## Разметка текста с помощью HTML

|  |
| --- |
| [Абзацы](https://npoed.htmlacademy.ru/htmlcss110/course/3/run/1) |
| [Заголовки и подзаголовки](https://npoed.htmlacademy.ru/htmlcss110/course/3/run/2) |
| [Неупорядоченный список](https://npoed.htmlacademy.ru/htmlcss110/course/3/run/3) |
| [Упорядоченный список](https://npoed.htmlacademy.ru/htmlcss110/course/3/run/4) |
| [Многоуровневый список](https://npoed.htmlacademy.ru/htmlcss110/course/3/run/5) |
| [Ещё более многоуровневый](https://npoed.htmlacademy.ru/htmlcss110/course/3/run/6) |
| [Список определений](https://npoed.htmlacademy.ru/htmlcss110/course/3/run/7) |
| [Важность. Теги strong и b](https://npoed.htmlacademy.ru/htmlcss110/course/3/run/8) |
| [Акцентируем внимание. Теги em и i](https://npoed.htmlacademy.ru/htmlcss110/course/3/run/9) |
| [Переносы и разделители. Теги br и hr](https://npoed.htmlacademy.ru/htmlcss110/course/3/run/10) |
| [Цитаты](https://npoed.htmlacademy.ru/htmlcss110/course/3/run/11) |
| [Верхние и нижние индексы](https://npoed.htmlacademy.ru/htmlcss110/course/3/run/12) |
| [Помечаем изменения. Теги del и ins](https://npoed.htmlacademy.ru/htmlcss110/course/3/run/13) |
| [Преформатированный текст](https://npoed.htmlacademy.ru/htmlcss110/course/3/run/14) |
| [Просто выделенный текст](https://npoed.htmlacademy.ru/htmlcss110/course/3/run/15) |
| [Испытание: разметка статьи](https://npoed.htmlacademy.ru/htmlcss110/course/3/run/16) |
| [Испытание: рецепт](https://npoed.htmlacademy.ru/htmlcss110/course/3/run/17) |

# Абзацы [1/17]

В разделе «Структура HTML-документа» вы познакомились с тегами, необходимыми для создания простейшей HTML-страницы, и с некоторыми служебными тегами, которые не отображаются в браузере.

В этом разделе мы будем изучать теги для логической разметки текста. Использовать их можно только внутри тега <body>.

Начнём с простейшего тега <p>, с помощью которого создаются абзацы. По умолчанию абзацы начинаются с новой строки и имеют вертикальные отступы, которыми можно управлять с помощью стилей.

# Заголовки и подзаголовки [2/17]

Для создания структуры больших текстов обычно используются заголовки. В текстовых редакторах есть возможность выделить часть текста, найти пункт «Заголовок» нужного уровня в меню, и применить его.

В языке HTML для выделения заголовков предусмотрено целое семейство тегов: от <h1> до <h6>. Тег <h1> обозначает самый важный заголовок (заголовок верхнего уровня), а тег <h6> обозначает подзаголовок самого нижнего уровня.

На практике редко встречаются тексты, в которых встречаются подзаголовки ниже третьего уровня. Поэтому самыми часто используемыми тегами заголовков являются: <h1>, <h2> и <h3>.

Стоит отметить, что поисковые системы придают особое значение заголовкам, поэтому необходимо учиться правильно их использовать.

# Неупорядоченный список [3/17]

Списки часто используются в различных документах. Иногда, чтобы сделать список, пользователь просто нумерует строчки текста. Такой подход не является хорошим, так как в документе отсутствует логическая сущность «список».

В HTML существует семейство тегов для создания списков: неупорядоченных, упорядоченных и списков определений. В последующих заданиях мы будем тренироваться работать с ними.

Неупорядоченные (или маркированные) списки создаются с помощью тега <ul>, который может содержать внутри себя теги <li>, обозначающие «элемент списка».

# Упорядоченный список [4/17]

Упорядоченный список создаётся с помощью тега <ol>, который может содержать внутри себя теги <li>.

Если элементы неупорядоченного списка по умолчанию отмечаются маркерами, то элементы упорядоченного списка — нумеруются.

Для упорядоченного списка можно задать атрибут start, который изменяет начало нумерации. Например, код:

<ol start="3">

<li>раз</li>

<li>два</li>

</ol>

Приведёт к такому результату:

1. раз
2. два

# Многоуровневый список [5/17]

Создать многоуровневый список достаточно просто.

Сначала нужно создать список первого уровня, а затем внутрь любого элемента этого списка, между тегами <li> и </li>, добавить список второго уровня. При этом необходимо аккуратно закрывать все теги.

