**Оглавление:**

[HTML теги, параметры тегов 2](#_Toc296115211)

[Структура HTML страницы 4](#_Toc296115212)

[Элемент HEAD 6](#_Toc296115213)

[Элемент BODY 10](#_Toc296115214)

[Гиперссылки 12](#_Toc296115215)

[Текстовые блоки. Заголовки. Абзацы. 19](#_Toc296115216)

[H1,H2,...H6 – Headers 19](#_Toc296115217)

[P – Paragraph 20](#_Toc296115218)

[DIV – Division 21](#_Toc296115219)

[ADDRESS 22](#_Toc296115220)

[BLOCKQUOTE 23](#_Toc296115221)

[BR – Break 24](#_Toc296115222)

[HR – Horizontal Rule 24](#_Toc296115223)

[PRE – Preformatted Text 25](#_Toc296115224)

[LISTING, PLAINTEXT, XMP 25](#_Toc296115225)

[Форматирование текста 26](#_Toc296115226)

[Base Font 27](#_Toc296115227)

[Font 29](#_Toc296115228)

[EM – Emphasis 30](#_Toc296115229)

[STRONG – Stronger emphasis 30](#_Toc296115230)

[TT – Tele Type 31](#_Toc296115231)

[KBD – Keyboard 31](#_Toc296115232)

[CITE – Citation 32](#_Toc296115233)

[Marquee - бегущая строка (текста) 32](#_Toc296115234)

[Списки 34](#_Toc296115235)

[LI – List Item 35](#_Toc296115236)

[Неупорядоченный список 36](#_Toc296115237)

[Упорядоченный список 37](#_Toc296115238)

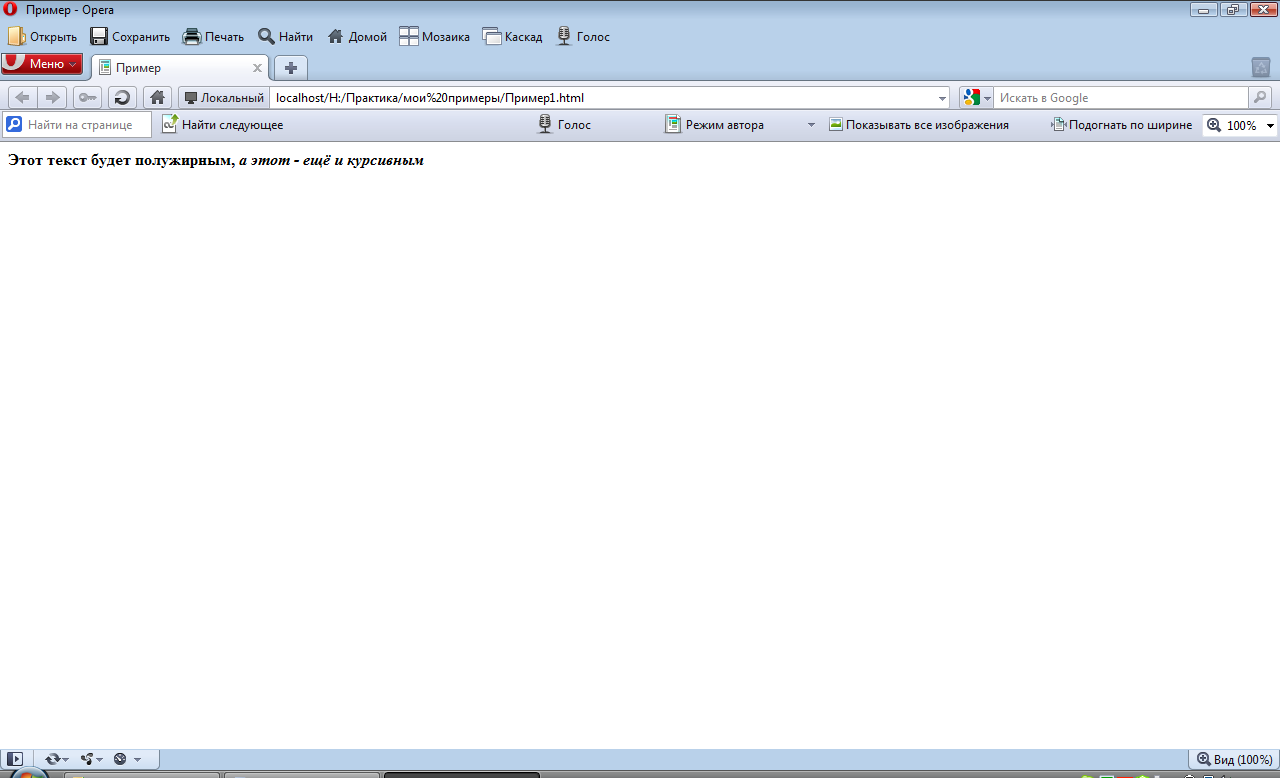
# HTML теги, параметры тегов

HTML — теговый язык разметки документов. Любой документ на языке HTML представляет собой набор элементов, причём начало и конец каждого элемента обозначается специальными пометками — тегами. Элементы могут быть пустыми, то есть не содержащими никакого текста и других данных (например, тег перевода строки <br>). В этом случае обычно не указывается закрывающий тег. Кроме того, элементы могут иметь атрибуты, определяющие какие-либо их свойства (например, размер шрифта для элемента font). Атрибуты указываются в открывающем теге. Вот примеры фрагментов HTML-документа:

* <strong>Текст между двумя тегами — открывающим и закрывающим.</strong>
* <a href="http://www.example.com">Здесь элемент содержит атрибут href.</a>
* А вот пример пустого элемента: <br>

Регистр, в котором набрано имя элемента и имена атрибутов, в HTML значения не имеет (в отличие от XHTML). Элементы могут быть вложенными. Пример:

<b> Этот текст будет полужирным, <i>а этот - ещё и курсивным</i> </b>



Тег – стартовый, конечный маркер элемента. Он определяет область действия элементов и отделяет элементы друг от друга. Тег – оформленная единица HTML-кода. Например, <HEAD>, <FONT>, <BODY>, <HTML> и так далее. Теги бывают начальными (открывающими) и конечными (закрывающими, начинающимися со знака "/"). Например, вышеуказанным тегам соответствуют закрывающие теги </HEAD>, </FONT>, </BODY>, </HTML>.

<тег> </тег> - не просто тег, это контейнер - тег, который может содержать внутри себя другие теги (и текст).

Обратите внимание: <тег1><тег2><тег3> ... </тег3></тег2></тег1>. Только такая очередность закрывающих тегов верна: тег, который мы открыли первым - закрываем последним, второй – предпоследним и т.д. Т.е. следующая очередность нежелательна и не верна, она может привести к ошибкам на вашей страничке: <тег1><тег2><тег3> ... </тег3></тег1></тег2> .

Итак, все, что находится между < и > - это тег. Текст, не находящийся между такими скобками < > - весь виден при просмотре в браузере.

Как мы видим на наших примерах: тегов много, и они разные. В html есть обязательные теги. Обязательные теги, это такие теги, которые вы всегда должны прописывать для каждой своей страницы, даже не смотря на бури, ураганы и другие стихийные бедствия.

Пример:

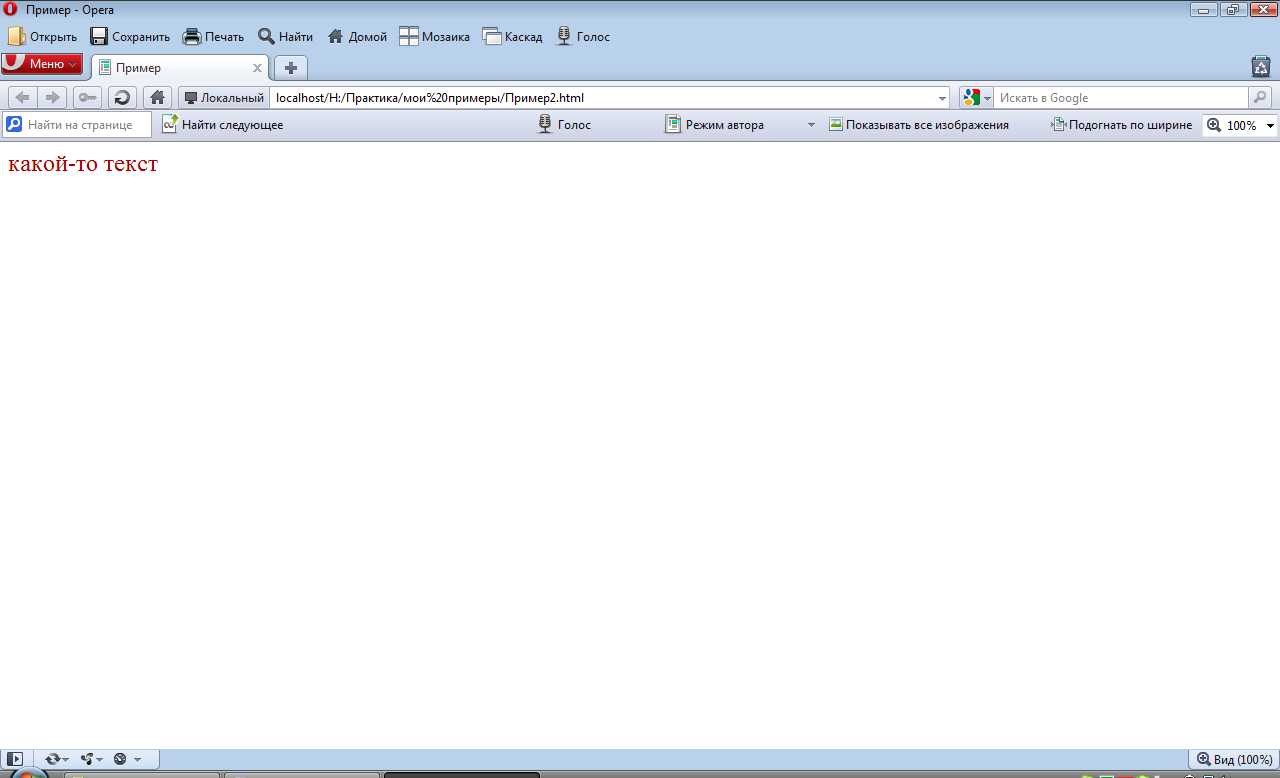
<html>

Этот тег должен открывать документ. Если есть открывающий тег, то должен быть и закрывающий:

</html>

Одному тегу может быть присвоено несколько атрибутов.

<FONT SIZE="+2" COLOR="#AA0000"> какой-то текст</FONT>



# Структура HTML страницы

HTML (от англ. HyperText Markup Language — «язык разметки гипертекста») — стандартный язык разметки документов во Всемирной паутине. Большинство веб-страниц создаются при помощи языка HTML (или XHTML). Язык HTML интерпретируется браузером и отображается в виде документа, в удобной для человека форме. HTML является приложением («частным случаем») SGML (стандартного обобщённого языка разметки) и соответствует международному стандарту ISO 8879. XHTML же является приложением XML.

Для того чтобы текстовый файл превратился в HTML-файл, поменять его расширение с ".txt" на ".html" недостаточно. Надо соблюсти "правило первой строки":

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 3.2 Final//EN">

Эта строка поможет браузеру определить, как правильно интерпретировать данный документ, но она не является обязательной. В данном случае мы говорим браузеру, что HTML соответствует международной спецификации версии 3.2, которая хоть и не отличается новизной, но, в отличие от более поздних версий, является полноценным, широко распространенным стандартом без каких-либо неопределенностей. Как видно из примера, самый короткий html-документ состоит буквально из одной строки.

На самом деле одной строкой дело не обходится. После объявления версии и типа документа необходимо обозначить его начало и конец. Это делается с помощью тега-контейнера <HTML>. Необходимо отметить, что любой HTML-документ открывается тегом <HTML> и им же закрывается.

Затем, между тегами <HTML> и </HTML> следует поместить заголовок и тело документа. Вот как должен выглядеть ваш базовый HTML-файл перед началом работы:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">

<HTML>

<HEAD>

<TITLE>Заголовок документа</TITLE>

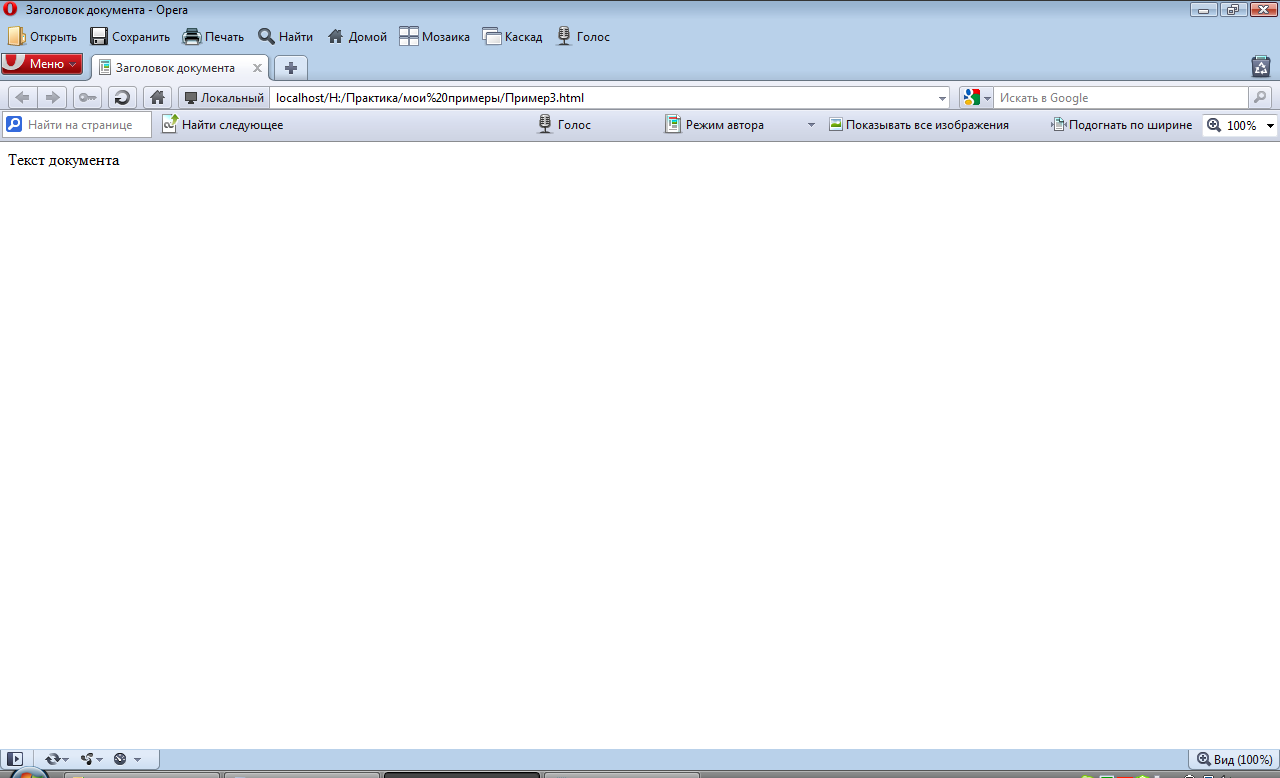
</HEAD>

<BODY>

Текст документа

</BODY>

</HTML>



## Элемент HEAD

HEAD определяет начало и конец заголовка документа. Является контейнером для элементов, содержащих техническую информацию о документе. (TITLE,BASE,STYLE,LINK,META).

