

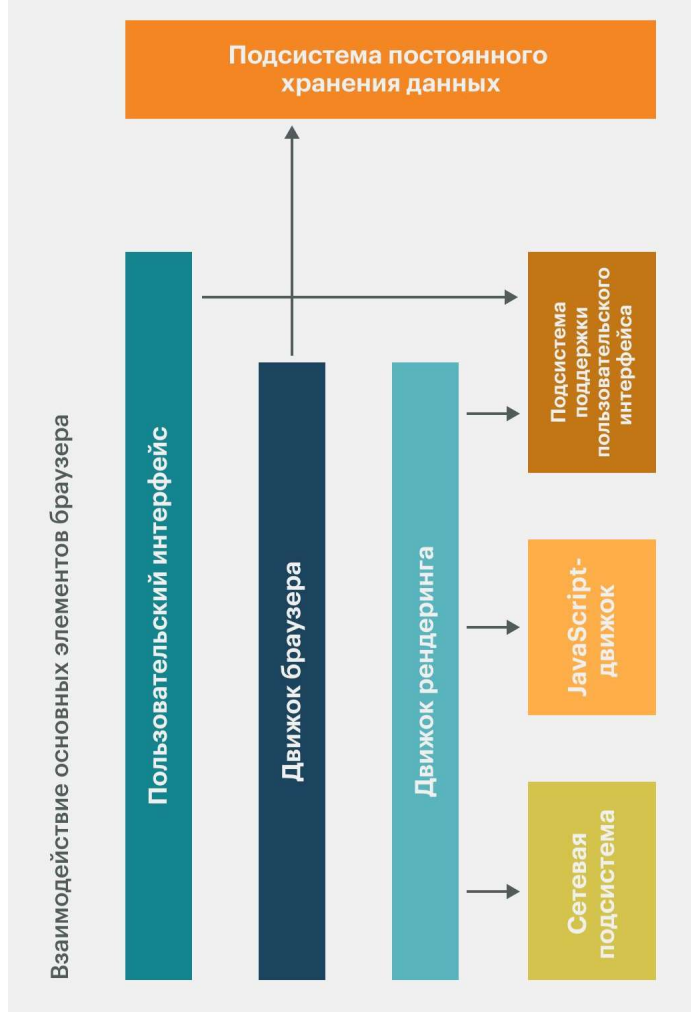
Браузер

Браузеры открывают сайты — это их основная функция.

Кроме того, они:

- позволяют скачивать на устройство пользователя файлы любого типа;
- хранят ссылки, пароли, историю посещений, платёжную информацию;
- позволяют общаться в социальных сетях, по email, через чаты и форумы;
 - позволяют получать [онлайн-образование](#);
 - защищают от вирусов и блокируют рекламу;
 - переводят страницы с иностранного языка;
- дают возможность для кастомизации интерфейса;
 - поддерживают расширения и виджеты.

Техническое устройство браузера:



Пользовательский интерфейс (user interface) — это все элементы браузера, с которыми может взаимодействовать пользователь. К интерфейсу относятся кнопки, поисковая строка, панель закладок, главная страница браузера, визуальный облик окна — всё, кроме содержимого сайтов.

Движок браузера (browser engine) — это механизм, который переводит закодированную информацию с сервера в интерактивное изображение на экране. Работает в тесной связке с движком рендеринга. Chrome, «Яндекс Браузер», Opera, Microsoft Edge работают на движке Blink. Mozilla Firefox на собственном браузерном движке Quantum. Safari использует WebKit.

Движок рендеринга (rendering engine) — механизм для визуализации графики.

Сетевая подсистема (networking) — обрабатывает сетевые запросы и отвечает за безопасность.

Подсистема поддержки пользовательского интерфейса (UI backend) отвечает за вывод основных компонентов интерфейса, таких как окна и элементы управления.

JavaScript-движок (JavaScript engine) — выполняет JS-код сайтов и передаёт его в механизм рендеринга для отображения.

Подсистема постоянного хранения данных (data persistence) — хранит пользовательские данные, такие как закладки, файлы cookie и настройки.

HTML

HyperText Markup Language - язык разметки гипертекста, используется для описания структуры содержимого веб-страниц, а именно, расположения элементов друг относительно друга, их внешнего представления, их функций.

1 Базовый синтаксис:

<!DOCTYPE html>

<html>

<head>

Информация в head

</head>

<body>

Информация в body

</body>

</html>

<head>

Тег <head> используется для описания метainформации о веб-странице (информация об информации). Эта информация не видна пользователю, а используется браузерами и поисковыми движками.

Один из наиболее важных тегов, используемых в <head> это незакрывающийся <meta>. С его помощью можно задать используемую кодировку, добавить описание страницы, ключевые слова для поисковиков, имя автора.

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

```
<meta name="description" content="Free Web tutorials">
```

```
<meta name="keywords" content="HTML,CSS,XML,JavaScript">
```

```
<meta name="author" content="John Doe">
```

Для того, чтобы сделать веб-страницу адаптивной, т.е. чтобы она подстраивалась под размеры экрана устройства, нужно использовать viewport. Этот параметр обязательно надо добавлять для всех страниц (если конечно хотите сделать веб-страницу пригодной для использования в 2019+ году :-)).

