# HTML

1. **HTML** (**HyperText Markup Language**) – язык, на котором описывается структура элементов сайта.
2. Структура состоит из иерархических «блоков» - **узлов** / **тэгов** / **заголовков**.

Структура строится иерархически из элементов (тегов, узлов), таких как <html>, <div>, <p>, <h1>.

Пример:

Так корневой узел в структуре html это <html></html>:  
  
<!DOCTYPE html>

<html>

<head>

<title>Минимальная страница</title>

</head>

<body>

<h1>Привет мир!</h1>

<p>Это самая простая HTML страница</p>

</body>

</html>

# Браузер

## Определения

Страница / сайт / документ -

## Общая механика работы браузера

1. Это обычное десктопное приложение, которое умеет делать запросы в интернет и расшифровывать ответы
2. Визуал сайта это данные которые браузер получил на запрос и расшифровывал.

Когда пользователь делает запрос и сайт визуально открывается – это и есть данные, которые сайт получил в ответ на совершенный запрос.

1. HTTP и прочие протоколы, и многие другие интернет стандарты устроены таким образом, чтобы с помощью сериализованных битовых данных можно было отправлять и принимать любые данные: визуал сайтов и изображения, разные типы файлов.
2. Таким образом браузер:
3. Отправляет запрос …
5. Более полной и корректный механизм работы браузера представлен в Механизм работы браузера

## Детали устройства работы браузера

1. Как и любая другая дескопная программа имеет в основе объектно-ориентированную логику и такие компоненты браузера как (вкладка, ..) – это **объекты** (то есть экземпляры некоторого класса).

## Корневой объект страницы

1. Так такой компонент как **вкладка (window)** – это корневой объект для всего визуала, который будет отображен для пользователя.

Вся информация о сайте, его структуре, его визуале итд содержится в window:

1. У этого класса есть множество свойств и методов, но основные свойства касающиеся сайта:  
   <https://developer.mozilla.org/ru/docs/Web/API/Window>

window.document // DOM документ

window.navigator // Информация о браузере

window.location // URL и навигация

window.history // История браузера

window.screen // Информация об экране

window.frames // iframes

window.parent // Родительское окно

window.top // Верхнее окно

window.self // Текущее окно

## HTML файл -> визуальная страница во вкладке

1. Сайт (или страница / документ) имеет определенную структуру заголовков (или **тэгов** / **узлов**), которая описывается в html файлах.

Заголовок / тэг / узел - в html это <div>, <table>, итд.

1. Однако браузер отображает не полученный html на прямую, а
2. Считывает html файл, разбивая структуру на узлы.
3. В оперативной памяти создаёт объекты (экземпляры классов) для каждого узла, так чтобы получилась точно такая же структура из соответсвующих объектов.
4. Каждый объект на странице вкладки отображает свой визуальный элемент, так что получается структура из визуальных элементов и как результат полная визуальная структура – копия исходного html.

Тк такая структура объектов (узлов) существует в оперативной памяти, то с ней уже можно взаимодействовать с помощью скриптов изменять визуал страницы (изменять DOM).

1. Более полный и детальный процесс рендеринга страницы из html писан в Рендеринг страниц.

## DOM

1. Таким образом вся структура сайта в оперативной памяти – это совокупность объектов узлов, которые объединили в чисто условное понятие **DOM** (**Document Object Model**).

DOM – это только обозначение для множества объектов узлов, DOM не представлен в коде вообще.

1. Можно сказать, что страница целиком представлена в объекте document, который обязан соответствоать DOM спецификации.

А саму структуру страницы можно назвать DOM и она представлена в виде структуры из объектов узлов каждый из которых является свойством другого объекта, кроме корневого document.

Иван иванов отец семьи – document

Семья ивановых - DOM

## Скрипты и изменение DOM

1. Для взаимодействия с каждым узлом DOM существует специальный объект **document** , представляющий текущую страницу на вкладке и являющийся свойством объекта window.  
     
   С таким объектом в оперативной памяти уже можно взаимодействовать и который может изменяться в отличии от html.

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Материалы

### Механизм работы браузера

1. Процесс рендеринга страницы (от получения байтов до отрисовки пикселей) — это сложный конвейер. Упрощенно его можно представить так:
2. Навигация: Браузер отправляет HTTP(S) запрос на сервер по введенному URL.

Получение ответа: Сервер отвечает. Ответ обычно включает:

Статус-код (например, 200 OK или 404 Not Found)

Заголовки (Headers) с мета-информацией (тип содержимого, кодировка и т.д.)

Тело ответа (Body) — сам HTML-документ.

1. Парсинг (Разбор HTML):

Браузер начинает обрабатывать полученные байты HTML.

Он строит Дерево DOM (Document Object Model). Это пошаговый процесс: читаются теги, создаются узлы, строится иерархия.

Загрузка дополнительных ресурсов: Парся HTML, браузер находит ссылки на CSS, JavaScript, изображения, шрифты. Для каждого из этих ресурсов он отправляет новые запросы.

Обработка CSS и построение CSSOM: Пока грузятся CSS-файлы, браузер парсит их и строит CSSOM (CSS Object Model) — дерево стилей.

1. Компоновка (Layout / Reflow): Браузер комбинирует DOM и CSSOM, чтобы вычислить геометрию каждого элемента на странице: его размеры, положение относительно viewport (области просмотра) и друг друга.
2. Отрисовка (Paint): На этом этапе браузер преобразует вычисленную геометрию в пиксели на экране. Он "рисует" элементы в их слоях.
3. Композиция (Composition): Слои объединяются в окончательное изображение, которое пользователь видит на экране.

1. Любое взаимодействие с JavaScript, изменяющее DOM или стили, может запустить часть этого конвейера заново (например, только перерисовку или, что дороже, перекомпоновку).

### Рендеринг страниц

1. Шаги рендера.
2. Парсинг HTML: Браузер читает HTML-код и строит из него DOM-дерево — иерархическую структуру объектов в памяти.
3. Парсинг CSS: Строится CSSOM — дерево стилей.
4. Формирование дерева рендеринга: DOM и CSSOM объединяются.
5. Компоновка (Layout): Вычисляется точное положение и размер каждого элемента.
6. Отрисовка (Paint): Браузер "рисует" пиксели на экране.