

Из чего состоит сайт

Основные компоненты сайта

Дизайн

- Визуальное представление сайта
- Расположение контента на странице
- Описание взаимодействия пользователя с сайтом

Верстка

- Написание кода HTML и CSS по дизайн-макету
- Кроссбраузерность
- Адаптивность под устройства разных размеров

Программирование

- Визуальные эффекты
- Логика взаимодействия пользователя с элементами
- Вычисления

Основные компоненты сайта

Серверная часть

- Вычисления
- Ответ на запросы клиентской части

Контент

- Содержимое сайта (текст, картинки, видео и др. файлы)
- Наполнение веб-страницы смыслом

Хостинг и домен

- Доменное имя - уникальный адрес в сети
- Хостинг - место хранения сайта

CMS

- Система управления содержимым
- Управление всеми элементами ресурса через административную часть

Работа с браузером

Браузер

Программа, основное предназначение которой - воспроизводить контент с веб-ресурсов



Основные компоненты браузера

- Пользовательский интерфейс
- Движок браузера
- Движок отображения
- Сетевая подсистема
- Исполнительная часть пользовательского интерфейса
- Интерпретатор JavaScript
- Хранилище данных

Механизм браузера

- Адресная строка для ввода URI (унифицированного идентификатора ресурсов)
- Элементы навигации (кнопки “вперед”, “назад”)
- Меню закладок
- Настройки
- Другие элементы браузера, кроме окна, в котором отображается запрашиваемая страница

Движок браузера

- управляет взаимодействием интерфейса и модуля отображения

Движок отображения

- отвечает за отображение информации на экране (обрабатывает HTML и CSS и выводит то, что получилось, на экран)

Сетевая подсистема

- отвечает за запросы по сети (например, HTTP-запросы)

Исполнительная часть пользовательского интерфейса

- отвечает за отрисовку базовых компонентов интерфейса, таких, как окна и элементы управления (например, чекбоксов)
- применяет методы пользовательского интерфейса конкретной операционной системы

Интерпретатор JavaScript

- используется для синтаксического анализа и выполнения кода JavaScript

Хранилище данных

- браузер сохраняет на жесткий диск данные различных типов, например файлы cookie, кэш

Инструменты разработчика (devtools)

Инструменты разработчика используются для отладки кода, выявления и исправления ошибок в коде

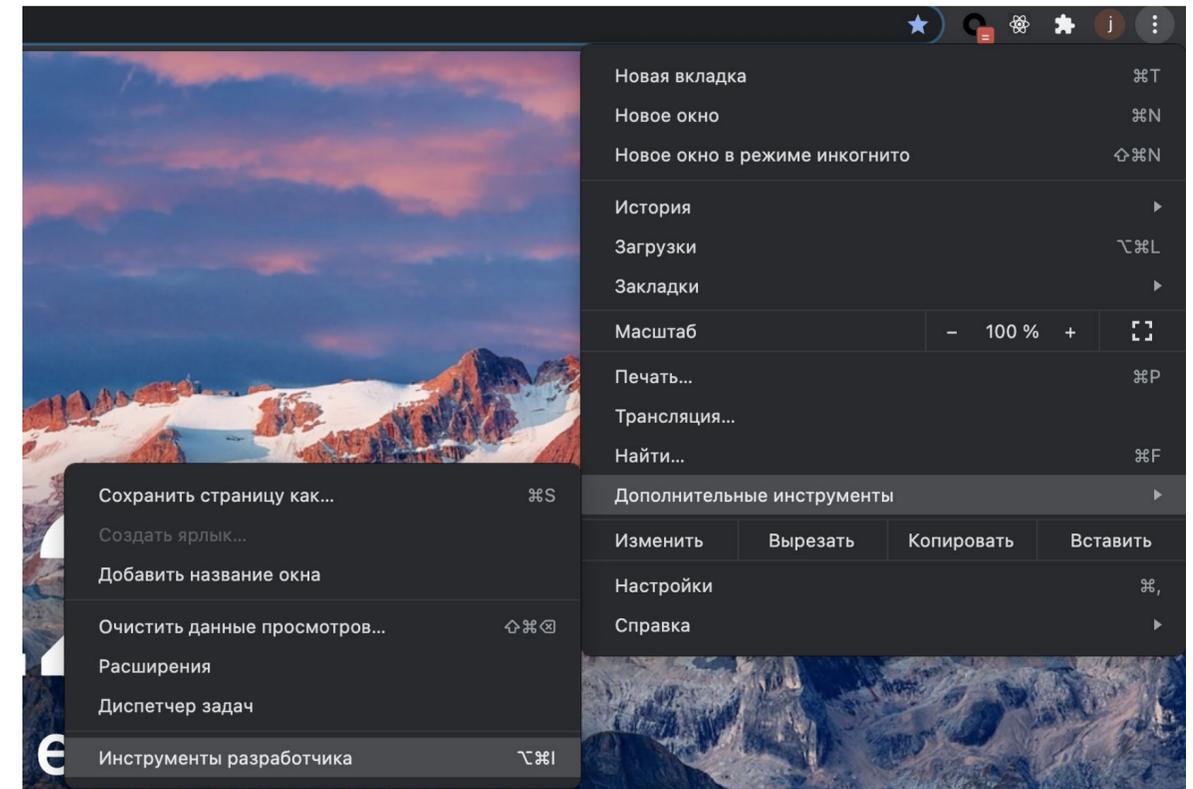
Самые удобные инструменты разработчика - у браузеров Chrome и Firefox



