

WEB DEVELOPMENT

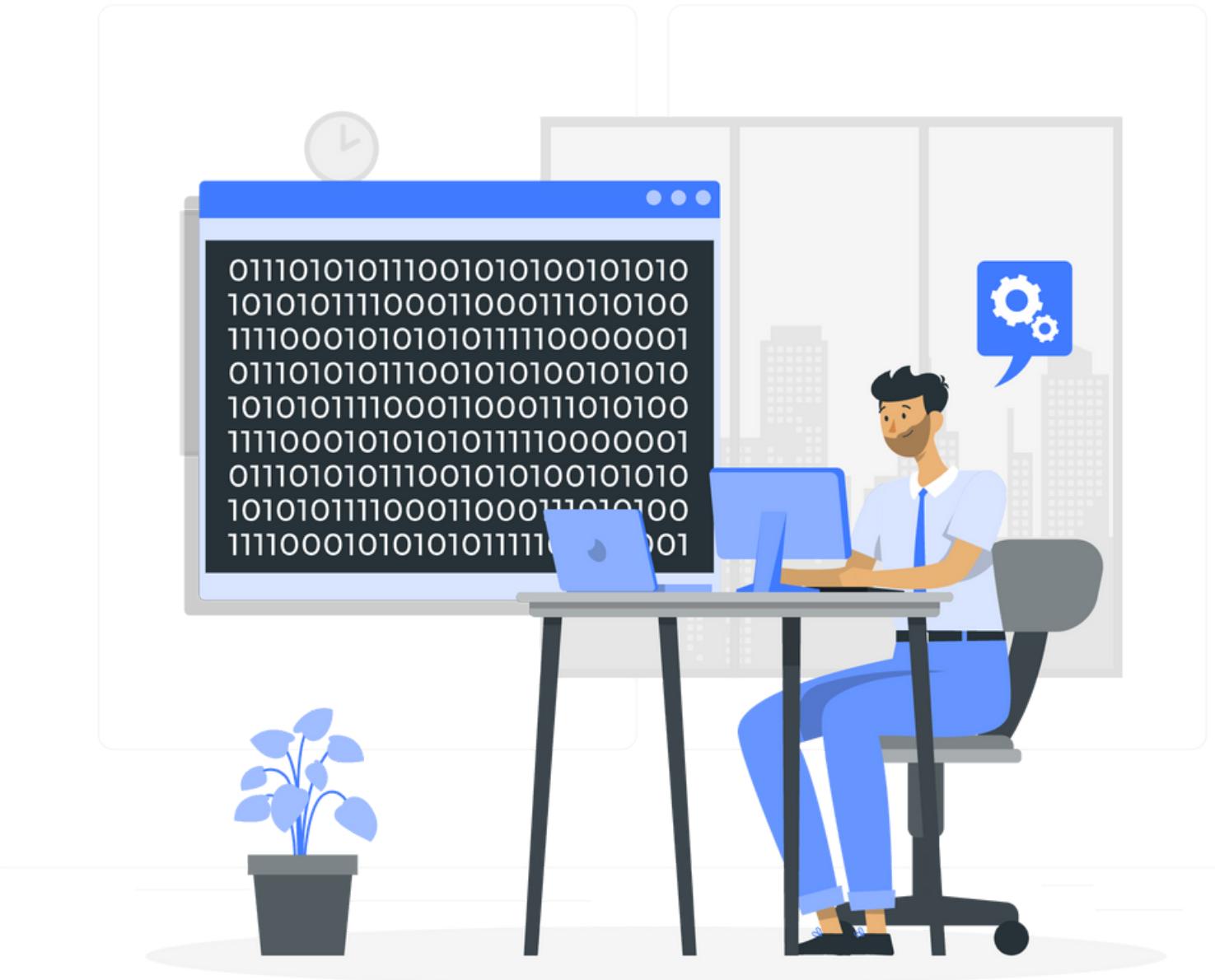
HTML, CSS, JavaScript



- 1** Ключевые элементы для создания пользовательского интерфеса веб приложений
- 2** Основы HTML
- 3** Структура кода
- 4** Основные теги
- 5** Изображения, атрибуты, списки

FRONTEND

Фронтенд относится к той части веб-разработки, которая занимается созданием пользовательского интерфейса (UI) -*user interface*, веб-приложений или веб-сайтов. Он отвечает за то, как пользователи взаимодействуют с веб-сайтом и как они видят и взаимодействуют с его содержимым.



