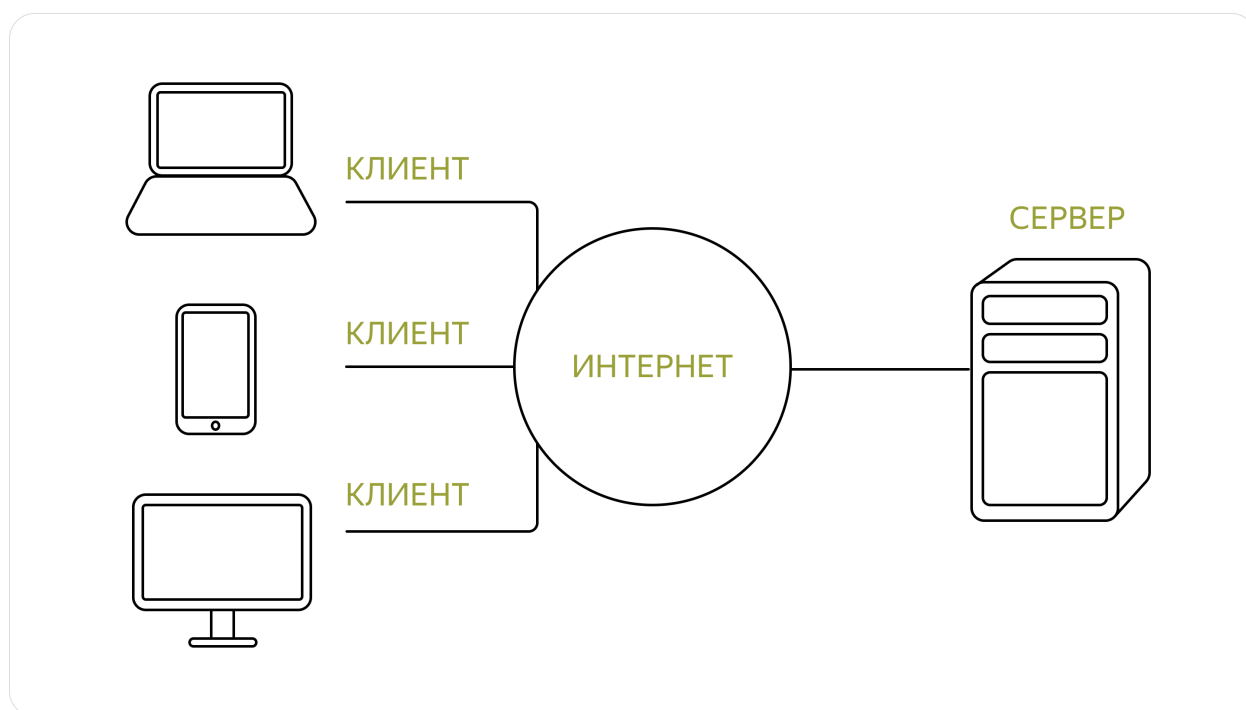


Шпаргалка: архитектура веб-приложений

Клиент-серверная архитектура

Архитектура — это способ организации работы приложения.

Самая распространённая архитектура веб-приложений — **клиент-серверная**.



Клиент — это система, которая связывается с сервером и запрашивает нужную пользователю информацию.

Сервер — система, которая обрабатывает запросы клиента и формирует ответ.

Интернет, или **сеть** — система связанных между собой устройств, которая помогает клиенту и серверу обмениваться данными.

Веб-приложение

Браузер отправляет запрос на сервер, получает ответ, обрабатывает информацию и показывает результат на странице веб-приложения.

Фронтенд (frontend) — видимая часть приложения. С ней пользователь взаимодействует. Например, нажимает кнопки или вводит текст. В модели «клиент-сервер» фронтенд — это код, который обрабатывается на стороне клиента.

Бэкенд (backend) — скрытая часть приложения. Она отвечает за вычисления, логику, хранение данных. В модели «клиент-сервер» бэкенд — это код, который работает на удалённом сервере.

URL

URL расшифровывается как Uniform Resource Locator — «унифицированный указатель ресурса». Это адрес веб-ресурса. URL указывает браузеру, где находится веб-приложение, веб-страница или фрагмент веб-страницы и как к ним обратиться.

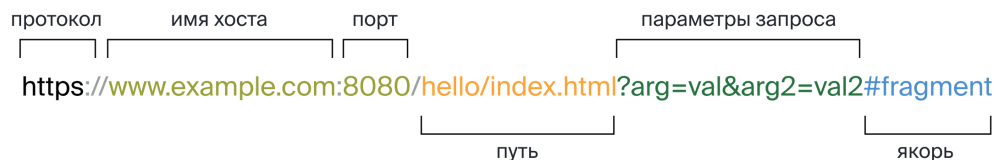


Схема (scheme) — протокол, по которому передаются данные. Например, HTTP и HTTPS.

