

Учимся создавать  
HTML-документы



# ОСНОВЫ HTML

- ◆ **Язык HTML (Hyper-Text Markup Language)** - язык гипертекстовой разметки - основное средство создания страниц для Internet.

Данные в формате HTML похожи на текстовый файл, за исключением того, что некоторые из символов интерпретируются как разметка. Разметка придает документу некую структуру.

- ◆ **Достоинства HTML-документов: малый объем, возможность просмотра на компьютерах с различными операционными системами, интерактивность.**



## Инструменты создания web-документа

- ◆ Текстовый редактор.
- ◆ Браузер для просмотра результатов.
- ◆ Валидатор — это программа для анализа HTML- кода Web страниц на наличие ошибок, согласно существующим стандартам.
- ◆ Графический редактор.
- ◆ Справочник по тегам HTML.

# Инструменты создания web-документа

**HTML-редакторы бывают двух видов:**

- ◆ **Кодовые редакторы.** HTML-документ можно создавать в любом текстовом редакторе, например, в Блокноте. Однако, удобно, чтобы были следующие возможности:
  - подсветка синтаксиса
  - работа с несколькими документами.
  - проверка текущего документа на ошибки.
- ◆ **Визуальные редакторы.** Наиболее популярным является **Macromedia Dreamweaver**, занимающий промежуточную позицию между кодовыми и визуальными редакторами. К тому же по нему достаточно много литературы на русском языке.

**Графический редактор**

- ◆ необходим для обработки изображений и их подготовки для публикации на веб-странице. Самый популярный – **Photoshop**. Но в большинстве случаев мощь Photoshop-а избыточна. Программа **Paint.Net** позволяет сделать все необходимые манипуляции с изображениями, вдобавок бесплатна для использования.

**Браузер**

- ◆ Браузер - программа, предназначенная для просмотра веб-страниц. Подойдет любой браузер. На сегодняшний день наибольшей популярностью пользуются три браузера: Firefox, Internet Explorer и Opera.

# Синтаксис HTML документа

- ◆ В HTML-документе есть две явные составляющие:
  - **сам текст**, который пользователь видит на экране браузера,
  - **теги** (англ.: *tag-ярлычок, этикетка*), которые определяют, как он выводится на экран, но остаются невидимыми для пользователя при просмотре документа.
- ◆ Все это в общем случае выглядит так:  
**<открывающий тег> текст, к которому применяется этот тег </закрывающий тег>**
- ◆ Итак, тег состоит из следующих друг за другом в определенном порядке элементов:
  - левой угловой скобки **<** (такого же, как "меньше чем" символа)
  - имени тега, например **title** или **pre**
  - атрибутов. Тег может быть без атрибутов или сопровождаться одним или несколькими атрибутами, например: **align=center**
  - правой угловой скобки **>** (такой же, как символа "больше чем").
  - если тег является конечным тегом, то после знака **<** ставится слеш **/**, такой тег закрывает некоторую структуру. Таким образом Вы можете читать символ **/**, как конец.
- ◆ Пример: **<h1>**, **<h1 align=left>**

# Начало html-документа

- ◆ Браузер загружает страницу и “считывает код” сверху вниз, слева направо.
- ◆ Каждый HTML-документ, отвечающий спецификации HTML какой-либо версии, обязан начинаться со строки декларации версии HTML **!DOCTYPE**, которая обычно выглядит так: :  
**<!DOCTYPE HTML PUBLIC “-//W3C//DTD HTML 4.0//EN”>**
- ◆ Элемент **<!DOCTYPE>** предназначен для указания типа текущего документа — DTD (**document type definition, описание типа документа**).
- ◆ Это необходимо, чтобы браузер понимал, как следует интерпретировать текущую веб-страницу, т.к. HTML существует в нескольких версиях, кроме того, есть XHTML (EXtensible HyperText Markup Language, расширенный язык разметки гипертекста), похожий на HTML, но различающийся с ним по синтаксису.
- ◆ Чтобы браузер «не путался» и понимал, согласно какому стандарту отображать веб-страницу и необходимо в первой строке кода задавать **<!DOCTYPE>**.

# Структура HTML-документа



**<HTML> ... </HTML>** - тэги показывают начало и окончание HTML документа

**<HEAD> ... </HEAD>** - это тэг заголовка документа, содержит справочную информацию о странице, **которая не отображается браузером**. Например, здесь располагается контейнер

**<TITLE> ... </TITLE>** - тэг названия страницы, который появится в строке названия в браузере.

**<BODY> ... </BODY>** - эта пара тэгов указывает на начало и конец **тела HTML-документа**. В *теле документа* содержится ваша информация: текст, графические изображения, бегущие строки, звук и т.д., которая отображается на экране монитора.

# Пример простейшего web-документа

1. Откройте Блокнот и запишите код:

```
<!DOCTYPE HTML PUBLIC  
“-//W3C//DTD HTML 4.0//EN”>
```

```
<html>
```

```
<head>
```

```
  <title> Изучение html </title>
```

```
</head>
```

