовый вид хостинга, подходящий для большинства простых веб-сайтов;**
- **Cloud computing:** запрос приходит на ферму серверов, клиент платит за количество запросов, за место для хранения данных и за объем передаваемых данных.

«WEB-программирование»



Основные понятия



Основные виды браузеров и особенности их использования

Обзор инструментария

Технологии Web-программирования

Основные вехи WEB

Перечень лабораторных работ

Обзор WEB-браузеров

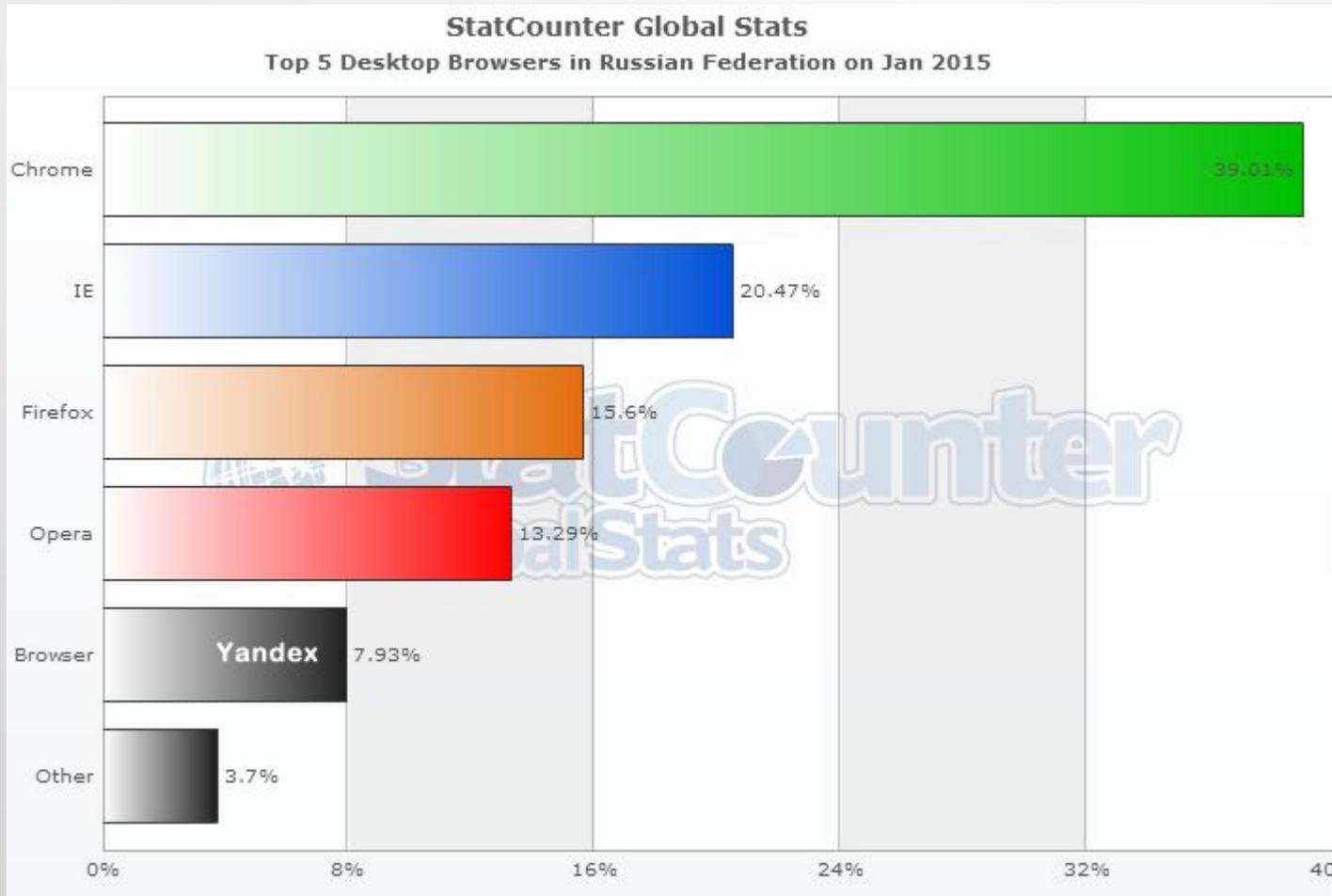
Основные виды браузеров и их особенности

Браузер (от англ. browse — просматривать, листать) — это специальная программа, позволяющая просматривать содержимое сети Интернет.

Сейчас популярны четыре ядра, или семейства:

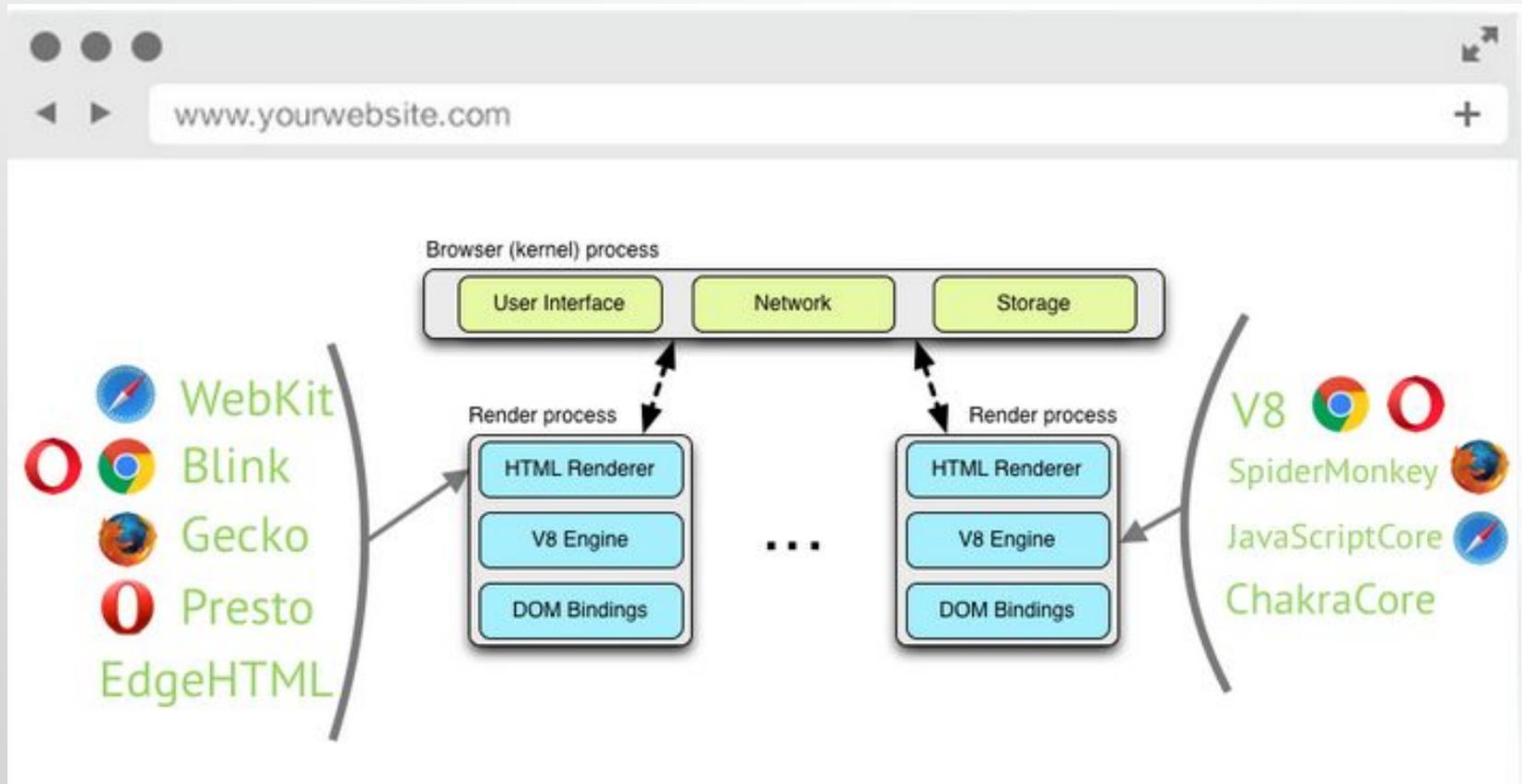
- На основе Gecko: FireFox, Mozilla, Netscape, SeaMonkey.
- На основе Trident: Internet Explorer, Maxthon (MyIE).
- На основе KHTML или WebKit: Konqueror, Safari, браузер телефонов Nokia 60 Series.
- На основе Presto и Merlin: Opera (с версии 15 переход на **WebKit**).

Обзор WEB-браузеров



Популярность браузеров на 2015

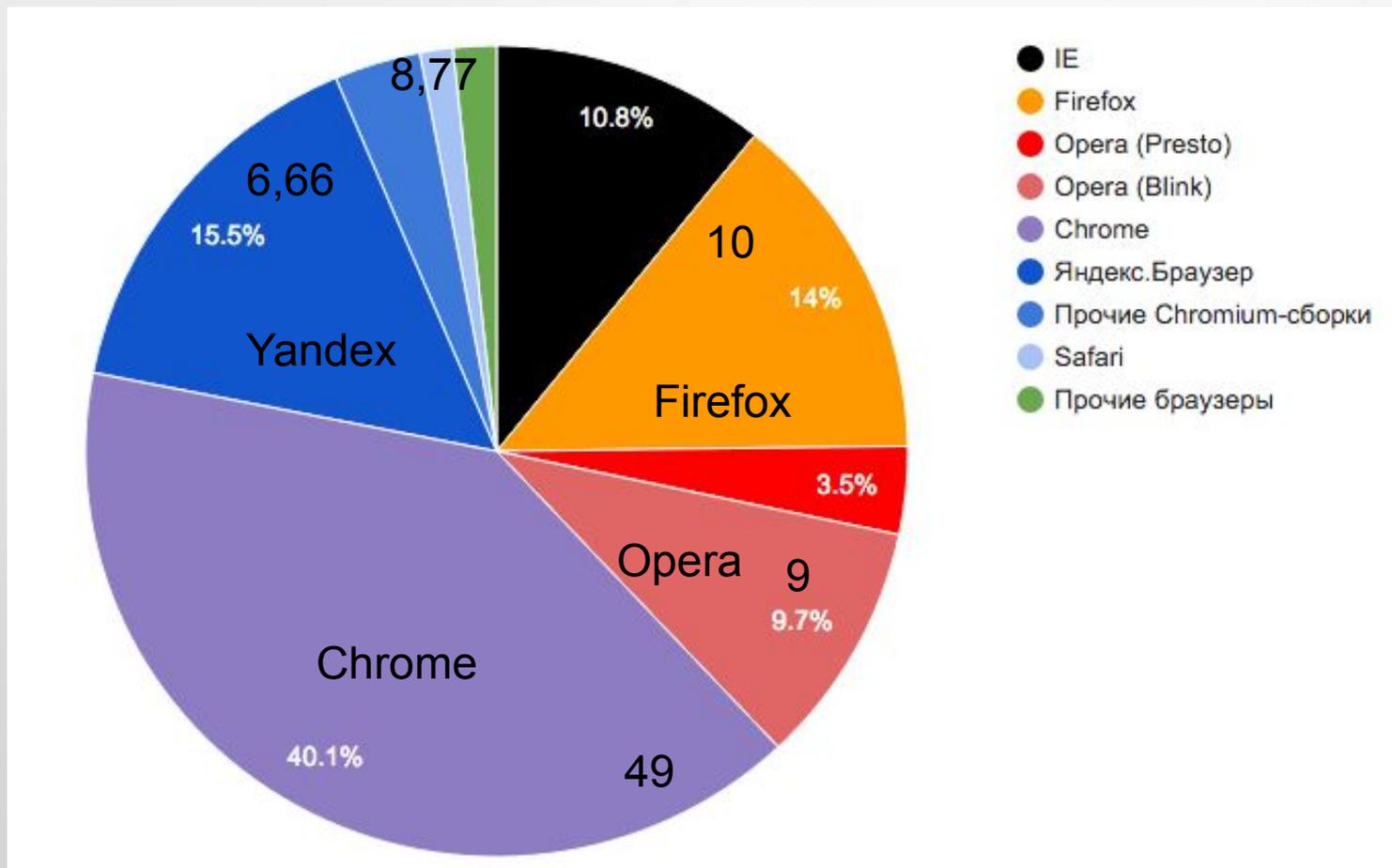
Что находится внутри браузера?