Пример:

<HTML>

<!-- Начинаем заголовок... -->

<HEAD>

<title>Справочник по HTML</title>

</HEAD>

<!-- ...закончили. Дальше пошло тело документа -->

<BODY>

Текст документа

</BODY>

</HTML>

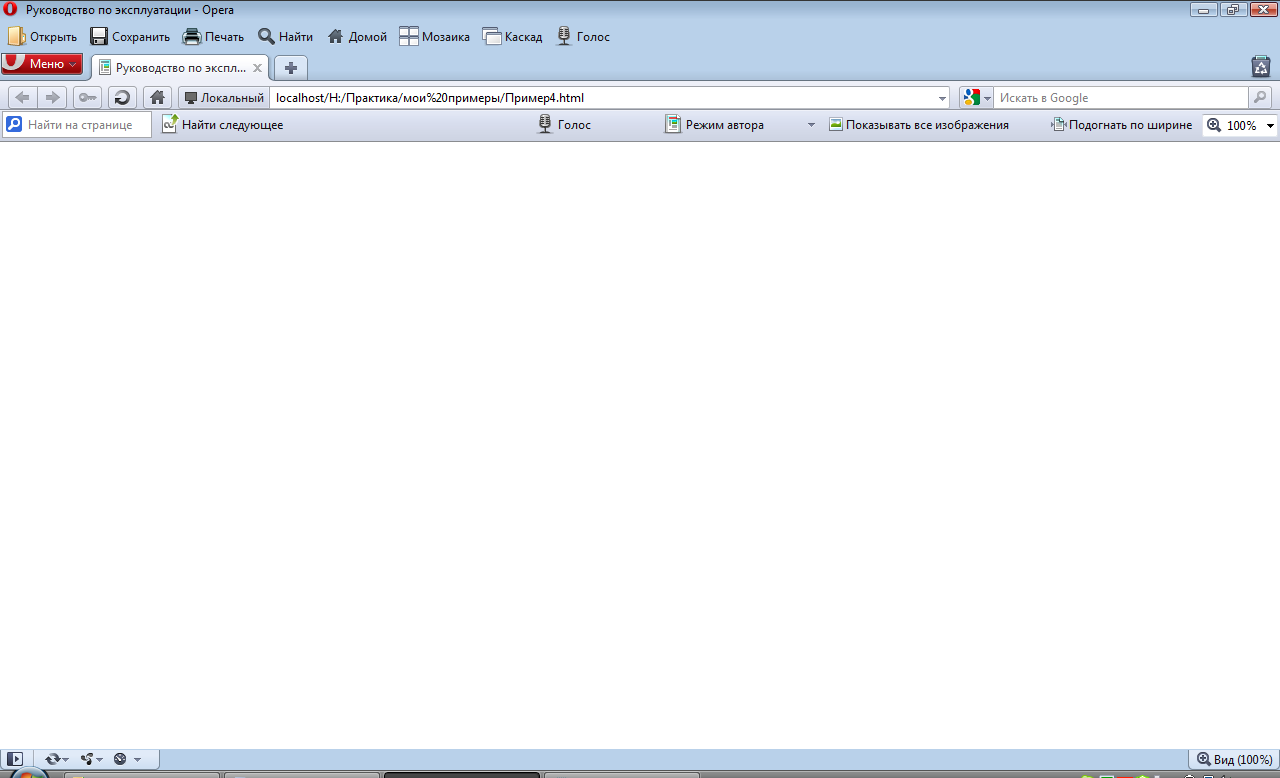
***Title*** определяет имя всего документа. Имя, как правило, отображается в заголовке окна браузера. Данный элемент обязателен для любого HTML-документа и может быть указан не более одного раза.

Пример:

<HEAD>

<TITLE>Руководство по эксплуатации</TITLE>

</HEAD>



***Base*** указывает базовый адрес текущего документа (URL), который станет отправной точкой для расчета относительных адресов внутри документа. Элемент не имеет конечного тега. Обязательно присутствие хотя бы одного из атрибутов.

Атрибуты:

HREF – определяет базовый адрес (URL) текущего документа.

TARGET – определяет имя фрейма, которое будет использоваться в гиперссылках по умолчанию. Это может вам пригодиться, если вы хотите открывать все ссылки документа в другом фрейме.

Пример:

<HEAD>

<!-- Пусть браузер думает, что находится по адресу: -->

<BASE HREF="http://www.igf.ru/other/index.html">

<TITLE>Руководство по эксплуатации</TITLE>

</HEAD>

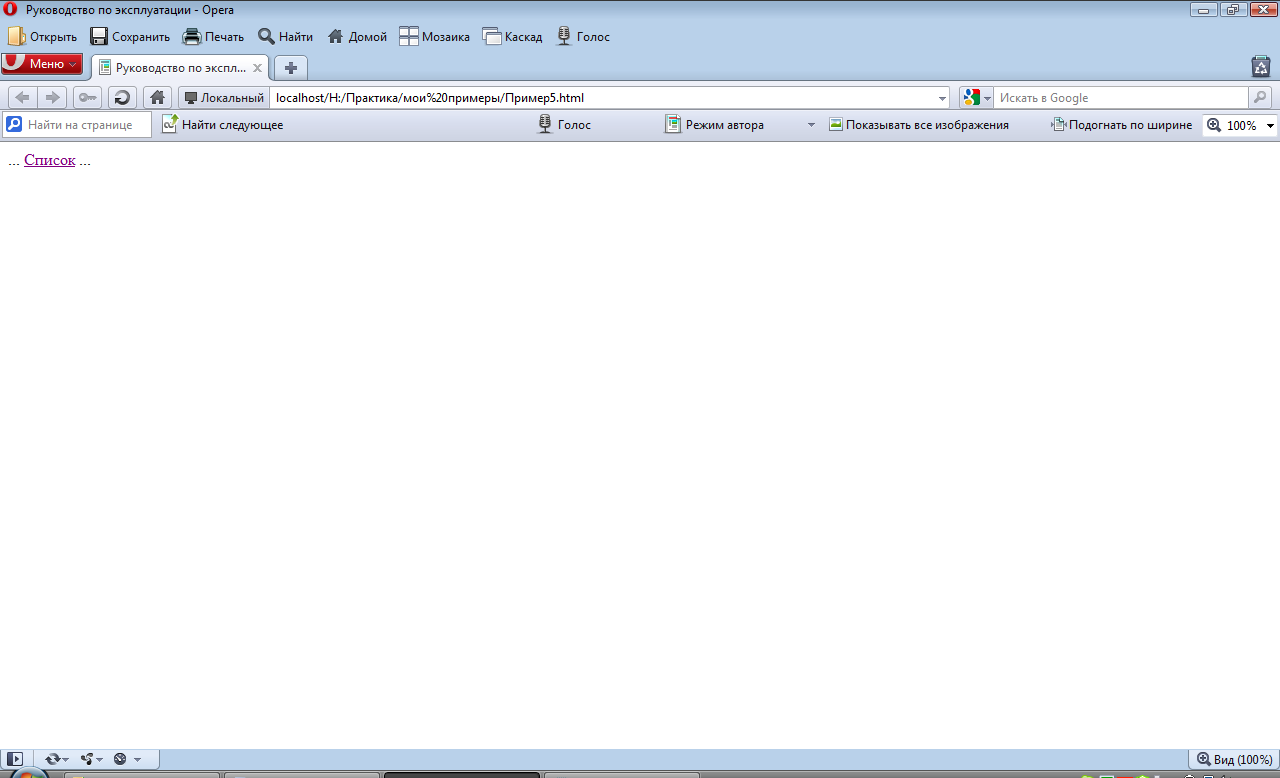
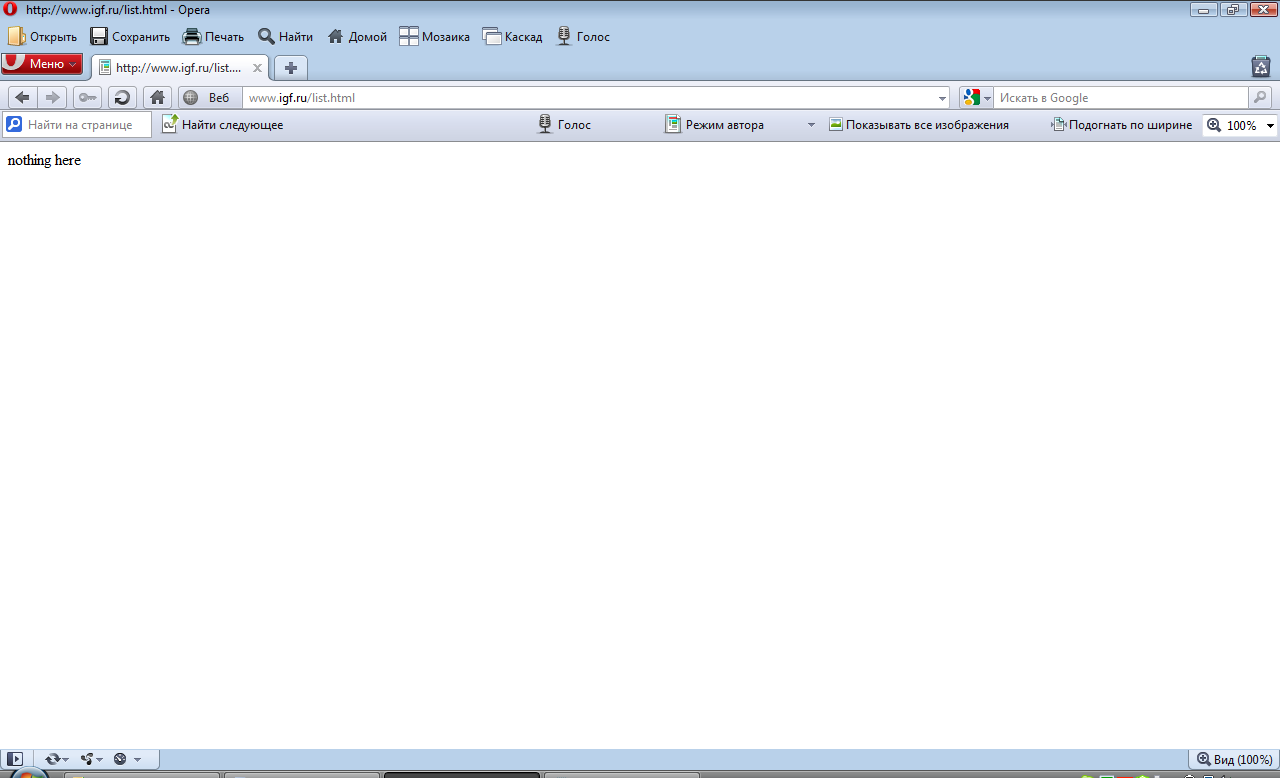
...

<!-- А теперь создадим относительную ссылку на документ -->

<!-- http://www.igf.ru/list.html -->

<A HREF="../list.html"> Список </A>

...

***Style*** используется для вставки в документ таблицы стилей (CSS – Cascade Style Sheet).

Атрибуты:

TYPE – обязательный атрибут. Определяет MIME-тип вставляемого блока стилей. Как правило, значением этого атрибута является "text/css".

TITLE – определяет имя создаваемой таблицы стилей. Необходим, если вы собираетесь использовать несколько элементов STYLE в одном документе. В этом случае браузер должен спросить пользователя, какой из предложенных стилей будет применен к документу.

Пример:

<HEAD>

<TITLE>Пример использования таблицы стилей</TITLE>

<!-- Вставляем таблицу стилей -->

<STYLE TYPE="text/css" TITLE="Cool table">

<!-- A {text-decoration : none;}

P {color : blue; font-size : 12pt; font-family : Arial;}

H1 {color : red; font-size : 18pt;} -->

</STYLE>

<!-- ... кончили втыкать -->

</HEAD>

Элемент ***LINK*** описывает взаимосвязь документа с другими документами на сайте, указывая его место в иерархической структуре сайта. Элемент не имеет конечного тега. В заголовке может содержаться несколько элементов LINK.

Атрибуты:

HREF – определяет URL объекта.

REL – определяет тип взаимосвязи текущего документа с объектом, определенным атрибутом HREF. Возможные значения:

stylesheet – указывает на файл, содержащий таблицу стилей (CSS) для текущего документа. Браузер загрузит css-файл с указанного в атрибуте HREF адреса и применит его к текущему документу.

home – указывает на заглавную страницу вашего сайта

toc, contents – указывают на файл, содержащий оглавление данного документа.

index – указывает на файл, содержащий информацию для индексного поиска по текущему документу.

glossary – указывает на файл, содержащий перечень терминов, относящихся к текущему документу.

copyright – указывает на страницу сайта, в которой говорится о его создателях, авторских правах и т.п.

up, parent – указывает на "родительскую" страницу (документ, стоящий на ступеньку выше в иерархической структуре вашего сайта).

child – указывает на "дочернюю" страницу (документ, стоящий на ступеньку ниже в иерархической структуре вашего сайта).

next – указывает на следующую страницу в последовательности документов (напр. следующую страницу электронного каталога, документации или словаря).

previous – указывает на предыдущую страницу в последовательности документов.

last, end – указывает на последнюю страницу в последовательности документов.

first – указывает на первую страницу в последовательности документов.

help – указывает на страницу с подсказкой (напр. по навигации по вашему сайту).

TYPE – определяет MIME-тип для объекта, указанного в атрибуте HREF.

Пример 1:

<HEAD>

<TITLE>Элемент DIV</TITLE>

<LINK REL="HOME" TITLE="HTML-справочник" HREF="index.html">

<LINK REL="UP" TITLE="Текстовые блоки" HREF="textblocks.html">

<LINK REL="PREVIOUS" TITLE="Элемент P" HREF="p.html">

<LINK REL="NEXT" TITLE="Элемент ADDRESS" HREF="address.html">

</HEAD>

Пример 2:

<HEAD>

<TITLE> Заголовок </TITLE>

<LINK REL="stylesheet" TYPE="text/css" HREF="deco1.css">

</HEAD>

## Элемент BODY

BODY указывает начало и конец тела HTML-документа. Между начальным и конечным тегами содержится текст документа, изображения и таблицы. Одним словом, все HTML-элементы, отвечающие за отображение документа, управление им и гипертекстовые ссылки. Элемент BODY должен встречаться в документе не более одного раза.

Атрибуты:

MARGINHEIGHT – определяет ширину (в пикселях) верхнего и нижнего полей документа. Работает только в браузерах Netscape.

TOPMARGIN – определяет ширину (в пикселях) верхнего и нижнего полей документа. Работает только в браузерах Internet Explorer.

MARGINWIDTH – определяет ширину (в пикселях) левого и правого полей документа. Работает только в браузерах Netscape.

LEFTMARGIN – определяет ширину (в пикселях) левого и правого полей документа. Работает только в браузерах Internet Explorer.

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.

LINK – определяет цвет гиперссылок в документе.

ALINK – определяет цвет подсветки гиперссылок в момент нажатия.

VLINK – определяет цвет гиперссылок на документы, которые вы уже просмотрели.

Значения атрибутов BGCOLOR, TEXT, LINK, ALINK и VLINK задаются либо RGB-значением в шестнадцатеричной системе, либо одним из 16 базовых цветов.

Пример:

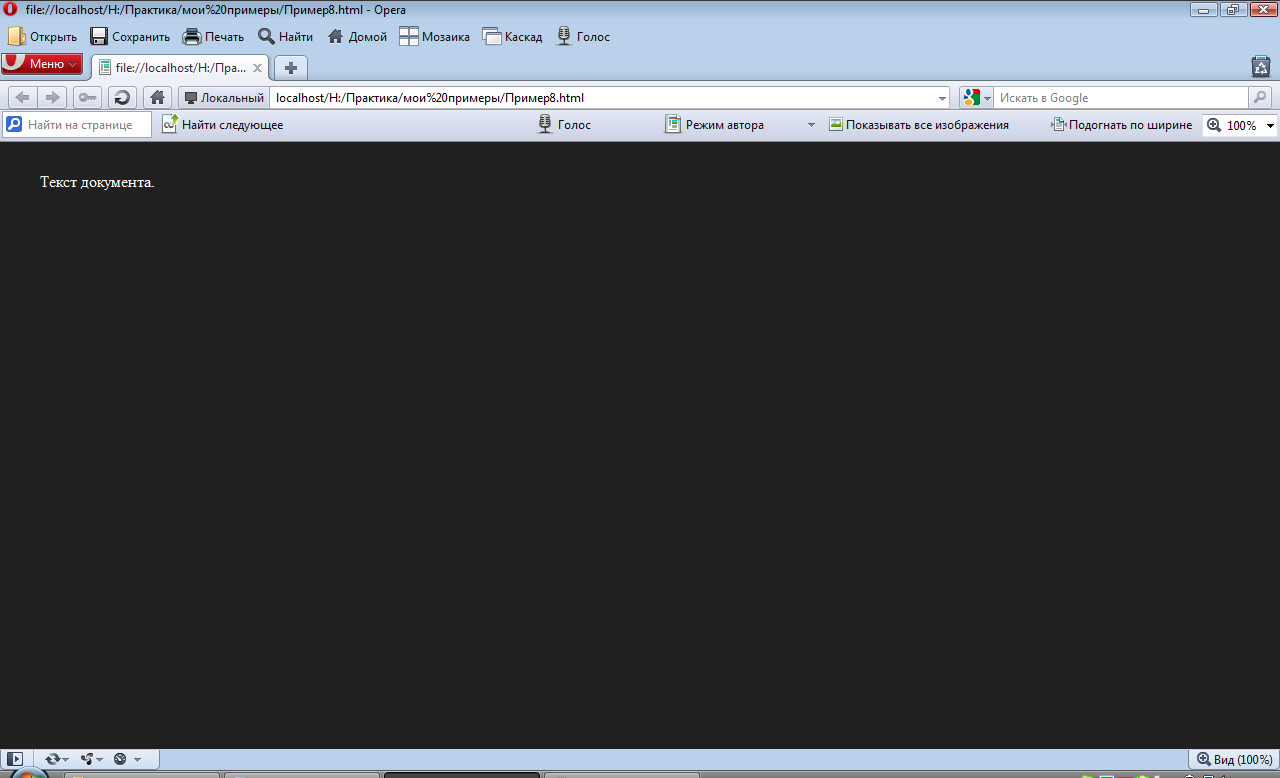
<HTML>

<BODY BACKGROUND="images/bricks.jpg" BGCOLOR="#202020" TEXT="#FFFFFF" LINK="#FF0000" VLINK="#505050" MARGINHEIGHT="30" TOPMARGIN="30" LEFTMARGIN="40" MARGINWIDTH="40">

Текст документа.

