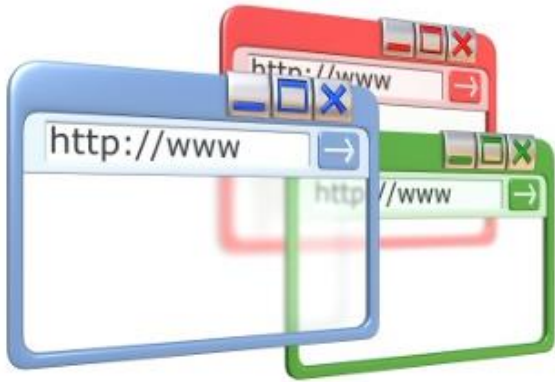
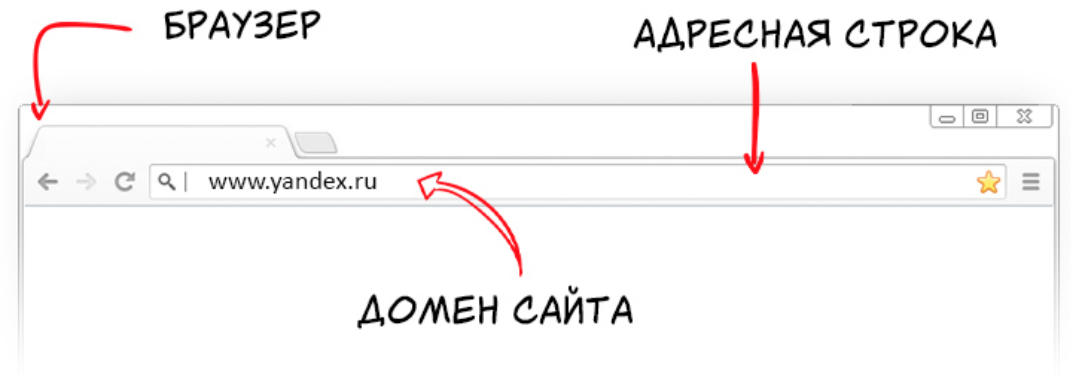


Что такое сайт?

➤ от англ. website: web — «паутина, сеть»
site — «место», букв. — «место, сегмент, часть в сети»



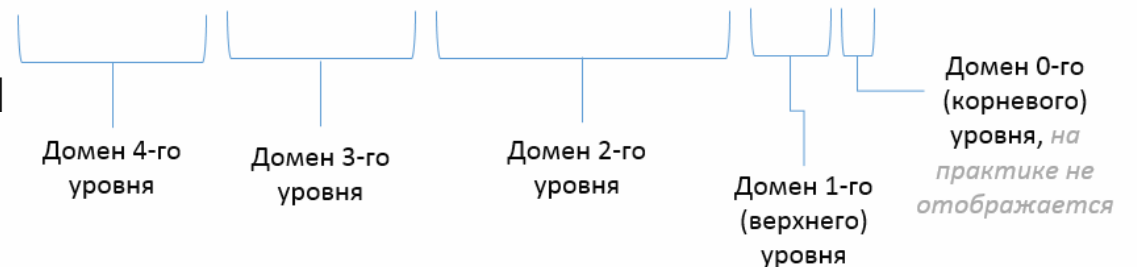
Веб-страница (англ. Web page)
— документ или
информационный ресурс
Всемирной паутины, доступ к
которому осуществляется с
помощью веб-браузера.



http://www.
https://www.

Hyper Text Transfer Protocol

cool.blog.mysite.ru.