Ограничения JS в браузере

- Нельзя взаимодействовать с файловой системой
- Нет доступа к сетевым функциям, кроме того, что предоставляет сам браузер
- Нет возможности организовывать многопоточные вычисления. Есть воркеры, но они имеют определённые ограничения
- Нельзя создавать новые процессы / запускать программы (открытие новых вкладок не считается)

Обзор WEB-браузеров



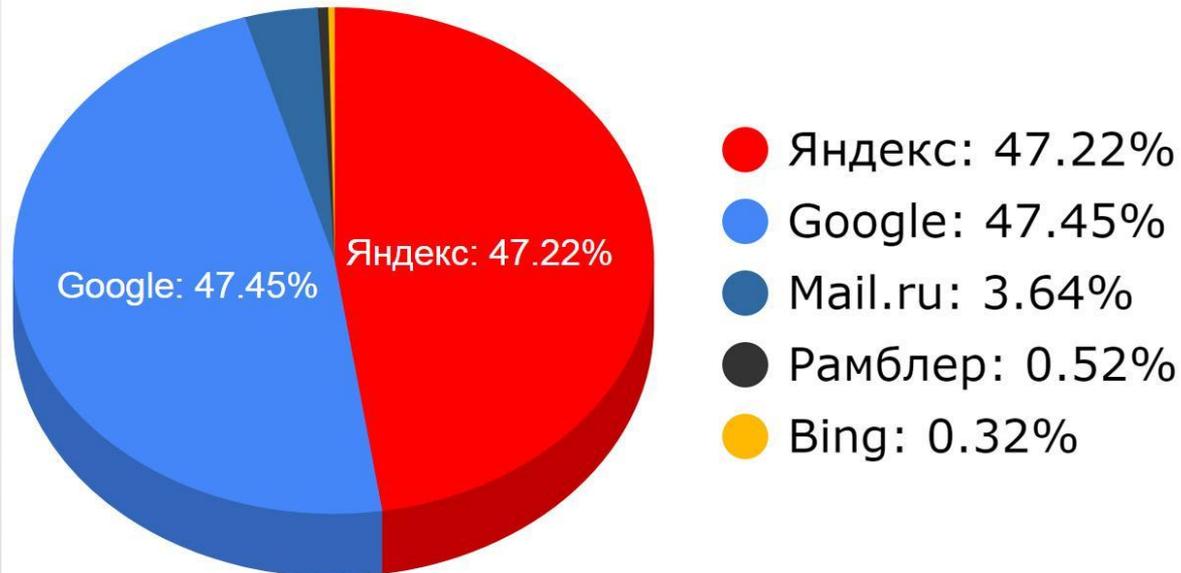
**Популярность браузеров по Liveinternet в рунете 2015
(белым)
на начало 2017г (черным)**



В чем отличие WEB-браузера от поисковика?

- **Браузер** (от английского «Web browser» – Веб-обозреватель, просмотрщик) является программой для просмотра страниц в интернете, а поисковая система – это программный комплекс, который с помощью веб-интерфейса позволяет искать необходимую информацию в интернете.
- **Поисковая система (ПС)** доступна только через браузер. Браузер – это программа, установленная на вашем компьютере. А поисковая система представляет какой-либо сайт, расположенный в интернете.
- В основе ПС лежат сложные алгоритмы ранжирования и базы данных с индексными подборками к различным запросам.
- Над развитием данных веб-сервисов и алгоритмов работают огромные компании с сотнями сотрудников. Они стараются **сделать поиск информации быстрым и удобным** для людей, а также хорошенько заработать на контекстно-медийной рекламе и других сервисах.

В чем отличие WEB-браузера от поисковика?



<http://infox45.ru/prodvizhenie-sajtov/o-poiskovyx-sistemax-runeta>

Сегодня все браузеры имеют свои встроенные системы поиска, что облегчает работу в сети Интернет. Многие крупные владельцы поисковых систем обзаводятся своими интернет-браузерами, например, Яндекс (Яндекс браузер), Google (Google Chrome) и другие. Конечно, встроенный поиск в этих браузерах – это поисковая система этой же компании.

На заметку

Различные браузеры показывают конкретную страницу по-разному. Отображение страниц зависит от работы алгоритмов коррекции, встроенных в браузер, а в разных браузерах эти алгоритмы различны.

Чтобы создать сайт, который будет корректно отображаться в большинстве браузеров, необходимо:

- проверять страницы сайта во всех распространенных браузерах - Chrome, Firefox, Mozilla, Opera, IE, — причем в разных версиях;
- предоставлять разные варианты доступа к информации (самое лучшее — создать дополнительно текстовую версию);
- по возможности не размещать важную информацию в графике и дополнять рисунки комментарием в атрибуте alt — некоторые пользователи ходят по сети Интернет с отключенной графикой.

Кроссбраузерность HTML страниц

Normalize.css является альтернативой [CSS Reset](#). Проект является продуктом сотен часов глубокого исследования различий между изначальными стилями браузера. Это исследование провели Николас Галахер, и Джонатан Нил/

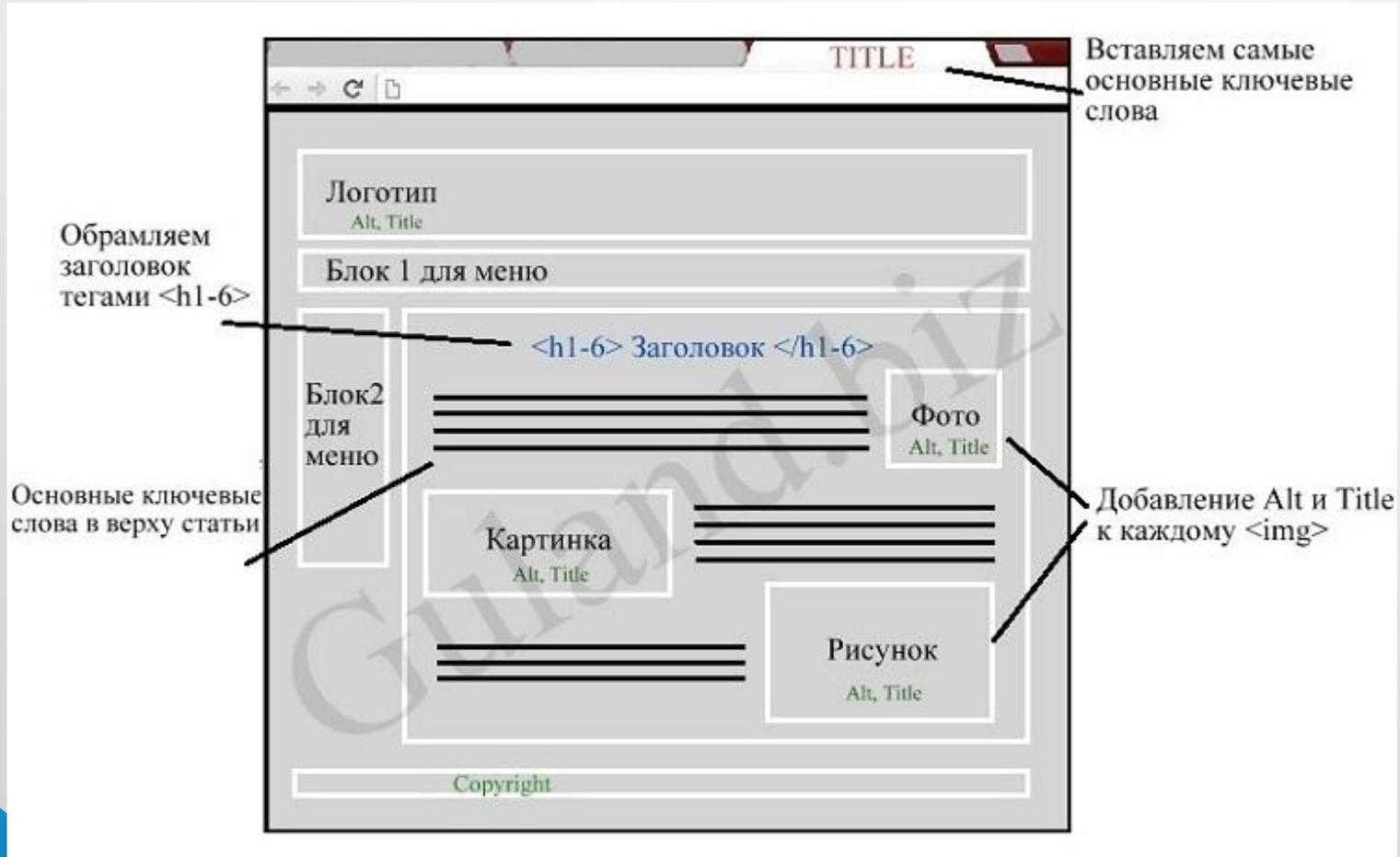
Цели normalize.css:

- сохранять полезные настройки браузера, а не стирать их;
- нормализовать стили для широкого круга HTML-элементов;
- корректировать ошибки и основные несоответствия браузера;
- совершенствовать юзабилити незаметными улучшениями;
- объяснять код, используя комментарии и детальную документацию.

Он поддерживает широкий диапазон браузеров (в том числе мобильных) и включает в себя CSS, который нормализует HTML5-элементы, типографику, списки, встраиваемый контент, формы и таблицы.

Перейдите по адресу cdnjs.com/libraries/normalize и найдите версию данного файла, заканчивающуюся на .min.css (эта версия меньше по размеру, чем остальные, за счет удаления пробельных символов)

Наиболее общая структура html страницы



«WEB-программирование»



Основные понятия

Основные виды браузеров и особенности их использования

 **Обзор инструментария**

Технологии Web-программирования

Основные вехи WEB

Перечень лабораторных работ

Инструментарий

Существует множество способов создать свою собственную web-страницу.

1. Создать персональную страницу с помощью конструкторов сайтов. Такой способ является простым и быстрым, но у него есть свои серьезные недостатки. Одним из главных является то, что подобные конструкторы предоставляют слишком мало возможностей по настройке web-страниц. Как правило, это возможность изменения фонового изображения, размера шрифта текста, форм и некоторые другие настройки. Однако подобные сайты выглядят, как "печенья", выпеченные в одной печи - все они похожи друг на друга и имеют схожий интерфейс.
2. Теми же недостатками обладают движки для создания персональных блогов.

Инструментарий

Лучшие конструкторы сайтов



Конструктор сайтов uKit Рекомендуем!

uKit — простой и эффективный конструктор, позволяющий построить сайт для вашего бизнеса самостоятельно. Основан на визуальном редакторе и имеет большое количество красивых шаблонов.

Создать сайт



Конструктор сайтов Wix

Wix — отличное решение для создания красивых flash-сайтов, творческих портфолио и сайтов-визиток. Встроенная система виджетов поможет расширить функциональность.

Создать сайт



Конструктор сайтов uCoz

uCoz — самый популярный конструктор России и СНГ. Универсальное решение для создания всех типов сайтов. Продвинутые возможности по редактированию кода. Функционал сравним с большинством популярных CMS.

Создать сайт



Конструктор сайтов Webasyst

Webasyst — это веб-платформа, объединяющая систему управления сайтом (CMS) и интернет-магазином с корпоративным интранетом, CRM, системой управления проектами и задачами.

Создать сайт



Конструктор сайтов Diafan

Diafan.Cloud — решение, которое хорошо подходит для создания любых типов сайтов: магазинов, блогов, визиток, лендингов и форумов. Хорошие шаблоны, мощный функционал, умеренная стоимость.

Создать сайт



Инструментарий

3. Создать свою собственную персональную страницу вручную, написав ее с помощью HTML- кода. Сам код можно написать как в простом текстовом редакторе (*прим. переводчика* - наподобие Блокнота в Windows или Kate в Linux), так и в специализированном редакторе HTML-кода, который поможет в создании правильной и грамотной структуры страницы. Многие программисты считают данный способ наиболее правильным и используют только его в своей работе, и утверждают, что этот способ чуть ли ни единственный правильный и HTML-код нужно создавать только вручную. Можно сказать, что этот способ, без сомнения, является наилучшим при создании персональных страниц. Но в этом случае можно легко допустить ошибки при написании кода.

Инструментарий

Разновидности HTML редакторов

Все HTML редакторы можно разделить на две основные категории:

- **WYSIWYG HTML редактор кода.** С помощью программ данного типа можно создавать веб-страницы без знания языков программирования, что следует из названия: What You See Is What You Get (что видишь, то и получаешь). Преимуществом является отсутствие углубления непосредственно в процесс строительства страницы, однако это же является и недостатком.

Редакторы этого типа зачастую формируют объёмные HTML коды, в результате чего документ получается невероятно громоздким и время его загрузки увеличивается.

Инструментарий

Разновидности HTML редакторов

- **Редактор HTML тегов.** Работая с программами данного типа вы можете наблюдать непосредственный код страницы и изменять его. HTML документ получается гораздо более компактным по сравнению с результатами работы редакторов первого типа.

Однако для комфортной работы и получения приемлемого результата вам необходимо знать языки веб-программирования на довольно высоком уровне.

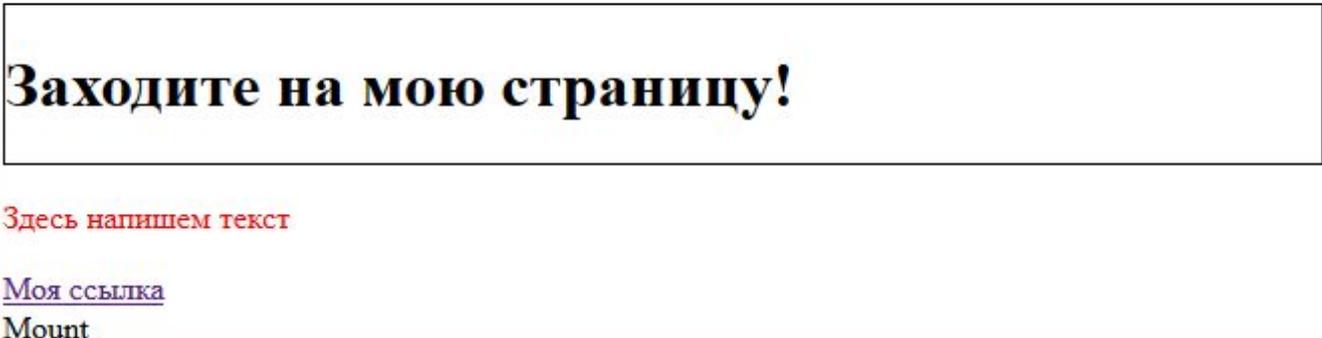
- Кроме того, HTML редакторы могут быть выполнены как в виде онлайн сервисов, так и в виде полноценных программных пакетов.