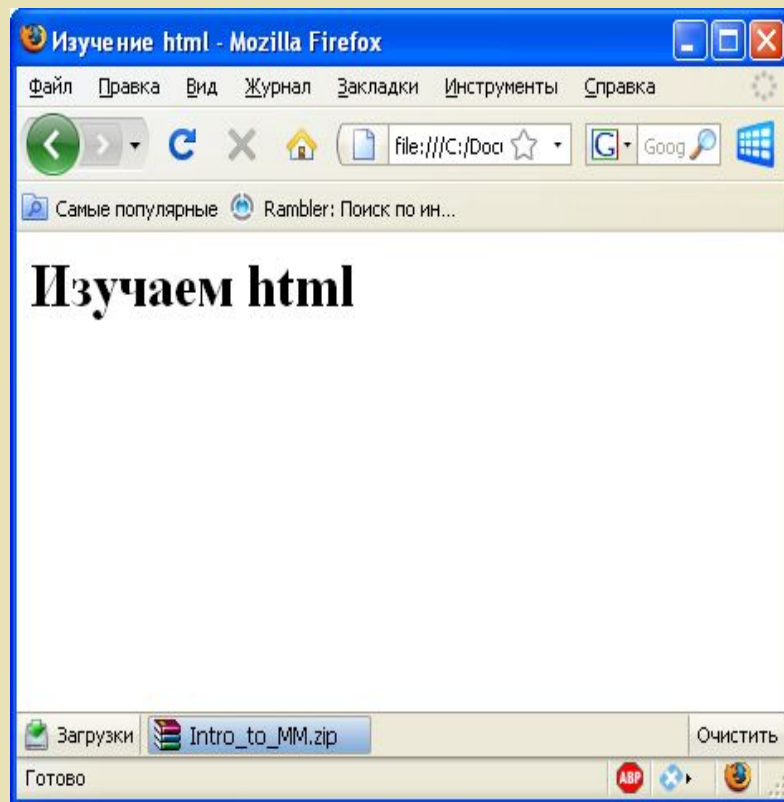
```
<body>
```

```
  <h1> Изучаем html </h1>
```

```
</body>
```

```
</html>
```

2. Сохраните файл под именем firstweb.html
3. Имена файлам будем давать **строчными латинскими буквами**
4. Откройте браузер и загрузите туда ваш файл firstweb.html





# Продолжаем создавать страничку

- ◆ Добавим строчку в Блокноте  
**<!DOCTYPE HTML PUBLIC  
“-//W3C//DTD HTML 4.0//EN”>**

**<html>**

**<head>**

**<title> Изучение html </title>**

**</head>**

**<body>**

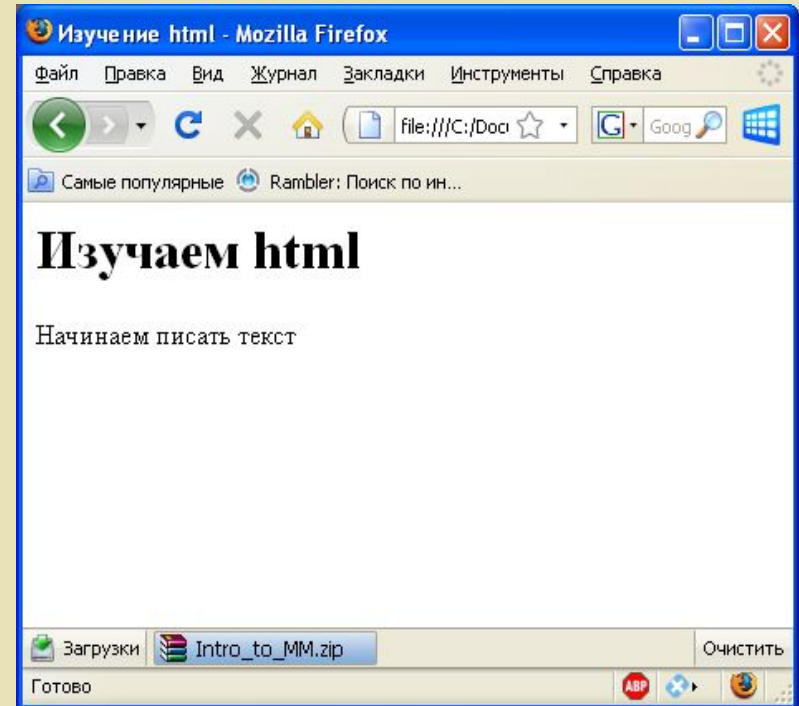
**<h1> Изучаем html </h1>**

**<p> Начинаем писать текст </p>**

**</body>**

**</html>**

Сохраним файл и обновим в  
браузере



# Правила применения тегов

## ◆ Атрибуты тегов и кавычки

- Согласно спецификации HTML все значения атрибутов тегов следует указывать в двойных ("пример") или одинарных кавычках ('пример').
- Отсутствие кавычек не приведет к ошибкам, браузеры во многих случаях достаточно корректно обрабатывают код и без кавычек, за исключением текста, содержащего пробелы

## ◆ Теги можно писать как прописными, так и строчными символами

- Любые теги, а также их атрибуты нечувствительны к регистру, поэтому форму записи вы вольны выбирать сами, как писать — `<BR>`, `<Br>` или `<br>`. В любом случае рекомендуется придерживаться выбранной формы записи на протяжении всех страниц сайта.

## ◆ Переносы строк

- Внутри тега между его атрибутами допустимо ставить перенос строк.
- Однако рекомендуется писать теги в одну строку, иначе ухудшается восприятие кода и его становится сложнее править.

## ◆ Закрывающие теги

- Существует три состояния закрывающего тега: обязателен, не требуется или не обязателен.
- Обязательный закрывающий тег должен присутствовать всегда, иначе это приведет к ошибке при отображении документа.
- Для некоторых тегов (например `<br>`) закрывающего тега нет в принципе.
- Необязательный закрывающий тег говорит о том, что разработчик может его как добавить, так и опустить, к ошибке это не приведет. Однако рекомендуется закрывать все подобные теги, включая необязательные, это дисциплинирует, создает более стройный и строгий код, который легко модифицировать.

# Атрибуты тэгов

Большинство тэгов имеет атрибуты, т.е. свойства, которые могут принимать различные значения.

## ◆ Синтаксис

`<ТЭГ атрибут 1="значение" .....атрибут n="значение">`

...

