**Из чего состоит сайт?**

**Дизайн сайта**

Это внешний вид и структура сайта: шрифты, цвета, кнопки, элементы меню, разделы и т.д. Его продумывают после того, как возникает идея сделать сайт. Дизайн есть у всех элементов сайта: от баннера или всплывающего меню на главной странице до кнопки перехода в чат в разделе «Контакты».



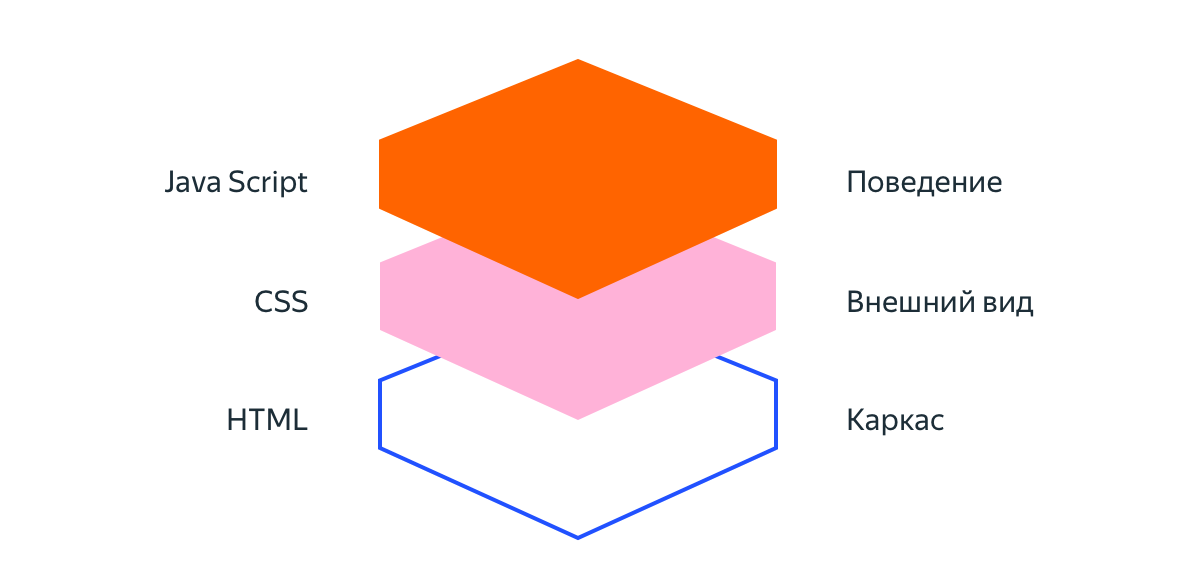
**Контент сайта**

Это любая информация, которую владелец сайта размещает на его страницах: текст, ссылки, фото, аудио и видеоматериалы. Контент сайта хранится на хостинге. Когда пользователь заходит на определённую страницу, программное обеспечение хостинга находит и загружает нужные файлы.

**Код сайта**

**Код** — это текст, в котором разработчики описывают дизайн и поведение сайта на понятном для компьютеров языке. Такие языки называют языками программирования. Языки бывают разные: с помощью одних описывают интерфейс, с помощью других — поведение сервера.

**Интерфейсный код (Front-end)** описывает дизайн. С его помощью компьютеры понимают внешний вид и расположение элементов на сайте. Чтобы написать интерфейсный код, программисты используют языки **HTML, CSS и Javascript**. Возможно, вам встречалось такое понятие как «вёрстка сайта». **Вёрстка сайта** — это процесс создания интерфейсного кода.



Треугольник интерактивной вёрстки состоит из HTML, CSS, JavaScript. Начинать нужно с HTML и CSS. Причём сразу с обоих.

Разбор конструкций языков HTML и CSS происходит по принципу «от общего к частному». Вы узнаете, что такое HTML-теги и атрибуты HTML, свойства и селекторы CSS.