Инструментарий

Пример online HTML редактора

```
<html>
<head>
<style>
  p {color:red;}
  div {border:1px solid black;}
</style>
</head>
<body>
<div>
<h1>Заходите на мою страницу!</h1>
</div>
<p>Здесь напишем текст</p>
<a href=""> Моя ссылка</a><br>

</body>
</html>
```



<http://htmledit.squarefree.com/>

http://www.internet-technologies.ru/articles/article_1857.html

обзор online редакторов

Инструментарий

Пример online HTML редактора

- JSFiddle

- [JSFiddle](#) — это место для написания и распространения кода. Сайт предоставляет панели для написания CSS-, HTML- и JS-кода вашего проекта. JSFiddle также позволяет подключать библиотеки, такие как JQuery, AngularJS и т.п. После этого вы можете запустить код в самом приложении или сохранить его

The screenshot displays the JSFiddle online editor interface. It is divided into four main sections:

- HTML:** Contains the following code:

```
1 <div class="container">
2 <div class="field">
3 <div class="row">
4 <div class="cell" data-rows=""></div>
5 <div class="cell" data-rows=""></div>
6 <div class="cell" data-rows=""></div>
7 <div class="cell entry" data-rows="5"></div>
```
- CSS:** Contains the following code:

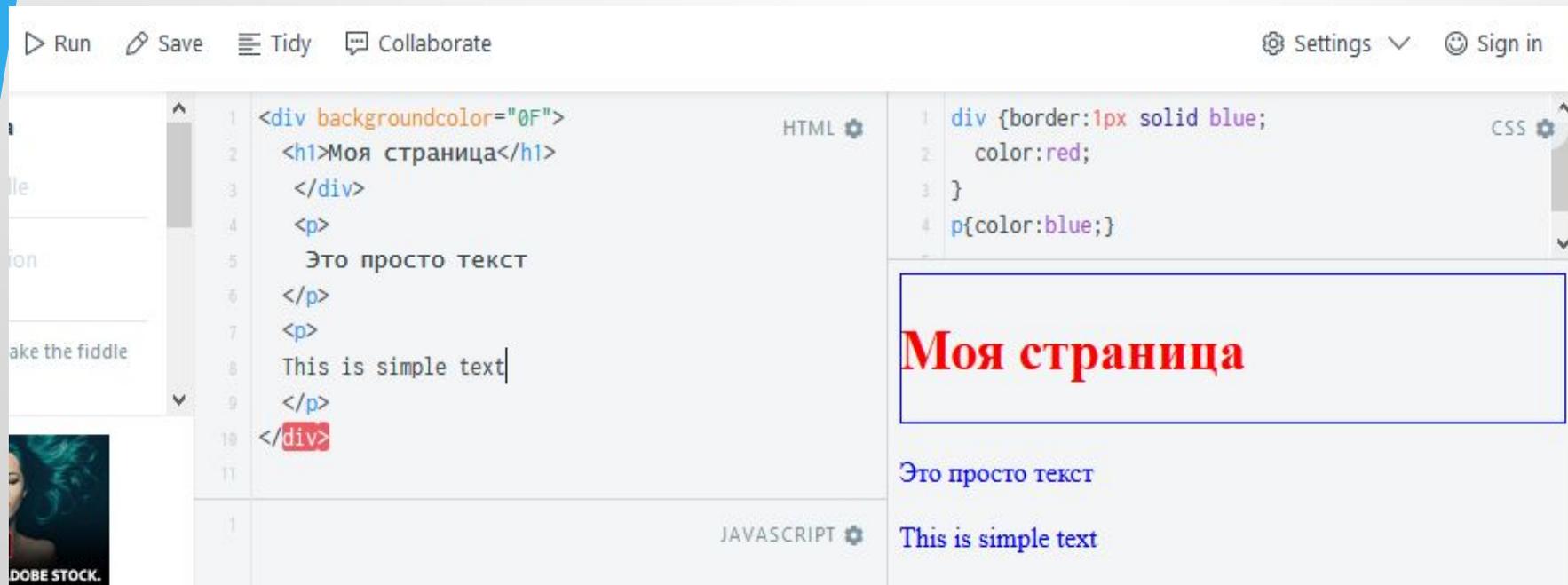
```
1 body {
2 background: #222;
3 padding: 32px;
4 font: 14px "Helvetica Neue", Arial, Helvetica, sans-serif;
5 }
6
7 .container {
```
- JavaScript:** Contains the following code:

```
1 var NORTH = 1, EAST = 2, SOUTH = 4, WEST = 8;
2 var TICK = 75;
3
4 var Field = new Class({
5 initialize: function(element) {
6 var self = this;
7 var firstRow = element.getChildren('.row')[0];
8 var wh = firstRow.getChildren('.cell')[0].getSize();
9 this.cellWidth = wh.x;
10 this.cellHeight = wh.y;
11 var width = firstRow.getChildren('.cell').length * this.cellWidth;
12 this.element = element.setStyle('width', width);
13 this.element.instance = this;
14 this.element.getParent().setStyle('width', width);
15 this.rows = this.element.getChildren('.row').length;
16 this.columns = this.element.getChildren('.row:first-child')
17 [0].getChildren('.cell').length;
18 this.cells = [];
19 this.entries = [];
20 this.exits = [];
21 this.creeps = [];
```
- Result:** Shows a preview of a maze game. The maze is a square with a grey path leading from a starting point to an exit. A green 'START' button is visible at the bottom right of the preview area.

Инструментарий

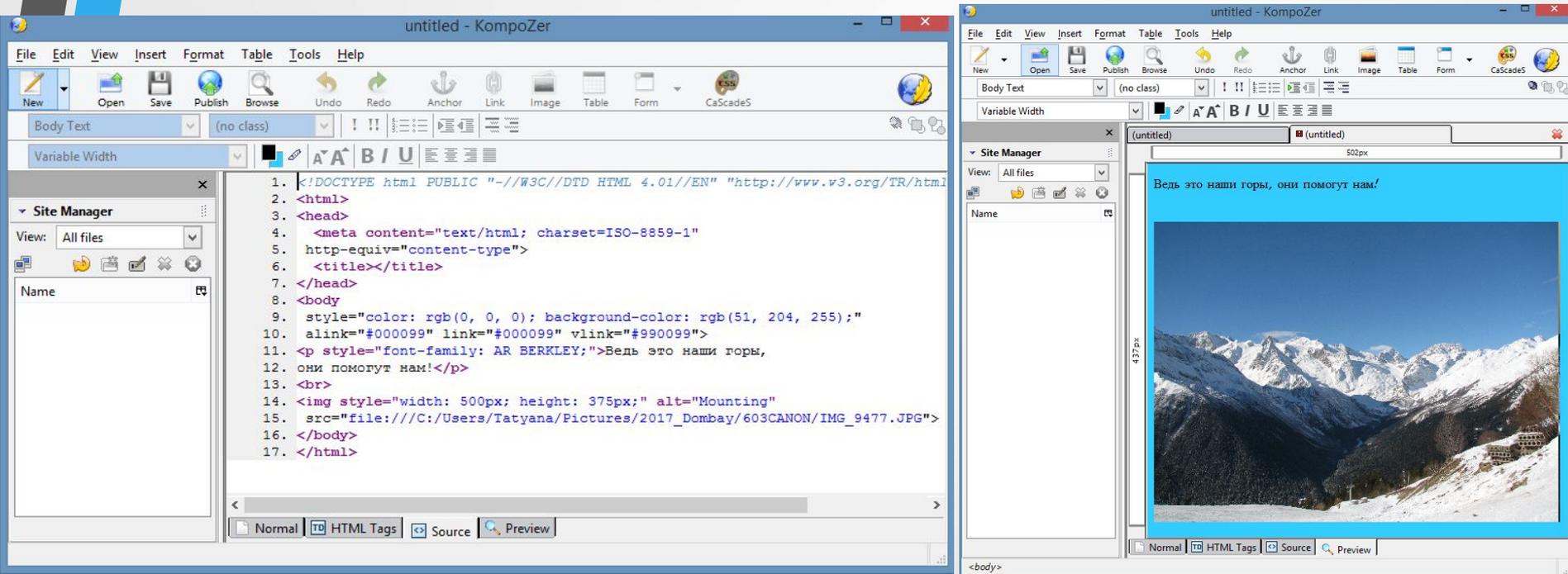
Пример online HTML редактора

- JSFiddle



Инструментарий

Пример WYSIWYG HTML редактора KompoZer



<https://sourceforge.net/projects/kompozer>

<http://rus-linux.net/MyLDP/internet/KompoZer-nachalo-raboty.html> -

Инструкция по работе с программой

<http://ruseller.com/lessons.php?rub=28&id=675>

10 бесплатных редакторов для веб страниц



Инструментарий

Пример локального HTML редактора - Brackets

Официальный сайт Официальный сайт - <http://brackets.io>

Отличный open-source продукт от Adobe, который имеет уже множество плагинов, расширений, дополнений. На взгляд разработчиков — перспективная вещь.

Написан на HTML, CSS, JavaScript.

Интерфейс

В стиле Adobe. Все очень просто и понятно, но при этом стильно.

Функционал

Стандартный, но к счастью полная поддержка LESS, SCSS. И **внимание:** тут есть некое подобие парсинга PSD. Т.е. редактор анализирует ваш PSD макет и вы свободно можете исходя из PSD сделать разметку стилей вашей страницы. Adobe есть Adobe. 43

Инструментарий



Пример локального HTML редактора - Brackets

```
103
104 <div id="hero-wrapper">
105   <div id="hero" class="row">
106     <div class="large-12 columns">
107       <h1>Brackets is an open source code editor for web designers and front-end developers. <a class="not
108         <div id="download">
109           <a id="hero-cta-button" href="https://github.com/adobe/brackets/releases/latest" class="large ro

X brackets.io.css : 378 New Rule ▾

378 #hero-cta-button {
379   font: 400 18px source-sans-pro, sans-serif;
380   text-align: center;
381   padding-left: 1.5em;
382   padding-right: 1.5em;
383 }

110     <span id="download-version" class="nowrap" data-i18n="index.page.hero.latest-release">Latest
111     </a>
112     <div id="os-alert" class="alert-box radius" data-alert>
113
114     </div>
```

- плагин для Live Preview — работает только с Google Chrome. Вносим какие-либо изменения в код в редакторе — в окне браузера автоматически отображаются изменения;
- подсказки при редактировании CSS, JS и HTML-файлов;

Инструментарий



WebStorm – среда разработки для JavaScript равно подходящая как для frontend'а, так и для создания приложений на Node.js и мобильных приложений на React Native. Этот инструмент разработан компанией JetBrains и является платным.

Многие разработчики считают **WebStorm** лучшей IDE для веб-разработки.

Функционал

Live Edit позволяет мгновенно видеть результаты ваших изменений в CSS, HTML и JavaScript файлах проекта в браузере без перезагрузки страницы. Live Edit доступен в режиме отладки JavaScript и работает в браузере Google Chrome с предустановленным плагином JetBrains IDE Support.

WebStorm поддерживает современные языки, компилируемые в JavaScript, такие как **TypeScript**, **CoffeeScript** и новую версию JavaScript – **ECMAScript 2017**, предлагая автодополнение кода, проверку ошибок, рефакторинг и отладку.

«WEB-программирование»



Основные понятия

Основные виды браузеров и особенности их использования

Обзор инструментария



Технологии Web-программирования

Основные вехи WEB

Перечень лабораторных работ

Технологии Web-программирования



- Hypertext Markup Language (HTML): используется для написания WEB-страниц

- XHTML: Более новая, более тщательно стандартизированная спецификация HTML

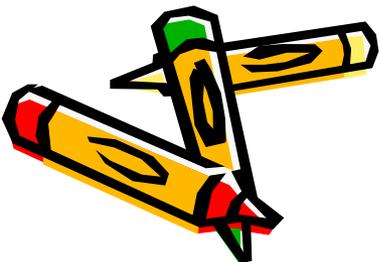
- Cascading Style Sheets (CSS): содержит стилистическое оформление web-страниц

- Javascript: позволяет создавать интерактивные и программируемые web-страницы



Технологии Web-программирования

- Asynchronous Javascript and XML (AJAX): подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате, при обновлении данных, веб-страница не перезагружается полностью, и веб-приложения становятся более быстрыми и удобными.
- **PHP** (*Hypertext Preprocessor*) позволяет WEB-серверу динамически создавать страницы
 - Structured Query Language **SQL** - обеспечивает взаимодействие с базами данных.