Имя хоста (hostname) — доменное имя или IP-адрес сервера, к которому обращается пользователь. Например, `music.yandex.ru:443`.

Доменное имя — это адрес веб-приложения, под которым оно зарегистрировано в интернете. Доменное имя состоит из символов, которые присвоили приложению в системе доменных имён (DNS).

Когда ты вводишь URL в адресной строке, браузер инициирует запрос к **DNS**-серверу (Domain Name System — «система доменных имён»). DNS-сервер возвращает браузеру IP-адрес.

IP-адрес — это уникальный идентификатор сервера, на котором находится нужная информация. Например, 198.221.33.12.

Порт (port) — вторая часть пары хост:порт. Это виртуальный номер порта для подключения к серверу. Например, 443 в music.yandex.ru:443.

Путь (path) — месторасположение ресурса. Например, [/home/index.html](#) или [/search/](#).

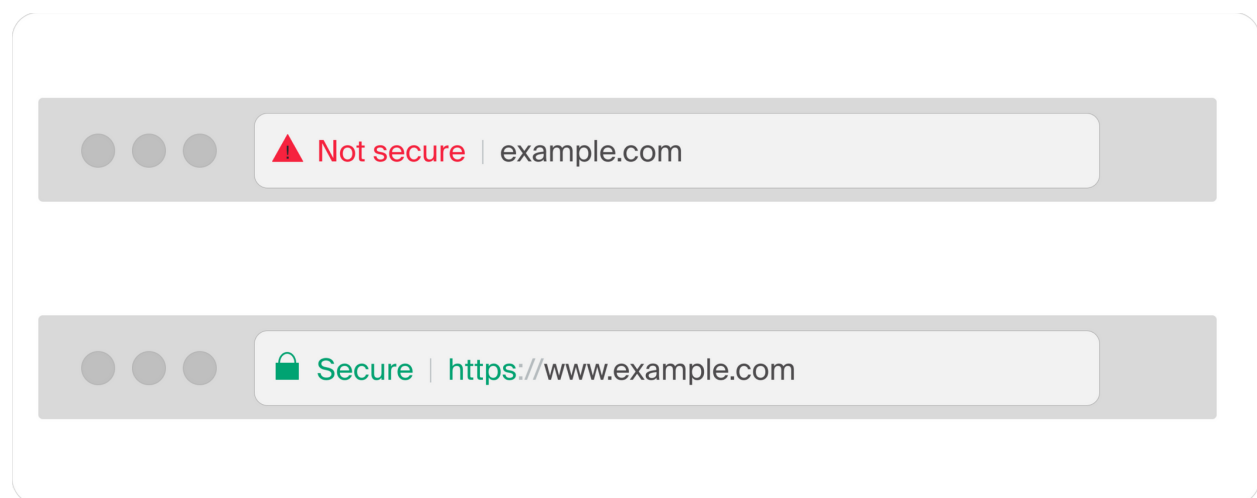
Параметры запроса (query) — дополнительные параметры вида `ключ=значение`. Перечисляются после знака `?` и разделяются через `&`. Например, поисковый запрос <https://yandex.ru/search/?text=практикум>.

Фрагмент текста (fragment), или **якорь** (anchor) — дополнительный указатель. Он позволяет сразу попасть в нужную часть веб-страницы. Например, к заголовку или абзацу в тексте. Вот так: <https://developers.google.com/web/tools/chrome-devtools/#application>.