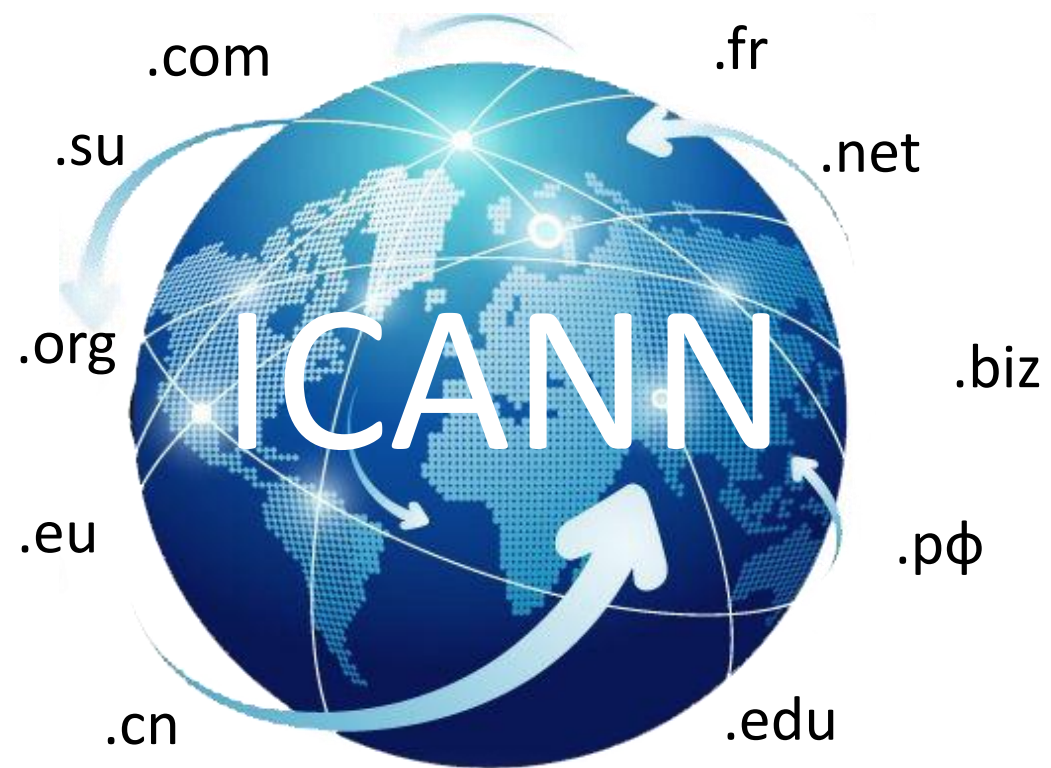
Задача HTTP — обмен данными между пользовательским приложением, осуществляющим доступ к веб-ресурсам (обычно это веб-браузер) и веб-сервером.

Административные Назначение

.gov	правительственные организации
.edu	образовательные ресурсы
.com	коммерческие ресурсы
.mil	военная сша
.net	компьютерная сеть
.org	некоммерческие ресурсы
.int	международные
.biz	ресурсы для бизнеса
.euro	европейские страны
.mobi	сайты для мобильных устройств
.travel	индустрия путешествий
.auto	автомобильная индустрия
.museum	музейное дело

Географические

	Страна
ru, рф	Россия
su	Бывший СССР
fr	Франция
us	США
de	Германия
uk	Англия
ca	Канада
cn	Китай



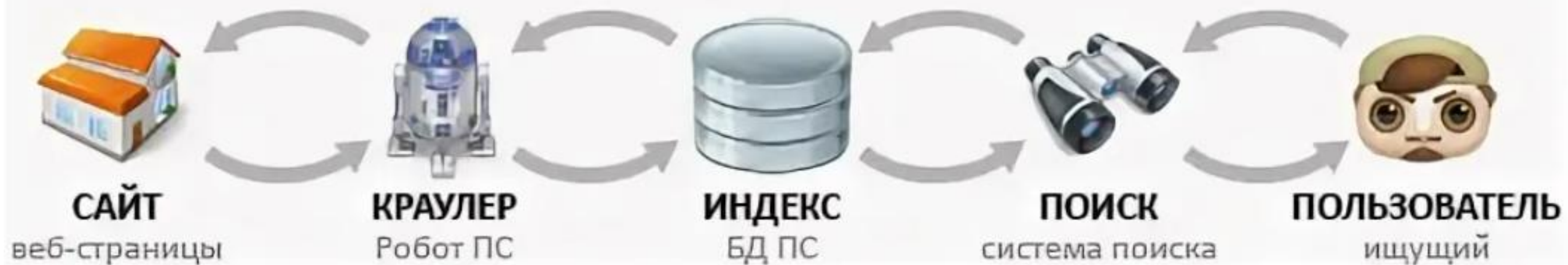


ИНДЕКСАЦИЯ

сбор и обработка
данных

ПОИСК

поиск и вывод
результатов



Основа сайтов — 3 технологии
обеспечивают представление информации
в различных ее видах:

технология HTML —
отвечает за
структурирование
информации и
выделение
логических блоков на
странице сайта

технология CSS —
используется для
оформления и
позиционирования,
представления
информации на
странице

технология JavaScript
— используется для
реализации
динамики и
интерактивного
взаимодействия с
пользователем

HTML (hyper text markup language) — язык разметки гипертекста

это набор **правил и знаков (тегов)**, которые используются для структурирования (разметки) текстовой информации, добавления в текст внешних изображений, создания таблиц, списков и т.д.

Текст, размеченный с помощью HTML интерпретируется браузером, в результате чего, пользователи видят не исходный код с элементами разметки, а итоговый результат обработки страницы.



HTML

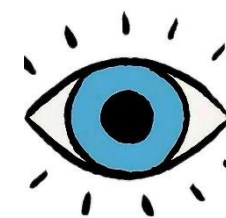
ТЕГИ

Структурный смысл
(таблица, список)

Семантический смысл
(информация для роботов
или браузеров)

Пошаговая инструкция

```
<ul>  
  <li> Достать ключ</li>  
  <li>Вставить ключ в замок</li>  
  <li>Повернуть ключ на два оборота</li>  
  <li> Достать ключ из замка </li>  
  <li>Открыть дверь</li>  
</ul>
```



Пошаговая инструкция

1. Достать ключ
2. Вставить ключ в замок
3. Повернуть ключ на два оборота
4. Достать ключ из замка
5. Открыть дверь

Кроссплатформенность — это

способность IT-продукта (сайта) полноценно работать на любом устройстве вне зависимости от типа операционной системы (ОС):

компьютерные Windows, Linux и MacOS или мобильные Android, iOS и т.д.