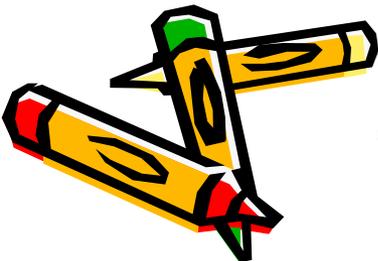


Технологии Web-программирования

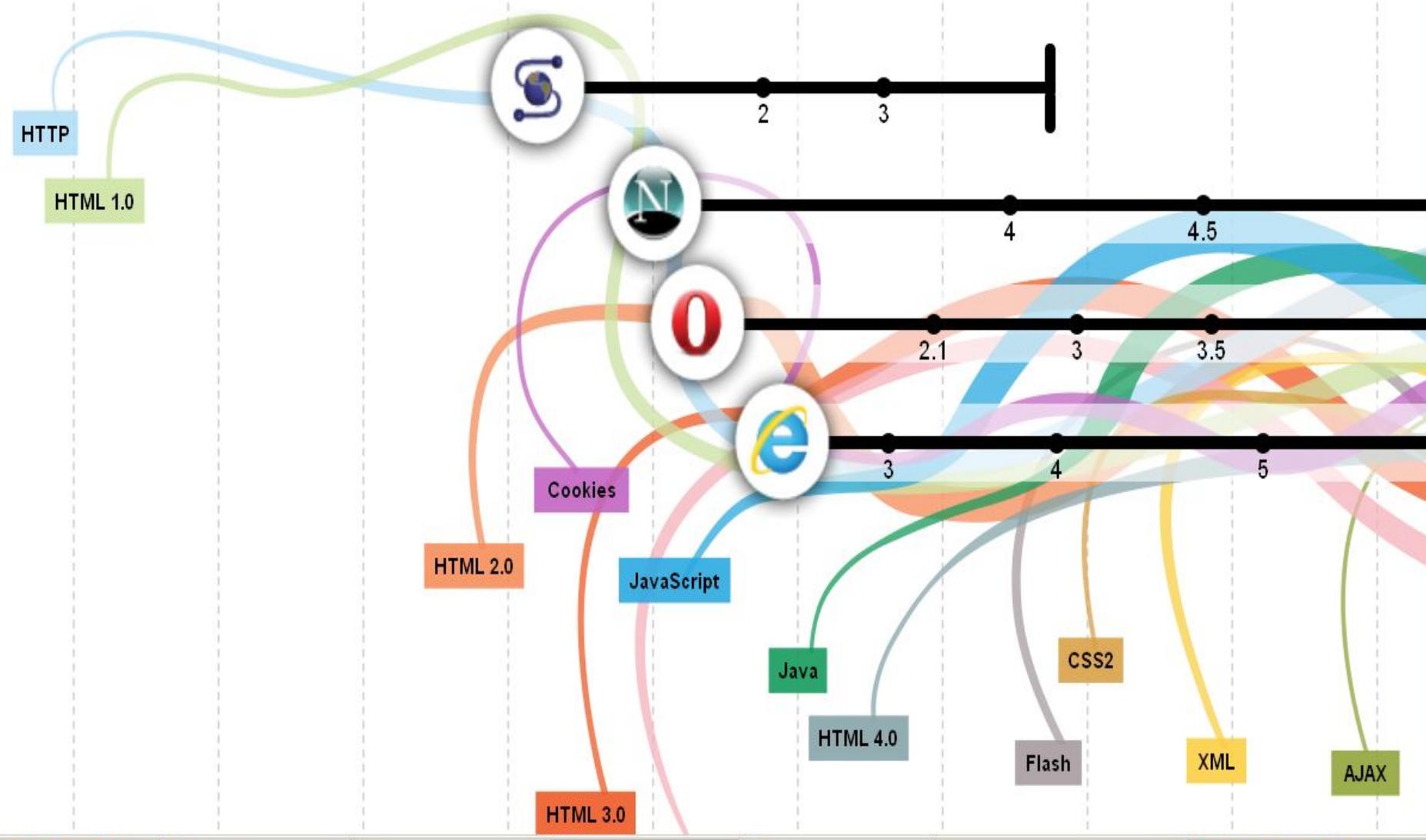
Следующие 3 страницы предоставляют вам ретроспективу развития WEB-технологий. Найти ее вы можете по адресу:

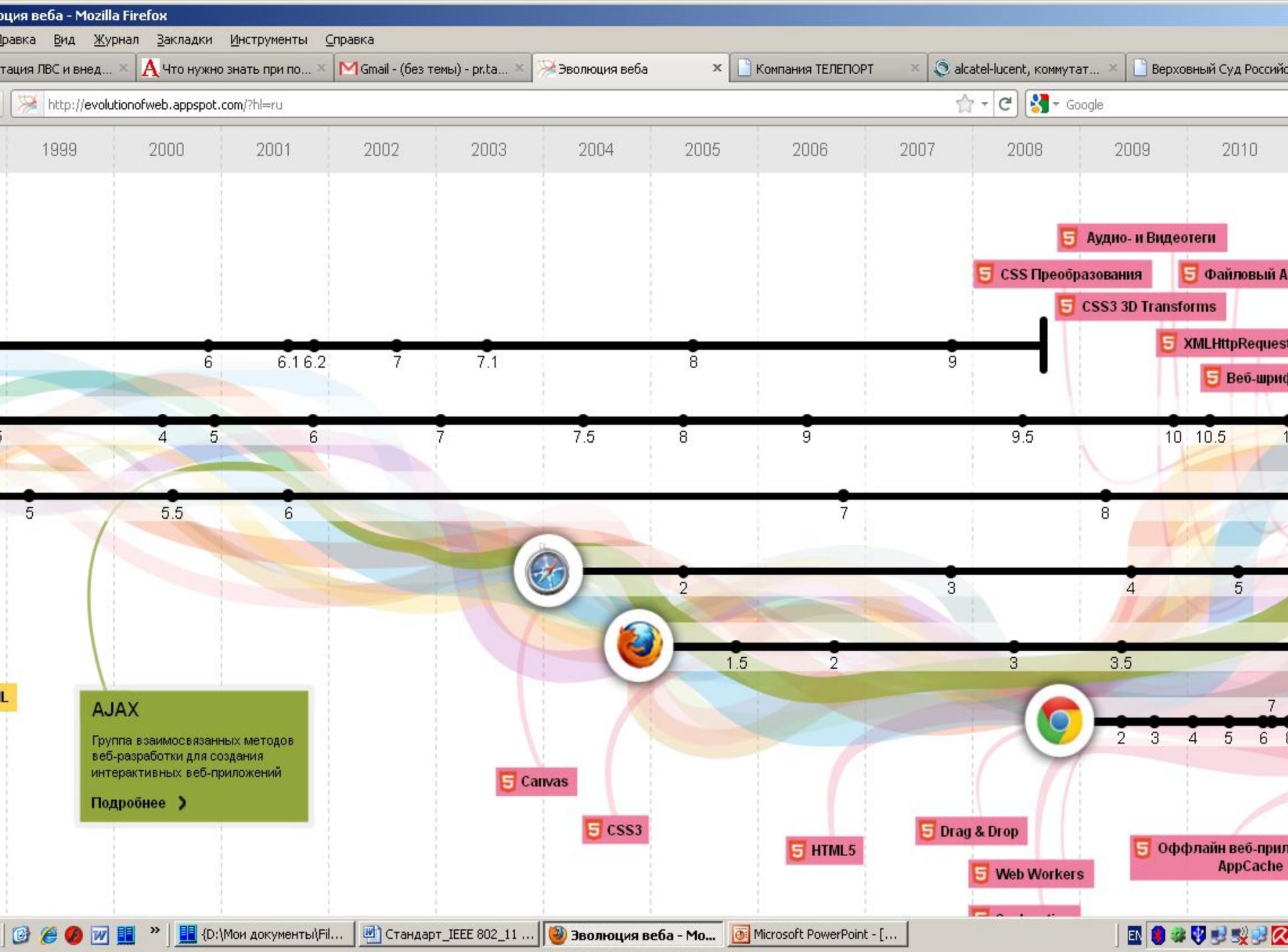
<http://evolutionofweb.appspot.com/>

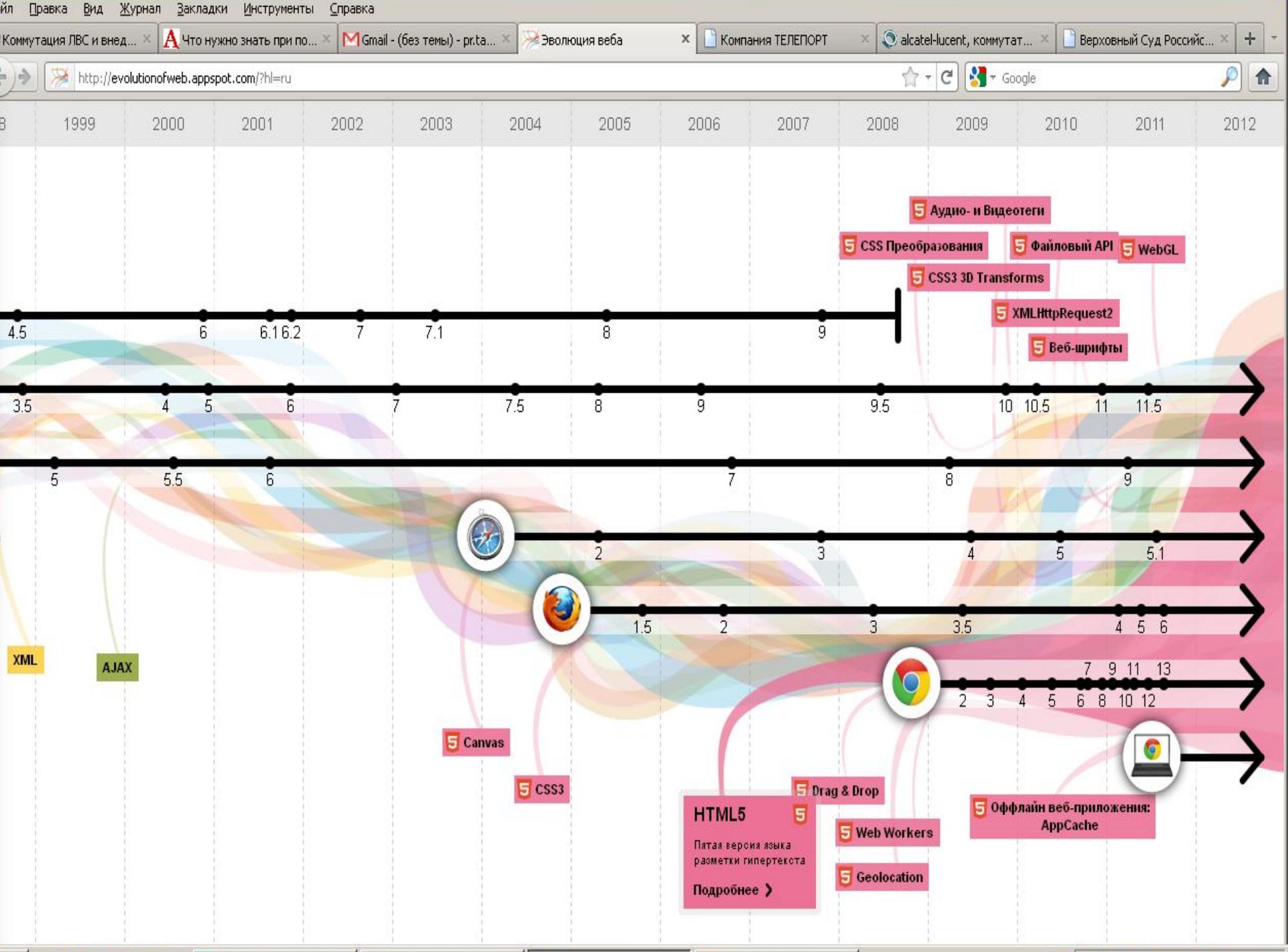
Цветные ленты на инфографике представляют взаимосвязи между веб-технологиями и браузерами, благодаря которым нам доступно столько функциональных веб-приложений.