</BODY>

</HTML>



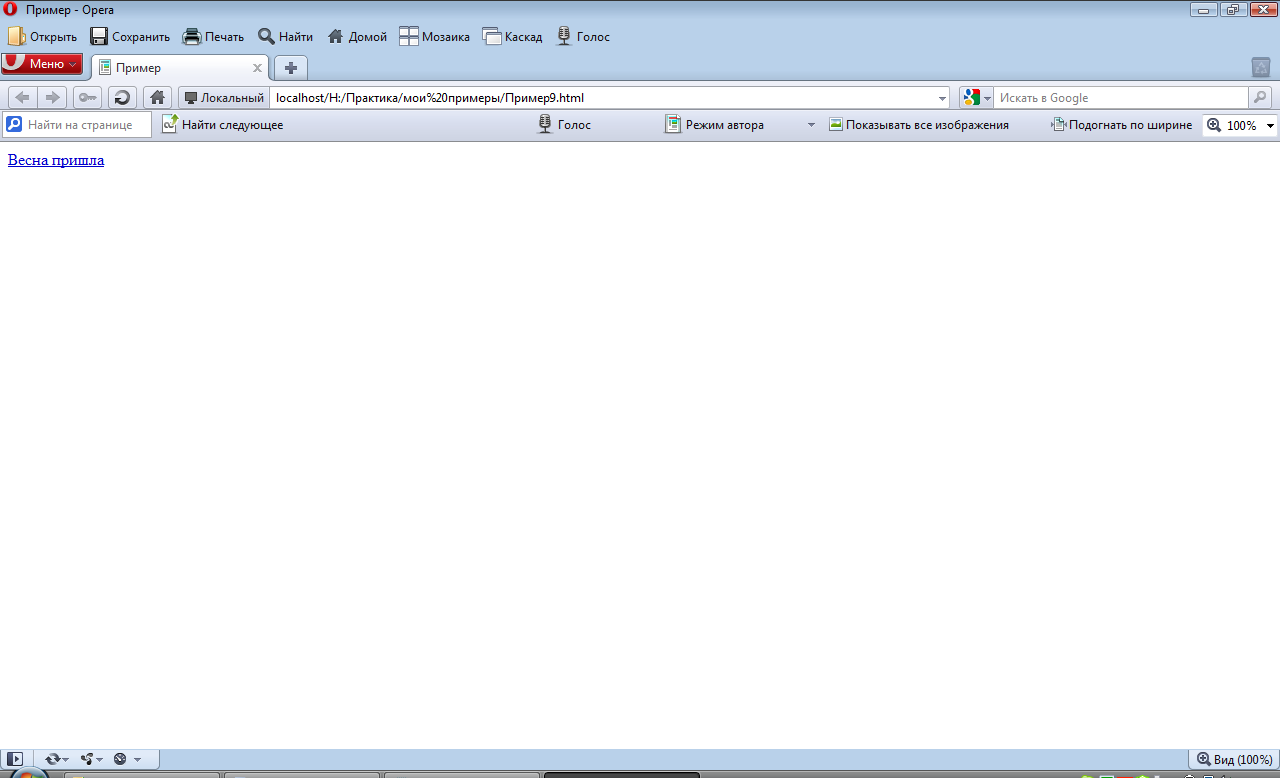
# Гиперссылки

Страница может состоять из нескольких документов. Один из них главный - он открывается первым и должен обязательно лежать на вашем сайте в интернете. Остальные документы могут лежать в одной директории (папке), а могут в разных. При помощи ссылок мы связываем эти документы.

Ссылки на другие документы в HTML создаются либо с помощью элемента A, либо с помощью навигационных карт. Элемент A применяется, если ссылкой планируется сделать часть текста или целое изображение. Навигационные карты имеет смысл применять, если ссылкой будет часть изображения.

***Тег <А></А>*** делает ссылкой заключенную в него картинку или фразу (текст). Пример:

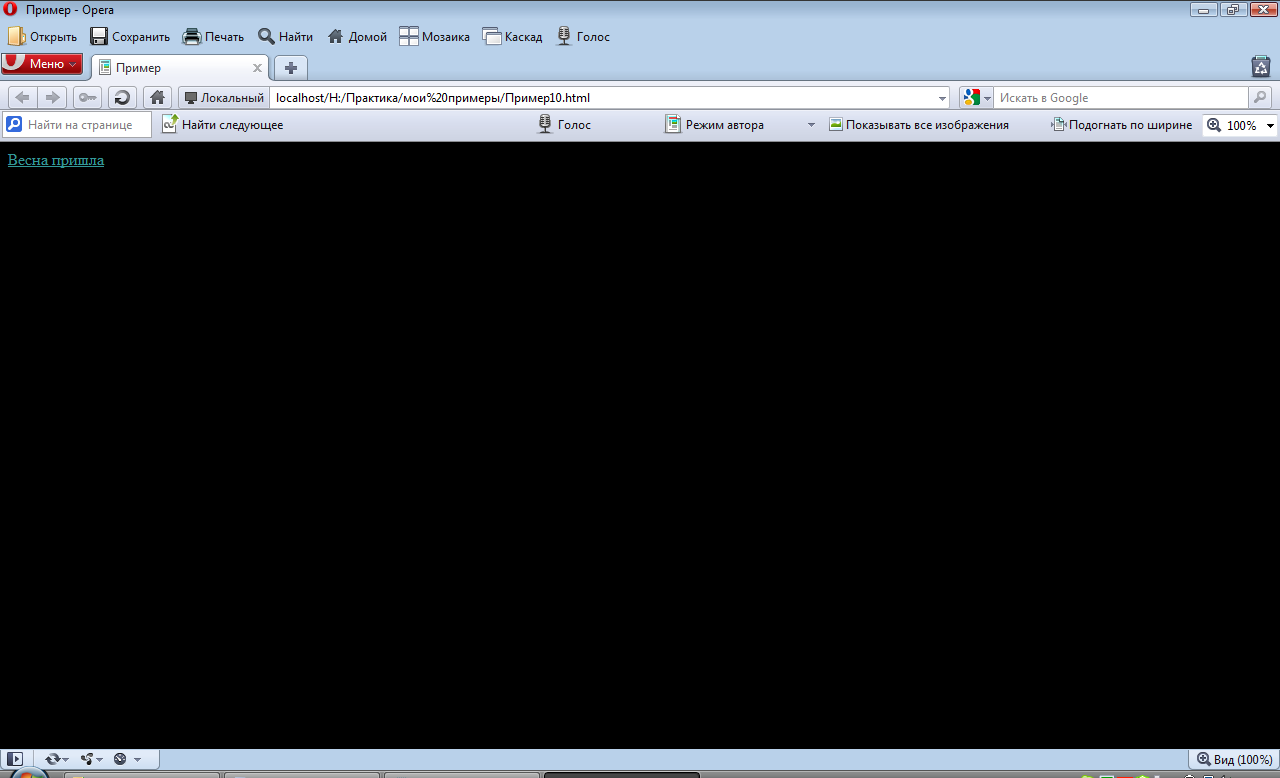
<A HREF=”1.html”> Весна пришла </A>



В этом примере 1.html – ссылка, Весна пришла – текст при нажатии на который будет работать ссылка.

Для всех ссылок в документе мы можем прописать цвета: link - цвет просто ссылки, alink - цвет активной ссылки (нажатой), vlink - цвет уже посещенной ссылки. Пример:

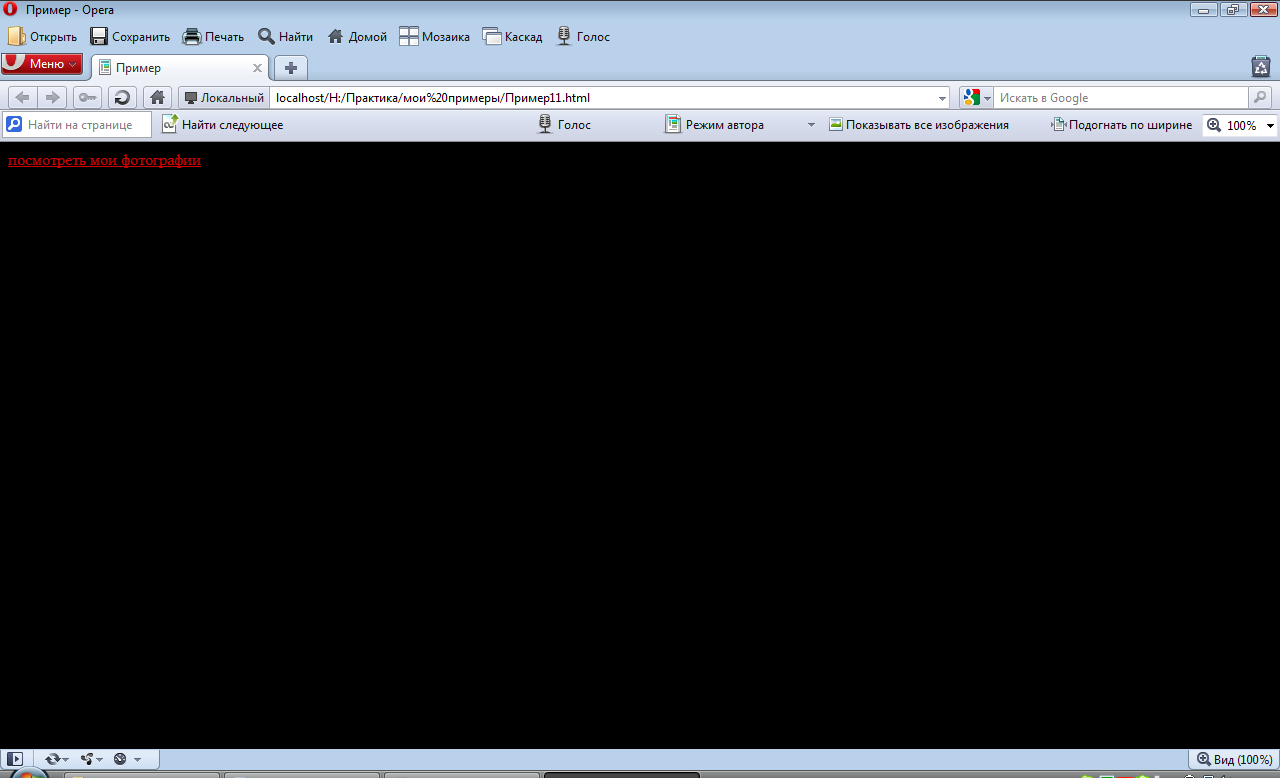
<body text="#336699" bgcolor="#000000" link="#339999" alink="#339999" vlink="#339999">



Как и цвет для всего текста в документе, цвета ссылок мы прописываем в <body>. В нашем примере цвета для просто ссылки, активной и посещенной - одинаковые, но они могут быть разными.

Для текстовых ссылок мы тоже можем задать разный цвет – это делается при помощи тега <font> и его атрибута color:

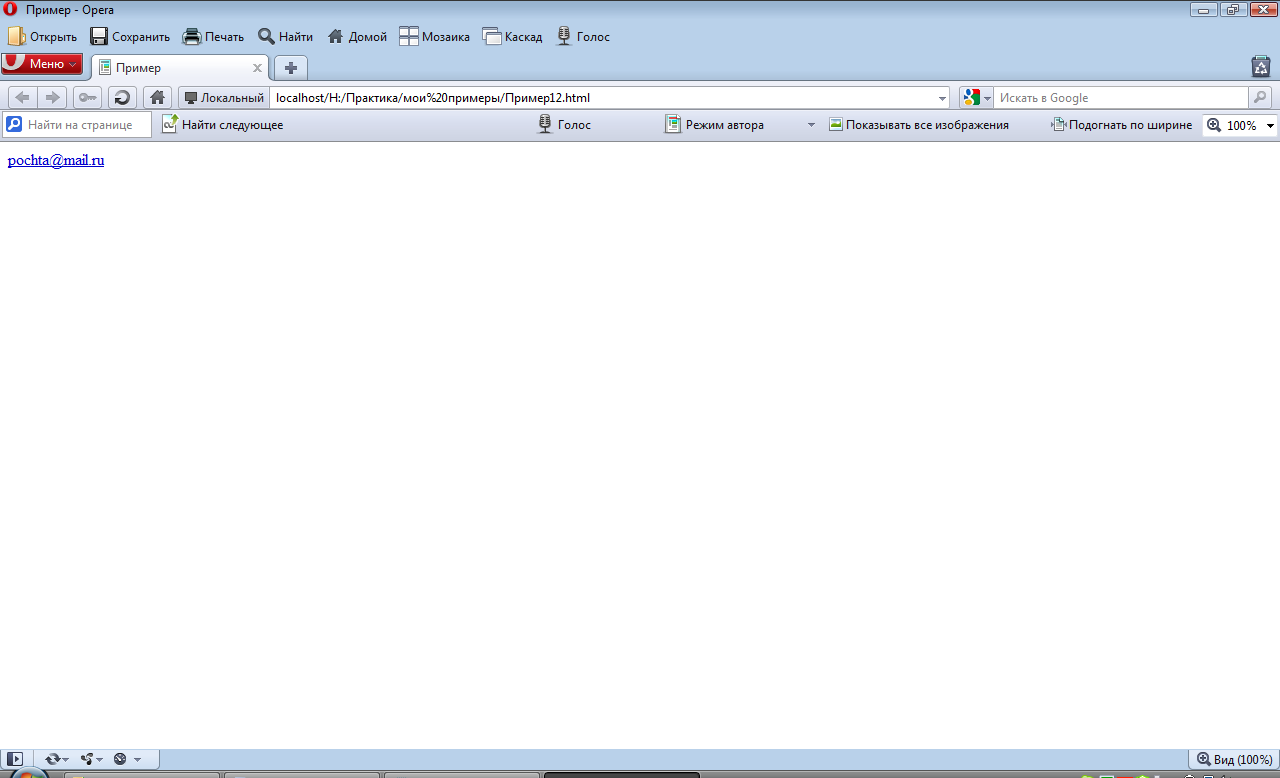
<a href="prf.html"><font color="#CC0000">посмотреть мои фотографии</font></a>



Обратите внимание, <font color="…"></font> - прописывается внутри тега <a></a>, если вы пропишите иначе, то у вас не получится задать вашей ссылке цвет отличный от цвета других ссылок в документе.

***Ссылка на почтовый ящик*** прописывается немного иначе, чем ссылка на другой документ (страницу, сайт):

<a href="mailto:pochta@mail.ru"> pochta@mail.ru </a>



Это надо запомнить и не пытаться прописывать по-другому. Ссылка на документ, картинку или файл - одно дело, а ссылка на почтовый ящик – другое.

У mailto есть еще некоторые опции:

* ?subject=Тема письма
* &Body=Текст вашего сообщения
* &cc=copy@mail.ru,copy2@mail.ru (копии письма через запятую)
* &bcc=hidden\_copy@mail.ru,hidden\_copy2@mail.ru (скрытые копии письма через запятую)

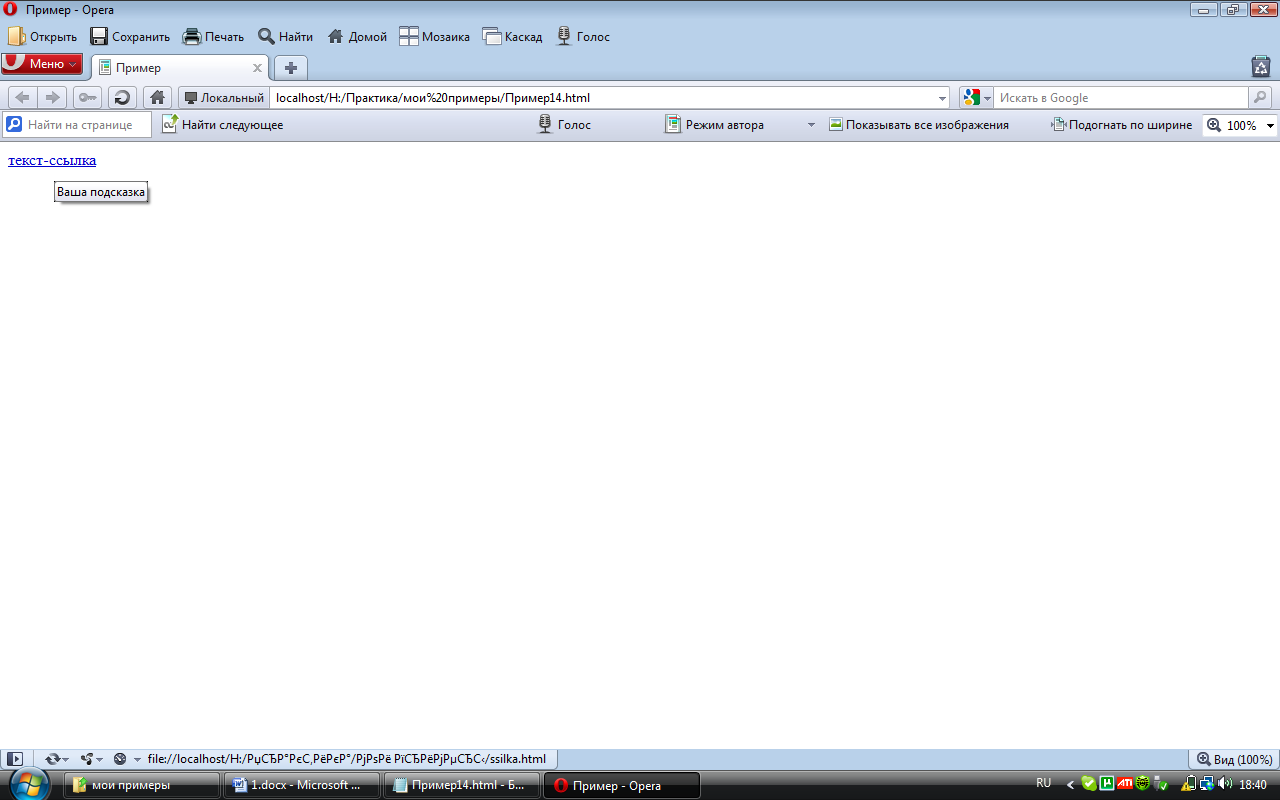
Все вместе это будет выглядеть так:

<a href="mailto:pochta@mail.ru?subject=Hello&Body=text&cc=copy@mail.ru&bcc=hidden\_copy@mail.ru"> pochta@mail.ru </a>

Как сделать, чтобы при наведении мышки на ссылку была подсказка: в старых спецификациях Html (3.2 и ниже) такое не предусмотрено, стало быть, и старые браузеры не поддерживают (IE ниже 5.0 и NN 4.x и ниже). Зато в спецификации Html 4.0 порывы пользователя задать примечание к тексту учитываются. Итак, за это отвечает ***атрибут title***.

