

«JavaScript в веб-разработке»



JavaScript
Courses

www.courses.dp.ua

О чём курс?

JavaScript – язык программирования (в отличии от **HTML** и **CSS**), поэтому фундамент нашего курса – основы программирования на базе языка **JavaScript**.

У **JavaScript** не обычная сфера применения - он работает в браузере и «управляет» HTML-документом. Как это происходит – также часть нашего курса.

JavaScript – инструмент для решения задач, поэтому в третьей части нашего курса мы поговорим о **практиках** его применения при решении задач возникающих перед веб-разработчиком.

А если

конкретнее:
*Базовый JavaScript. Основы программирования.
Базовые концепции. JavaScript как язык
программирования.*

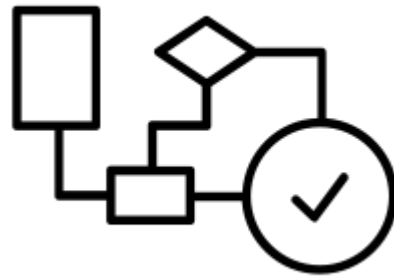
*Клиентский JavaScript. JavaScript в браузере.
Инфраструктура браузера. Объектная модель
документа (DOM), событийная модель, AJAX,
работа с Web API и др.*

*Технологии на базе JavaScript. JavaScript
библиотеки и фреймворки (jQuery, AngularJS).
JavaScript на стороне сервера/back-end (Node.js).
Хранённые данных (MongoDB, BaaS)*

План нашего курса



JavaScript



ECMAScript-5/2015/2016

...



HTML



CSS



+



План нашего курса



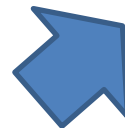
JavaScript



AJAX

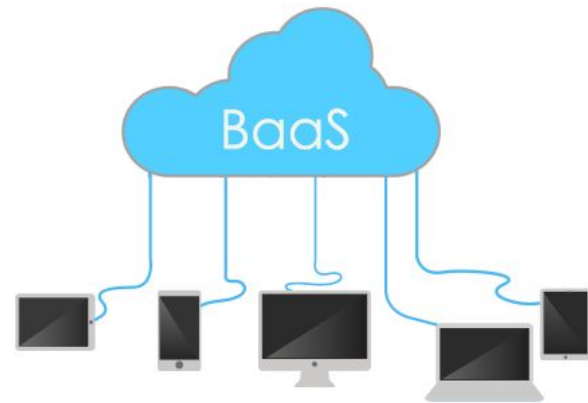


План нашего курса



План нашего курса

 mongoDB®



BaQend

На кого ориентирован курс

На тех кто знаком с языками HTML и CSS, кто умеет сверстать страницу и знаком с Bootstrap'ом.

Акценты

**Что является
основой**

**Понять суть и
принципы**

Кто проводит курс?

Анатолий Кигель

>11 лет в сфере веб-разработки.

7 лет опыта преподавательской деятельности в НГУ.

Технический консультант дизайн-студии «Свой стиль».

anatoliy.kigel@gmail.com

Взаимодействие

Презентация доступна по
адресу:

courses.dp.ua

Ваша группа:

JS  vk.com/js.courses

*Группа для вопросов, обсуждений, объявлений
(и презентации там тоже будут).*

О

вопросах
Хороший вопрос принесёт знания,
плохой вопрос добавит путаницу и
проблемы.

Хороший вопрос

У меня есть проблемы с обработкой события `onhappens` я пишу такой код:

```
tag.onhappens = function(e) {}
```


, однако при наступлении события этот обработчик не вызывается, в чём может быть причина?

Плохой вопрос

Я там сбросил свой сайт, он чёта не работает, гляньте и поправьте там, чтобы работал.

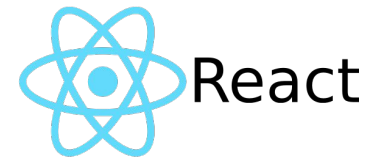
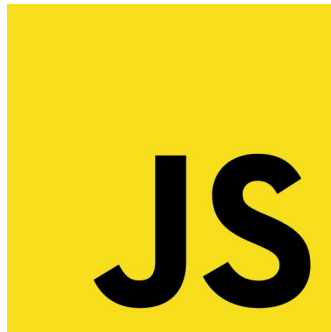
Зачем нужен JavaScript

HTML + CSS + JavaScript



*Безальтернативная тройка технологий **front-end**.
Технологии которые работают в браузере
пользователя.*

«Чистый» JS против JS-библиотек/фреймворков



Чистый JS позволяет сделать всё то, что предлагают JS-библиотеки/фреймворки.

HTML

HTML (*HyperText Markup Language*) – язык разметки текста, по сути **структурирует** (определяет структуру текста) и выполняет роль **контейнера для текста** (данных, информации).

Язык **HTML** - инструкция для браузера, указывающая как отобразить текст в удобной для человека форме.

HTML

*Составные части
HTML*

Данные + Метаданные

Текст + Как текст отобразить

<title> Page title </title>

Синтаксис языка

HTML
состоит

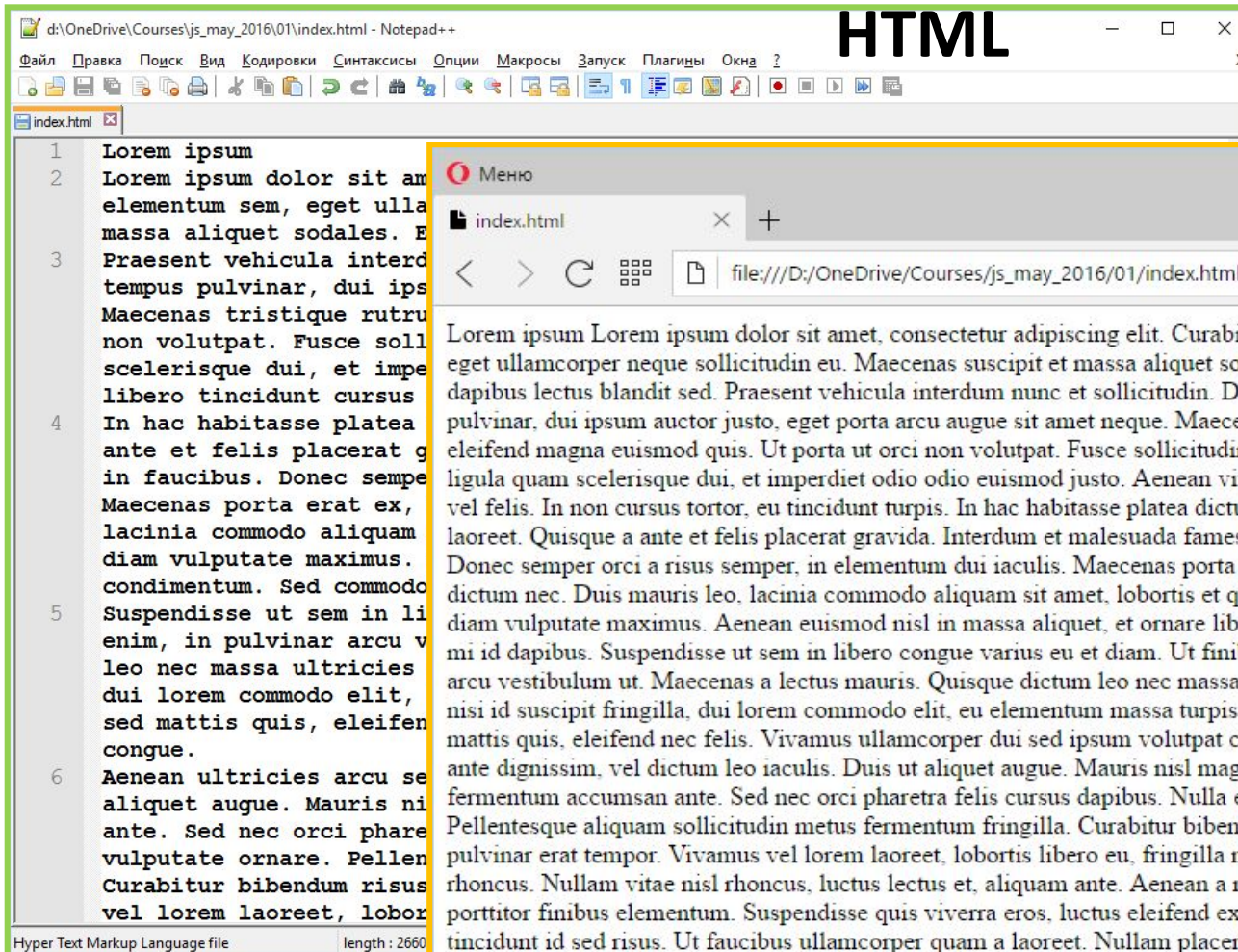
из:

```
<tag attr="value">Text data</tag>
```

Теги как контейнер для блока текста
+ **атрибуты** (*свойства, уточняют задачи тега, теги могут быть без атрибутов*);

Текстовые данные (содержимое, контент).