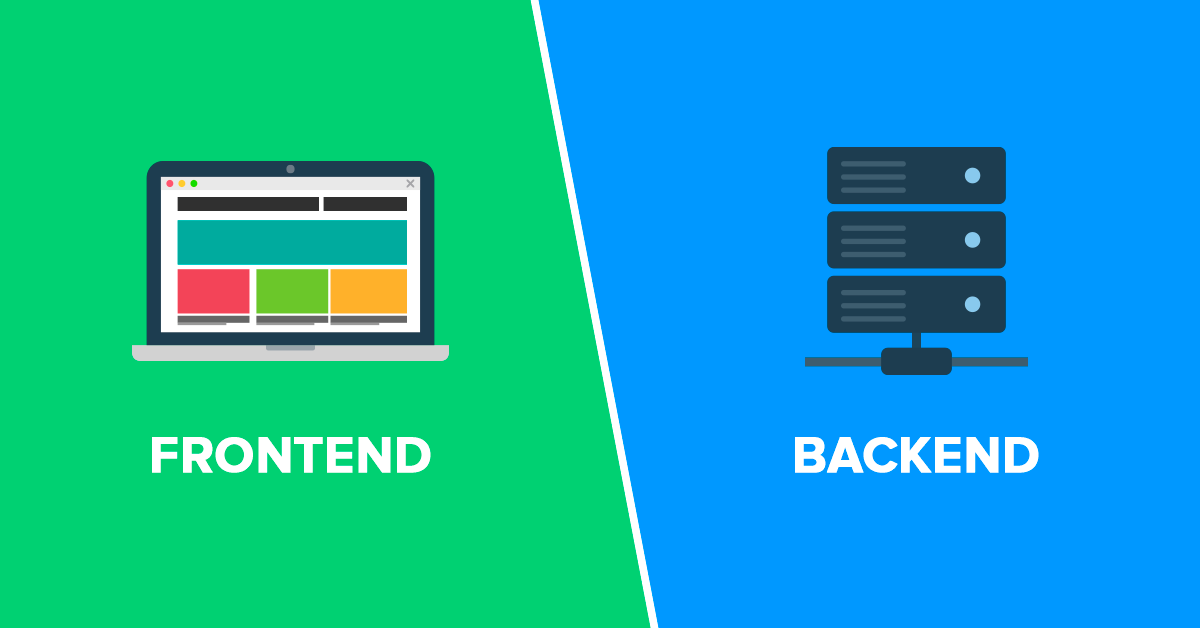


**Серверный код (Back-end)** описывает поведение сайта. С его помощью сервер определяет, что произойдёт, если пользователь нажмёт на кнопку или введёт текст. У одностраничных сайтов такого кода может и не быть. Серверный код нужен, если на сайте есть возможность создать аккаунт, купить товар или услугу. Серверный код пишут на **PHP, Java, Ruby** и т.д.

Для создания сайта нужно придумать дизайн, найти контент и написать код. Всё это можно заказать у веб-студии или сделать самому.

# Что такое Front-end и Back-end?

Вы наверняка уже слышали эти модные в сфере программирования слова «фронтенд» и «бэкенд», но что за ними стоит?



Давайте начнем с определений.

**Front-end** — все, что браузер может читать, выводить на экран и / или запускать. То есть это **HTML, CSS и JavaScript**.

**HTML (HyperText Markup Language)** говорит браузеру, каково содержание страницы, например, «заголовок», «параграф», «список», «элемент списка».

**CSS (Cascading Style Sheets)**говорит браузеру, как отображать элементы, например, «после первого параграфа отступ в 20 пикселей» или «весь текст должен быть темно-серым и написан шрифтом Verdana».

**JavaScript** говорит браузеру, как реагировать на некоторые взаимодействия, используя легкий язык программирования. Большинство сайтов на самом деле не используют много JavaScript, но если вы нажмете на что-то и содержимое страницы поменяется без белого мигания экрана, значит, где-то использовался JavaScript.



**Back-end** — все, что работает на сервере, то есть «не в браузере» или «на компьютере, подсоединенном к сети (обычно к Интернету), который отвечает на сообщения от других компьютеров».

Для бэкенда вы можете использовать любые инструменты, доступные на вашем сервере (который, по сути, является просто компьютером, настроенным для ответов на сообщения). Это означает, что вы можете использовать любой универсальный язык программирования: **Ruby, PHP, Python, Java, JavaScript / Node, bash**. Это также означает, что вы можете использовать системы управления базами данных, такие как **MySQL, PostgreSQL, MongoDB, Cassandra, Redis, Memcached**.