Задача разработчика:

сделать портал пригодным для удобного использования в любом браузере.

Не путать с **кроссбраузерностью**.

Это способность сайта работать в Google Chrome, Opera, Mozilla Firefox и других браузерах, не теряя при этом **скорости работы и читабельности страниц**.

Адаптивный веб-дизайн

Это автоматическая адаптация сайта под устройства с разным размером экрана. Не нужно разрабатывать мобильную версию или специальный комплект иконок.

Существует несколько вариантов адаптации:

- Растягивание или сжатие блоков до размеров экрана. Если это невозможно, они выстраиваются в ряд.
- Блоки выстраиваются по вертикали. Ширина экрана - это ширина одного блока.
- Для небольших ресурсов используйте масштабирование объектов и изображений.
- Страница делится на несколько блоков. На большом экране все блоки отображаются одновременно. На меньшем экране блоки отображаются по очереди. Пользователи перелистывает блоки и находят нужный.
- Изменение навигации.

Типы верстки:

1. Фиксированная верстка
2. Резиновая верстка
3. Табличная верстка
4. Верстка с div-блоками
5. Адаптивная верстка.
6. Гибкая верстка (Семантическая).



Современные стандарты HTML

HTML5 завершен в 2014 году и рекомендован для использования в современной веб-разработке.

HTML 4 (с 1997 года).

Определить его можно на основании **доктайпа**.

Это первая строка любого **HTML-документа**, позволяющего браузерам понять способ работы с содержимым страницы.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

Базис современной разработки – стандарт **HTML5**.

1.<p>Абзац<\p>

2.<P>Абзац<\P>

3.<p>Абзац

Специфика HTML5:

- добавлены семантические элементы (<nav>, <section>, <article> и др.), которые облегчают чтение структуры страницы как разработчикам, так и поисковым машинам;
- внедрена поддержка векторной графики и специальных форматов (**svg, canvas, MathML**);
- представлены новые элементы управления (**dates, email, tel**);
- удалены устаревшие теги (**big, center, isindex**).

Консорциум Всемирной сети

Некоммерческая организация **W3-Консорциум** (www.w3.org) разрабатывает единые принципы и рекомендации для производителей Интернет-контента, чтобы обеспечить максимальную совместимость между продуктами, платформами, программами.

Помимо непосредственно стандартов **HTML** компания занимается утверждением и других **связанных положений**:

- **CSS** (каскадные таблицы стилей),
- **DOM** (объектная модель документа),
- **PNG** (формат хранения растровых изображений),
- **HTTP** (протокол передачи данных),
- **RDF** (модель представления метаданных),
- **XPath** (язык для доступа к частям **XML-документов**) и др.



<https://validator.w3.org/>

The screenshot shows the W3C Markup Validation Service interface. At the top, there's a navigation bar with the W3C logo and the text 'Markup Validation Service' and 'Check the markup (HTML, XHTML, ...) of Web documents'. Below this, there are three tabs: 'Validate by URI', 'Validate by File Upload', and 'Validate by Direct Input'. The 'Validate by URI' tab is active, showing a form with an 'Address:' label and an input field. Below the input field is a 'More Options' link and a 'Check' button. The main content area contains a paragraph explaining the service: 'This validator checks the [markup validity](#) of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as [RSS/Atom feeds](#) or [CSS stylesheets](#), [MobileOK content](#), or to [find broken links](#), there are [other validators and tools](#) available. As an alternative you can also try our [non-DTD-based validator](#).' Below this is a 'W3C DEVELOPERS' badge and a text block: 'Interested in understanding what new technologies are coming out of W3C? Follow [@w3cdevs on Twitter](#) to keep track of what the future looks like!' followed by a 'Donate' link and the text 'and help us build better tools for a better web.' At the bottom, there's a footer with navigation links: 'Home', 'About...', 'News', 'Docs', 'Help & FAQ', 'Feedback', and 'Contribute'. The footer also includes the W3C logo, the text 'This service runs the W3C Markup Validator, v1.3+hg.', and a copyright notice: 'COPYRIGHT © 1994-2013 W3C® (MIT, ERCIM, KEIO, BEIHANG), ALL RIGHTS RESERVED. W3C LIABILITY, TRADEMARK, DOCUMENT USE AND SOFTWARE LICENSING RULES APPLY. YOUR INTERACTIONS WITH THIS SITE ARE IN ACCORDANCE WITH OUR PUBLIC AND MEMBER PRIVACY STATEMENTS.' There is also a small 'VALIDATOR' badge in the bottom right corner.

