

**федеральное государственное автономное образовательное
учреждение высшего образования
«Московский политехнический университет»**

Лекция №1 Основы HTML

Преподаватель: Шульга М.В

Москва 2021

HTML – это язык разметки документов в среде WEB. То, что вы видите при просмотре страницы в Internet, это интерпретация вашим браузером HTML-текста.

Чтобы браузер правильно отображал форматирование к примеру текста т.е. разделял его на абзацы, выделял цитаты, заголовки, списки и.т.д. ему надо как-то сообщить , что мол это заголовок, а это – параграф и.т.д. Этим как раз и занимается язык html.

Тег – оформленная единица HTML-кода. Например, <html>, <body>, <h1>,<h2> и так далее. Все тэги имеют одинаковый формат: они начинаются знаком "<" и заканчиваются знаком ">".

Обычно имеются два тэга – открывающий и закрывающий. Различие в том, что в закрывающем имеется слэш "/". Желательно вводить все теги в нижнем регистре(маленькими буквами), хотя не обязательно.

Пример:

<h1> Крупный заголовок **</h1>**

<h2> Заголовок чуть поменьше **</h2>**

Где тег **<h1>** означает заголовок первого уровня, а тег **<h2>** - второго уровня. В начале тег открывается, затем идет его содержимое, потом он закрывается.

Атрибут – это своего рода дополнительная информация. Атрибуты всегда записываются внутри тега, затем следует знак равенства и детали атрибута, заключённые в двойные кавычки.

Пример:

`<h1> Крупный заголовок </h1>`

`<h2 align="center"> Заголовок чуть поменьше </h2>`

В теге `<h2>` присутствует атрибут `align="center"`, что означает выравнивание по центру, в результате чего содержимое тега выравнивалось по центру ячейки. В теге `<h1>` атрибута `align` (выравнивание) нет, и поэтому выравнивание происходит по умолчанию (по левому краю).

Элемент — понятие, введенное чисто для удобства объяснения. Например, элемент **HEAD** состоит из двух тегов — открывающего **<HEAD>** и закрывающего **</HEAD>**. Таким образом, элемент — более общее понятие, обозначающее пару тегов и участок документа между тегами, на который распространяется их влияние.

Структура web-документа. Вставка комментария.

Каждый HTML-документ, отвечающий спецификации HTML какой-либо версии, должен начинаться со строки объявления версии HTML `<!DOCTYPE...>`, которая обычно выглядит примерно так:

```
<!DOCTYPE HTML>
```

В данном случае мы говорим браузеру, что HTML соответствует международной спецификации версии 5. Именно эта строка помогает сделать сайт, который будет одинаково смотреться во всех браузерах(обычно это три основных Internet Explorer, Opera, Firefox) .

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

Обычно основой головы документа является элемент **TITLE** - заголовок документа. Также там содержится вся техническая информация, различные таблицы стилей и.т.п. А основное содержимое: текст, таблицы, картинки, - находится в теле документа.

На следующем рисунке представлена структура HTML документа.

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<HTML>



<HEAD>

<TITLE>Заголовок документа</TITLE>

</HEAD>

<BODY>

Основной текст документа

</BODY>

</HTML>

Вставка комментария: <!-- -->

Очень полезным при создании сайтов является вставка комментария. Почему? да потому, что через месяц после создания, вам будет трудно разобраться что к чему, ведь кода будет очень много. А если вы будете комментировать свои действия - тогда разобраться будет значительно легче!

Примечание: тег <!-- --> внутри элемента TITLE не действует.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//EN">
```

```
<html>
```

```
<head>
```

```
<title> Сайт о собаках </title>
```

```
</head>
```

```
<!-- начнем работать с телом документа -->
```

```
<body>
```

```
<!--вставляем таблицу с перечнем основных пород собак-->
```

```
...
```

```
</body>
```

```
<!--закончили с телом документа-->
```

```
</html>
```

Создание первой HTML странички.

ШАГ 1:

Создайте на своем компьютере папку с названием вашего будущего сайта, ну к примеру в данном случае актуально будет назвать папку **obuchenie_html**. Внутри папки создайте еще одну папку с названием **www**, а внутри нее еще одну для изображений с названием **img** (позже вставим туда картинки).

ШАГ 2:

Зайдите в созданную на предыдущем шаге папку **www** и создайте внутри нее текстовый документ. Для этого зайдите в нее, кликните правой кнопкой мыши в выпадающем меню выберите Создать - Текстовый документ. Перед собой вы должны видеть примерно следующее:

ШАГ 3:

Теперь, напишите в Блокноте текст простейшей странички:

Приведу пример:

```
<!DOCTYPE HTML PUBLIC>
```

```
<html>
```

```
<head>
```

```
<title> Это моя первая страничка! </title>
```

```
</head>
```

```
<body>
```

```
<!--сейчас я напишу свою первую строчку текста!-->
```

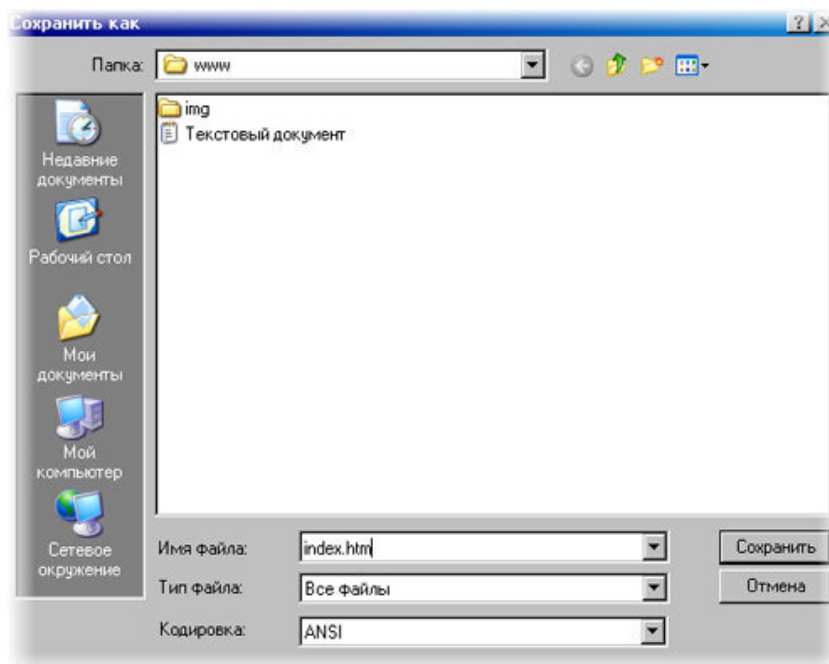
```
Ура!!!! Я создал свою первую страницу на html!!!
```