Открыть devtools можно с помощью

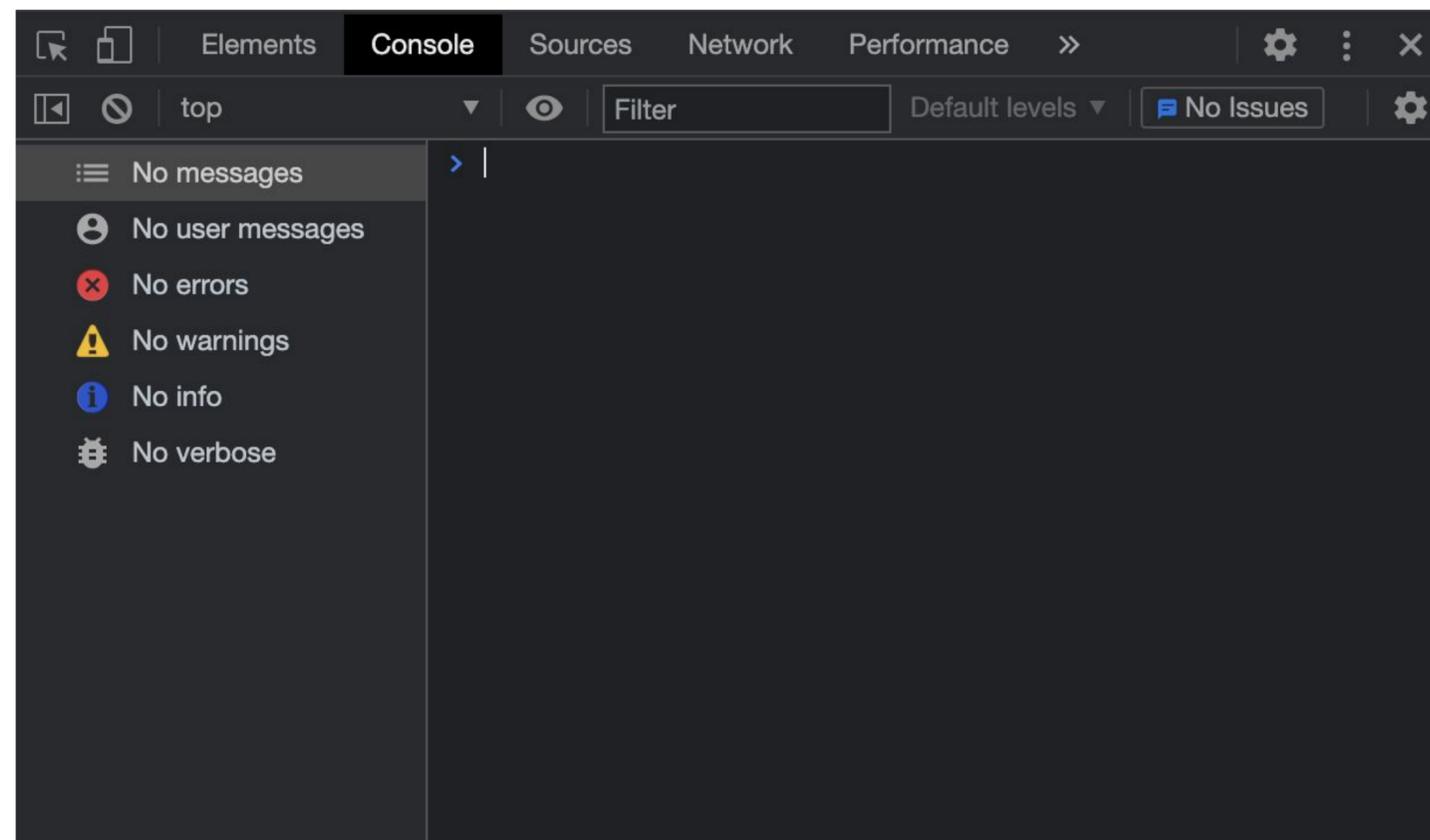
- меню (справа показан пример в браузере Chrome)
- F12 (работает во многих браузерах), на macOS сочетанием `Cmd + Opt + J`

Далее будем рассматривать инструменты разработчика в браузере Chrome



Инструменты разработчика (devtools)