## ◆ Порядок атрибутов в тегах

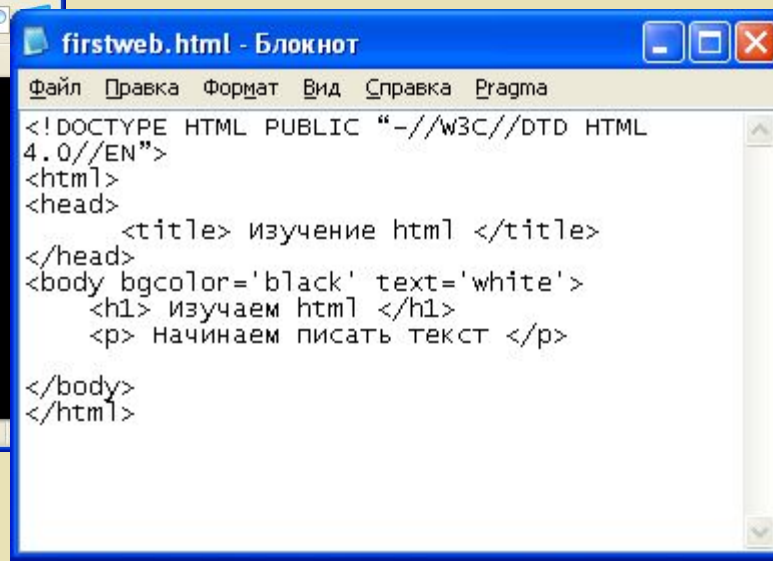
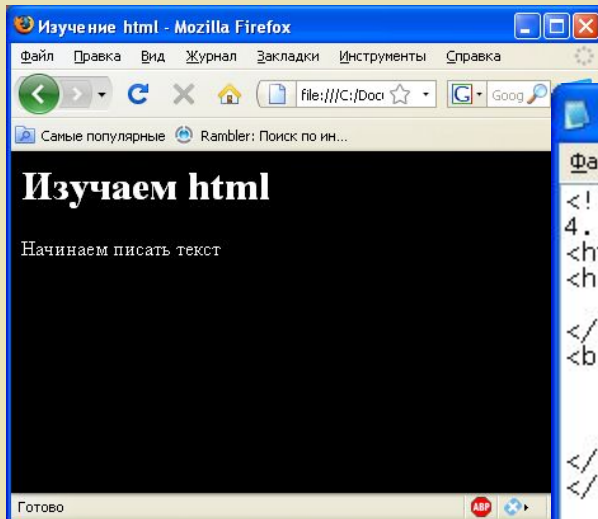
- Порядок атрибутов в любом теге не имеет значения и на результат отображения элемента не влияет.

## ◆ Формат атрибутов

- Каждый атрибут тега относится к определенному типу (например: текст, число, путь к файлу и др.), который обязательно должен учитываться при написании атрибута.

## ◆ Пример

- текст документа написан белым цветом на черном фоне



# Комментарии

◆ <!-- -->

- ◆ Используется для создания комментариев в любой части документа. Все, что находится внутри <!-- --> – будь то элемент или текст – будет проигнорировано браузером (не будет обрабатываться и выводиться на экран).

◆

◆ **Пример:**

```
<HTML>
```

```
  <HEAD>
```

```
    <TITLE> Заголовок документа</TITLE>
```

```
  </HEAD>
```

```
    <!-- Начинаем работу с телом документа -->
```

```
<BODY>
```

```
  <!-- Помещаем нужную информацию -->
```

```
  Текст документа
```

```
</BODY>
```

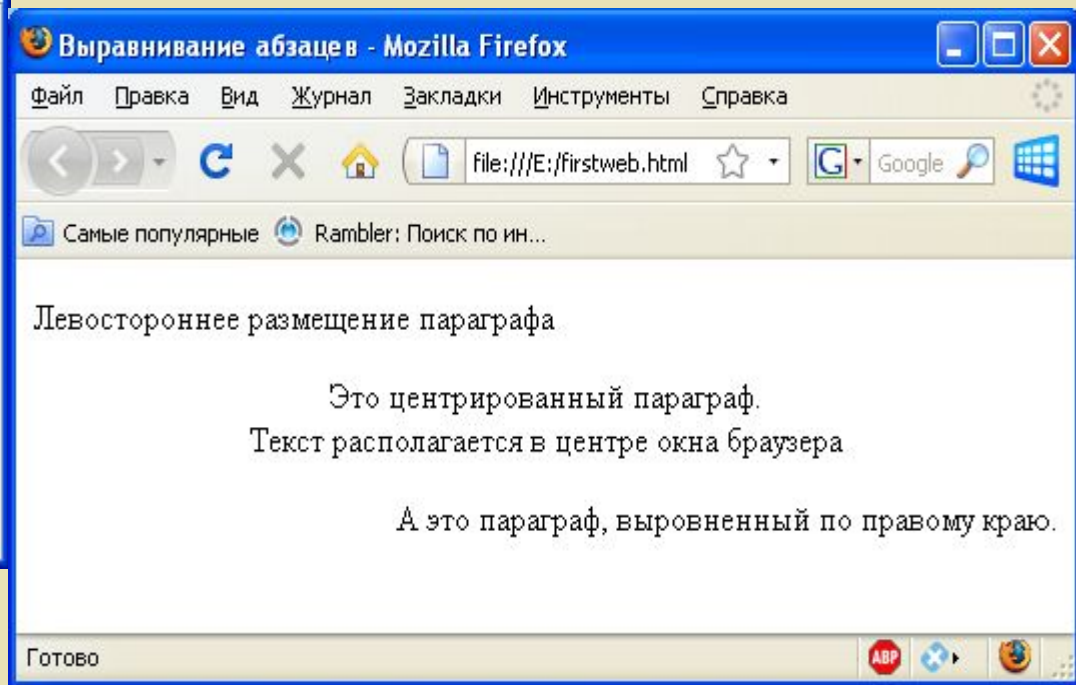
```
<!-- Все. Готово -->
```

```
</HTML>
```

# Теги абзаца и перевода строки

- ◆ Абзацы в html отделяются друг от друга пустой строкой
  - `<p> </p>` (**paragraf**)– тег нового абзаца (парный)
- ◆ Атрибут **ALIGN** – позволяет изменить выравнивание абзаца.  
`<p align= «?» >`, где ?=left, right, center, justify
- ◆ `<br>` -одиночный тэг образования новой строки без образования нового абзаца