# ****HTML****

Каждый сайт — конструктор из элементов. За их создание отвечает язык **HTML**. Любая веб-страница содержит код на этом языке и сохраняется в **HTML-файле**. Именно его открывают и смотрят в браузере. HTML расшифровывается как **Hyper Text Markup Language** (англ. «язык гипертекстовой разметки»).

**Гипертекст**- это, вообще говоря, некоторый текст, который содержит ссылку на другой текст. В веб сфере эти ссылки являются ссылками на другие веб-документы.

Разметка состоит в том, что блоки текста заключают в управляющие конструкции — **теги** (от англ. tag, «бирка»). Такие «бирки» указывают браузеру, например, как отображать фразу, которая в них «обернута».  
  
Таким образом, **HTML-документ** - это веб-документ, который был сформирован при помощи языка разметки HTML. По своей сути - это текстовый файл, в котором был использован язык разметки. Исходный код HTML-документа может быть просмотрен и отредактирован в любом текстовом редакторе, например, в Блокноте. Это одно из удобных свойств HTML-документа. Программа, которая позволяет отображать HTML-документ, на основе исходного кода разметки называется браузером.

# Структура HTML документа

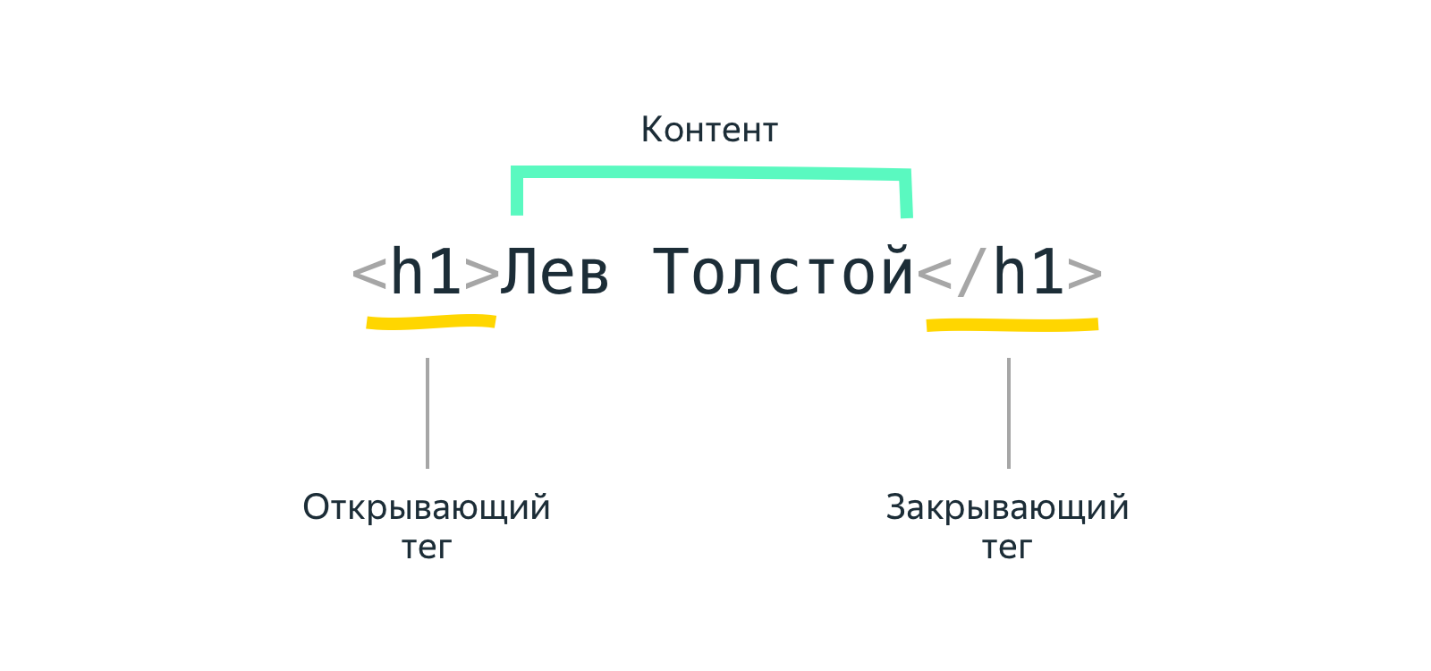
Язык HTML состоит из **тегов**. Каждый тег является элементом разметки гипертекста, и выполняет свою определенную функцию. Одни теги нужны для создания структуры html-документа, указания различной информации о веб-странице, другие – для форматирования и создания элементов на веб-странице, например, чтобы создать кнопку, необходим тег <button>Кнопка</button>, а чтобы перенести текст на новую строку – тег <br>.