Тег – отдельный HTML элемент, позволяющий вкладывать или оборачивать части документа для определенного отображения или поведения.

Типичная структура web-элемента:

Открывающий тег

Атрибут, свойство

Содержимое

Закрывающий тег



```
<a href="https://yandex.ru">Ссылка </a>
```

Одиночные тэги 16 штук

Наиболее часто используемые:

```
<!doctype>  
<meta>  
<hr>  
<br>  
<img>  
<input>
```

Парные тэги

Парные тэги обязательно имеют **закрывающий тег** и содержимое (пусть даже и пустое).

В HTML их больше всего (около 100).

```
<p>Текст</p>  
<a href="#">Текст ссылки</a>  
<small>Текст</small>  
<mark>Текст</mark>
```

Блочные тэги занимают всю ширину страницы или родительского элемента.

Блочный тег	1. <div></div>
	2. <table></table>
	3. <p>Параграф</p>
Блочный тег	4. <h1>Большой заголовок</h1>
	5. <form></form>

Строчные тэги не имеют строго размера. Он зависит от того, сколько символов в них содержится.

Строчный тег Строчный тег

1.
2. Выделение жирным шрифтом
3. Важный текст
4. Ссылка в никуда
5. <sub>, <sup>

В CSS-свойствах отображение тегов можно менять по следующим типам:

1. **Блочные (display: block)** – получает свойство блочности, занимает всю ширину страницы;
2. **Строчные (display: inline)** – тег становится строчным;
3. **Флекс (display: flex)** – элемент приобретает свойство особого типа блоков – флекс (ведет себя как блок, но внутреннее содержимое гибко настраивается);
4. **Грид (display: grid)** – еще один блочный тип с особенностями внутреннего поведения элементов (строятся на основании «сетки», состоящей из строк и колонок);
5. **Таблица (display: table)** – тег и его содержимое будут наследовать свойства таблиц;
6. **Строчный блок (display: inline-block)** – блочный элемент ведет себя как строка, но сохраняет часть свойств (можно задать размер, границы).
7. **Спрятанный (display: none)** – делает элемент невидимым и полностью удаляет его отображение со страницы.

В **HTML5** прямого разделения тегов на блочные и строчные фактически нет. Сейчас говорят о **категориях контента**, который содержится внутри элемента.

Контентная модель **HTML5** позволяет более точно описать типы содержимого, с которым сталкиваются авторы при разработке сайтов. Она включает такие типы:

1. **Поток (flow)** – отображает основное содержимое страницы;
2. **Метаданные (metadata)** – применяются в заголовке документа и определяют поведение остального содержимого и связь с иными данными;
3. **Текст (phrasing)** – включает непосредственно текст страницы и используемые для его форматирования теги;
4. **Встроенный контент (embedded)** – часть текстового, строчного содержимого, включающая импортированный контент;
5. **Интерактивные элементы (interactive)** – позволяют посетителю сайта взаимодействовать с ресурсом;
6. **Заголовки (heading)** – для определения заголовков сайта, статей, подразделов;
7. **Секции (sectioning)** – выделение изолированного контента в блоки.

Большая часть тегов включается именно сюда. Например: **a, button, header, nav, ol, section**.

Пример тегов: **title, meta, script, style**

Фактически, это строчные элементы: **a, button, i, img, span, textarea**.

Пример тегов: **audio, video, img, svg, canvas**.

Включает теги: **button, select, video, a, input, textarea**.

Теги от **h1** до **h6**.

Включает теги: **article, aside, nav, section**.

HTML теги. Семантические теги

К основным тегам такого типа относят:

1. `<article>Будущая статья</article>`

Для выделения независимого, самоопределяемого контента. Тег подразумевает содержимое, которое несет ценность само по себе, обособливается в общем контексте сайта. Хорошим примером такого вида служит пост на форуме или в блоге, отдельная статья сайта, новость;

2. `<aside>Побочная информация</aside>`

Некие дополнительные данные, косвенно связанные с основным содержанием страницы. Например, вы пишете про списки в Python. В блоке `aside` можно рассказать про списки и их свойства в других языках программирования;

3. `<details>Дополнительные сведения</details>`

Справочная, дополняющая информация, которую пользователь по своему желанию может не смотреть. Это раскрывающийся блок с надписью **Подробнее** (по умолчанию). Сюда могут включать сведения об авторе, некие уточнения;

4. `<summary></summary>`

Дает краткую характеристику содержимого блока `details`.

5. `<figcaption>Описание</figcaption>`

Комментарий к картинке, пояснения, название;

6. `<figure></figure>`

Выделяет рисунок и его описание в единый блок.