При открытии инструментов разработчика по умолчанию откроется вкладка Console

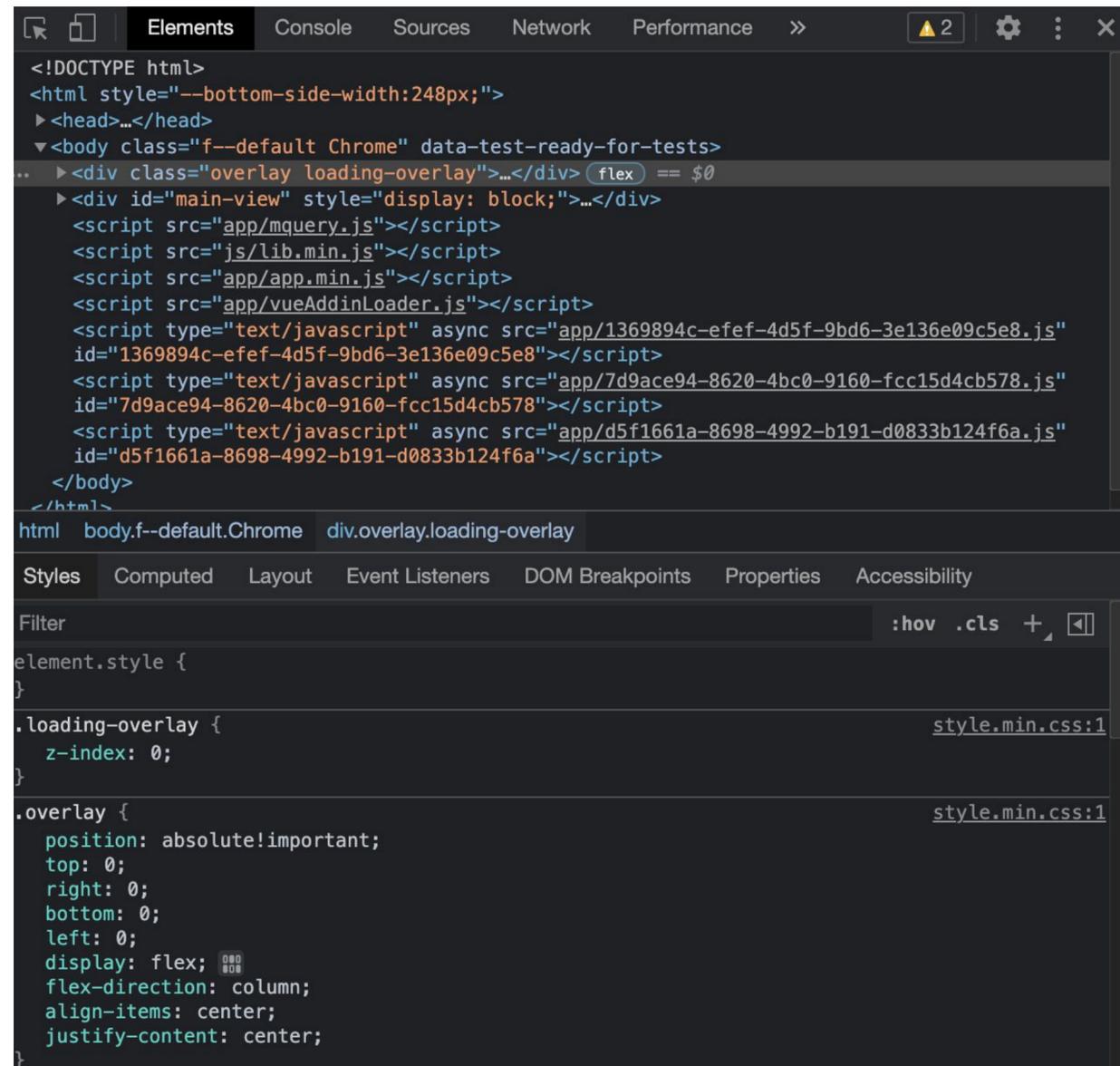


- Сообщения об ошибках (красным цветом)
- Предупреждения (желтым цветом)
- Синий символ > обозначает командную строку (в ней можно запускать JavaScript-команды)

Точный внешний вид инструментов разработки зависит от используемой версии браузера

Инструменты разработчика (devtools)

Вкладка Elements используется для просмотра и редактирования любых HTML элементов на странице, CSS-стилей



The screenshot shows the Chrome DevTools 'Elements' panel. The top part displays the DOM tree with the following structure:

```
<!DOCTYPE html>
<html style="--bottom-side-width:248px;">
  <head>...</head>
  <body class="f--default Chrome" data-test-ready-for-tests>
    <div class="overlay loading-overlay">...</div>
    <div id="main-view" style="display: block;">...</div>
  </body>
</html>
```

The selected element is a `div` with class `loading-overlay`. The 'Styles' panel below shows the following CSS rules:

```
element.style {
}

.loading-overlay {
  z-index: 0;
}

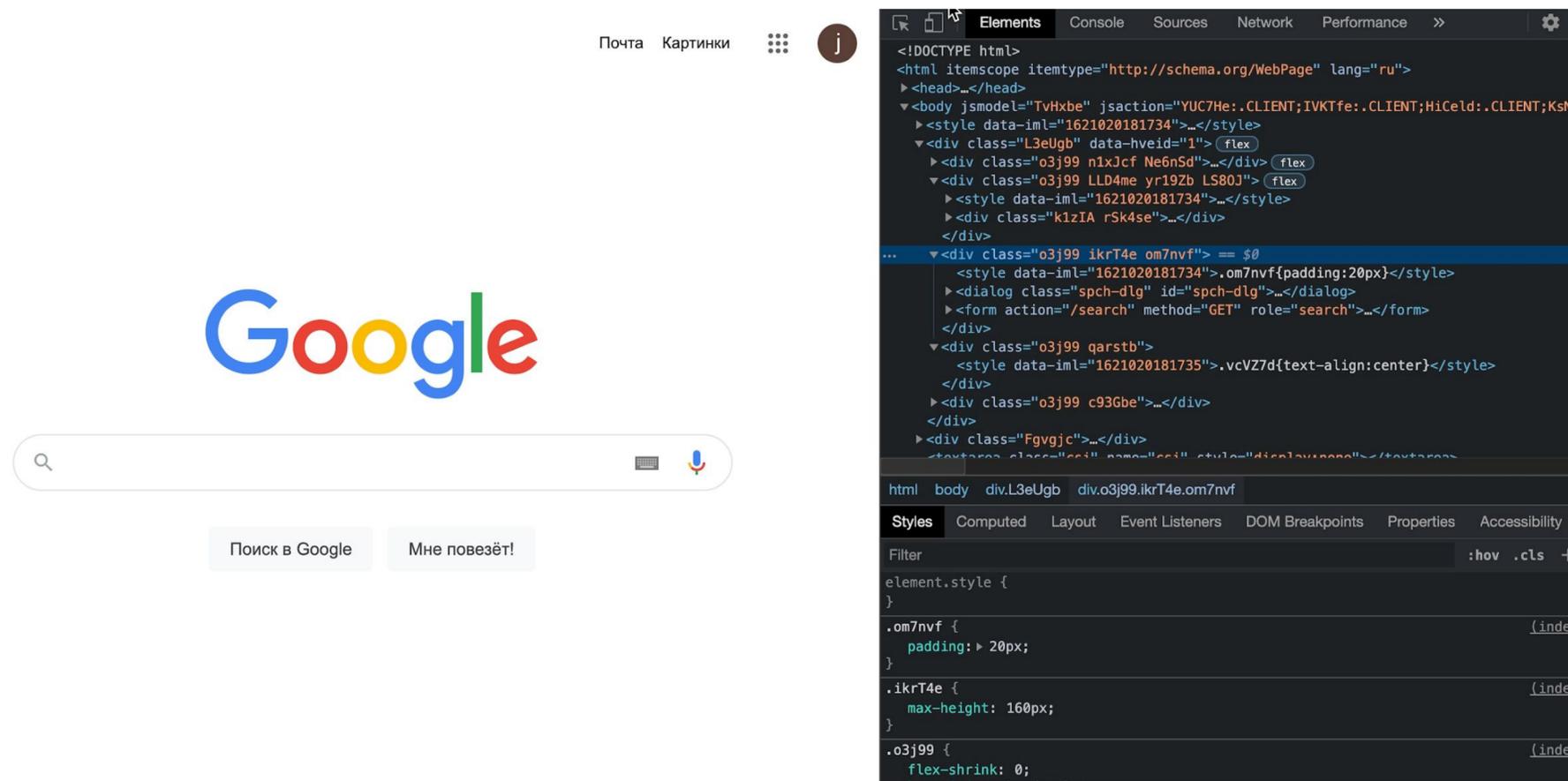
.overlay {
  position: absolute!important;
  top: 0;
  right: 0;
  bottom: 0;
  left: 0;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
}
```