ОСНОВНЫЕ ТЕХНОЛОГИИ, ИСПОЛЬЗУЕМЫЕ ВО ФРОНТЕНД РАЗРАБОТКЕ:

HTML **(HyperText Markup Language):**

Это основной язык разметки, используемый для создания структуры веб-страницы. HTML определяет различные элементы, такие как заголовки, параграфы, списки, изображения и ссылки.

CSS **(Cascading Style Sheets):**

CSS отвечает за внешний вид веб-страницы. Он определяет стили, цвета, шрифты, расположение элементов и другие аспекты дизайна.

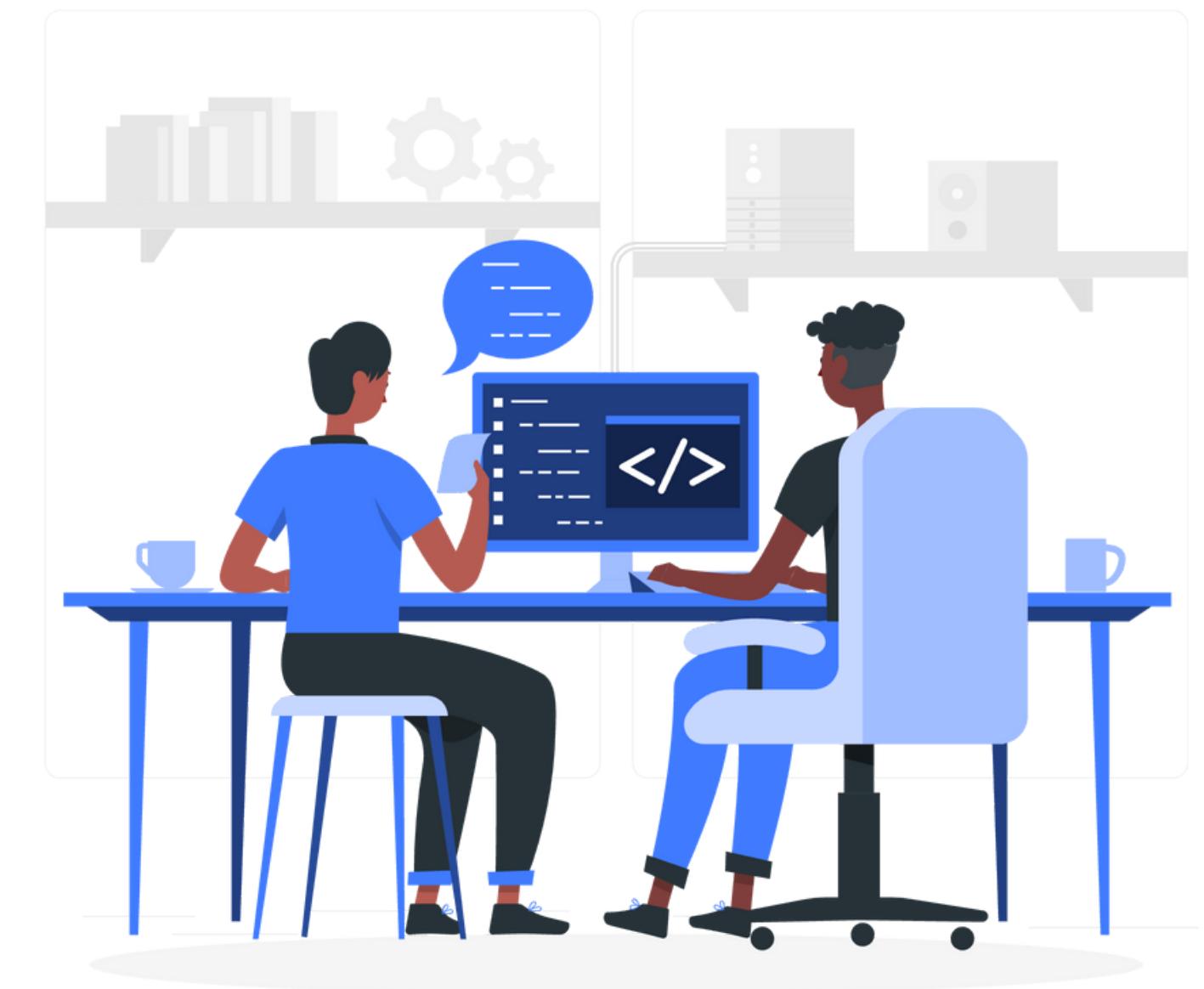
JavaScript:

JavaScript является языком программирования, который добавляет динамическое поведение на веб-страницы. Он позволяет создавать интерактивные элементы, обрабатывать события, взаимодействовать с сервером и многое другое.

HTML

Основная цель HTML - предоставить структурированный способ представления информации на веб-страницах и обеспечить совместимость между различными веб-браузерами.

HTML работает в сочетании с другими технологиями, такими как CSS (Cascading Style Sheets) для определения внешнего вида веб-страниц и JavaScript для добавления динамического поведения и интерактивности на страницы.



HTML позволяет создавать информационную и интерактивную среду, включая текстовое содержимое, гиперссылки, мультимедиа элементы, формы для ввода данных и многое другое.

С помощью **HTML** вы можете структурировать содержимое вашей веб-страницы, определить заголовки, абзацы, таблицы и создавать ссылки на другие веб-страницы или внешние ресурсы.



HTML документ состоит из HTML тегов. Каждый HTML тег заключен в угловые скобки (< и >). HTML теги обычно идут парами: открывающий тег <tag> и закрывающий тег </tag>.

В приведенных примерах <p>, <h1> и <div> - это открывающиеся теги, а </p>, </h1> и </div> - соответствующие закрывающиеся теги.

Эти теги используются для включения различных элементов веб-страницы без необходимости указывать закрывающий тег. Вместо этого они заканчиваются символом /. Примеры таких тегов:

```
<> index.html > ⚒ input
1  <br/>
2  
3  <input type="text" name="username"/>
```

```
<> index.html > ⚒ div
1  <p>Это абзац.</p>
2  <h1>Заголовок</h1>
3  <div>Контейнер</div>
```

index.html > html

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Моя первая веб-страница</title>
5  </head>
6  <body>
7  |   <h1>Привет, мир!</h1>
8  |   <p>Это моя первая веб-страница.</p>
9  </body>
10 </html>
```

- <!DOCTYPE html> объявляет версию HTML документа. В данном случае, это HTML5.
- <html> - корневой элемент HTML страницы.
- <head> - содержит метаинформацию о документе, которая не отображается на самой веб-странице.
- <title> - задает заголовок веб-страницы, который отображается в заголовке окна браузера.
- <body> - это основное содержимое веб-страницы, которое отображается в браузере.
- <h1> и <p> - это примеры тегов, которые мы используем для добавления содержимого на веб-страницу. <h1> определяет заголовок, а <p> определяет абзац текста.

Виды Doctype

HTML 5

```
<!DOCTYPE html>
```

Самая свежая версия, сейчас используется на подавляющем большинстве сайтов. Содержит указания, что сайт свёрстан с использованием html5-тегов и сообщает браузеру, что тот должен предоставить доступ ко всем свойствам и элементам html5.

HTML 4.01

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

HTML 4.01 Strict — содержит все HTML элементы и атрибуты, но не включает в себя устаревшие презентационные элементы (например, font) и фреймсет.

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

HTML 4.01 Transitional — содержит все HTML элементы и атрибуты, также устаревшие презентационные. Фреймсеты запрещены.

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

HTML 4.01 Frameset — аналогичен HTML 4.01 Transitional, но позволяет использовать фреймсет.

XHTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
strict.dtd">
```