Протоколы HTTP и HTTPS

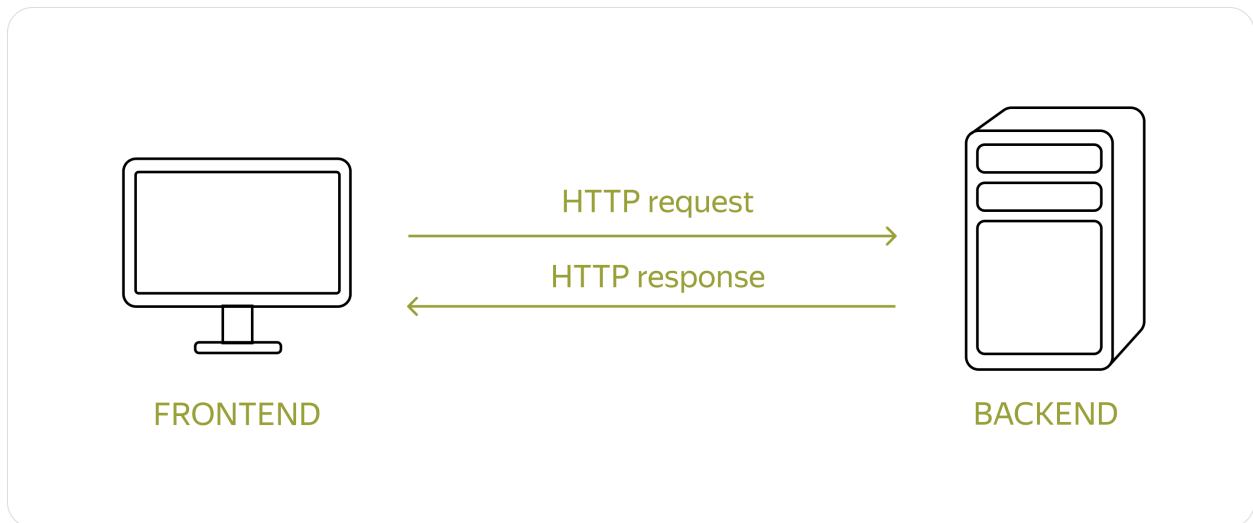
Протокол передачи данных — набор правил, по которым устройства обмениваются информацией. Один из таких протоколов — **HTTP** (HyperText Transfer Protocol — «протокол передачи гипертекста»).

Чтобы обеспечить безопасность, применяют протокол HTTP с расширением защиты — **HTTPS** (HyperText Transfer Protocol Secure — «протокол защищённой передачи гипертекста»).



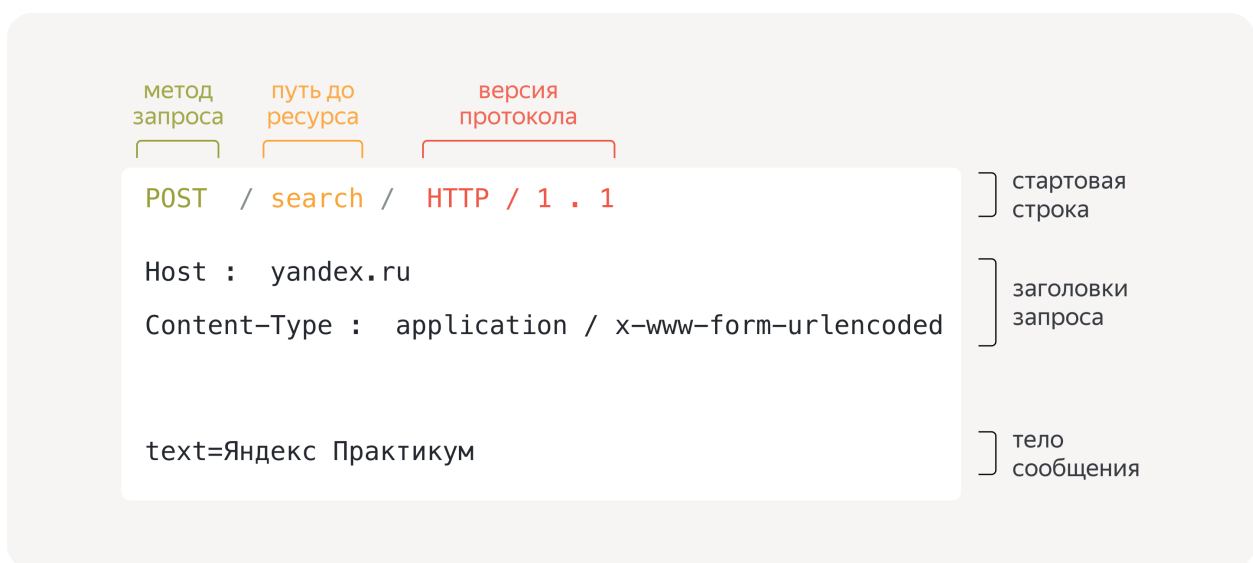
Структура HTTP-запроса

По протоколу HTTP фронтенд передаёт запрос (request), а бэкенд — ответ (response).



Запрос состоит из трёх блоков:

- стартовая строка,
- заголовки,
- тело сообщения.



Стартовая строка (start line) состоит из трёх элементов: метода, пути до ресурса и версии протокола. Самые распространённые методы — GET, POST, PUT, DELETE.

Заголовки запроса (Request Headers) — дополнительная информация от фронтенда бэкенду.

Тело сообщения — это данные, которые передаёт фронтенд.