Инструменты разработчика (devtools)

Во вкладке Elements есть кнопки

- выбор элемента с помощью курсора
- переключение в режим выбора устройств

выбор элемента с помощью курсора

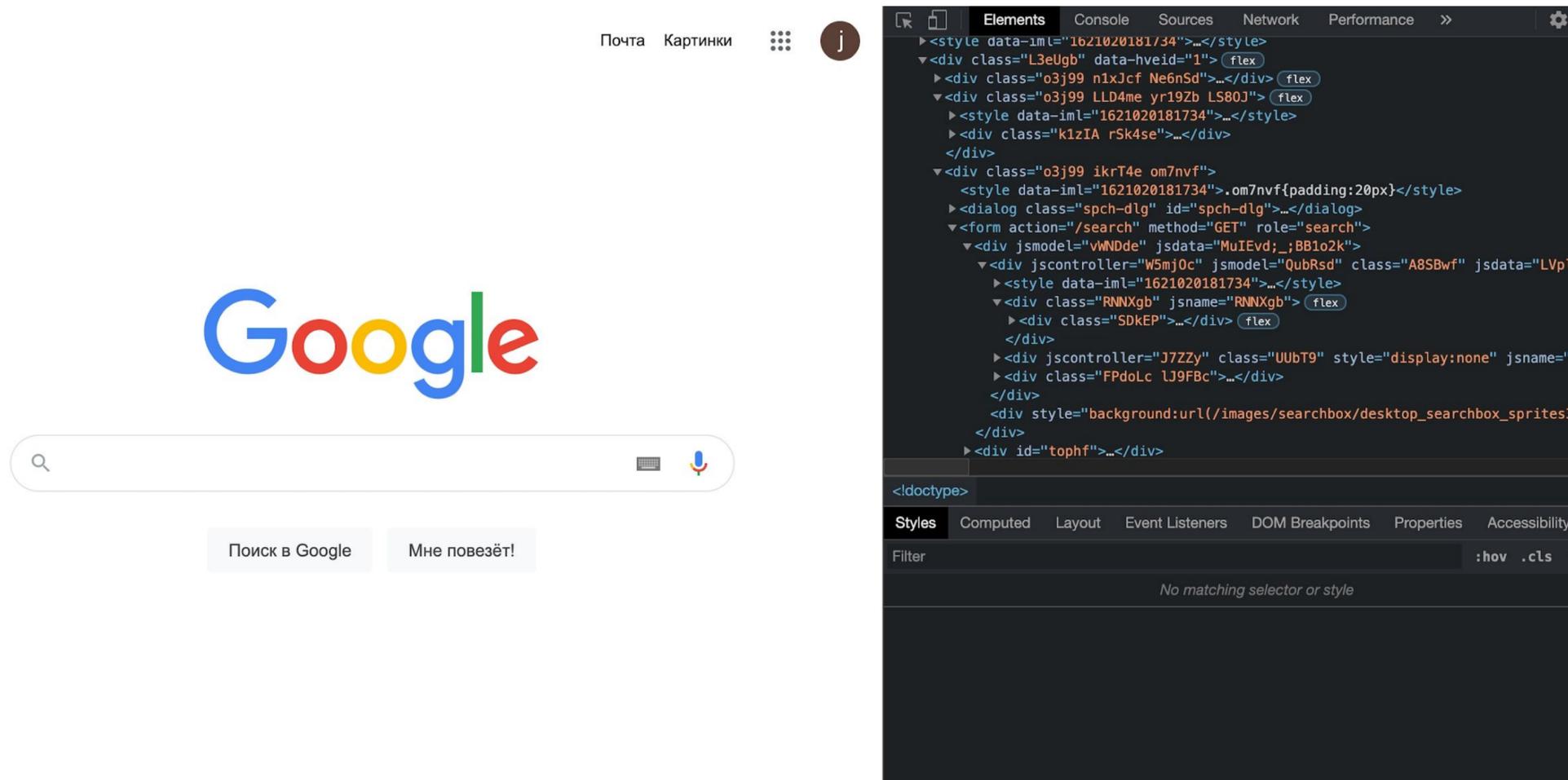


Инструменты разработчика (devtools)