Атрибут title (не путать с тегом <TITLE>) предлагает информацию об элементе, для которого он устанавливается. Мы можем установить его, например, для тега <a>:

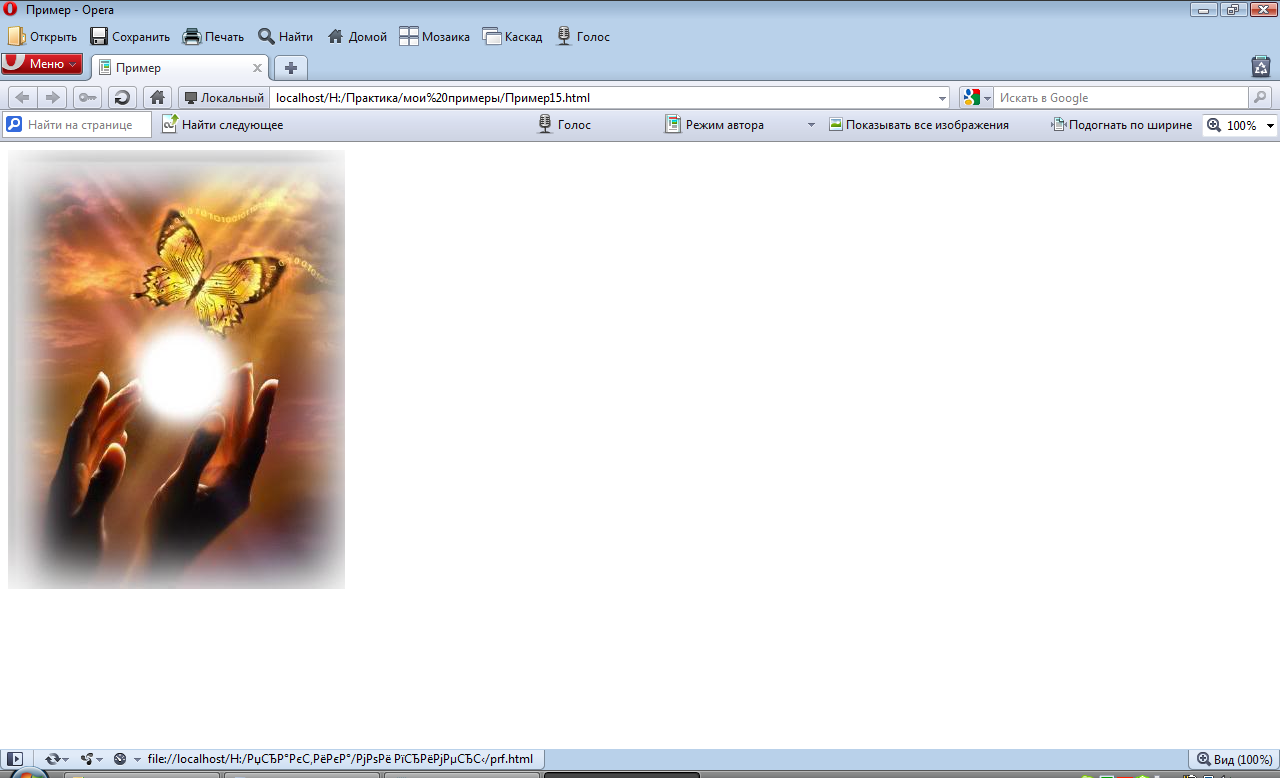
<a href="ssilka.html" title="Ваша подсказка"> текст-ссылка </a>



Наведите курсор мыши на слова "текст-ссылка", и, если ваш браузер поддерживает спецификацию Html 4.0, то вы увидите свою подсказку.

Как уже упоминалось, ссылкой может быть и картинка. Принцип ссылки тот же, что и в случае с текстом, только в тег <a></a> вставляется не текст, а картинка:

<a href="prf.html"><img src="primtocodephoto.gif"></a>



Если вы хотите сделать картинку ссылкой на ваш почтовый ящик, то принцип тот же:

<a href="mailto:pochta@mail.ru"><img src="primtocodephoto.gif"></a>

Ссылка может быть не только на документ с расширением \*.html, но и на многие другие (\*.doc, \*.mp3, \*.jpg, \*.gif, \*.txt, \*.zip, \*.exe и т.д.:), потому же уже известному принципу:

<a href="http://www.melody.ru/music.mp3"> скачать песню </a>

При нажатии на такую ссылку у пользователя появится окно, предлагающее сохранить ему музыкальный файл на его компьютере. Т.е. если файл не является текстовым документом (\*.doc, \*.txt) или, например, \*.html документом, то браузер вашего пользователя обязательно предложит ему скачать такой файл.

Многие уже сталкивались с таким явлением: когда нажимаешь на маленькую картинку, то загружается большая в том же или в новом окне. Делаем ссылкой картинку (допустим small.jpg) и ссылаемся на другую картинку (допустим big.jpg):

<a href="big.jpg"> <img src="small.jpg"> </a>

В этом случае большая картинка откроется в том же окне. Но как же сделать так, чтобы картинка (или любой другой файл-документ) открылась в новом окне? Дело в том, что у тега <a></a> есть ***атрибут target***:

<a href="big.jpg" target="\_blank"> <img src="small.jpg"> </a>

Итак, target="\_blank" - указывает на то, что документ (картинка в нашем случае), на который ведет ссылка, откроется в новом окне браузера.

Иногда возникает такая ситуация: нам нужно сделать ссылку не на другой документ, а внутри того же документа - закладку, в народе называемую ***якорем***. Такая навигация внутри одного и того же документа весьма удобна. Создаваться она может двумя способами.

Способ первый - с помощью атрибута name (имя закладки) тега A:

Пример:

<H2>Григорий Остер, "Вредные советы.<BR>

Книга для непослушных детей и их родителей".</H2>

<A href="#stih1">Ссылка на стих первый</A><BR>

<A href="#stih2">Ссылка на стих второй</A><BR>

<A href="#stih3">Ссылка на стих третий</A><BR><BR>

<PRE>

Недавно ученые открыли, что на свете бывают непослушные дети, которые все делают наоборот. Им дают полезный совет: "Умывайтесь по утрам" - они берут и не умываются. Им говорят: "Здоровайтесь друг с другом" - они тут же начинают не здороваться. Ученые придумали, что таким детям нужно давать не полезные, а вредные советы. Они все сделают наоборот, и получится как раз правильно.

Пример:

</PRE>

<H3><A name="stih1">Стих первый</A></H3>

<PRE>

... тра-ля-ля 1...

</PRE>

<H3><A name="stih2">Стих второй</A></H3>

<PRE>

... тра-ля-ля 2...

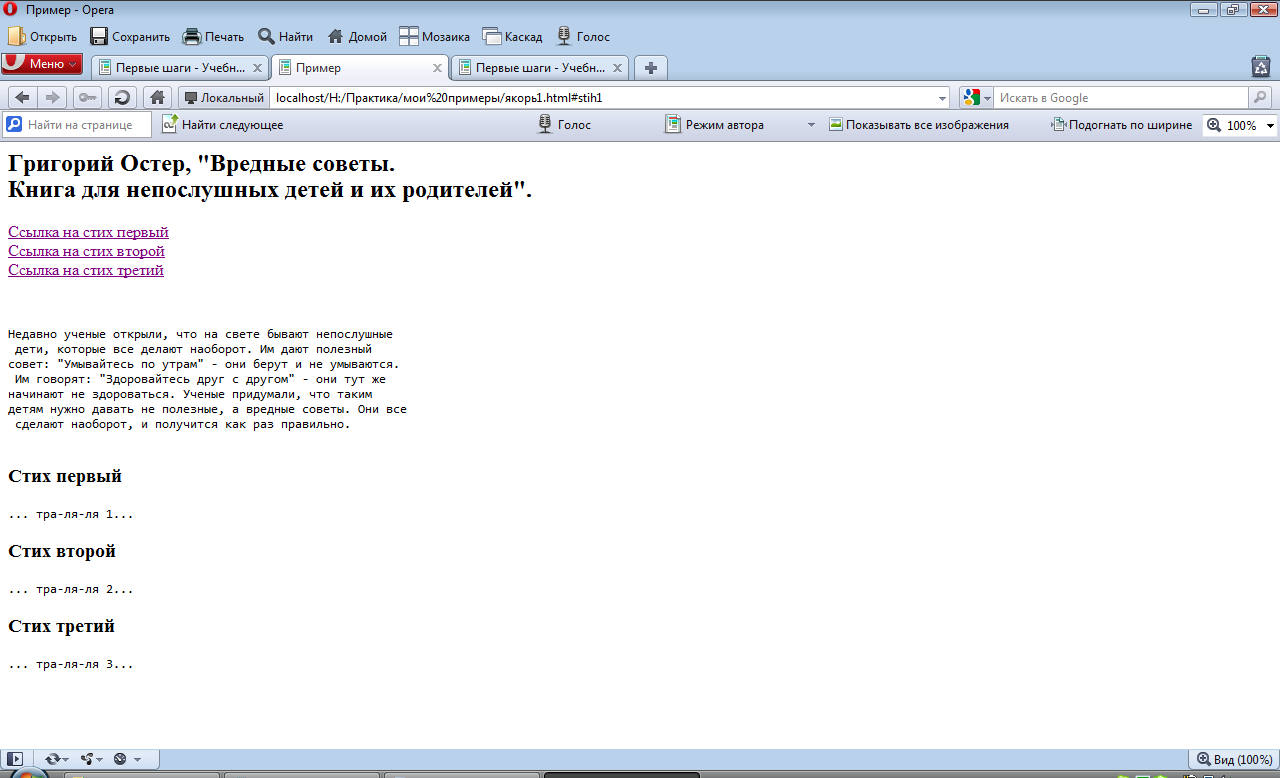
</PRE>

<H3><A name="stih3">Стих третий</A></H3>

<PRE>

... тра-ля-ля 3...

</PRE>



Заголовки (стих первый, стих второй, стих третий) в примере мы сделали закладками, использовав атрибут тега A - name:

<A name="stih3">Стих третий</A>

Затем сделали на них ссылки:

<A href="#stih3">Ссылка на стих третий</A>

Заметьте - <A href="#stih3"> - символ решетки (#) перед именем закладки, на которую мы ссылаемся, обязателен. Вы можете ссылаться на закладку и из других документов, тогда ссылка будет выглядеть следующим образом:

<A href="ancorpri.html#stih3"> Ссылка на стих третий из какого-то другого документа </A>

или

<A href="http://www.mysite.ru/ancorpri.html#stih1"> Ссылка на стих третий из какого-то другого документа </A>

(имя сайта, имя документа + имя закладки)

Способ второй - есть такой атрибут - id, который назначает тегу (элементу) уникальное имя в пределах одного документа (кроме тегов Base, Head, Html, Meta, Script, Style, Title).

Пример:

<H2>Григорий Остер, "Вредные советы.<BR>

Книга для непослушных детей и их родителей".</H2>

<A href="#stih1">Ссылка на стих первый</A><BR>

<A href="#stih2">Ссылка на стих второй</A><BR>

<A href="#stih3">Ссылка на стих третий</A><BR><BR>

<PRE>

Недавно ученые открыли, что на свете бывают непослушные дети, которые все делают наоборот. Им дают полезный совет: "Умывайтесь по утрам" - они берут и не умываются. Им говорят: "Здоровайтесь друг с другом" - они тут же начинают не здороваться. Ученые придумали, что таким детям нужно давать не полезные, а вредные советы. Они все сделают наоборот, и получится как раз правильно.

</PRE>

<H3 id="stih1">Стих первый</H3>

<PRE>

... тра-ля-ля 1...

</PRE>

<H3 id="stih2">Стих второй</H3>

<PRE>

... тра-ля-ля 2...

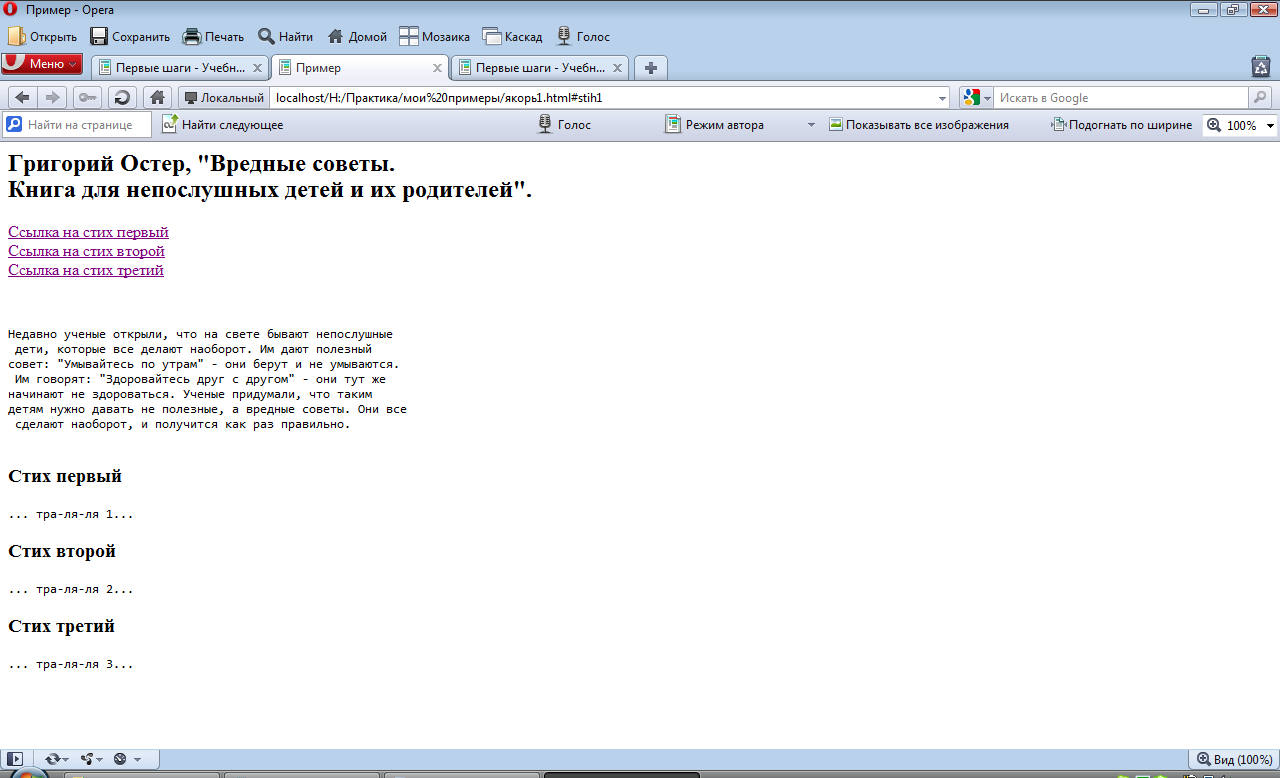
</PRE>

<H3 id="stih3">Стих третий</H3>

<PRE>

... тра-ля-ля 3...

</PRE>



Внешне ничего не изменилось и все действует так же, как и в первом примере, поэтому можно использовать любой из этих двух способов (какой больше нравится).

Замечание!!! никогда не делайте так:

<A href="#stih">Ссылка на ...</A><BR>

...

<A name="STIH">Закладка</A>

Если вы, например, напишите имя закладки большими буквами, а, ссылаясь, укажете это имя маленькими, то браузер может посчитать это за два разных имени (а может и за одно), поэтому во избежание ошибок учитывайте это, и пишите имена в одном регистре.