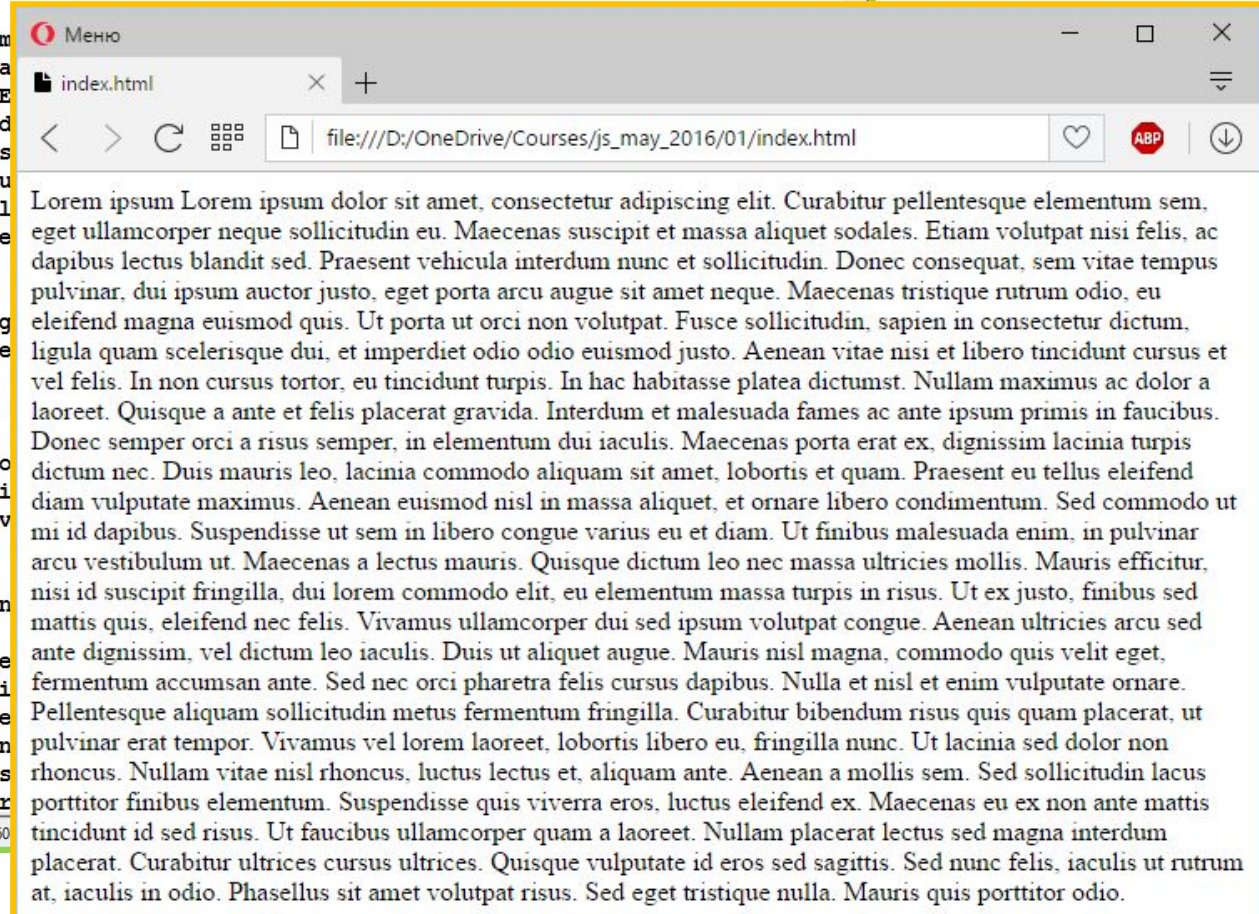
Цели и задачи языка



The image shows a Notepad++ window titled "index.html - Notepad++". The menu bar includes "Файл", "Правка", "Поиск", "Вид", "Кодировки", "Синтаксисы", "Опции", "Макросы", "Запуск", "Плагины", "Окна", and "?". The toolbar contains various icons for file operations and editing. The main text area shows the following HTML code:

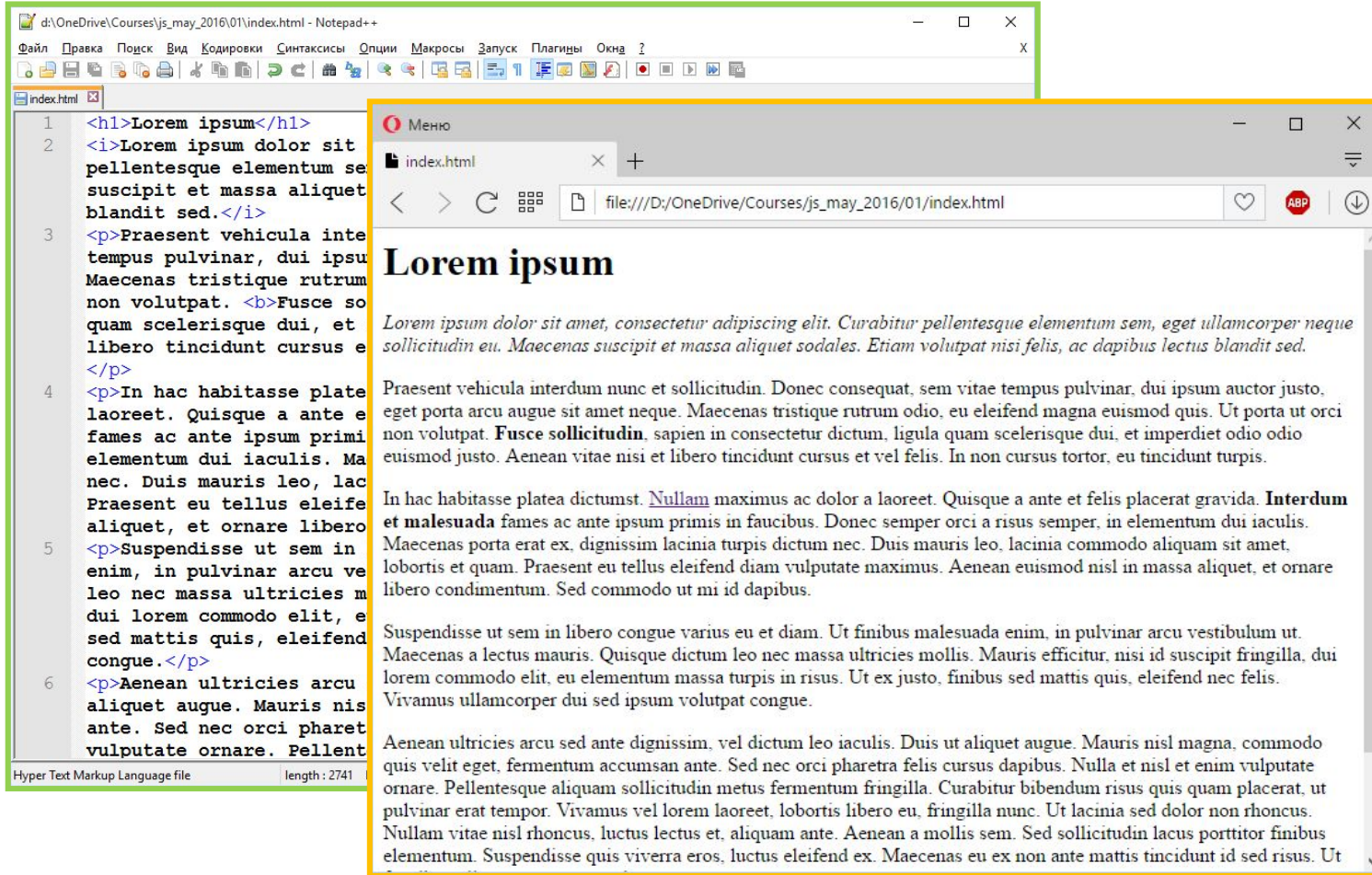
```
1 Lorem ipsum
2 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur pellentesque elementum sem,
  eget ullamcorper neque sollicitudin eu. Maecenas suscipit et massa aliquet sodales. Etiam volutpat nisi felis, ac
  dapibus lectus blandit sed. Praesent vehicula interdum nunc et sollicitudin. Donec consequat, sem vitae tempus
  pulvinar, dui ipsum auctor justo, eget porta arcu augue sit amet neque. Maecenas tristique rutrum odio, eu
  eleifend magna euismod quis. Ut porta ut orci non volutpat. Fusce sollicitudin, sapien in consectetur dictum,
  ligula quam scelerisque dui, et imperdiet odio odio euismod justo. Aenean vitae nisi et libero tincidunt cursus et
  vel felis. In non cursus tortor, eu tincidunt turpis. In hac habitasse platea dictumst. Nullam maximus ac dolor a
  laoreet. Quisque a ante et felis placerat gravida. Interdum et malesuada fames ac ante ipsum primis in faucibus.
  Donec semper orci a risus semper, in elementum dui iaculis. Maecenas porta erat ex, dignissim lacinia turpis
  dictum nec. Duis mauris leo, lacinia commodo aliquam sit amet, lobortis et quam. Praesent eu tellus eleifend
  diam vulputate maximus. Aenean euismod nisl in massa aliquet, et ornare libero condimentum. Sed commodo ut
  mi id dapibus. Suspendisse ut sem in libero congue varius eu et diam. Ut finibus malesuada enim, in pulvinar
  arcu vestibulum ut. Maecenas a lectus mauris. Quisque dictum leo nec massa ultricies mollis. Mauris efficitur,
  nisi id suscipit fringilla, dui lorem commodo elit, eu elementum massa turpis in risus. Ut ex justo, finibus sed
  mattis quis, eleifend nec felis. Vivamus ullamcorper dui sed ipsum volutpat congue. Aenean ultricies arcu sed
  ante dignissim, vel dictum leo iaculis. Duis ut aliquet augue. Mauris nisl magna, commodo quis velit eget,
  fermentum accumsan ante. Sed nec orci pharetra felis cursus dapibus. Nulla et nisl et enim vulputate ornare.
  Pellentesque aliquam sollicitudin metus fermentum fringilla. Curabitur bibendum risus quis quam placerat, ut
  pulvinar erat tempor. Vivamus vel lorem laoreet, lobortis libero eu, fringilla nunc. Ut lacinia sed dolor non
  rhoncus. Nullam vitae nisl rhoncus, luctus lectus et, aliquam ante. Aenean a mollis sem. Sed sollicitudin lacus
  porttitor finibus elementum. Suspendisse quis viverra eros, luctus eleifend ex. Maecenas eu ex non ante mattis
  tincidunt id sed risus. Ut faucibus ullamcorper quam a laoreet. Nullam placerat lectus sed magna interdum
  placerat. Curabitur ultrices cursus ultrices. Quisque vulputate id eros sed sagittis. Sed nunc felis, iaculis ut rutrum
  at, iaculis in odio. Phasellus sit amet volutpat risus. Sed eget tristique nulla. Mauris quis porttitor odio.
```

At the bottom of the window, it says "Hyper Text Markup Language file" and "length : 2660".