```
</body>
```

```
</html>
```

ШАГ 4:

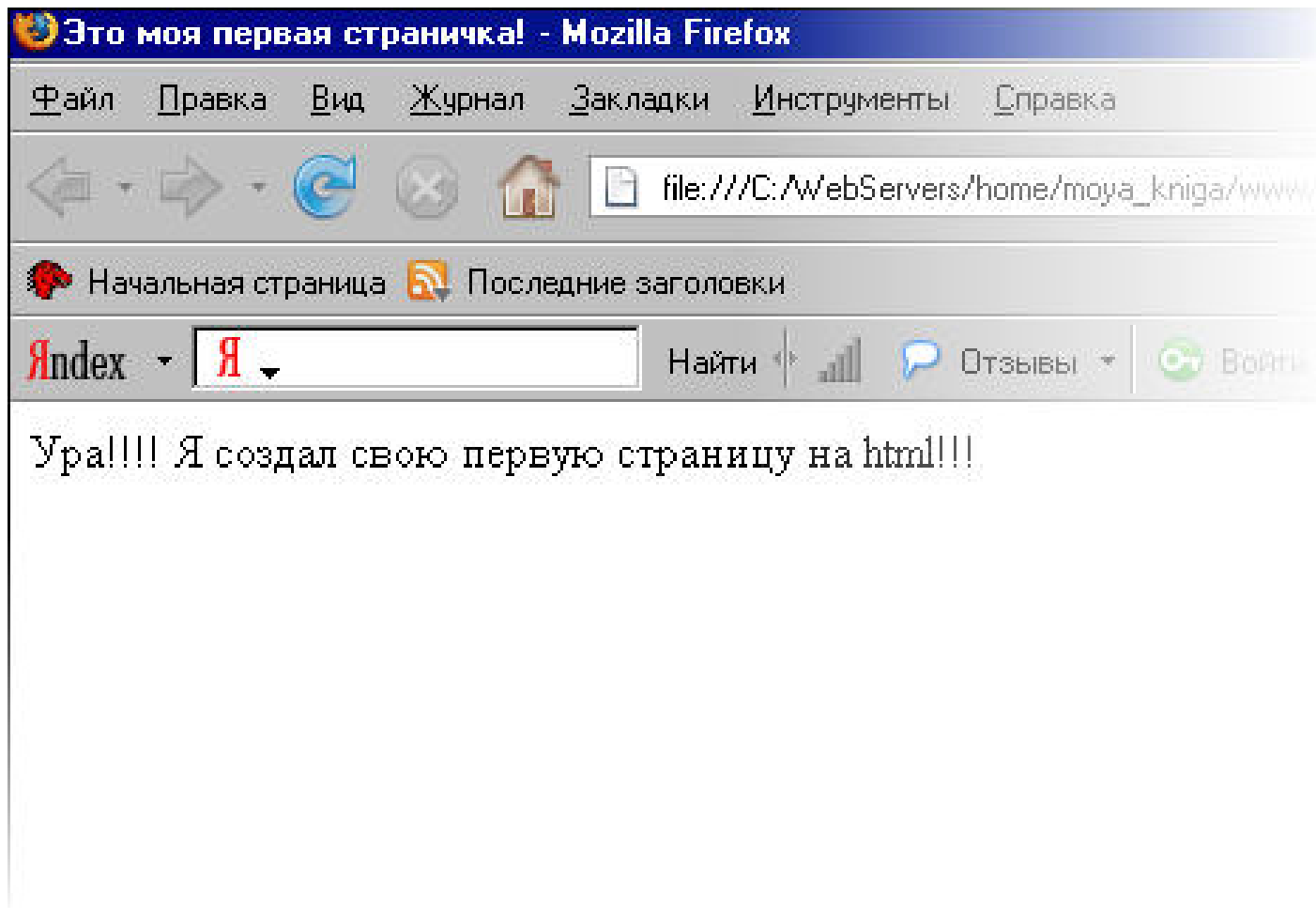
Теперь сохраните ваш документ как «***.htm" (расширение ".htm" указывает, что это HTML-документ. ".html" даст тот же результат. Это не имеет значения). Для этого в меню Файл выберите Сохранить Как:



В пункте Тип файла выберите расширение файла.
После сохранения можете удалить текстовый документ.

ШАГ 5:

Теперь если вы запустите получившийся файл *****.htm** увидите примерно следующее:



Работаем с текстом

Основными элементами форматирования текста в html являются:

- **P** - (Используется для разбивания текста на параграфы)
- **H1,H2,...H6** - (Применяются для создания заголовков 1,2...6 уровней)
- **BR** - (Используется для переноса строки)
- **DIV** , **SPAN** - (Используются для выделения части документа определенным способом)

P - используется для разметки параграфов в html документах.

Атрибуты:

ALIGN - определяет способ горизонтального выравнивания параграфа .

Возможные значения: left, center, right, justify. Это соответственно по левому краю, по центру, по правому краю, и по ширине .По умолчанию имеет значение left.

Приведу пример:

Если внутри тела документа написать следующее:

`<p align="center"> Этот параграф по центру </p>`

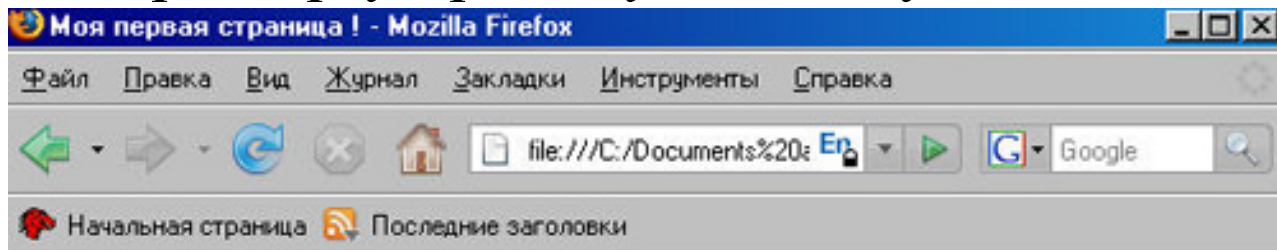
`<p align="left"> Этот по левому краю</p>`

`<p > Этот тоже по левому (т.к. по умолчанию) </p>`

`<p align="right"> Этот по правому краю</p>`

`<p align="justify"> В этом параграфе текст будет выравниваться по ширине (одновременно по левому и правому краям документа). Не понимающие justify браузеры будут выравнивать текст по левому краю</p>`