Во вкладке Elements есть кнопки

- выбор элемента с помощью курсора
- переключение в режим выбора устройств

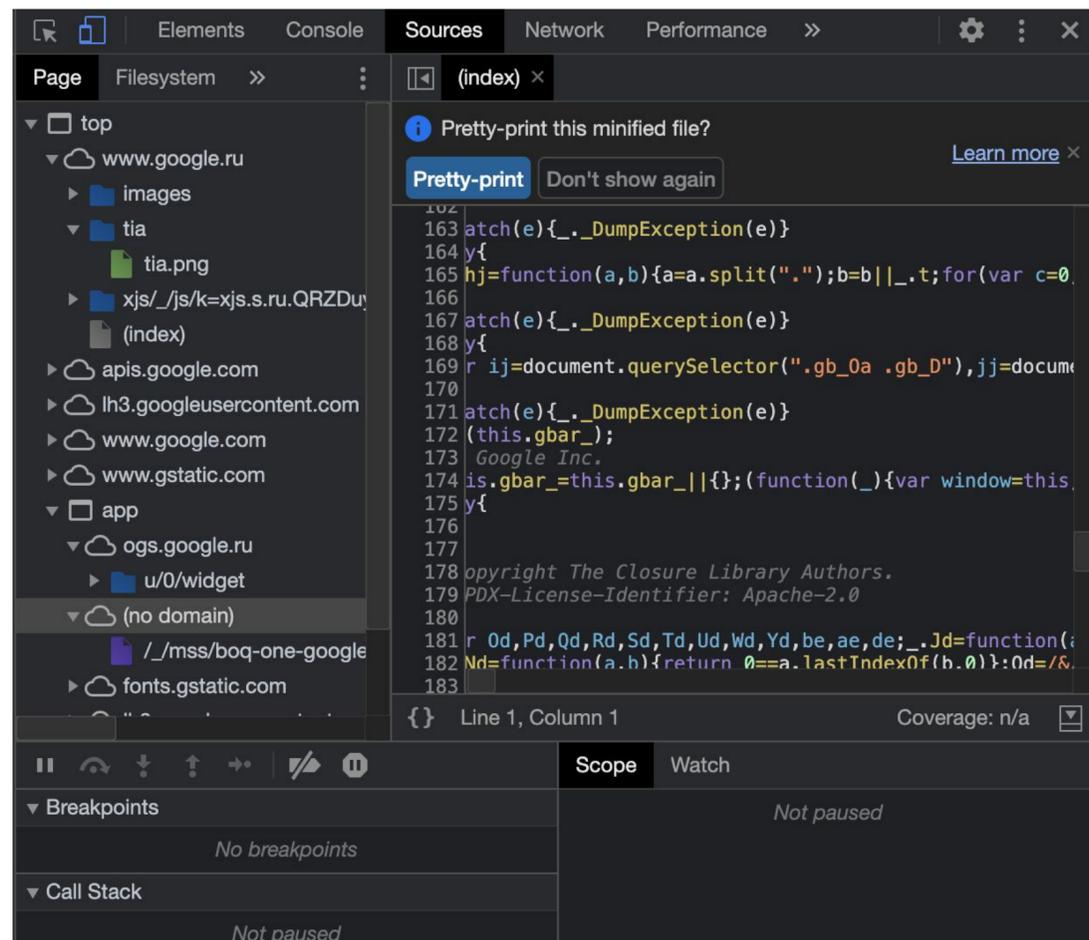
переключение в режим выбора устройств



Инструменты разработчика (devtools)

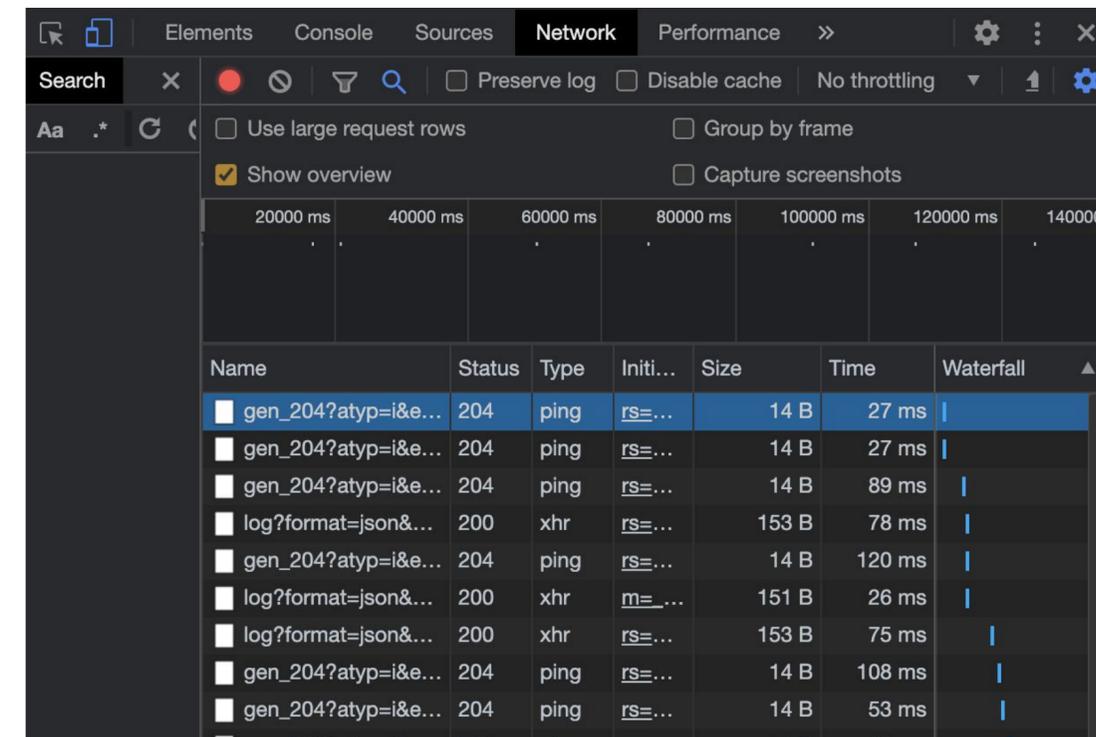
Вкладка Source:

- можно посмотреть все файлы подключенные на странице
- используется для отладки JavaScript-кода



Вкладка Network:

- мониторинг процесс загрузки страницы и всех файлов, которые подгружаются при загрузке.
- оптимизация загрузки страниц
- Подробная информация о каждом запросе



Домашнее задание

- Найдите инструменты разработки в вашем браузере и попробуйте открывать их на различных сайтах (лучше использовать браузер Chrome или Firefox)
- Обратите внимание, есть ли во вкладке Console какие-либо сообщения, ошибки и т.д.
- Зайдите во вкладку Elements и попробуйте перейти к коду какого-нибудь элемента с помощью инструмента «выбор элемента с помощью курсора»
- Посмотрите, как сайт будет выглядеть на различных устройствах с помощью инструмента «переключение в режим выбора устройств»

Редакторы кода, настройка редактора

Редакторы кода

Для написания кода может подойти любой редактор, наподобие обычного текстового редактора, даже «блокнот»

НО!

