

**HTML**  
**<<HyperText Markup Language>>**

# WWW – World Wide Web

Изобретателями всемирной паутины считаются **Тим Бернерс-Ли**. Он является автором технологий **HTTP, URI/URL** и **HTML**

В **1989** году, работая в **CERN** (conseil européen pour la recherche nucléaire) над внутренней сетью организации, **Тим Бернерс-Ли** предложил глобальный гипертекстовый проект, теперь известный как «**Всемирная паутина**». Для осуществления проекта **Тимом Бернерсом-Ли** (совместно с его помощниками) были изобретены идентификаторы **URI**, протокол **HTTP** и язык **HTML**.

В период с **1991** по **1993** год **Бернерс-Ли** усовершенствовал технические спецификации этих стандартов и опубликовал их. Но, всё же, официально годом рождения **Всемирной паутины** нужно считать **1989** год

В рамках проекта **Бернерс-Ли** написал первый в мире веб-сервер, называвшийся «**httpd**», и первый в мире **гипертекстовый веб-браузер**, называвшийся «**WorldWideWeb**». Этот браузер был одновременно и **WYSIWYG**-редактором (what you see is what you get — что видишь, то и получишь). Программа работала в среде **NeXTStep**.

**Первый в мире веб-сайт был размещён Бернерсом-Ли 6 августа 1991 года на первом веб-сервере**

# HTML

## <HyperText Markup Language>

- стандартизированный **язык разметки** документов во **Всемирной паутине**
- большинство **веб-страниц** содержат описание разметки на языке **HTML** (или **XHTML**)
- интерпретируется браузерами, полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства
- во **Всемирной паутине HTML-страницы**, как правило, передаются браузерам от сервера по протоколам **HTTP** или **HTTPS**, в виде простого текста или с использованием шифрования
- **XHTML** является более строгим вариантом **HTML**, он следует синтаксису **XML** и является приложением языка **XML** в области разметки гипертекста

Язык **HTML** до 5-й версии определялся как приложение **SGML** (стандартного обобщенного языка разметки по стандарту **ISO 8879**). Спецификации **HTML5** формулируются в терминах **DOM** (объектной модели документа).

# Тег

**Тег** – элемент языка разметки гипертекста

Текст, содержащийся между начальным и конечным тегом, отображается и размещается в соответствии со свойствами, указанными в начальном теге

Регистр, в котором набрано имя элемента и имена атрибутов, в **HTML** значения не имеет (в отличие от **XHTML**)

Обычно используются **парные теги** — **открывающий**, или начальный, и **закрывающий**, или конечный. Возможно также применение **одиного тега**. Например, тег отступа абзаца может оформлять абзац между тегами **<p>** и **</p>**, а может в одиночной форме до следующего первого попавшегося тега **<p>**

**Набор и рекомендуемые интерпретации тегов определены организацией W3C**

# Структура Тегов

Весь текст, заключенный между начальным и конечным тегом, включая и сами эти теги, называется **элементом**. Сам же текст между тегами — **содержанием элемента**.

Содержание элемента может включать в себя любой текст, в том числе и другие элементы.

У тега могут быть свойства, называемые **атрибутами**, дающие дополнительные возможности форматирования текста. Они записываются в виде сочетания: **имя атрибута = значения**, причем текстовые значения заключаются в **кавычки**.

**Атрибуты** тегов расширяют возможности самих тегов и позволяют гибко управлять различными настройками отображения элементов веб-страницы

Элементы должны быть строго вложенными друг в друга, «перехлест» элементов недопустим:

неправильно → **<a><b></a></b>**

правильно → **<person><i>John Doe</i></person>**

# Типы тегов

Каждый **тег HTML** принадлежит к определенной группе тегов

Все теги делятся на две основных группы:

- **строчные**
- **блочные**

Также теги условно можно разделить на:

- теги верхнего уровня
- теги заголовка документа
- блочные элементы
- строчные элементы
- ссылки
- списки
- изображения
- таблицы
- формы

Один и тот же тег может одновременно принадлежать разным группам

# Структура HTML-документа

**HTML** — **теговый** язык разметки документов.

Любой документ на языке **HTML** представляет собой набор элементов, причём начало и конец каждого элемента обозначается специальными пометками — **тегами**.