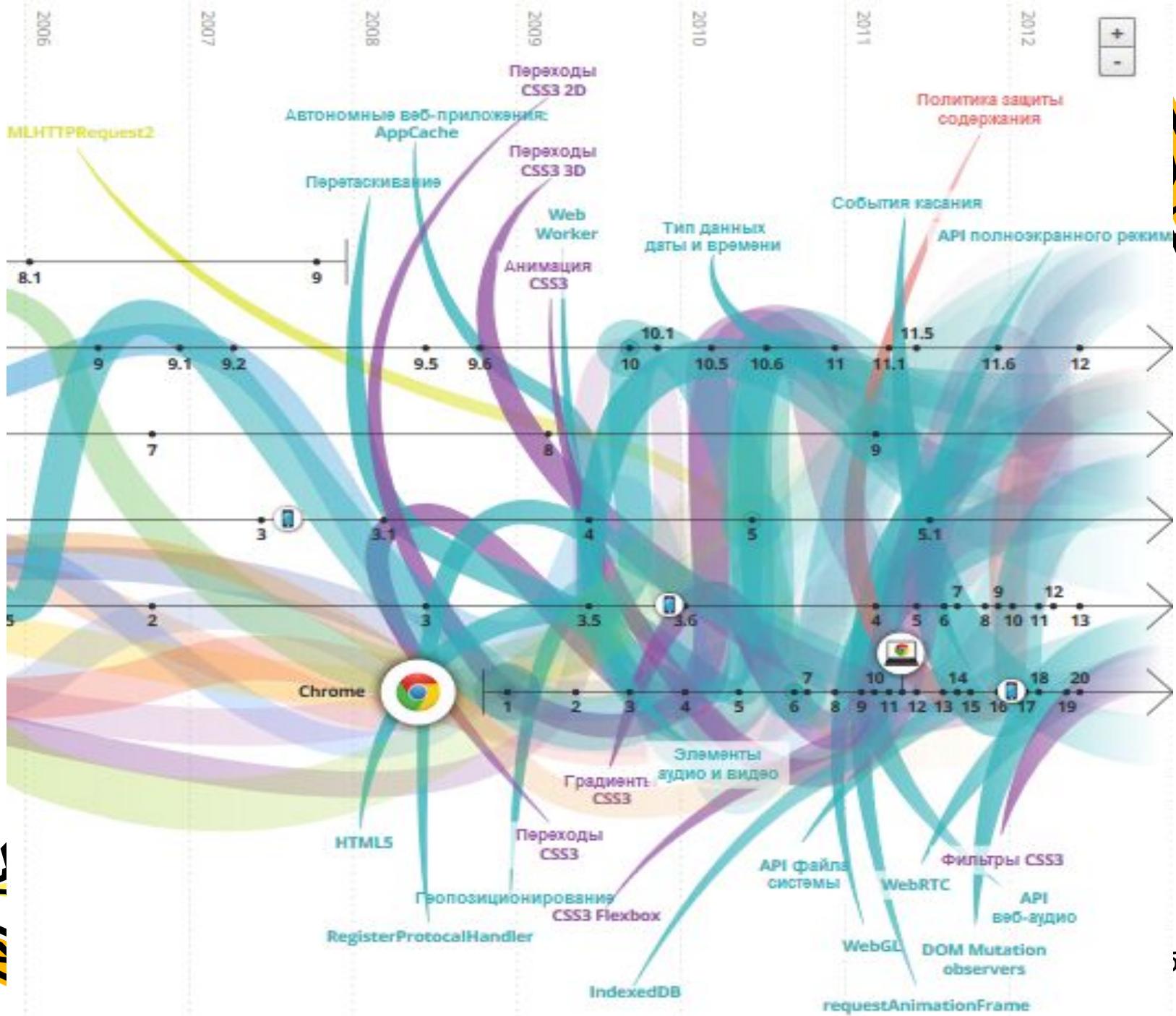


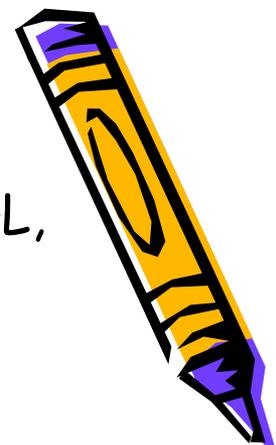
1990 1991 1992 1993 1994 1995 1996 1997 1998 1999



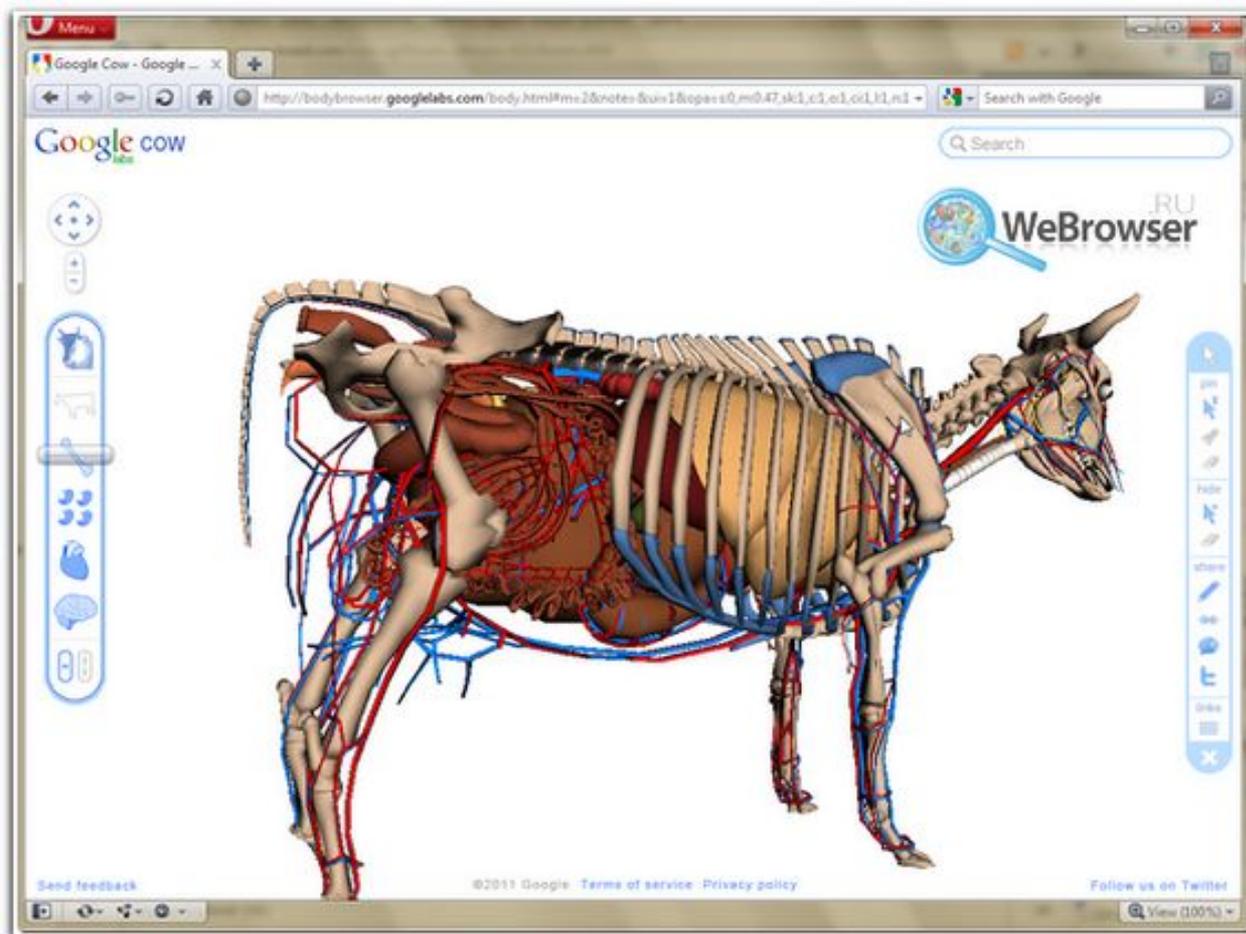




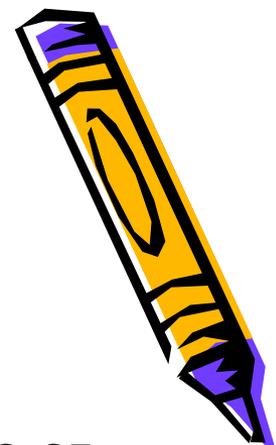




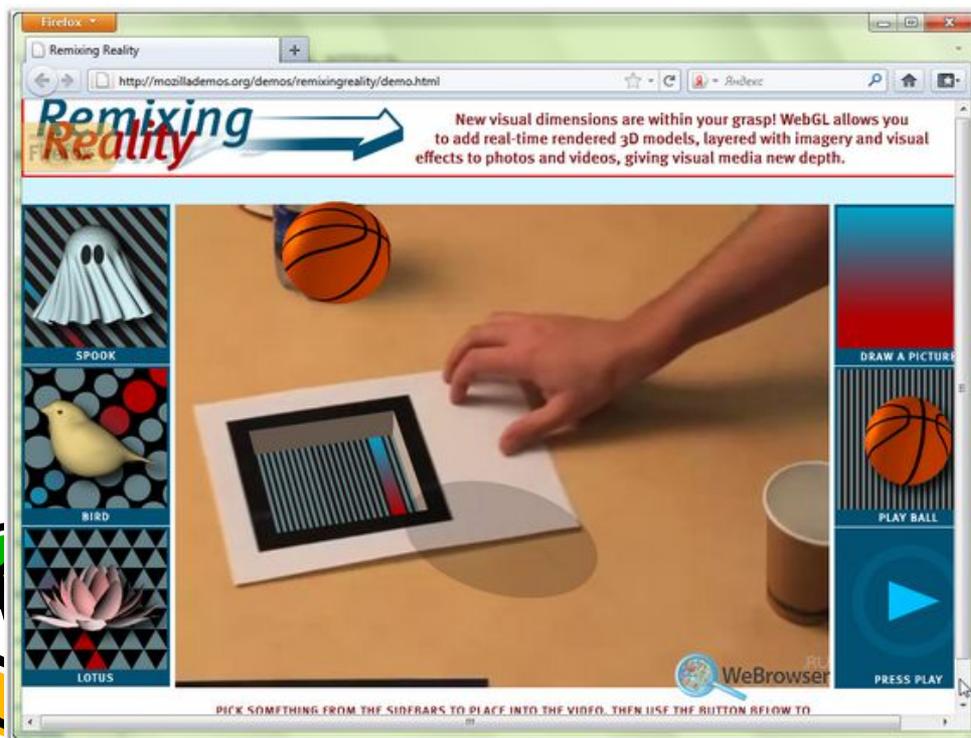
Среди совсем недавно появившихся технологий - WebGL, о котором много говорят последнее время. Очень эффективный пример использования - <http://bodybrowser.googlelabs.com/moo.html>



Демонстрация строения тела коровы. Позволяет бегунком слева отображать тело «по уровням». Объект можно отдалять и приближать, а также перемещать мышью.



Итак, **WebGL** позволяет получать доступ к ресурсам видеокарты для отображения в реальном времени 3D-графики на интернет-страницах. На данный момент, в финальных версиях всех браузеров поддержка WebGL имеется только в **Firefox** и **Chrome**, **Opera** и **Apple**. Явно от реализации уклонилась пока **Microsoft**, но, скорее всего, передумают и они.



Взаимодействие видео с 3D-графикой и JavaScript-объектами. Первые шаги к расширенной реальности в Интернете.

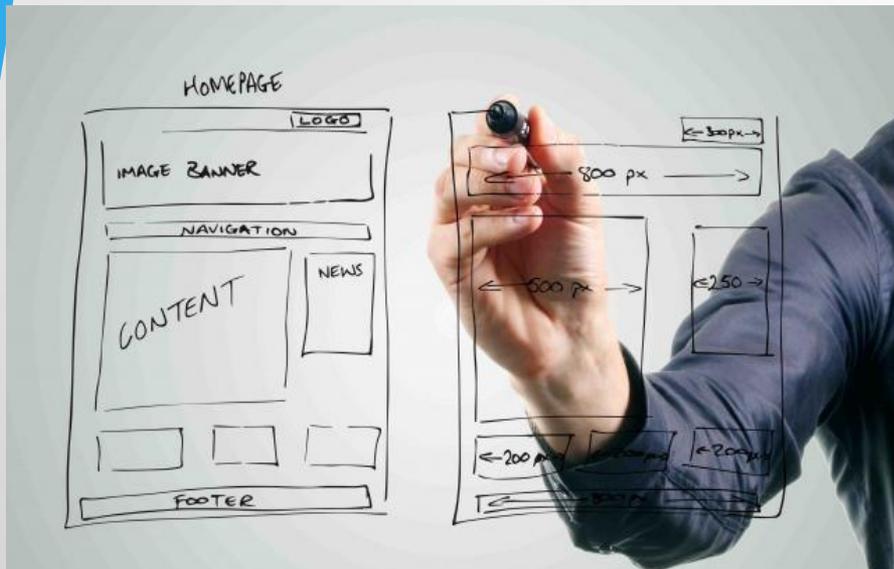


Направления в WEB – разработке

Направления в WEB – разработке

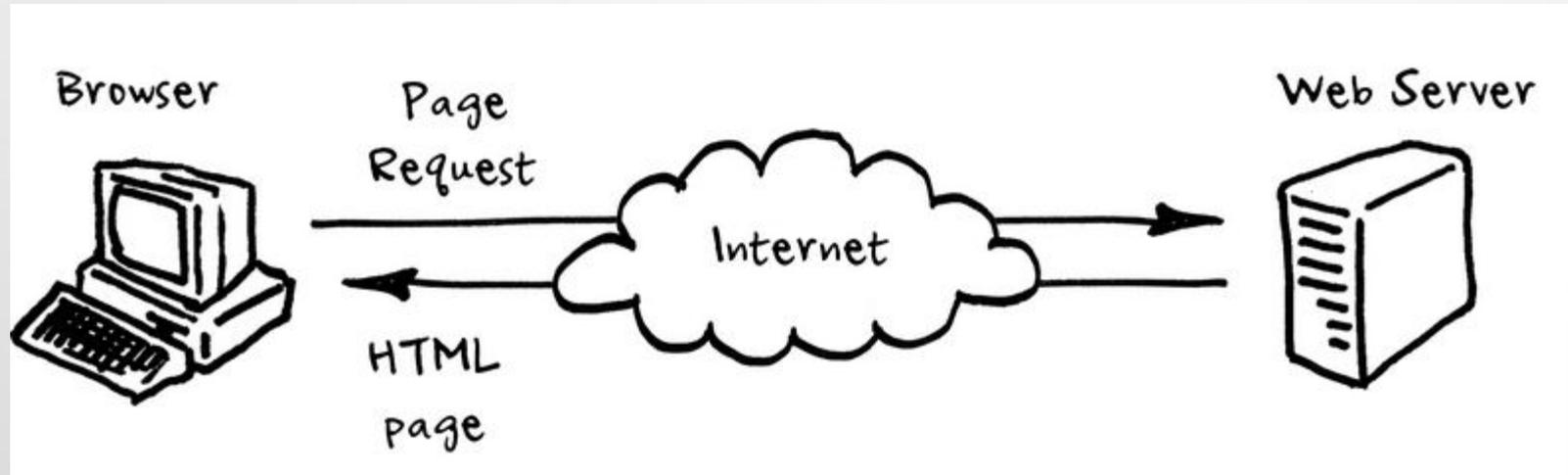
Фронтенд-разработчик

Бекенд-разработчик



Направления в WEB – разработке

Структура web-приложений: этап 1



Frontend

- Отображение статических HTML-страниц (HTML, CSS)
- Контент на страницах, переходы по гиперссылкам