То при просмотре в браузере получим следующее:



Этот параграф по центру

Этот по левому краю

Этот тоже по левому (т.к. по умолчанию)

Этот по правому краю

В этом параграфе текст будет выравниваться по ширине (одновременно по левому и правому краям документа). Не понимающие justify браузеры будут выравнивать текст по левому краю



Примечание: следите чтобы содержимое в кавычках было написано без пробелов, т.е. `<p align="right">` а не `<p align="right ">` иначе не будет работать!

H1, H2, ... H6 - Применяются для разбивания текста на смысловые уровни - разделы и подразделы. Существует шесть уровней заголовков, различающихся по величине шрифта.

Атрибуты:

ALIGN - определяет способ горизонтального выравнивания заголовков.

Возможные значения: **left, center, right** .По умолчанию - **left**.

Приведу пример:

Если внутри тела документа написать следующее:

<!-- примеры заголовков от 1 до 6 уровня-->

<h1> Большой заголовок **</h1>**

<h2> Заголовок поменьше **</h2>**

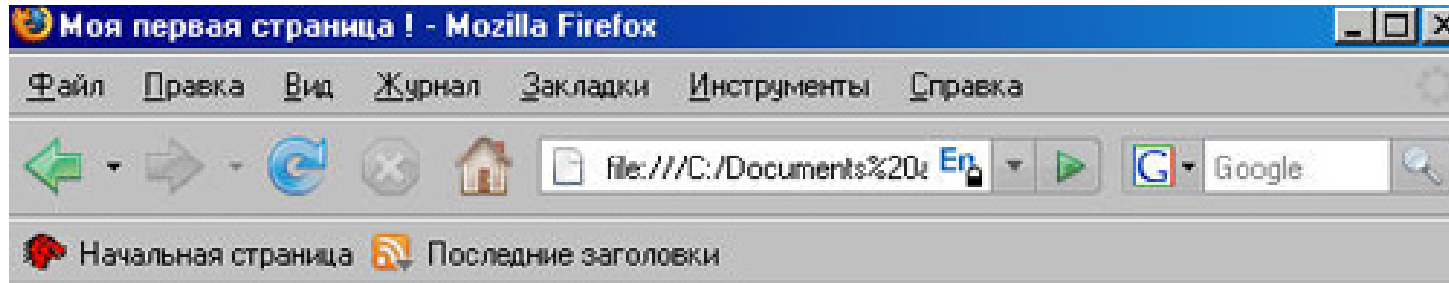
<h3> Еще меньше **</h3>**

<h4> Совсем маленький **</h4>**

<h5> Малюсенький заголовочек **</h5>**

<h6> Ну просто лилипутский заголовочек **</h6>**

В результате получим:



Большой заголовок

Заголовок поменьше

Еще меньше

Совсем маленькій

Малюсенький заголовочек

Ну просто дьявольский сагитационик



Примечание: закрывающий тег обязателен!

BR - данный элемент осуществляет перенос строки. Не имеет закрывающего тега.

Приведу пример:

Если внутри тела документа написать следующее:

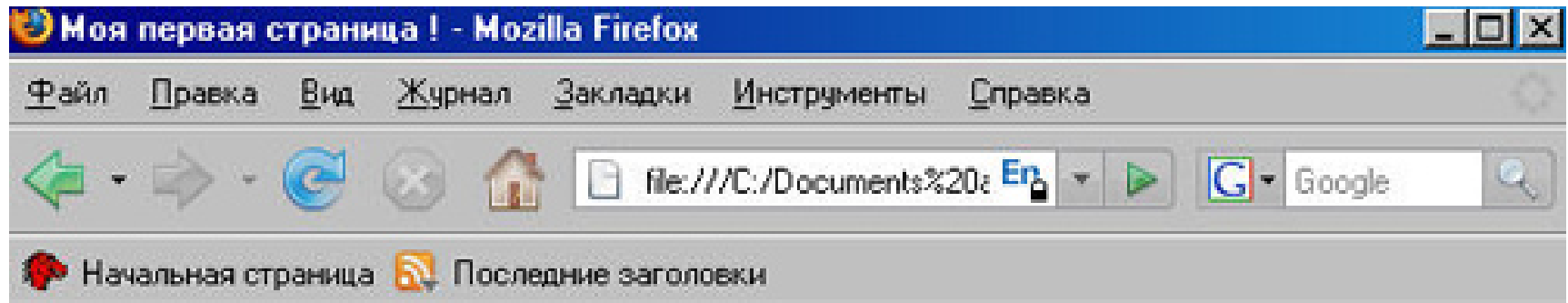
Начинается первая строка **
**

теперь идет вторая **
**

а вот уже и третья

<p> А это параграф и внутри него тоже можно использовать тег **
** переноса строки.Его надо использовать всегда, когда надо перенести строку:) **</p>**

Увидим следующее:



Начинается первая строка
теперь идет вторая
а вот уже и третья

А это параграф и внутри него тоже можно использовать тег
переноса строки.Его надо использовать всегда, когда хочеш перенести строку.)

Примечание: при переносе строки не начинается новый
параграф! Перенос- он и есть перенос!

Чтобы придать тексту ту или иную уникальность в html используются такие элементы как:

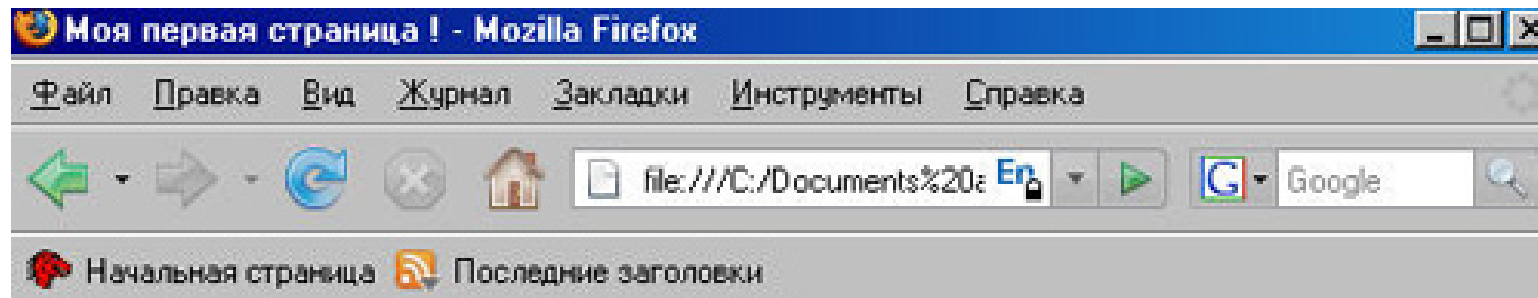
- **STRONG** - Используется для выделения текста жирным
- **EM** - Используется для выделения текста курсивом
- **U** - Выделение текста подчеркиванием
- **S** - Перечеркивание текста
- **SUP** - Создание верхнего индекса
- **SUB** - Создание нижнего индекса
- **FONT** - Изменение цвета, типа, размера шрифта

STRONG - Выделяет текст, заключенный между открывающим и закрывающим тегами, жирным шрифтом. Раньше везде использовался тег **<BOLD>** (или ****), в принципе его можно использовать и сейчас, но это не приветствуется поисковыми системами.

Если в теле документа написать:

**** Этот текст будет выделен жирным **** **
**
**** И этот тоже, но этим тегом лучше не пользоваться! ****

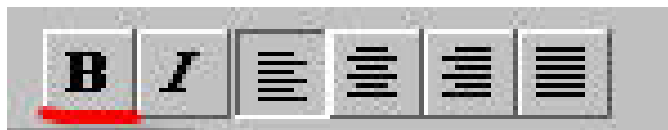
В браузере увидим:



Этот текст будет выделен жирным

И этот тоже, но этим тегом лучше не пользоваться!