Пример правильного кода:

<ul>

<li>1

<ul>

<li>1.1</li>

<li>1.2</li>

</ul>

</li>

<li>2</li>

</ul>

Пример кода с ошибкой:

<ul>

<li>1</li>

<ul>

<li>1.1</li>

<li>1.2</li>

</ul>

<li>2</li>

</ul>

В примере с ошибкой вложенный список вставлен не внутрь элемента списка, а между элементами, что недопустимо.

Количество уровней в списках не ограничено. В многоуровневом списке можно использовать как упорядоченные, так и неупорядоченные списки.

В этом задании мы потренируемся работать с многоуровневыми списками.

# Многоуровневый список [5/17]

Создать многоуровневый список достаточно просто.

Сначала нужно создать список первого уровня, а затем внутрь любого элемента этого списка, между тегами <li> и </li>, добавить список второго уровня. При этом необходимо аккуратно закрывать все теги.

Пример правильного кода:

<ul>

<li>1

<ul>

<li>1.1</li>

<li>1.2</li>

</ul>

</li>

<li>2</li>

</ul>

Пример кода с ошибкой:

<ul>

<li>1</li>

<ul>

<li>1.1</li>

<li>1.2</li>

</ul>

<li>2</li>

</ul>

В примере с ошибкой вложенный список вставлен не внутрь элемента списка, а между элементами, что недопустимо.

Количество уровней в списках не ограничено. В многоуровневом списке можно использовать как упорядоченные, так и неупорядоченные списки.

В этом задании мы потренируемся работать с многоуровневыми списками.

# Ещё более многоуровневый [6/17]

Хорошо. Вы создали двухуровневый список. Теперь задание посложнее.

В этом задании вам нужно будет создать четырёхуровневый список, наподобие этого:

1. Разметка
   1. Основы HTML
      1. HTML-теги
         1. парные
         2. одиночные
   2. Основы CSS
      1. Селекторы
         1. по типу
         2. по классу
         3. вложенные
   3. Стиль кодирования
2. Работа с фотошопом
3. Построение сеток
4. Декоративные элементы
5. Введение в JavaScript
6. Прогрессивное улучшение

# Список определений [7/17]

Список определений создаётся с помощью трёх тегов:

1. <dl> обозначает сам список определений;
2. <dt> обозначает термин;
3. <dd> обозначает определение термина.

Теги <dt> и <dd> пишутся парами внутри <dl>.

Например:

<dl>

<dt>Термин</dt>

<dd>Определение</dd>

<dt>Второй термин</dt>

<dd>И его определение</dd>

<dt>Кошка</dt>

<dd>Шерстяное изделие развлекательного характера</dd>

</dl>

# Важность. Теги strong и b [8/17]

Ещё раз отметим, что этот раздел посвящён **логической** разметке текста, поэтому уделяется особое внимание смыслу элементов, их предназначению, а не визуальному форматированию.

В предыдущих заданиях вы познакомились с элементами, которые предназначены для разметки крупных блоков текста: заголовков, абзацев и списков. В этом и последующих заданиях мы познакомимся с элементами, предназначенными для разметки небольших фраз и отдельных слов.

Первые два тега предназначены, чтобы указать на важность слова или фразы.

Тег <strong> определяет **важность** отмеченного текста.

Тег <b> предназначен для выделения текста без придания ему особой важности.

Лучше всего отличия этих тегов будут заметны людям, которые используют специальные настройки ОС, в частности, слепым и слабовидящим. Когда они включают функцию чтения текста, то «говорилка» будет интонацией выделять слова с тегом <strong>. То же самое касается и тегов <em> и <i>. Тег <em> «говорилка» будет выделять интонацией.

Визуально оба тега одинаковы, они выделяют текст полужирным.

Отметим, что новый смысл тегу <b> придали в HTML5. Раньше это был тег, который просто делает текст полужирным. То есть он был предназначен только для визуального форматирования.

# Акцентируем внимание. Теги em и i [9/17]

Следующие два тега предназначены для акцентирования внимания на слово или фразу.

Тег <em> определяет текст, на который сделан особый акцент, меняющий смысл предложения.

Например, если мы хотим подчеркнуть, что Кекс не любит питаться укропом (он больше за тунца), а любит только гонять его по полу, то разметим текст так:

Инструктор Кекс любит <em>играть</em> с укропом.

Тег <i> обозначает текст, который отличается от окружающего текста, но не является более важным. Обычно так выделяют *названия*, *термины*, *иностранные слова*.

Например, если мы хотим указать, что *инспектор* — это какой-то специальный термин, то разметим текст так:

Обычно Кекс пользовался <i>инспектором</i> браузера для поиска ошибок.

Визуально оба тега одинаковы, они выделяют текст курсивом.

Новый смысл тегу <i> придали в HTML5. Раньше это был просто тег для выделения текста курсивом.

# Переносы и разделители. Теги br и hr [10/17]

Иногда возникает необходимость вставить в текст перенос строки, не создавая при этом абзац. Например, при разметке стихов или текстов песен.

