

Дмитрий Кудрец
Основы CSS

Составитель Дмитрий Кудрец

Основы CSS



ISBN 9785449621740

Аннотация

В книге рассказывается об использовании таблиц стилей, позиционирования и фильтров в HTML-документах. Рекомендована для учащихся школ, гимназий, а также для всех желающих расширить свои знания о языке HTML.

Основы CSS

Составитель Дмитрий Кудрец

ISBN 978-5-4496-2174-0

Создано в интеллектуальной издательской системе Ridero

Каскадные таблицы стилей

При оформлении HTML-документов приходится по несколько раз прописывать однотипные теги. Использование каскадных таблиц стилей (CSS) позволяет значительно упростить эту работу. Поддержка стандарта каскадных таблиц стилей и их программируемых элементов появилась, начиная с версии 4.0 и Netscape, и Internet Explorer.

Существует три основных варианта использования таблиц стилей.

Вариант первый

Описание стиля располагается непосредственно внутри тега элемента, который вы описываете. Это делается с помощью параметра **STYLE**. Этот метод нежелателен, и понятно почему: он приводит к потере одного из основных преимуществ CSS – возможности отделения информации от описания оформления информации. Впрочем, если необходимо описать лишь один элемент, этот вариант расположения описания стилей также вполне применим.

<H1 STYLE =«font-family: Arial. Helvetica. San Serif»> Мой сайт </H1 >

<P STYLE =«font-size: 12 pt»> Добро пожаловать на мой сайт. </P >

 Щелкните Щелкните здесь!
здесь!

Если нужно задать сразу несколько свойств, их разделяют точкой с запятой:

<P STYLE="ALIGN: RIGHT; FONT-STYLE: ITALIC; FONT-WEIGHT: BOLD; BACKGROUND: YELLOW">Жирный, наклонный шрифт на желтом фоне.</P>

Жирный, наклонный шрифт на желтом фоне.

Атрибут **STYLE** применяется лишь тогда, когда нужно изменить свойства одного-двух конкретных элементов, расположенных в конкретном месте страницы.

Вариант второй

Если необходимо многократное применение заданного стиля, его описание таблиц стилей размещается в разделе **<HEAD>** в теге **<STYLE TYPE=«text/css»>** ... **</STYLE>**. Параметр **TYPE=«text/css»** является обязательным и служит для указания браузеру использовать CSS.

<STYLE TYPE=«text/css»>

P FONT-STYLE: SMALL-CAPS

</STYLE>

Выбранный элемент (в данном случае <P>) в терминах CSS называется *селектором*, а все, что располагается между фигурными скобками, – *определением*. Все вместе называется *правилом*.

Селекторы это те буквы, из которых состоят определения элементов HTML: **P**, **H1**, **UL** и т. д. При создании правила вы сопоставляете выбранному элементу определение стиля. Например, **UL List-Style: Disc**

Элемент **<STYLE>** может содержать и несколько правил. Каждое из них оканчивается закрывающей скобкой, поэтому даже на одной строке вы можете разместить более одного правила.

```
<STYLE TYPE="TEXT/CSS">
P {FONT-STYLE: SMALL-CAPS}
H1 {COLOR: BLUE}
</STYLE>
<BODY>
<H1>Заголовок</H1>
<P>Текст</P>
<UL>
<LI>Элемент списка
<LI>Элемент списка
</UL>
</BODY>
```

Заголовок

Текст

- Элемент списка
- Элемент списка

Далее, каждое определение может включать в себя несколько свойств. Они отделяются друг от друга точкой с запятой, но для улучшения читабельности кода лучше выносить каждое из них в отдельную строку:

```
<STYLE TYPE="TEXT/CSS">
P {FONT-STYLE: SMALL-CAPS;
BACKGROUND: RED;
PADDING-LEFT: 18PX}
</STYLE>
<BODY>
<P>Текст</P>
</BODY>
```

Текст

Одно и то же определение можно применить сразу к нескольким элементам. Для этого надо написать несколько селекторов подряд, разделяя их запятыми.

```

<STYLE TYPE="TEXT/CSS">
P, H1, H2, H3, BLOCKQUOTE, {FONT-
FAMILY: Arial, Helvetica}
</STYLE>
<BODY>
<P>
<H1>Заголовок 1</H1>
<H2>Заголовок 2</H2>
</P>
</BODY>

```

Заголовок 1

Заголовок 2

Такой подход позволяет сделать элементы выглядящими однородно.

Вариант третий

Чтобы не загромождать сам HTML-документ, таблицу стилей можно вынести в отдельный файл. Расположение описания стилей в отдельном файле имеет смысл в случае, если вы планируете применять эти стили к большему, чем одна, количеству страниц. CSS-файлы создаются теми же средствами, что и сами HTML-документы (например, в редакторе Блокнот). При этом одна и та же таблица стилей может использоваться неограниченное число раз. Этот документ сохраняется с расширением. CSS, например STYLES.CSS.

Для подключения файла CSS к HTML-документу используется элемент **<LINK>**, который должен быть помещен в заголовок между **<HEAD>** и **</HEAD>**.

<HEAD>

<TITLE> Главная страница **</TITLE>**

<LINK REL =«STYLESHEET» TYPE =«TEXT/CSS» HREF =«1.CSS»>

</HEAD>

Файл CSS может содержать различные комментарии, поясняющие назначение тех или иных участков кода. Комментарии в стандарте CSS заключаются в последовательность символов **«/»** и **«*»** следующим образом: **/* текст комментария */**

Создание специальных классов

Таблицы стилей позволяют не только задавать свойства отдельных элементов, но и создавать целые классы.

```

<HEAD>
<STYLE>
H1.krasota {color: red}
</STYLE>
</HEAD>
<BODY>
<H1>обычный заголовок</H1>
<H1 Class="krasota">Необычный   красный
заголовок</H1>
</Body>

```

Обычный заголовок

**Необычный
красный заголовок**

В данном примере используется селектор **H1.krasota** вместо обычного **H1**. Указанный стиль будет применяться к **H1** только там, где вы укажете атрибут **CLASS**.

Таким образом можно определять разные классы для одного и того же элемента.


```

<STYLE>
P.body {FONT-FAMILY: Arial, Helvetica; FONT-SIZE: 15 pt }
P.ft {FONT-FAMILY: Times New Roman, Times;
FONT-SIZE: 18 pt }
</STYLE>
<BODY>
<P CLASS="body">Текст</P>
<P CLASS="ft">Текст</P>
</BODY>

```

Текст
Текст

Можно создать и универсальный класс, привязанный к любому элементу, и он будет использовать указанный стиль.

```

<STYLE>
.ft {FONT-FAMILY: Times New Roman;
FONT-SIZE: 18 pt; COLOR=red }
</STYLE>
<BODY>
<P CLASS="ft">Текст</P>
<P>Текст</P>
<UL CLASS=ft>
<LI>Элемент списка
<Li>Элемент списка
</UL>
</BODY>

```

Текст
Текст
• Элемент списка
• Элемент списка

Элемент ****

При работе с таблицами стилей также широко используется элемент ****. Используя его, можно применять стиль вообще к чему угодно. Иначе говоря, **** — это такой контейнер, в котором определяется стиль и который действует вплоть до своего закрывающего тега.

Например, **** Текст ****,

или

<STYLE >

SPAN FONT-FAMILY: arial, helvetica: FONT-SIZE: 12 pt

</STYLE >

**** можно использовать в качестве специфического элемента, к которому можно применить определенное форматирование.

```

<STYLE>
SPAN {FONT-FAMILY: arial, helvetica: FONT-SIZE: 12 pt;
COLOR=blue }
</STYLE>
<BODY>
<P><SPAN>Текст</SPAN></P>
<P>Текст</P> </BODY>

```

Текст
Текст

На основе элемента **** можно создать класс.

```

<STYLE>
SPAN.DR {FONT: times new roman, times; FONT-SIZE:
28 pt; COLOR=red}
SPAN.SM {FONT: arial, helvetica;
FONT-VARIANT:small-caps}
</STYLE>
</HEAD>
<BODY>
<P><SPAN CLASS="DR">Э</SPAN>
<SPAN CLASS="SM">то было темным, дождливым
вечером.</SPAN> Тучи неслись с огромной скоростью,
одна норовила догнать другую. Изредка небо
прорезали яркие молнии.</P>

```

ЭТО БЫЛО ТЕМНЫМ, ДОЖДЛИВЫМ ВЕЧЕРОМ. Тучи неслись с огромной скоростью, одна норовила догнать другую. Изредка небо прорезали яркие молнии.

Элемент **** можно использовать с определенными заранее независимыми классами.

```

<HEAD>
<STYLE>
.dr {FONT: Times New Roman, Times;
FONT-SIZE: 18 pt; COLOR=red}
</STYLE>
</HEAD>
<BODY>
<P><SPAN CLASS="dr"> Текст с оформлением
</SPAN></P>
<P>Текст без оформления</P></BODY>

```

Текст с оформлением

Текст без оформления

Написав ****, можно применить указанный стиль к тексту. Точно так же универсальные классы можно применять и в любых других контейнерах.

С помощью элемента **STYLE** тега **** ... **** можно задавать выделение текста любым цветом:

```

<SPAN STYLE="BACKGROUND-COLOR: lightgreen">
Светлозеленый </SPAN>
<SPAN STYLE="BACKGROUND-COLOR: yellow"> Желтый
</SPAN>
<SPAN STYLE="BACKGROUND-COLOR: lightblue">
Светлосиний </SPAN>

```

Светлозеленый
Желтый
Светлосиний

Элемент <DIV>

<DIV> – еще один элемент, который имеет отношение к стилям и применяется для разделения всего HTML-документа на секции. По своей значимости и области действия он сопоставим с **<BODY >** или **<HEAD >**.