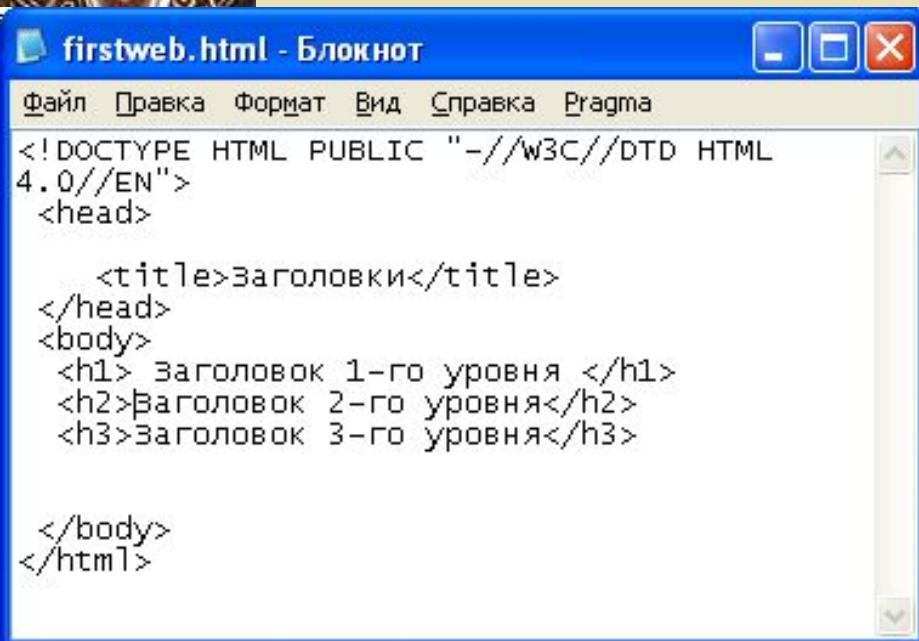
```
firstweb.html - Блокнот
Файл  Правка  Формат  Вид  Справка  Прагма
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0//EN">
<head>
  <title>выравнивание абзацев</title>
</head>
<body>
  <p align="left"> левостороннее размещение
    параграфа </p>
  <p align="center">Это центрированный
    параграф.<br>Текст располагается в центре
    окна браузера</p>
  <p align="right">А это параграф,
    выровненный по правому краю.</p>
</body>
</html>
```



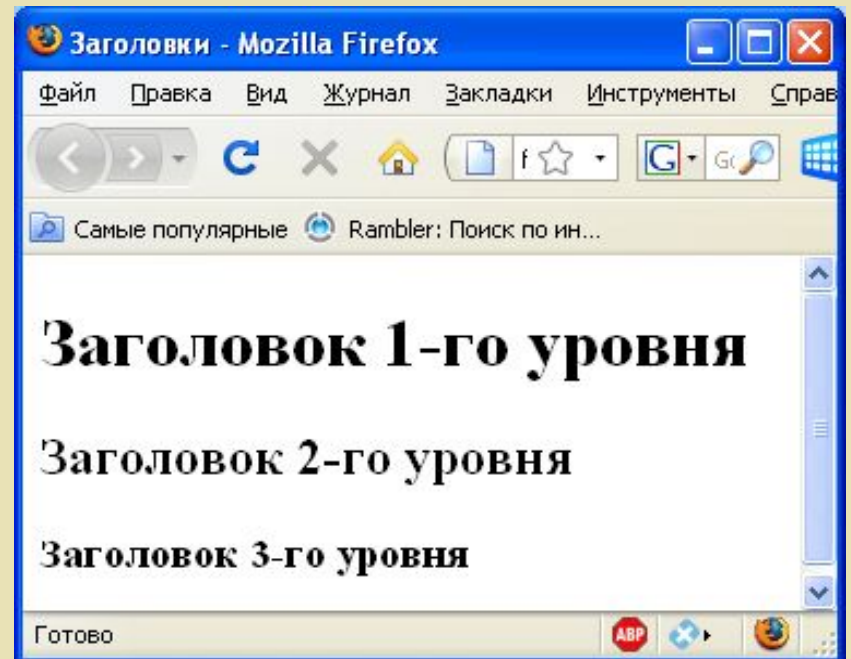
# Использование заголовков

В HTML есть шесть уровней заголовков

- ◆ `<h1>` `<h2>`...`<h6>` - теги заголовков с заранее заданным форматированием - инструмент для структуризации документа. `h1` - самый главный, `h6` - самый мелкий.
- ◆ Для того, чтобы оформить текст заголовком следует поместить тэг перед и после текста, например:  
`<h1> Заголовок самый главный </h1>`
- ◆ В тело данного тега также можно добавить атрибут **align**:  
`<h1 align="?" >` где `?=left, right, center, justify`



```
firstweb.html - Блокнот
Файл Правка Формат Вид Справка Прагма
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0//EN">
<head>
  <title>заголовки</title>
</head>
<body>
  <h1> Заголовок 1-го уровня </h1>
  <h2>заголовок 2-го уровня</h2>
  <h3>заголовок 3-го уровня</h3>
</body>
</html>
```



# Использование предварительного форматирования

- ◆ `<pre>` – вставляет предварительно отформатированный текст (Preformatted Text)
- ◆ Используется для включения в документ уже отформатированного текста. Браузеры воспроизводят содержимое этого элемента с помощью моноширинного шрифта, сохраняя пробелы и символы конца строки.

Пример:

```
<pre>
```

```
Раз!
```

```
    Два!
```

```
        Три!
```

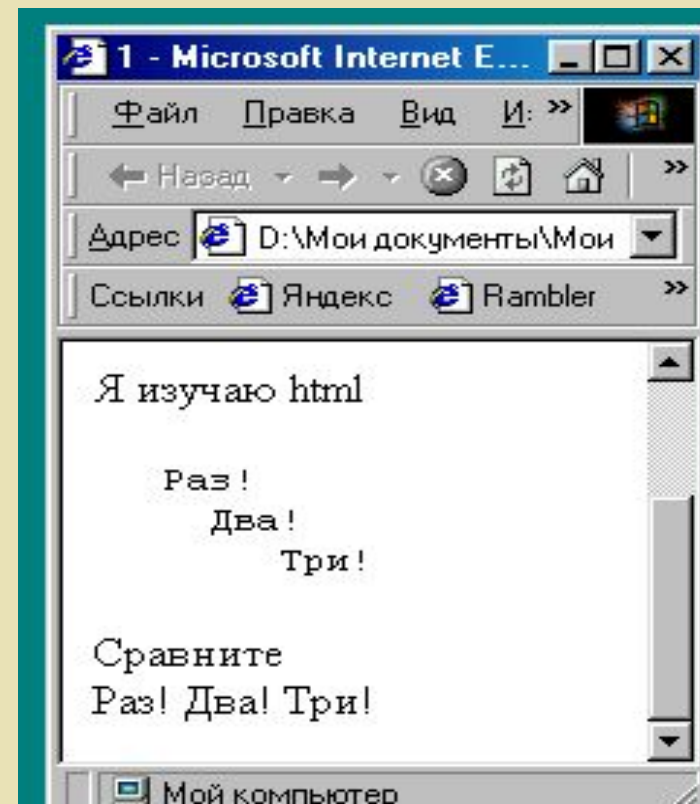
```
</pre>
```

Сравните `<br>`

```
Раз!
```

```
    Два!
```

```
        Три!
```



# Другие тэги разбиения текста на блоки

## ◆ CITE – цитата по тексту

Используется для цитат, названий книг. *Обычно наклонный текст*

**Атрибуты:** TITLE - всплывающая подсказка

**Пример:**

- ◆ Это обычный текст  
<CITE >А это уже текст с использованием тэга CITE</CITE>

## ◆ BLOCKQUOTE – оформление цитаты

Оформляет находящийся между начальным и конечным тегами текст как цитату. Используется для длинных цитат. *Цитируемый текст отображается отдельным абзацем с увеличенным отступом.*

**Пример:**

- ◆ Все знают чудесные строки сонета Шекспира:  
<BLOCKQUOTE>  
Кто под звездой счастливою рожден, <BR>  
Гордится славой, титулом и властью.<BR>  
А я судьбой скромнее награжден,<BR>  
И для меня любовь – источник счастья.<BR>

...  
</BLOCKQUOTE>

## ◆ ADDRESS – оформление текста как почтового адреса

Находящийся между начальным и конечным тегами текст оформляется как почтовый адрес. Чаще всего оформление выражается в выделении строки адреса курсивом.

**Пример:**

- ◆ Пишите по следующему адресу:  
<ADDRESS>  
Харьков, ул. Лермонтовская, 27 <BR>  
ХГУ «НУА»  
</ADDRESS>



# Цветовое оформление странички

Цветовая гамма HTML-документа определяется атрибутами, размещенными внутри тэга **<body>**:

- ◆ **bgcolor** - (BackGround Color, фоновый цвет) - атрибут, который отвечает за **цвет фона** вашей странички. По умолчанию - белый.
- ◆ **text** - отвечает за основной **цвет текста**. По умолчанию - черный.

**<body bgcolor="?" text="?">**

- ◆ Пример для задания желтого фона и синего текста:

**<body bgcolor="yellow" text="blue">**

- ◆ О задании цветов см. ссылку

# Основной шрифт текста на странице

## **basefont** – основной шрифт страницы

- ◆ Определяет основной шрифт, которым должен отображаться текст документа. *Не имеет конечного тега.*
- ◆ Впоследствии вы можете легко изменить шрифт в любой части документа (тэг <font>).

## Атрибуты тэга **basefont**:


- ◆ **size** – обязательный атрибут. Определяет базовый размер шрифта. Возможные значения: целые числа *от 1 до 7* включительно.
- ◆ **face** – определяет используемый тип шрифта (гарнитуру).

## Пример 1:

```
<body>  
  <basefont size="3" face="Tahoma">  
  ...  
  Текст документа оформлен  
  шрифтом 3 размера  
  ...  
</body>
```

## Пример 2:

```
<body>  
  <basefont size="5">  
  ...  
  Текст документа шрифтом 5  
  размера  
  ...  
  <font size="+1">  
  Слегка увеличиваем шрифт  
  </font>  
  ...  
  Продолжаем шрифтом 5 размера  
  ...  
</BODY>
```



# Задание размера, цвета и гарнитуры шрифта для отдельного фрагмента текста

- Изменить цвет, размер и тип шрифта отдельного фрагмента текста позволяет тэг

```
<font size="?" color="?" face="?"> .....</font>
```

- Атрибуты этого тэга:

**size:** определяет размер шрифта, целое число от 1 до 7;

Принято считать, что размер "нормального" шрифта соответствует значению 3.

Размер шрифта указывает как абсолютной величиной (SIZE=2), так и относительной (SIZE=+1).

**face:** определяет **гарнитуру шрифта**. Можно указать как один, так и несколько названий шрифтов, разделяя их запятыми.

Список шрифтов просматривается слева направо. Если на компьютере пользователя нет шрифта, указанного в списке первым, то делается попытка найти следующий шрифт и т.д. Если такого шрифта не будет найдено, то данное указание будет проигнорировано и будет использован шрифт, установленный по умолчанию. Совет: число различных шрифтов, используемых в одном документе, не должно превышать трех.

**color:** определяет цвет текста. Задается либо RGB-значением в шестнадцатеричной системе, либо названием цвета ([подробнее](#)).

# Определение цветов

- ◆ Для задания цветов в соответствующих атрибутах различных тэгов, например, `<font color=?>` используются следующие два приема:
  - Цвет задается просто названием на английском языке:  
`<font color="blue">` - шрифт синего цвета;
  - Цвет задается кодом, состоящим из шести знаков шестнадцатиричной системы исчисления:  
`<font color="#FF0000">` - шрифт красного цвета.
- ◆ При задании цвета с помощью шестнадцатиричного кода (т.е. с использованием цифр 0..9 и символов A,B,C,D,E,F) используется цветовая схема **RGB**, принцип которой состоит в представлении цветов с помощью трех составляющих – red (красный), green (зеленый), blue(синий). Для каждой составляющей отводится пара знаков, таким образом FF – означает **наибольшее насыщение** данного цвета, а 00 – наименьшее. Например, #FFFFFF - шестнадцатиричный код БЕЛОГО цвета, а #000000 - ЧЕРНОГО
- ◆ *Для задания цветов используйте таблицу цветов Для задания цветов используйте таблицу цветов или цветовую палитру*



## Особенности расположения текста

- ◆ Любое количество пробелов идущих подряд, в браузере отображается как один
  - Исключением из этого правила является тег `<pre>`, внутри которого любое число пробелов отображается именно так, как оно указано в коде.
- ◆ Нет расстановки переносов в тексте
  - HTML не поддерживает расстановку переносов в словах как это делают текстовые редакторы
- ◆ Текст занимает ширину окна браузера
  - Если вы просто напишите одну длинную строку в коде HTML, то в браузере она будет отформатирована, чтобы текст поместился по ширине в окно. Переносы текста будут добавлены автоматически в местах пробела или дефиса.

# Тэги начертания

## ◆ Жирное начертание

- применяется два тега: `<b>` и `<strong>`.
- `<b>`Жирное начертание шрифта`</b>`  
`<strong>`Сильное выделение текста`</strong>`

## ◆ Курсивное начертание

- определяют два тега: `<i>` и `<em>`.
- `<i>`Курсивное начертание шрифта`</i>`  
`<em>`Выделение текста`</em>`

## ◆ Подчеркивание

- используется тэг `<u>`

## ◆ Перечеркивание

- используется тэг `<s>`

## ◆ Увеличение шрифта (относительно текущего).

- используется тэг `<big>`

## ◆ Уменьшение шрифта (относительно текущего).

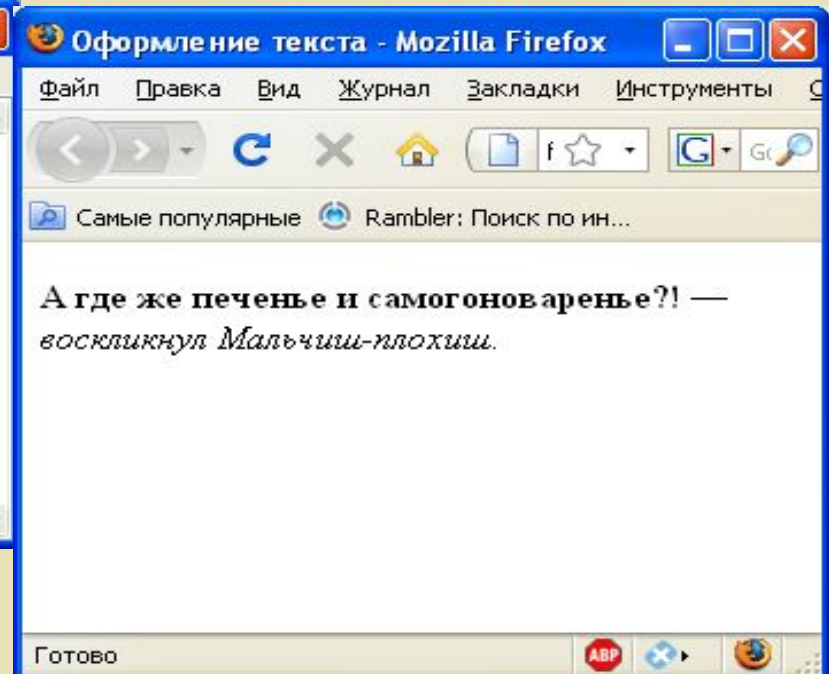
- используется тэг `<small>`

```
firstweb.html - Блокнот
Файл  Правка  Формат  Вид  Справка  Прагма
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0//EN">
<html>
<head>
  <title>оформление текста</title>
</head>
<body>

  <p><strong>А где же печенье и
самогоноваренье?!</strong> –

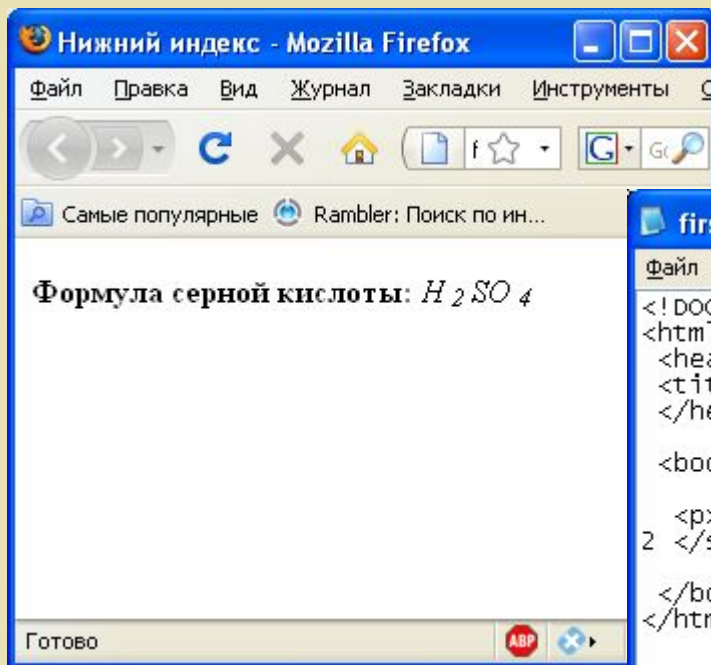
  <em>воскликнул Мальчиш-плохиш</em>.</p>

</body>
</html>
```