Каждый **HTML-документ**, отвечающий спецификации **HTML** какой-либо версии, должен начинаться со строки объявления версии **HTML** → **<!DOCTYPE...>**

Далее обозначается начало и конец документа тегами **<html>** и **</html>** соответственно. Внутри этих тегов должны находиться теги заголовка (**<head></head>**) и тела (**<body></body>**) документа

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Example</title>
</head>
<body>
<h1>Hello world</h1>
</body>
</html>
```

# Теги верхнего уровня

предназначены для формирования структуры веб-страницы и определяют раздел заголовка и тела документа

**<html>** – является контейнером, который заключает в себе всё содержимое веб-страницы, включая теги **<head>** и **<body>**

**<head>** – предназначен для хранения других элементов, цель которых помочь браузеру в работе с данными. Также внутри элемента **<head>** находятся **метатеги**, которые используются для хранения информации, предназначенной для браузеров и поисковых систем. Механизмы поисковых систем обращаются к метатегам для получения описания сайта, ключевых слов и других данных

**<body>** – предназначен для хранения содержания веб-страницы, отображаемого в окне браузера



# Теги заголовка документа

К этим тегам относятся элементы, которые располагаются в элементе **<head>**. Все эти теги напрямую не отображаются в окне браузера, за исключением тега **<title>**, который определяет название веб-страницы

**<title>** – используется для отображения строки текста в левом верхнем углу окна браузера, а также на вкладке. Такая строка сообщает пользователю название сайта и другую информацию, которую добавляет разработчик

**<meta>** – мета теги используются для хранения информации, предназначенной для браузеров и поисковых систем. Например, механизмы поисковых систем обращаются к метатегам для получения описания сайта, ключевых слов и других данных. Хотя тег `<meta>` всего один, он имеет несколько атрибутов, поэтому к нему и применяется множественное число

# Блочные элементы

характеризуются тем, что занимают всю доступную ширину, высота элемента определяется его содержимым, и он всегда начинается с новой строки

**<blockquote>** – предназначен для выделения длинных цитат внутри документа. Текст, обозначенный этим тегом, традиционно отображается как выровненный блок с отступами слева и справа, а также с пустым пространством сверху и снизу

**<div>** – относится к универсальным блочным контейнерам и применяется в тех случаях, где нужны блочные элементы без дополнительных свойств

**<h1>...<h6>** – группа тегов определяет текстовые заголовки разного уровня, которые показывают относительную важность секции, расположенной после заголовка

**<hr>** – рисует горизонтальную линию, которая по своему виду зависит от используемых атрибутов

**<p>** – определяет параграф (абзац) текста

**<pre>** – задает блок предварительно форматированного текста. Такой текст отображается обычно моноширинным шрифтом и со всеми пробелами между словами

# Строчные элементы

элементы веб-страницы, которые являются непосредственной частью другого элемента, например, текстового абзаца

**<a>** – является одним из важных элементов HTML и предназначен для создания ссылок

**<b>** – определяет жирное начертание шрифта

**<big>** – увеличивает размер шрифта на единицу по сравнению с обычным текстом

**<br>** – устанавливает перевод строки в том месте, где этот тег находится

**<em>** – предназначен для акцентирования текста. Браузеры отображают такой текст курсивным начертанием

**<i>** – устанавливает курсивное начертание шрифта

**<img>** – предназначен для отображения на веб-странице изображений в графическом формате **GIF, JPEG** или **PNG**

**<small>** – уменьшает размер шрифта на единицу по сравнению с обычным текстом

**<span>** – универсальный тег, предназначенный для определения строчного элемента внутри документа

**<strong>** – предназначен для акцентирования текста. Браузеры отображают такой текст жирным начертанием

**<sub>** – отображает шрифт в виде нижнего индекса

**<sup>** – отображает шрифт в виде верхнего индекса

# Основная разница между блочными и строчными элементами

- строчные элементы могут содержать только данные или другие строчные элементы, а в блочные допустимо вкладывать другие блочные элементы, строчные элементы, а также данные. Иными словами, строчные элементы никак не могут хранить блочные элементы.
