**ОСНОВЫ HTML**

**Содержание:**

1.Синтаксис HTML

2.Блочные и строчные элементы

3. Иерархия в HTML

4.Семантика в HTML

5. Форматирование HTML

6. Валидный HTML-документ

7. Проверка знаний

**1.Синтаксис HTML**

HTML расшифровывается как HyperText Markup Language (**язык разметки гипертекста**):

* **язык** означает, что он может быть прочитан как человеком, так и компьютером;
* **разметка** означает, что написанный вами код помечается с помощью ключевых слов;
* **гипертекст** означает, что он использует HTTP как часть Интернета.

Как и любой язык, HTML поставляется с набором правил. Эти правила относительно простые и сводятся к определению границ, чтобы знать, где что-то начинается и где заканчивается.

Ниже приведён пример абзаца в HTML:

<p>Если Тетрис и научил меня чему-то,

так это тому, что ошибки накапливаются, а достижения исчезают.</p>

То, что вы видите в угловых скобках < и > называется тегами HTML. Они определяют, где что-то начинается и где оно заканчивается.

Каждый из тегов несёт определённый смысл. В нашем случае <р> обозначает абзац текста.

Как правило, они идут парами:

* открывающий тег <р> определяет начало абзаца;
* закрывающий тег </p> определяет его конец.

Единственным различием между открывающим и закрывающим тегами является слэш /, который предшествует имени тега.

При объединении открывающего, закрывающего тегов и всего между ними, вы получите элемент HTML. Строка целиком представляет собой элемент HTML, который использует теги HTML <р> и </p>.

Если вы просмотрите этот пример в вашем браузере, то заметите, что теги HTML в браузере не отображаются. Они читаются только браузером, чтобы знать, какой тип контента вы написали.

**Где писать HTML**

Вы, вероятно, сталкивались c простыми текстовыми файлами, теми, которые имеют расширение .txt. Чтобы такой текстовый файл стал HTML-документом (вместо текстового), вам нужно использовать расширение .html.

Откройте текстовый редактор, скопируйте и вставьте следующее содержимое:

*<p>Это моя первая веб-страница!</p>*

Сохраните этот файл как my-first-webpage.html, просто откройте его вашим браузером и вы увидите:

Это моя первая веб-страница!

Помните:

* используйте текстовый редактор, вроде Notepad++, для создания HTML-документов;
* используйте браузер, вроде Firefox, для открытия HTML-документов.

**Атрибуты**

Атрибуты действуют как дополнительная информация, привязанная к элементу HTML. Они пишутся внутри тега HTML. Таким образом, они также не отображаются в браузере.

Например, атрибут href используется для определения назначения ссылки (которая создаётся тегом <a>):

***<a href="http://www.mozilla.com/firefox">Скачать Firefox</a>***

Есть 16 атрибутов HTML, которые могут быть использованы в любом элементе HTML. Все они не являются обязательными.

Вы в основном будете применять class (который используется для CSS) и title (подсказка, которая появляется при наведении курсора на объект, вроде этого).

Некоторые элементы HTML содержат обязательные атрибуты. Например, при вставке изображения вы должны указать его расположение с помощью атрибута src:

***<img src="#" alt="Описание изображения">***

Принимая во внимание, что цель элемента <img> заключается в показе изображения, то имеет смысл путь к изображению сделать обязательным.

**Комментарии**

Если вы пишете что-то в своём коде без нарушения отображения страницы браузером, то вы можете писать комментарии. Они будут игнорироваться браузером и могут быть использованы только для людей, которые пишут код.

Комментарий начинается с <!-- и заканчивается -->.

***JSFiddle***

***<!-- Это предложение будет игнорироваться браузером -->***

***<p>Привет, мир!</p>***

**Самозакрывающиеся элементы**

Некоторые элементы HTML имеют только открывающий тег:

***JSFiddle***

***<br> <!-- Перевод строки -->***

***<img src="http://placehold.it/50x50" alt="Описание"> <!-- изображение -->***

***<input type="text"> <!-- текстовое поле -->***

Поскольку у них нет закрывающего тега и, следовательно, они не могут содержать ничего внутри, самозакрывающие элементы обычно несут с собой несколько атрибутов, которые предоставляют дополнительную информацию.