# Текстовые блоки. Заголовки. Абзацы.

Здесь описаны элементы, разбивающие текст документа на блоки тем или иным способом. Типичными примерами текстовых блоков являются параграфы, абзацы и главы. Для отделения одной части текста от другой также используются разделительные горизонтальные линии и символы возврата каретки.

Элементы:

* H1,H2,...H6 — Используются для создания заголовков текста
* P — Используется для разметки параграфов.
* DIV — Отделяет блок HTML-документа от остальной его части
* ADDRESS — Оформляет текст как почтовый адрес
* BLOCKQUOTE — Оформляет текст как цитату
* BR — Осуществляет перевод строки
* HR — Вставляет в текст горизонтальную разделительную линию.
* PRE — Включает в документ (моноширинным шрифтом) блок предварительно отформатированного текста
* LISTING, PLAINTEXT, XMP — Включают в документ (моноширинным шрифтом) блок предварительно отформатированного текста (устаревшие элементы)

## H1,H2,...H6 – Headers

Используются для создания заголовков текста. Существует шесть уровней заголовков, различающихся величиной шрифта. С их помощью можно разбивать текст на смысловые уровни – разделы и подразделы.

Атрибуты:

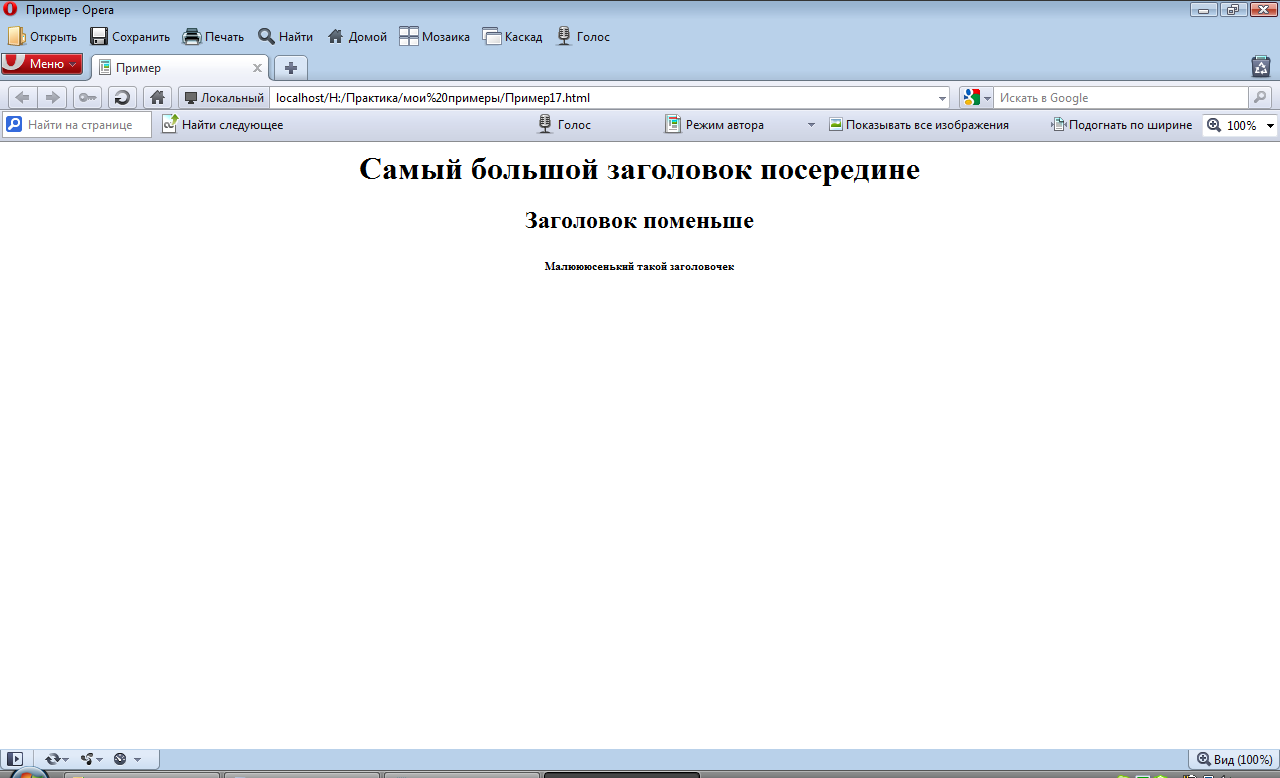
ALIGN – определяет способ выравнивания заголовка по горизонтали. Возможные значения: left, right, center.

Пример:

<H1 ALIGN="center">Самый большой заголовок посередине</H1>

<H2 ALIGN="center">Заголовок поменьше</H2>

<H6 ALIGN="center">Малюююсенький такой заголовочек</H6>

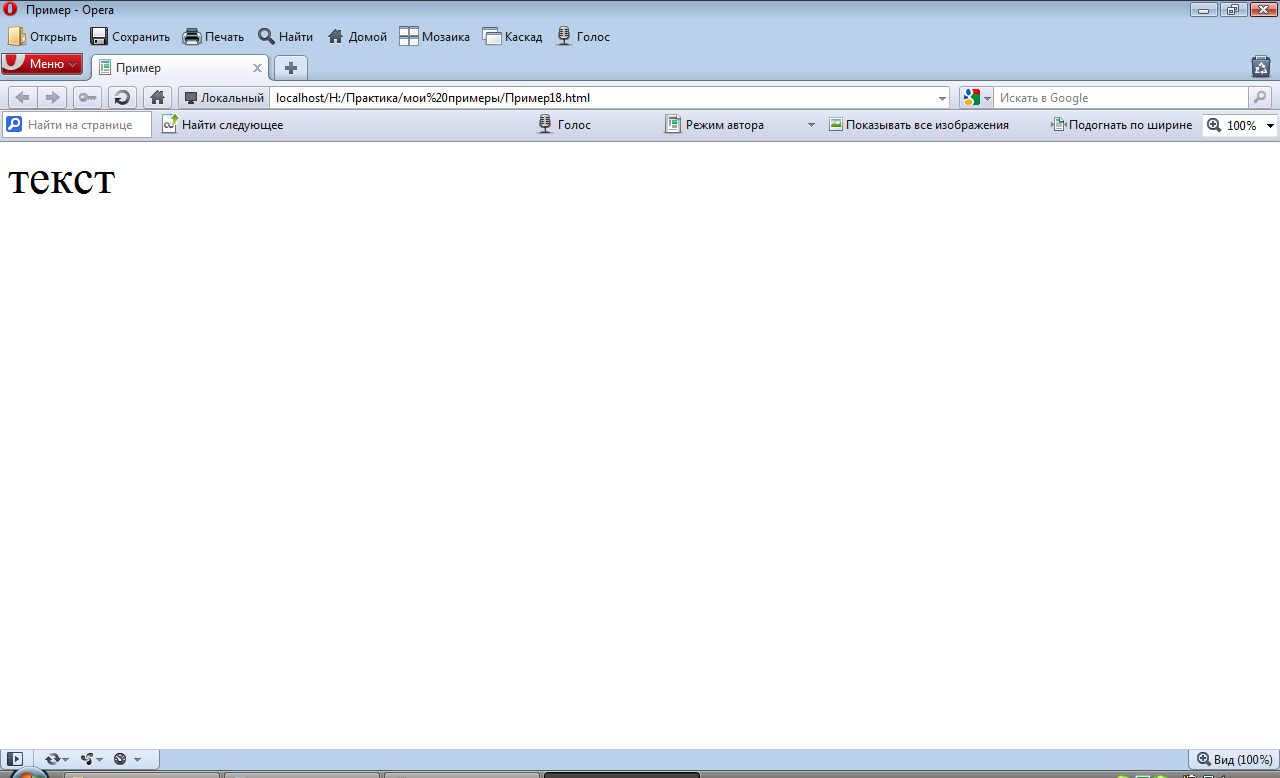


Итак, H1 – самый важный, H6 – наименее важный. Вводя заголовки в текст, вы позволяете вашему посетителю более легко ориентироваться на странице. Заголовками вы определяете структуру внутри текста.

Как мы видим, визуально заголовки отображаются не только более крупным шрифтом, но к тому же и полужирным. Также после закрывающегося </hx> происходит автоматически перенос на другую строку.

Замечание!!! Заголовки предназначены для выделения небольшой части текста (строки, фразы). Но, если вы хотите выделить большой фрагмент текста, то заголовки для этого использовать нельзя. Для этого предназначен атрибут size тега <font></font>, который устанавливает желаемый размер шрифта:

<font size="+4"> текст </font>



## P – Paragraph

Используется для разметки параграфов.

Атрибуты:

ALIGN – определяет способ горизонтального выравнивания параграфа.

Возможные значения: left, center, right, justify. По умолчанию имеет значение left.

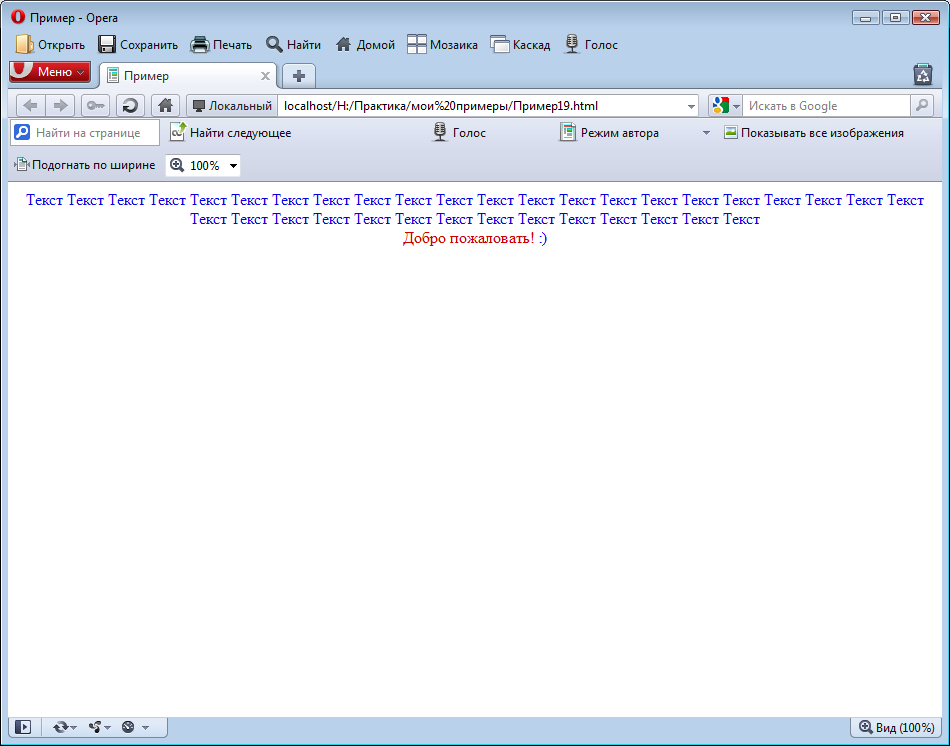
Пример:

<body text=”blue”>

<p align="center">

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст </p>

</body>

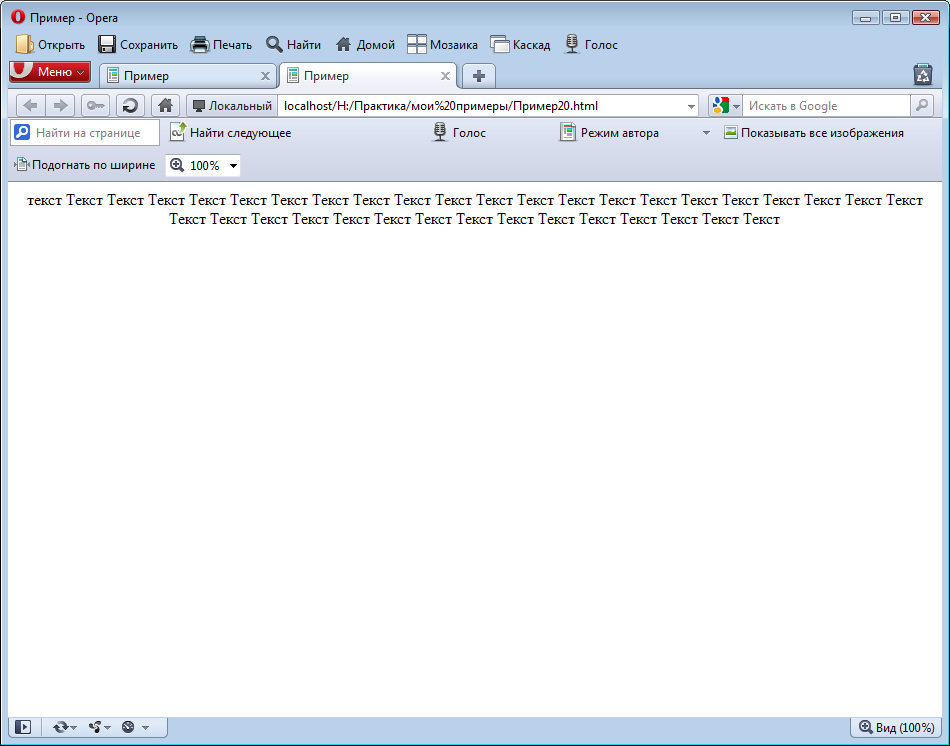


Запомните: никогда нельзя вводить в документ подобную конструкцию:

<p></p>

Пустые элементы <p> без какого-либо содержания (других тегов или текста) могут игнорироваться браузерами. Не забывайте это. Заметьте, что текст в документе, если не задавать параграфы, всегда выравнивается по умолчанию по левому краю. Также запомните, что после закрывающего тега </p> автоматически происходит перенос строки. Но что делать, если вам этот перенос никак не нужен? Есть тег альтернативный <p align="center">:

<center> текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст </center>



Конечно, тег <center> хорош, но достался нам он еще с прошлых версий HTML. Пока что этот тег никто не отменял, и его можно использовать, но это не слишком желательно.

## DIV – Division

Используется для логического выделения блока HTML-документа. Элемент группировки, как и элемент SPAN. В современном сайтостроении используется как удобный контейнер для объектов страницы, которым легко динамически манипулировать – перемещать, включать/выключать, создавать слои, регулировать отступы и т.п.

В браузеронезависимой вёрстке обычно используется для выравнивания блока html-кода в окне браузера. Находящиеся между начальным и конечным тегами текст или HTML-элементы по умолчанию оформляются как отдельный параграф.

Атрибуты:

ALIGN – определяет выравнивание содержимого элемента DIV. Атрибут может принимать значения: left, right, center, justify.

Пример:

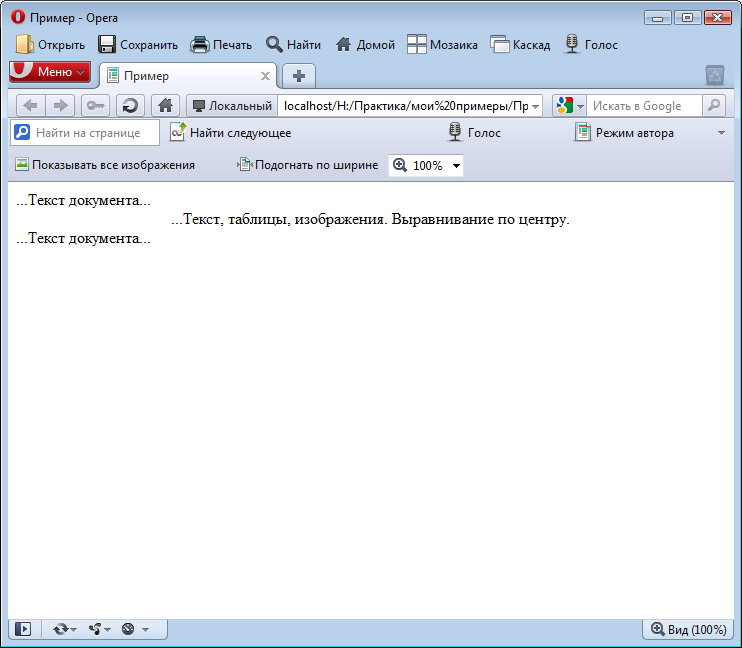
...Текст документа...

<DIV ALIGN="center">

...Текст, таблицы, изображения. Выравнивание по центру.

</DIV>

...Текст документа...