*Без HTML, веб-страница всего лишь
длинная текстовая строка.*

Цели и задачи языка



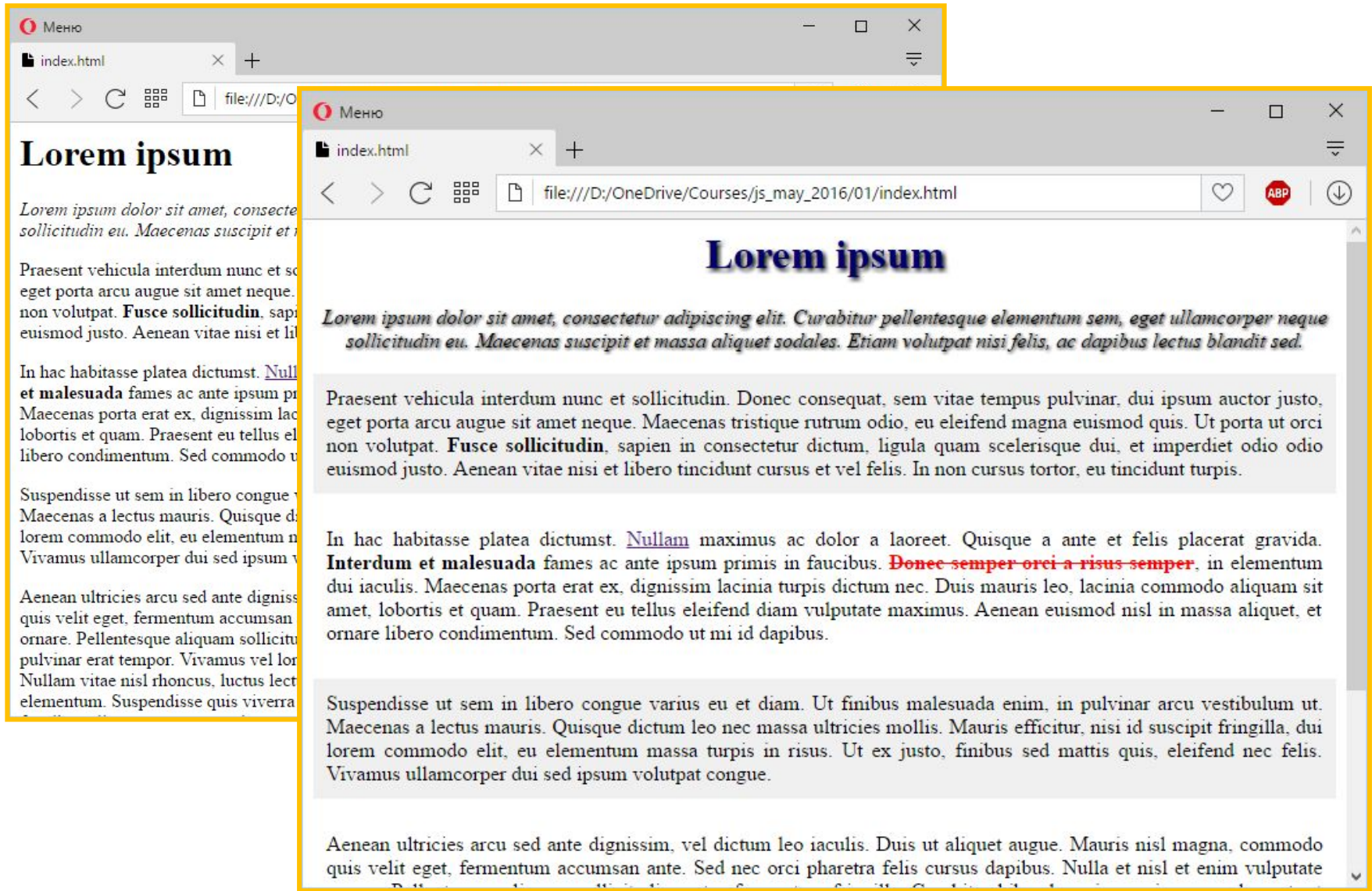
HTML предназначен для задания структуры документа, и выполняет роль «контейнера» для текста.

HTML- ДОКУМЕНТ

```
1 <html>
2   <head>
3     <title>Sample Document</title>
4   </head>
5   <body>
6     <h1>An HTML Document</h1>
7     <p>This is a <i>simple</i> document.</p>
8   </body>
9 </html>
```

*Древовидная структура HTML-
документа*

Цели и задачи языка



Задать оформления элементов страницы
(тегов и их содержимого).

CSS

CSS ([англ.](#) *Cascading Style Sheets* — *каскадные таблицы** стилей) — язык **описания внешнего вида документа**, написанного с использованием HTML.

** таблицы здесь вообще не причём.*

Синтаксис CSS

CSS селектор, говорит к каким элементам будет применяться описываемый стиль (**css selector**).

Значение которое устанавливается для свойства (**value**).

```
div { color: red; font-size: 16pt; }
```

Имя свойства, которое устанавливается (**property**).

CSS отвечает за такие аспекты

- 1. Внешний вид элемента (цвет, шрифт, прозрачность и т.д.);*
- 2. Размеры элемента (высота, ширина, границы, отступы и т.д.);*
- 3. Положение элемента на странице;*
- 4. Спецэффекты, анимация.*

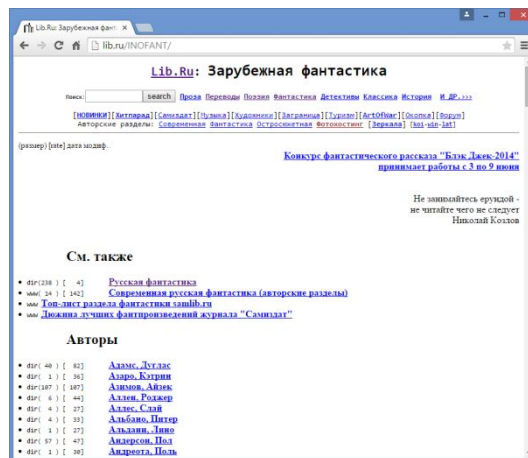
** под элементом, подразумевается тег.*

Есть проблема - HTML

статичен*

*Что неудивительно, ведь HTML не является языком программирования.

Страница, после загрузки в браузер, остаётся неизменной до перезагрузки, или ухода на другую страницу.



Типичный сценарий работы с сайтом без JavaScript: любое действие требует перехода на другой URL и/или полной перезагрузки страницы.

Так зачем же нужен

JavaScript?

JavaScript позволяет «оживить» веб-страницу, чтобы страница могла изменять своё содержание (и внешний вид) без перезагрузки.

Задача JavaScript – Манипуляция HTML-документов.

Цель JavaScript - повысить интерактивность веб-страниц.

Задача JavaScript – манипуляция HTML-документом

- 1. Добавление элементов (тегов) на страницу (в документ);*
- 2. Удаление элементов со страницы;*
- 3. Изменение содержимого элемента и/или его стилевых свойств (стилей CSS);*

Система управления основанная на событиях



Каждая из этих вещей делает что-то, только в ответ на действия пользователя. Можно сказать каждое действие пользователя это событие, и на него нужно как-то отреагировать.

Задача JavaScript – манипуляция HTML-документом

4. Реагирование на действия пользователя;

Суть **интерактивности** – делать что-то в ответ на действия пользователей. В программировании такой подход называют **событийно-ориентированным**, а каждое действие пользователя – **событием**.

Версии JavaScript

JavaScript – реализация языка

ECMAScript

ECMAScript 3

ECMAScript 5

ECMAScript 6 => ECMAScript 2015

ECMAScript 2016

?!?!?!?...

<http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf>

** Стандарт определяет, что входит в сам языке, но не инфраструктуру где он работает. Инфраструктуру определяет стандарт DOM (Document Object Model).*

Версии JavaScript

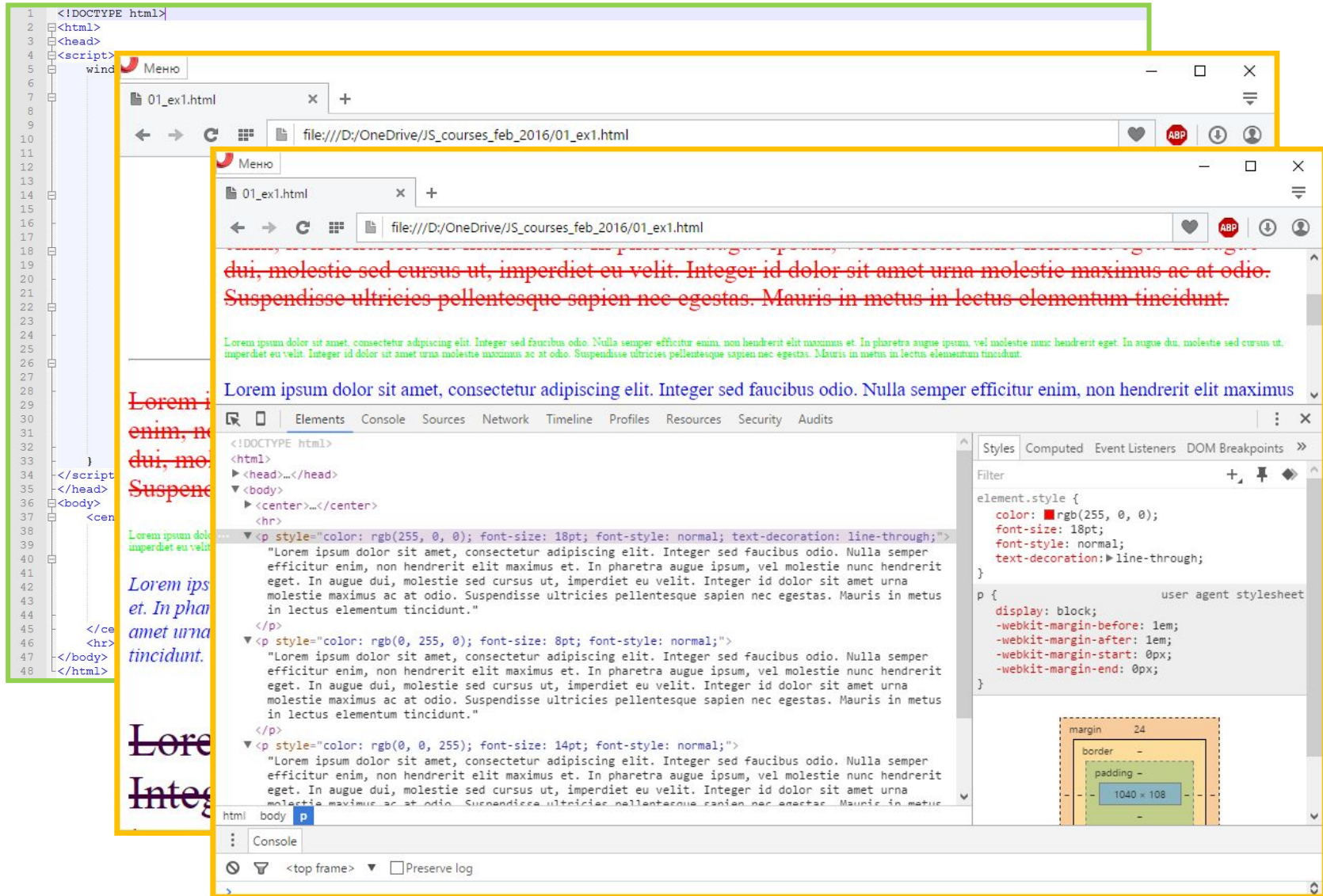
Стандарт впереди планеты всей, однако, поддержка....