**Проверочный тест:**

**1.**Что является элементом HTML?

</p>

<p>

<p>…</p>+++

2. Выберите все открывающие теги

* </a>
* <p>
* <a>
* </p>

3. Выберите все закрывающие теги

* </a>
* <a>
* </p>
* <p>

4. Расставьте в правильном порядке

*<a href=”search.html”>Поиск</a>*

5.Расставьте в правильном порядке

<*img src=”picture.jpg” alt=”Описание”>*

6. Паша написал следующий код HTML, но браузер не показал никакого текста. Почему?

<!-- <p>Тетрис</p> -->

* Неверное вложение тегов
* Файл сохранен с расширением тхт
* Это комментарий, он не отображается в браузере+++
* Не хватает атрибута титл

**Блочные и строчные элементы**

В HTML вам, в основном, будут попадаться два типа элементов HTML.

Блочные элементы, вроде:

* абзацы <р>;
* списки: неупорядоченные (с маркером) <ul> или упорядоченные списки (с числами) <ol>;
* заголовки: от первого уровня <h1> до шестого уровня <h6>;
* статьи <article>;
* разделы <section>;
* длинные цитаты <blockquote>.

Строчные элементы, вроде:

* ссылки <a>;
* выделенные слова <em>;
* важные слова <strong>;
* короткие цитаты <q>;
* аббревиатуры <abbr>.

Блочные элементы предназначены для структурирования основных частей вашей страницы, путём разделения содержимого на логически связанные блоки.

Строчные элементы предназначены, чтобы разграничить часть текста и придать ему определённую функцию или смысл. Строчные элементы, как правило, содержат одно или несколько слов.

<p>Вы уже видели это <a href="http://www.youtube.com">замечательное видео</a> на YouTube?</p>

Открывающие и закрывающие теги

Все блочные элементы содержат открывающие и закрывающие теги.

Как результат, самозакрывающие элементы являются строчными, просто потому, что их синтаксис не позволяет им содержать любой другой элемент HTML.

|  |  |  |
| --- | --- | --- |
|  | Содержит открывающие и закрывающие теги | Самозакрывающие |
| Блочные элементы | <p> </p>  <ul> </ul> <ol> </ol> | Невозможно |
| Строчные элементы | <a> </a>  <strong> </strong> <em> </em> | <input> <br> <img> |

Другие типы элементов HTML

Есть несколько исключений у блочных/строчных элементов. Вот те, с которыми вы чаще всего столкнётесь:

* пункты списка <li>;
* таблица <table>;
* строки таблицы <tr>;
* ячейки таблицы <td>.

**Проверочный тест:**

1. Сопоставьте элемент и его тип

* Блочный <ul>
* Строчный <strong>
* Пункт списка <li>
* Ячейка таблицы <td>
* Таблица <table>
* Строка таблица <tr>

1. Выберите все блочные элементы

* <strong>
* <ul> +++
* <p> +++
* <blockquote> +++
* <q>
* <em>

1. Выберите все строчные элементы

* <h1>
* <article>
* <section>
* <q> +++
* <em> +++
* <strong> +++

1. Сопоставьте элемент с его назначением

<strong> Важные слова

<em> Выделенные слова

<h1> Заголовок первого уровня

<ol> Упорядоченный список

<p> Абзац текста

<section> раздел

1. Выберите правильное утверждение

* Все самозакрывающие элементы являются блочными
* Некоторые самозакрывающие элементы являются блочные
* Самозакрывающие элементы не относятся ни к блочным ни к строчным
* Все самозакрывающие элементы являются строчными +++

1. Выберите правильное утверждение
   * Строчные элементы разбивают весь документ на отдельные строки

* Строчные элементы задают взаимосвязанные заголовки
* Строчные элементы определяют основную статью и разделы документа
* Строчные элементы придают части текста определенный смысл +++

**Иерархия в HTML**

HTML-документ — это как большое семейное древо, с родителями, братьями, детьми, предками и потомками.

Всё это происходит из возможности вкладывать одни элементы HTML внутрь других.

**Вложения**

Давайте напишем простой абзац и улучшим его путём разделения частей текста, вставив два строчных элемента:

<p>Сэр <strong>Алекс Фергюсон</strong> однажды сказал о Филиппо Индзаги:

<q>Этот парень должен был родиться в положении вне игры</q>.</p>

В этой структуре:

* элемент <strong> задаёт слова «Алекс Фергюсон» более важными;
* <q> отмечает его цитату об Индзаги.