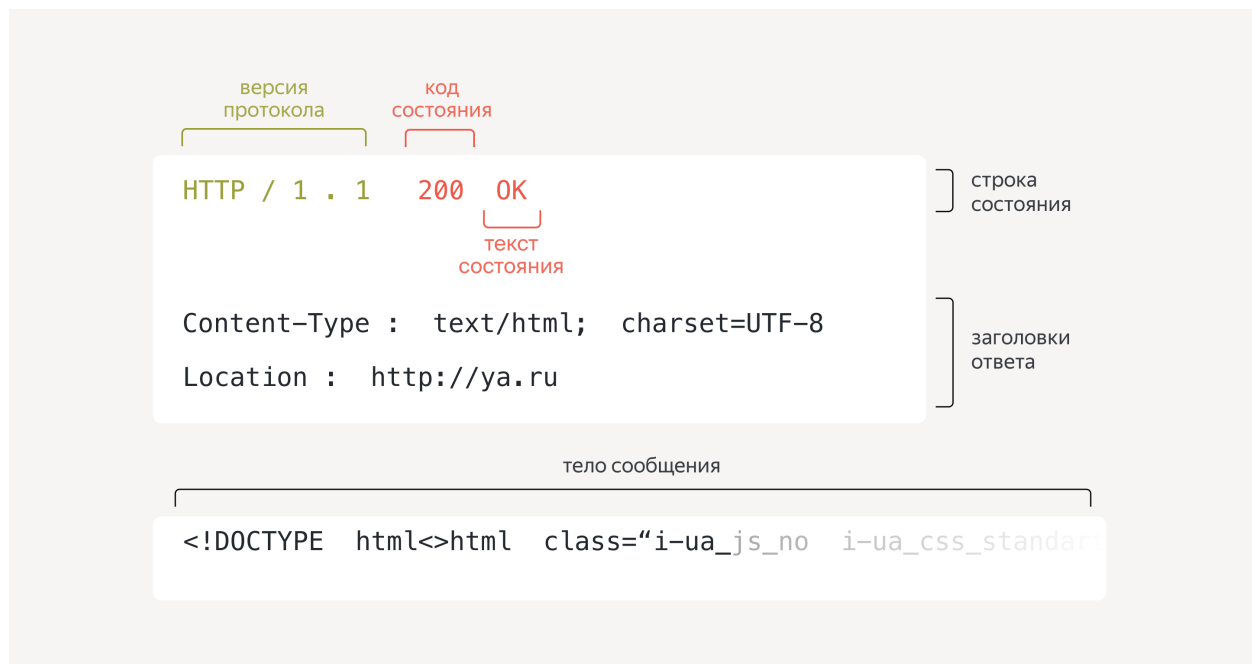
В GET-запросе тело отсутствует.



Структура HTTP-ответа

HTTP-ответ включает:

- строку состояния,
- заголовки,
- тело сообщения.



Строка состояния включает версию протокола, код состояния, текст состояния.

Самые распространённые ошибки в коде состояний:

- 403 Forbidden — у клиента недостаточно прав, чтобы получить доступ к данному ресурсу.
- 404 Not Found — указанная в URL страница не найдена.
- 500 Internal Server Error — внутренняя ошибка сервера.
- 503 Service Unavailable — сервис временно недоступен, сервер не может выполнять запросы.
- 504 Gateway Timeout — время ожидания ответа от сервера истекло.

Заголовки ответа (Response Headers) — дополнительная информация о бэкенде и типе данных для фронтенда.

В теле сообщения бэкенд отдаёт результат выполнения запроса.

Веб-страница: HTML

HTML — это HyperText Markup Language, то есть «язык гипертекстовой разметки». На этом языке создают структуру страницы, которую считывает браузер.

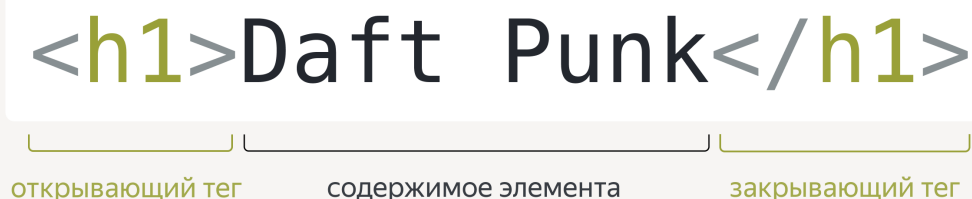
Разметка указывает браузеру, в каком формате отобразить данные. Также она может содержать дополнительную информацию — например пути к HTML-, CSS- и JS-файлам.

```
<h1>Пропала собака</h1>
<h2>6 месяцев</h2>
<b>Нашедшим просьба позвонить по телефону +123456789</b>

```

HTML: теги и атрибуты

У всех HTML-элементов есть имена, а у большинства — открывающий и закрывающий теги. Например, тег самого крупного на странице заголовка — это `h1` (heading 1, «заголовок первого уровня»).



`<h1>Daft Punk</h1>`

открывающий тег содержимое элемента закрывающий тег

В HTML-элементе могут быть **атрибуты**, которые сообщают о нём дополнительные сведения.