| Feature name | Current browser | Compilers/polyfills | | | | | | | | | | Desktop browsers | | | | | | | | | | | | | | | Servers/... | | | | | | | | | |
|--|-----------------|---------------------|--------------------------------|---------|----------------------|----------|-------|------------------------|------------------------|------------------------|-----------|------------------|-------|-------|-------|-------|-----------------------------|-----------------------------|-----------------------------|--------------|--------------|-------|-------|-------|------------------------|------|--------------------------|-----------------------|-----------------------|-------|-------|-------|------|------|-----|-----|
| | | Tracour | Babel + core-js ^[1] | Closure | TypeScript + core-js | es6-shim | IE 11 | Edge 12 ^[3] | Edge 13 ^[3] | Edge 14 ^[3] | FF 38 ESR | FF 45 ESR | FF 46 | FF 47 | FF 48 | FF 49 | CH 50, OP 37 ^[0] | CH 51, OP 38 ^[0] | CH 52, OP 39 ^[0] | SF 6.1, SF 7 | SF 7.1, SF 8 | SF 9 | SFTP | WK | KQ 4.14 ^[4] | PJS | Node 0.12 ^[5] | Node 4 ^[6] | Node 5 ^[5] | | | | | | | |
| Optimisation | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| proper tail calls (tail call optimisation) | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 |
| Syntax | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| default function parameters | 7/7 | 4/7 | 4/7 | 4/7 | 5/7 | 0/7 | 0/7 | 0/7 | 0/7 | 7/7 | 3/7 | 4/7 | 4/7 | 4/7 | 4/7 | 4/7 | 7/7 | 7/7 | 7/7 | 0/7 | 0/7 | 0/7 | 7/7 | 7/7 | 0/7 | 0/7 | 0/7 | 0/7 | 0/7 | 0/7 | 0/7 | 0/7 | 0/7 | 0/7 | 0/7 | 0/7 |
| rest parameters | 5/5 | 4/5 | 3/5 | 2/5 | 4/5 | 0/5 | 0/5 | 0/5 | 5/5 | 5/5 | 4/5 | 5/5 | 5/5 | 5/5 | 5/5 | 5/5 | 5/5 | 5/5 | 5/5 | 0/5 | 0/5 | 0/5 | 5/5 | 5/5 | 0/5 | 0/5 | 0/5 | 0/5 | 0/5 | 0/5 | 0/5 | 0/5 | 0/5 | 0/5 | 0/5 | |
| spread (...) operator | 15/15 | 15/15 | 13/15 | 12/15 | 4/15 | 0/15 | 0/15 | 12/15 | 15/15 | 15/15 | 15/15 | 15/15 | 15/15 | 15/15 | 15/15 | 15/15 | 15/15 | 15/15 | 15/15 | 0/15 | 5/15 | 9/15 | 15/15 | 15/15 | 0/15 | 0/15 | 0/15 | 0/15 | 0/15 | 0/15 | 0/15 | 0/15 | 0/15 | 0/15 | | |
| object literal extensions | 6/6 | 6/6 | 6/6 | 4/6 | 6/6 | 0/6 | 0/6 | 6/6 | 6/6 | 6/6 | 6/6 | 6/6 | 6/6 | 6/6 | 6/6 | 6/6 | 6/6 | 6/6 | 6/6 | 0/6 | 1/6 | 5/6 | 6/6 | 6/6 | 0/6 | 0/6 | 0/6 | 0/6 | 6/6 | 6/6 | 6/6 | 6/6 | 6/6 | 6/6 | | |
| for...of loops | 7/9 | 9/9 | 9/9 | 6/9 | 3/9 | 0/9 | 0/9 | 6/9 | 7/9 | 7/9 | 7/9 | 7/9 | 7/9 | 7/9 | 7/9 | 7/9 | 7/9 | 7/9 | 7/9 | 0/9 | 2/9 | 8/9 | 9/9 | 9/9 | 0/9 | 0/9 | 7/9 | 7/9 | 7/9 | 7/9 | 7/9 | 7/9 | 7/9 | 7/9 | | |
| octal and binary literals | 4/4 | 2/4 | 4/4 | 4/4 | 4/4 | 2/4 | 0/4 | 4/4 | 4/4 | 4/4 | 4/4 | 4/4 | 4/4 | 4/4 | 4/4 | 4/4 | 4/4 | 4/4 | 4/4 | 0/4 | 0/4 | 4/4 | 4/4 | 4/4 | 0/4 | 0/4 | 0/4 | 0/4 | 4/4 | 4/4 | 4/4 | 4/4 | 4/4 | 4/4 | | |
| template literals | 5/5 | 4/5 | 4/5 | 3/5 | 3/5 | 0/5 | 0/5 | 4/5 | 5/5 | 5/5 | 5/5 | 5/5 | 5/5 | 5/5 | 5/5 | 5/5 | 5/5 | 5/5 | 5/5 | 0/5 | 0/5 | 5/5 | 5/5 | 5/5 | 0/5 | 0/5 | 0/5 | 0/5 | 0/5 | 0/5 | 0/5 | 0/5 | 0/5 | 0/5 | | |
| RegExp "v" and "u" flags | 5/5 | 3/5 | 3/5 | 0/5 | 0/5 | 0/5 | 0/5 | 2/5 | 4/5 | 4/5 | 2/5 | 2/5 | 5/5 | 5/5 | 5/5 | 5/5 | 5/5 | 5/5 | 0/5 | 0/5 | 0/5 | 5/5 | 5/5 | 0/5 | 0/5 | 0/5 | 0/5 | 0/5 | 0/5 | 0/5 | 0/5 | 0/5 | 0/5 | 0/5 | | |
| destructuring declarations | 21/22 | 20/22 | 21/22 | 18/22 | 15/22 | 0/22 | 0/22 | 0/22 | 0/22 | 21/22 | 19/22 | 19/22 | 19/22 | 21/22 | 21/22 | 21/22 | 21/22 | 22/22 | 0/22 | 9/22 | 19/22 | 22/22 | 22/22 | 0/22 | 0/22 | 0/22 | 0/22 | 0/22 | 0/22 | 0/22 | 0/22 | 0/22 | 0/22 | 0/22 | | |
| destructuring assignment | 23/24 | 23/24 | 24/24 | 15/24 | 19/24 | 0/24 | 0/24 | 0/24 | 0/24 | 23/24 | 20/24 | 21/24 | 21/24 | 23/24 | 23/24 | 23/24 | 23/24 | 23/24 | 24/24 | 0/24 | 12/24 | 21/24 | 24/24 | 24/24 | 0/24 | 0/24 | 0/24 | 0/24 | 0/24 | 0/24 | 0/24 | 0/24 | 0/24 | 0/24 | | |
| destructuring parameters | 22/23 | 19/23 | 20/23 | 17/23 | 15/23 | 0/23 | 0/23 | 0/23 | 0/23 | 22/23 | 18/23 | 18/23 | 18/23 | 19/23 | 19/23 | 19/23 | 22/23 | 22/23 | 23/23 | 0/23 | 10/23 | 18/23 | 23/23 | 23/23 | 0/23 | 0/23 | 0/23 | 0/23 | 0/23 | 0/23 | 0/23 | 0/23 | 0/23 | 0/23 | | |
| Unicode code point escapes | 2/2 | 1/2 | 1/2 | 1/2 | 1/2 | 0/2 | 0/2 | 2/2 | 2/2 | 2/2 | 0/2 | 1/2 | 1/2 | 1/2 | 1/2 | 1/2 | 2/2 | 2/2 | 2/2 | 0/2 | 0/2 | 2/2 | 2/2 | 2/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | | |
| new.target | 2/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 1/2 | 2/2 | 0/2 | 2/2 | 2/2 | 2/2 | 2/2 | 2/2 | 2/2 | 2/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | 0/2 | | |
| Bindings | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| const | 16/16 | 14/16 | 14/16 | 14/16 | 14/16 | 0/16 | 12/16 | 12/16 | 12/16 | 16/16 | 10/16 | 12/16 | 12/16 | 12/16 | 12/16 | 12/16 | 16/16 | 16/16 | 16/16 | 1/16 | 1/16 | 1/16 | 16/16 | 16/16 | 2/16 | 1/16 | 1/16 | 8/16 | 8/16 | 8/16 | 8/16 | 8/16 | | | | |
| let | 12/12 | 10/12 | 10/12 | 10/12 | 10/12 | 0/12 | 10/12 | 10/12 | 10/12 | 12/12 | 0/12 | 10/12 | 10/12 | 10/12 | 10/12 | 10/12 | 12/12 | 12/12 | 12/12 | 0/12 | 0/12 | 0/12 | 12/12 | 12/12 | 0/12 | 0/12 | 0/12 | 6/12 | 6/12 | 6/12 | 6/12 | 6/12 | | | | |
| block-level function declaration ^[12] | Yes | Yes | Yes | Yes | No | No | Yes | Yes | Yes | Yes | No | No | Yes | Yes | Yes | Yes | Yes | Yes | No | No | No | Yes | Yes | No | No | No | Flag | Yes | Yes | Yes | Yes | Yes | Yes | | | |
| Functions | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| arrow functions | 13/13 | 11/13 | 9/13 | 11/13 | 9/13 | 0/13 | 0/13 | 8/13 | 12/13 | 13/13 | 8/13 | 13/13 | 13/13 | 13/13 | 13/13 | 13/13 | 13/13 | 13/13 | 0/13 | 0/13 | 0/13 | 13/13 | 13/13 | 0/13 | 0/13 | 0/13 | 9/13 | 10/13 | 10/13 | 10/13 | 10/13 | 10/13 | | | | |
| class | 24/24 | 17/24 | 19/24 | 9/24 | 19/24 | 0/24 | 0/24 | 0/24 | 24/24 | 24/24 | 0/24 | 24/24 | 24/24 | 24/24 | 24/24 | 24/24 | 24/24 | 24/24 | 0/24 | 0/24 | 16/24 | 24/24 | 24/24 | 0/24 | 0/24 | 0/24 | 0/24 | 0/24 | 0/24 | 0/24 | 0/24 | 0/24 | 0/24 | | | |
| super | 8/8 | 7/8 | 4/8 | 4/8 | 7/8 | 0/8 | 0/8 | 0/8 | 8/8 | 8/8 | 0/8 | 8/8 | 8/8 | 8/8 | 8/8 | 8/8 | 8/8 | 8/8 | 0/8 | 0/8 | 6/8 | 8/8 | 8/8 | 0/8 | 0/8 | 0/8 | 0/8 | 0/8 | 0/8 | 0/8 | 0/8 | 0/8 | | | | |
| generators | 27/27 | 24/27 | 24/27 | 16/27 | 0/27 | 0/27 | 0/27 | 0/27 | 27/27 | 27/27 | 20/27 | 25/27 | 25/27 | 25/27 | 25/27 | 25/27 | 27/27 | 27/27 | 27/27 | 0/27 | 0/27 | 0/27 | 27/27 | 27/27 | 0/27 | 0/27 | 0/27 | 20/27 | 20/27 | 20/27 | 20/27 | 20/27 | | | | |