Тот факт, что <strong> отображается жирным шрифтом, является только поведением браузера по умолчанию. Помните, что вы должны выбирать элементы HTML в соответствии с их значением, а не как они выглядят.

В данном случае:

* <p> — родительский элемент для <strong> и <q>;
* <strong> и <q> — дочерние элементы для <р>;
* <strong> и <q> — братские элементы.

**Порядок**

Как работает вложение, зависит от расположения открывающего и закрывающего тегов.

Поскольку элемент HTML содержит открывающий тег, закрывающий тег и всё между ними, дочерний элемент должен быть закрыт до закрытия родительского элемента.

<!-- Это НЕВЕРНЫЙ код! :-( -->

<p>

Этот код HTML не будет работать, потому что тег strong я открыл здесь,

<strong>но закрыл его только после абзаца.

</p></strong>

Поскольку <strong> был открыт после <р> (и, таким образом, считается потомком <р>), элемент <strong> должен быть закрыт перед родительским </р>.

<!-- Это правильный код. :-) -->

<p>

Этот код HTML будет работать, потому что тег strong я открыл <strong>и закрыл</strong> правильно.

</p>

**Глубина**

Поскольку дочерние элементы сами по себе могут содержать другие дочерние элементы, то можно написать более глубокую иерархию внутри HTML-документа.

Наш абзац выше может быть частью статьи блога:

<article>

<h1>Известные футбольные цитаты</h1>

<p>

Сэр <strong>Алекс Фергюсон</strong> однажды сказал о Филиппо Индзаги:

<q>Этот парень должен был родиться в положении вне игры</q>.

</p>

<p>

При критике со стороны Джона Карью, <strong>Златан Ибрагимович</strong> ответил:

<q>То, что делает Карью с мячом, я могу сделать с апельсином.</q>

</p>

<p>

<strong>Джордж Бест</strong> сказал:

<q>Я потратил много денег на выпивку, девчонок и быстрые автомобили.

Остальное я просто промотал</q>, —

когда его спросили о его образе жизни.

</p>

</article>

В этой структуре:

* <article> является предком для любого другого элемента;
* <article> является родителем для <h1> и трёх <р>;
* <h1> и три <р> являются братскими друг для друга;
* каждый <р> является родителем для <strong> и <q>, которые в них содержатся;
* каждый <h1>, <p>, <strong> и <q> — это всё потомки <article>.

Аналогия с семейным древом также применяется при прохождении нескольких слоёв вложений в HTML:

* потомок элемента X — это любой элемент внутри X;
* ребёнок — это просто прямой потомок;
* предком элемента Y является любой элемент;
* родитель — это лишь первый прямой предок.

**Вложения блоков и строк**

Блочные элементы могут содержать блочные или строчные элементы.

Однако, строчные элементы могут содержать только другие строчные элементы (за исключением элемента <a>).

<!-- Это НЕВЕРНЫЙ код! :-( -->

<strong>

<p>Вы не можете поместить абзац внутрь тега strong.

</strong>

Просто запомните — предок/потомок, родитель/ребёнок, братья. Эта иерархия будет полезна в CSS.

**Проверочный тест:**

* 1. <article>  
     <h1>Астрономия</h1>  
     <section>  
     <h2>Далёкий Тукан</h2>  
     <p>Бесспорно, эффективный диаметр оценивает Тукан.</p>  
     </section>  
     </article>

Для данного кода, какой элемент является родителем <h2>?

* <h1>
* <section> +++
* <article>
* <p>
  1. (тот же код)Для данного кода, какой элемент является братским <h2>?

<h1>

<section>

<article>

<p> +++

* 1. Для данного кода, какой элемент является родителем <section>?
* <h1>
* <section>
* <article> +++
* <p>
  1. Для данного кода, какой элемент является братским <h1>?
* <p>
* <article>
* <h2>
* <section>+++
  1. <article>  
     <h1>Астрономия</h1>  
     <section>  
     <h2>Далёкий Тукан</h2>  
     <p>Бесспорно, эффективный диаметр оценивает Тукан.</p>  
     </section>  
     </article>

Для данного кода укажите всех предков для элемента <p>.