Как видите, оба этих тега действуют одинаково. В программе Adobe Dreamweaver на панели форматирования текста написано "**B**"



(просто это название более старое и большинство веб-мастеров привыкли к нему), хотя при нажатии на эту кнопочку в коде пишется тег `` .

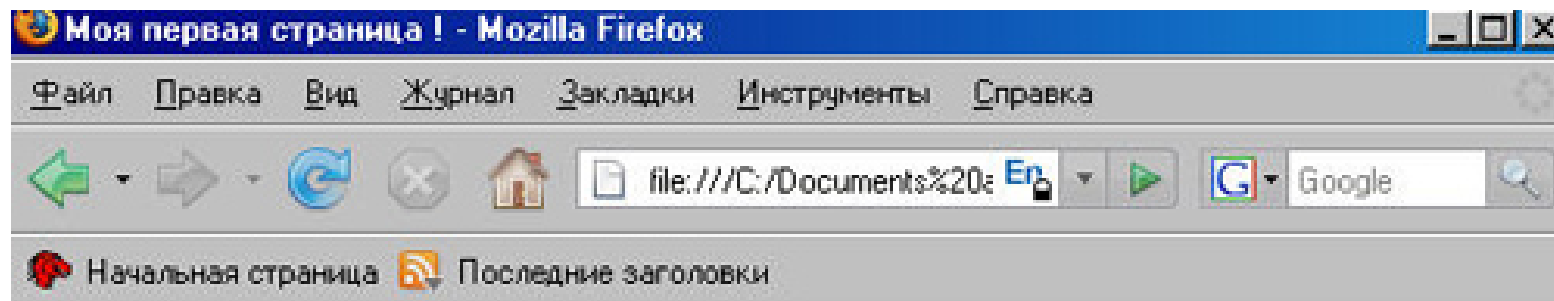
ЕМ (Emphasis) - Выделяет текст, заключенный между открывающим и закрывающим тегами, курсивом. Также аналогичен по действию тег `<I>` , однако его лучше не использовать т.к. это не приветствуется поисковыми системами.

Приведу пример:

`` Этот текст будет выделен курсивом `` `
`

`<i>` И этот тоже, но лучше пользоваться тегом `ЕМ</i>`

В браузере увидим:



Этот текст будет выделен курсивом

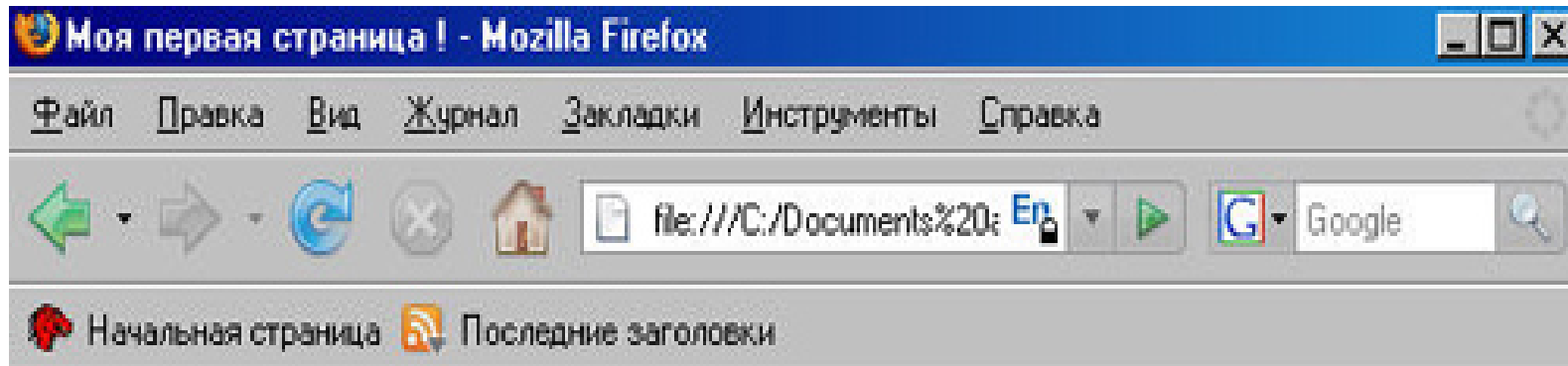
И этот тоже, но лучше пользоваться тегом ЕМ

U(Underline) - Отображает помещенный между открывающим и закрывающим тегами текст как подчеркнутый:

Если в теле документа написать:

`<u> Этот текст будет подчеркнутым!!! </u>`

Увидим в браузере:



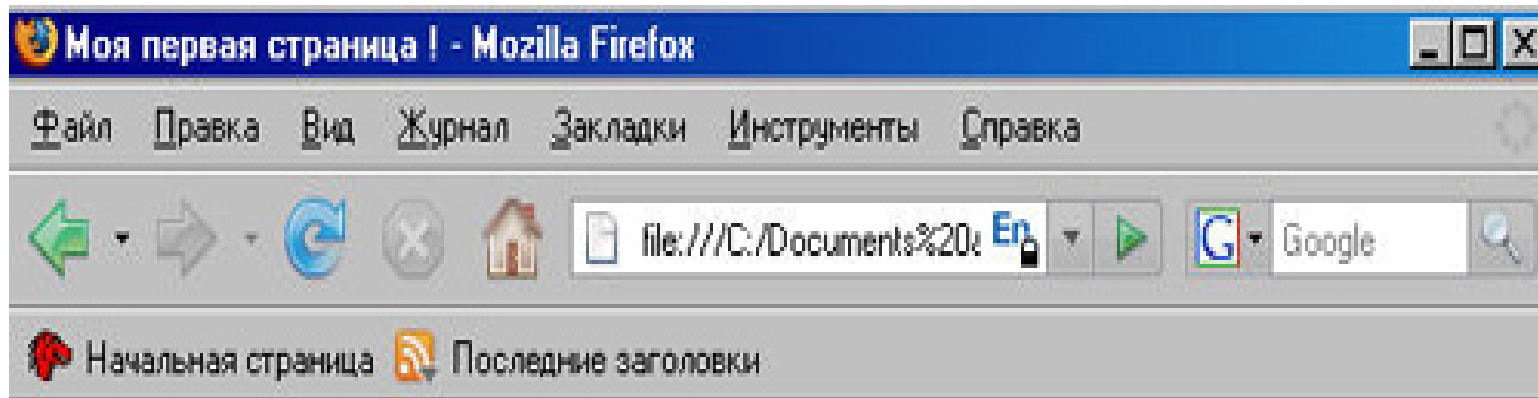
Этот текст будет подчеркнутым!!!

S(Strike) - Текст, помещенный между открывающим <s> и закрывающим </s> тегами, будет перечеркиваться!

Например:

Внимание акция! новая цена <s> 50 000 </s> 20 000 рублей!

Результат:



Внимание акция! новая цена ~~50 000~~ 20 000 рублей!

SUP (Superscript) - Отображает текст, заключенный между открывающим ^{и закрывающим} тегами, как верхний индекс от основного текста.

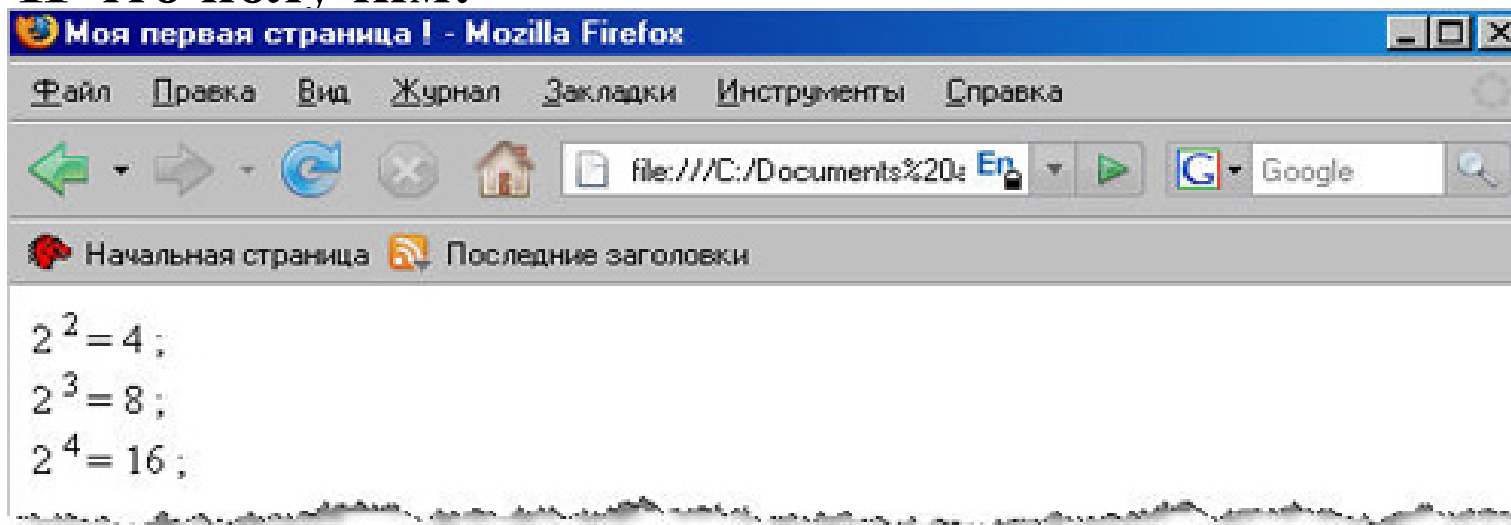
Например:

2 ² = 4 ;

2 ³ = 8 ;

2 ⁴ = 16 ;

И что получим:

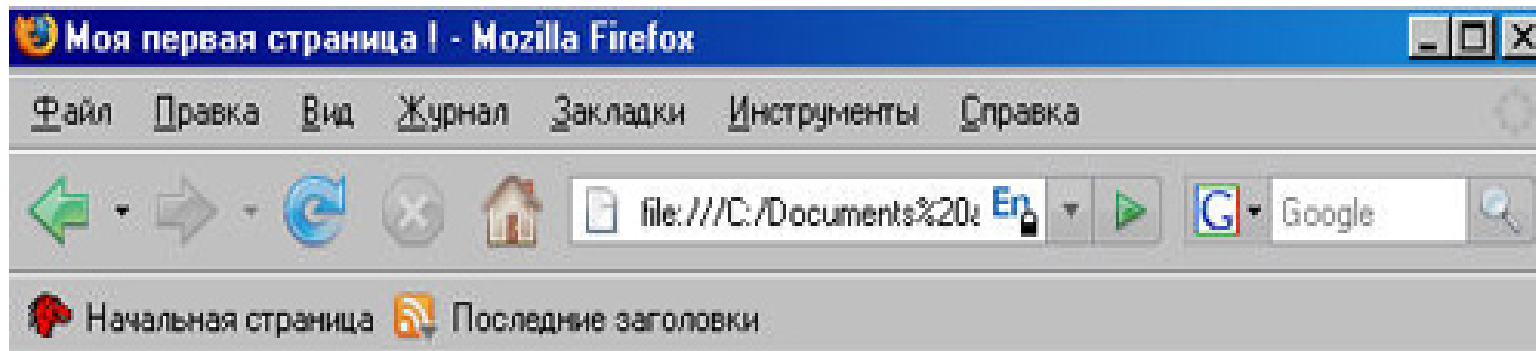


SUB(Subscript) - Отображает текст, заключенный между открывающим _{и закрывающим} тегами, как нижний индекс от основного текста.

Приведу пример:

Формула спирта в химии - C ₂ H ₅ OH

И что мы получим:



Формула спирта в химии - C₂H₅OH

FONT - Позволяет изменять цвет, размер и тип шрифта текста, находящегося между открывающим и закрывающим тегами.

Атрибуты:

SIZE - Определяет размер шрифта. Возможные значения - 1 , 2 , 3 , 4 , 5 , 6 , 7.

COLOR - Определяет цвет текста. Задается либо RGB-значением в шестнадцатиричной системе, либо одним из 16 базовых цветов.

FACE - определяет используемый шрифт. Используйте Times New Roman, Arial, Tahoma, Courier, Courier New. Они установлены почти у всех. Иначе гарантий нет!

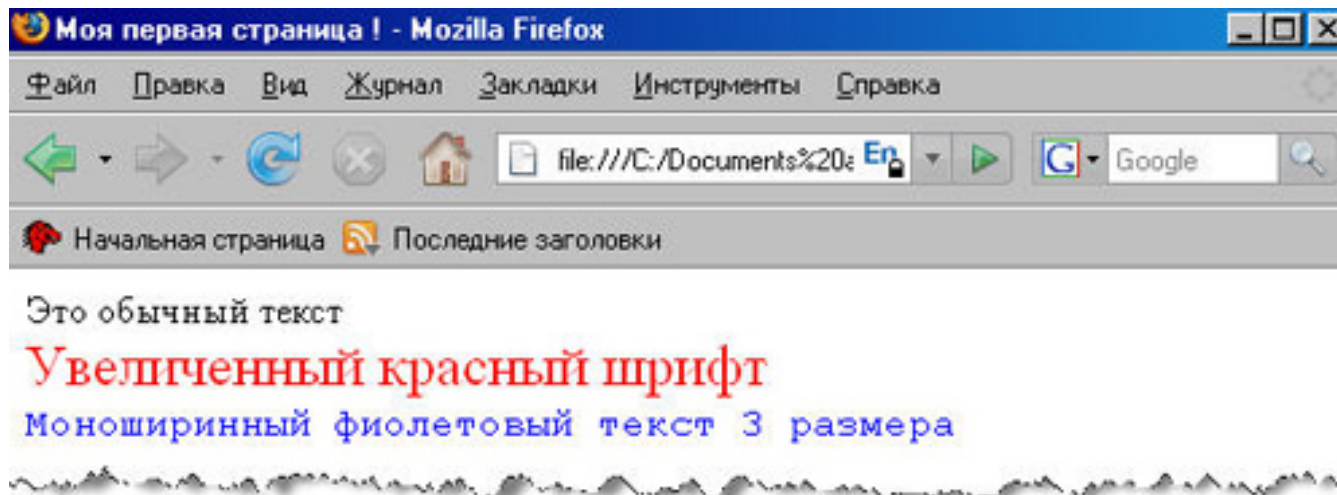
Если в теле документа написать:

Это обычный текст

 Увеличенный красный шрифт

Моноширинный фиолетовый текст 3 размера

При просмотре в браузере увидим:



Если написано `SIZE="+2"` то это означает увеличить на 2 единицы шрифт по сравнению со стандартным. Обычно стандартный размер шрифта равен 3 .

Примечание: В дальнейшем(после изучения `css`) лучше вообще все что связано с типом, размером и цветом шрифта делать через таблицы стилей. Это гораздо удобнее, тем более многие элементы форматирования текста расположенные внутри элемента `FONT`, работают некорректно.

Цвет фона и текста

Мы уже знаем как менять цвет текста, но для этого нам нужно было заключать его в теги `font`, а это не всегда удобно. Иногда, лучше задать цвет текста для всего документа. Также, можно задать и фоновое изображение.