Элемент **<DIV >** может использоваться примерно так же, как ****. Для него могут быть определены правила таблиц стилей, он может иметь атрибут **CLASS** .

<pre><H1>Список</H1> <DIV STYLE="BACKGROUND-COLOR: yellow"> Элемент 1 Элемент 2 Элемент 3 </DIV></pre>	<h1>Список</h1> <ul style="list-style-type: none"> • Элемент 1 • Элемент 2 • Элемент 3
---	---

Элемент `<DIV >` может включать в себя совершенно разные типы контейнеров, например, такие, как `<P >` и ``. Именно для этого он и предназначен – для разделения страницы на секции, выполненные в едином стиле.

Стилевое оформление страницы

Также параметры элемента `<BODY>` можно задавать в стиле.

Например,

```
<STYLE TYPE=«text/css»>
```

```
<! —
```

BODY

```
{
BACKGROUND-ATTACHMENT: fixed;
BACKGROUND-REPEAT: no-repeat;
BACKGROUND-POSITION: 13px 13px;
BACKGROUND-COLOR: #FFFF00
}
```

```
— >
```

```
</STYLE>
```

BACKGROUND-COLOR: #ffff00; — определяет цвет фона всего документа

в шестнадцатеричном виде или названиями цветов.

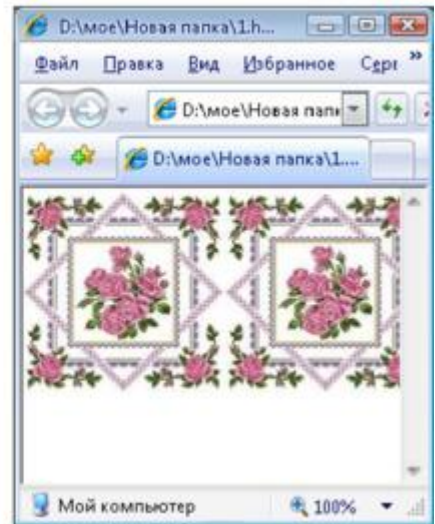
```
<STYLE TYPE="text/css">
BODY
{
BACKGROUND-COLOR: #FFFF00;
}
</STYLE>
```



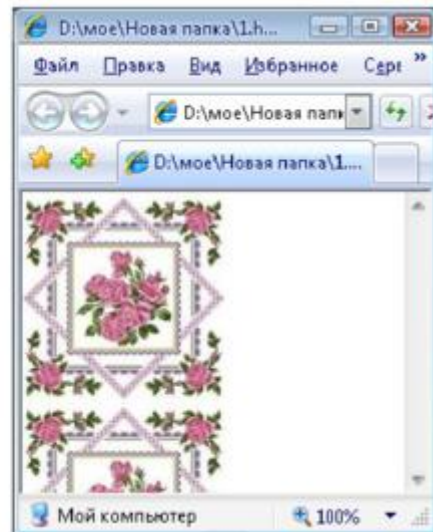
BACKGROUND-ATTACHMENT: FIXED; — определяет неподвижность фонового изображения при прокрутке страницы. Параметр `SCROLL` — заставляет фоновое изображение двигаться при прокрутке документа.

BACKGROUND-REPEAT: NO-REPEAT; — данный параметр запрещает размножить изображение, убрав его, изображение будет размножено по всей странице. Параметр `REPEAT` позволяет повторять изображение во всех направлениях; `REPEAT-X` — только по горизонтали; `REPEAT-Y` — только по вертикали.

```
<STYLE TYPE="text/css">
BODY
{
BACKGROUND-REPEAT: REPEAT-X;}
</STYLE>
```

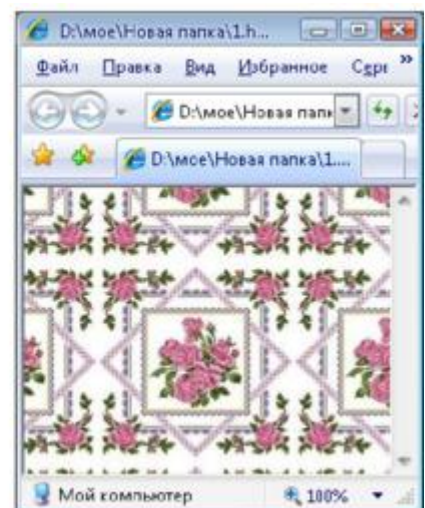


```
<STYLE TYPE="text/css">
BODY
{
BACKGROUND-REPEAT: REPEAT-Y;}
</STYLE>
```



BACKGROUND-POSITION: 13px 13px; — определяет позицию фонового изображения: от верхнего края и от левого края. В качестве значения этого свойства расстояние от левой и верхней границы видимой части экрана до изображения можно указать в пикселях, миллиметрах или сантиметрах.

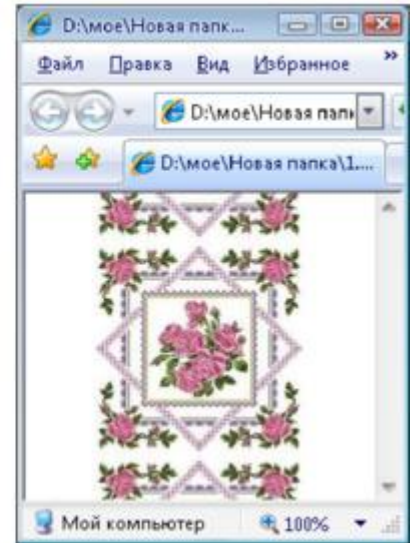
```
<STYLE TYPE="text/css">
BODY
{
BACKGROUND-POSITION: 50px 50px;}
</STYLE>
```



В случае если значение свойства **BACKGROUND-REPEAT** установлено **REPEAT-X** или **REPEAT-Y**, позволяет зафиксировать положение фонового изображения. Оно может принимать следующие значения: **TOP** — позиционирование по верхней границе экрана, **BOTTOM** — по нижней

границе, **LEFT** — выравнивание по левой границе экрана или ячейки таблицы, **RIGHT** — выравнивание по правой границе, **CENTER** — по центру экрана.

```
<STYLE TYPE="text/css">
BODY
{
BACKGROUND-POSITION: 50px 50px;
BACKGROUND-POSITION: center;
}
</STYLE>
```



Полоса прокрутки

Данные настройки применимы ко всем элементам, имеющим полосы прокрутки: сама страница (элемент **BODY**), текстовый блок (**TEXTAREA**) и т. д.

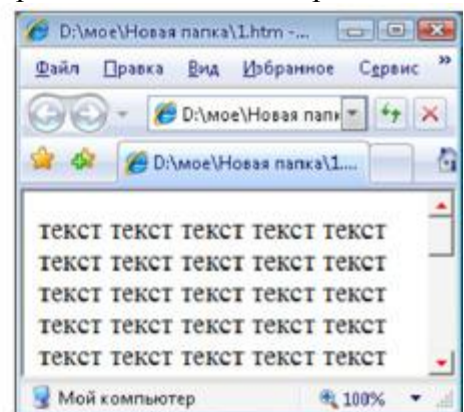
SCROLLBAR-3DLIGHT-COLOR — определяет цвет верха и левой части ползунка и кнопок со стрелками на полосе прокрутки.

```
<STYLE TYPE="text/css">
BODY
{
SCROLLBAR-3DLIGHT-COLOR: red;
}
</STYLE>
```



SCROLLBAR-ARROW-COLOR — устанавливает цвет стрелок на кнопке со стрелками.

```
<STYLE TYPE="text/css">
BODY
{
SCROLLBAR-ARROW-COLOR: red;
}
</STYLE>
```

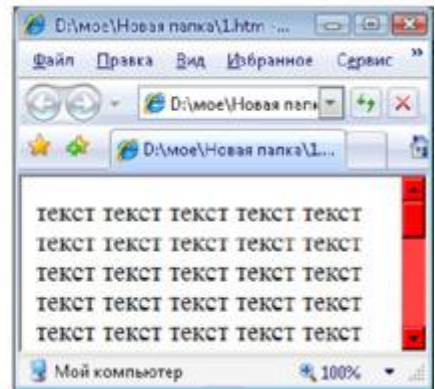


SCROLLBAR-BASE-COLOR — устанавливает цвет основных элементов ползунка: ползунка, кнопок со стрелками, дорожки для ползунка, если не определены параметры в **SCROLLBAR-FACE-COLOR**.

```

<STYLE TYPE="text/css">
BODY
{
SCROLLBAR-BASE-COLOR: red;
}
</STYLE>

```

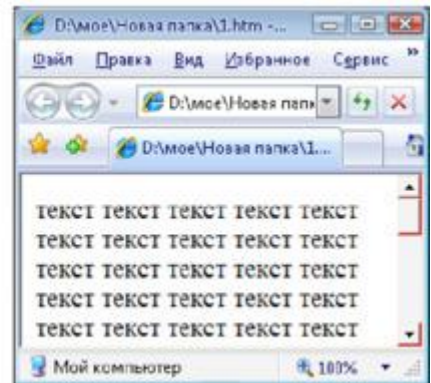


SCROLLBAR-DARKSHADOW-COLOR — устанавливает цвет тени для ползунка и кнопок со стрелками.

```

<STYLE TYPE="text/css">
BODY
{
SCROLLBAR-DARKSHADOW-COLOR:
red;
}
</STYLE>