<http://kangax.github.io/compat-table/es6/>

На практике

Как всё вышеописанное

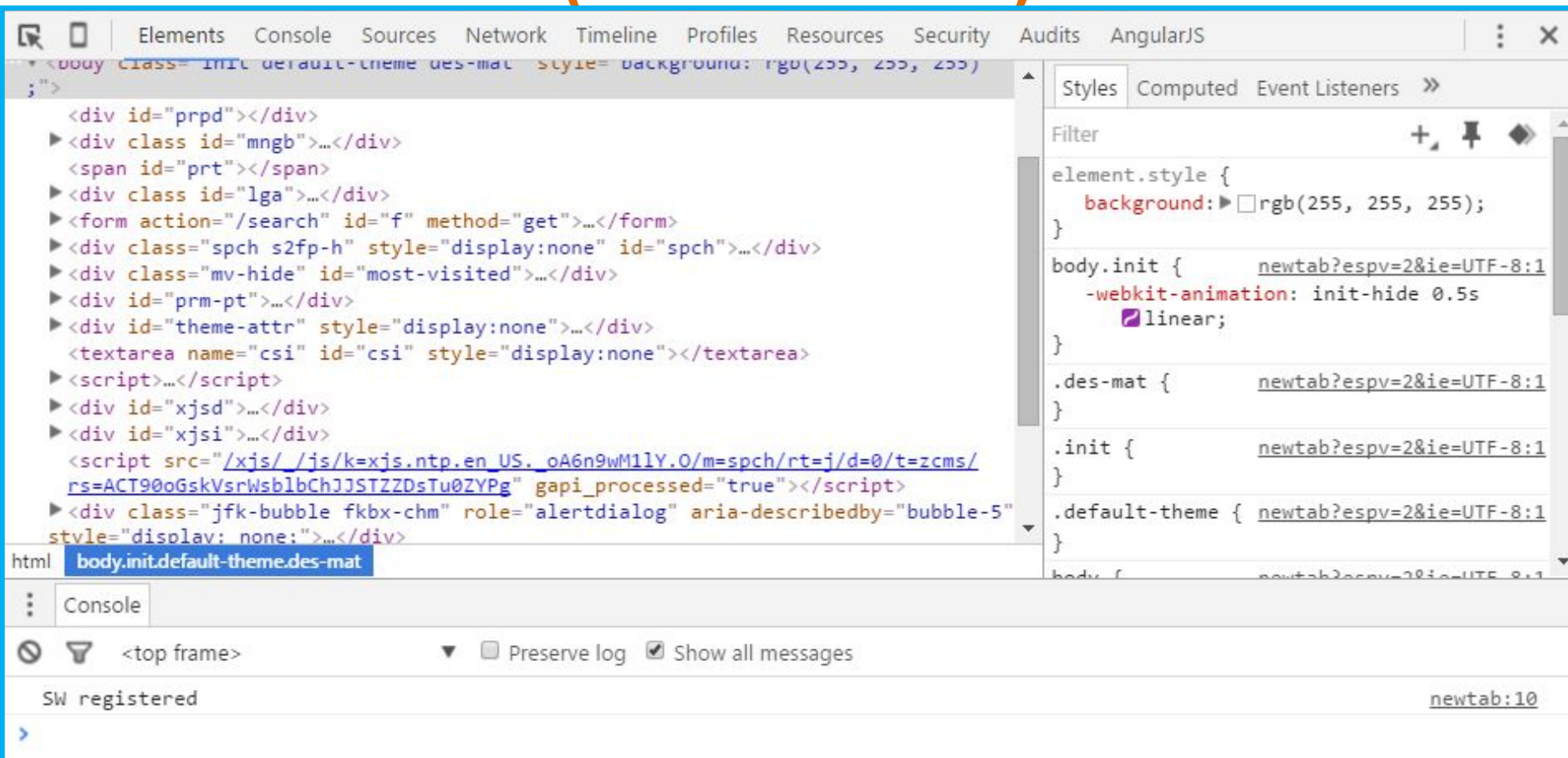


Первый пример: <http://courses.dp.ua/files/js/01/ex01.html>

Инструменты

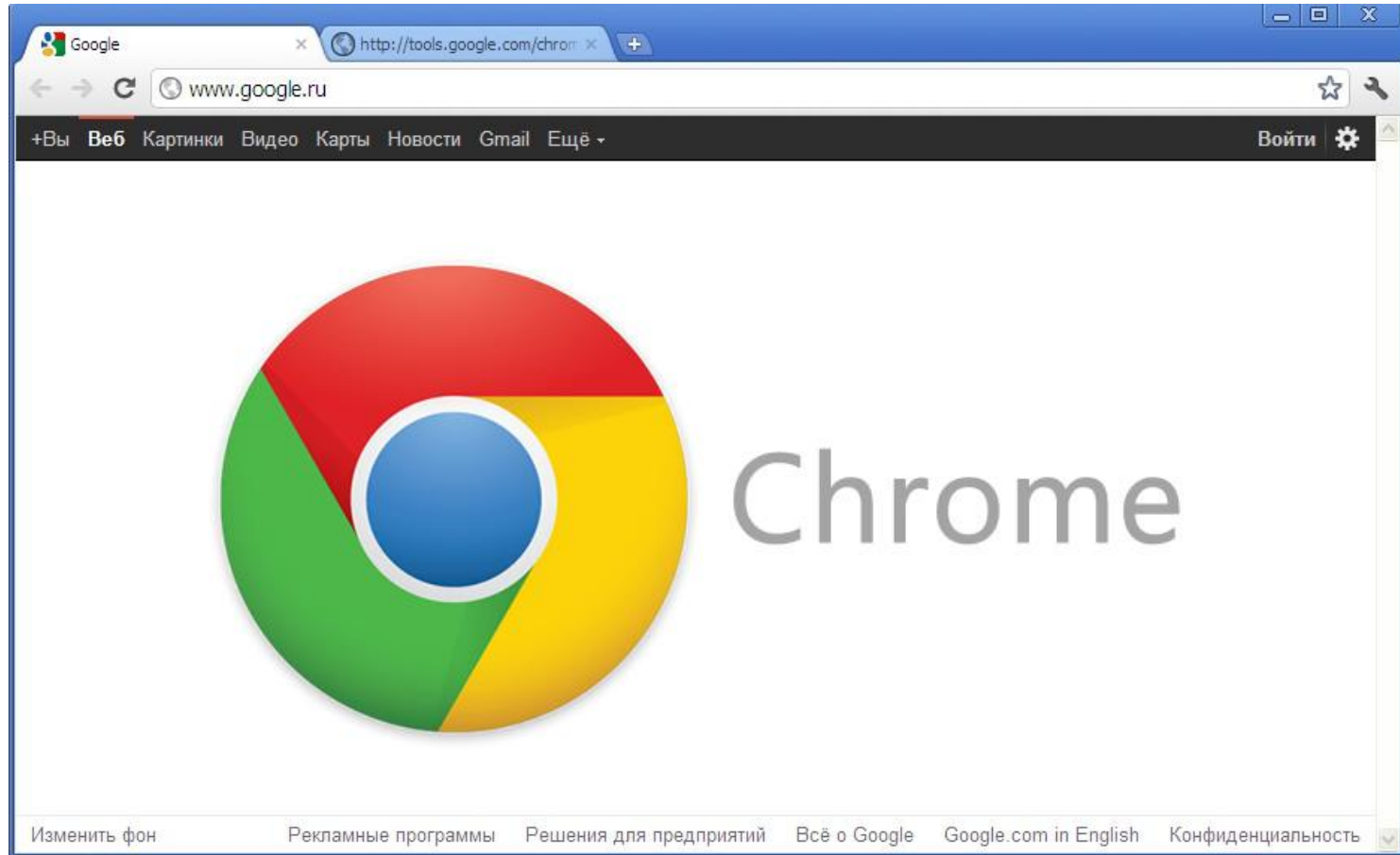
Которые нам пригодятся

Инструменты: Консоль разработчика в браузере (клавиша F12)



Подробная информация о том, как браузер «понимает» созданный вами код.

Инструменты: Браузер Chrome



***Н.В.** JavaScript выполняется браузерами, поэтому поддержка тех или иных функций JavaScript зависит от того сделали её поддержку разработчики браузера или нет.*