Нам нужен редактор, в котором будет удобно работать, в котором будет отображаться структура проекта, в котором будут подсказки по синтаксису кода

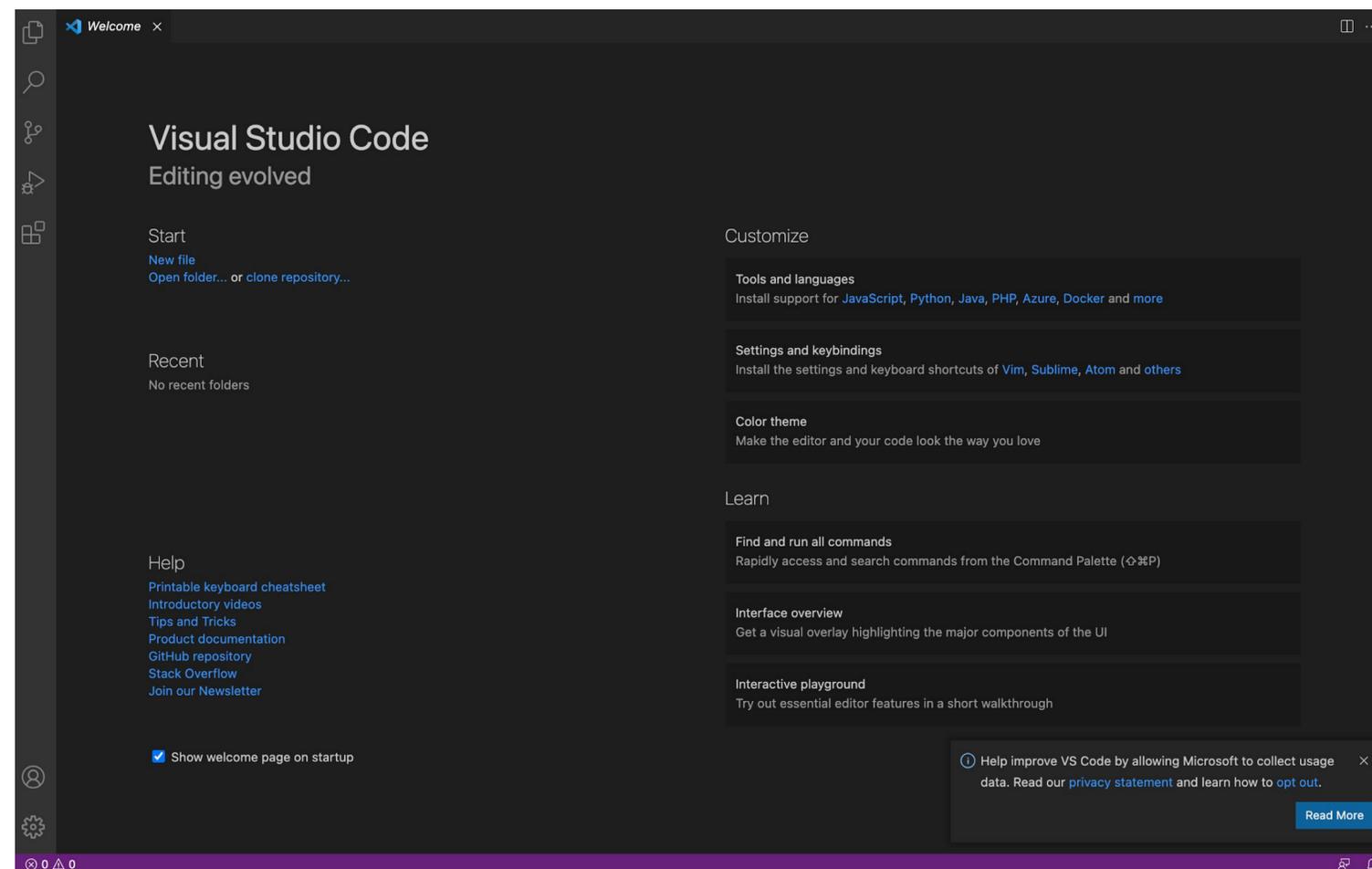
В дальнейшем мы будем работать с редактором Visual Studio Code

Установка Visual Studio Code

Скачать Visual Studio Code: <https://code.visualstudio.com/>

Стандартная установка программы

После установки при открытии Visual Studio Code появится приветственная страница



Что такое HTML

HTML

HTML (от англ. HyperText Markup Language) - язык разметки гипертекста.

HyperText

Ссылки, которые соединяют веб-страницы друг с другом

Markup

Разметка для отображения текста, изображений, блоков и других элементов в веб-браузере

- позволяет создавать и структурировать элементы на веб-странице
- дает понять браузеру, как нужно отобразить сайт

HTML

- Создан физиком из исследовательского института ЦЕРН в Швейцарии Тимом Бернерсом-Ли
- Первая версия HTML появилась в 1989 -1991 г.
- На данный момент существует 5 версий HTML
- Самое большое обновление - в 2014 г. вышла версия HTML5
- HTML считается официальным веб-стандартом
- НЕ является языком программирования

HTML

HTML-файл состоит из тегов - команд, которые преобразовываются в визуальные объекты в браузере

Пример простейшего HTML-документа

```
1  <!DOCTYPE html>
2  <html>
3    <body>
4
5      <h1>My First Heading</h1>
6      <p>My first paragraph.</p>
7
8    </body>
9  </html>
10
```