```



SCROLLBAR-FACE-COLOR — устанавливает цвет ползунка и кнопок со стрелками. Если вы не задали параметр **SCROLLBAR-TRACK-COLOR**, у вас изменится цвет дорожки.

```

<STYLE TYPE="text/css">
BODY
{
SCROLLBAR-FACE-COLOR: red;
}
</STYLE>

```

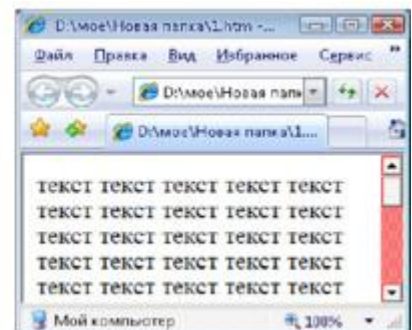


SCROLLBAR-HIGHLIGHT-COLOR — устанавливает цвет подсветки, создающий эффект объёмности. Когда кнопка не нажата, то цвет заливает левый верхний угол и стороны между ним, когда нажата — нижний правый угол.

```

<STYLE TYPE="text/css">
BODY
{
SCROLLBAR-HIGHLIGHT-COLOR: red;
}
</STYLE>

```

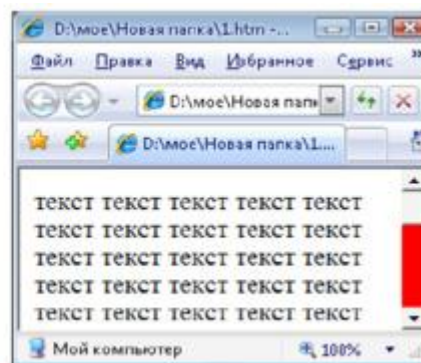


SCROLLBAR-TRACK-COLOR — устанавливает цвет дорожки для ползунка.

```

<STYLE TYPE="text/css">
BODY
{
SCROLLBAR-TRACK-COLOR: red;
}
</STYLE>

```



Комбинируя данные элементы, можно задать произвольное оформление для полосы прокрутки.

```

<STYLE TYPE="text/css">
BODY
{
SCROLLBAR-TRACK-COLOR: red;
SCROLLBAR-HIGHLIGHT-COLOR: blue;
SCROLLBAR-ARROW-COLOR: 880000;}
</STYLE>

```



Стилевое оформление текстов

Свойства цвета и фона

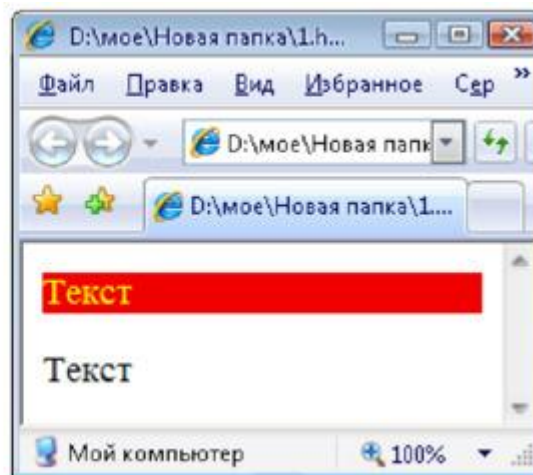
Свойство **COLOR** определяет цвет любого элемента CSS.

Свойство **BACKGROUND-COLOR** указывает на цвет фона какого-либо элемента, причем этот цвет устанавливается именно для отображения элемента, а не всего HTML-документа. Интерпретаторы разных браузеров обрабатывают данное свойство неодинаковым образом: например, Netscape Navigator выделяет фоновым цветом сам элемент страницы, причем выделение ограничивается его видимой шириной, а Internet Explorer растягивает выделение на всю ширину страницы.

```

<STYLE>
.ft {
COLOR=yellow;
background-color: #f00000;
}
</STYLE>
<BODY>
<P CLASS="ft">Текст</P>
<P>Текст</P>
</BODY>

```



Эффекты оформления шрифта

Оформление шрифтов задается при помощи свойства **TEXT-DECORATION**. Данное свойство может принимать одно из следующих значений:

- NONE** — отсутствие всяких эффектов;
- UNDERLINE** — подчеркнутый текст;

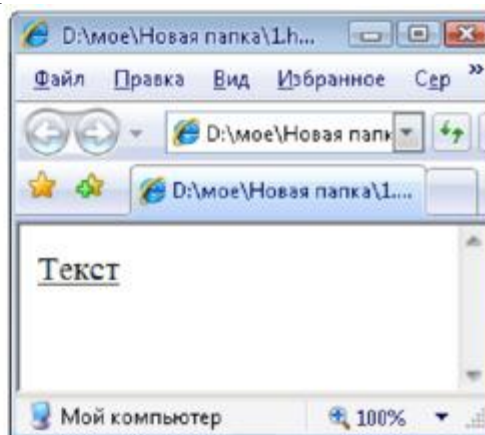
LINE-THROUGH — перечеркнутый текст;

OVERLINE — надстрочный текст;

BLINK — мерцающий текст.

Значение **OVERLINE** не поддерживается браузером Netscape Navigator, а значение **BLINK** — программой Microsoft Internet Explorer.

```
<STYLE>
.ft {
TEXT-DECORATION: UNDERLINE ;
}
</STYLE>
<BODY>
<P CLASS="ft">Текст</P>
</BODY>
```



Размещение элементов

Свойство **TEXT-ALIGN** задает расположение текста относительно границ рабочего окна браузера или ячейки таблицы. Соответственно, оно применяется совместно с так называемыми «блоковыми элементами», то есть элементами, поддерживающими отображение текстовых массивов: **P**, **H1-H6**, **TD**, **TR** и некоторыми другими.

Можно использовать одно из четырех значений данного свойства:

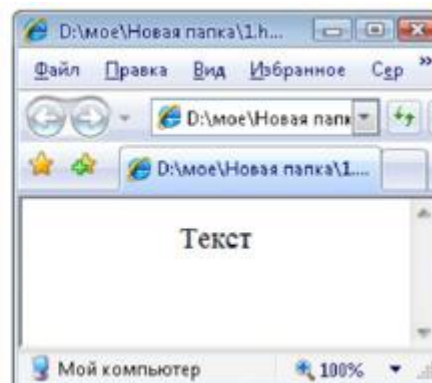
LEFT — выравнивание по левой границе экрана или ячейки таблицы;

RIGHT — по правой границе;

CENTER — выравнивание по центру;

JUSTIFY — растягивание текста по всей ширине экрана или табличной ячейки.

```
<STYLE>
.ft {
text-align: CENTER;
}
</STYLE>
<BODY>
<P CLASS="ft">Текст</P>
</BODY>
```



Свойство **VERTICAL-ALIGN** задает вертикальное положение элемента. Может принимать следующие значения:

BASELINE — по условной базовой линии;

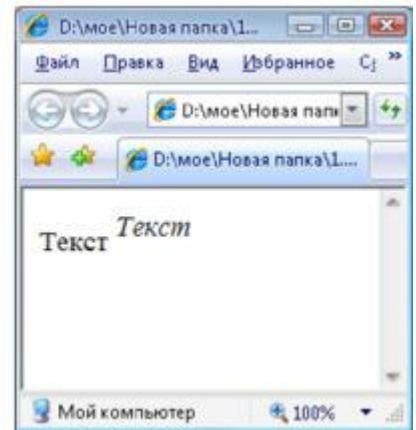
SUB — нижний индекс;

SUPER — верхний индекс;


```

<STYLE>
i.ss { vertical-align: super }
</STYLE>
<BODY>
<P>Текст<i class=ss> Текст</i></p>
</BODY>

```



Вертикальное расположение имеет смысл изменять, когда текст находится внутри какого-то элемента. Также вертикальное расположение можно задавать и в процентах.

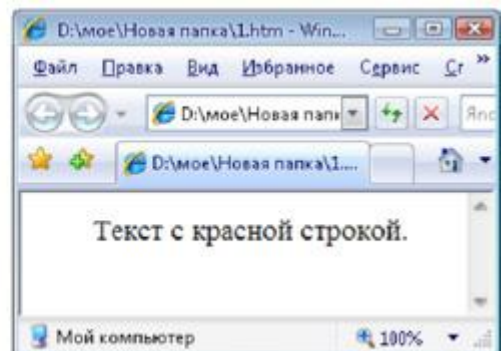
Абзацный отступ

В основном абзацные отступы создаются с помощью неразрывных пробелов (** **), но есть и другой способ — использование стилевого свойства **TEXT-INDENT**.

```

<DIV STYLE="text-indent: 2em;">Текст с
красной строкой. </DIV>

```



Здесь абзацный отступ равен двум символам максимальной ширины в данном шрифте (2em). Абзацный отступ можно задать в пунктах (100pt), процентах (10%) и т. д.