XHTML 1.0 Strict — включает все HTML элементы и атрибуты, кроме устаревших презентационных. Фреймсеты запрещены. Код должен соблюдать стандарты XML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">
```

XHTML 1.0 Transitional — включает все HTML элементы и атрибуты, вместе с устаревшими презентационными. Фреймсеты запрещены. Код должен соблюдать стандарты XML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Frameset//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
frameset.dtd">
```

XHTML 1.0 Frameset — аналогичен XHTML 1.0 Transitional, но позволяет использовать фреймсеты. Код должен соблюдать стандарты XML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML  
1.1//EN"  
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd"  
>
```

XHTML 1.1 This DTD — аналогичен XHTML 1.0 Strict, но позволяет добавлять модули.

Любым доктайпам, кроме самого первого в этой таблице, подпевают динозавры. Вы вряд ли не будете их использовать в трезвом виде или не на спор, но знать, какими они были, — отнюдь не лишнее для истинного джентельмена.

①

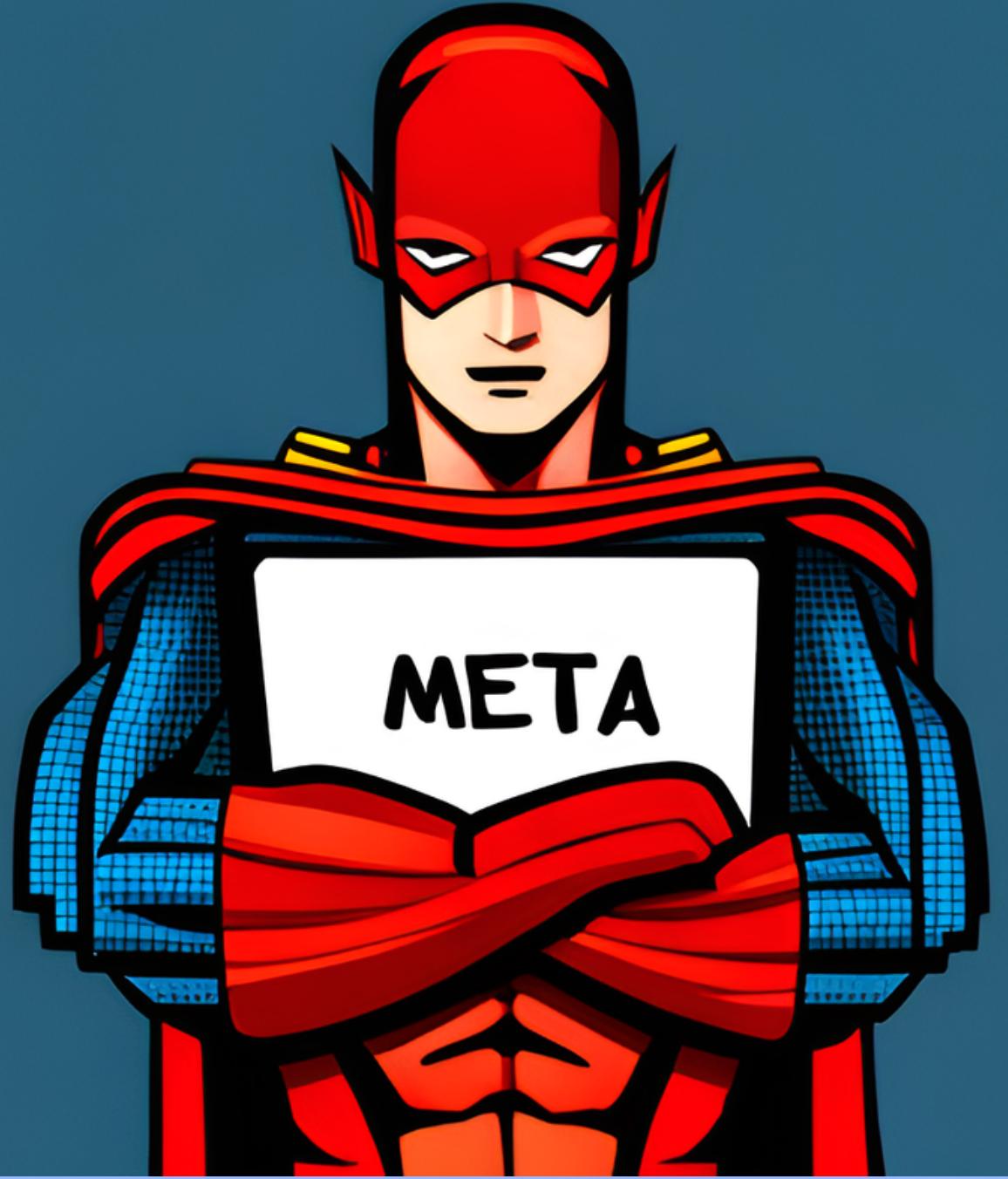
🛠 Если сравнивать HTML с устройством человека, то `<!DOCTYPE>` — это его вид. Важно знать, что было раньше, но не надо быть австралопитеком. Единственный вариант, который можно использовать — это `<!DOCTYPE html>`, все остальные устарели.