* <section>+++
* <h1>
* <article>+++
* <h2>
  1. <article>  
     <h1>Астрономия</h1>  
     <section>  
     <h2>Далёкий Тукан</h2>  
     <p>Бесспорно, эффективный диаметр оценивает Тукан.</p>  
     </section>  
     </article>

Для данного кода укажите все дочерние элементы у <section>.

* <p>+++
* <article>
* <h2>+++
* <h1>

**Семантика в HTML**

Целью тегов HTML является передача смысла документу. Не беспокойтесь о том, как ваша веб-страница выглядит. Сосредоточьтесь на значении каждого тега, который вы будете использовать.

В зависимости от написанного вами содержимого, вы можете выбрать подходящий элемент, соответствующий смыслу текста.

Диапазон элементов достаточно широк, чтобы он подходил и для материалов общего назначения (например, абзацы или списки) и для более конкретного содержимого, вроде <output> (для отображения результата вычисления) или <progress> (для отображения хода выполнения задачи).

**Структурные элементы: организация страницы**

Структурные элементы позволяют организовать основные части вашей страницы. Они обычно содержат другие элементы HTML.

Вот что типичная веб-страница может в себя включать:

* <header> в качестве первого элемента страницы, который может включать в себя логотип и слоган;
* <nav> в качестве списка ссылок, которые ведут на разные страницы сайта;
* <h1> в качестве заголовка страницы;
* <article> в качестве основного содержимого страницы, вроде статьи блога;
* <footer> в качестве последнего элемента страницы, расположенного внизу.

**Текстовые элементы: определение контента**

Внутри структурных элементов вы обычно находите текстовые элементы, призванные определить цель вашего содержимого.

Вы, в основном, будете использовать:

* <p> для абзацев;
* <ul> для (неупорядоченных) списков;
* <ol> для (упорядоченных) списков;
* <li> для отдельных пунктов списка;
* <blockquote> для цитат.

**Строчные элементы: различный текст**

Поскольку текстовые элементы могут быть длинными, но с разным содержанием, строчные элементы позволяют различатьчасти текста.

Есть много строчных элементов, но вы обычно столкнётесь со следующими:

* <strong> для важных слов;
* <em> для *выделенных* слов;
* <a> для [ссылок](https://webref.ru/course/html-basics/semantics);
* <small> для менее важных слов;
* <abbr> для аббревиатур, вроде W3C.

Просто читая этот код HTML, вы можете легко понять, что означает каждый элемент HTML.

* ***HTML***
* ***Результат***

***JSFiddle***

***<article>***

***<h1>Основной заголовок страницы</h1>***

***<h2>Подзаголовок</h2>***

***<p>***

***Какие-то всякие разные штуки и некоторые <em>выделенные</em> и***

***даже <strong>важные</strong> слова.***

***</p>***

***<p>***

***Другой абзац.***

***</p>***

***<ul>***

***<li>Один</li>***

***<li>Два</li>***

***<li>Три</li>***

***</ul>***

***<blockquote>***

***Однажды сказано***

***</blockquote>***

***</article>***

***<aside>***

***<h3>Мои последние сообщения</h3>***

***<ul>***

***<li><a href="#">Один</a></li>***

***<li><a href="#">Два</a></li>***

***<li><a href="#">Три</a></li>***

***</ul>***

***</aside>***

**Общие элементы**

Когда ни один семантический элемент не подходит для вашего содержимого, но вы всё ещё хотите вставить элемент HTML (в целях группирования или стилизации), то можете остановиться на одном из двух общих элементов:

* <div> для блочных элементов;
* <span> для строчных элементов.

Хотя эти элементы HTML на самом деле не несут какого-либо *смысла*, они пригодятся когда мы начнём использовать CSS.

**Не заморачивайтесь на семантике**

Существует около 100 семантических элементов HTML на выбор. Это много. Может оказаться непреодолимым пройтись по этому списку и выбрать соответствующий элемент для вашего контента.