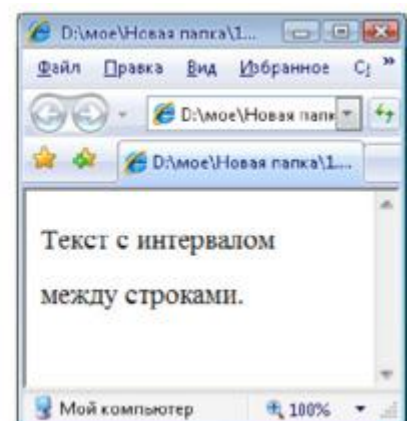
Интервалы

Интервалы между строками задаются при помощи свойства **LINE-HEIGHT**.

```

<DIV STYLE="line-height: 2em;">Текст с
интервалом между строками. </DIV>

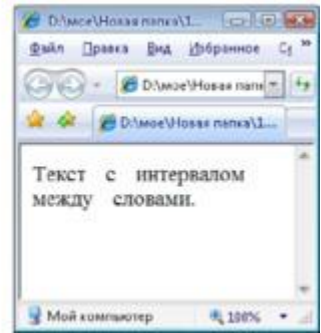
```



Интервал можно задавать и в процентах.

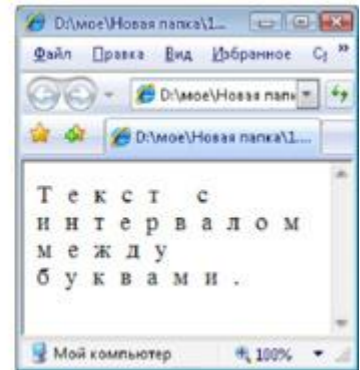
Интервал между словами задается свойством **WORD-SPACING**. Можно использовать отрицательные значения

`<DIV STYLE="WORD-SPACING: 12pt;">Текст с интервалом между словами. </DIV>`



Интервал между буквами задается свойством **LETTER-SPACING**.

`<DIV STYLE="LETTER-SPACING: 12pt;">Текст с интервалом между буквами. </DIV>`



Преобразование регистра

Свойство **TEXT-TRANSFORM** задает трансформацию определяемого элементом текстового блока:

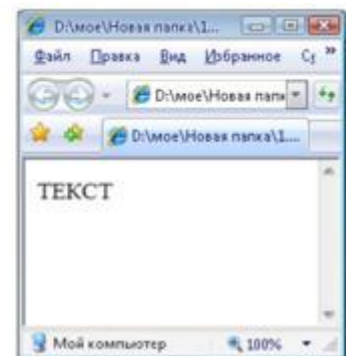
NORMAL — стандартное отображение текста без каких-либо изменений;

CAPITALIZE — каждое слово абзаца начинается с заглавной буквы, при этом регистр символов, определенный в коде HTML, игнорируется;

UPPERCASE — все символы текста становятся заглавными;

LOWERCASE — все символы становятся строчными. Это свойство поддерживается только браузером Microsoft Internet Explorer.

`<P STYLE="TEXT-TRANSFORM: UPPERCASE">текст</p>`



Свойства шрифтов

Свойство **FONT-FAMILY** — используется для указания шрифта или шрифтового семейства, которым будет отображаться элемент. Можно указать до трех семейств шрифта. В том случае, когда в правилах указано несколько семейств, браузер будет последовательно перебирать все перечисленные, пока не найдет того, которое присутствует на машине клиента.

Также вместо названия шрифта можно указать его URL. URL шрифта можно указывать в виде как абсолютного адреса (http://www.myserver.ru/fonts/font_name.ttf), так и относительного ([fonts/fontjame.ttf](#)).

Свойство **FONT-STYLE** — задает способ начертания шрифта:

NORMAL — нормальный (по умолчанию);

ITALIC — курсив;

OBLIQUE — наклонный.

Свойство **FONT-VARIANT** — задает варианты начертания шрифта:

NORMAL — нормальный (по умолчанию);

SMALL-CAPS — все буквы заглавные.

Свойство **FONT-VARIANT** поддерживает только Microsoft Internet Explorer.

Свойство **FONT-WEIGHT** — определяет степень жирности шрифта с помощью параметров:

NORMAL — стандартное написание;

BOLD — жирный шрифт;

BOLDER — очень жирный шрифт;

LIGHTER — тонкий шрифт.

Необходимо учитывать, что значения свойства **FONT-WEIGHT** обрабатываются интерпретаторами различных браузеров по-разному. Например, Microsoft Internet Explorer отображает текст, которому присвоен параметр **BOLDER**, так же, как текст, отмеченный значением **BOLD**, а Netscape Navigator отобразит его идентично тексту с параметром **NORMAL**. К тому же Internet Explorer не различает значения **NORMAL** и **LIGHTER**, поэтому эти отрывки текста отображаются в нем одинаковым образом.

Свойство **FONT-SIZE** — устанавливает размер шрифта. Значение этого свойства может быть указано как в абсолютных, так и в относительных величинах. Под абсолютными величинами подразумевается размер шрифта в условных пунктах (pt), сантиметрах (cm), миллиметрах (mm) или пикселах (px). Относительные величины содержат указание на размер шрифта в процентах от высоты символов, принятой по умолчанию, или в виде следующих определений:

SMALLER — самый маленький шрифт по отношению к установленному по умолчанию;

SMALL — маленький шрифт;

MEDIUM — средний;

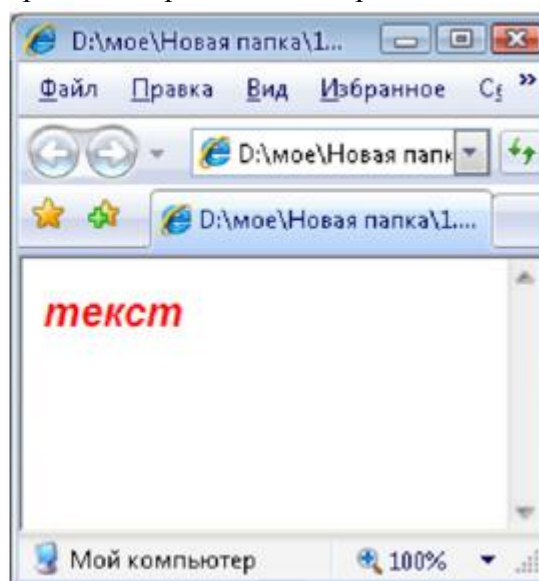
LARGE — большой шрифт;

LARGER — самый большой шрифт.

Необходимо помнить, что использование абсолютных значений более предпочтительно, поскольку относительные значения браузеры могут трактовать различным образом.

```
<Style Type="text/css">
```

```
p {  
font-family:Arial;  
font-weight: bolder;  
font-style: italic;  
FONT-SIZE: 14 pt.;  
color:red;  
}  
</style>  
<p>текст</p>
```



Атрибут LANG

По умолчанию язык, на котором написана страница, является частью определения XML, которое пишется в начале кода. Атрибут **LANG** определяет национальный язык различных элементов и позволяет браузеру принять решение о том, как отображать элемент, как его произносить (если браузер речевой), он также помогает поисковым системам в классификации сайтов.

Значение представляет собой двухбуквенное обозначение, например,

`<P LANG=«ru»> Этот абзац написан по-русски. </P>`

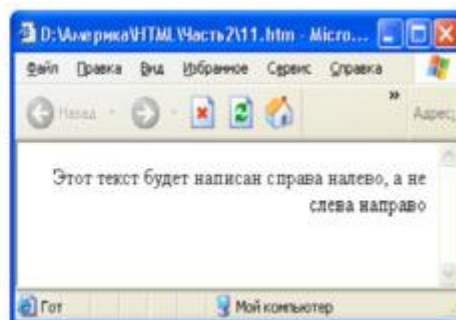
Вот некоторые двухбуквенные обозначения языков: **DE** — немецкий, **ES** — испанский, **FR** — французский, **HE** — иврит, **IT** — итальянский, **JA** — японский, **RU** — русский, **CH** — китайский.

Атрибут **LANG** может использоваться с элементом `<q>`, если цитата приводится на иностранном языке: `<P> <Q LANG=«en»> I can't believe she said that </Q>.- сказал Филипп. </P> <P> <Q Lang=«es»> Si. es verdad </Q>, — ответила Мария. </P>`

Направление текста

Во многих местах HTML-документа можно задать направление текста. Для этого используется атрибут **DIR**.

`<P DIR="RTL">Этот текст будет написан справа налево, а не слева направо</P>`



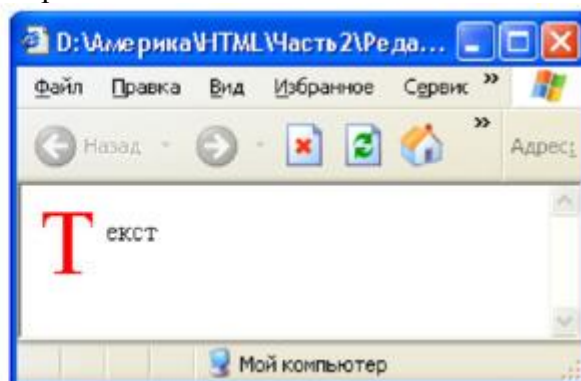
Атрибут **DIR** может быть включен в элементы `<TR>` или `<TD>`, тогда в строке или ячейке изменится направление текста. Эта возможность особенно часто используется при написании текста на таких языках, в которых принято писать справа налево (например, на иврите).

Следует помнить, что в разных браузерах результат применения этого атрибута будет разным. Он реализован далеко не во всех браузерах, даже современных.