Отображение данного HTML-документа в браузере

My First Heading

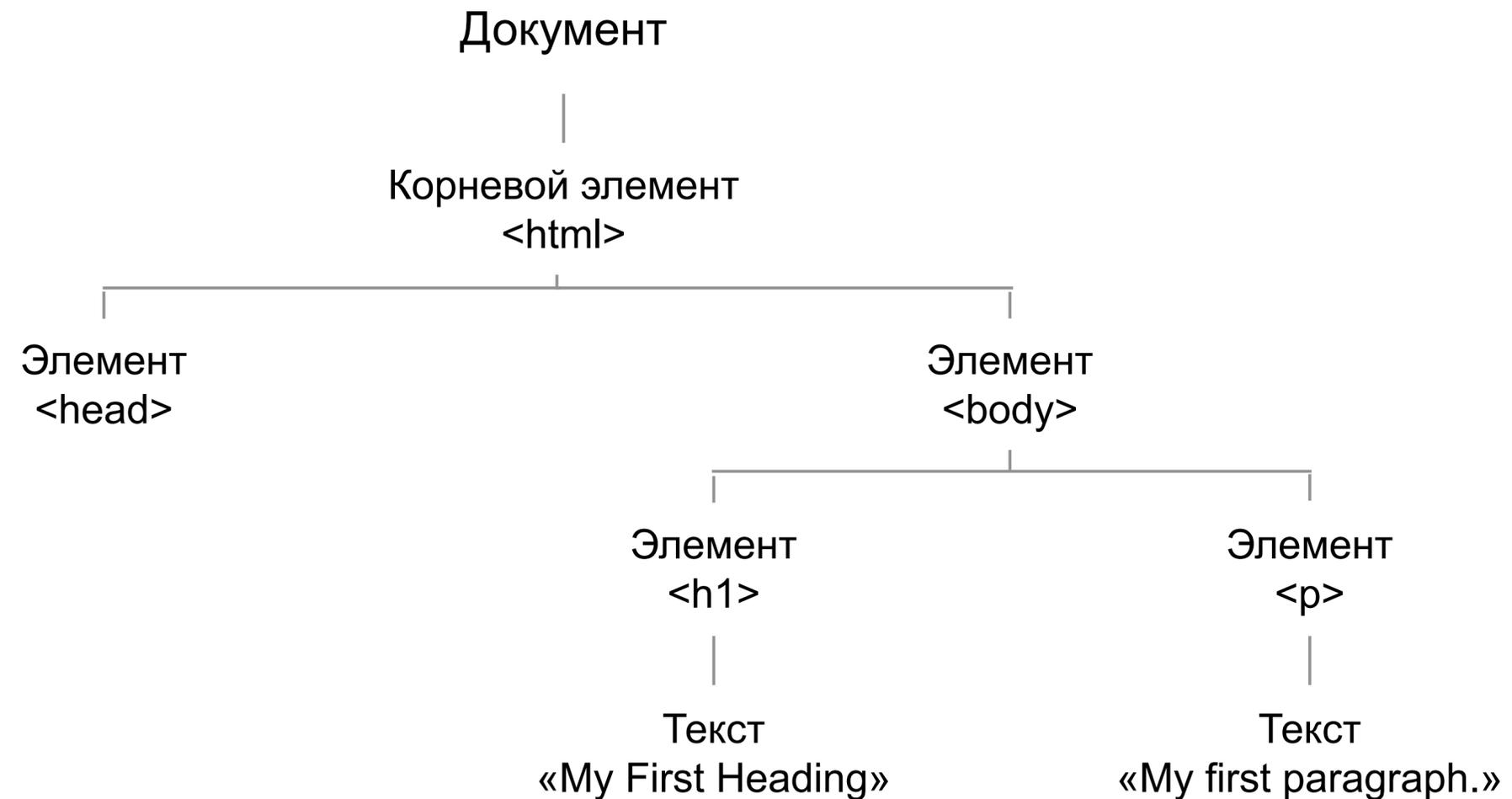
My first paragraph.

Что такое DOM

DOM

DOM (Document Object Model) - объектная модель документа

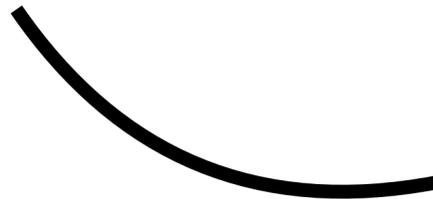
HTML-код анализируется и преобразуется браузером впоследствии в DOM. Модель HTML DOM представляется в виде дерева объектов.



DOM

Визуальное представление DOM можно увидеть в инструментах разработчика во вкладке Elements

DOM-дерево



The screenshot shows the browser's developer tools interface. The 'Elements' pane on the left displays the DOM tree structure:

```
<!DOCTYPE html>
<html>
  <head></head>
  <body> == $0
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
    <div>
      <p>Hello, world!</p>
    </div>
  </body>
</html>
```

The 'Styles' pane on the right shows the computed styles for the selected element (the body):

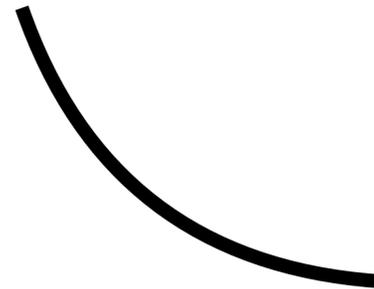
```
element.style {
}

body {
  display: block;
  margin: 8px;
}
```

Below the styles, a box model diagram is displayed, showing the margin (8px), border, padding, and the content area (713 x 110.438).

DOM

DOM-дерево



```
<!DOCTYPE html>
<html>
  <head></head>
  .. ▼ <body> == $0
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
    ▼ <div>
      <p>Hello, world!</p>
    </div>
  </body>
</html>
```

- каждый узел дерева - объект
- теги являются узлами - элементами (образуют структуру дерева)
- у узлов есть потомки