Замечание!!! Параграф не может содержать в себе другие параграфы, и также тег <div></div>. Т.е. следующие конструкции будут не верны, и вводить их в документ нельзя:

<p align="right">

<p>текст</p>

<p>текст</p>

<p>текст</p>

</p>

и

<p align="right">

<div>текст</div>

<p>текст</p>

<div>текст</div>

</p>

Однако <div> может содержать в себе параграфы: с помощью него мы можем сгруппировать их, допустим, по правому краю:

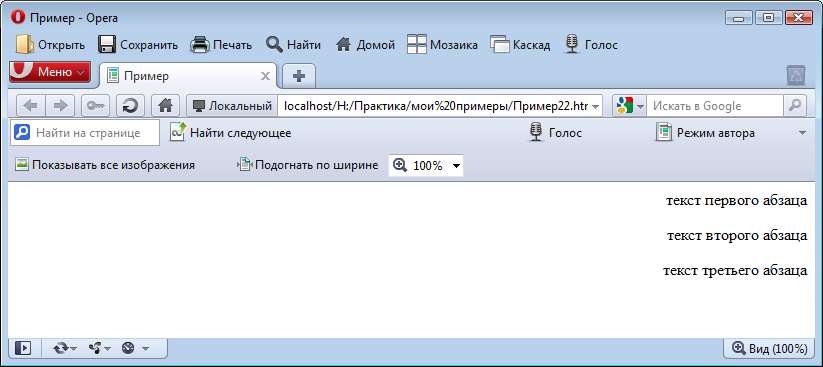
<div align="right">

<p>текст первого абзаца</p>

<p>текст второго абзаца</p>

<p>текст третьего абзаца</p>

</div>



## ADDRESS

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

Пример:

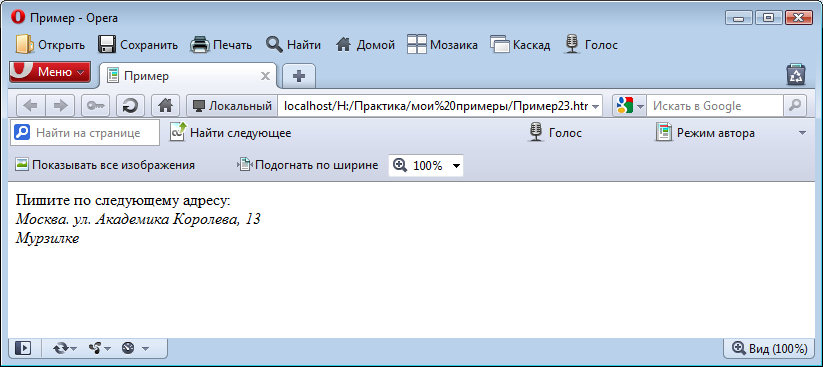
Пишите по следующему адресу:

<ADDRESS>

Москва. ул. Академика Королева, 13 <BR>

Мурзилке

</ADDRESS>



Замечание!!! Некоторые браузеры неверно интерпретирует конечный тег </ADDRESS>, поэтому весь текст, находящийся после адреса, будет отображаться курсивом. Эту ошибку можно легко устранить, поставив после </ADDRESS> любой элемент форматирования текста.

Пример:

<ADDRESS>

Мой адрес не дом и не улица, <BR>

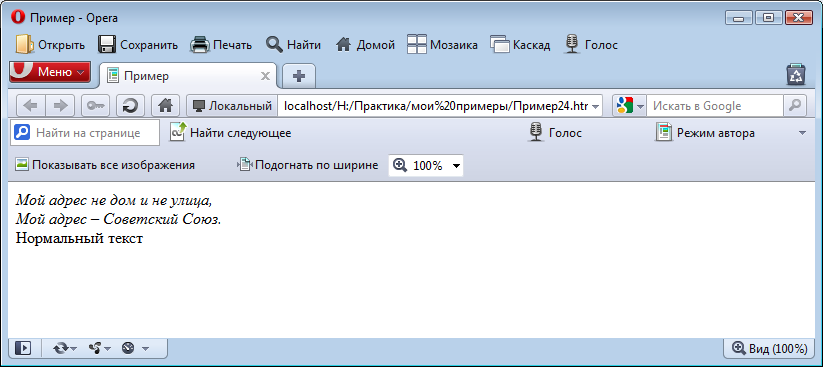
Мой адрес &#150; Советский Союз.

</ADDRESS>

<!-- избавляемся от глюка -->

<I></I>

Нормальный текст



## BLOCKQUOTE

Оформляет находящийся между начальным и конечным тегами текст как цитату. Используется для длинных цитат (в отличие от элемента CITE). Цитируемый текст отображается отдельным абзацем с увеличенным отступом.

Пример:

Редакция журнала "Домосед" выражает благодарность Бухаресту Магарычу Шницелю за замечательное стихотворение:

<BLOCKQUOTE>

Синели красные ромашки,<BR>

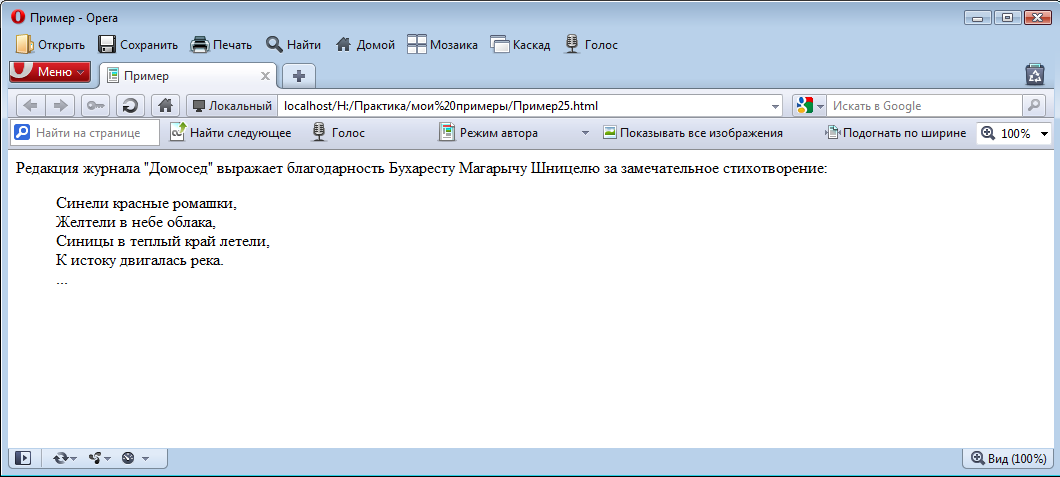
Желтели в небе облака,<BR>

Синицы в теплый край летели,<BR>

К истоку двигалась река.<BR>

...

</BLOCKQUOTE>



## BR – Break

Данный элемент осуществляет перевод строки, то есть практически аналогичен нажатию Enter в текстовом редакторе. После того, как в браузерах появилась возможность обтекания изображения текстом, понадобился дополнительный атрибут CLEAR. Элемент не имеет конечного тега.

Атрибуты:

CLEAR – указывает на необходимость завершения обтекания изображения текстом. Может принимать следующие значения:

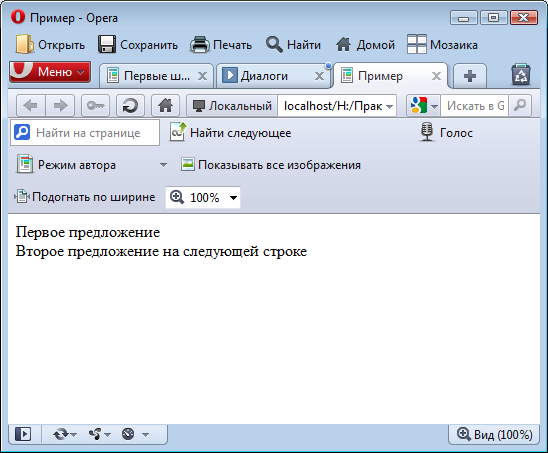
all – завершить обтекание изображения текстом.

left – завершить обтекание текстом изображения, выровненного по левому краю.

right – завершить обтекание текстом изображения, выровненного по правому краю.

Пример:

Первое предложение<BR> Второе предложение на следующей строке



## HR – Horizontal Rule

Вставляет в текст горизонтальную разделительную линию.

Атрибуты:

WIDTH – определяет длину линии в пикселях или процентах от ширины окна браузера.

SIZE – определяет толщину линии в пикселях.

ALIGN – определяет выравнивание горизонтальной линии. Атрибут может принимать следующие значения:

left – выравнивание по левому краю документа.

right – выравнивание по правому краю документа.

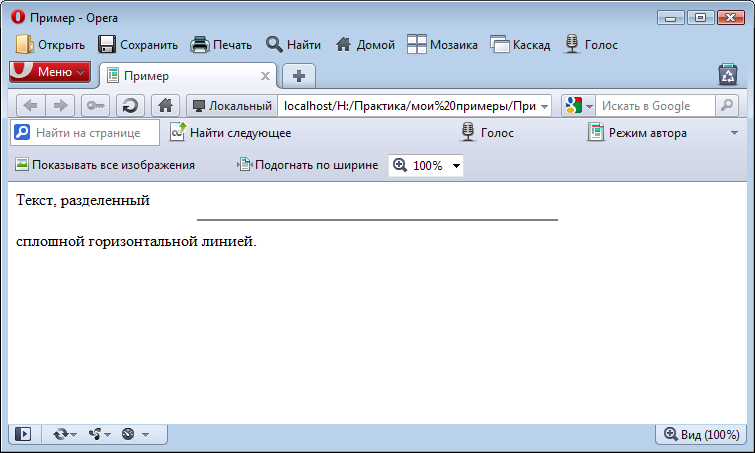
center – выравнивание по центру документа (используется по умолчанию).

NOSHADE – определяет способ закраски линии как сплошной. Атрибут является флагом и не требует указания значения. Без данного атрибута линия отображается объемной.

COLOR – определяет цвет линии. Задается либо RGB-значением в шестнадцатиричной системе, либо одним из 16 базовых цветов. Атрибут работает только в Internet Explorer.

Пример:

Текст, разделенный <HR NOSHADE WIDTH="50%"> сплошной горизонтальной линией.



## PRE – 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:

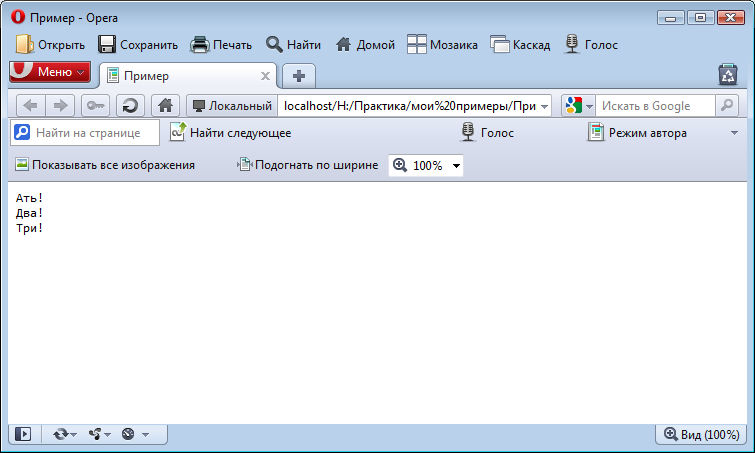
<PRE>

Ать!

Два!

Три!

</PRE>



Замечание!!! Желательно избегать использования символа горизонтальной табуляции внутри PRE, т.к. он может быть неадекватно интерпретирован некоторыми браузерами. Вместо символа табуляции рекомендуется использовать число пробелов, кратное четырем.

## LISTING, PLAINTEXT, XMP

Данные элементы предназначены для воспроизведения предварительно отформатированного текста. Они использовались до появления элемента PRE и на данный момент являются устаревшими.

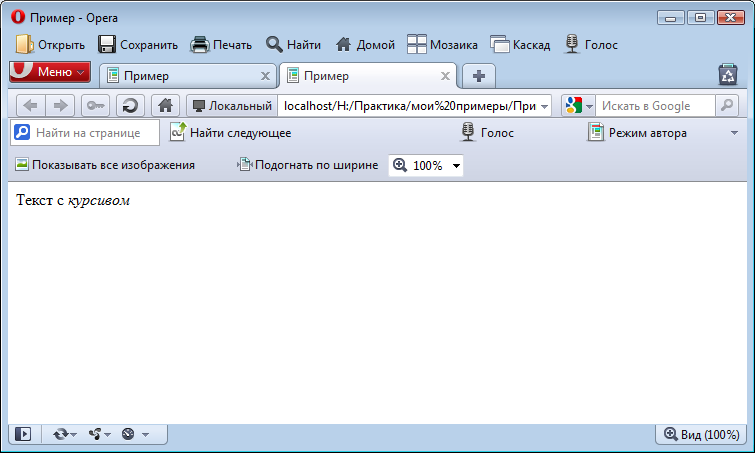
Эти элементы не вошли в спецификацию HTML 4.0, однако браузеры еще могут распознавать их в целях обратной совместимости версий.

# Форматирование текста

Элементы форматирования текста:

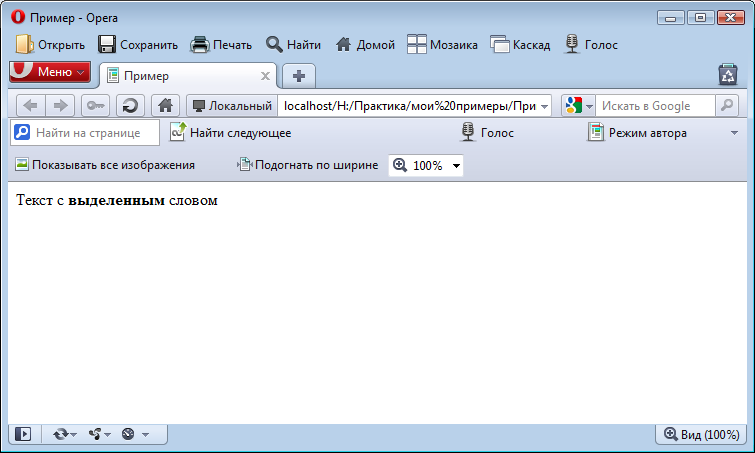
* BASEFONT — Определяет основной шрифт, которым должен отображаться текст документа
* FONT — Позволяет изменять цвет, размер и тип шрифта текста
* I — Выделяет текст курсивом. Пример:

Текст с <I>курсивом</I>



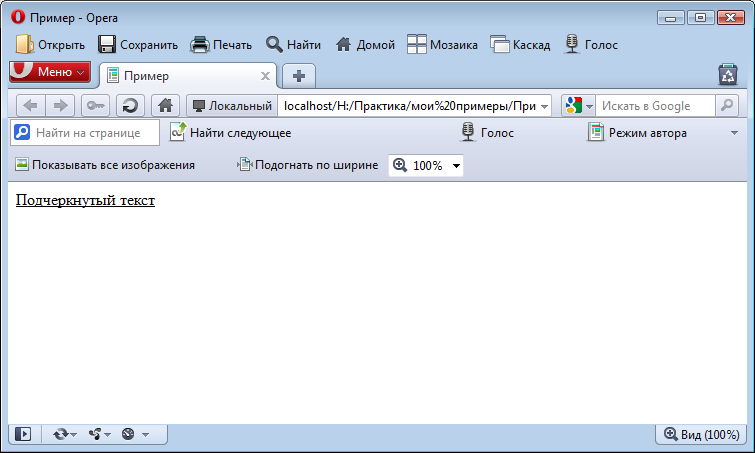
* EM — Используется для смыслового выделения текста (курсивом)
* B — Выделяет текст жирным шрифтом. Пример:

Текст с <B>выделенным</B> словом

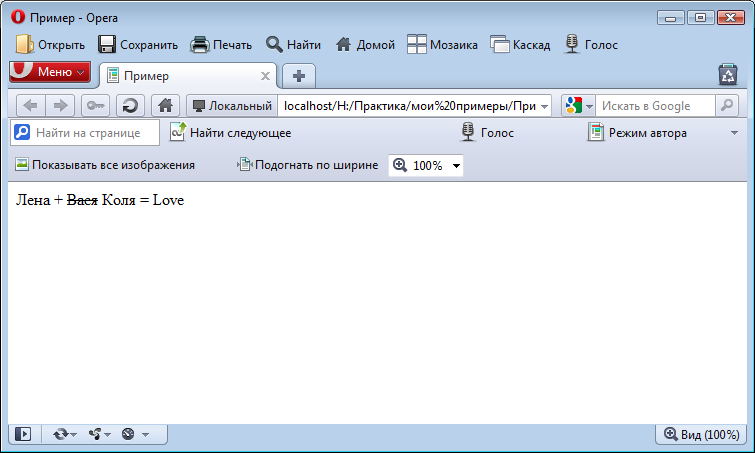


* STRONG — Усиленное выделение текста (жирным)
* U — Выделяет текст подчеркнутым. Пример:

<U> Подчеркнутый текст </U>



* S, STRIKE — Выделяет текст перечеркнутым. Пример:

Лена + <STRIKE>Вася</STRIKE> Коля = Love

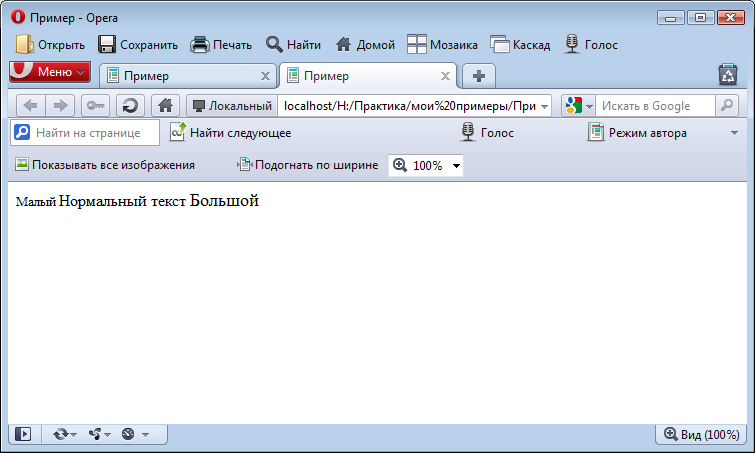
* BIG — Отображает текст увеличенным шрифтом (относительно текущего)
* SMALL — Отображает текст уменьшенным шрифтом (относительно текущего)