Например, атрибут с именем `href` позволяет присвоить тексту адрес ссылки, по которой можно перейти.

```
<a href="https://ikea.com">Текст ссылки</a>
```

имя
атрибута

значение
атрибута

Атрибут встраивают только в открывающий тег. Если атрибутов несколько, их пишут через пробел. Порядок значения не имеет:

```
<a href="http://info.cern.ch" target="_blank">Сайт о первом веб-сайте</a>  
<!-- то же самое -->  
<a target="_blank" href="http://info.cern.ch">Сайт о первом веб-сайте</a>
```

HTML-документы построены по определённой структуре.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

Информация
о странице

Кодировка
Название страницы
Прочие метаданные
Подключение файлов,
шрифтов

Видимое
содержание

Элементы веб-страницы

В HTML можно вкладывать один элемент в другой. Элемент, который лежит внутри, называется **дочерним**. Тот, в который вложен дочерний, — **родительским**.

У элемента на странице может быть **идентификатор**. Разработчики присваивают его атрибуту `id`. Например: `<h1 id="part-8">HTML: теги и атрибуты</h1>`. Здесь `part-8` — это идентификатор элемента `HTML: теги и атрибуты`.

Веб-страница: CSS

Язык **CSS** (Cascading Style Sheets — «каскадные таблицы стилей») задаёт стиль оформления HTML-элементов.

Есть два способа подключить стили:

1. Внутри элемента `<head></head>` создают элемент `style`. Пространство между тегами `<style></style>` — это зона языка CSS.

```
<head>
  <style rel="stylesheet">
    h1 {
      color: red;
    }
  </style>
</head>
```

2. Подключают стили через отдельный файл с расширением `.css`. Когда стили выносят в отдельный CSS-файл, HTML-страница запрашивает его через тег `<link>`:

```
<head>
  <link rel="stylesheet" href="./style.css">
</head>
```

В CSS есть правила стиля. Правило состоит из селектора и свойств со значениями.

Вот код, который задаёт всем заголовкам с именем `h2` голубой цвет и размер шрифта 32px (пикселя). В этом случае селектор — имя тега.



Веб-страница: JavaScript

Логику работы приложения пишут на языке программирования **JavaScript** (JS). На нём программируют веб- и мобильные приложения, а часто и бэкенд.

JavaScript-код нужно подключать к HTML-странице так же, как и стили.

Первый способ: написать весь JavaScript-код в теле HTML-документа между тегами `<script>` и `</script>`.

```
<head>
  <script>
    <!-- Здесь можно расположить JS-код -->
  </script>
</head>
```

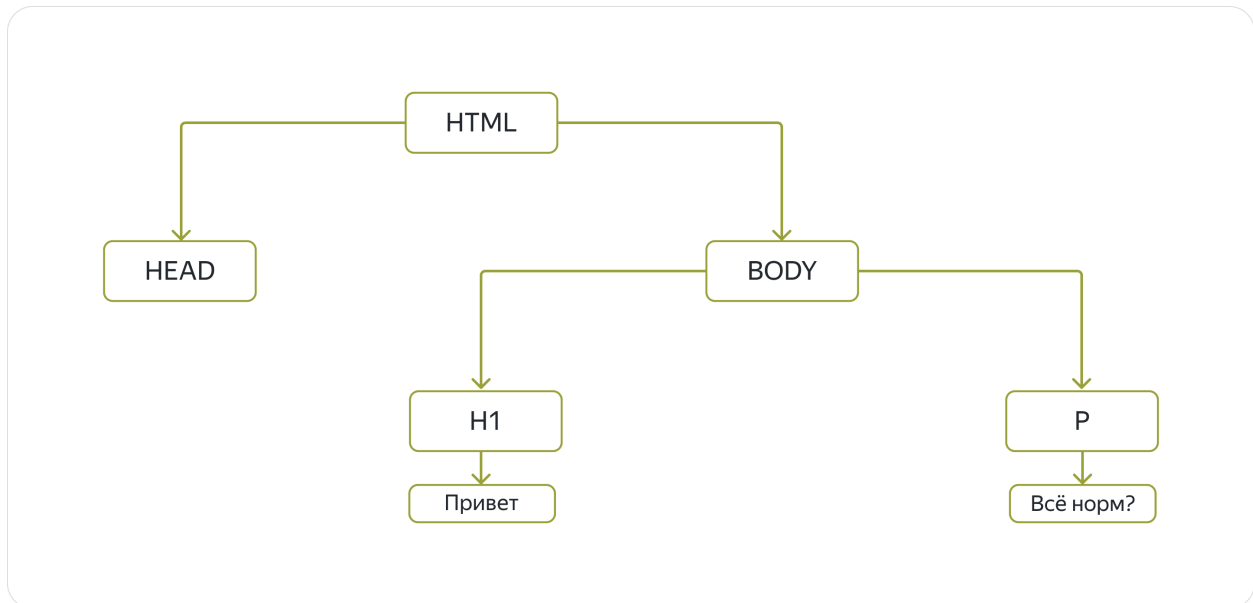
Второй способ: описать весь JS-код в отдельном файле, а в HTML-документе сослаться на этот файл.

```
<script src="./app.js"></script>
```

DOM

DOM (Document Object Model) — объектная модель страницы. Фактически это и есть HTML-разметка: блоки, из которых состоит документ. Браузер превращает

HTML-файл в DOM-дерево и на его основе отрисовывает страницу.