- блочные элементы всегда начинаются с новой строки, а строчные таким способом не акцентируются.
- блочные элементы занимают всю доступную ширину, например, окна браузера, а ширина строчных элементов равна их содержимому плюс значения отступов, полей и границ.

# Атрибуты

это дополнительные свойства для тега которые могут влиять на отображение тега на странице, а также предоставлять дополнительные возможности работы с тегами но никак не влияют на его отображение на странице

Универсальные атрибуты(существуют практически у каждого тега):

**id** – указывает имя стилевого идентификатора.

**class** – определяет имя класса, которое позволяет связать тег со стилевым оформлением.

**style** – применяется для определения стиля элемента с помощью правил **CSS**.

**title** – описывает содержимое элемента в виде всплывающей подсказки.

**lang** – браузер использует значение параметра для правильного отображения некоторых национальных символов.

**hidden** – скрывает содержимое элемента от просмотра.

**dir** – задает направление и отображение текста — слева направо или справа налево.

Также существует множество атрибутов которые специфичны для каждого тега отдельно. События которые происходят на теге также описываются атрибутами

# Работа с текстом

- любое количество пробелов идущих подряд, в браузере отображается как один
- нет расстановки переносов в тексте
- текст занимает ширину окна браузера
- по умолчанию между параграфами (**<p></p>**) существует небольшой вертикальный отступ, называемый отбивкой
- тег переноса строки **<br>** не создает дополнительных вертикальных отступов между строками и может применяться практически в любом тексте
- заголовки (**<h1>...<h6>**) выполняют важную функцию на веб-странице. Во-первых, они показывают важность раздела, к которому относятся. Чем больше заголовков и его “вес”, тем более он значимый. Как правило, на веб-странице применяют заголовки с первого по третий уровень, их вполне достаточно

# Выравнивание текста

Выравнивание текста определяет его внешний вид и ориентацию краев абзаца

Для установки выравнивания текста обычно используется тег параграфа **<p>** с атрибутом **align**, который определяет способ выравнивания. Также блок текста допустимо выравнивать с помощью тега **<div>** с аналогичным атрибутом **align**

Атрибут **align** можно применять как для текста, так и для заголовков

Значения для атрибута **align**:

**left** — выравнивание по левому краю, задается по умолчанию;

**right** — выравнивание по правому краю;

**center** — выравнивание по центру;

**justify** — выравнивание по ширине (одновременно по правому и левому краю). Это значение работает только для текста, длина которого более, чем одна строка.

# Спецсимволы

Для отображения символов, которых нет на клавиатуре, применяются специальные знаки, начинающиеся с амперсанд **&** и заканчивающиеся точкой с запятой **;**;

Имя	Код	Вид	Описание
&amp;	&#38;	<b>&amp;</b>	амперсанд
&nbsp;	&#160;		неразрывный пробел
&pound;	&#163;	<b>£</b>	фунт стерлингов
&euro;	&#8364;	<b>€</b>	знак евро
&para;	&#182;	<b>¶</b>	символ параграфа
&sect;	&#167;	<b>§</b>	параграф
&copy;	&#169;	<b>©</b>	знак copyright
&reg;	&#174;	<b>®</b>	знак зарегистрированной торговой марки
&trade;	&#8482;	<b>™</b>	знак торговой марки
&deg;	&#176;	<b>°</b>	градус
&plusmn;	&#177;	<b>±</b>	плюс-минус
&frac14;	&#188;	<b>¼</b>	дробь - одна четверть
&frac12;	&#189;	<b>½</b>	дробь - одна вторая
&frac34;	&#190;	<b>¾</b>	дробь - три четверти
&times;	&#215;	<b>×</b>	знак умножения
&divide;	&#247;	<b>÷</b>	знак деления
&fnof;	&#402;	<b>f</b>	знак функции



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Basic Html</title>
</head>
<body bgcolor="gainsboro">
<h1 align="center">First HTML Page</h1>
<hr>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
<hr>
<p>
Default paragraph with <strong>STRONG,</strong> <b>BOLD</b> and <em>CURSIVE</em> text!<br>
New line using tag <br>
Specsymbols: &nbsp; &nbsp; &copy; &nbsp; &frac12; &nbsp; &reg; &nbsp; &trade; &nbsp; &fnof;
</p>
<p align="right">
Lorem ipsum dolor sit amet consectetur adipisicing elit. Ut excepturi cupiditate deserunt,
neque nam ad, saepe a officiis omnis officia, maiores nobis magni? Quia, quas! At hic id nam
ab.
</p>
<p>
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Sint quia, officiis voluptatum eos
nemo vel natus ducimus cupiditate non perferendis quos voluptatem aperiam dolores quibusdam
sit porro nostrum voluptas alias!
</p>
</body>
</html>
```