# Верхний и нижний индексы

- ◆ `<sup>` — верхний индекс и `<sub>` — нижний индекс. Текст, помещенный в один из этих контейнеров, обозначается меньшим размером, чем базовый текст и смещается относительно горизонтали.



firstweb.html - Блокнот

```
Файл Правка Формат Вид Справка Прагма
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<html>
  <head>
    <title>нижний индекс</title>
  </head>
  <body>
    <p> <b> формула серной кислоты: </b> <i> H <sub>
2 </sub> SO <sub> 4 </sub> </i> k/p>
  </body>
</html>
```

## Специальные символы

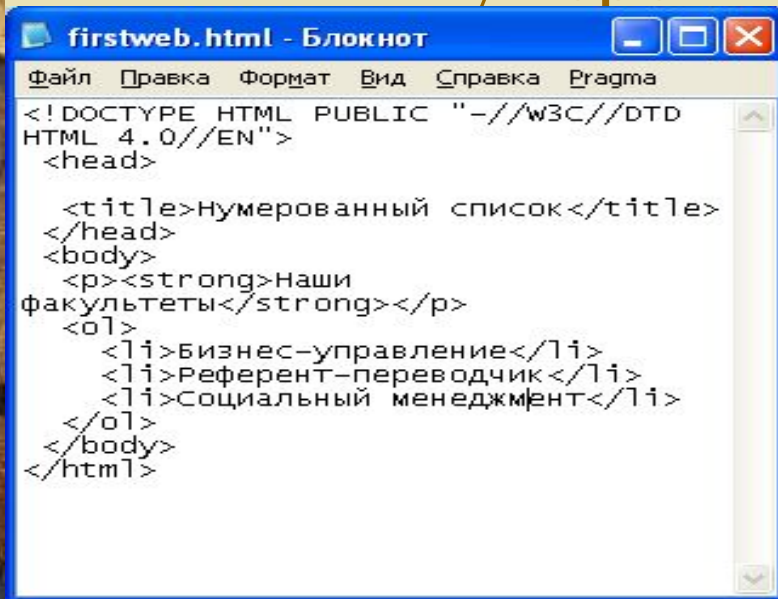
- ◆ Браузер видит символы < или > как начало или конец тега и не показывает их. Встает вопрос как показать эти символы на экране?
- ◆ Для этих символов предусмотрено специальное кодирование. Код строится из символа "&" (амперсанд), фиксированного имени и завершающего символа ";" . В общем это выглядит так: **&имя;**
- ◆ Например, для задания двух пробелов напишем **&nbsp; &nbsp;**
- ◆ Вот фиксированные имена некоторых символов:

Символ	<	&	"	>	пробел	©
Имя	lt	amp	quot	gt	nbsp	copy

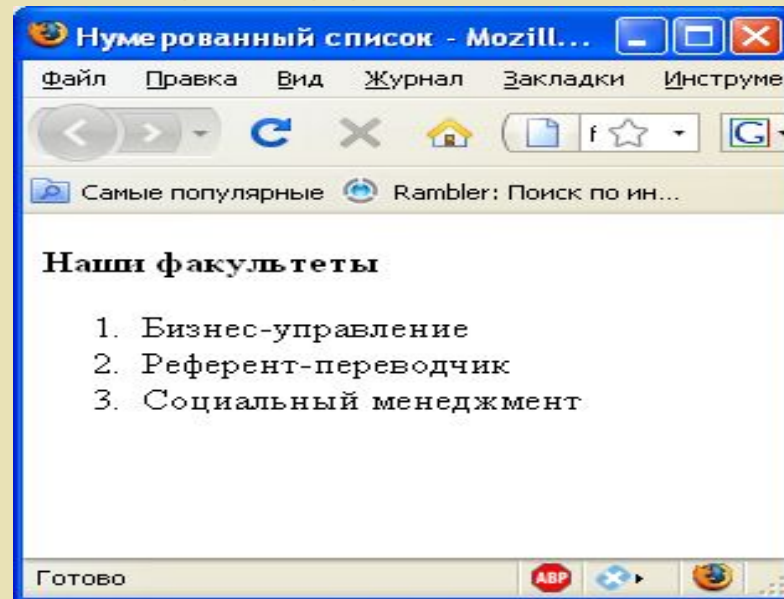
Прочитайте файл «Таблица СИМВОЛОВ»



# Нумерованный список

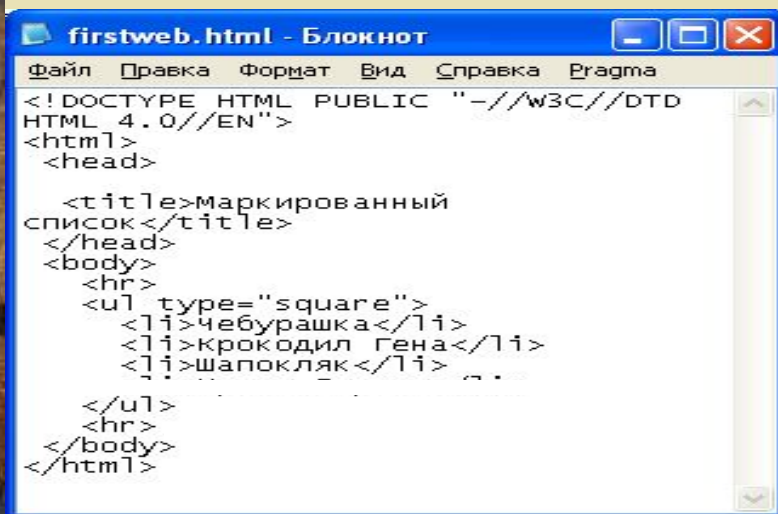


```
firstweb.html - Блокнот
Файл  Правка  Формат  Вид  Справка  Прагма
<!DOCTYPE HTML PUBLIC "-//W3C//DTD
HTML 4.0//EN">
<head>
  <title>нумерованный список</title>
</head>
<body>
  <p><strong>Наши
факультеты</strong></p>
  <ol>
    <li>бизнес-управление</li>
    <li>Референт-переводчик</li>
    <li>Социальный менеджмент</li>
  </ol>
</body>
</html>
```