Пример:

<small> Малый </small>

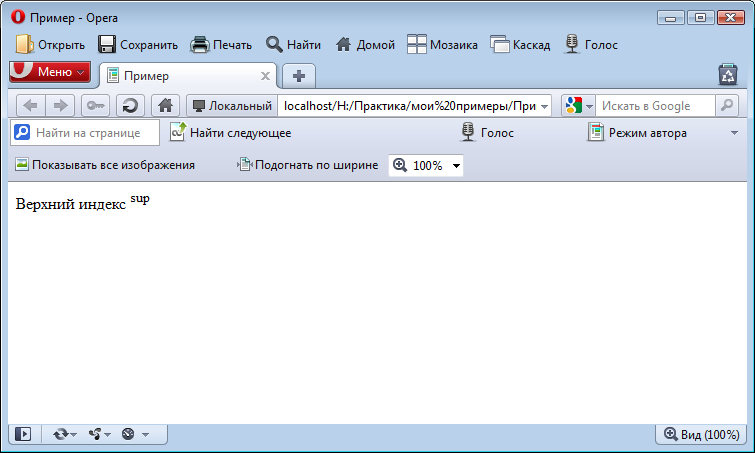
Нормальный текст

<big> Большой </big>



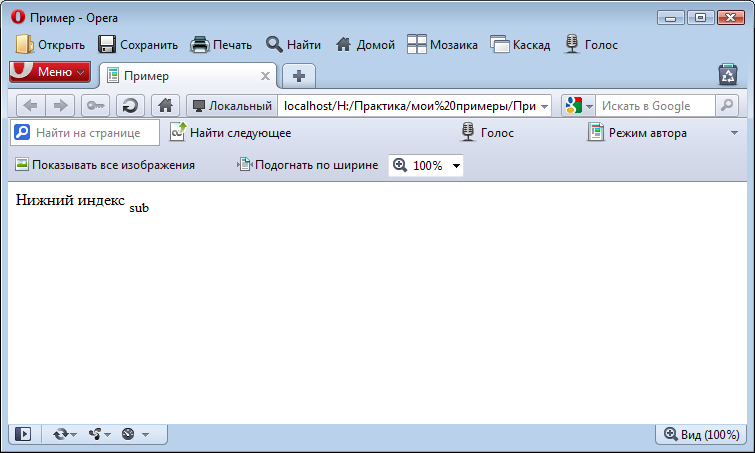
* SUP — Отображает текст со сдвигом вверх (верхний индекс). Пример:

Верхний индекс <sup> sup </sup>



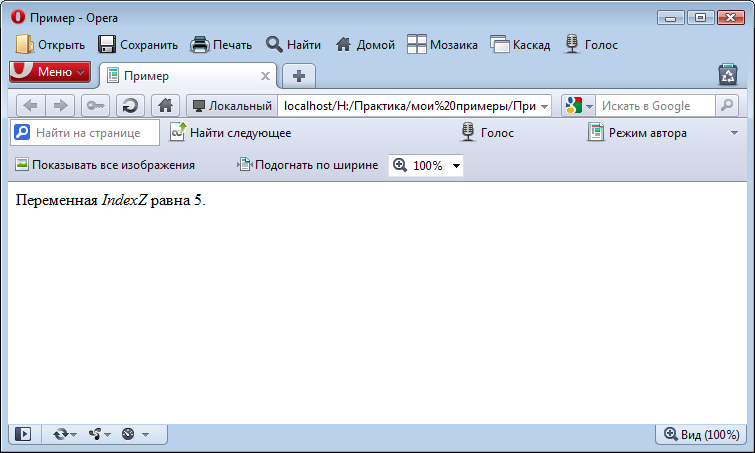
* SUB — Отображает текст со сдвигом вниз (нижний индекс). Пример:

Нижний индекс <sub> sub </sub>



* CODE, SAMP — Оформляют текст как формулу или программный код
* TT — Отображает текст моноширинным шрифтом
* KBD — Выделяет текст, который предлагается набрать на клавиатуре
* VAR — Используется для обозначения в тексте переменных. Как правило, отображается курсивом. Пример:

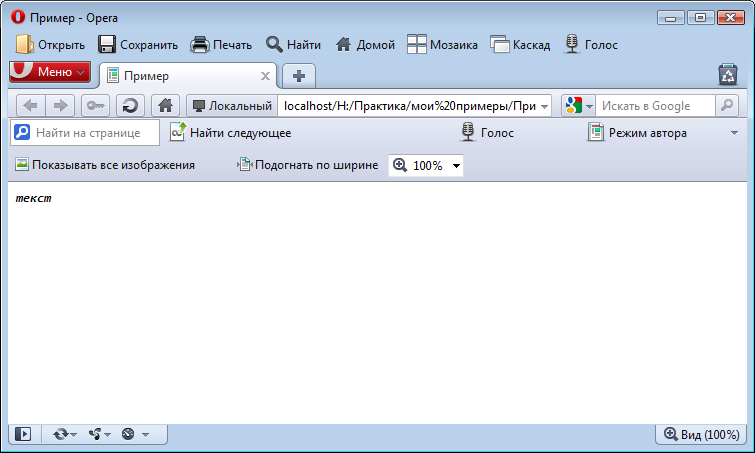
Переменная <VAR>*IndexZ*</VAR> равна 5.



* CITE — Оформляет текст как цитату или ссылку на источник

К одному фрагменту текста может применяться сразу несколько тегов:

<tt><b><i> текст </i></b></tt>



В нашем примере текст моноширинный, полужирный, и выделен курсивом.

## Base Font

Не имеет конечного тега. Определяет основной шрифт, которым должен отображаться текст документа.

Впоследствии вы можете легко изменить шрифт в любой части документа, используя элемент FONT. Действие элемента BASEFONT не распространяется на текст, заключенный в ячейки таблиц.

Атрибуты:

SIZE – обязательный атрибут. Определяет базовый размер шрифта. Возможные значения: целые числа от 1 до 7 включительно.

FACE – определяет используемый шрифт (гарнитуру).

Пример:

<BODY>

<BASEFONT SIZE="3">

... Текст документа шрифтом 3 размера ...

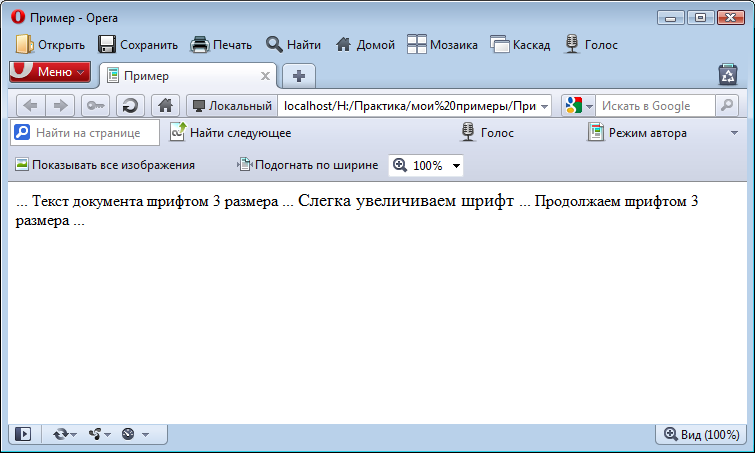
<FONT SIZE="+1">

Слегка увеличиваем шрифт

</FONT>

... Продолжаем шрифтом 3 размера ...

</BODY>



Замечание!!! Атрибут FACE игнорируется большинством браузеров. Если вы хотите явно указать тип шрифта для всего документа, используйте элемент FONT: — Пример:

<BODY>

<FONT FACE="Arial">

...

Основной текст документа шрифтом Arial

...

<FONT FACE="Times New Roman" SIZE="5">

Вставка текста увеличенным шрифтом Times New Roman

</FONT>

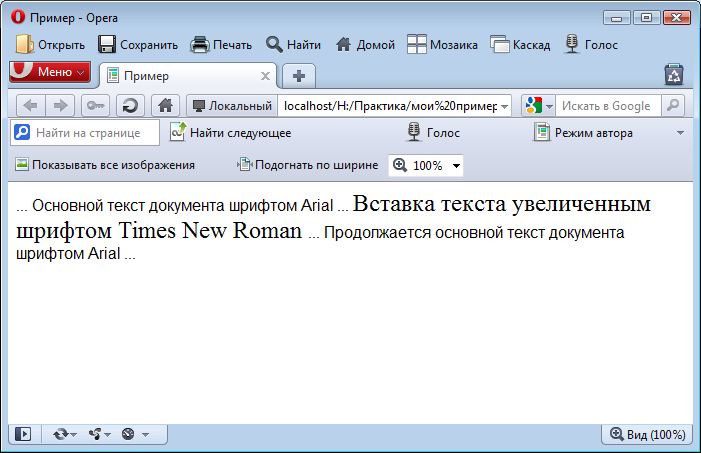
...

Продолжается основной текст документа шрифтом Arial

...

</FONT>

</BODY>



## Font

Позволяет изменять цвет, размер и тип шрифта текста, находящегося между начальным и конечным тегами. Вне тегов <FONT> и </FONT> используется шрифт, указанный в элементе BASEFONT.

Атрибуты:

SIZE – определяет размер шрифта. Возможные значения:

* целое число от 1 до 7;
* относительный размер с указанием знака, вычисляется путем сложения с базовым размером, определенным с помощью атрибута SIZE элемента BASEFONT.

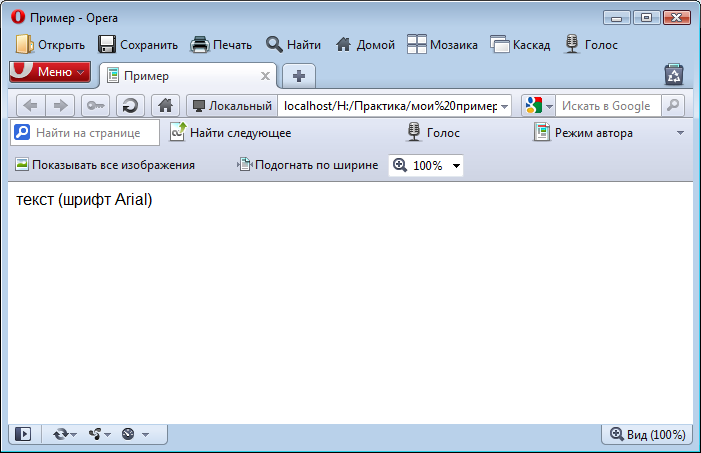
COLOR – определяет цвет текста. Задается либо RGB-значением в шестнадцатиричной системе, либо одним из 16 базовых цветов. Обратите внимание, что значению цвета обязательно должен предшествовать значок «решетка» - #, если он задается RGB-значением.

FACE – определяет используемый шрифт. Стандартными шрифтами, которые находятся на компьютере каждого пользователя, являются:

* Times;
* Times New Roman;
* Arial;
* Helvetica;
* Courier;
* Verdana;
* Tahoma;
* Cosmic Sans;
* Garamond

Любой из этих шрифтов можно использовать безбоязненно. В атрибуте face можно указать сразу несколько типов шрифтов:

<font face="arial, verdana, courier"> текст (шрифт Arial) </font>

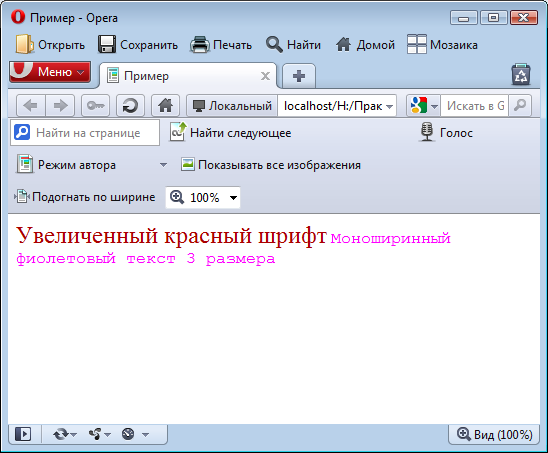


В этом случае если не окажется на компьютере шрифта Arial, то текст будет отображен шрифтом Verdana. Если и Verdana нет на компьютере, то текст будет отображен шрифтом Courier. Т.е. в атрибуте face мы можем задать список разделенных запятыми названий шрифтов, которые браузер вашего посетителя должен попытаться найти у него на компьютере и отобразить в порядке приоритета.

Пример:

<FONT SIZE="+2" COLOR="#AA0000">Увеличенный красный шрифт</FONT>

<FONT SIZE="3" FACE="Courier New" COLOR="Magenta">Моноширинный фиолетовый текст 3 размера</FONT>

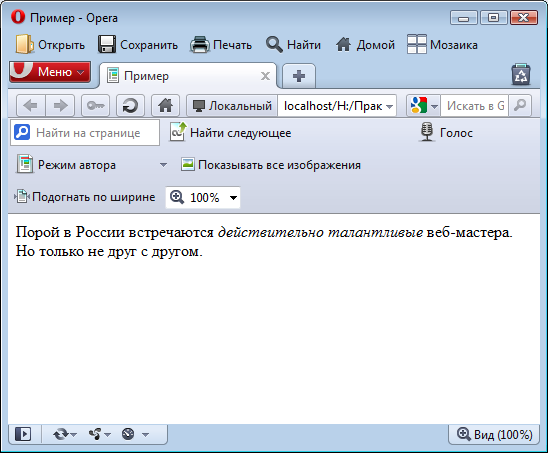


## EM – Emphasis

Логическое ударение. Используется для смыслового выделения текста, стоящего между начальным и конечным тегами. Результат обычно отображается курсивом. То есть элемент EM по действию практически аналогичен элементу I.

Пример:

Порой в России встречаются <EM>действительно талантливые</EM> веб-мастера. Но только не друг с другом.

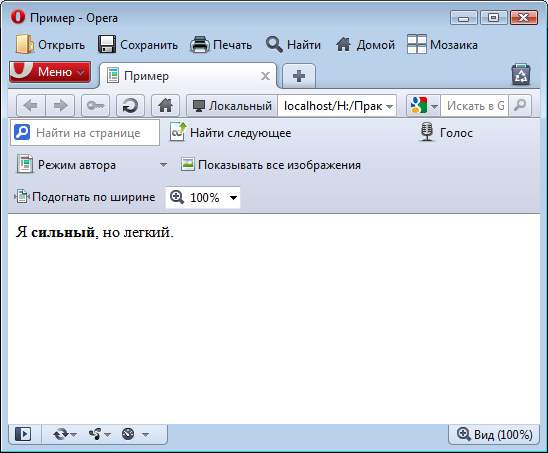


## STRONG – Stronger emphasis

Усиленное выделение. Текст, заключенный между начальным и конечным тегами, будет выделен жирным шрифтом. То есть элемент STRONG практически аналогичен по действию элементу B.

Пример:

Я <STRONG>сильный</STRONG>, но легкий.

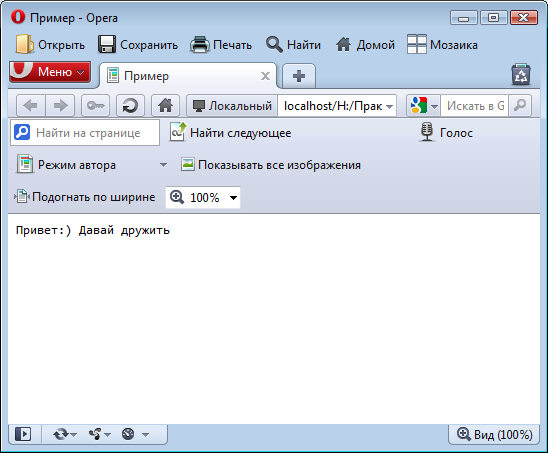


## TT – Tele Type

Данный элемент отображает помещенный между начальным и конечным тегами текст моноширинным шрифтом. Это шрифт с символами одинаковой фиксированной ширины, как шрифт у пишущей машинки.