# Ссылки – <a>

позволяют переходить с одной веб-страницы на другую.

Особенность их состоит в том, что сама ссылка может вести не только на **HTML-файлы**, но и на файл любого типа, причем этот файл может размещаться совсем на другом сайте.

Атрибут **href** определяет адрес документа, на который следует перейти, а содержимое контейнера **<a>** является ссылкой. Адрес ссылки может быть как **абсолютным**, так и **относительным**

Файл по ссылке открывается в окне браузера только в тех случаях, когда браузер знает тип документа

Атрибут **target** определяет в какой вкладке будет открыта ссылка:

**\_blank** — загружает страницу в новое окно браузера.

**\_self** — загружает страницу в текущее окно (это значение задается по умолчанию).

**\_parent** — загружает страницу во фрейм-родитель, если фреймов нет, то это значение работает как **\_self**.

**\_top** — отменяет все фреймы и загружает страницу в полном окне браузера, если фреймов нет, то это значение работает как **\_self**.

Атрибут **title** добавляет поясняющий текст к ссылке в виде всплывающей подсказки. Такая подсказка отображается, когда курсор мыши задерживается на ссылке, после чего подсказка через некоторое время пропадает

# Якорные ссылки

– закладка с уникальным именем на определенном месте веб-страницы, предназначенная для создания перехода к ней по ссылке. Якорные ссылки удобно применять в документах большого объема, чтобы можно было быстро переходить к нужному разделу

Для создания якорной ссылки следует вначале сделать закладку в соответствующем месте и дать ей имя при помощи атрибута **name** тега **<a>**

В качестве значения href для перехода к этому якорю используется имя закладки с символом решетки (#) впереди

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Basic Html</title>
</head>

<p><a name="top">Anchor</a></p>

<p><a href="https://tel-ran.com" target="_blank">Te-Ran Home Page</a></p>
<p><a href="./about.html">About</a></p>
<p><a href="https://image.com/tr_logo.png">Tel-Ran Logo</a></p>
<p>...</p>

<p><a href="#top">Up to anchor</a></p>

</body>
</html>
```

# Списки – <ul> <ol> <li>

Списки предоставляют возможность упорядочить разные данные и представить их в наглядном и удобном для пользователя виде.

типы:

**<ul>** – маркированные

**<ol>** – нумерованные

Тип маркера или нумерации задается с помощью атрибута type:

**<ul>** – **disc** | **circle** | **square**

**<ol>** – **1** | **A** | **a** | **I** | **i**

Атрибут start задает начало нумерации для нумерованных списков

Элемент списка задается тегом **<li>**

```
<ul type="circle">  
  <li>Item 1</li>  
  <li>Item 2</li>  
  <li>Item 3</li>  
</ul>
```

```
<ol type="I" start="5">  
  <li>Item 1</li>  
  <li>Item 2</li>  
  <li>Item 3</li>  
</ol>
```

## Список определений – **<dl>** **<dt>** **<dd>**

Список определений хорошо подходит для расшифровки терминов, словаря, справочника и т.д.

Список определений состоит из двух элементов — термина и его определения. Сам список задается с помощью контейнера **<dl>**, термин — тегом **<dt>**, а его определение — с помощью тега **<dd>**