7. `<footer>Заключение</footer>`

Позволяет выделить нижний колонтитул статьи или раздела сайта. Здесь часто подводят итоги, пишут информацию об авторе, резюмируют написанное;

```
<div>Чтобы создавать сайты необходимо знать HTML, CSS, JavaScript и серверную технологию (например, Node.js, Python или php)</div>
```

```
<div>Первый сайт, к слову, создал Тим Бернес-Ли</div>
```

```
<details>
```

```
<summary>W3-Консорциум</summary>
```

```
Возглавляется Тимом Бернесом-Ли, разработчиком языка разметки HTML.
```

```
</details>
```

8. `<header>Введение</header>`

Служит для вступительного содержания страницы сайта или статьи. Тут указывают заголовок, подзаголовок, некие вводные данные;

9. `<main>Основное содержимое</main>`

Основной контент документа. Если речь о статье, то именно тут находится ее тело за исключением вводной и заключительной частей;

10. `<nav>Разделы сайта</nav>`

Основное меню сайта заключают в тег `nav`. Содержит ссылки на разделы сервиса;

11. `<section>Некий блок сайта</section>`

Отдельный раздел документа, не обязательно независимый. Представляет некую тему с содержимым, которые требуется выделить в изолированный блок;

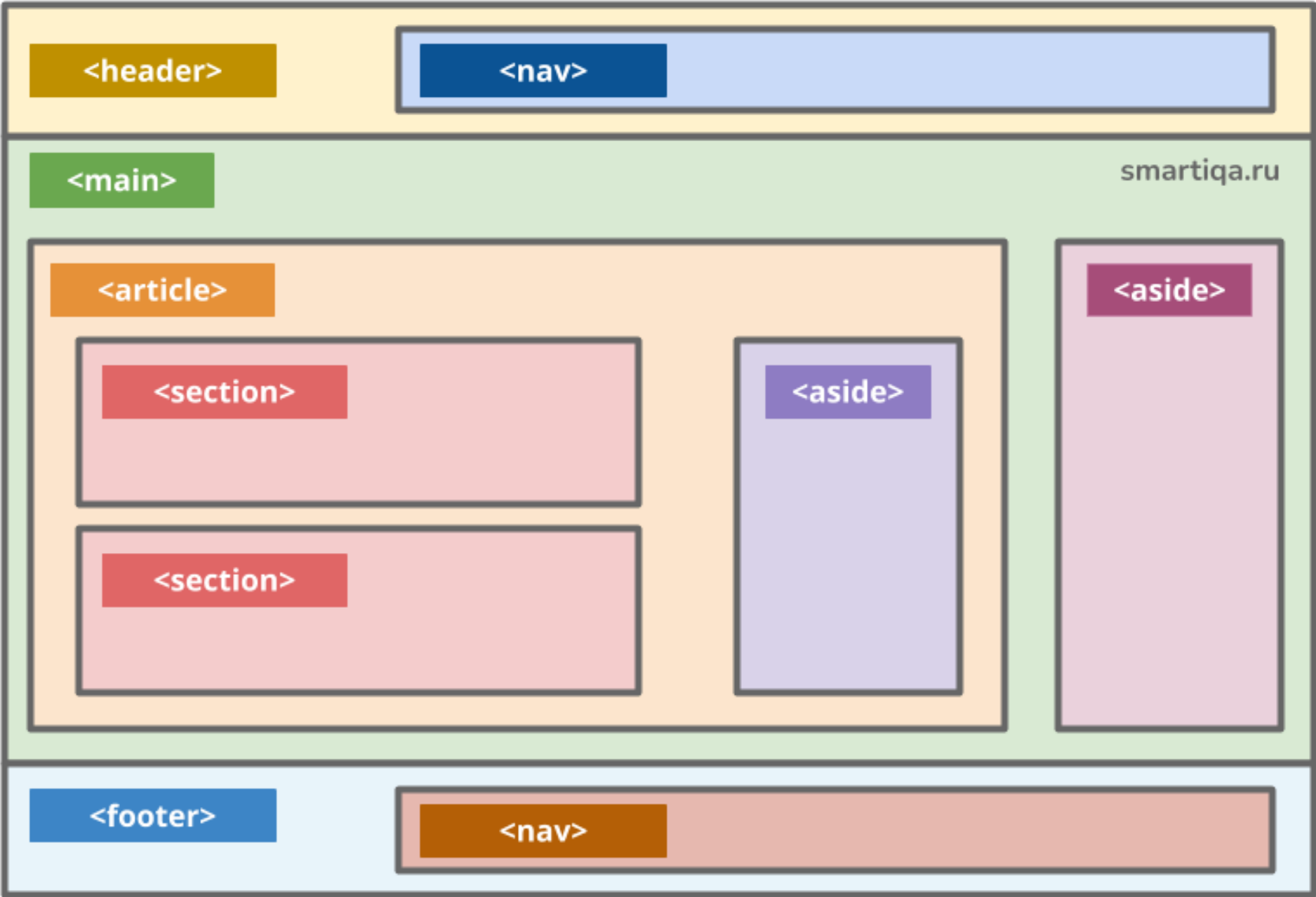
12. `<time></time>`

Позволяет браузеру и поисковым системам понять, что заключенные в тег данные представляют собой дату.