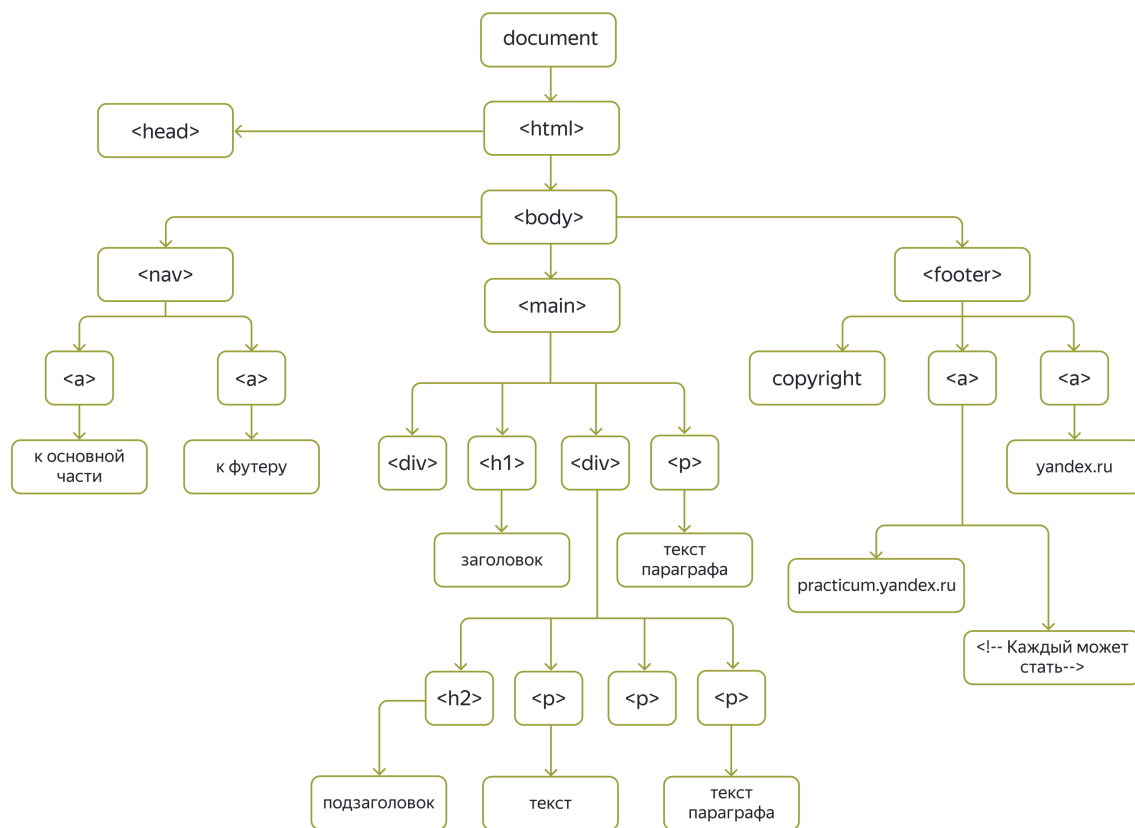


```
<!doctype html>
<html lang="ru">
  <head></head>
  <body>
    <h1>Привет</h1>
    <p style="color: red">Всё норм?</p>
  </body>
</html>
```