```
<dl>  
  <dt>Java</dt>  
  <dd>Java - Object Oriented Programming language</dd>  
  <dt>HTML</dt>  
  <dd>HTML - Hypertext Markup Language</dd>  
</dl>
```

# Изображения – <img>

Добавление изображения происходит в два этапа: вначале готовится графический файл желаемого размера, затем он добавляется на страницу через тег **<img>**. Сам **HTML** предназначен только для того, чтобы отобразить требуемую картинку, при этом никак ее не меняя.

Атрибуты:

**src** – путь к изображению может быть локальный или глобальный

**alt** – альтернативный текст если изображение не загрузилось или еще грузится

**width** – задает ширину картинки в пикселях

**height** – задает высоту картинки в пикселях

**align** – [**bottom** | **left** | **middle** | **right** | **top**] задает положение относительно текста или других изображений

```

```

Поскольку веб-страница загружается по сети, существенным фактором выступает объем графического файла, встроенного в документ. Чем он меньше, тем быстрее отобразится изображение

# Формы – **<form>** **<input>** **<select>** **<label>**

**<input>** является одним из разносторонних элементов формы и позволяет создавать разные элементы интерфейса и обеспечить взаимодействие с пользователем.

Главным образом **<input>** предназначен для создания текстовых полей, различных кнопок, переключателей и флажков.

Хотя элемент **<input>** не требуется помещать внутрь контейнера **<form>**, определяющего форму, но если введенные пользователем данные должны быть отправлены на сервер, где их обрабатывает серверная программа, то указывать **<form>** обязательно.

Тег **<label>** устанавливает связь между определенной меткой, в качестве которой обычно выступает текст, и элементом формы (**<input>**, **<select>**, **<textarea>**). Такая связь необходима, чтобы изменять значения элементов формы при нажатии курсором мыши на текст.

Существует два способа связывания объекта и метки. Первый заключается в использовании идентификатора **id** внутри элемента формы и указании его имени в качестве атрибута **for** тега **<label>**. При втором способе элемент формы помещается внутрь контейнера **<label>**.



# Атрибуты <input>

Основной атрибут тега **<input>**, является атрибут **type**.

Он позволяет задавать следующие элементы формы:

**text** – текстовое поле

**password** – поле с паролем

**radio** – переключатель

**checkbox** – флажок

**hidden** – скрытое поле

**button** – кнопка

**submit** – кнопка для отправки формы

**reset** – кнопка для очистки формы

**file** – поле для отправки файла

**image** – кнопка с изображением

Для каждого элемента существует свой список атрибутов, которые определяют его вид и характеристики.

Кроме того, в **HTML5** добавлено еще более десятка новых элементов.

<b>accept</b>	Устанавливает фильтр на типы файлов для поле загрузки файлов
<b>align</b>	Определяет выравнивание изображений
<b>alt</b>	Альтернативный текст для кнопки с изображением
<b>autocomplete</b>	Включает или отключает автозаполнение
<b>border</b>	Толщина рамки вокруг изображения
<b>checked</b>	Предварительно активированный переключатель или флажок
<b>disabled</b>	Блокирует доступ и изменение элемента
<b>max</b>	Верхнее значение для ввода числа или даты
<b>maxlength</b>	Максимальное количество символов разрешенных в тексте
<b>min</b>	Нижнее значение для ввода числа или даты
<b>multiple</b>	Позволяет загрузить несколько файлов одновременно
<b>name</b>	Имя поля, предназначено для того, чтобы обработчик формы мог его идентифицировать
<b>placeholder</b>	Выводит подсказывающий текст
<b>required</b>	Обязательное для заполнения поле
<b>size</b>	Ширина текстового поля
<b>src</b>	Адрес графического файла для поля с изображением
<b>type</b>	Сообщает браузеру, к какому типу относится элемент формы
<b>value</b>	Значение элемента