13. `<mark>Выделенные данные</mark>`

Содержит информацию, требующую выделения или подчеркивания.

Структура web-страницы, построенной на семантических тегах



HTML теги. Классификация атрибутов

Атрибуты или свойства тегов позволяют расширить функционал HTML-элементов либо предоставить дополнительные сведения о них.

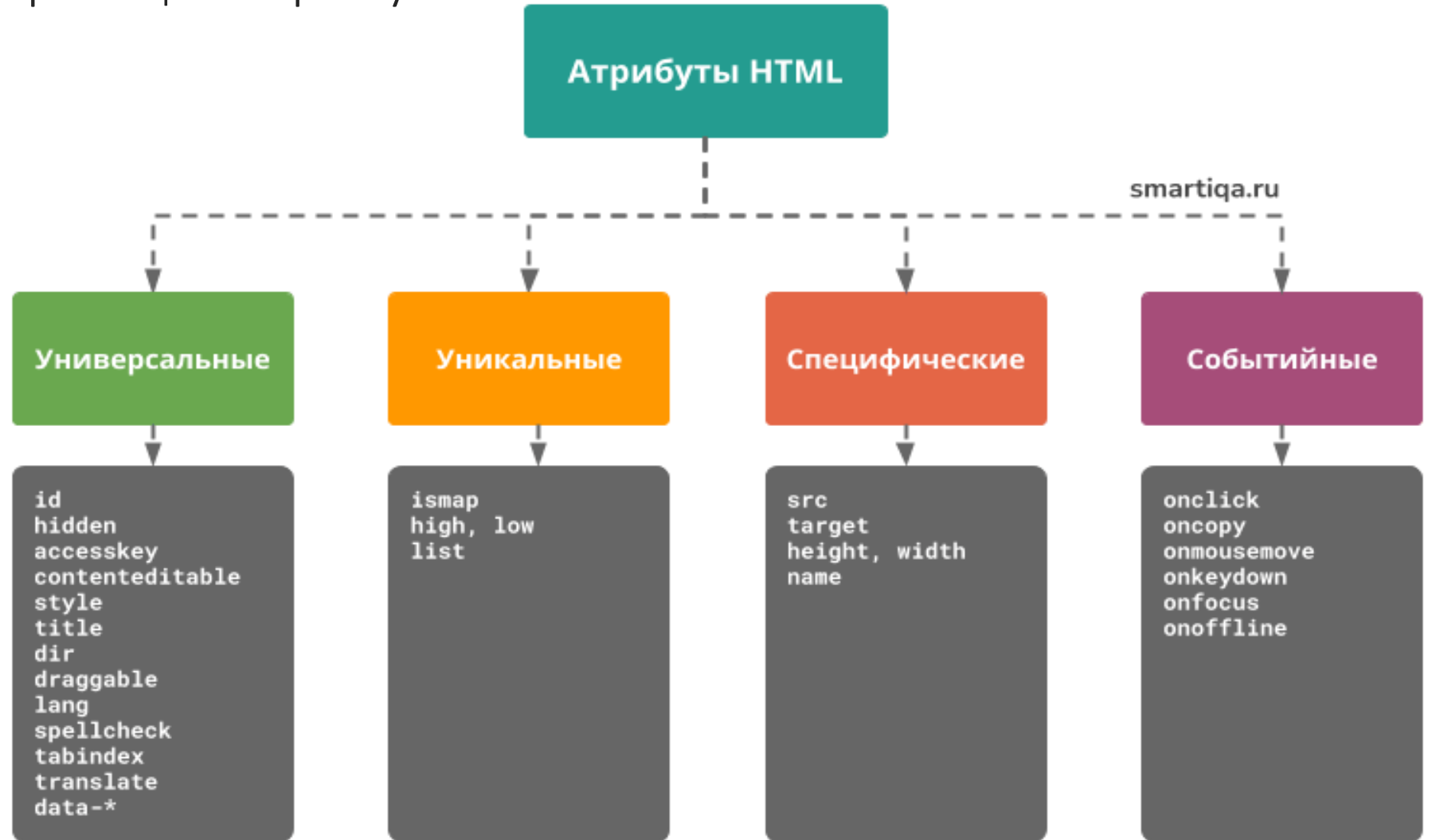
1. У всех тегов есть атрибуты (универсальные, уникальные, событийные или специфические);
2. Они необходимы для дополнения элемента новыми особенностями;
3. Они приводятся в открывающем теге через пробел;
4. Типичный вид атрибута – пара ключ="значение".