Но не тратьте слишком много времени, беспокоясь об этом. Если вы будете придерживаться следующего списка на данный момент, этого будет достаточно.

|  |  |  |
| --- | --- | --- |
| Структурные | Текстовые | Строчные |
| <header> <h1> <h2> <h3> <nav> <footer> <article> <section> | <p> <ul> <ol> <li> <blockquote> | <a> <strong> <em> <q> <abbr> <small> |

Проверочный тест:

* 1. Для чего применяется элемент <small>
* Для второстепенного содержимого страницы
* Для выделения менее важных слов +++
* Для подзаголовка текста
* Для уменьшения размера текста
  1. Для чего применяется элемент <header>?
* Для заголовка страницы
* Для навигации по сайту
* Для информации об авторах
* Для первого элемента сайта +++
  1. Для чего применяется элемент <article>?
* Для абзаца текста
* В качестве универсального блока
* Для разделов сайта
* Для основного содержимого страницы +++
  1. Для чего применяется элемент <div>?
* В качестве универсального блока +++
* Для логотипа и слогана
* Для основного содержимого страницы
* Для указания разделов сайта
  1. Сопоставьте элемент и его роль
* <p> абзац
* <blockquote> цитата
* <ul> список
* <nav>навигация
* <a> ссылка
* <abbr> аббревиатура
  1. Сопоставьте элемент и его роль
     + Для важных слов <strong>
     + Для выделенных слов <em>
     + Для цитат<blockquote>
     + Для ссылок<a>
     + Для менее важных слов <small>
     + Для заголовков<h1>

**Форматирование HTML**

Существует разница между тем, что **написано** в вашем коде HTML и что **отображается** в браузере.

Как мы уже видели, **теги** HTML, такие как <р>, только читаются браузером (чтобы знать, какой *тип* содержимого написан), но **не отображаются** в нём.

Мы также видели, как можно писать **комментарии** в коде HTML, чтобы помочь человеку с чтением кода, без необходимости отображать эти комментарии в браузере.

Другим видом написанного кода, который *игнорируется* браузером, является **пробел**, он включает в себя:

* переносы строк;
* пустые строки;
* табуляция (или отступы).

**Переносы строк**

Переносы строк и пустые строки (которые представляют собой последовательность переносов строк) в коде HTML **игнорируются** браузером. Они составляют лишь **единое** пространство.

* ***HTML***
* ***Результат***

***JSFiddle***

***<blockquote>***

***Первоначальная идея веба была в том, что он должен быть совместным***

***пространством,***

***где вы можете общаться путём обмена информацией.***

***</blockquote>***

Чтобы реально **вставить** перенос строки вам нужно использовать элемент <br>:

* ***HTML***
* ***Результат***

***JSFiddle***

***<p>В лучшем случае, жизнь совершенно<br>непредсказуема</p>***

**Табуляция**

**Табуляция** — это специальный символ, полученный с помощью клавиши Tab. Она обычно перемещает курсор на следующую позицию табуляции, но иногда превращается в два пробела.

Во всяком случае, как и обычный пробел, табуляция **невидима**. Она также игнорируется браузером:

* ***HTML***
* ***Результат***

***JSFiddle***

***<p>***

***Давайте толкнём этот текст***

***табуляцией.***

***</p>***

Если вы хотите добавить пространство *перед* словом, вам придётся использовать CSS.

Если вы хотите закрыть элемент HTML, то должны сперва закрыть все его дочерние элементы.

**Форматирование в виде дерева**

Поскольку элементы HTML могут быть вложены друг в друга, вы должны следить за **порядком**, в котором они были открыты, так как это будет влиять на порядок, в котором они закрыты.

* ***HTML***
* ***Результат***

***JSFiddle***

***<article><p>Данный код написан в <strong>одну</strong> строку.</p></article>***

Из-за того, что может быть сложно следить за порядком, в котором были открыты элементы HTML, рекомендуется писать HTML в **виде дерева**:

* ***HTML***
* ***Результат***

***JSFiddle***

***<article>***

***<p>***

***Данный код написан в***

***<strong>несколько</strong>***

***строк, но тем не менее, будет***

***отображаться как***

***<em>одна</em>***

***строка.***

***</p>***

***</article>***

Форматирование в виде дерева позволяет *визуально* воспроизвести **уровень вложенности** вашего кода HTML. Это позволяет легко увидеть, что:

* <article> является **предком**;
* <p> — **родитель** для <strong> и <em>;
* <strong> и <em> — это **братские** элементы.

**Пишите HTML для себя, чтобы его читать**

Табуляция, пустые строки, набор пробелов и переносы строк опускаются компьютером и все они превращаются в **один пробел**.