DOM

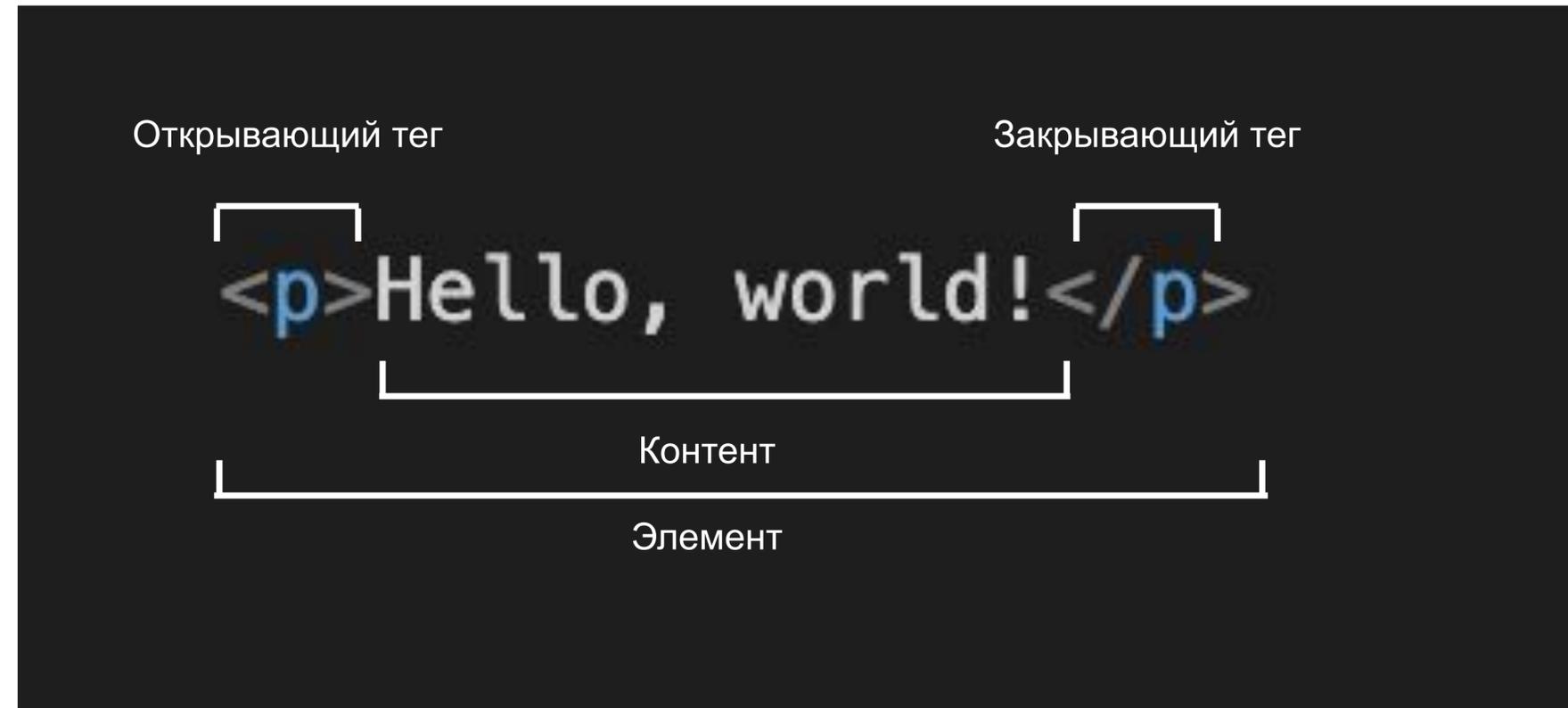
- В DOM закрываются незакрытые в HTML- документе теги
- В DOM добавляются обязательные теги, даже если они не проставлены в HTML- документе
- С элементами DOM можно проводить манипуляции с помощью JavaScript (динамическое добавление элементов, изменение, удаление)
- DOM - инструмент, с помощью которого JavaScript видит содержимое HTML-страницы и состояние браузера

Синтаксис и теги HTML

Синтаксис HTML

- HTML состоит из элементов
- Элемент - базовая сущность веб-страницы
- Все элементы делятся по типу и назначению
- Для обозначения начала и конца элемента применяются теги

Теги HTML



Элемент = открывающий тег + контент + закрывающий тег

Теги HTML

Теги не чувствительны к регистру, могут быть записаны как строчными, так и заглавными буквами

Например `<HEAD>` и `<head>` - сработают оба варианта

НО! Принято записывать теги в нижнем регистре, то есть в данном случае лучше вариант `<head>`

Типы элементов HTML

По типу элементы бывают:

- пустые - у которых нет закрывающего тега и содержимого (например, `
` или `<meta>`)
- Необрабатываемые текстовые элементы - предназначены для вывода скриптов или стилей, имеющих синтаксис отличный от HTML (например, `<script>`, `<style>`)
- Инородные элементы (например, элементы `svg`)
- Обычные элементы - все, которые не вошли в предыдущие группы

Комментарии в HTML

В HTML, как и в языках программирования, есть возможность писать комментарии в коде

Комментарии игнорируются обозревателем и не видны на странице в браузере

Комментарий помещается в специальные маркеры `<!--` и `-->`

Обычный элемент →

```
<p>Hello, world!</p>
```

Комментарий →

```
<!-- Этот текст не отобразится на странице -->
```

Базовая структура страницы

Базовая структура HTML документа

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <meta name="description" content="краткое описание страницы">
```

```
    <meta name="keywords" content="ключевые, слова">
```

```
    <title>Тестовая страница</title>
```

```
  </head>
```

```
  <body>
```

```
    <p>Hello, world!</p>
```

```
  </body>
```

```
</html>
```

Домашнее задание

- Создайте в редакторе кода новый файл с расширением .html
- Скопируйте код с предыдущего слайда и вставьте его в ваш файл
- Сохраните этот файл и попробуйте открыть его в браузере
- Попробуйте изменить в коде заголовок страницы, который отображается на вкладке в браузере
- Попробуйте добавить на страницу еще один текстовый элемент (в элементе `<p>` с любым текстом), и проверьте, появился ли он на странице в браузере

Атрибуты

Атрибуты

Атрибуты тегов расширяют возможности самих тегов и позволяют гибко управлять различными настройками отображения элементов веб-страницы

```
<p class="main-text">Hello, world!</p>
```

Атрибут class

```
<a href="newlink.html">Это ссылка на какой-нибудь документ</a>
```

Атрибут href

Атрибуты

```
<p class="main-text">Hello, world!</p>  
      └──────────┘  
      Атрибут class  
<a href="newlink.html">Это ссылка на какой-нибудь документ</a>  
      └──────────┘  
      Атрибут href
```

Атрибут должен иметь:

- Пробел между атрибутом и именем элемента (или предыдущим атрибутом, если у элемента уже есть один или несколько атрибутов).
- Имя атрибута и следующий за ним знак равенства.
- Значение атрибута, заключённое в кавычки.

Порядок атрибутов в любом теге не имеет значения и на результат отображения элемента не влияет.

Типы атрибутов

- **class** (используется для связывания элемента со списком стилей и задает класс элементу)
- **id** (для однозначной идентификации любого элемента внутри html-страницы)
- **style** (применяется для определения стиля элемента с помощью правил CSS)
- **title** (дает название элементу. Определяет дополнительную информацию, задавая всплывающую подсказку для страницы)
- **атрибуты интернационализации** (`dir` — позволяет указать браузеру направление потока текста, `lang` — позволяет указать основной язык, используемый в документе)