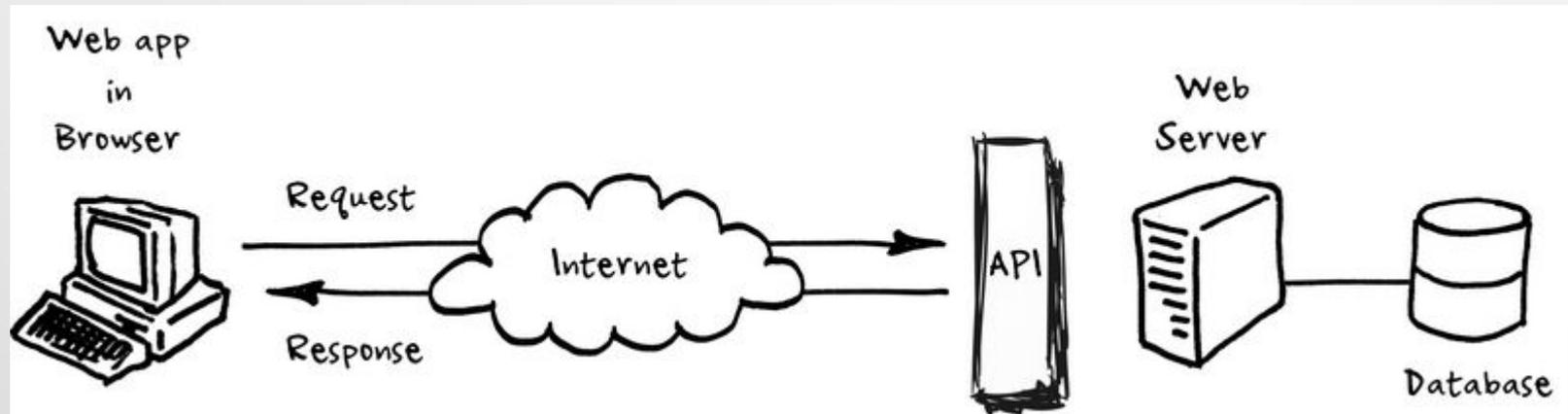
Backend

- Хранение статических документов и отдача по запросу по протоколу HTTP

На данном этапе сайты назывались web-ресурсами

Направления в WEB – разработке

Структура web-приложений: этап 2



Frontend

- Отображение статических HTML-страниц (HTML, CSS)
- Контент на страницах, переходы по гиперссылкам
- Взаимодействие с сервисом посредством форм

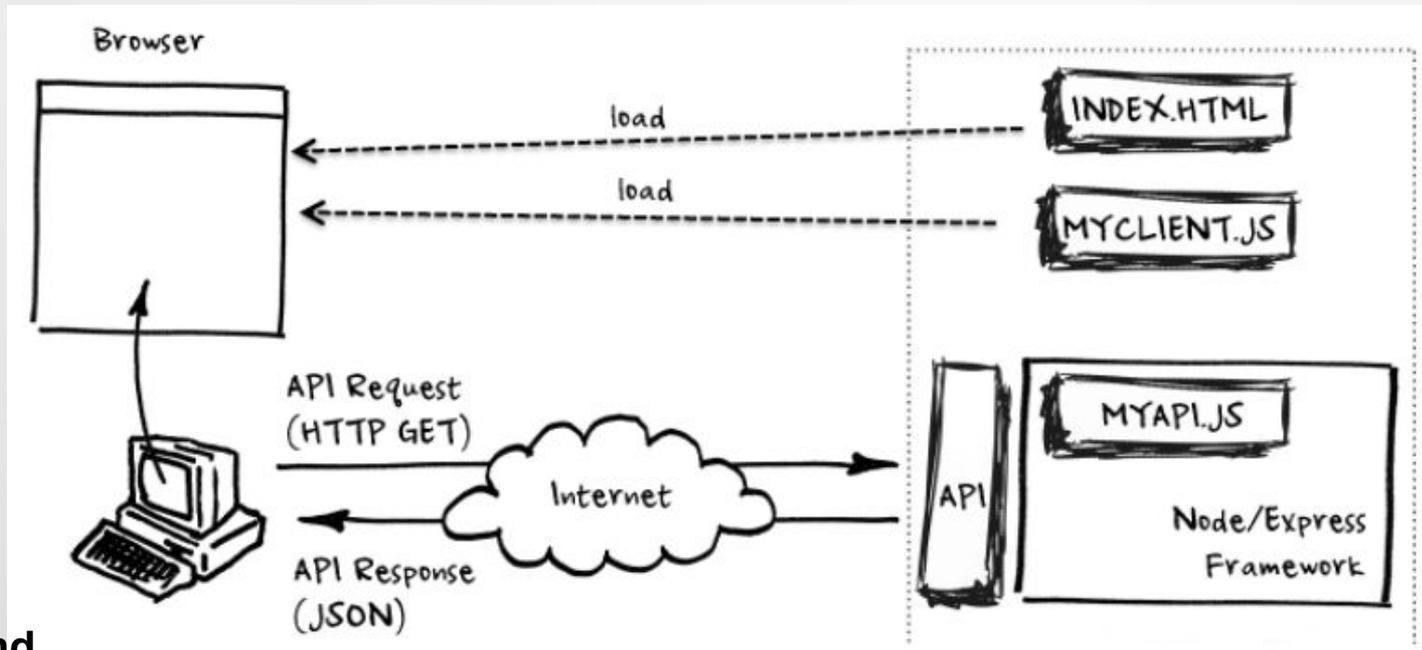
Backend

- Хранение статических документов и отдача по запросу по протоколу HTTP
- **Обработка пользовательских запросов и генерация динамических страниц**
- **Хранение данных в базе данных.**

На данном этапе сайты назывались web-сервисами

Направления в WEB – разработке

Структура web-приложений: этап 3



Frontend

- Хранение и доступ к статическому контенту (файлы стилей, скрипты)
- Генерация и отображение пользовательского интерфейса
- Взаимодействие с пользователем и выполнение запросов к API
- Обновление пользовательского интерфейса в ответ на действия пользователя.

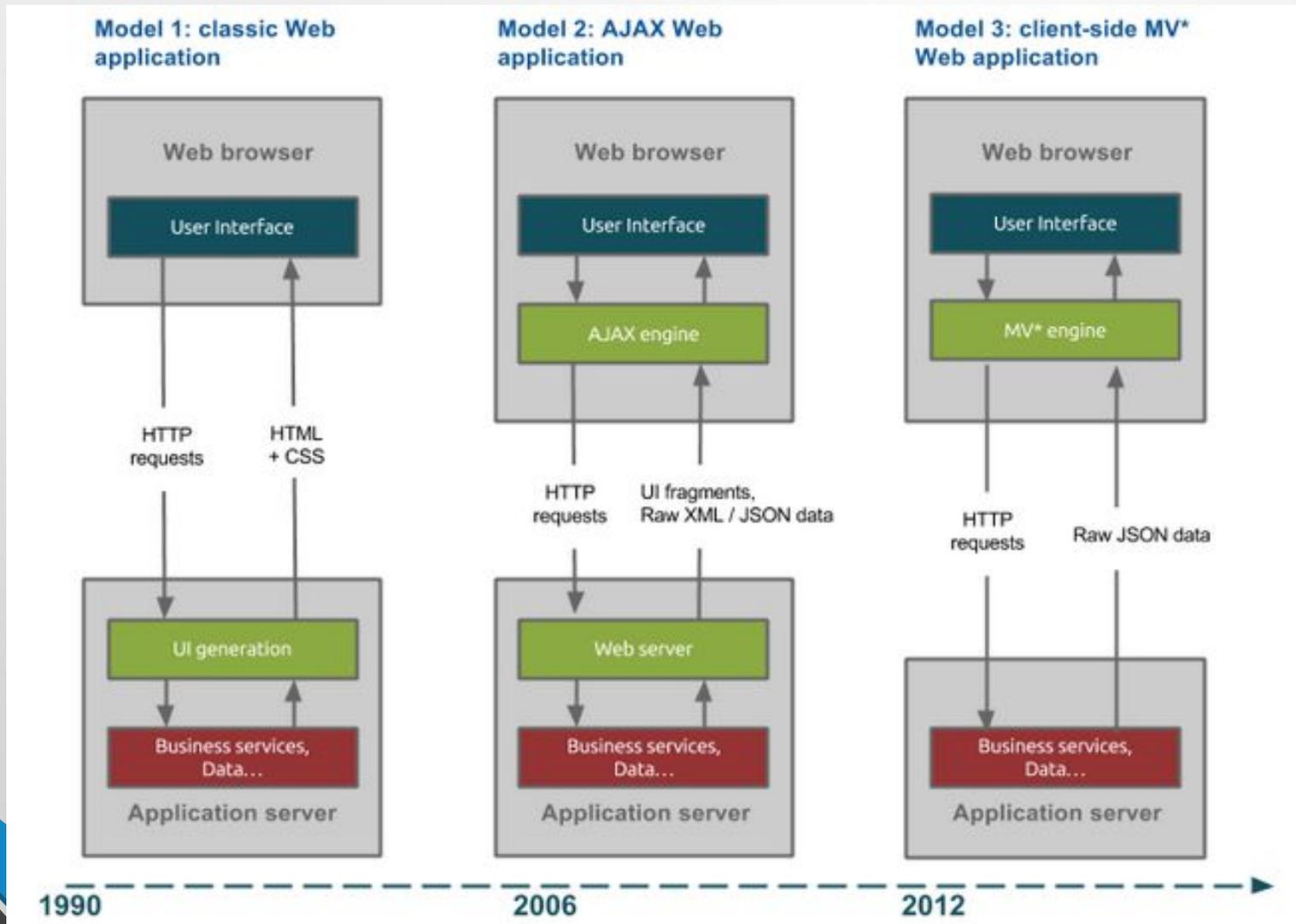
Backend

- Реализация публичного API
- Хранение данных в базе данных и работа с ними

Теперь сайты называют web-приложениями

Направления в WEB – разработке

Структура web-приложений



Направления в WEB – разработке



Часть работодателей не отличают (или не хотят отличать) верстальщика от фронтенд-разработчика:

- **Верстальщик** — боец узкого фронта. Его задача — сверстать полученный от дизайнера макет, используя HTML+CSS. Он, возможно, немного умеет в JavaScript, но чаще ограничивается умением прикрутить какой-нибудь плагин jQuery.
- **Фронтенд-разработчик** — это универсальный солдат. Он и макет завершает, и веб-приложение построит, и серверную часть, если надо, освоит. Знать надо немало: HTML, CSS, JavaScript, библиотеки и фреймворки JS, препроцессоры и фреймворки CSS, системы контроля версий и таскраннеры, технологии бэкенда, юнит-тестирование и многое другое.

Направления в WEB – разработке



Фронтенд-разработчик

Карьерный путь фронтендера обычно начинается с верстальщика — это самый логичный и общепринятый вариант. Сначала изучается связка **HTML+CSS**, затем на нее «наслаиваются» знания **JavaScript**, библиотек и фреймворков. Будущий специалист также изучает ключевые понятия построения серверной части, добавляет сюда инструменты, необходимые для выбранной специализации. Затем все это шлифуется умением работать с контролем версий, графическими редакторами и пониманием принципов UI/UX-дизайна.

User Experience Design в переводе означает «опыт взаимодействия» и включает в себя различные UX-компоненты: информационную архитектуру, проектирование взаимодействия, графический дизайн и контент.

Направления в WEB – разработке

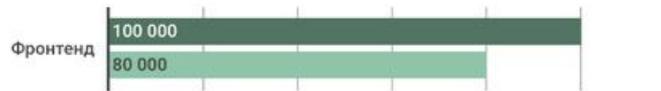
Фронтенд-разработчик



Направления в WEB – разработке



Средняя зарплата фронтенд-специалиста по России, рублей/месяц



Средняя зарплата фронтенд-специалиста по Москве, рублей/месяц



По данным «Моего круга»

Направления в WEB – разработке

Фронтенд-разработчик

Какие технологии должен освоить фронтенд-разработчик:

- HTML и CSS (в том числе сетки и CSS-фреймворки, спецификации W3C и WHATWG, HTML5/CSS3 Polyfills)
- Препроцессоры CSS (Sass, Less, Stylus и т. д.)
- JavaScript
- Популярные фреймворки и библиотеки: jQuery, Angular.JS, React.JS, Backbone.js и т. д.)
- SVG
- DOM
- HTML5 API
- ECMAScript 6
- Популярные CMS (WordPress, Drupal, Joomla и т.д.)
- Понимать принципы построения бекэнда и разбираться в серверных технологиях (Node.js, PHP, Ruby, .NET и т. д.)
- Инструменты дебаггинга (Chrome Dev Tools, Firebug и прочие)
- Инструменты контроля версий (Git, GitHub, CVS и т. д.)
- Базы данных и языки запросов (SQL, MySql, NoSQL, MongoDB и т. д.)
- Графические редакторы (Photoshop, Illustrator и т. д.)