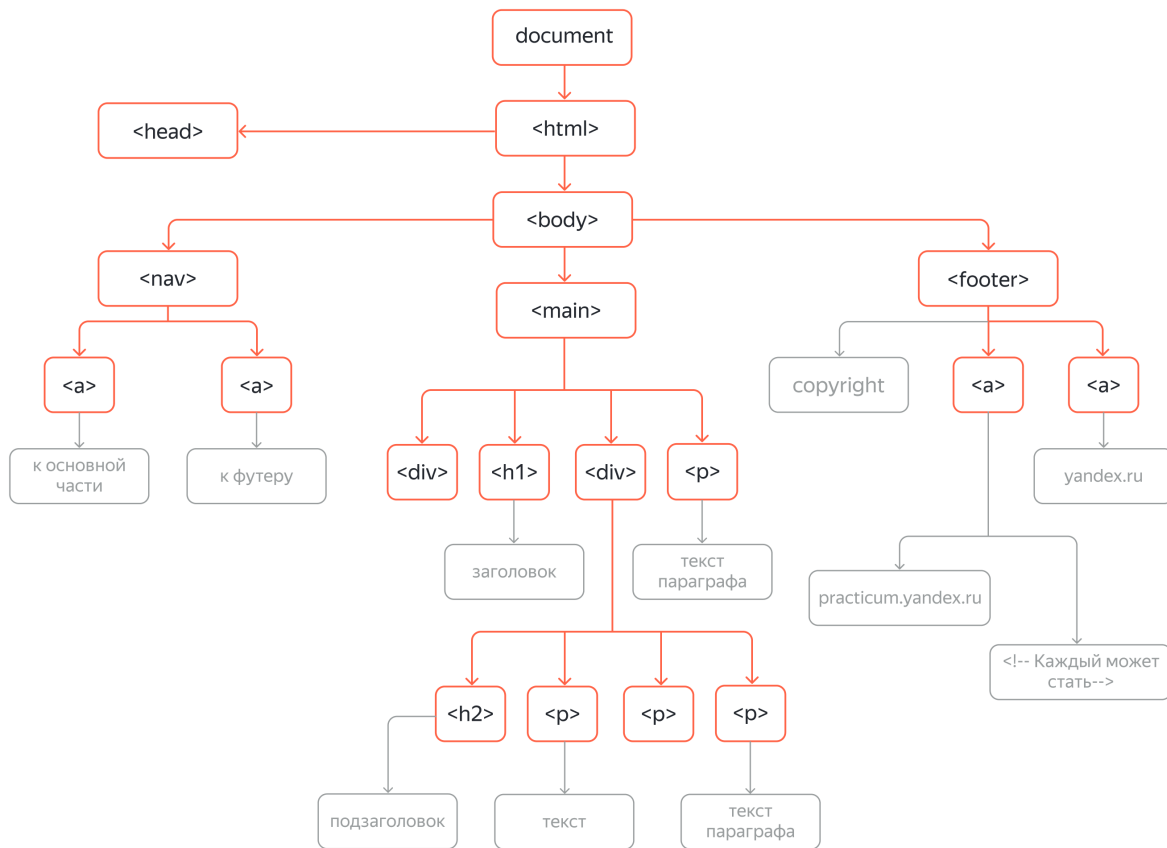
DOM-дерево состоит из **узлов**. Он возникает там, где есть:

- открывающий тег,
- текст,
- HTML-комментарий.