Теги бывают **парными**, например тег <button>...</button>, и **одиночными**, например, тег <br>. **Парные теги** состоят из начального или открывающего тега, и конечного или закрывающего тега. Закрывающий тег записывается со слэшем.

# Как устроены HTML-элементы?

Обычно HTML-элемент состоит из **тегов** и размещённого между ними содержания — **«контента»**. У любого тега HTML есть имя и угловые скобки. Например, тег самого главного заголовка носит имя **h1** (от англ. heading 1, «заголовок-1»).

Теги расставляются как дорожные знаки с названием города. На въезде в город знак объявляет его имя, а на выезде перечеркнутое имя сообщает, что город закончился. Так и в начале HTML-элемента ставят **открывающий тег** с именем тега, а в конце — **закрывающий тег**, где имя будто перечеркнуто косой чертой. Созданный элемент называют по имени тега.



Ещё одна полезная вещь — это комментарии. Код, заключённый между символами <!-- и -->, работать не будет. Если эти символы удалить, то код заработает, это называется «раскомментировать». С помощью комментариев обычно временно отключают какой-то код или оставляют подсказки и разъяснения.

**Doctype в HTML, объявление типа документа**

В первой строке html-документа принято указывать тип документа. Делать это нужно обязательно, потому что существует несколько версий языка html, и у каждой есть свои правила и стандарты. Если браузеру не сообщить о версии HTML, то он не будет знать какому стандарту нужно следовать при отображении веб-страницы, что может привести к ее некорректному отображению.

Для старой версии HTML доктайп выглядел так:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"

"http://www.w3.org/TR/html4/strict.dtd">

А для современной версии HTML уже намного проще:

<!doctype html>

При вёрстке веб-страниц мы будем использовать пятую версию языка – **HTML5**. Для этой версии тип документа записывается таким образом: <!doctype html>.

# Основные теги HTML

Простейшая HTML-страница состоит как минимум из трёх тегов: <html>, <head> и <body>.

Тег <html> располагается в документе сразу после доктайпа и содержит все остальные теги, включая <head> и <body>. Тегу <html> обычно добавляют важный атрибут lang(сокращение от английского «language»), в котором задаётся язык документа. Наш сайт будет русскоязычным, поэтому в lang пропишем такое значение:

<html lang="ru">

Тег <head> хранит важную служебную информацию, а в теге <body> хранится содержание страницы, которое отображается в окне браузера. Все тексты и картинки мы будем добавлять внутрь <body>. Оба тега в документе могут быть в единственном числе.

Давайте начнём работу над прототипом главной страницы нашего сайта!

Атрибут lang можно задавать любым тегам, если нужно уточнить, на каком языке написан текст внутри тега. Но если задать его тегу <html>, то действие атрибута распространится на весь документ целиком

# Тег head, служебная информация о странице

Тег <head> предназначен для хранения служебной информации о странице. Он располагается первым в теге <html>, сразу перед <body>.

Внутри <head> обычно содержится заголовок, ключевые слова, описание страницы и другие служебные данные, а также подключаются стили и другие внешние ресурсы. Содержимое этого тега не отображается на странице напрямую.

Важный тег, который включается в <head> — тег <meta>. Он одиночный, то есть не требует парного закрывающего тега в конце.

С помощью <meta> можно сообщать браузеру, поисковому роботу или другому устройству различную служебную информацию (или метаинформацию) о вашем сайте: кодировку текста, описание контента и так далее. Для этого используются теги <meta> с разными атрибутами и их значениями. Вот некоторые из атрибутов: charset, content, http-equiv, name и scheme.