Пример:

<TT>Привет:) Давай дружить</TT>



Также моноширинным шрифтом отображается текст, заключенный в тег ***<pre></pre>:***

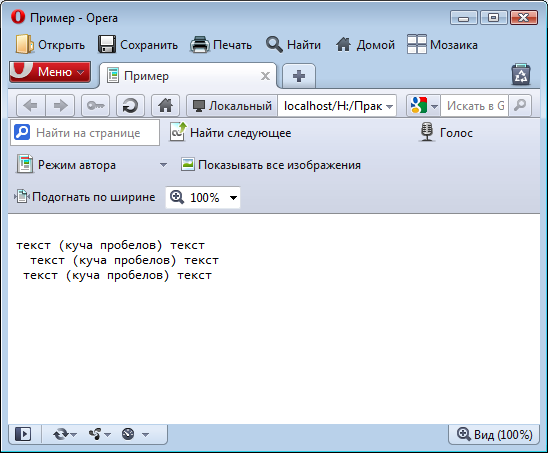
<pre>

текст (куча пробелов) текст

текст (куча пробелов) текст

текст (куча пробелов) текст

</pre>



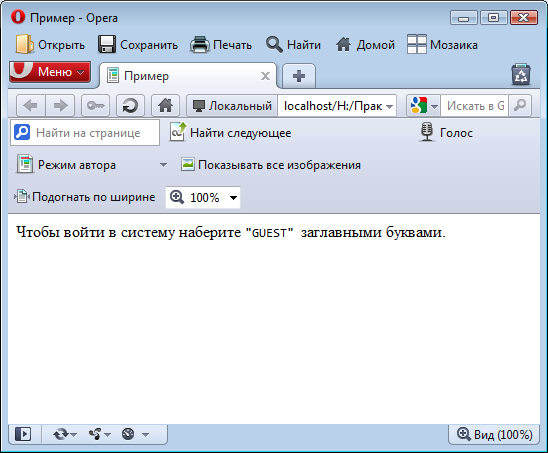
У тега <pre> есть одна замечательная особенность: текст, заключенный в него, выводится с точностью до пробела так, как вы его набили в блокноте. Этот тег вам может быть полезен, допустим, для форматирования стихотворений.

## KBD – Keyboard

Предназначен для отображения текста, который пользователь должен набрать на клавиатуре (например, при заполнении формы, введении пароля и т.п.). Как правило, текст, заключенный между начальным и конечным тегами, выделяется моноширинным шрифтом.

Пример:

Чтобы войти в систему наберите <KBD> "GUEST" </KBD> заглавными буквами.



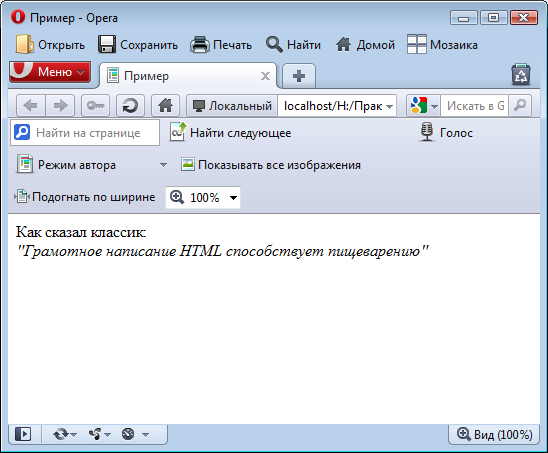
## CITE – Citation

Оформляет находящийся между начальным и конечным тегами текст как цитату или ссылку на источник. Обычно используется для коротких цитат (в отличие от элемента BLOCKQUOTE). Цитируемый текст отображается курсивом.

Пример:

Как сказал классик:<BR>

<CITE>"Грамотное написание HTML способствует пищеварению"</CITE>



## Marquee - бегущая строка (текста)

Текст, который мы помещаем в теге <marquee></marquee>, становится бегущей строкой.

Атрибуты:

bgcolor - цвет фона бегущей строки.

height - высота строки.

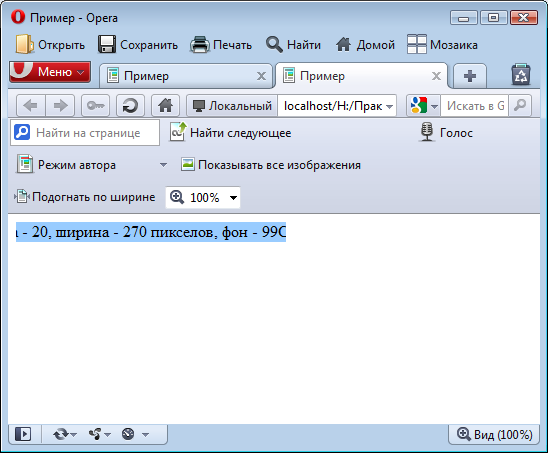
width - ширина строки.

Пример:

<marquee height="10" width="270" bgcolor="#99CCFF">

Бегущая строка, высота - 10, ширина - 270 пикселов, фон - 99CCFF

</marquee>



vspace - задает расстояние между текстом и бегущей строкой по вертикали. Расстояние задается в пикселях.

hspace - задает расстояние между текстом и бегущей строкой по горизонтали. Расстояние задается в пикселях.

Align - отвечает за выравнивание бегущей строки на экране.

loop – задает, сколько раз прокрутится строка. Здесь она прокручивается всего 2 раза:

<marquee height="10" width="270" loop="2">

Бегущая строка

</marquee>

Direction - задает направление движения бегущей строки - direction="left" (right, up, down) - движение влево (вправо, вверх, вниз).

Behavior - поведение строки - behavior="scroll" (slide, alternate). Scroll - обычная прокрутка (можно не прописывать, оно так и есть по умолчанию). Slide - прокрутка с остановкой, строка пробегает до края и останавливается. Если одновременно с behavor="slide" употребить параметр loop, то строка прокрутится установленное количество раз и остановится у края. Alternate - строка будет двигаться от края к краю.

Scrollamount - скорость движения строки, scrollamount="1". Может принимать значения от 1 до 10. 1 - самое медленное движение, 10 - самое быстрое.

# Списки

Списки в HTML бывают двух видов: упорядоченные (пронумерованные) и неупорядоченные (непронумерованные). Отличаются они лишь способом оформления. Перед пунктами неупорядоченных списков обычно ставятся символы-буллеты (bullets), например, точки, ромбики и т.п., в то время как пунктам упорядоченных списков предшествуют их номера.

Элементы:

* UL — Создает неупорядоченный список
* OL — Создает упорядоченный список
* LI — Создает пункт меню внутри элементов OL или UL.
* MENU, DIR — Создает неупорядоченный список, подобный UL. Эти элементы не вошли в спецификацию HTML 4.0, однако браузеры могут распознавать их в целях обратной совместимости. Элемент DIR был разработан для создания списка из нескольких столбцов. Элемент MENU был разработан для создания списка в один столбец. Оба элемента имеют ту же структуру, что и UL, но другое представление. Авторам же рекомендуется использовать вместо MENU и DIR элемент UL.
* DL — Открывает и закрывает список определений. Определения задаются с помощью элементов DT и DD. Атрибуты:

COMPACT – включает режим "компактного" отображения списка. Удобно использовать, если список определений очень велик. Данный атрибут является флагом и не требует присвоения значения.

Пример:

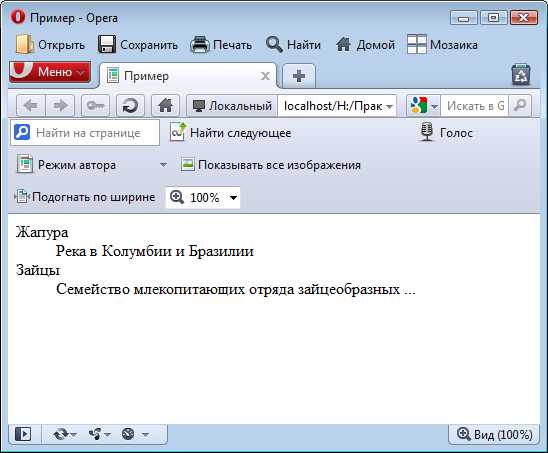
<DL COMPACT>

<DT>Жапура<DD>Река в Колумбии и Бразилии

<DT>Зайцы<DD>Семейство млекопитающих отряда зайцеобразных

...

</DL>



Другая сфера применения DL это, например, создание диалогов, где каждый DT именует говорящего, а каждый DD содержит его, её слова.

* DT — Создает термин в списке определений внутри элемента DL
* DD — Создает определение термина внутри элемента DL

## LI – List Item

Создает пункт в списке. Располагается внутри элементов OL или UL.

Атрибуты:

VALUE – изменяет порядок нумерации элементов списка. Используется, только если элемент LI находится внутри элемента OL. В качестве значения указывается порядковый номер элемента.

Пример:

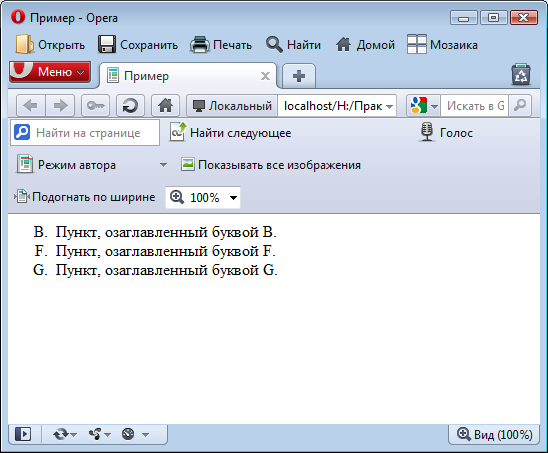
<OL TYPE="A" START="2">

<LI> Пункт, озаглавленный буквой B. </LI>

<LI VALUE="6"> Пункт, озаглавленный буквой F. </LI>

<LI> Пункт, озаглавленный буквой G. </LI>

</OL>



Type — отвечает за стиль элемента списка, визуально это выглядит так:

Disc – жирный круг ●

Circle – полый круг ○

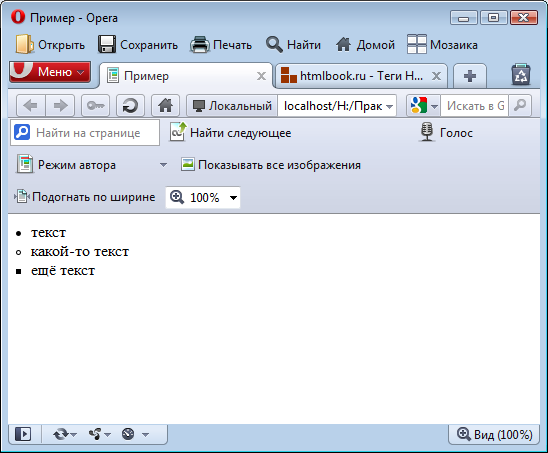
Square – жирный квадрат ■

Пример:

<li type="disc"> текст

<li type="circle"> какой-то текст

<li type="square"> ещё текст



Атрибут type может вводиться не только для каждого элемента списка <li>, но и для всего списка в целом:

<ul type="square">

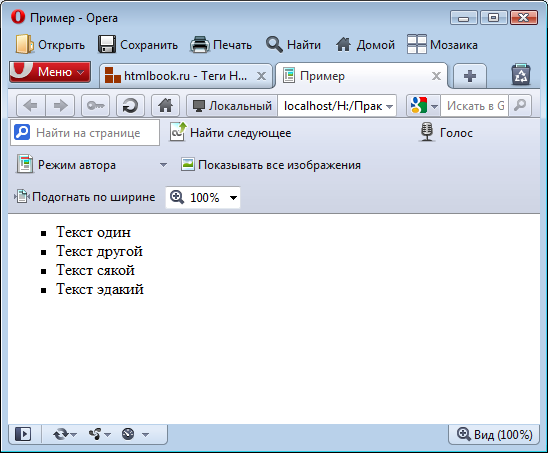
<li>Текст один

<li>Текст другой

<li>Текст сякой

<li>Текст эдакий

</ul>



## Неупорядоченный список

UL (Unsorted List) создает неупорядоченный список. Между начальным и конечным тегами должны присутствовать один или несколько элементов LI, обозначающих отдельные пункты списка.

Пример:

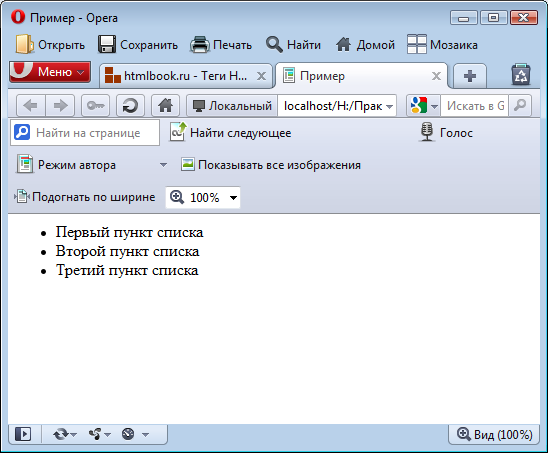
<UL>

<LI> Первый пункт списка </LI>

<LI> Второй пункт списка </LI>

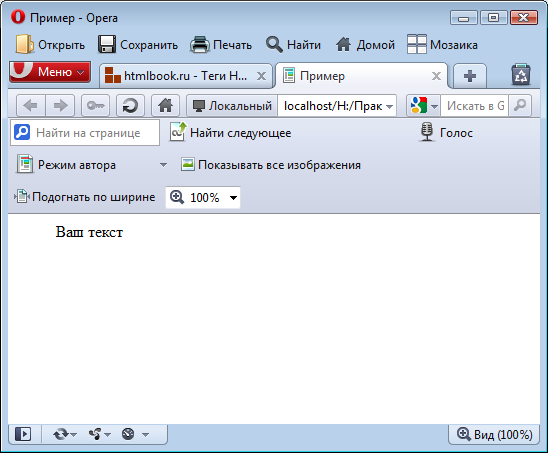
<LI> Третий пункт списка </LI>

</UL>



Очень часто тег <ul></ul> используется кодерами немного нестандартно, для создания отступа (табуляции). Если вы пользуетесь текстовыми редакторами (допустим Microsoft Word), то вы знаете, что там есть табуляция (это когда вы жмете на кнопку Tab и весь текст сдвигается, допустим, вправо), тоже самое можно сделать средствами html:

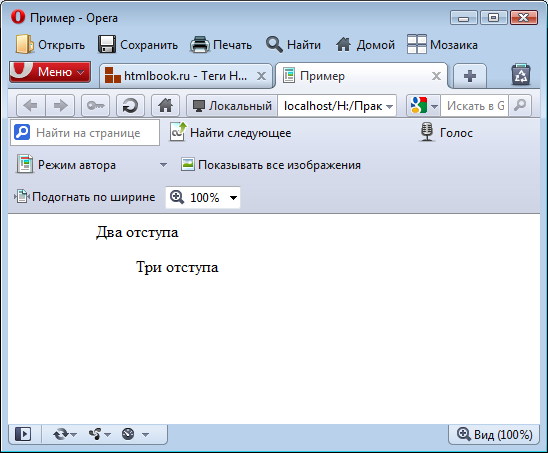
<ul>Ваш текст</ul>



В этом случае мы не использовали тег <li>, и не ввели элементы списка. Такое применение тега <ul></ul> не стандартно, а значит нежелательно, т.к. не предусмотрено спецификацией, однако, если вам это нужно, то в данном случае, не страшно нестандартно воспользоваться тегом <ul>, ведь визуально это отобразиться всеми браузерами, а также избавит вас от необходимости вводить лишние таблицы в ваш документ, чтобы достичь того же эффекта стандартными средствами. Чтобы отступ был больше, можно вложить тег <ul> в самого себя:

<ul><ul>Два отступа</ul></ul>

<ul><ul><ul>Три отступа</ul></ul></ul>



## Упорядоченный список

OL (Ordered List) создает упорядоченный список. Между начальным и конечным тегами должны присутствовать один или несколько элементов LI, обозначающих отдельные пункты списка.

Атрибуты:

START – определяет первое число, с которого начинается нумерация пунктов (только целые числа)

TYPE – определяет стиль нумерации пунктов. Может иметь значения:

"A" – заглавные буквы A, B, C ...

"a" – строчные буквы a, b, c ...

"I" – большие римские числа I, II, III ...

"i" – маленькие римские числа i, ii, iii ...

"1" – арабские числа 1, 2, 3 ...

По умолчанию <UL TYPE="1">.

Пример:

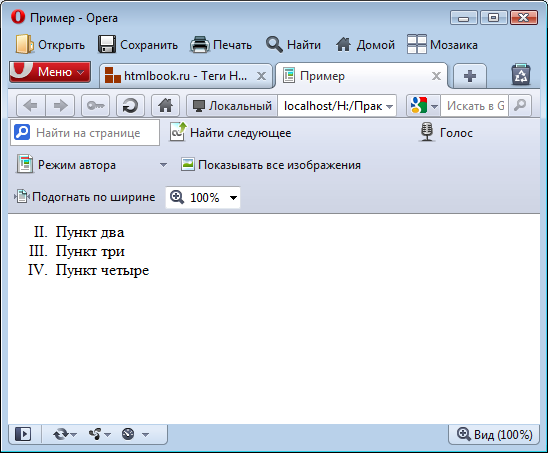
<OL TYPE="I" START="2">

<LI> Пункт два </LI>

<LI> Пункт три </LI>

<LI> Пункт четыре </LI>

</OL>



Различные типы списков можно употреблять одновременно друг с другом.