

HTML

HTML (HyperText Markup Language) — стандартизованный язык разметки, который позволяет создавать и структурировать веб-страницы.

Теги — это единицы разметки, которые сообщают браузеру, как отображать содержимое документа

```
<h1>HTML заголовок H1</h1>  
<h2>HTML заголовок H2</h2>  
<h3>HTML заголовок H3</h3>  
<h4>HTML заголовок H4</h4>  
<h5>HTML заголовок H5</h5>  
<h6>HTML заголовок H6</h6>
```

Html заголовок H1

Html заголовок H2

Html заголовок H3

Html заголовок H4

Html заголовок H5

Html заголовок H6

Самая простая страница

```
<!DOCTYPE html>
```

```
<!-- это «голова» страницы -->
```

```
<head>
```

```
</head>
```

```
<!-- это «тело» страницы -->
```

```
<body>
```

```
</body>
```

```
</html>
```

Основные теги

`<p>`, `
` — для абзаца и переноса на другую строку.

`<h1>`, `<p>`, `<div>`, `` и другие — для организации текста и мультимедийных элементов.

``, `<i>`, `<u>`, ``, `` и так далее — для форматирования текста.

``, `` и `` — для создания маркированных и нумерованных списков.

`<a>` — для создания ссылки на определённый раздел страницы или другой веб-документ.

``, `<video>`, `<audio>` и так далее — для вставки изображений, видео, аудио и других мультимедийных элементов.

`<button>`, `<form>`, `<input>`, `<textarea>` — для создания кнопок и форм для ввода данных и так далее.

Парные и одиночные теги

`<p>Текст абзаца</p>`

Есть одиночные теги, которые не требуют закрытия, например `
` для переноса строки или `` для вставки изображения.

Атрибуты тегов

Атрибуты тегов предоставляют дополнительную информацию о том, как элемент должен быть обработан или отображён

`` может включать такие атрибуты:

- 'src' (источник изображения)
- 'alt' (альтернативный текст)
- 'width' и 'height' (ширина и высота)

Раздел Head

Раздел <head> содержит метаданные и другую служебную информацию, которая не отображается напрямую на веб-странице, но играет важную роль в её работе и отображении

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

Назначение: Указывает кодировку символов документа.

Если не указать кодировку, браузер может неправильно отобразить текст (например, вместо букв появятся "кракозябры").

Пример проблемы: Без UTF-8 текст "Привет, мир!" может отобразиться как "ÐŸÑ€Ð, Ð²Ð¼Ñ,, Ð¼Ð,Ñ€!".

Настройка адаптивного дизайна для мобильных устройств

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

Назначение:

Без этого тега мобильные устройства могут неправильно масштабировать страницу, что сделает её неудобной для использования.

Атрибуты:

- `width=device-width` — ширина страницы адаптируется под ширину экрана устройства.
- `initial-scale=1.0` — страница отображается без масштабирования (масштаб 100%).

Пример проблемы: Без этого тега страница может отображаться слишком мелко на мобильных устройствах, и пользователю придется увеличивать её вручную.

Пробуем разные теги

`<label for="email">` — это HTML-элемент, который используется для создания текстовой метки, связанной с определенным элементом формы (например, полем ввода). Атрибут `for` указывает, с каким элементом формы связана метка, путем указания `id` этого элемента.

1.Связь метки и поля ввода: Атрибут `for` в `<label>` должен совпадать с атрибутом `id` у элемента формы (в данном случае `<input>`). Это связывает метку с полем ввода.

2.Удобство для пользователей: Когда пользователь кликает по тексту метки, фокус автоматически переходит на связанное поле ввода. Это улучшает доступность и удобство использования формы.

 (List Item)

Назначение: Элемент списка.

Использование: используется внутри упорядоченного списка () или неупорядоченного списка (). В данном случае это часть неупорядоченного списка.

Атрибут type

Назначение: Определяет тип поля ввода.

Значение: В данном случае `type="text"` указывает, что это текстовое поле, в которое пользователь может вводить текст.

Примеры других типов:

- `type="email"` — для ввода email.
- `type="password"` — для ввода пароля (текст скрывается символами).
- `type="number"` — для ввода чисел.
- `type="date"` — для выбора даты.

Атрибут id (связан с for)

`id="name"`

Назначение: Уникальный идентификатор элемента на странице.

Зачем нужен:

Связь с `<label>`: Атрибут `for` в `<label>` использует `id`, чтобы связать метку с полем ввода. Например:

Использование в JavaScript и CSS: `id` позволяет обращаться к элементу через JavaScript (`document.getElementById("name")`) или стилизовать его в CSS (`#name { ... }`).

Атрибут name

Назначение: Имя поля, которое используется при отправке данных формы на сервер.

Зачем нужен:

Когда форма отправляется, данные передаются в виде пар ключ=значение. Атрибут name задает ключ (в данном случае name), а значение — это то, что ввел пользователь.

div и span

`<div>` и `` — это два основных HTML-элемента, которые используются для группировки и стилизации содержимого.

`<div>` (блочный элемент)

Назначение: Используется для создания блочных контейнеров, которые занимают всю доступную ширину родительского элемента.

Поведение:

- Начинается с новой строки.
- Занимает всю доступную ширину (если не заданы другие стили).
- Может содержать другие блочные и строчные элементы.

```
<div style="background-color: lightblue;">  
  <p>Это параграф внутри div.</p>  
  <p>Это еще один параграф.</p>  
</div>
```

div и span

`` (строчный элемент)

Назначение: Используется для выделения части текста или других строчных элементов.

Поведение:

- Не начинается с новой строки.
- Занимает только столько места, сколько нужно для содержимого.
- Может содержать только текст и другие строчные элементы.

`<p>Это важный текст.</p>`

div и span

Основные отличия:

Характеристика	<code><div></code>	<code></code>
Тип элемента	Блочный (block-level)	Строчный (inline-level)
Начало новой строки	Всегда начинается с новой строки	Не начинается с новой строки
Ширина	Занимает всю доступную ширину	Занимает только место под текст
Использование	Для группировки блоков контента	Для выделения части текста
Пример	Контейнеры, секции, макеты	Выделение текста, стилизация

br

`
` — это HTML-элемент, который используется для переноса строки в тексте. Он не имеет закрывающего тега (является самозакрывающимся) и просто добавляет разрыв строки в том месте, где он указан

Как работает `
`:

Когда браузер встречает `
`, он переносит текст на новую строку.

Это полезно, когда нужно вручную управлять разрывами строк, например, в стихах, адресах или других случаях, где важно сохранить форматирование.

Семантический HTML

Семантический HTML — это HTML-теги, которые чётко описывают смысл содержимого, то есть его семантику

Вот основные примеры семантических HTML-тегов:

`<header>` — заголовок страницы или раздела.

`<nav>` — навигационное меню.

`<main>` — основное содержание документа.

`<article>` — независимый, автономный контент, который может быть отдельно распространён или использован.

`<section>` — раздел, группирующий связанный контент.

`<aside>` — контент, косвенно связанный с основным содержанием (например, боковая панель).

`<footer>` — нижний колонтитул страницы или раздела.

Таблицы в HTML

Для этого используют теги <table>, <tr>, <td>, <th>, <tbody>, <thead>, <tfoot>