Инструменты: <http://google.com>

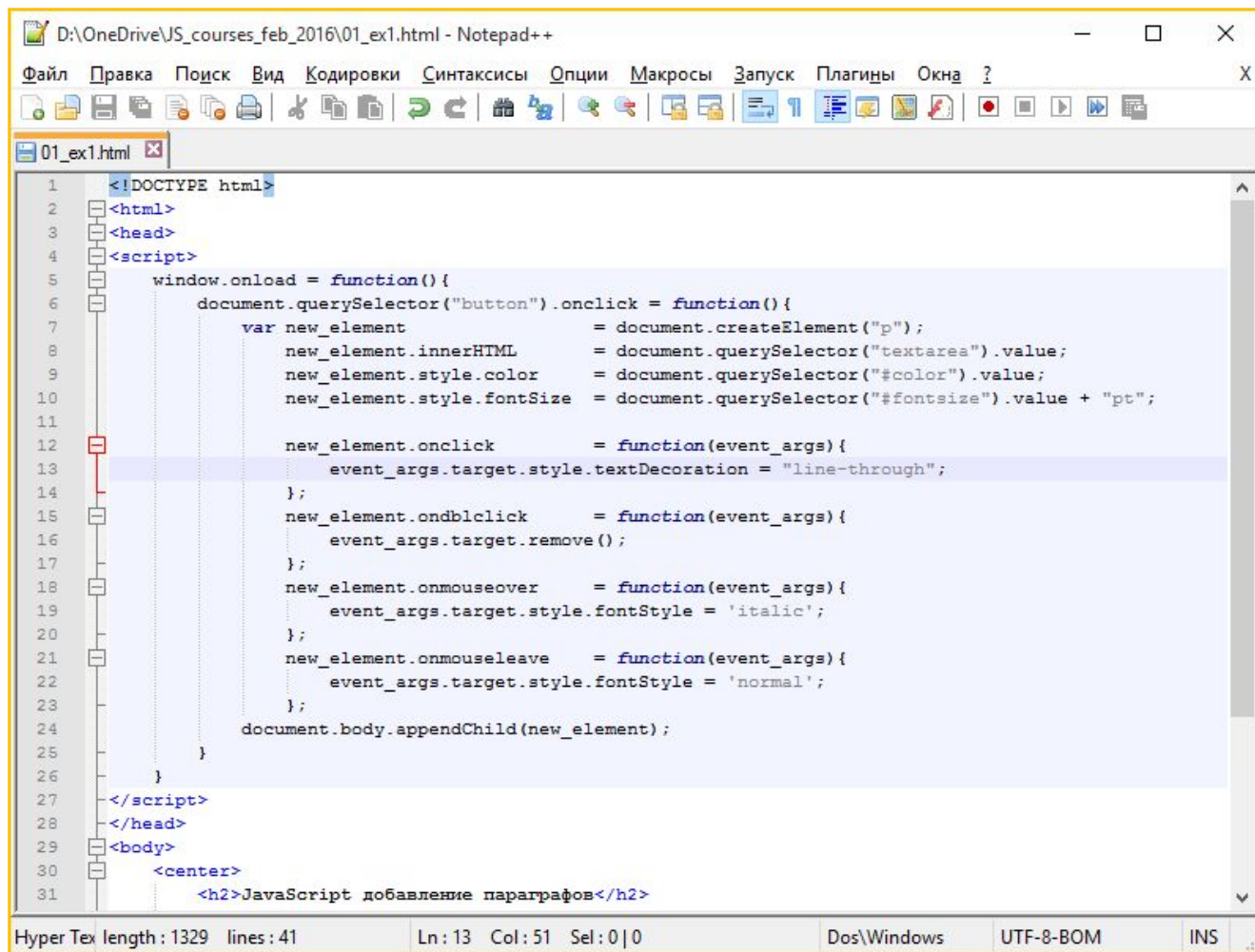
A large, empty search input field with a thin grey border. On the right side of the field, there are two small icons: a keyboard icon and a microphone icon.

Поиск в Google

Мне повезёт!

Google.com.ua предлагается на: [українська](#)

Инструменты: Notepad++



The screenshot shows the Notepad++ interface with a file named '01_ex1.html'. The code is as follows:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <script>
5     window.onload = function(){
6         document.querySelector("button").onclick = function(){
7             var new_element
8                 = document.createElement("p");
9             new_element.innerHTML
10                = document.querySelector("textarea").value;
11             new_element.style.color
12                = document.querySelector("#color").value;
13             new_element.style.fontSize
14                = document.querySelector("#fontsize").value + "pt";
15
16             new_element.onclick
17                = function(event_args){
18                 event_args.target.style.textDecoration = "line-through";
19             };
20             new_element.ondblclick
21                = function(event_args){
22                 event_args.target.remove();
23             };
24             new_element.onmouseover
25                = function(event_args){
26                 event_args.target.style.fontStyle = 'italic';
27             };
28             new_element.onmouseleave
29                = function(event_args){
30                 event_args.target.style.fontStyle = 'normal';
31             };
32             document.body.appendChild(new_element);
33         }
34     }
35 </script>
36 </head>
37 <body>
38     <center>
39         <h2>JavaScript добавление параграфов</h2>
```

The status bar at the bottom indicates: Hyper Text length: 1329 lines: 41 Ln: 13 Col: 51 Sel: 0|0. The system tray shows 'Dos\Windows', 'UTF-8-BOM', and 'INS'.

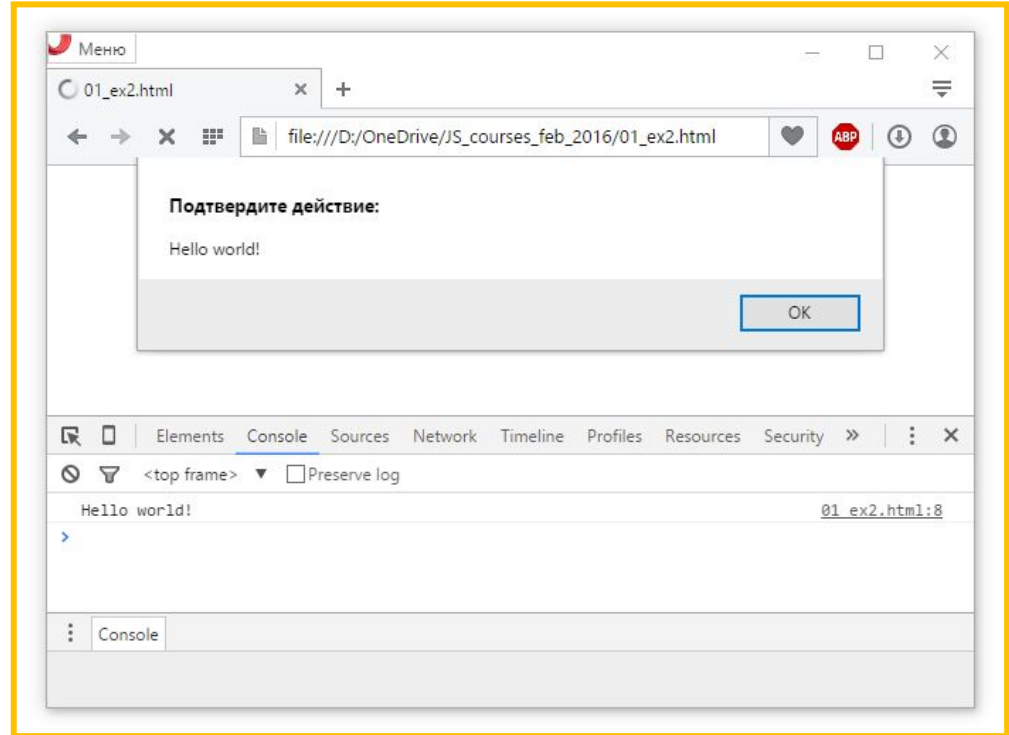
Инструменты: «допустимый» синтаксис

```
4 <script>  
5  
6     var message = "Text, text, text.";  
7  
8     console.log(message);  
9  
10 </script>
```

Инструменты: служебные функции

```
4 <script>
5
6   var message = "Hello world!";
7
8   console.log(message);
9
10  alert(message);
11
12 </script>
```

console.log(...)
;
alert(...)
;

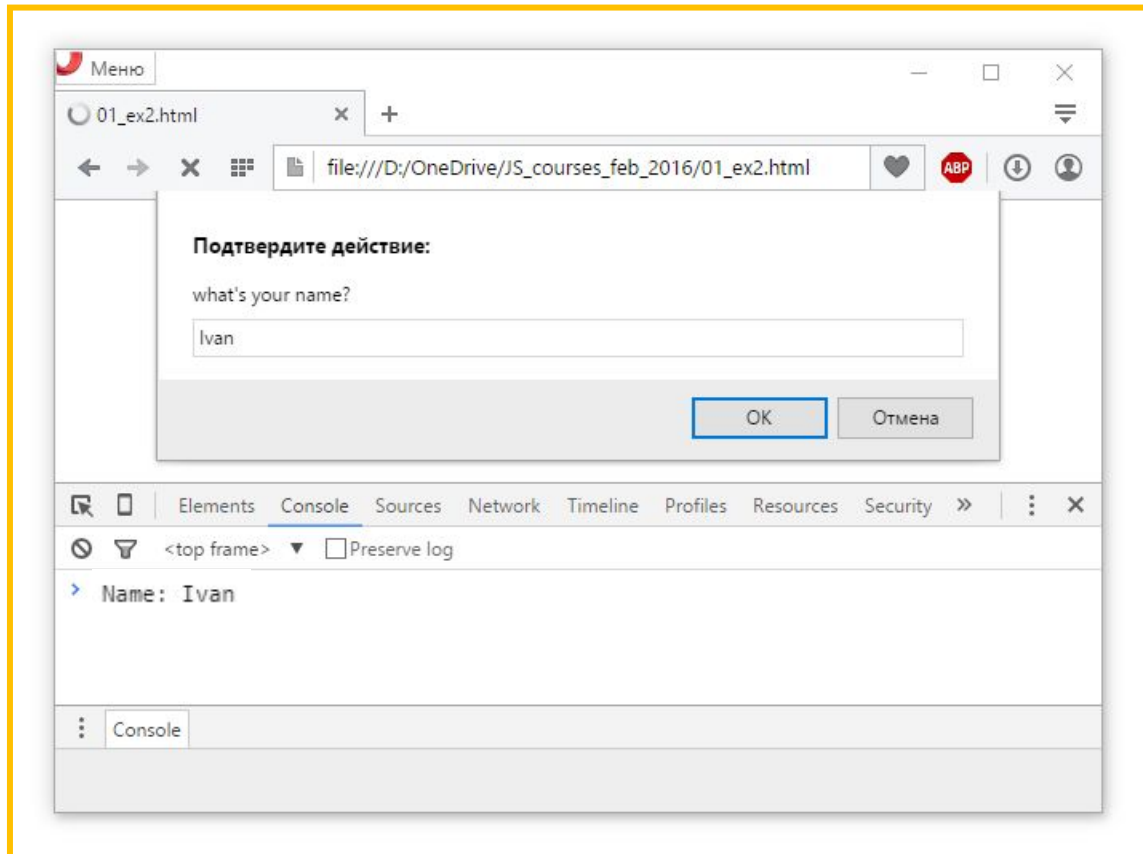


Создаём новый файл в *notepad++*, сохраняем файл как *test.html*, и запускаем (**Ctrl+Shift+Alt+R**).

Инструменты: служебные функции