```
<> index.html > html
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="UTF-8">
5      <title>Моя веб-страница</title>
6      <meta name="description" content="Описание моей веб-страницы">
7      <meta name="keywords" content="мои, ключевые, слова">
8      <meta name="author" content="Имя автора">
9      <meta name="viewport" content="width=device-width, initial-scale=1.0">
10     <link rel="stylesheet" href="styles.css">
11     <script src="script.js"></script>
12 </head>
13 <body>
14     <!-- Тело документа -->
15 </body>
16 </html>
```

<head> - это один из основных элементов HTML документа. Он содержит метаинформацию о документе, которая не отображается на самой веб-странице, но нужна для браузеров, поисковых систем и других веб-сервисов.

Вот некоторые важные элементы, которые обычно размещаются внутри тега **<head>**:

Тег **<title>** используется для определения заголовка веб-страницы. Заголовок отображается в заголовке окна браузера и в результатах поисковых систем. Он важен для **SEO (Search Engine Optimization** - оптимизация для поисковых систем).



Тег `<meta>` используется для хранения различной метаинформации о веб-странице, такой как кодировка символов, описание страницы, ключевые слова для поисковых систем и другое.

В `<head>` можно также включить ссылки на внешние **CSS** и **JavaScript** файлы.

Тег `<link>` используется для связи веб-страницы с внешним ресурсом, обычно это стилевой файл **CSS**. Он также используется для добавления иконок сайта, которые отображаются рядом с заголовком страницы на вкладке браузера.

- 1.<meta charset="UTF-8">: Этот тег используется для определения кодировки символов документа. "UTF-8" - это универсальная кодировка, которая поддерживает любые символы и языки.
- 2.<meta name="description" content="Описание моей веб-страницы">: Этот тег используется для определения описания веб-страницы. Описание может быть использовано поисковыми системами для отображения сниппета (краткого описания) в результатах поиска.
- 3.<meta name="keywords" content="мои, ключевые, слова">: Этот тег ранее использовался для определения ключевых слов веб-страницы для поисковых систем. Однако современные поисковые системы, включая Google, больше не используют этот тег для ранжирования веб-страниц.
- 4.<meta name="author" content="Имя автора">: Этот тег используется для указания автора веб-страницы.
- 5.<meta name="viewport" content="width=device-width, initial-scale=1.0">: Этот тег используется для контроля размера области просмотра и масштабирования на мобильных устройствах. Свойство width=device-width устанавливает ширину области просмотра равной ширине устройства, а initial-scale=1.0 устанавливает начальный масштаб при просмотре страницы. Это важно для адаптивного дизайна, чтобы страница корректно отображалась на различных устройствах.



- 1.<h1> до <h6>: Эти теги используются для определения заголовков разного уровня. <h1> используется для основного заголовка страницы, а <h2> до <h6> для подзаголовков.
- 2.<p>: Этот тег используется для определения абзацев текста.
- 3.<a>: Этот тег используется для создания гиперссылок. Атрибут href указывает URL-адрес, на который ведет ссылка.
- 4.: Этот тег используется для вставки изображений. Атрибут src указывает URL-адрес изображения, а alt - альтернативный текст, который отображается, если изображение не может быть загружено.

1. **** и ****: Эти теги используются для создания ненумерованных списков. **** определяет сам список, а **** - каждый элемент списка.
2. **** и ****: Эти теги используются для создания нумерованных списков. **** определяет сам список, а **** - каждый элемент списка.
3. **<table>**, **<tr>**, **<th>**, **<td>**: Эти теги используются для создания таблиц. **<table>** определяет таблицу, **<tr>** - строку таблицы, **<th>** - заголовочную ячейку, а **<td>** - ячейку таблицы.