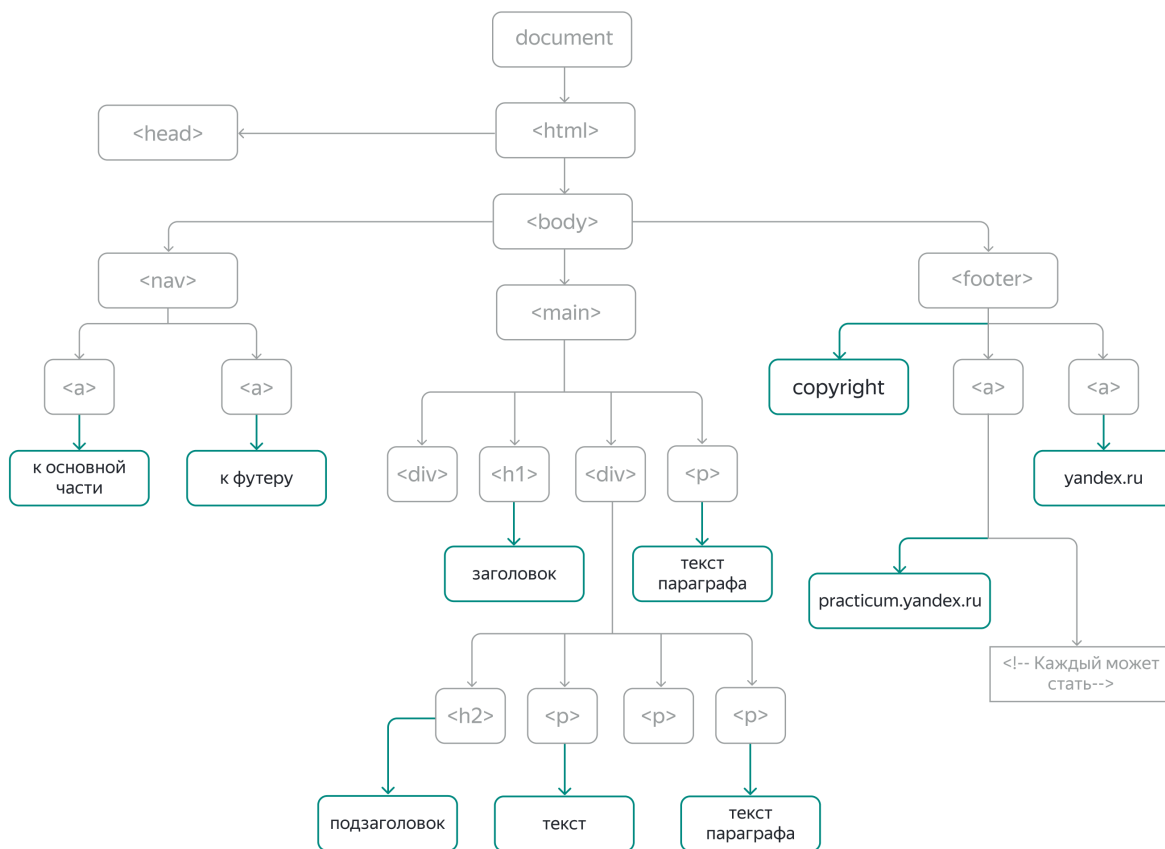
Узлы



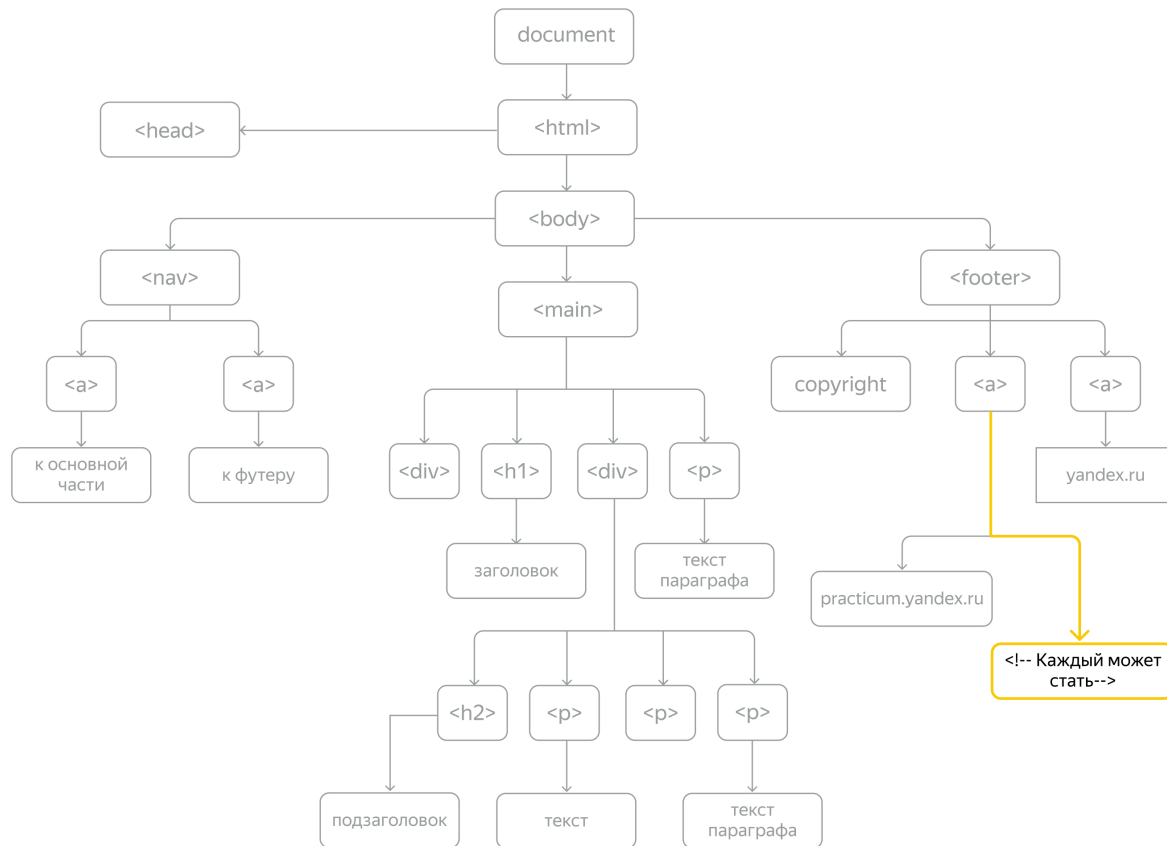
Элементы



Текстовые узлы



Узлы-комментарии



Элементы — то, что в разметке называется тегами. Они могут быть парными или одиночными.

Узлы — все сущности, которые можно увидеть на странице, в том числе и элементы. Текст, комментарии, теги — это узлы.