```
4 <script>
5
6     var result = prompt("what's your name?");
7
8     console.log("Name: " + result);
9
10 </script>
```

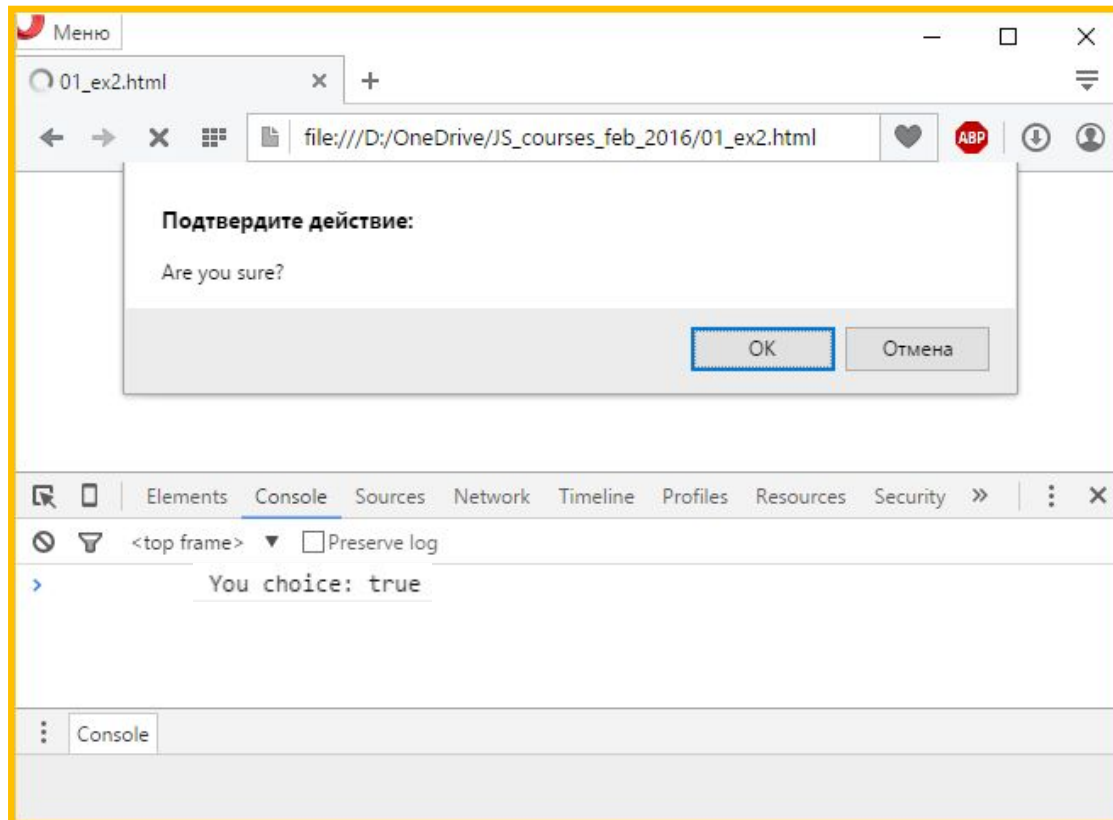
prompt(...)
;



Инструменты: служебные функции

```
4 <script>
5
6     var result = confirm("Are you sure?");
7
8     console.log("You choice: " + result);
9
10 </script>
```

confirm(...)
;



Hello world!

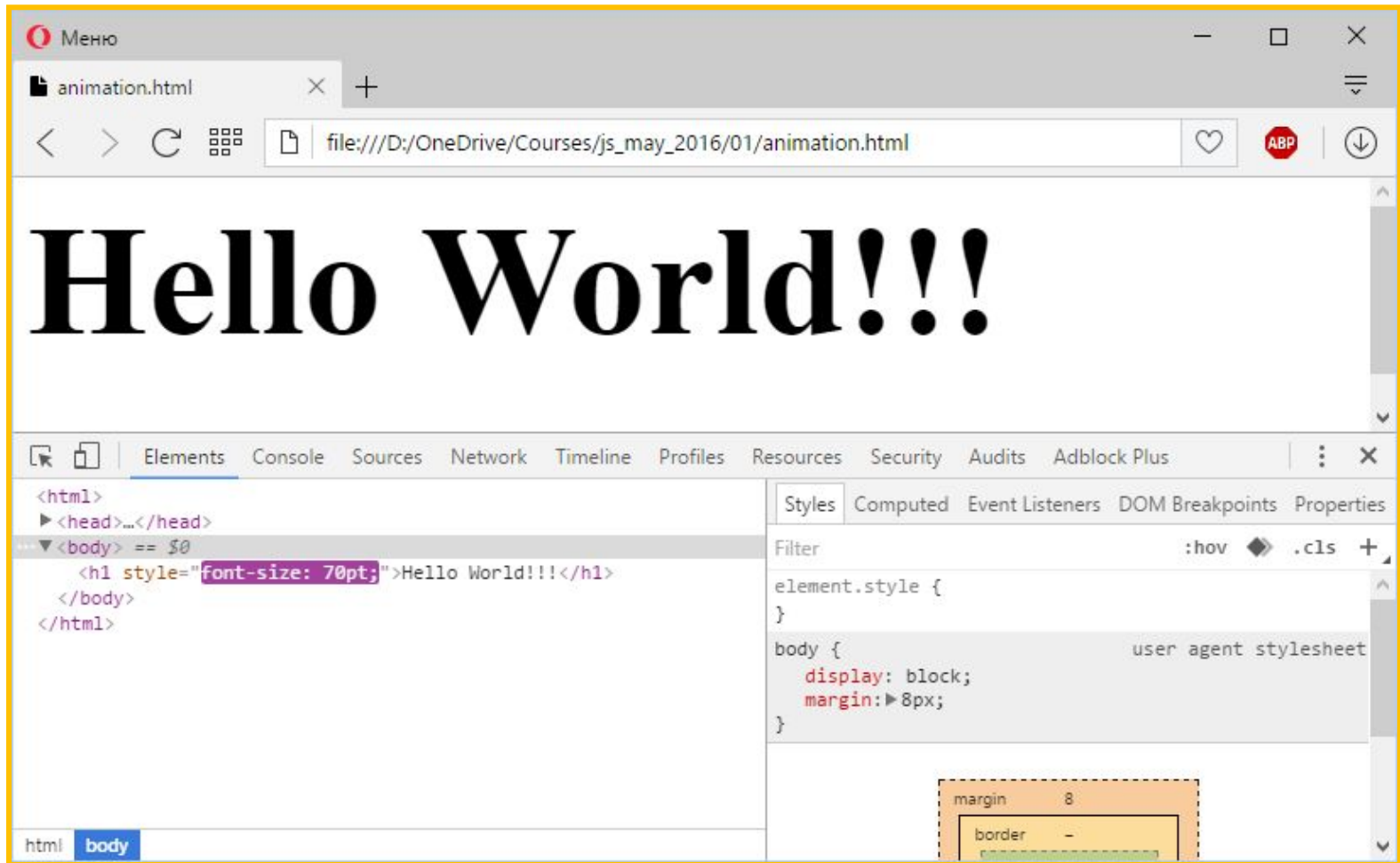
Давайте попробуем или “Hello World!!!”

```
1 <script>
2   document.write("<h1>Hello World!!!</h1>");
3   var tag = document.querySelector("h1");
4
5   var size = 6;
6   var step = 1;
7
8   function beat() {
9     requestAnimationFrame(beat);
10    tag.style.fontSize = size + "pt";
11    size = size + step;
12
13    if(size > 70 || size < 6) {
14      step = -step;
15    }
16  }
17
18  beat();
19
20 </script>
```



Создаём новый файл в *notepad++*, сохраняем файл как *index.html*, и запускаем (**Ctrl+Shift+Alt+R**).

Давайте попробуем или “Hello World!!!”



*В результате
получаем...*

Node.JS

JavaScript не только в браузере: Node.JS

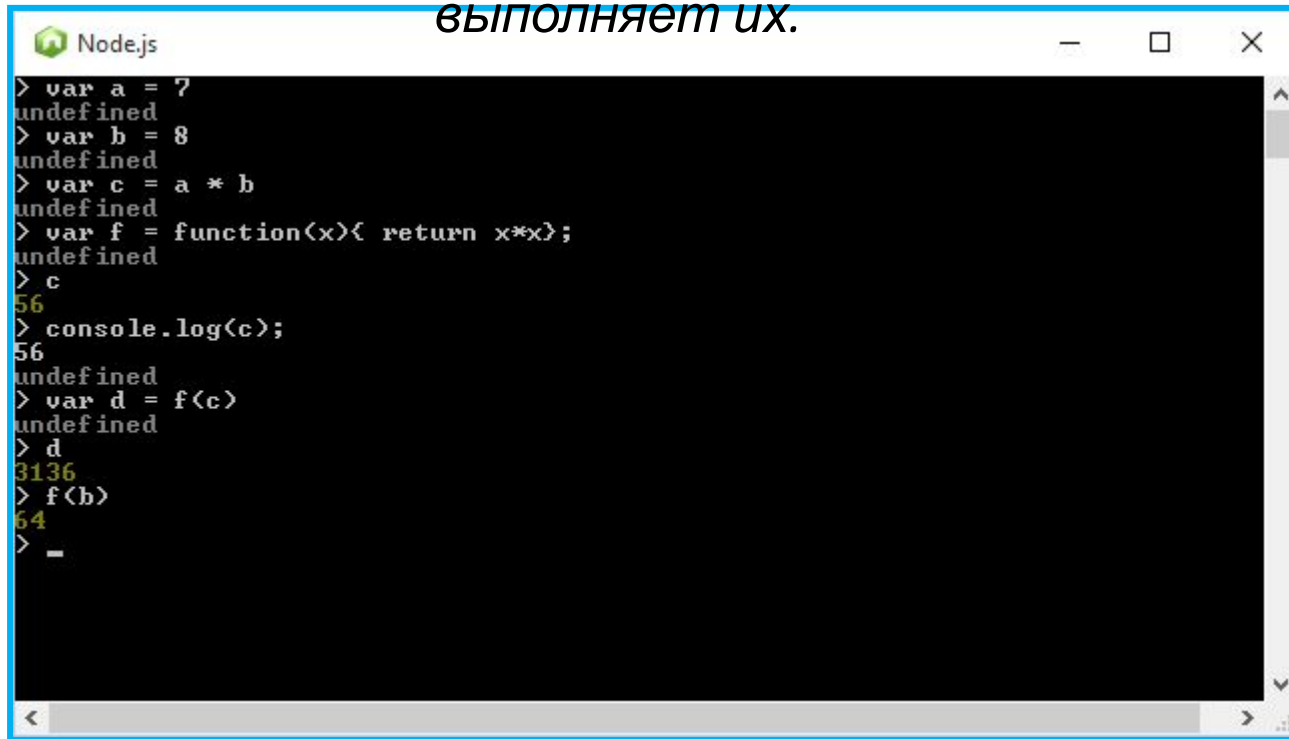


<https://nodejs.org/en/>

NodeJS – программа, которая ставиться на компьютер и позволяет выполнять код на языке **JavaScript** прямо на вашем компьютере (с полным доступом к компьютеру, без ограничений браузера, и без самого браузера). Т.е. **NodeJS** ставит **JavaScript** в один ряд с C#, Java, Ruby, Python и прочими.

Интерпретатор JavaScript-

кода
В составе Node.js входит программа-интерпретатор,
которая разбирает текстовые файлы с кодом и
выполняет их.