Атрибут, свойство

```
<html lang="ru">
```

...

```
</html>
```



Структура веб-страницы и DOM-модель

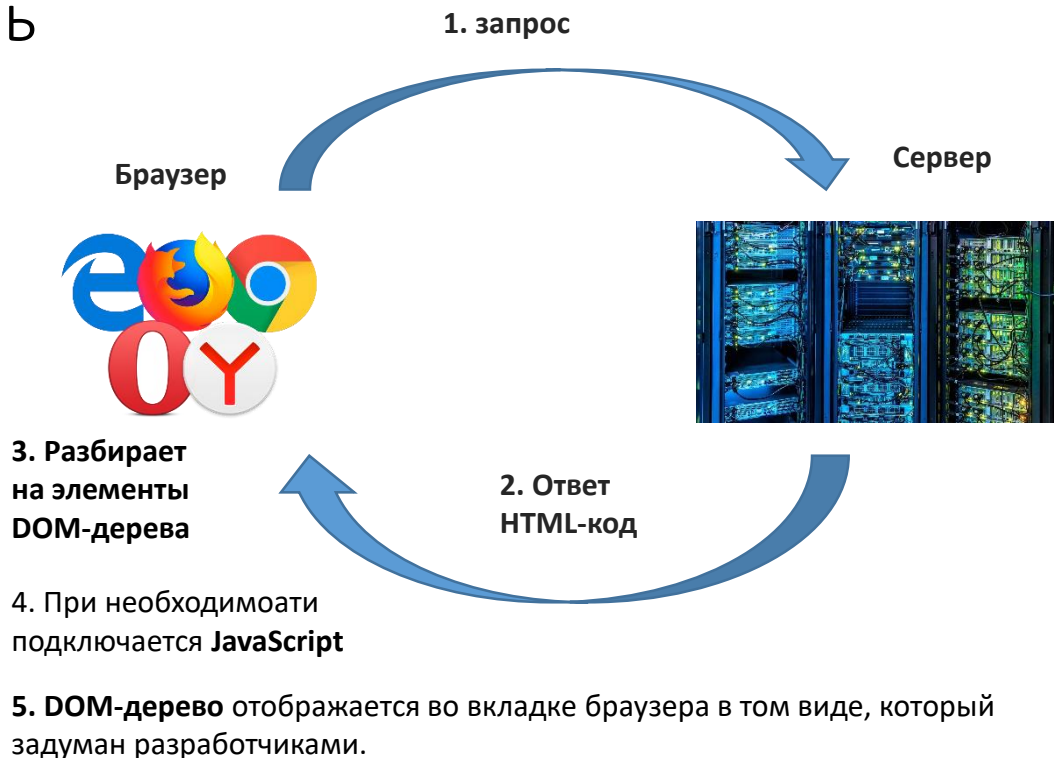
Открываемая в браузере конкретным пользователем **HTML-страница** является **результатом**:

1. Соединения разных документов (**HTML**, **CSS** и скриптов) в единую структуру;
2. Генерации тегов языками программирования и их фреймворками (**Python** и **Django**, **JavaScript** и **Angular**);
3. Обращения к данным из некоторой базы (реляционной или **NoSQL**);
4. Воздействия **JavaScript** и **AJAX-технологии** (позволяет динамически подгружать элементы сайта);
5. Работы браузера по построению дерева документа.

Последний пункт связан с моделью **DOM (Document Object Model)**, объектная модель документа).

DOM – модель документа как объекта, создаваемая веб-браузером в памяти вашего устройства на основании того **HTML-кода**, который был получен от сервера.

Сам документ в виде **HTML** браузер не показывает в исходном виде.



HTML-код, который пишут программисты – это всего лишь **текстовый файл** определенного формата, а **DOM** – **результат действий браузера**, который создает объекты при парсинге текстовых файлов.

Парсинг — это процесс автоматического сбора данных и их структурирования. Специальные программы или сервисы-парсеры «обходят» сайт и собирают данные, которые соответствуют заданному условию.