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Внутри <head> можно задавать внутренние стили для документа:

```
<style>
```

```
</style>
```

И внешние стили:

```
<link rel="stylesheet" href="mystyle.css">
```

Также внутри <head> обычно указываются скрипты:

```
<script>
```

```
</script>
```

<body> (базовая структура страницы)

```
<body>
  <header>
    <nav>
      Navigation
    </nav>
  </header>
  <section>
    <main>
      <article>
        Article
      </article>
      <article>
        Article
      </article>
    </main>
    <aside>
      Aside element
    </aside>
  </section>
  <footer>
    </footer>
```


[`<main>`](#) предназначен для содержимого, *уникального для этой страницы*. Используйте `<main>` только *один раз* на странице и размещайте прямо внутри [`<body>`](#). В идеале он не должен быть вложен в другие элементы.

[`<article>`](#) окружает блок связанного содержимого, который имеет смысл сам по себе без остальной части страницы (например, один пост в блоге).

[`<section>`](#) подобен `<article>`, но больше подходит для группирования одной части страницы, которая представляет собой одну часть функциональности (например, мини-карту или набор заголовков статей и сводок). Считается хорошей практикой начинать каждый раздел с [`<заголовка>`](#). Также обратите внимание, что в зависимости от контекста вы можете разбить `<article>` на несколько `<section>` или, наоборот, `<section>` на несколько `<article>`.

[`<aside>`](#) содержит контент, который не имеет прямого отношения к основному содержимому, но может содержать дополнительную информацию, косвенно связанную с ним (словарь, биография автора, связанные ссылки и т. д.).

[`<header>`](#) представляет собой группу вводного содержимого. Если он дочерний элемент [`<body>`](#), то он определяет глобальный заголовок веб-страницы, но если он дочерний элемент [`<article>`](#) или [`<section>`](#), то определяет конкретный заголовок для этого раздела (постарайтесь не путать его с [`titles`](#) и [`headings`](#)).

[`<nav>`](#) содержит основные функции навигации для страницы. Так же часто в нем можно увидеть логотип и / или название сайта или компании. Вторичные ссылки и т. д. не входят в навигацию.

[`<footer>`](#) представляет собой группу конечного контента для страницы.

Атрибуты

- это дополнительные значения, которые настраивают элементы или регулируют их поведение различным способом, чтобы соответствовать критериям пользователей.

Пример:

- Class - Задаёт стилевой **класс**, который позволяет связать определённый тег со стилевым оформлением. В значении допускается указывать сразу несколько **классов**, разделяя их между собой пробелом.
- Id - определяет уникальный идентификатор элемента **HTML**. Значение **атрибута id** должно быть уникальным в **HTML** документе. **Атрибут id** используется для указания на конкретную декларацию стиля в таблице стилей. Он также используется JavaScript для доступа и управления элементом с указанным **id**. data-attribute добавляем данные к элементу
- style - добавляем стили элементу
- Title - используется для указания дополнительной информации об элементе. При наведении курсора мыши на элемент отображается информация.
- alt - краткое описание того, что изображено на картинке target: "_blank"
- href - ссылка
- type (input) Сообщает браузеру, к какому типу относится элемент формы. посмотреть (button , password, radio, text, color, date, email range, search tel)

Строчные элементы - генерируют внутристрочные контейнеры. Они не формируют новые блоки контента.

<a>, <abbr>, <acronym>, , <bdo>, <big>,
, <button>, <cite>, <code>, <dfn>, , <i>, , <input>, <kbd>, <label>, <map>, <object>, <q>, <samp>, <script>, <select>, <small>, , , <sub>, <sup>, <textarea>, <time>, <tt> и< var>.

Блочные элементы — элементы высшего уровня, которые формируются визуально как блоки, располагаясь на странице в окне браузера вертикально.

- [`<address>`](#) Контактная информация.
- [`<article>`](#) Содержание статьи.
- [`<aside>`](#) Побочное содержание.
- [`<blockquote>`](#) Длинная («блочная») цитата.
- [`<details>`](#) Раскрывающийся блок с подробностями.
- [`<dialog>`](#) Диалоговое окно.
- [`<dd>`](#) Описание определения.
- [`<div>`](#) Фрагмент документа.
- [`<dl>`](#) Список определений.
- [`<dt>`](#) Термин в списке определений.
- [`<table>`](#) [`\(en-US\)`](#) Таблица.
- [``](#) Неупорядоченный список.

- [`<fieldset>`](#) Группирование элементов формы.
- [`<figcaption>`](#) Описание для тега [`<figure>`](#).
- [`<figure>`](#) Группирование медиа-контента с подписью (см. [`<figcaption>`](#)).
- [`<footer>`](#) Нижняя часть («подвал») раздела или страницы.
- [`<form>`](#) Форма ввода.
- [`<h1>`](#) (en-US), [`<h2>`](#) (en-US), [`<h3>`](#) (en-US), [`<h4>`](#) (en-US), [`<h5>`](#) (en-US), [`<h6>`](#) (en-US)
- Заголовки от первого до шестого уровня.
- [`<header>`](#) Верхняя часть («шапка») раздела или страницы.
- [`<hgroup>`](#) Группирование заголовков раздела или страницы.
- [`<hr>`](#) Горизонтальная разделительная линия.
- [``](#) Пункт списка.
- [`<main>`](#) Содержит основной контент, уникальный для страницы.
- [`<nav>`](#) Содержит навигационные ссылки.
- [``](#) Упорядоченный список.
- [`<p>`](#) Текстовый абзац.
- [`<pre>`](#) Предварительно форматированный текст.
- [`<section>`](#) Раздел веб-страницы.