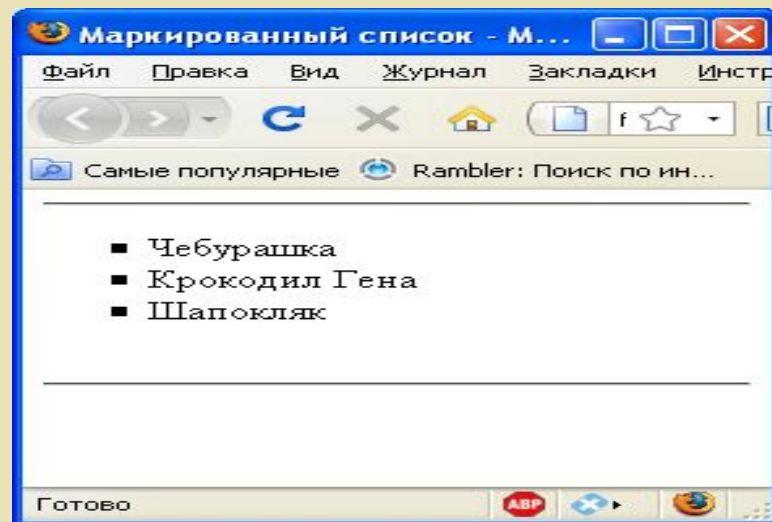


- ◆ Для создания нумерованного списка используется тег **OL (Ordered List)** с необязательными атрибутами: `<OL type="?" start="?" >`
- ◆ Каждый новый элемент списка следует начинать с метки `<LI>` (List Item).
- ◆ Атрибут **type** задает тип списка:
  - **A** устанавливает маркер в виде прописных букв (A, B, C ...)
  - **a** устанавливает маркер в виде строчных букв (a, b, c ...)
  - **I** устанавливает маркер в виде больших римских цифр (I, II, III ...)
  - **i** устанавливает маркер в виде маленьких римских цифр (i, ii, iii ...)
  - **1** устанавливает маркер в виде арабских цифр (1, 2, 3 ...).
- ◆ Атрибут **start** устанавливает начальный маркер в текущем списке.

# Маркированный список



```
firstweb.html - Блокнот
Файл  Правка  Формат  Вид  Справка  Фрагма
<!DOCTYPE HTML PUBLIC "-//W3C//DTD
HTML 4.0//EN">
<html>
<head>
  <title>маркированный
  список</title>
</head>
<body>
  <hr>
  <ul type="square">
    <li>Чебурашка</li>
    <li>крокодил гена</li>
    <li>шапокляк</li>
  </ul>
  <hr>
</body>
</html>
```



- ◆ Для создания маркированного списка используется тег **UL (Unordered List)** с необязательным атрибутом: `<UL type="?" >`
- ◆ Атрибут **type** задает символ маркера
  - **circle** - кружок (устанавливаемый по умолчанию),
  - **square** - квадрат
  - **disk** - закрашенный круг

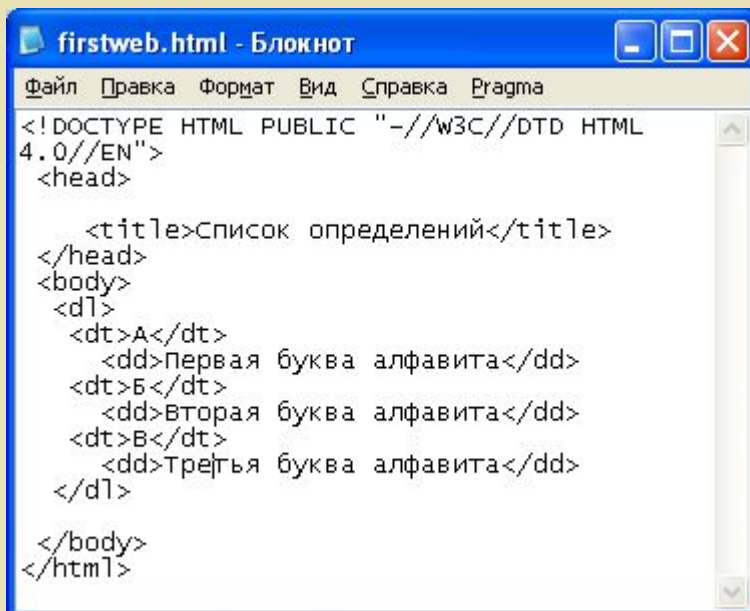
*Попробуйте!*

# Список определений (терминологический список)

Данный список служит для создания списков типа "термин"- "описание".

```
<dl>
  <dt>Термин 1</dt>
  <dd>Определение 1</dd>
  <dt>Термин 2</dt>
  <dd>Определение 2</dd>
</dl>
```

- ◆ Список определений начинается с тэга `<dl>` (Definition List) и завершается тэгом `</dl>`.
- ◆ Каждый термин начинается тэгом `<dt>` (Definition Term), а описание - тэгом `<dd>` (Definition Description).



firstweb.html - Блокнот

```
Файл Правка Формат Вид Справка Прагма
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0//EN">
<head>
  <title>Список определений</title>
</head>
<body>
  <dl>
    <dt>А</dt>
    <dd>Первая буква алфавита</dd>
    <dt>Б</dt>
    <dd>Вторая буква алфавита</dd>
    <dt>В</dt>
    <dd>Третья буква алфавита</dd>
  </dl>
</body>
</html>
```