Вот необходимые атрибуты:

BACKGROUND – определяет изображение для "заливки" фона. Значение задается в виде полного URL или имени файла с картинкой в формате GIF или JPG.

BGCOLOR – определяет цвет фона документа.

TEXT – определяет цвет текста в документе.

Все они прописываются для элемента `BODY`. Значения цветов задаются либо RGB-значением в шестнадцатеричной системе, либо одним из 16 базовых цветов.

Пример:

<!-- задаем фоновый цвет и цвет текста -->

<body bgcolor="#FFF8D2" text="red">

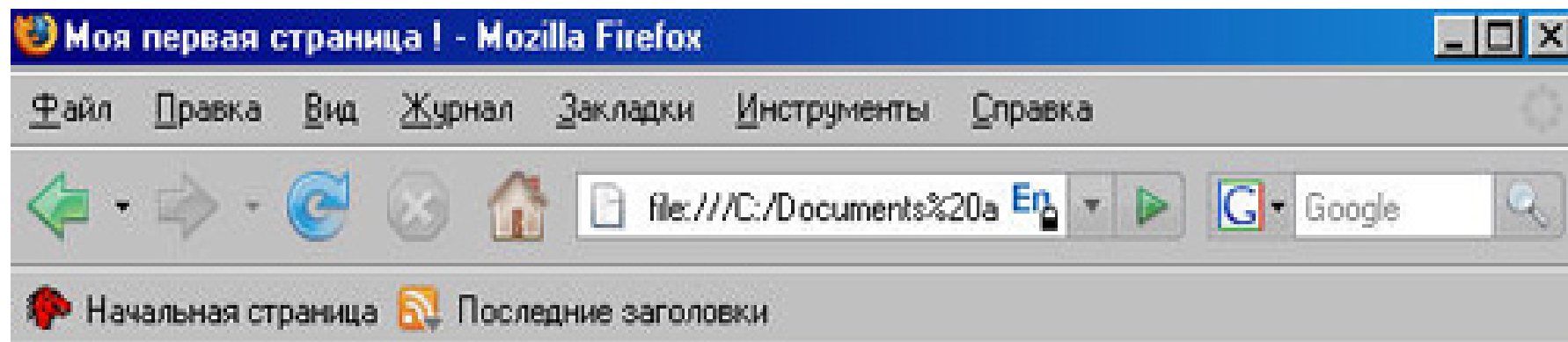
<p> Этот текст будет красный, потому что мы изменили цвет текста в теге БОДИ и теперь весь текст на странице по умолчанию будет красный </p>

<p> В этом абзаце текст будет зеленый, потому что мы заключили его в теги font и придали соответствующий цвет </p>

<p> Теперь текст снова будет красный </p>

</body>

Что мы увидим:



Этот текст будет красный, потому что мы изменили цвет текста в теге БОДИ и теперь весь текст на странице по умолчанию будет красный

В этом абзаце текст будет зеленый, потому что мы заключили его в теги font и придали соответствующий цвет

Теперь текст снова будет красный

Пример:

<!-- задаем фоновое изображение и цвет текста -->

<body background="fon.jpg" text="red">

<p> Этот текст будет красный, потому что мы изменили цвет текста в теге БОДИ и теперь весь текст на странице по умолчанию будет красный </p>

<p>Теперь тут тоже красное и только эти слова зеленые </p>

**<p> Тут как вы поняли текст тоже красный</p>
</body>**

Что мы увидим:

