

Лабораторная работа № 1. Знакомство с HTML5

(трудоемкость 2 часа)

Цель: изучить структуру HTML документа; принципы создания списков, таблиц, гиперссылок; особенности применения тегов `` и `<div>`.

Порядок выполнения лабораторной работы:

1. Изучить основные теги HTML5 (см. приложения, лекцию, специальную литературу, on-line справочники)
2. Выполнить задание согласно своему варианту¹.
3. Оформить отчет, содержащий²:
 - титульный лист;
 - цель работы;
 - краткое описание хода выполнения работы с листингом кода html-страниц.

¹ Вариант индивидуального задания можно как выбрать из предложенного преподавателем, так и предложить свой. Во избежание путаницы, фамилии студентов и выбранные варианты фиксируются в документе с общим доступом.

² Для студентов, сдающих работу в указанный преподавателем срок, допускается сдача отчета в электронном виде.

Задания к лабораторной работе № 1

Используя данные из теоритической части лабораторной работы создайте набор связанных документов (сайт), отвечающим следующим требованиям:

- На сайте должны присутствовать изображения, как средства иллюстрации.
- На сайте должны быть использованы списки различных типов.
- На сайте должны содержаться табличные данные.
- Каждая страница сайта должна содержать разбитые на параграфы текстовые данные. (не менее 5 параграфов на каждую страницу)
- Станицы сайта должны иметь ссылки друг на друга.
- Сайт должен содержать не менее трех страниц.

Тематика вашего сайта должна совпадать с вашим вариантом.

Варианты

1. Личная информация о студенте (Личные данные, Интересы, Достижения и т.п.)
2. Сайт вашей группы (Описание специальности, Список группы, Список дисциплин изучаемых в этом семестре и т.п.)
3. Сайт посвященный вашему хобби (Описание увлечение, описание ваших достижений, портфолио и т.п.)
4. Сайт посвященный отдельным комплектующим вашего или учебного компьютера (Описание монитора, описание системного блока, описание процессора, видеокарты, мышки, ауди системы и т.п.)
5. Сайт туристической фирмы (Описание фирмы, перечень туров, отзывы и т.п.)
6. Сайт вашей школы (О школе, список классов, список учителей, факультативы и т.п.)
7. Сайт вашего города (О городе, достопримечательности, история и т.п.)
8. Сайт студенческого кафе (Описание кафе, меню, график работы и т.п.)
9. Сайт посвящённый вашему питомцу (Описание питомца, фотогалерея, умения и т.п.)
10. Сайт вашего вуза (Описание ВУЗа, контакты, список институтов и т.п.)

Приложение 1

Источник: <https://zarabotat-na-sajte.ru/uroki-html/osnovnie-tegi-html.html>

Оглавление

1. [<p></p> — создание абзацев на странице](#)
2. [и — выделение жирным](#)
3. [<i> — курсивный шрифт](#)
4. [<u> — подчеркнутый шрифт](#)
5. [<a> — создание гиперссылок](#)
6. [<h1> — заголовки в контенте](#)
7. [<center> — выравнивание по центру](#)
8. [<sub> — подстрочный шрифт](#)
9. [<sup> — надстрочный шрифт](#)
10. [<big>, <small> — уменьшить/увеличить размер шрифта](#)
11. [— создание списка](#)
12. [<table> — создание таблиц](#)
13. [
 — перенос строки](#)
14. [<hr> — горизонтальная линия](#)
15. [— вывод изображения](#)
16. [— для форматирования текста](#)
17. [<form> — вывод форм](#)
18. [<div> — блочный элемент](#)
19. [<!--ком--> — комментарии в коде](#)

Список тегов html

1. HTML тег <p></p> (для абзацев)

<p></p> — выводит текстовый абзац (допускает атрибуты style, class, id). Самый распространенный тег, поскольку в него чаще всего помещают текст (впрочем он для этого и создан).

Например, html код:

```
<p>Текстовый абзац номер один</p>
<p>А это другой абзац</p>
```

Преобразуется на странице в следующее:

Текстовый абзац номер один
А это другой абзац

К тегу можно ещё приписать параметр style:

```
<p style="стиль"></p>
```

С помощью этих различных значений можно редактировать внешний вид шрифта. Про эти параметры можно прочитать в отдельном уроке: [стили в html](#) и [свойство css font](#).иц

Также можно прописать атрибуты **CLASS** и **ID**. Например:

```
<p class="style1"></p>
<p id="id_style1"></p>
<p class="style1" id="id_style1"></p>
```

2. HTML тег **** и **** (выделение жирным)

**** и **** — два тега, которые позволяют сделать шрифт жирным. Разницы между этими тегами нет.

Приведем пример. Html код:

```
<b>жирный текст</b>
```

Преобразуется на странице в следующее:

жирный текст

Также можно прописать атрибуты CLASS и ID (как и в случае с <p>).

Примечание

Эти теги оказывают небольшое влияние на ранжирование документов в поисковых системах, поэтому бесцельно их лучше не использовать.

Читайте также: [как сделать жирный шрифт html](#)

3. HTML тег **<i>** (создание курсива)

<i></i> — курсивный шрифт (допускает параметр style, class, id)

Например, html код:

```
<i>курсивный текст</i>
```

Преобразуется на странице в следующее:

курсивный текст

Читайте также: [как сделать курсивный текст в html](#)

4. HTML тег **<u>** (подчеркнутый текст)

<u></u> — подчеркнутый шрифт (допускает параметр style, class, id)

Например, html код:

`<u>подчеркнутый текст</u>`

Преобразуется на странице в следующее:

подчеркнутый текст

Читайте также: [как сделать подчеркнутый текст html](#)

5. HTML тег `<a>` (создание гиперссылки)

`` — создает ссылку на странице (допускает параметр `style`, `class` и другие).

Например, html код:

`текст ссылки`

Преобразуется на странице в следующее:

[текст ссылки](#)

Все параметры и атрибуты тега `<a>` будут рассмотрены в отдельном уроке: [html тег `<a>`](#).

6. HTML тег `<h1>` (заголовки в контенте)

`<h1></h1>`, ..., `<h6></h6>` — заголовочные теги внутри контента (допускает параметр `style`, `class`, `id`). Причем чем меньше цифра, тем больший вес и размер (по умолчанию) имеет текст, заключенный в эти теги.

Например, html код:

`<h1>Заголовок h1</h1>`

Тег `<h1>` используют для названия страницы (также как и тайтл)

Необходимо использовать эти теги только по назначению, т.е. только когда в статье необходим заголовок. Это связано с тем, что теги `<h1></h1>`, ..., `<h6></h6>` имеют большое влияние на поисковые машины. Если их использовать правильно, то шансы попасть на верхние строчки выдачи очень высоки.

Более подробно про эти теги читайте в 14 уроке: [HTML теги h1-h6](#)

7. HTML тег `<center>` (выравнивание)

`<center></center>` — выравнивает контент по центру.

Например, html код:

`<center>`Этот текст будет в центре`</center>`

Преобразуется на странице в следующее:

Этот текст будет в центре

Примечание

Рекомендуется не использовать тег `<center></center>`, а вместо него использовать такие конструкции:

- `<p style="text-align:center">...</p>` — для текста
- `<div style="text-align:center">...</div>` — для всего (например, изображение)

8. HTML тег `<sub>` (подстрочный текст)

`` — выводит подстрочный шрифт.

Например, html код:

Обычный текст, `_{`подстрочный текст`}`

Преобразуется на странице в следующее:

Обычный текст, подстрочный текст

9. HTML тег `<sup>` (надстрочный текст)

`` - выводит надстрочный шрифт.

Например, html код:

Обычный текст, `^{`надстрочный текст`}`

Преобразуется на странице в следующее:

Обычный текст, надстрочный текст

10. HTML тег `<big>`, `<small>`

`<big></big>`, `<small></small>` — выводит шрифт на один пиксель больше/меньше текущего размера (допускает параметр style, class, id).

Например, html код:

Обычный шрифт, `<big>`этот шрифт больше на один пиксель`</big>`

Преобразуется на странице в следующее:

Обычный шрифт, этот шрифт больше на один пиксель

11. HTML тег (создание списков)

... — выводит список (допускает параметр style, class, id). Каждый новый элемент записывается между **** и ****.

Например, html код:

Список:

```
<ul>
  <li>первый элемент списка</li>
  <li>второй элемент списка</li>
</ul>
```

Преобразуется на странице в следующее:

Список:

- первый элемент списка
- второй элемент списка

Более подробно про списки читайте в **приложении 2**

12. HTML тег <table> (создание таблиц)

<table></table> — создает таблицу (допускает параметр style, class). Каждая новая строка создается тегами **<tr></tr>**, а столбец **<td></td>**.

Например, html код:

```
<table border="1">
<tr> <td>1-строка 1 элемент</td> <td>1-строка 2 элемент</td> </tr>
<tr><td>2-строка 1 элемент</td><td>2-строка 2 элемент</td></tr>
</table>
```

Преобразуется на странице в следующее:

1-строка 1 элемент	1-строка 2 элемент
1-строка 1 элемент	1-строка 2 элемент

Все возможности тега **<table>** будут рассмотрены в **приложении 3**

13. HTML тег
 (перенос строки)

**
** — переход на следующую строку, представляет собой одиночный тег.

Например, html код:

Строка 1

**
**2-строка

**
**3-строка

А этот текст будет на 3ей строке, поскольку перехода не было

Преобразуется на странице в следующее:

1-Строка

2-строка

3-строка А этот текст будет на 3ей строке, поскольку перехода не было

14. HTML тег <hr> (горизонтальная линия)

<hr/> — чертит линию, представляет собой одиночный тег (допускает параметр style, class).

Например, html код:

Какой-то текст над линией

<hr>

А этот текст будет уже под линией

Преобразуется на странице в следующее:

Какой-то текст над линией

А этот текст будет уже под линией

15. HTML тег (вывод картинки)

**** — выводит изображение, представляет собой одиночный тег (допускает параметр style, class).

Например, html код:

Преобразуется на странице в следующее:



Все возможности тега **** будут рассмотрены в отдельном уроке: [html тег](#).

16. HTML тег **** (форматирование текста)

**** — для форматирования текста, изменения стиля и т.п. (допускает параметр `style`, `class`, `id`). Его вес в глазах поисковых машин отсутствует, поэтому его можно использовать сколько хотите.

Например, html код:

```
<font style="color:green; font-size:17px;">Этот текст зеленый,  
а его размер 15 пикселей</font>
```

Преобразуется на странице в следующее:

Этот текст зеленый, а его размер 17 пикселей

Примечание

Аналогичным тегом является ****.

Более подробно с `font` можете ознакомиться в статье: [тег](#)

17. HTML тег **<form>** (создание формы)

<form></form> — создание формы на странице (допускает параметр `style`, `class`).

Например, ввод логина и пароля, любые кнопки, любая форма регистрации — все это формы.

Подробное описание этого тега читайте: [тег <form> — создание html форм](#).

18. HTML тег <div> (создание блоков)

<div></div> — служит для создания блоков на странице (допускает параметр style, class). Раньше для разметки на страницах пользовались в основном таблицами. После появления тега **<div>** задача упростилась. Практически все сайты содержат **<div>** блоки, как удобную альтернативу таблицам.

Подробное описание этого тега читайте в **приложении 4**.

19. HTML тег <!--ком--> (комментарии)

<!--комментарий--> — служит для добавления [комментариев в коде html](#). Все, что заключено между <!-- и --> является комментарием и не отображается на интернет странице.

Комментарии html очень удобны и позволяют упростить код, делая его более понятным. Приведем простой пример:

```
<body>

    <!--верхушка сайта-->

    <div class="top">Здесь содержится верхушка сайта</div>

    <!--конец верхушки сайта-->


    <!--левое меню-->

    <div class="leftblock">Левая часть сайта, здесь может быть
        различное меню, реклама и т.д.</div>

    <!--конец левого меню-->


    <!--контент страницы-->

    <div class="content">Содержательная часть страницы</div>

    <!--конец контента страницы-->

</body>
```

В html есть ещё пару важных тегов, о которых выше не было рассказано. Я имею ввиду [заголовочные теги](#), однако на данном уровне Ваших познаний, мне кажется, что их изучение пока преждевременно. Для начала нужно разобраться с базовыми тегами.

Уважаемый читатель, теперь Вы имеете представление о html тегах. В следующих уроках мы рассмотрим самые важные из этих тегов во всех подробностях. Теперь советую перейти к следующим урокам.

Пример создания html страницы в блокноте³

В [первом уроке](#) мы уже рассмотрели маленький пример создания html-страницы. Теперь создадим более сложную страничку. Хочу отметить, что я буду объяснять как это сделать без применения специальных программных средств. Все, что понадобится для работы — это обычный блокнот от Windows. Для более удобной работы рекомендую [скачать Notepad++](#) (продвинутый блокнот с возможностью подсвечивания html-тегов). Итак, начнем.

Откройте блокнот и скопируйте в него следующее:

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <title>Главная страница - страница обо мне</title>
</head>
<body>
<center><h1>Информация обо мне</h1></center>
Краткая биография
Родился в 1985 году в городе Москва. Закончил в 2008 году МАИ.
На данный момент работаю ведущим инженером в крупной авиакомпании.
Моя мечта стать ведущим по машине.

Поскольку я люблю авиацию, то хотел бы поделиться несколькими интересными
фотографиями на эту тему
<br/><br/>
<center>
</center>
<br/><br/>
<font style="color:green">Этот текст зеленый</font>
<br/><br/>
<b>Просто пример жирного текста</b>
<br/><br/>
Низ страницы
<br/><br/>
В данном примере мы рассмотрели кратко основные теги HTML, теперь можно пробовать
создавать
```

³ Можно использовать любой удобный Вам текстовый редактор

несколько связанных страниц через ссылки и выкладывать сайт в интернет.

```
<hr>
```

Этот материал был написан благодаря сайту

```
<a href=http://zarabotat-na-sajte.ru/>
```

http://zarabotat-na-sajte.ru/ - за что я ему благодарен.

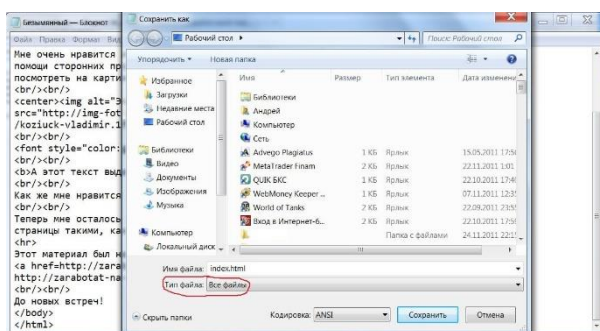
```
<br/><br/>
```

Спасибо. До новых встреч!

```
</body>
```

```
</html>
```

Далее нажмите "сохранить как", в поле тип файла выберите "все файлы", а в названии напишите **index.html**. Обязательно в конце названия должно быть расширение .html (не .txt), иначе браузеры не будут интерпретировать его как веб-документ.



Если по каким-то причинам у Вас не получается создать html страницу, то Вы можете скачать предыдущий пример по ссылке: [index.rar](#).

Теперь пару слов о тегах, которые мы использовали, чтобы сделать эту страницу.

Описание html тегов из примера

1. **<html></html>** — эти теги должны присутствовать на каждой веб-странице обязательно. Они сообщают браузерам и поисковым машинам, что это html-страница.

Любая html страница имеет следующую структуру:

```
<html>
```

```
<head>
```

```
...
```

Заголовочные теги

```
...
```

```
</head>
```

```
<body>
```

```
...
```

Тело страницы

```
...  
</body>  
</html>
```

2. **<body></body>** — между этими тегами заключается весь видимый контент страницы.

3. **<head></head>** — внутри этих тегов должны располагаться все заголовочные теги. Этот раздел можно опустить, но я не советую это делать, поскольку это важная часть документа, особенно для поисковых систем. Более подробно про **<head></head>** читайте в описание [заголовочных тегов](#)

4. **<title></title>** — между этими тегами прописывается заголовок страницы, которое отображается в самом верху браузера. Кстати, когда Вы что-то ищите в поисковиках, то первое, что отображается, это как раз название страницы. Тег **<title></title>** часто сокращенно называют "тайтлом". Советую ознакомиться со статьей: [как составить тег <title>](#)

Теперь перейдем к тегам, которые находятся в теле html страницы (внутри **<body>** и **</body>**).

5. **<center></center>** — эти теги выравнивают все, что находится внутри них по центру. В данном случае центром будет - центр экрана. В будущем рекомендуется отказаться от использования этих тегов.

6. **<h1></h1>** — это один из класса тегов заголовочных тегов **<h1>..**<h6>****, обычно в него заключают название страницы. Например, у этой странице заголовочный тег "Пример создания html страницы".

Примечание

эти теги имеют большой вес в [ранжировании сайта](#), поэтому ими необходимо пользоваться аккуратно и с умом.

При составлении html-кода нужно придерживаться простого правила: сначала должен идти заголовочный тег **<h1>**, а дальше могут идти уже **<h2>**, **<h3>** и т.д. Главное, чтобы не было сначала **<h2>**, потом **<h1>**, потом **<h3>** и т.п. Должна быть строгая иерархия. Заголовков **<h2>**, **<h3>** и т.п. может быть много.

Более подробно про эти теги читайте в уроке 14 [HTML теги <h1>-<h6> - заголовочные теги внутри страницы](#)

7. **
** — это одиночный тег, который не требует закрывающего тега. Он осуществляет переход на следующую строку. В моем примере написаны два одиночных тега подряд, чтобы дважды перейти на следующую строку.

8. **** — это одиночный тег, который выводит изображение.

1. **src** — обязательный параметр, в котором указывается адрес изображения (вместо URL_ИЗОБРАЖЕНИЯ необходимо прописать адрес, где хранится Ваше изображение).

Примечание:

- Если изображение находится в одной папке с Вашей html-страницей, то достаточно написать название изображения, иначе нужно прописывать либо [абсолютный или относительный URL](#);
- Не забудьте указать расширение изображения. Например, [.jpg](#), [.gif](#), [.jpeg](#).

2. **alt** или **title** — в эти параметры можно записать описание к Вашему изображению. Когда Вы наведете мышкой на изображение, то появится это описание. Эти параметры важны для [продвижения сайта](#), особенно в поиске по картинкам. Если изображение не удалось загрузить, то будет выведен этот текст, что также является плюсом.

Более подробно про **** читайте в специальном уроке: [html тег](#)

9. **** — эти теги созданы для изменения шрифта, фона, размера и т.д. Короче говоря, все, что связано с форматированием текста можно настроить в одном теге. Это тег имеют довольно много атрибутов, о которых я расскажу в отдельном уроке.

Примечание: **** — аналогичный тег.

Есть также свойство CSS [font](#), в котором можно задавать все эти параметры.

10. **** — выделить жирным. Все, что заключено между **** и **** будет выделено жирным. Например, если Вы напишете его в самом начале контента, а закроете в самом конце, то весь текст на странице будет выделен жирным. Это довольно часто встречающийся тег, аналогом которого является ****.

Поисковые системы обращают внимание на этот тег в плане усиления влияния ключевых слов. Но нужно быть аккуратным, поскольку выделять каждый раз жирным ключевые слова будет воспринято за спам.

Читайте также: [как сделать жирный шрифт html](#)

11. **<hr/>** — одиночный тег, который выводит горизонтальную линию. Имеет несколько параметров, о которых будет сказано позднее.

12. **текст_ссылки** — тег для создания ссылок.

href="URL" — этому параметру необходимо присвоить URL ссылки, на которую ведет ссылка. Является обязательным параметром.

Примечание

Если документ, на который Вы хотите перейти, находится в одной папке с Вашей html-страницей, то достаточно написать название этого документа. Например:

```
<a href="stranica_50.html">stranica_50.html</a>
```

Можно писать и полный адрес страницы

```
<a  
href="http://САЙТ.РУ/КАТЕГОРИЯ/stranica_50.html">http://САЙТ.РУ/КАТЕГОРИЯ/stranica_50.  
html</a>
```

Этому важному тегу посвящен специальный урок: [HTML тег <a>](#).

Приложение 2

Источник: <https://zarabotat-na-sajte.ru/uroki-html/teg-ul.html>

HTML тег - маркированный список

В языке разметки гипертекста HTML имеется тег ``, используемый для создания маркированных списков. Он поддерживается всеми современными браузерами и позволяет выводить элементы в порядке, не нуждающемся в нумерации. Например, очень часто с помощью него выводятся пункты меню и все что касается списков на странице: блюд, продуктов, оборудования, инструментов и многое другое, что нужно перечислить без указания на приоритетность того или иного пункта.

Синтаксис тега

```
<ul>

    <li>Элемент #1</li>

    <li>Элемент #2</li>

    <li>Элемент #3</li>

    <li>...</li>

</ul>
```

Этот код преобразуется в маркированный список на сайте:

- Элемент #1
- Элемент #2
- Элемент #3
- ...

Тег `` требует обязательного использования закрывающего тега ``.

Для формирования элементов списка используется парный тег `..`. Между открывающим и закрывающим тегами располагаются отдельные слова, фразы, абзацы, изображения, куски кода и многое другое, являющееся содержимым маркированного списка.

Что может являться содержимым маркированного списка?

Это могут быть различные тексты, включая одиночные слова, фразы и абзацы, изображения, вложенные списки, куски php-кода и многое другое, нуждающееся в простой маркировке.

Каждый элемент маркированного списка отступает по умолчанию на 40 пикселей вправо. Используя стили CSS, мы можем изменить отображение данного списка по своему усмотрению. Тег `` является блочным, поэтому он занимает всю доступную

для него область, ограниченную краем экрана, рамкой таблицы или другими элементами страницы.

Допускается вложения "список в списке"

Например

```
<ul>

  <li>Элемент #1</li>

  <li>

    <ul>

      <li>Элемент #2-1</li>

      <li>Элемент #2-2</li>

      <li>Элемент #2-3</li>

    </ul>

  </li>

  <li>Элемент #3</li>

  <li>...</li>

</ul>
```

Атрибуты и свойства тега

Широко распространённым атрибутом тега является атрибут **type**, указывающий на то, как будет выглядеть маркер списка. Может принимать следующие значения

1. **type="disc"** — маркер в виде закрашенного кружка (это значение стоит по умолчанию). Пример с диском был чуть выше.

2. **type="circle"** — маркер в виде прозрачного кружка

Например:

```
<ul type="circle">

  <li>Элемент #1</li>

  <li>Элемент #2</li>

</ul>
```

А вот как это выглядит на странице:

- Элемент #1

- Элемент #2

3. **type="square"** — маркер в виде квадратика

Например:

```
<ul type="square">
  <li>Элемент #1</li>
  <li>Элемент #2</li>
</ul>
```

А вот как это выглядит на странице:

- Элемент #1
- Элемент #2

Примечание 1

В CSS тип маркера задается с помощью атрибута **list-style-type**:

```
<ul style="list-style-type: square">
  <li>...</li>
</ul>
```

Рассмотрим какие значения может принимать **list-style-type**:

- **disc** — маркер в виде кружка (пример был выше)
- **circle** — маркер в виде прозрачного кружка (пример был выше)
- **square** — маркер в виде квадратика (пример был выше)
- **decimal** — маркер в виде нумеренного списка арабскими цифрами: 1, 2, 3, ...
- **decimal-leading-zero** — маркер в виде нумеренного списка арабскими цифрами с нулем в начале: 01, 02, 03, ...
- **lower-roman** — маркер в виде нумеренного списка римским алфавитом маленькими буквами: i, ii, iii, iv, v
- **upper-roman** — маркер в виде нумеренного списка римским алфавитом большими буквами: I, II, III, IV, V
- **lower-latin** — маркер в виде списка латинским алфавитом маленькими буквами: a, b, c, d, ...
- **upper-latin** — маркер в виде списка латинским алфавитом большими буквами: A, B, C, D, ...
- **lower-greek** — маркер в виде списка греческим алфавитом маленькими буквами
- **upper-greek** — маркер в виде списка греческим алфавитом большими буквами

Примечание 2

Атрибут можно назначить как самому тегу ****, так и тегам ****. При задании атрибута тегу **** все элементы списка будут отображаться так, как на это указывает атрибут. Но мы можем задать тому или иному элементу своё отображение. Пример на рисунке:

Код при этом выглядит так:

```
<ul type="circle">
  <li>Элемент #1</li>
  <li style="list-style-type: disc">Элемент #2</li>
  <li>Элемент #3</li>
</ul>
```

А так это выглядит на странице:

- Элемент #1
- Элемент #2
- Элемент #3

Изменение маркеров тега `` с помощью CSS⁴

Элементы маркированного списка, создаваемые тегом ``, могут маркироваться произвольными изображениями. Для изменения типа маркера используется CSS. Например

```
<ul style="list-style-image: url('primer.gif') ">
  <li>Элемент #1</li>
  <li>Элемент #2</li>
  <li>Элемент #3</li>
</ul>
```

А так это выглядит на странице:

- Элемент #1
- Элемент #2
- Элемент #3

С помощью CSS мы можем задать и другие типы отображения маркеров. Но нужно помнить, что при задании какого-либо стиля тегу ``, он распространяется на все элементы списка.

⁴ Для выполнения лабораторной работы №1 с данным разделом можно только ознакомиться, но не использовать его как руководство к действию.

HTML тег - нумерованный список

В целом синтаксис тега **** такой же как и у обычного html списка ****. Разница лишь в том, что этот тег строго сделан для нумерации списков. Название тега пришло от английского сокращения "Ordered List" - нумерованный список.

Синтаксис тега

```
<ol [type=value] [start=value2] [reversed]>
    <li>Элемент #1</li>
    <li>Элемент #2</li>
    <li>Элемент #3</li>
    <li>...</li>
</ol>
```

Где атрибут **type="value"** может принимать следующие значения

- **A** — задает маркеры в виде прописных латинских букв (A, B, C..);
- **a** — задает маркеры в виде строчных латинских букв (a, b, c..);
- **I** — задает маркеры в виде больших римских цифр (I, II, III, IV..);
- **i** — задает маркеры в виде маленьких римских цифр (i, ii, iii, iv..);
- **1** (по умолчанию) — задает маркеры в виде арабских цифр (1, 2, 3..);

Атрибут **start="value"** задает начальное значение (стартовое значение) отчета.

Атрибут **reversed** задает обратный счет (в случае необходимости).

Тег **** требует обязательного использования закрывающего тега ****

Для формирования элементов списка используется парный тег ****. Между открывающим и закрывающим тегами располагаются отдельные слова, фразы, абзацы, изображения, куски кода и многое другое, являющееся содержимым маркированного списка.

Примечание

Внутри списка есть возможность изменять счет на свой. Для этого есть специальный атрибут **value=""** у тега ****, которому присваивается какое-то числовое значение. Например

```
<ol>

  <li>Элемент #1</li>

  <li value="10">Элемент #2</li>

  <li>Элемент #3</li>

</ol>
```

Примеры с нумерованными списками в html ()

Пример 1. Нумерованный список html в виде латинских букв

Пример с заглавными буквами

```
<ol type="A">

  <li>Элемент #1</li>

  <li>Элемент #2</li>

  <li>Элемент #3</li>

</ol>
```

Вот как это выглядит на странице:

1. Элемент #1
2. Элемент #2
3. Элемент #3

Пример со строчными буквами

```
<ol type="a">

  <li>Элемент #10</li>

  <li>Элемент #11</li>

  <li>Элемент #12</li>

</ol>
```

Вот как это выглядит на странице:

1. Элемент #1
2. Элемент #2
3. Элемент #3

Пример 2. Нумерованный список html в виде римских букв

Пример с заглавными буквами

```
<ol type="I">
  <li>Элемент #1</li>
  <li>Элемент #2</li>
  <li>Элемент #3</li>
</ol>
```

Вот как это выглядит на странице:

1. Элемент #1
2. Элемент #2
3. Элемент #3

Пример со строчными буквами

```
<ol type="i">
  <li>Элемент #1</li>
  <li>Элемент #2</li>
  <li>Элемент #3</li>
</ol>
```

Вот как это выглядит на странице:

1. Элемент #1
2. Элемент #2
3. Элемент #3

Пример 3. Нумерованный список html разная позиция старта

Пример, который показывает возможности атрибута **start**, который позволяет задавать стартовое значение счетчика.

```
<ol start="50">
  <li>Элемент #1</li>
  <li>Элемент #2</li>
  <li>Элемент #3</li>
</ol>
```

Вот как это выглядит на странице:

50. Элемент #1
51. Элемент #2
52. Элемент #3

Пример 4. Изменение счета в нумерованных списках html

Ниже представлен пример с возможностью менять значения счетчика с помощью атрибута **value** при выводе новых элементов в тегах ``.

```
<ol start="10">
  <li>Элемент #1</li>
  <li>Элемент #2</li>
  <li value="100">Элемент #3</li>
  <li>Элемент #4</li>
</ol>
```

Вот как это выглядит на странице:

-
- 10. Элемент #1
 - 11. Элемент #2
 - 100. Элемент #3
 - 101. Элемент #4
-

Пример 5. Реверсивный нумерованный список в html

Ниже приведен пример реверсивного нумерованного списка (счет в обратном порядке).

```
<ol start="10" reversed>
  <li>Элемент #1</li>
  <li>Элемент #2</li>
  <li>Элемент #3</li>
  <li>Элемент #4</li>
</ol>
```

Вот как это выглядит на странице:

-
- 10. Элемент #1
 - 11. Элемент #2
 - 12. Элемент #3
 - 13. Элемент #4
-

Приложение 3

Источник: <https://zarabotat-na-sajte.ru/uroki-html/teg-table.html>

HTML тег <table>

С недавнего времени тег **<table>** стал заметно сдавать свои позиции по популярности другому **<div>**, поскольку у него больше возможностей в плане дизайна. Однако человек, который разбирается в html, просто обязан хорошо разбираться в таблицах.

Синтаксис тега <table>

```
<table [атрибуты]>
  <tr>
    <td>
      ...
    </td>
  ...
</tr>
...
</table>
```

Между открывающим тегом **<table>** и закрывающим тегом **</table>** находится тело таблицы. Тело состоит из строк и столбцов. Таблица заполняется построчно.

Каждый тег **<tr>** создает новую строку. Далее во вложенных **<td>** создаются столбцы. Можно создавать множество столбцов. При этом нужно следить за количеством столбцов в каждой строке. Например, если в первой строке было 5 столбцов, то и в следующих строках должно быть 5 столбцов. Иначе таблица поплывет. Есть возможность объединять ячейки.

Как сделать таблицу в html

Приведем пример, html код:

```
<html>
<body>
```

```

<table align="center" width="100%" border="1">
    <tr>
        <td colspan="2">
            Пример таблицы
        </td>
    </tr>
    <tr>
        <td>
            Столбец 1
        </td>
        <td>
            Столбец 2
        </td>
    </tr>
</table>
</body>
</html>

```

Преобразуется на странице в следующее:

Пример таблицы	
Столбец 1	Столбец 2

Обратите внимание на ячейку `<td colspan="2">`. Мы используем специальный атрибут **colspan** для объединения ячеек по горизонтали. Его численное значение указывает число объединяемых столбцов. Также существует аналог этого атрибута: тег **<th>** (заголовок таблицы), где нужно также прописать colspan. Результат будет таким же. Но зачастую используют обычный td.

Теперь рассмотрим подробно все атрибуты тега `<table>`.

Атрибуты и свойства тега `<table>`

К открывающему тегу `<table>` можно прописывать различные атрибуты.

1. Свойство **align="параметр"** - задает выравнивание таблицы. Может принимать следующие значения:

- **left** - выравнивание по левому краю
- **center** - выравнивание по центру
- **right** - выравнивание по правому краю

В разобранный выше примере мы выравнивали таблицу по центру **align="center"**.

Этот атрибут можно применять не только к таблице, но и к отдельным ячейкам таблицы `<td>` или строкам `<tr>`. Таким образом, в разных ячейках выравнивание будет разное.

Например

```
<td align="left">...<td>
<td align="center">...<td>
...
```

2. Свойство **`background="URL"`** - задает фоновый рисунок. Вместо URL должен быть написан адрес фонового изображения.

Пример

```
<html>
<body>
<table background="img/fon.gif" align="center"
width="100%" border="1" style="color:white;">
<tr>
    <td colspan="2">
        Пример таблицы
    </td>
</tr>
<tr>
    <td>
        Столбец 1
    </td>
    <td>
        Столбец 2
    </td>
</tr>
</table>
</body>
</html>
```

Преобразуется на странице в следующее:

В рассмотренном примере наше фоновое изображение находится в папке `img` (которая находится в той же директории, что и `html`-страница), а называется изображение `fon.gif`. Обратите внимание на то, что в тегах мы

добавили **style="color:white;"**. Поскольку фон почти черный, то для того, чтобы текст не слился с фоном, мы сделали текст белым.

Более подробно про фон читайте в статье: [как сделать фон для сайта](#)

3. Свойство **bgcolor="цвет"** - задает цвет фона таблицы. В качестве цвета можно выбрать любой из всей палитры (см. [коды и названия html цветов](#))

4. Свойство **border="число"** - задает толщину рамки таблицы. В предыдущих примерах мы указывали **border="1"**, что означает толщина рамки - 1 пиксель.

5. Свойство **bordercolor="цвет"** - задает цвет рамки. Если **border="0"**, то рамки не будет и цвет рамки не будет иметь смысла.

6. Свойство **cellpadding="число"** - отступ от рамки до содержимого ячейки в пикселях.

7. Свойство **cellspacing="число"** - расстояние между ячейками в пикселях.

8. Свойство **cols="число"** - число столбцов. Если его не задать, то браузер сам определит число столбцов. Разница лишь в том, что указание этого параметра, скорее всего, ускорит загрузку таблицы.

9. Свойство **frame="параметр"** - как отображать границы вокруг таблицы. Может принимать следующие значения:

- **void** - не отрисовывать границы
 - **border** - граница вокруг таблицы
 - **above** - граница по верхнему краю таблицы
 - **below** - граница снизу таблицы
 - **hsides** - добавить только горизонтальные границы (сверху и снизу таблицы)
 - **vsides** - рисовать только вертикальные границы (слева и справа от таблицы)
 - **rhs** - граница только на правой стороне таблицы
 - **lhs** - граница только на левой стороне таблицы
-

10. Свойство **height="число"** - задает высоту таблицы: либо в пикселях, либо в процентах.

11. Свойство **rules="параметр"** - где отображать границы между ячейками. Может принимать следующие значения:

- **all** - линия рисуется вокруг каждой ячейки таблицы
 - **groups** - линия отображается между группами, которые образуются тегами **<thead>**, **<tfoot>**, **<tbody>**, **<colgroup>** или **<col>**
 - **cols** - линия отображается между колонками
 - **none** - все границы скрываются
 - **rows** - граница рисуется между строками таблицы, созданными через тег **<tr>**
-

12. Свойство **width="число"** - задает ширину таблицы: либо в пикселях, либо в процентах.

13. Свойство **class="имя_класса"** - можно указать имя класса, которому принадлежит таблица.

14. Свойство **style="стили"** - стили можно задать индивидуально для каждой таблицы.

Теперь настало время погрузиться внутрь таблицы и рассмотреть атрибуты ячеек таблицы. Эти атрибуты надо писать в открывающем теге **<td>**.

Атрибуты и свойства **<tr>** и **<td>**

1. Свойство **align="параметр"** - задает выравнивание отдельной ячейки таблицы. Может принимать следующие значения:

- **left** - выравнивание по левому краю
 - **center** - выравнивание по центру
 - **right** - выравнивание по правому краю
-

2. Свойство **background="URL"** - задает фоновое изображение ячейки. Вместо URL должен быть написан адрес фонового изображения.

3. Свойство **bgcolor="цвет"** - задает цвет фона ячейки.

4. Свойство **bordercolor="цвет"** - задает цвет рамки ячейки.

5. Свойство **char="буква"** - задает букву, от которой нужно сделать выравнивание. Значение атрибута align должно быть установлено как char.

6. Свойство **colspan="число"** - задает число объединяемых горизонтальных ячеек.

7. Свойство **height="число"** - задает высоту таблицы: либо в пикселях, либо в процентах %.

8. Свойство **width="число"** - задает ширину таблицы: либо в пикселях, либо в процентах %.

9. Свойство **rowspan="число"** - задает число объединяемых вертикальных ячеек.

10. Свойство **valign="параметр"** - выравнивание содержимого ячейки по вертикали.

- **top** - выравнивание содержимого ячейки по верхнему краю строки
 - **middle** - выравнивание по середине
 - **bottom** - выравнивание по нижнему краю
 - **baseline** - выравнивание по базовой линии
-

Примечание 1

Для тега <tr> доступны такие же параметры, что и для <td>. Параметры для одного тега <tr> будут иерархично применены ко всем <td> внутри него

Как сделать, чтобы границы ячеек в таблице не склеивались

В случае использования `border` (граница ячеек) и нулевых отступов между ячейками, они все равно склеиваются и получается удвоенный [border](#). Чтобы это избежать нужно прописать в стилях таблице **`border-collapse: collapse;`**

```
<table style="border-collapse: collapse;">
...
```

Более подробно про это свойство читайте в специальной статье [border-collapse CSS](#)

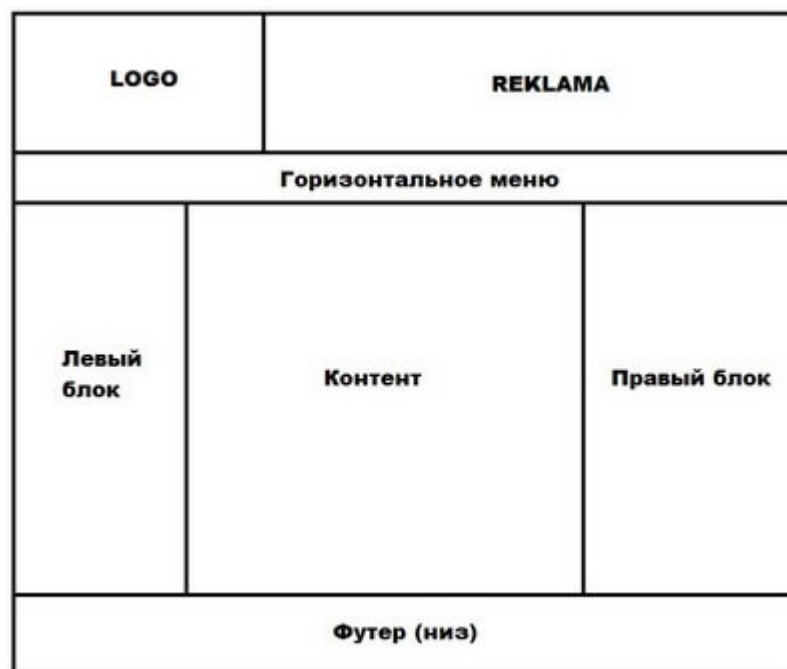
Уважаемый читатель, теперь Вы узнали гораздо больше о html теге table. Теперь советую перейти к следующему уроку.

Приложение 4

Источник: <https://zarabotat-na-sajte.ru/uroki-html/teg-div.html>

HTML тег <div>

Пару лет назад "скелетом" для сайта всегда служила таблица ([тег <table>](#)). Создавалась одна большая таблица, которая потом делилась на несколько областей: заголовок, левый блок, правый блок, центр и низ. С появлением тега **<div>** все вебмастера стали пользоваться им для создания "скелета" будущего сайта. Например, так может выглядеть структура страницы сайта:



Синтаксис тега <div>

```
<div class="Имя класса">
...
</div>
```

HTML тег **<div>** является блочным элементом. Самым главным его атрибутом является **class**. С помощью него можно создавать блоки с нужными [стилями CSS](#).

Пример. Использование блока div

```
<html>
```



```

<head>
<style type="text/css">
    .primer {
        width: 300px;
        background: #9affe8;
        padding: 5px;
        padding-right: 25px;
        border: solid 1px orange;
        float: right;
    }
</style>
</head>
<body>
    <div class="primer">
        Пример использования блока div
    </div>
</body>
</html>

```

Преобразуется на странице в следующее:

Пример использования блока div

Это элементарный пример использования тега **<div>**. Его возможности уходят далеко за рамки этого примера. Например, с помощью этого тега можно создавать красивые меню (см. [как сделать выпадающее меню на css](#) »)

Теперь рассмотрим подробно все атрибуты тега **div**.

Атрибуты и свойства тега <div>

1. **align="параметр"** — задает выравнивание. Может принимать следующие значения:
 - **center** — выравнивание текста по центру
 - **left** — выравнивание текста по левому краю
 - **right** — выравнивание текста по правому краю
 - **justify** — выравнивание по левому и правому краю
2. **title="текст"** — всплывающая подсказка к тегу. Почти никогда не используется.
3. **class="имя"** — определяет принадлежность к классу.
4. **style="стили через запятую"** — возможность задать стили.

Практические примеры использования <div>

Давайте рассмотрим пару практических примеров с тегом `<div>`.

Пример. Вывод контента по центру с использованием div

```
<html>
<head>
<style type="text/css">
    .center {
        text-align: center;
    }
</style>
</head>
<body>
    <div class="center">
        Этот текст будет по центру
    </div>
</body>
</html>
```

Преобразуется на странице в следующее:

Этот текст будет по центру

Теперь вместо тега `<center></center>` можно писать `<div class="center"></div>` и текст будет выведен по центру. Это соответствует требованиям стандарта HTML ([валидация сайта](#)).

Пример. Красивая рамка с div

```
<html>
<head>
<style type="text/css">
    .ramka {
        background: #eeeeee5;
        border: 1px dashed #abab9a;
        padding: 5px;
        font: 8pt Tahoma;
        color: #2c2c2c;
    }
</style>
```

```

</head>
<body>

    <div class="ramka">

        Вывод текста в красивом блоке

    </div>

</body>
</html>

```

Преобразуется на странице в следующее:

Вывод текста в красивом блоке

Пример. Реализация тизеров на сайте

Удобство тега `div` состоит в том, что он размещается блочно относительно друг друга. Практическим примером является вывод подряд идущих блоков, которые будут автоматически переноситься на новую строку при достижении конца.

```

<head>
<style type="text/css">
    .block {
        padding: 0;
        font-size: 14px;
    }
    .block .element{
        border: solid 1px #C7CADD;
        margin: 0px 3px 3px 0px;
        padding: 0px 0px 0px 0px;
        float: left;
        height: 200px;
        width: 150px;
    }
    .block .element:hover{
        border: solid 1px #000;
        margin: 0px 3px 3px 0px;
        padding: 0;
        float: left;
        height: 200px;
        width: 150px;
        background: #FFFAE7;
    }

```

```

        .block .img{
            text-decoration: none;
        }

        .block .img img{
            margin: 5px 0px 0px 5px;
            padding: 0;
            width: 140px;
            height: 130px;
        }

        .block .text{
            text-align: center;
            padding: 0;
            margin: 0;
        }
    </style>
</head>
<body>
<div class="block">
    <div class="element">
        <a href="">
            <div class="img">
                
            </div>
            <div class="text">Море</div>
        </a>
    </div>

    <div class="element">
        <a href="">
            <div class="img">
                
            </div>
            <div class="text">Горы</div>
        </a>
    </div>
</div>
</body>

```

Таких элементов можно сделать множество. В итоге Вы получаете что-то подобное:



Море



Горы



Снег

Уважаемый читатель, теперь Вы узнали гораздо больше о разметке html. Теперь Вы в состоянии сделать свой собственный сайт. Позднее будут выложены ссылки на практические примеры, которые часто необходимы в верстке.