4. **<div>**: Этот тег используется как контейнер для других HTML элементов. Он часто используется совместно с **CSS** для стилизации содержимого.
5. ****: Этот тег также является контейнером для текста или других HTML элементов. Он обычно используется с **CSS** для стилизации специфических частей текста.
6. Тег **<hr>** определяет тематический разрыв в HTML-странице (например, смещение темы)
7. **<form>**, **<input>**, **<label>**, **<button>**: Эти теги используются для создания форм. **<form>** определяет саму форму, **<input>** и **<button>** - поля для ввода и кнопки, а **<label>** - метки для полей ввода.
8. Это лишь некоторые из многих HTML тегов, но они являются основными и часто используются при создании HTML страниц.

Разница между блочными и строчными (инлайновыми) тегами состоит в их поведении и способе отображения на веб-странице:

1. Блочные теги:

- Занимают всю доступную ширину горизонтального пространства.
- Выводятся с новой строки и прерывают текущий блок.
- Можно задавать им размеры в виде ширины и высоты.
- Примеры блочных тегов: **<div>**, **<p>**, **<h1>**, ****, ****.

2. Строчные (инлайновые) теги:

- Занимают только необходимую ширину содержимого.
- Не прерывают текущий блок и выводятся на одной строке с соседними элементами.
- Нельзя задавать им размеры в виде ширины и высоты.
- Примеры строчных тегов: ****, **<a>**, ****, ****, ****.

Атрибуты в HTML предоставляют дополнительную информацию о HTML-элементах и обычно присутствуют в открывающем теге



class: Этот атрибут используется для определения одного или нескольких имен классов для элемента.

id: Этот атрибут используется для определения уникального идентификатора для элемента. Идентификатор может быть использован **CSS** и **JavaScript** для доступа к элементу и применения стилей.

src: Этот атрибут используется в тегах ``, `<script>`, `<iframe>`, `<audio>` и `<video>` для указания URL-адреса источника.

href: Этот атрибут используется в теге `<a>` для указания URL-адреса, на который ведет ссылка.

style: Этот атрибут используется для применения встроенных стилей CSS к элементу.

width и **height**: Эти атрибуты используются для определения ширины и высоты элемента, например изображения.

value, **name** и **type****: Эти атрибуты используются в теге `<input>` для определения значения, имени и типа поля ввода.

disabled, **readonly** и **required****: Эти атрибуты используются для контроля поведения полей формы.

alt: Этот атрибут используется в теге `` для предоставления альтернативного текста, который отображается, если изображение не может быть загружено.

```
<input type="text" name="username">
```

text: стандартное текстовое поле для одной строки текста.

```
<input type="password" name="password">
```

password: текстовое поле, в котором введенные символы скрываются для безопасности.

```
<input type="checkbox" name="option1" value="Milk">
```

```
<input type="checkbox" name="option2"  
value="Butter">
```

checkbox: поле для установки флагка, которое позволяет выбрать несколько значений из нескольких вариантов.

```
<input type="radio" name="gender" value="male">
```

```
<input type="radio" name="gender" value="female">
```

radio: радиокнопка, которая позволяет выбрать одно значение из нескольких вариантов.

```
<input type="email" name="email">
```

email: поле для ввода электронной почты. Браузеры могут предоставлять проверку на валидность электронной почты и интерактивные элементы управления для работы с этим полем.

```
<input type="date" name="birthday">
```

date: поле для ввода даты. Браузеры могут предоставлять интерактивный выбор даты для работы с этим полем.

```
<input type="number" name="quantity" min="1"
```

```
max="10">
```

number: поле для ввода числа. Можно задать минимальное и максимальное значения.