```
Node.js
> var a = 7
undefined
> var b = 8
undefined
> var c = a * b
undefined
> var f = function(x){ return x*x};
undefined
> c
56
> console.log(c);
56
undefined
> var d = f(c)
undefined
> d
3136
> f(b)
64
> -
```

Позволяет проверить JavaScript код, и сразу
получить результат его работы
построчно.

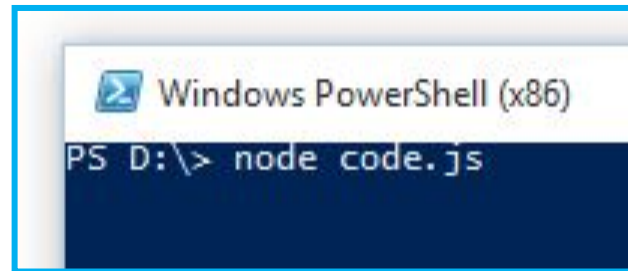
Интерпретатор JavaScript-

Создайте файл **code.js**

```
1  var http = require('http');  
2  
3  function onRequest(request, response){  
4      response.end("<h1>Hi from Node.JS Server!!!</h1>");  
5  }  
6  
7  var server = http.createServer(onRequest);  
8  server.listen(80);
```

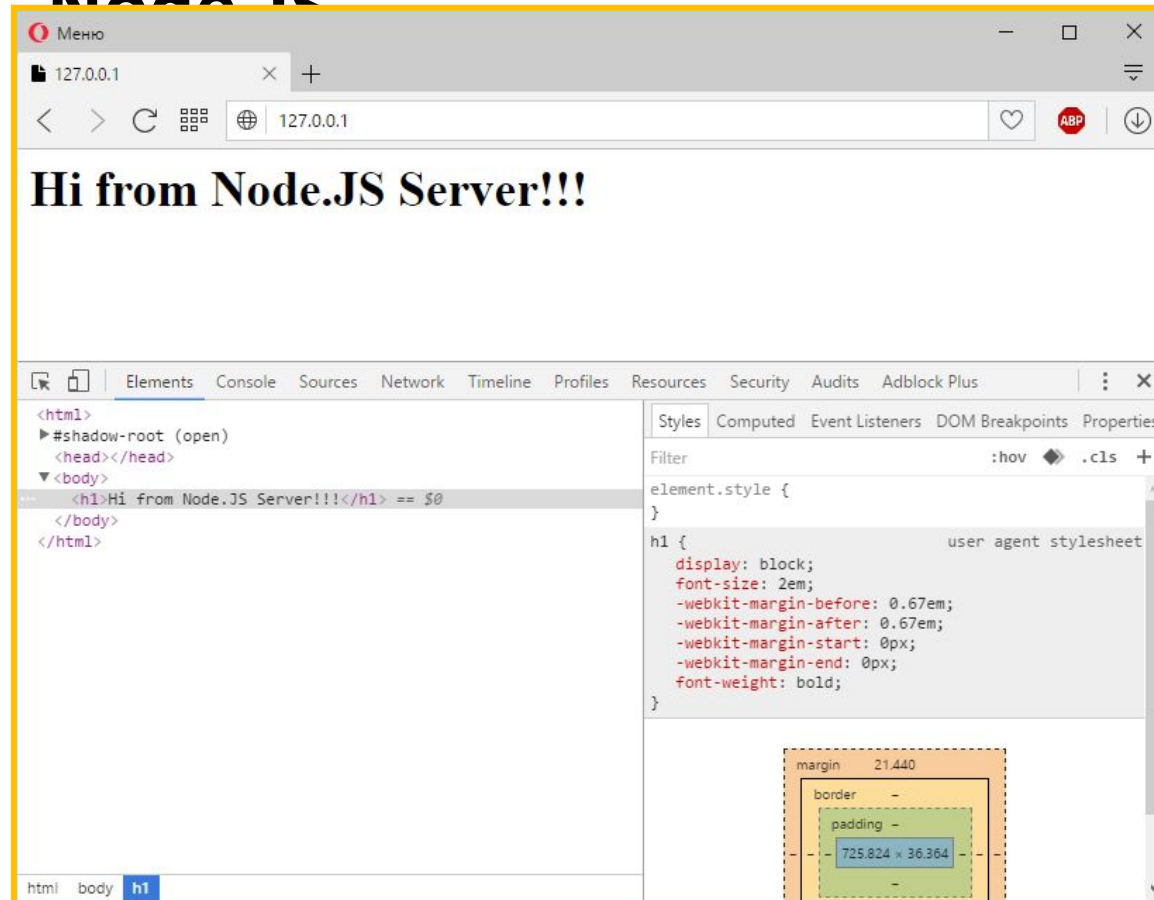


Выполните в консоли команду **node code.js** (вместо **code.js** может быть путь к любому js-файлу).



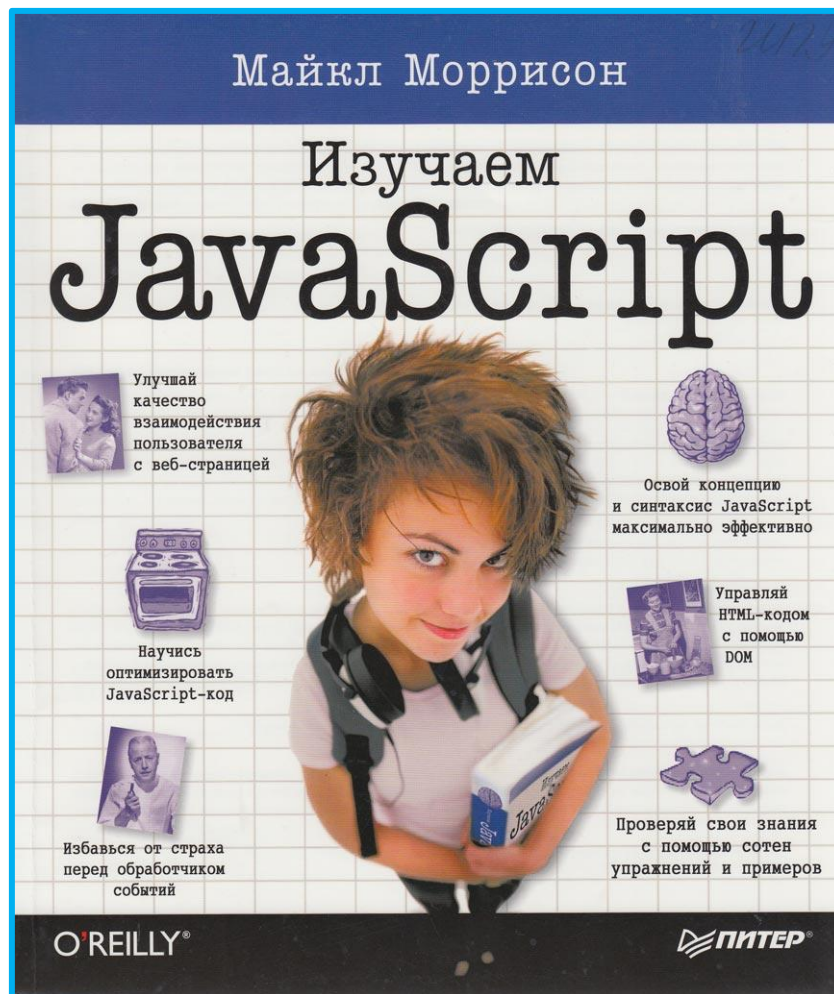
Консольная команда **node code.js** позволяет запустить на выполнение файл с JS-кодом целиком. В данном случае будет запущен http-сервер.

http-сервер на базе Node JS



Работоспособность запущенного http-сервера легко проверить запустив браузер и открыв адрес <http://localhost> или <http://127.0.0.1>

Майкл Моррисон «Изучаем JavaScript»



JavaScript.ru

The screenshot displays the homepage of the JavaScript.ru website. At the top, the title 'Современный учебник JavaScript' is prominently featured. Below it, a short introductory paragraph describes the book's scope. A search bar with the placeholder text 'Искать в учебнике' and a green 'Найти' button is positioned next. The main content area is titled 'Основной курс' and includes a 'Карта учебника' (Table of Contents) icon. It is divided into two columns: 'Часть первая: Язык JavaScript' and 'Часть вторая: Документ, события, интерфейсы'. The first column lists various topics such as 'Введение', 'Основы JavaScript', and 'Современные возможности ES-2015'. The second column lists 'Документ и объекты страницы', 'Основы работы с событиями', and 'Создание графических компонентов'.

Современный учебник JavaScript

Перед вами учебник по JavaScript, начиная с основ, включающий в себя много тонкостей и фишек JavaScript/DOM.

Найти

Основной курс

[Карта учебника](#)

Часть первая

Язык JavaScript

Эта часть позволит вам изучить JavaScript с нуля или упорядочить и дополнить существующие знания.

Мы будем использовать браузер в качестве окружения, но основное внимание будет уделяться именно самому языку JavaScript.

- Введение
- Основы JavaScript
- Качество кода
- Структуры данных
- Замыкания, область видимости
- Методы объектов и контекст вызова
- Некоторые другие возможности
- ООП в функциональном стиле
- ООП в прототипном стиле
- Современные возможности ES-2015

Часть вторая

Документ, события, интерфейсы

Изучаем работу со страницей – как получать элементы, манипулировать их размерами, динамически создавать интерфейсы и взаимодействовать с посетителем.

- Документ и объекты страницы
- Основы работы с событиями
- События в деталях
- Формы, элементы управления
- Создание графических компонентов

Современный учебник
JavaScript
<http://learn.javascript.ru/>

Не забываем

Презентация доступна по
адресу:

courses.dp.ua

Ваша группа:

[JS4 vk.com/js.courses](https://vk.com/js4)

*Группа для вопросов, обсуждений, объявлений
(и презентации там тоже будут).*