Первая буква, первая строка

Для задания формата первой буквы текста или первой строки используются классы **FIRST-LETTER** и **FIRST-LINE**, которые прописываются в таблице стилей.

```
<STYLE TYPE="TEXT/CSS">
P.TT:FIRST-LETTER
{
FONT-SIZE: 50;
COLOR:RED;
FLOAT: LEFT;
MARGIN-RIGHT: 5px;
}</STYLE>
<P CLASS="TT">Текст</P>
```



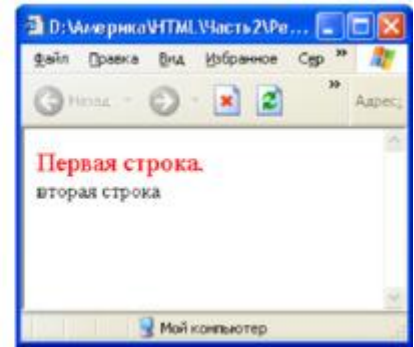
Применение этого кода заключается в создании буквицы в начале абзаца. Задается он атрибутом **CLASS=«TT»** в составе элемента `<P>`.

Для создания формата первой строки **FIRST-LETTER** нужно заменить на **FIRST-LINE**


```

<STYLE TYPE="TEXT/CSS">
P.TT:FIRST-LINE
{
FONT-SIZE: 50;
COLOR:red;
FLOAT: left;
MARGIN-RIGHT: 5px;
}</STYLE>
<P CLASS="TT">Первая строка<BR> вторая строка</P>

```



Стили списков

Стили списков применяются для настройки отображения элементов нумерованных и маркированных списков, определяемых в коде HTML тегом ****.

Для элементов списка можно использовать различную заливку, включив в тег **** атрибут **STYLE=«BACKGROUND: RED;»**.

```

<UL>
<LI style="background:red;"> Элемент 1
<LI style="background:green;"> Элемент 2
<LI style="background:cc0033;"> Элемент 3
</UL>

```



Стили ссылок

Используя таблицы стилей можно менять цвет ссылки, ее вид при наведении мыши и т. д.

:LINK — вид ссылки до того, как на ней щелкнули;

:VISITED — вид ссылки после того, как на ней щелкнули;

:HOVER — вид ссылки или объекта в тот момент, когда на них находится указатель;

:FOCUS — вид ссылки или объекта во время ввода текста или при выделении;

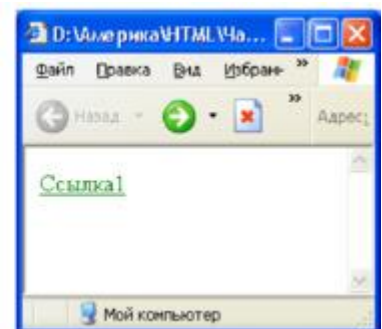
:ACTIVE — вид ссылки или объекта во время их выбора (при нажатой кнопке мыши или клавише Enter).

Эти классы определяются с помощью свойств **COLOR**, **FONT** и **BACKGROUND**.

```

<STYLE TYPE="text/css">
A:LINK {COLOR: red; BACKGROUND: white}
A:VISITED {COLOR: green; BACKGROUND: white}
A:HOVER {COLOR: gray; BACKGROUND: yellow}
A:ACTIVE {COLOR: blue; BACKGROUND: yellow}
</STYLE>
<A HREF=11.htm>Ссылка</A>

```



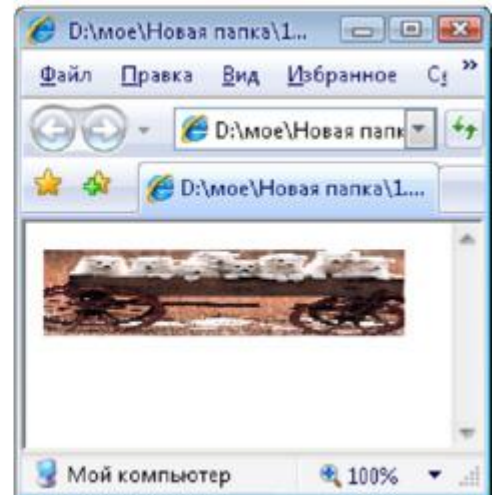
При указанном определении стили автоматически присваиваются элементам **<A>**, находящимся в соответствующих состояниях на странице. Но не все браузеры поддерживают работу этих стилей.

Стили изображений

Стиль изображения задается в таблице стилей и содержит теги по оформлению изображения. Использование такого подхода позволяет применять заданные стили для всех изображений на странице.

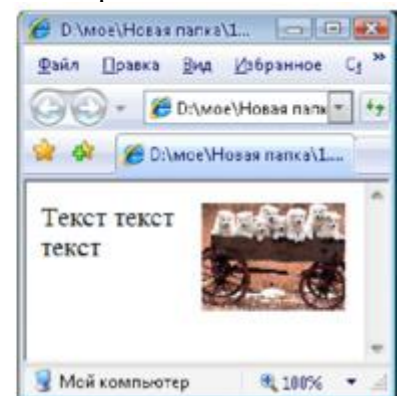
Свойства **WIDTH** и **HEIGHT** определяют соответственно ширину и высоту объекта в пикселях, миллиметрах, сантиметрах или процентах от значения, принятого по умолчанию. Применяются они в основном для задания геометрических размеров изображений различной величины в случаях, когда их необходимо выровнять по вертикали или горизонтали. Данные свойства можно использовать как вместе, так и по отдельности.

```
<STYLE>
IMG {
HEIGHT: 50%;
WIDTH: 200;
}
</STYLE>
<IMG SRC=11.jpg>
```



Свойство **FLOAT** указывает на расположение какого-либо объекта в окне браузера относительно других объектов, размещенных рядом. Оно может принимать одно из следующих значений: **NONE** — расположение объекта по умолчанию, **LEFT** — расположение объекта слева или **RIGHT** — расположение объекта справа.

```
<STYLE>
IMG {
FLOAT: right;
}
</STYLE>
<IMG SRC=11.jpg>
<P>Текст текст текст</P>
```



Свойство **BORDER** позволяет задать вид границ вокруг объектов. Могут быть установлены следующие параметры границы:

BORDER-WIDTH — ширина границы. Может быть задана значением в пикселях или с помощью зарезервированных слов **THIN** (тонкая), **MEDIUM** (средняя), **THICK** (толстая).

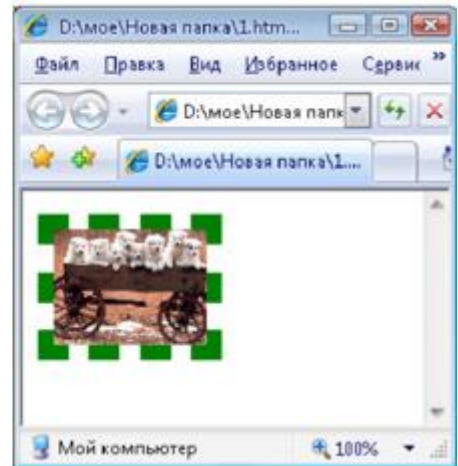
BORDER-COLOR — цвет границы. Может принимать значение **TRANSPARENT** для задания невидимой, но имеющей ширину, границы.

BORDER-STYLE — задает стиль рисования границы. Может принимать следующие значения: **NONE** (по умолчанию), **HIDDEN**, **DOTTED**, **DASHED**, **SOLID**, **DOUBLE**, **GROOVE**, **RIDGE**, **INSET**, **OUTSET**.

```

<STYLE>
IMG {
WIDTH:100;
BORDER-WIDTH: 10px;
BORDER-COLOR: green;
BORDER-STYLE: dashed;
</STYLE>
<IMG SRC=11.jpg>

```

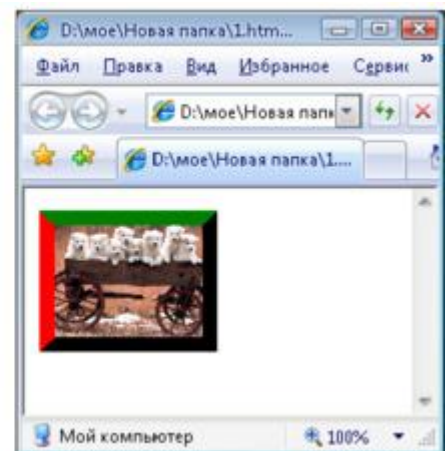


Также можно задать стили для каждой из сторон границы. Например: **BORDER-TOP** — верхняя граница, **BORDER-BOTTOM** — нижняя граница, **BORDER-LEFT** — левая граница и **BORDER-RIGHT** — правая граница.

```

<STYLE>
IMG {
WIDTH:100;
BORDER-WIDTH: 10px;
BORDER-TOP-COLOR: green;
BORDER-LEFT-COLOR: red;
BORDER-STYLE: solid;
</STYLE>
<IMG SRC=11.jpg>

```

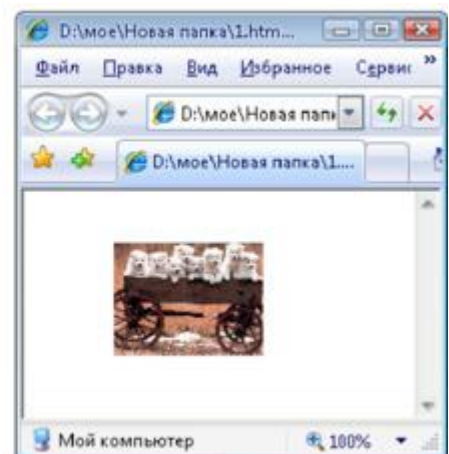


Свойства **MARGIN-LEFT**, **MARGIN-RIGHT**, **MARGIN-TOP** и **MARGIN-BOTTOM** определяют значения отступов вокруг содержащего их элемента в пикселях, сантиметрах или миллиметрах соответственно слева, справа, сверху и снизу. Отступ представляет собой пустое пространство, не заполненное никакими другими элементами. Данные свойства могут применяться совместно с элементами изображений, абзацев, таблиц, любых других компонентов.

```

<STYLE>
IMG
{
WIDTH:100;
MARGIN-TOP: 20;
MARGIN-LEFT: 50;
MARGIN-BOTTOM: 20;
}
</STYLE>
<IMG SRC=1.jpg>