С помощью атрибута charset указывается кодировка текста HTML-страницы:

<meta charset="название кодировки">

Лучше всегда указывать кодировку явно. Если этого не делать, браузер может неправильно угадать её, и вместо текста будут отображаться «иероглифы».

Самая распространённая современная кодировка — utf-8. Используйте её во всех своих проектах. Раньше часто использовали кодировку windows-1251, стандартную кодировку для кириллицы в Windows. Но сейчас это считается плохой практикой.

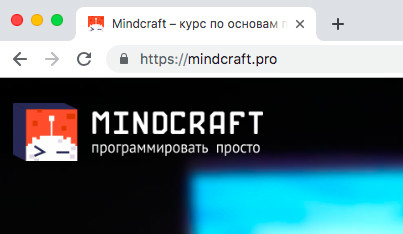
<head>

<meta charset="utf-8">

</head>

# Тег title, заголовок страницы

Ещё один элемент, который располагается в <head> — это тег <title>. В нём задаётся заголовок страницы, который отображается во вкладках браузера:



А вот пример использования <title> в коде:

<head>

<title>Mindcraft - курс по основам программирования</title>

</head>

Текст в <title> должен описывать содержимое документа в целом. По заголовку должно быть понятно, о чём эта страница, даже когда она не открыта в браузере, а отображается в результатах поиска или в браузерных закладках.

# Заголовки

Для создания основной структуры текста используют заголовки. В HTML существует целое семейство заголовочных тегов: от <h1> до <h6>. Тег <h1> обозначает самый важный заголовок (заголовок верхнего уровня), а тег <h6> обозначает подзаголовок самого нижнего уровня. Буква «h» в названии тега — это первая буква английского «heading».

На практике в текстах редко встречаются подзаголовки ниже третьего уровня. Поэтому чаще всего используются теги <h1>, <h2> и <h3>:

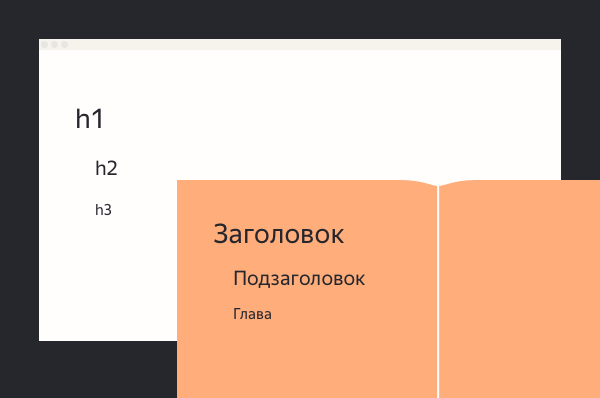
<h1>Спецификация HTML</h1>

<h2>Раздел 1 Введение</h2>

<h3>Раздел 1.1 Происхождение языка</h3>

Поисковые системы придают особое значение заголовкам, также правильно расставленные заголовки важны для доступности документа. Поэтому нужно учиться грамотно использовать заголовки.

Заголовок <h1> — самый важный на странице. В него нужно включать текст, который в целом описывает содержание страницы. Очень важно, чтобы заголовок первого уровня на странице был только один.



# Параграфы

Основную структуру текста создают с помощью заголовков, а более мелкую выстраивают с помощью параграфов (или абзацев). Для разметки параграфов предназначен тег <p>(от английского «paragraph»).

По умолчанию параграфы начинаются с новой строки и отделяются от остального контента отступами сверху и снизу. Так что, если нужно отделить один блок текста от другого, верным решением будет заключить их в теги <p>.

<p>Абзац про то, как я решил стать верстальщиком</p>

<p>Абзац про моего инструктора</p>

<p>Абзац про мой дневничок</p>

# Вложенные теги

Теги могут вкладываться друг в друга. У вложенных тегов всегда нужно следить за правильным порядком закрытия. Вложенный тег не может закрываться позже родительского:

<p><a href="http://vk.com">Ссылка</p></a> <!-- Плохо -->

<p><a href="http://vk.com">Ссылка</a></p> <!-- Хорошо -->

И ещё, не все теги можно вкладывать в другие теги, например уже знакомый вам тег <h1>нельзя вкладывать в <p>. Эти правила вложенности для каждого тега вы узнаете постепенно по ходу изучения HTML.