Направления в WEB – разработке

Back-end разработчик



Back-end разработчики несут ответственность за создание "серверной" части в веб-приложениях, иначе говоря, они имеют дело со всем, что относится к программно-административной части веб-приложения, внутреннему содержанию системы, серверным технологиям — базам данных, архитектуре, программной логике.

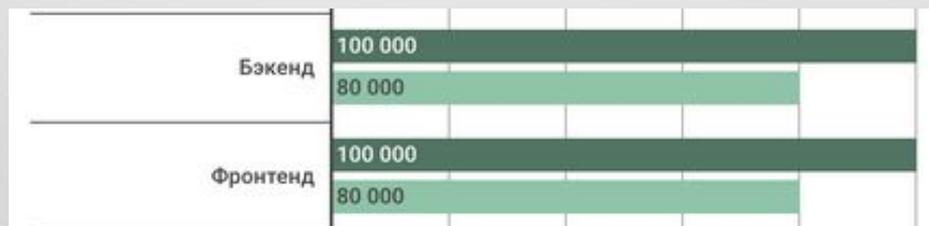
Back-end разработчик должен знать серверные языки программирования, такие как PHP, Python, Ruby, Java, Perl, Node JS (программные платформы). Дополнительно к Node JS этому специалисту полезно будет изучить Express (библиотеку для взаимодействия платформы Node JS с сервером) и Mongo DB (базу данных для получения и хранения информации). В качестве дополнительных средств применяются фреймворки Symfony, Codeigniter, Yii, Zend Framework, Kohana и др.

Направления в WEB – разработке

Back-end разработчик

Основные навыки:

- Свободное владение серверными языками PHP, Python, Ruby, Java, Perl, Node JS, а также фреймворками Symfony, Codeigniter, Yii, Zend Framework, Kohana и др;
- Понимание интерфейсных веб-технологий, таких как HTML, CSS, и JavaScript;
- Навыки работы с большими объемами информации;
- Способность организовывать серверное окружение, включая администрирование баз данных, а также масштабирование приложений;
- Знание всех стандартов безопасности;
- Опыт работы с системами контроля версий, таких как Git;
- Знание английского языка для работы с технической документацией.



Средний уровень ЗП по России

Направления в WEB – разработке

DevOps



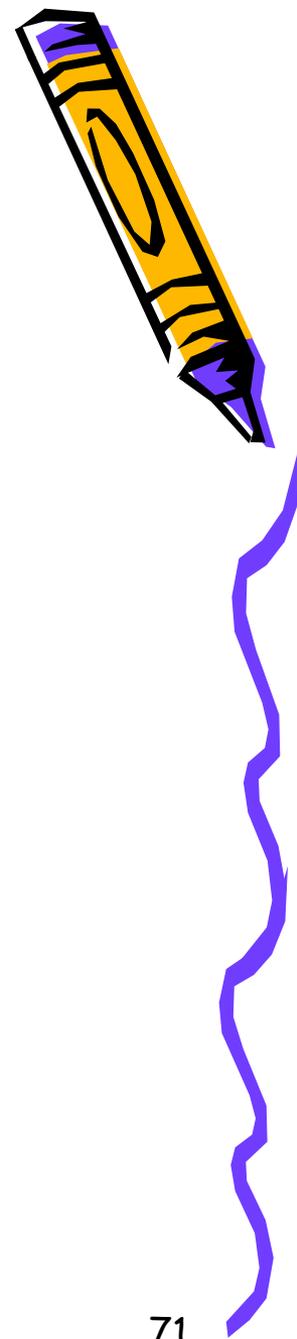
Иллюстрация, показывающая DevOps как пересечение разработки, эксплуатации и тестирования

DevOps (акроним от *development* и *operations*) — набор практик, нацеленных на активное взаимодействие специалистов по разработке со специалистами по информационно-технологическому обслуживанию и взаимную интеграцию их рабочих процессов друг в друга. Базируется на идее о тесной взаимозависимости разработки и эксплуатации программного обеспечения и нацелен на то, чтобы помогать организациям быстрее создавать и обновлять программные продукты и услуги.

Направления в WEB – разработке DevOps

1. Поскольку DevOps — это командная работа (между сотрудниками, занимающимися разработкой, операциями и тестированием), нет единого инструмента «DevOps»: это скорее набор (или «инструментальная цепочка DevOps»), состоящий из нескольких инструментов. Как правило, инструменты DevOps вписываются в одну или несколько из этих категорий, что отражает ключевые аспекты разработки и доставки программного обеспечения:
2. Code — разработка и анализ кода, инструменты контроля версий, слияние кода;
3. Build — инструменты непрерывной интеграции, статус сборки;
4. Test — инструменты непрерывного тестирования, которые обеспечивают обратную связь по бизнес-рискам;
5. Пакет — репозиторий артефактов, предварительная установка приложения;
6. Release — управление изменениями, официальное утверждение выпуска, автоматизация выпуска;
7. Конфигурация — Конфигурация и управление инфраструктурой, Инфраструктура как инструменты кода;
8. Мониторинг — мониторинг производительности приложений, опыт работы с конечным пользователем.

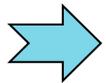
«WEB-программирование»



Основные понятия

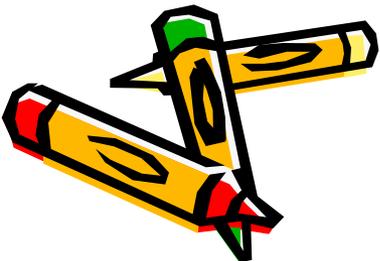
Основные виды браузеров и особенности их использования

Технологии Web-программирования

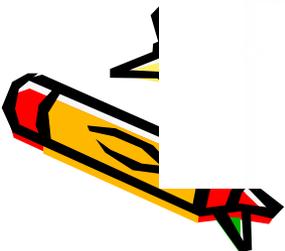
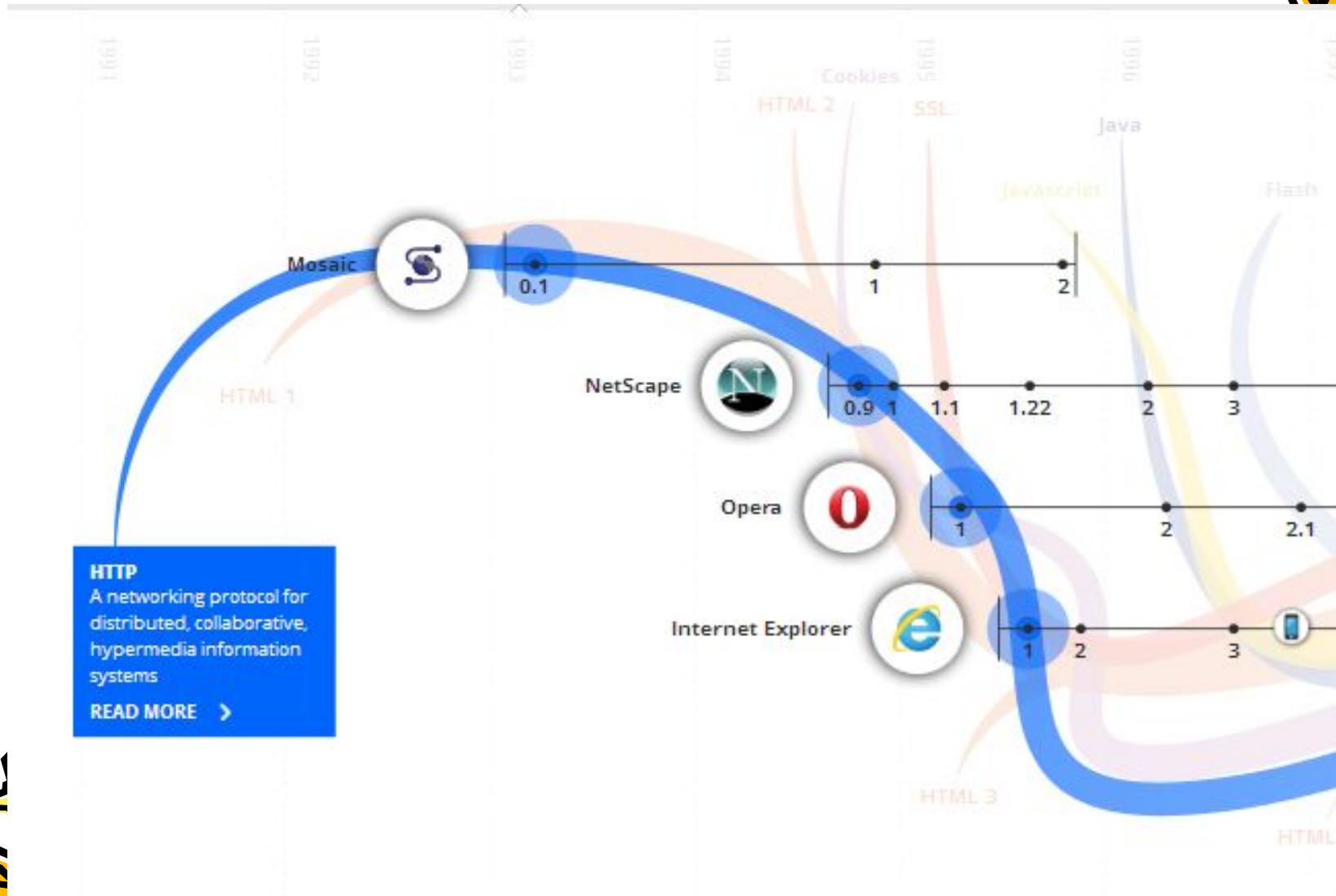
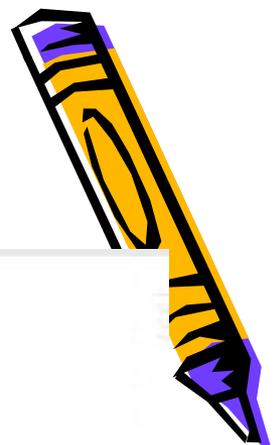


Основные вехи WEB

Перечень лабораторных работ



Основные вехи WEB



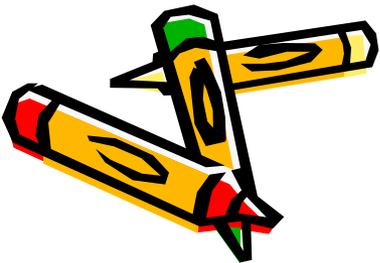


Основные вехи WEB

HTTP/0.9

HTTP был предложен в марте 1991 года **Тимом Бернерсом-Ли**, работавшим тогда в CERN (The European Organization for Nuclear Research), как механизм для доступа к документам в Интернете и облегчения навигации посредством использования гипертекста.

Самая ранняя версия протокола HTTP/0.9 была впервые опубликована в январе 1992 г. (хотя реализация датируется 1990 годом). Спецификация протокола привела к упорядочению правил взаимодействия между клиентами и серверами HTTP, а также чёткому разделению функций между этими двумя компонентами.



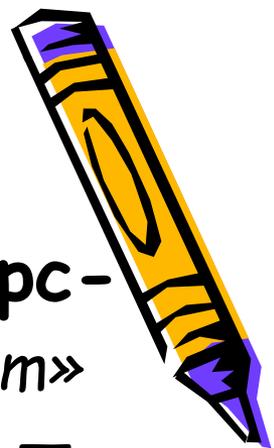
Были задокументированы основные синтаксические и семантические положения.

Основные вехи WEB



Сэр Тимоти Джон Бернерс-Ли (англ. *Sir Timothy John «Tim» Berners-Lee*; род. 8 июня 1955) — знаменитый британский учёный, изобретатель URI, URL, HTTP, HTML, изобретатель Всемирной паутины (совместно с Робертом Каиллау) и действующий глава Консорциума Всемирной паутины (W3C).

Автор концепции семантической паутины. Автор множества других разработок в области информационных технологий.



Основные вехи WEB

HyperText Markup Language — «язык разметки гипертекста», является приложением **SGML** (стандартного обобщённого языка разметки), соответствующего международному стандарту **ISO 8879**

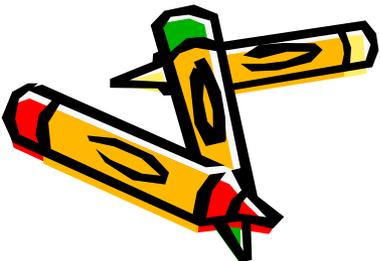
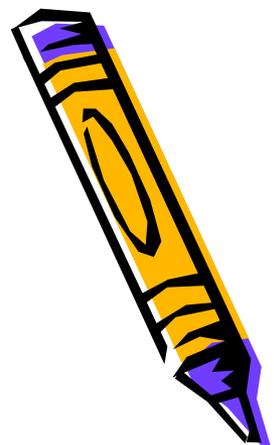
4 этапа в развитии HTML

1990-1995 - Начальное развитие («относительно хаотичное»)

1995-2000 - Стандартизация

2000-2005 - Поиск путей дальнейшего развития

2005-наст. время - Разработка нового поколения HTML (HTML 5)



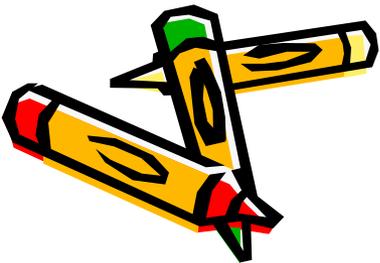
Основные вехи WEB

Первый WEB-браузер* - 1993г.