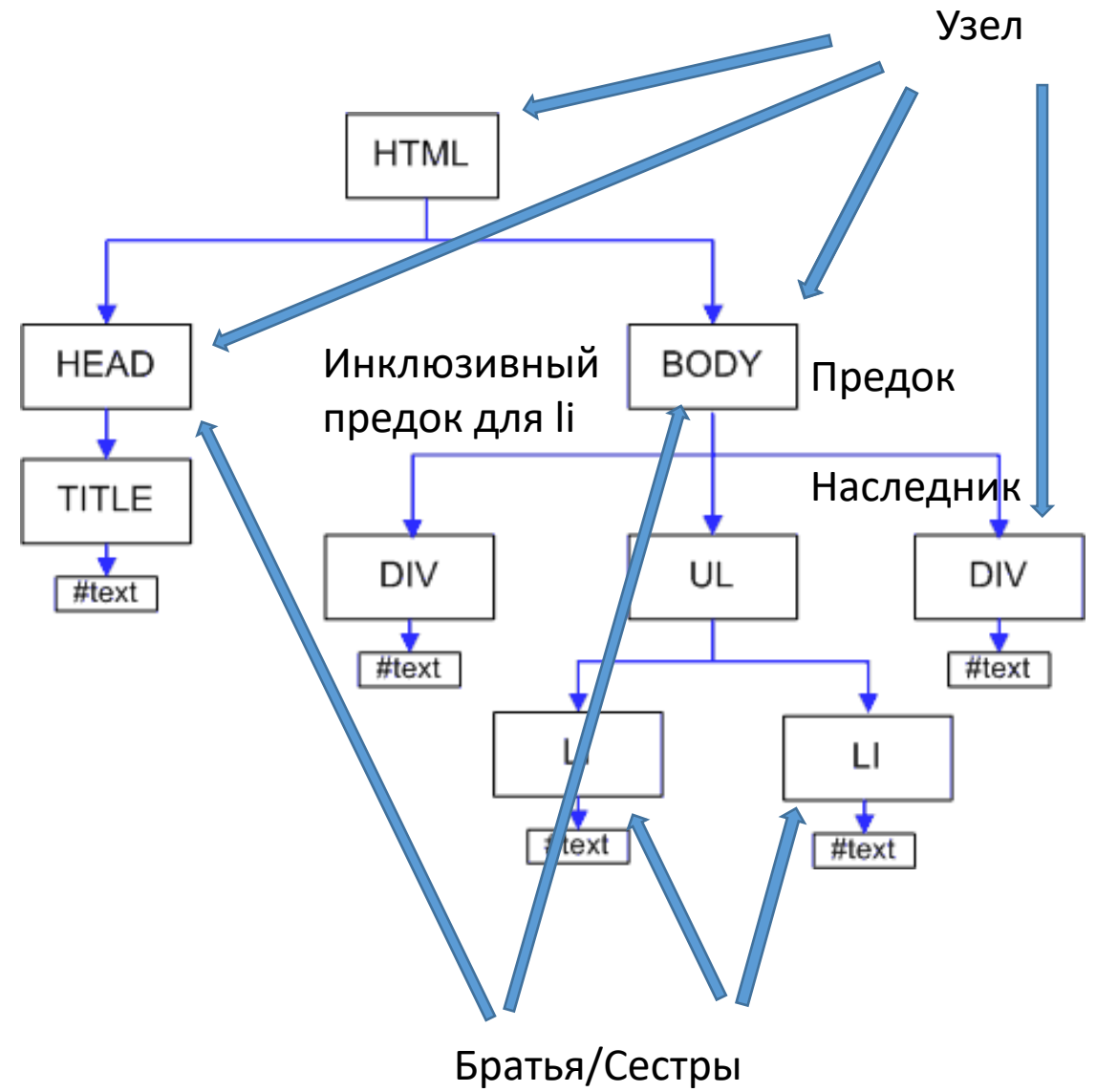
Основные понятия модели DOM

Благодаря **W3-консорциуму** выработан единый стандарт построения и разбора содержимого веб-страницы.

Актуальный **DOM-стандарт** официальный сайт: <https://dom.spec.whatwg.org/>

Основные структуры, используемые в объектной модели документа:

1. **Дерево** (веб-страница может быть представлена как иерархическое перевернутое дерево, начинающееся с главного элемента и расширяющееся к низу. У каждого объекта дерева есть родитель, который может быть пустым, т.е. **null**, и дети, если он находится не в самом низу);
2. **Наследник** (любой дочерний элемент узла дерева. Наследник может быть инклюзивным, т.е. не прямым. Пример: у тега `<html>` есть дочерний элемент `<body>`, внутри которого содержится элемент `<article>`, тогда тег `<article>` это не прямой наследник `<html>`);
3. **Предок** (если у объекта **DOM-дерева** есть потомки, то для них он является предком. Здесь также возможна инклюзивность. Из примера выше у тега `<article>` есть прямой предок – тег `<body>`, а также инклюзивный родитель – тег `<html>`);
4. **Братья / сестры** (в стандарте используется термин **sibling**, что с английского переводится как родной брат или сестра. Так объект подразумевает наличие у него «соседей», стоящих на том же уровне дерева и единого предка, не являющегося пустым);
5. **События** (к объектам дерева можно применять разные события при помощи **JavaScript**. Они могут срабатывать на действия пользователя или сетевую активность. Так, при наведении мыши, клике, нажатии клавиши или обрыве связи возможно разное поведение всего сайта или его элементов);
6. **Узел** (все элементы **DOM-дерева** являются узлами, **нодами**, **nodes**. Их можно создавать, изменять, вызывать свойства и методы с помощью **JS**).



Код HTML страницы

Заголовок

Тело

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
...
```

```
</head>
```

```
<body>
```

```
...
```

```
...
```

```
...
```

```
...
```

```
...
```

```
</body>
```

```
</html>
```