```
<form action="#" method="post">
  <p>
    <label for="email">Email: </label>
    <input type="text" id="email">
  </p>
  <p>
    <label for="password">Password: </label>
    <input type="password" id="password">
  </p>
  <p>
    <label for="lesson">Lesson: </label><select name="lesson" id="lesson">
      <option value="java">Java</option>
      <option value="web">Web</option>
      <option value="fullstack">Full stack</option>
    </select>
  </p>
  <p>
    <label>Haifa: <input name="city" type="radio"></label>
    <label>Rehovot: <input name="city" type="radio"></label>
  </p>
  <p>
    <textarea name="comment" cols="40" rows="3"></textarea>
  </p>
  <p>
    <input type="submit" value="Send">
    <input type="reset" value="Reset">
  </p>
</form>
```

# Таблицы – `<table>` `<tr>` `<td>`

Благодаря универсальности таблиц, большому числу параметров, управляющих их видом, таблицы надолго стали определенным стандартом для верстки веб-страниц. Таблица с невидимой границей представляет собой словно модульную сетку, в блоках которой удобно размещать элементы веб-страницы. Но такой подход не совсем правильный, ведь каждый объект **HTML** определен для своих собственных целей и если он используется не по назначению, причем повсеместно, это значит, что альтернатив нет. Так оно и было долгое время, пока на смену таблицам при верстке сайтов не пришли слои.

Атрибуты **`<table>`**:

**`align`** – [**`left`** | **`right`** | **`center`**] задает выравнивание таблицы по краю окна браузера. Когда используются значения **`left`** и **`right`**, текст начинает обтекать таблицу сбоку и снизу.

**`bgcolor`** – устанавливает цвет фона таблицы

**`border`** – устанавливает толщину границы в пикселах вокруг таблицы

**`cellpadding`** – определяет расстояние между границей ячейки и ее содержимым

**`cellspacing`** – задает расстояние между внешними границами ячеек

**`cols`** – указывает количество столбцов в таблице, помогая браузеру в подготовке к ее отображению

**`rules`** – сообщает браузеру, как отображать границы между ячейками. По умолчанию рамка рисуется вокруг каждой ячейки, образуя тем самым сетку

**`width`** – Задает ширину таблицы

# Ячейки таблицы

Каждая новая строка задается тегом **<tr>** который содержит внутри себя ячейки которые задаются тегом **<td>**

Атрибуты **<td>**:

**align** – [**left** | **right** | **center**] задает выравнивание содержимого ячейки по горизонтали

**bgcolor** – устанавливает цвет фона ячейки

**colspan** – устанавливает число ячеек, которые должны быть объединены по горизонтали

**rowspan** – устанавливает число ячеек, которые должны быть объединены по вертикали

**valign** – [**top** | **middle** | **bottom** | **baseline**] устанавливает вертикальное выравнивание содержимого ячейки

**width** – задает ширину ячейки. Суммарное значение ширины всех ячеек может превышать общую ширину таблицы только в том случае, если содержимое ячейки превышает указанную ширину.

```

<table bgcolor="#d6d6d6" border="1" align="center" width="400">
  <tr>
    <td>Cell 1</td>
    <td colspan="2" align="center"> Cell 2</td>
  </tr>
  <tr>
    <td rowspan="2">Cell 3</td>
    <td>Cell 4</td>
    <td>Cell 5</td>
  </tr>
  <tr>
    <td>Cell 5</td>
    <td>Cell 6</td>
  </tr>
</table>

```