HTML-документ пишется и читается человеком, но компьютером только читается. Учитывая, что табуляция, пробелы и переносы строк не влияют на то, как браузер будет читать и впоследствии отображать веб-страницу, вы можете отформатировать свой документ наиболее читаемым для **себя** способом.

Нет конкретных правил, касающихся форматирования HTML, но есть неявные **соглашения**, в частности:

* используйте **табуляцию**, чтобы помочь визуализировать **вложения** элементов HTML;
* вставляйте открывающие и закрывающие теги блочных элементов на **отдельные строки**;
* пишите строчные элементы в одну строку (включая открывающие и закрывающие теги).

Проверочный тест:

1. Какой элемент используется для переноса строк?
   * <br>+++
   * <p>
   * <div>
   * <tab>
2. Как рекомендуется писать теги блочных элементов в коде HTML?
   * <div>Содержимое

</div>

* + <div>Содержимое</div>
  + <div>

Содержимое

</div>+++

* + <div>

Содержимое</div>

1. Как рекомендуется писать теги строчных элементов в коде HTML?
   * <span>

Содержимое</span>

* + <span>Содержимое</span> +++
  + <span>

Содержимое

</span>

* + <span>Содержимое

</span

1. Что из перечисленного в коде HTML браузер будет игнорировать?

* Табуляция.+++
* Перенос строки. +++
* Комментарии. +++
* Текcт.
* Списки.
* Изображения.
* Несколько пробелов подряд. +++

1. Выберите неверное утверждение
   * Форматирование кода позволяет увидеть отношения элементов
   * Форматирование кода позволяет следить за порядком элементов html
   * Форматирование кода нужно чтобы сделать документ более читаемым
   * Форматирование кода позволяет не вкладывать одни элементы внутрь других+++

**Валидный HTML-документ**

До сих пор мы рассматривали отдельные фрагменты кода HTML. Но HTML-документ (или веб-страница, что означает то же самое) требует определённой структуры для того, чтобы стать валидным.

Почему мы заботимся о валидации HTML-документа?

* Правильность: валидный документ корректно отображается в браузере.
* Отладка: некорректный код HTML может вызвать ошибки, сложные для выявления.
* Поддержка: валидный документ легче обновлять позже, даже кому-то другому.

**Доктайп**

Первой информацией которую мы пишем, является *тип* HTML-документа — доктайп.

Подумайте о доктайпе как о версии автомобиля на протяжении многих лет: Ford Fiesta, купленный в 1986 году, был Fiesta 2. Если вы покупаете его сегодня, то это Fiesta 7.

Раньше сосуществовало несколько версий HTML (XHTML и HTML 4.01 были конкурирующими стандартами). В настоящее время нормой является HTML5.

Чтобы сообщить браузеру, что HTML-документ представляет собой HTML5, просто начните свой документ со следующей строки:

<!DOCTYPE html>

Вот и всё. Просто установите и забудьте об этом.

Вы можете задаться вопросом, почему это доктайп HTML5 не упоминает цифру 5. В W3C подумали, что предыдущие определения доктайпа были слишком запутанными и воспользовались возможностью, чтобы упростить его, удалив упоминание о версии HTML.

**Элемент <html>**

Помимо строки с доктайпом, весь ваш HTML-документ должен располагаться внутри элемента <html>:

<!DOCTYPE html>

<html>

<!-- Здесь остальная часть вашего кода HTML -->

</html>

<html> технически является предком всех элементов HTML.

<head>

Как атрибуты несут дополнительную информацию для элемента HTML, так и элемент <head> несёт дополнительную информацию для всей веб-страницы.

Например, заголовок страницы (отображается на вкладке) находится в <head>:

<head>

<title>Мой сказочный блог</title>

</head>

Следующие элементы HTML могут появляться в <head> и только в <head>:

* <link>
* <meta>
* <style>

<body>

В то время как <head> содержит только метаданные, не предназначенные для отображения вообще (за исключением <title>), то элемент <body> это место, где мы пишем всё наше содержимое. Всё внутри <body> будет отображаться в окне браузера.