NCSA Mosaic (по-русски произносят [маза́ик]) — это первый веб-браузер под операционную систему Microsoft Windows с графическим интерфейсом пользователя и развитыми возможностями, на котором были основаны и Netscape Navigator, и Microsoft IE

Основными разработчиками Mosaic были Марк Андерссен и Эрик Бина, работавшие в Национальном центре применения суперкомпьютеров (*National Center for Supercomputing Applications*) Иллинойского университета.

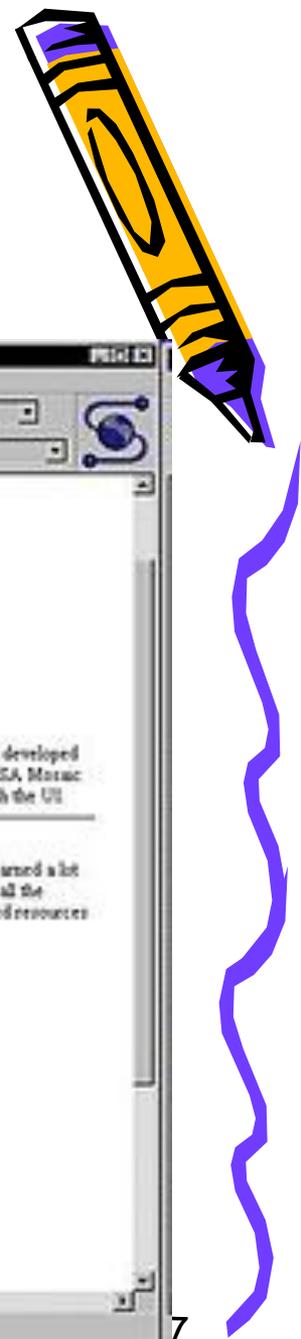
Были и другие примеры успешных реализаций графических браузеров, но все они - до Mosaic - имели лишь небольшой круг пользователей.



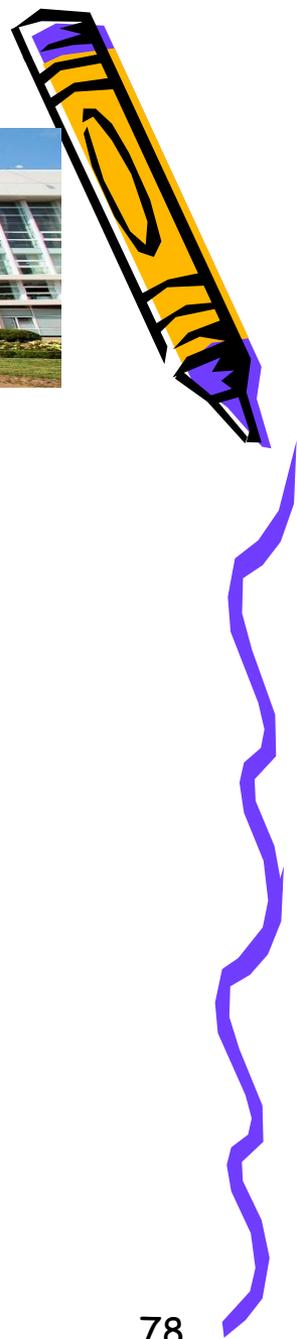


Основные вехи WEB

Первый WEB-браузер - 1993г.



«WEB-программирование»



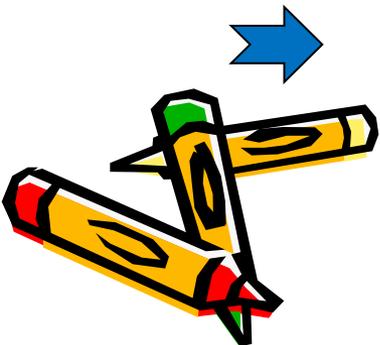
Основные понятия

Основные виды браузеров и особенности их использования

Технологии Web-программирования

Основные вехи WEB

Перечень лабораторных работ

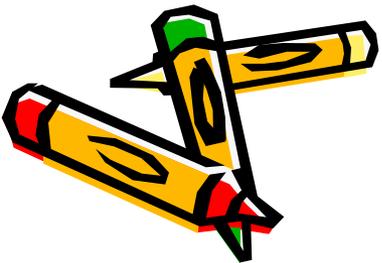
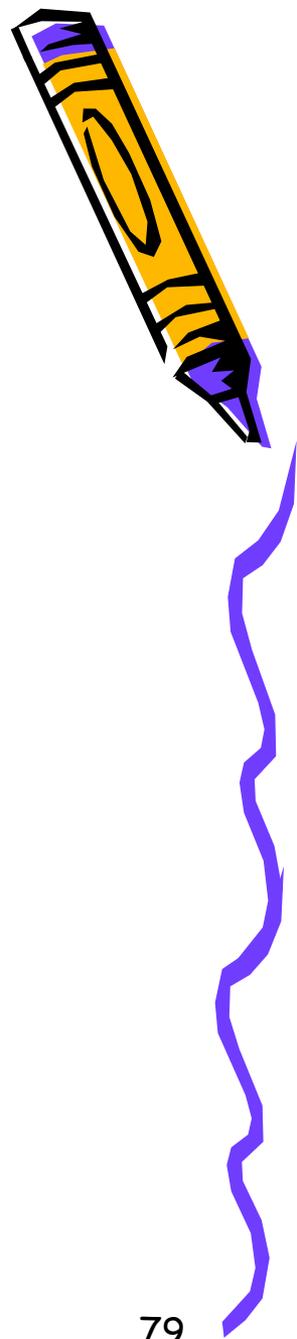


Структура курса

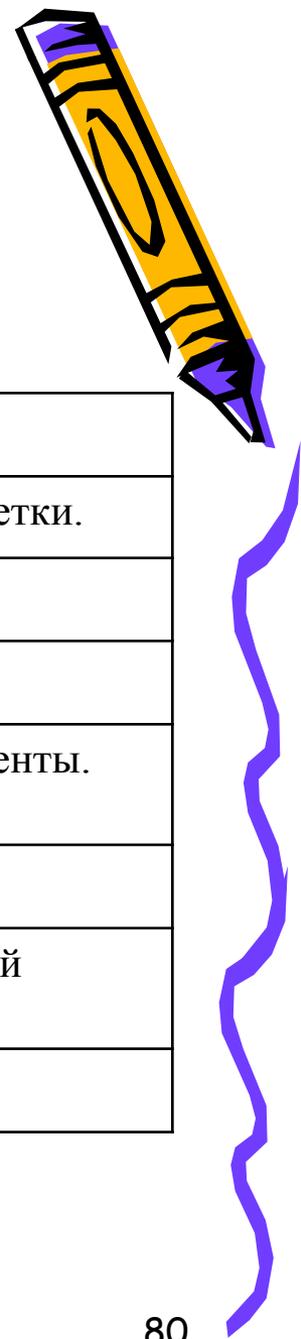
Курс разделен на 2 части:

- 1.Фронтэнд разработка
- 2.Бэкенд разработка

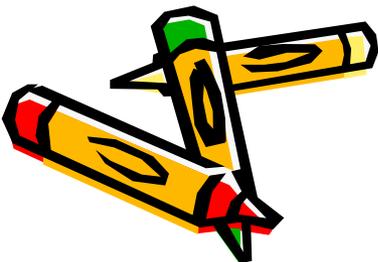
17 лекций и 17 лабораторных работ



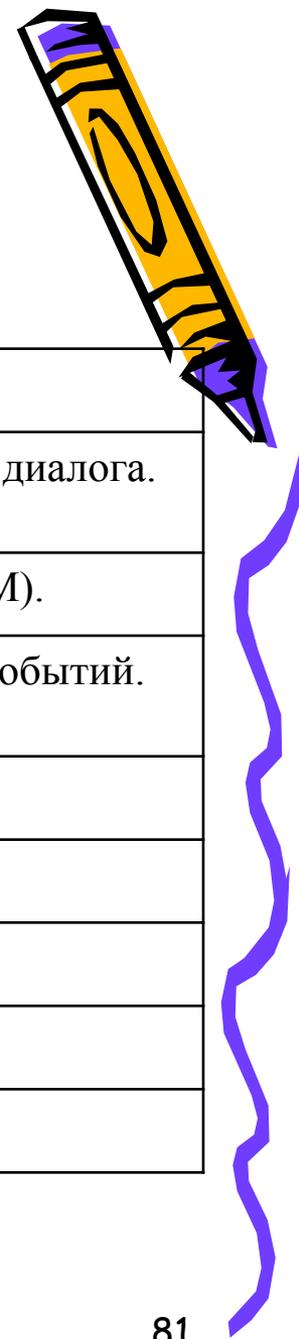
Перечень лабораторных работ к первой части курса



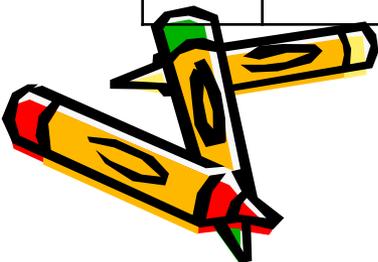
№	Название
1.	Введение в HTML(XHTML). Структура HTML-документа. Элементы разметки.
2.	HTML. Теги и атрибуты.
3.	CSS. Способы применения Каскадных таблиц к HTML-странице.
4.	CSS. Особенности использования селекторов. Псевдоклассы и псевдоэлементы. Блочная модель.
5.	CSS.Создание динамического меню навигации.
6.	HTML. Виды верстки: плоская, табличная, блочная. Изучение особенностей табличной и блочной верстки.
7.	Разработка персонального сайта студента.



Перечень лабораторных работ ко второй части курса

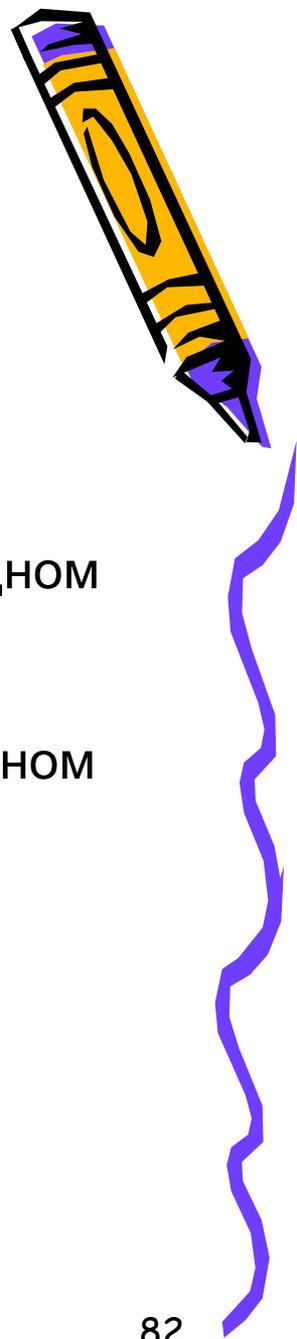


№	Название
8.	Типы данных в JavaScript. Изучение синтаксиса языка. Работа с окнами диалога. Объекты JavaScript. Дата-время. Работа с массивами.
9.	JavaScript. Объектная модель документа — Document Object Model (DOM).
10.	Динамический HTML. Формы, события, программирование обработки событий. Обработка форм с помощью регулярных выражений.
11.	Разработка серверной части приложения – PHP и Node.js
12.	
13.	
14.	
15.	Зачет.



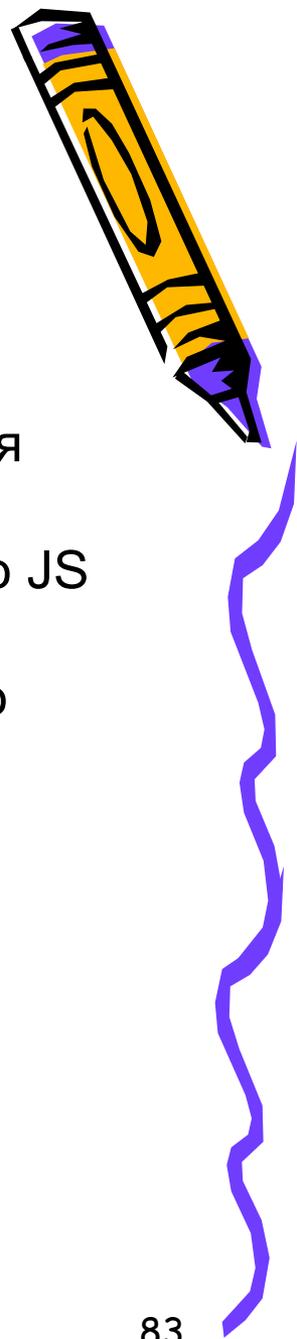
Использованные источники

1. Как стать веб-разработчиком в 2017 году — план действий. В свободном доступе:
<https://habrahabr.ru/company/netologyru/blog/328426/>
2. Что должен уметь фронтенд-разработчик. В свободном доступе:
<https://habrahabr.ru/company/netologyru/blog/327294/>
3. Brackets для сомневающихся и новичков. В свободном доступе: <https://habrahabr.ru/post/242623/>



Полезные ссылки

1. [Скринкаст](#) по Node.js
2. [MDN](#) (Mozilla Developer Network — веб-технологии для разработчиков)
3. [learn.javascript.ru](#) — основной источник информации о JS для начинающих
4. [WebReference.ru](#) — основной источник информации о HTML5 и CSS3 для начинающих



Вопросы?