```

<table border="1" align="center" width="400" cellpadding="4" cellspacing="0">
  <caption>Table about nothing</caption>
  <tr>
    <th>&nbsp;</th><th>2019</th><th>2018</th><th>2017</th>
  </tr>
  <tr>
    <td>Item 1</td><td>43</td><td>51</td><td>79</td>
  </tr>
  <tr>
    <td>Item 2</td><td>29</td><td>34</td><td>48</td>
  </tr>
  <tr>
    <td>Item 3</td><td>38</td><td>57</td><td>36</td>
  </tr>
</table>

```

# HTML 5 Семантические тэги

Семантические элементы являются одним из значимых нововведений в **HTML5**. До их появления вся разметка веб-страниц строилась при помощи элементов **<div>**, которым присваивались идентификаторы (**id**) или классы (**class**). Для размещения боковых панелей, верхних и нижних колонтитулов, элементов навигации и прочих структурных блоков использовались блоки **div** с соответствующими значениями (например, **class="footer"**).

## Предназначение семантические теги

Несмотря на то, что уже в **HTML4** можно было создавать сайты с понятной для пользователей структурой, для поисковых систем и браузеров элементы страницы оставались однотипными, ведь каждый из элементов был создан при помощи тега **<div>**, который, по сути, является обычной прямоугольной областью на странице, не несущей какую-то конкретную информацию, кроме той, которую в нее заложит разработчик. Для решения этой проблемы было принято решение ввести **семантические теги**, используя которые можно сделать структуру страницы более понятной и логичной.

## Применение семантические теги

Это нововведение уже нашло свое применение в мире смартфонов и ридеров – например, используя браузер **Safari** на **iPhone** или **iPad**, вы можете перейти в режим чтения, когда на экране выводится только текстовая информация страницы – и воспринимать ее гораздо легче, ведь нет отвлекающих элементов в виде картинок, баннеров, рекламы. Однако этот режим будет доступен лишь в том случае, если страница имеет семантическую разметку, благодаря которой браузер может понять, где находится заголовок, где текст, а где навигация. Как раз ту разметку, о которой идет речь.

Не стоит забывать и о поисковых роботах – семантическая разметка поможет им правильно проанализировать страницу, отделив важную информацию, касающуюся содержимого страницы, от второстепенной (шапка, подвал, навигация). И это обязательно положительно скажется на предназначение сайта.

Таким образом, главное предназначение семантической разметки – делать структуру документа более понятной на всех этапах использования, от разработки до публикации в сети и индексации поисковыми машинами.



# Структура документа

Так выглядит общая структура документа в HTML4.



# Структура документа

Так выглядит общая структура документа в HTML5 с использованием новых тэгов разметки

**<header>**

здесь располагается логотип и лозунг,  
а также поле поиска и ссылки на различные разделы сайта

**<nav>**

здесь располагается  
навигационное меню  
связывающие  
документы сайта

**Содержимое**

здесь располагается  
содержимое HTML-документа

**<footer>**

здесь располагается информация об авторских правах,  
авторе документа и могут присутствовать ссылки на другие  
ресурсы

# Список семантических тегов

Тег	Назначение
<b>&lt;header&gt;</b>	“шапка” документа либо секции
<b>&lt;nav&gt;</b>	основная навигация
<b>&lt;main&gt;</b>	главный контент страницы
<b>&lt;article&gt;</b>	статья или новость
<b>&lt;section&gt;</b>	семантически обособленная секция
<b>&lt;aside&gt;</b>	вспомогательный контент
<b>&lt;footer&gt;</b>	“подвал” документа либо секции
<b>&lt;address&gt;</b>	адрес
<b>&lt;mark&gt;</b>	выделенный текст
<b>&lt;figure&gt;</b>	автономный контент (картинка и т.п.)
<b>&lt;figcaption&gt;</b>	подпись для figure
<b>&lt;details&gt;</b>	дополнительные сведения
<b>&lt;summary&gt;</b>	видимая подпись для details
<b>&lt;time&gt;</b>	время / дата

# Описание некоторых семантических тегов

**<header>** – формирует верхнюю часть элемента или страницы (шапку, основной заголовок или группу заголовков), при этом может располагаться в любой части страницы или вообще отсутствовать

**<nav>** – создает раздел с навигационными ссылками на странице или на сайте; в качестве элементов панели навигации могут выступать теги списков **<ul>**, **<li>** и тег адреса документа **<a href="URL">**, допускается использование заголовков (к примеру, **<h2>**);

**<article>** – группирует записи, публикации и статьи

**<section>** – разделяет страницы или записи на тематические блоки, но при этом, в отличие от тега **<div>**, не является блоком-оберткой – тег **<section>** направлен на определение смысловой секции содержимого и обязательно должен содержать заголовок

**<aside>** – группирует содержимое, которое будет показано в блоке сбоку от основного содержания страницы (например, для выделения цитат, карточек и т.д.)

**<footer>** – формирует нижнюю часть элемента или страницы, которая обычно состоит из технической и юридической информации, контактных данных

**<address>** – определяет контактную информацию, касающуюся автора текста или страницы

**<mark>** – выделяет "важную" часть в тексте

**<figure>** – группирует элементы (например, иллюстрации и подписи, но не ограничивается именно этими элементами – вы можете группировать видео, аудио контент, цитаты, таблицы и многое другое по своему желанию), для создания подписи необходимо внутри данного тега использовать тег **<figcaption>**