Для этого в HTML предусмотрен одиночный тег <br>.

Иногда этот тег используется для разбиения текста на «как бы абзацы», что является плохим подходом. Используйте для разметки абзацев тег <p>.

Одиночный тег <hr> используется для того, чтобы создать горизонтальную линию-разделитель. На внешний вид этой линии можно влиять с помощью атрибутов, но правильней делать это с помощью CSS (это будет изучаться в последующих разделах).

# Переносы и разделители. Теги br и hr [10/17]

Иногда возникает необходимость вставить в текст перенос строки, не создавая при этом абзац. Например, при разметке стихов или текстов песен.

Для этого в HTML предусмотрен одиночный тег <br>.

Иногда этот тег используется для разбиения текста на «как бы абзацы», что является плохим подходом. Используйте для разметки абзацев тег <p>.

Одиночный тег <hr> используется для того, чтобы создать горизонтальную линию-разделитель. На внешний вид этой линии можно влиять с помощью атрибутов, но правильней делать это с помощью CSS (это будет изучаться в последующих разделах).

# Цитаты [11/17]

В HTML существует несколько тегов для обозначения цитат:

* <blockquote> предназначен для выделения длинных цитат, которые могут состоять из нескольких абзацев. Тег выделяет цитату как отдельный блок текста с отступами.
* <q> предназначен для выделения коротких цитат в тексте предложения. Текст внутри этого тега автоматически обрамляется кавычками.
* <cite> используется для того, чтобы выделить источник цитаты, название произведения, но не автора цитаты.

# Верхние и нижние индексы [12/17]

Следующие два тега обычно используются не для выделения слов, а для выделения отдельных символов. Их используют для указания единиц измерения или для написания простых формул.

Например: 20м2, H2O, X3+X2=1

Тег <sup> отображает текст в виде верхнего индекса.

Тег <sub> отображает текст в виде нижнего индекса.

Стоит отметить, что эти теги являются чисто презентационными и не имеют собственной семантики.

Эти теги можно использовать внутри друг друга для создания более сложных формул.

Если вам нужно вставить очень сложную формулу в HTML-документ, лучше воспользоваться специальным языком разметки [MathML](http://ru.wikipedia.org/wiki/MathML" \t "_blank).

# Помечаем изменения. Теги del и ins [13/17]

Любой документ на протяжении своей «жизни» может изменяться. С распространением динамических веб-приложений вносить изменения в HTML-документы стало проще простого.

Иногда возникает вопрос: а что же именно было изменено в документе, что было добавлено, а что удалено?

Как раз для описания изменений предназначены теги <del> и <ins>.

<del> выделяет текст, который был удалён в новой версии документа.

<ins> выделяет текст, который был добавлен в новой версии документа.

Оба тега имеют атрибут datetime, в котором можно указать дату и время, когда была внесена та или иная правка.

Простейшим примером применения этих тегов может служить список ошибок. Когда ошибка исправлена, её помечают тегом <del>, если найдена новая ошибка, то её добавляют в список и помечают тегом <ins>.

Атрибут datetime предназначен не для людей, а для компьютеров, поэтому дату и время там пишут в стандартизованном формате. При такой разметке программам легче разбирать документы и анализировать, когда произошли те или иные изменения.

# Преформатированный текст [14/17]

Наверное, вы уже заметили, как отличается отображение кода в HTML-редакторе и в мини-браузере.

Вы можете ставить сколько угодно пробелов в HTML-коде, но браузер отобразит их как один. Вы также можете ставить сколько угодно переносов строки в HTML-коде, а в браузере переноса не будет, если только не использовать специальные теги, например <p> или <br>.

Изменить это поведение браузера можно с помощью тега <pre>, который обозначает «предварительно отформатированный текст». Браузер сохраняет и отображает все пробелы и переносы, которые есть внутри тега <pre>.

Наиболее часто этот тег используется при отображении примеров кода.

# Просто выделенный текст [15/17]

В HTML5 появился новый тег <mark>, который обозначает выделенный текст.

Иногда при работе с объёмными текстами мы используем маркер, чтобы выделять ключевые слова, идеи или что-то другое на что стоит обратить внимание. Такое же назначение и у тега <mark>.

В современных браузерах текст внутри <mark> подсвечивается жёлтым фоном.

# Испытание: разметка статьи [16/17]

Чтобы пройти испытание, вам необходимо разметить текст в HTML-редакторе точно так же, как и в образце.

При создании образца были использованы только те HTML-теги, которые изучались в данном курсе. CSS-стили не использовались, так что изменять код CSS-редактора нельзя.

Если отображение в мини-браузере сильно отличается от образца, то используйте вкладку «Различия», в которой отображается то, как наш сервер видит ваш HTML-код.

# Испытание: рецепт [17/17]

Задача аналогична предыдущей. В этом испытании использованы другие теги из данного раздела.