Полностью валидный HTML-документ

Объединив все эти требования, мы можем написать простой и валидный HTML-документ:

* ***HTML***
* ***Результат***

***JSFiddle***

***<!DOCTYPE html>***

***<html>***

***<head>***

***<meta charset="utf-8">***

***<title>MarkSheet</title>***

***<meta name="description" content="Простое руководство по HTML">***

***</head>***

***<body>***

***<p>Привет, мир!</p>***

***</body>***

***</html>***

Если вы просмотрите этот пример в браузере, то увидите, что:

* «MarkSheet» написано на вкладке браузера;
* «Привет, мир!» — это единственный текст, отображаемый в окне, потому что это единственное содержимое <body>.

W3C предлагает [Markup Validation Service](http://validator.w3.org/" \l "validate_by_input) для проверки любого HTML-документа на наличие ошибок и предупреждений.

**Проверочный тест:**

* + 1. **Для чего применяется доктайп?**
       - Для указания типа документа +++
       - Сообщает, откуда должен начинаться код html
       - Хранит метаданные документа
       - Для отображения содержимого веб страницы
    2. Для чего предназначен элемент <head>?
       - В качестве первого элемент для логотипа и слогана
       - Для отображения содержимого веб страницы
       - Для хранения метаданных +++
       - Для заголовка веб страницы
    3. Для чего предназначен элемент <body>?
       - Для основного содержания страницы, вроде статьи блога
       - Для хранения всего содержимого веб страницы +++
       - Хранит метаданные документа
       - В качестве универсального блочного элемента
    4. Содержимое какого элемента отображается на веб странице?
       - <title> +++
       - <head>
       - <link>
       - <meta>
    5. Расставьте в правильном порядке

<!DOCTYPE html>

<html>

<head>

<title>Заголовок</title>

</head>

<body>

<p>Привет, мир!</p>

</body>

</html>

* + 1. Укажите все элементы допустимые в <head>

<header>

<meta>+++

<style>+++

<p>

<link>+++

<abbr>

Проверка знаний:

Вопрос 1

Сопоставьте друг с другом элементы.

|  |  |  |
| --- | --- | --- |

| **Вариант** | **Правильно?** | **Правильный ответ** |
| --- | --- | --- |
| <a> | Правильно | Открывающий тег |
| </a> | Правильно | Закрывающий тег |
| <img> | Правильно | Самозакрывающий элемент |
| <a>...</a> | Правильно | Элемент HTML |

Вопрос 2

Сопоставьте элемент и его назначение.

|  |  |  |
| --- | --- | --- |

| **Вариант** | **Правильно?** | **Правильный ответ** |
| --- | --- | --- |
| <strong> | Правильно | Для важных слов |
| <em> | Правильно | Для выделенных слов |
| <small> | Правильно | Для менее важных слов |
| <abbr> | Правильно | Для аббревиатур |

Вопрос 3

Сопоставьте элемент и его назначение.

|  |  |  |
| --- | --- | --- |

| **Вариант** | **Правильно?** | **Правильный ответ** |
| --- | --- | --- |
| <head> | Правильно | Хранит метаданные HTML-документа |
| <body> | Incorrect | Хранит всё содержимое веб-страницы |
| <!DOCTYPE html> | Incorrect | Указывает, что HTML-документ представляет собой HTML5 |
| <html> | Правильно | Предок всех элементов HTML |

Вопрос 4

Сопоставьте элемент и его тип.

| **Вариант** | **Правильно?** | **Правильный ответ** |
| --- | --- | --- |

| **Вариант** | **Правильно?** | **Правильный ответ** |
| --- | --- | --- |
| <span> | Incorrect | Строчный элемент |
| <table> | Incorrect | Таблица |
| <td> | Incorrect | Ячейка таблицы |
| <div> | Правильно | Блочный элемент |

Вопрос 5

Сопоставьте элемент и его назначение.

|  |  |  |
| --- | --- | --- |

| **Вариант** | **Правильно?** | **Правильный ответ** |
| --- | --- | --- |
| <ol> | Incorrect | Упорядоченный список |
| <ul> | Правильно | Неупорядоченный список |
| <blockquote> | Incorrect | Цитата |
| <footer> | Incorrect | Подвал страницы |

Вопрос 6

Сопоставьте элемент и его назначение.

|  |  |  |
| --- | --- | --- |

| **Вариант** | **Правильно?** | **Правильный ответ** |
| --- | --- | --- |
| <nav> | Incorrect | Навигация по сайту |
| <h1> | Правильно | Заголовок страницы |
| <p> | Правильно | Абзац текста |
| <article> | Incorrect | Основное содержимое |