```



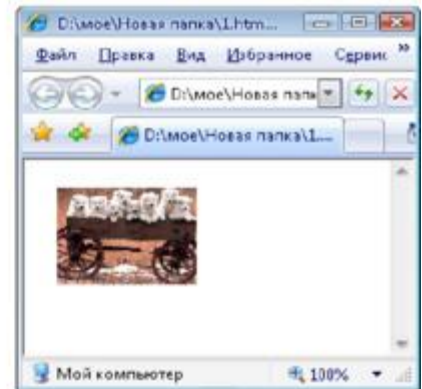
Одиночное значение **MARGIN: 5pt** означает, что устанавливаются поля верхней, нижней, левой и правой границ;

Два значения **MARGIN: 5pt 10pt** говорят о том, что первое из них применяется к верхней и нижней границе, а второе — к левой и правой;

Могут присутствовать и три значения **MARGIN: 5pt 10pt 10pt** — первое устанавливает верхнюю границу, второе — правую и левую, третье — нижнюю;

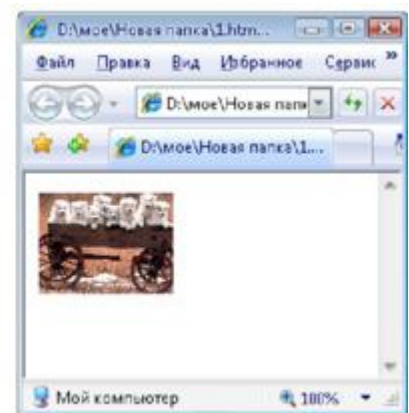
Четыре значения соответствуют верхнему, правому, нижнему и левому полям.

```
<STYLE>
IMG
{
WIDTH:100;
MARGIN: 5pt 10 pt 10 pt
}
</STYLE>
<IMG SRC=11.jpg>
```



Свойства **PADDING-LEFT**, **PADDING-RIGHT**, **PADDING-TOP** и **PADDING-BOTTOM** описывают значения отступов от видимых рамок объектов. Величина отступов также определяется в пикселях, сантиметрах или миллиметрах. Они могут применяться как вместе, так и по отдельности.

```
<STYLE>
IMG
{
WIDTH:100;
MARGIN-TOP: 20;
MARGIN-LEFT: 50;
MARGIN-BOTTOM: 20;
}
</STYLE>
<IMG SRC=1.jpg>
```

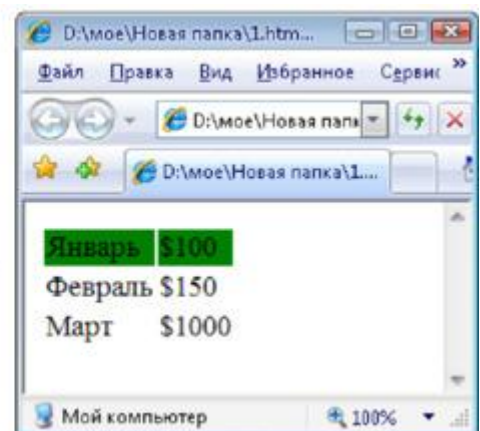


Свойства **MARGIN** и **PADDING** означают примерно одно и то же. Разница между свойствами **MARGIN** и **PADDING** состоит в том, что первое устанавливает дополнительные пробелы за границами блока, а второе включает их между границами и текстом.

Табличные стили

При создании таблиц приходится описывать их оформление для каждого элемента. Можно задавать оформление одновременно для групп элементов (ячеек, строк, столбцов). Для этого используются таблицы стилей.

```
<STYLE>
TR.1
{
BACKGROUND-COLOR: green;
}
</STYLE>
<TABLE>
<TR CLASS=1>
<TD>Январь</TD><TD>$100</TD></TR>
<TR><TD>Февраль</TD><TD>$150</TD></TR>
<TR><TD>Март</TD><TD>$1000</TD></TR>
</TABLE>
```



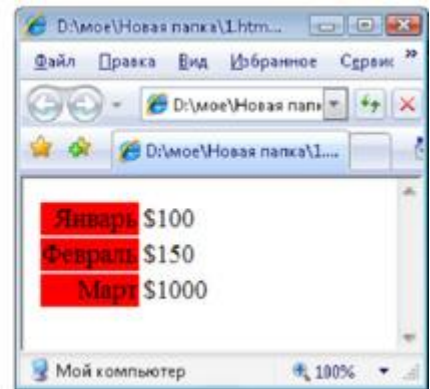
В примере создан класс **tr.1**, который используется для создания строк таблицы с фоном зеленого цвета. Таким же образом можно задавать стиль и для оформления ячеек таблицы.

Для создания стилей столбцов используются элементы **COLGROUP** и **COL**.


```

<STYLE>
COL.RED{BACKGROUND-COLOR: RED }
</STYLE>
<TABLE>
<COLGROUP></COL>
<COL
ALIGN="RIGHT"></COLGROUP>
CLASS="red"
<TR><TD>Январь</TD><TD>$100</TD></TR>
<TR><TD>Февраль</TD><TD>$150</TD></TR>
<TR><TD>Март</TD><TD>$1000</TD></TR>
</TABLE>

```

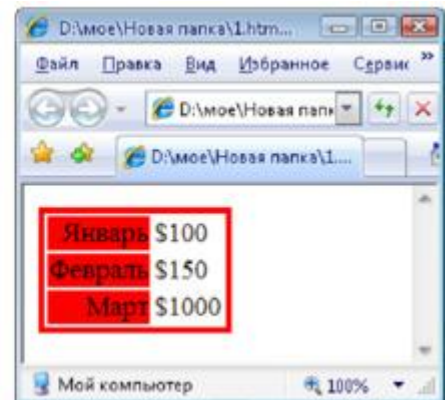


Границы и отступы внутри таблицы можно изменять, используя свойства **BORDER** и **MARGIN**.

```

<STYLE>
COL.RED{BACKGROUND-color: red }
</STYLE>
<TABLE STYLE="BORDER: solid red">
<COLGROUP></COL>
<COL
ALIGN="RIGHT"></COLGROUP>
CLASS="red"
<TR><TD>Январь</TD><TD>$100</TD></TR>
<TR><TD>Февраль</TD><TD>$150</TD></TR>
<TR><TD>Март</TD><TD>$1000</TD></TR>
</TABLE>

```



Вид границ можно настроить с помощью атрибута **BORDER**.

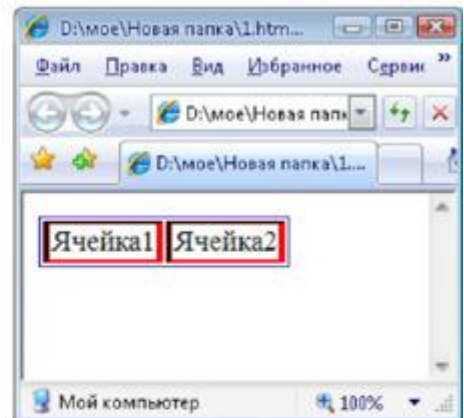
Свойства **BORDER-TOP-WIDTH**, **BORDER-BOTTOM-WIDTH**, **BORDER-RIGHT-WIDTH** и **BORDER-LEFT-WIDTH** определяют толщину обрамления таблиц и их ячеек.

Свойство **BORDER-COLOR** описывает цвет видимого обрамления таблиц.

```

<style type="text/css">
TABLE {
border-color: blue;
padding-left: 10px;
padding-right: 10px;
padding-top: 15px;
padding-bottom: 15px;
}
TD{
border-color: red;
border-top-width: 2px;
border-bottom-width: 2px;
border-right-width: 1mm;
border-left-width: 1mm;
}
</style>
<table border=1>
<tr><td>Ячейка1</td><td>Ячейка2</td></tr>
</table>

```



Свойство **VERTICAL-ALIGN** задает вертикальное положение элемента. Может принимать следующие значения:

TOP — позиционирование по верхней границе ячейки;

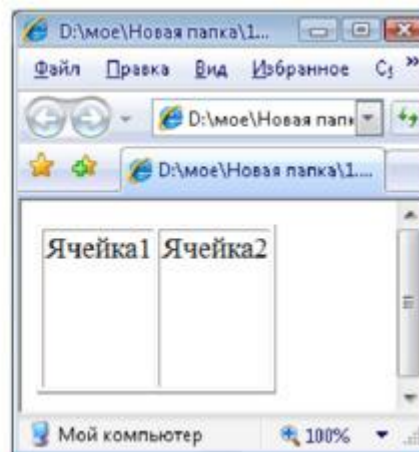
BOTTOM — по нижней границе ячейки;

MIDDLE — по центру ячейки.

```

<style type="text/css">
TD{
{ vertical-align: top }
}
</style>
<table border=1>
<tr height=100>
<td width=100>Ячейка1</td>
<td width=100 >Ячейка2</td>
</tr> </table>

```



Курсоры в таблицах

В таблицах можно использовать различные формы курсоров. Для отображения соответствующего курсора нужно создать таблицу и поместить текст между любыми ячейками.

cursor:crosshair	<td style=CURSOR:crosshair;> текст </td>
cursor:move	<td style=CURSOR:move;> текст </td>
cursor:e-resize	<td style=CURSOR:e-resize;> текст </td>
cursor:ne-resize	<td style=CURSOR:ne-resize;> текст </td>
cursor:nw-resize	<td style=CURSOR:nw-resize;> текст </td>
cursor:n-resize	<td style=CURSOR:n-resize;> текст </td>
cursor:sw-resize	<td style=CURSOR:sw-resize;> текст </td>
cursor:se-resize	<td style=CURSOR:se-resize;> текст </td>
cursor:s-resize	<td style=CURSOR:s-resize;> текст </td>
cursor:w-resize	<td style=CURSOR:w-resize;> текст </td>
cursor:text	<td style=CURSOR:text;> текст </td>
cursor:wait	<td style=CURSOR:wait;> текст </td>
cursor:help	<td style=CURSOR:help;> текст </td>
cursor:no-drop	<td style=CURSOR:no-drop;> текст</td>
cursor:not-allowed	<td style=CURSOR:not-allowed;> текст</td>
cursor:pointer	<td style=CURSOR:pointer;>текст </td>
cursor:row-resize	<td style=CURSOR:row-resize;>текст </td>

Позиционирование элементов и слои

Позиционирование элементов позволяет точно указывать их положение на странице. Благодаря своей совместимости практически со всеми стандартными браузерами, страницы, на которых используется позиционирование будут выглядеть примерно одинаково и в IE, и в Netscape 4, и в Netscape 6.

Общая схема позиционирования элементов выглядит так:

```
<STYLE TYPE=«TEXT/CSS»>.имя
{
POSITION: параметр (absolute, relative, static)
LEFT: значение;
TOP: значение;
WIDTH: значение;
HEIGHT: значение;
}
</STYLE>
<BODY>
<P CLASS=«имя»> текст </P>
</BODY>
```

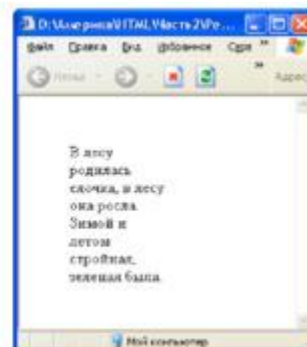
Свойство **POSITION** означает тип позиционирования и имеет следующие параметры:

ABSOLUTE — абсолютное позиционирование,
RELATIVE — относительное позиционирование,
STATIC — без позиционирования.

Свойства **LEFT** и **TOP** задают отступы слева и сверху. Отсчет ведется от верхнего левого угла страницы.

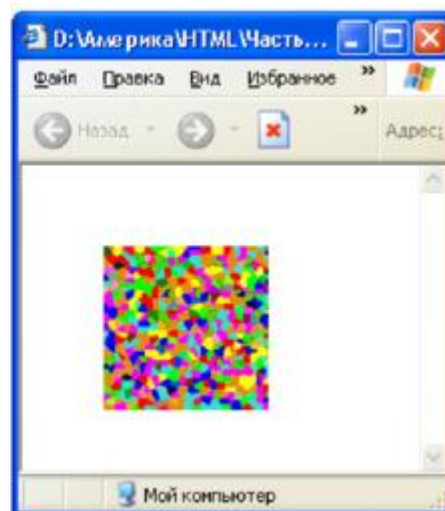
WIDTH и **HEIGHT** задают соответственно ширину и высоту элемента.

```
<STYLE TYPE="TEXT/CSS">
.1 {
POSITION: absolute;
LEFT: 50; TOP: 50;
WIDTH: 100; HEIGHT: 100;
}
</STYLE>
<BODY>
<P CLASS="1">В лесу родилась елочка, в лесу она росла. Зимой
и летом стройная, зеленая была.</P>
</BODY>
```



Аналогично можно позиционировать и изображения.

```
<STYLE TYPE="TEXT/CSS">
.1 {
POSITION: absolute;
LEFT: 50;
TOP: 50;
WIDTH: 100;
HEIGHT: 100;
}
</STYLE>
<BODY>
<IMG SRC=1.jpg CLASS="1">
</BODY>
```

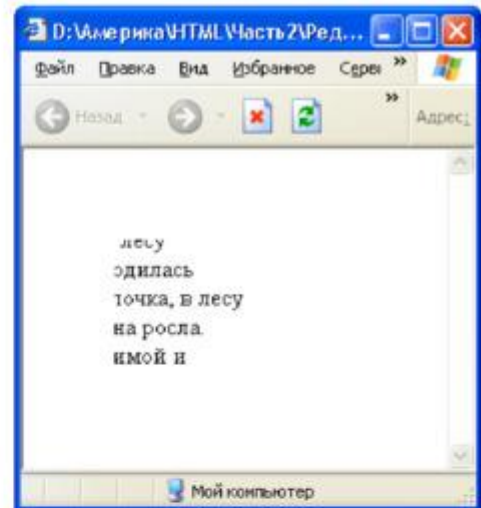


В дополнение используется свойство **CLIP: RECT** (вершина, правый угол, низ, левый угол). Оно предназначено для того, чтобы сделать видимой только какую-то часть страницы. Оно напоминает прием из графических редакторов под названием «обрезание изображения».


```

<STYLE TYPE="TEXT/CSS">
.1 {
POSITION: absolute;
LEFT: 50;
TOP: 50;
WIDTH: 100;
HEIGHT: 100;
CLIP: RECT(10px, 100px, 100px, 10px);
}
</STYLE>
<BODY>
<P CLASS="1">В лесу родилась елочка, в
лесу она росла. Зимой и летом стройная,
зеленая была.</P>
</BODY>

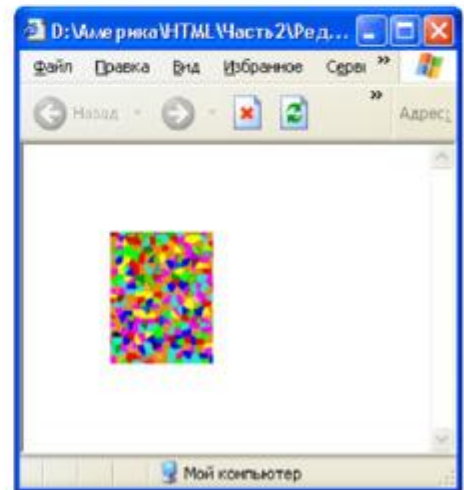
```



```

<STYLE TYPE="TEXT/CSS">
.1 {
POSITION: absolute;
LEFT: 50;
TOP: 50;
WIDTH: 100;
HEIGHT: 100;
CLIP: RECT(10px, 80px, 100px, 10px);
}
</STYLE>
<BODY>
<IMG SRC=1.jpg CLASS="1">
</BODY>

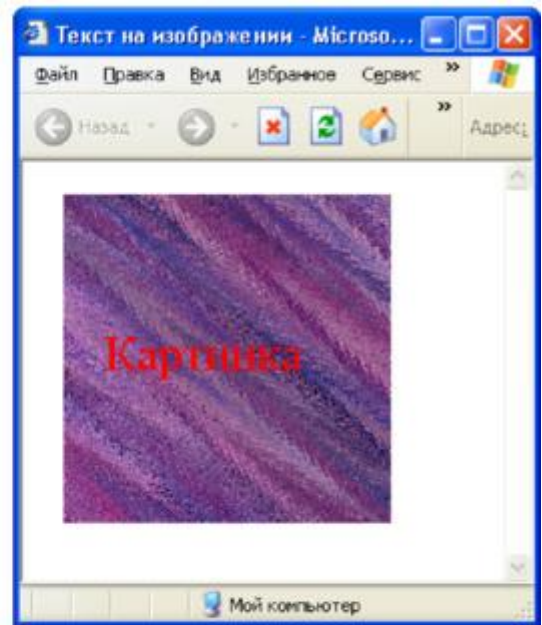
```



Наложение элементов

С помощью средств CSS-слоев можно располагать элементы Web-страниц слоями. Например, картинку размещать поверх картинки, текст поверх картинки, текст поверх текста и т. д. Для этого используется позиционирование каждого элемента на странице. Для порядка расположения слоев используется свойство **Z-INDEX**. Чем больше его значение, тем ближе к верхнему слою находится элемент.


```
<STYLE TYPE="TEXT/CSS">
.image
{
POSITION: absolute;
LEFT: 25; TOP: 20;
WIDTH: 200; HEIGHT: 200;
Z-INDEX: 0;}
.text
{
POSITION: absolute;
LEFT: 50; TOP:100;
Z-INDEX: 1;
FONT-SIZE:30;
COLOR: red;
}
</STYLE>
<BODY>
<IMG SRC="2.jpg" CLASS="image">
<P CLASS="text">Картинка</p>
</BODY>
```

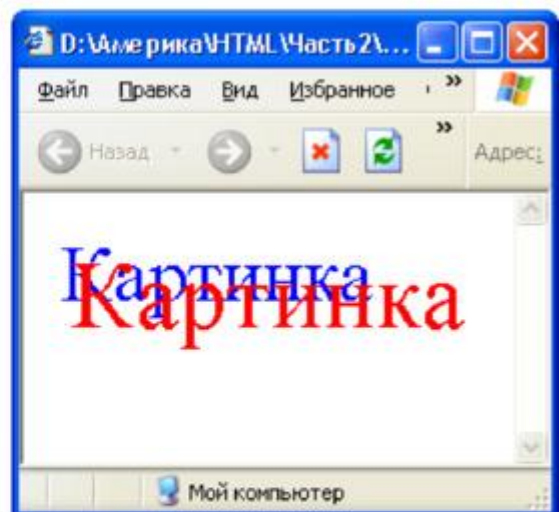
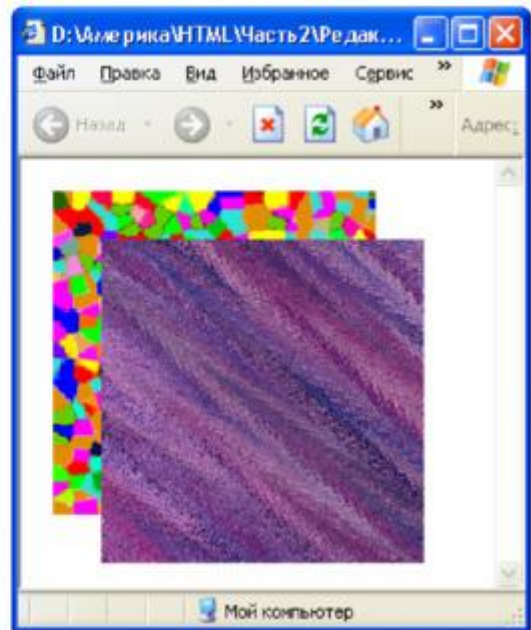


Аналогичным образом накладывается изображение на изображение и текст на текст.

```

<STYLE TYPE="TEXT/CSS">
.image1
{
POSITION: absolute;
LEFT: 20; TOP: 20;
WIDTH: 200; HEIGHT: 200;
Z-INDEX: 0;}
.image2
{
POSITION: absolute;
LEFT: 50; TOP:50;
WIDTH: 200; HEIGHT: 200;
Z-INDEX: 1;
}
</STYLE>
<BODY>
<IMG SRC="1.jpg" CLASS="image1">
<IMG SRC="2.jpg" CLASS="image2">
</BODY>
<STYLE TYPE="TEXT/CSS">
.text1
{
POSITION: absolute;
LEFT: 20; TOP: 20;
FONT-SIZE:40; COLOR: blue;
Z-INDEX: 0;}
.text2
{
POSITION: absolute;
LEFT: 25; TOP: 25;
Z-INDEX: 1;
FONT-SIZE:50; COLOR: red;
}
</STYLE>
<BODY>
<P CLASS="text1">Картинка</p>
<P CLASS="text2">Картинка</p>
</BODY>

```



Таким образом можно накладывать различное число слоев.

Использование свойства **VISIBILITY: HIDDEN** позволяет скрыть слой.

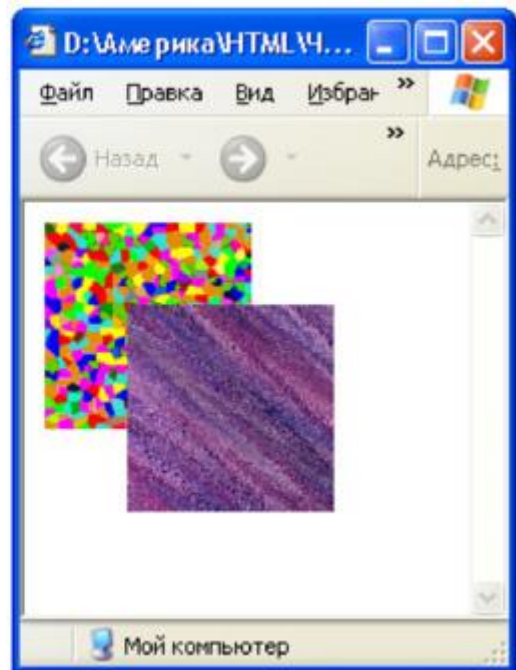
Использование элемента ID

Вместо элемента **CLASS** можно применять элемент **ID**. Его использование показано на примере:

```

<STYLE TYPE="TEXT/CSS">
#1
{
POSITION: absolute;
TOP: 10; LEFT: 10;
WIDTH: 100; HEIGHT:100;
}
#2 {
POSITION: absolute;
TOP: 50; LEFT: 50;
HEIGHT: 100; WIDTH: 100;
}
</STYLE>
<BODY>
<IMG SRC="1.jpg" ID="1">
<IMG SRC="2.jpg" ID="2">
</BODY>

```



Фильтры

Фильтры, применяемые к тексту

Для оформления текстов можно использовать различные фильтры.

- **Фильтр Mask** (Выделяет текст);
- **Фильтр Shadow** (Создает тень);
- **Фильтр DropShadow** (Еще один вид тени);
- **Фильтр FlipH** (Переворачивает текст горизонтально);
- **Фильтр FlipV** (Переворачивает текст вертикально);
- **Фильтр Glow** (Сияющий текст);
- **Фильтр Wave** (Волнистый текст);
- **Фильтр Blur** (Размытие текста).

Фильтр Mask

Выделяет текст, как будто его выделили его мышью.

STYLE=«FILTER: MASK (COLOR=«Color'»)»

COLOR — цвет выделения в шестнадцатеричном виде или название цвета на английском языке. Такое определение цвета применяется во всех фильтрах.

```

<P STYLE="FILTER:MASK(COLOR='RED'); WIDTH: 200;
HEIGHT: 28; FONT-SIZE: 16pt; FONT-WEIGHT: BOLD;
TEXT-ALIGN: CENTER;"> Пример </P>

```

Пример

Фильтр Shadow

Создает тень текста.

STYLE=«FILTER: SHADOW (COLOR=„COLOR“, DIRECTION=„DIRECTION“»)»

COLOR — цвет тени;

DIRECTION — направление тени (от 0 до 315).

```

<P STYLE="FILTER:SHADOW(COLOR='#000000',
DIRECTION='130'); WIDTH: 200; HEIGHT: 28; FONT-SIZE: 16PT;
FONT-WEIGHT: BOLD; TEXT-ALIGN: CENTER; COLOR: RED;">
Пример </P>

```

Пример

Фильтр DropShadow

Добавляет тень к тексту.

STYLE=«FILTER: DROPSHADOW (COLOR=„COLOR“, OFFX=„OFFX“, OFFY=„OFFY“, POSITIVE=„POSITIVE“»

COLOR — цвет тени;

OFFX — смещение тени по X;

OFFY — смещение тени по Y;

POSITIVE — тень слева или справа (0 или 1 соответственно).

```
<P STYLE="FILTER: DROPSHADOW (COLOR='#7FFF00',  
OFFX='3',OFFY='3', POSITIVE='1'); WIDTH: 200; HEIGHT: 28; FONT-  
SIZE: 16PT; FONT-WEIGHT: BOLD; TEXT-ALIGN: CENTER; COLOR:  
GREEN;"> Пример </P>
```

Пример

Фильтр FlipH

Переворачивает текст горизонтально.

STYLE=«FILTER: FLIPH»

```
<P STYLE="FILTER:FLIPH; WIDTH: 200; HEIGHT: 28; FONT-SIZE:  
16PT; FONT-WEIGHT: BOLD; TEXT-ALIGN: CENTER; COLOR:  
RED;"> Пример </P>
```

Пример

Фильтр FlipV

Переворачивает текст вертикально.

STYLE=«FILTER: FLIPV»

```
<P STYLE="FILTER:FLIPV; WIDTH: 200; HEIGHT: 28; FONT-SIZE:  
16PT; FONT-WEIGHT: BOLD; TEXT-ALIGN: CENTER; COLOR:  
RED;"> Пример </P>
```

Пример

Фильтр Glow

Создает эффект сияния.

STYLE=«FILTER: GLOW (STRENGTH=«Strength’, COLOR=«Color’»

STRENGTH — сила, с которой текст засияет (0—100);

COLOR — цвет, которым текст засияет.

```
<P STYLE="FILTER:GLOW(STRENGTH='3', COLOR='#00FF00');  
WIDTH: 200; HEIGHT: 28; FONT-SIZE: 16PT; FONT-WEIGHT: BOLD;  
TEXT-ALIGN: CENTER; COLOR: DARKGREEN;"> Пример </P>
```

Пример

Фильтр Wave

Делает текст волнистым.

STYLE=«FILTER: WAVE (FREQ=„FREQ“, ADD=„ADD“, LIGHTSTRENGTH=„LIGHTSTRENGTH“, PHASE=„PHASE“, STRENGTH=„STRENGTH“»

FREQ — количество волн;

ADD — показать/скрыть окантовку (1 или 0 соответственно);

LIGHTSTRENGTH — сила волн;

PHASE — угол волн;

STRENGTH — интенсивность волн.

```
<P STYLE="FILTER:WAVE(FREQ='5', ADD='0',  
LIGHTSTRENGTH='1', PHASE='2', STRENGTH='2');  
WIDTH: 200; HEIGHT: 28; FONT-SIZE: 16PT; FONT-  
WEIGHT: BOLD; TEXT-ALIGN: CENTER; COLOR:  
GREEN;"> Пример </P>
```

Пример

Фильтр Blur

Размывает текст в определенную сторону.

**STYLE=«FILTER: BLUR (ADD=„ADD“, DIRECTION=„DIRECTION“,
STRENGTH=„STRENGTH“)»**

ADD — умеренное или сильное размытие (соответственно 1 или 0);

DIRECTION — в какую сторону произойдет размытие (от 0 до 315);

STRENGTH — смещение размытия.

```
<P STYLE="FILTER:BLUR(ADD='1', DIRECTION='30',  
STRENGTH='10'); WIDTH: 200; HEIGHT: 28; FONT-SIZE: 16PT;  
FONT-WEIGHT: BOLD; TEXT-ALIGN: CENTER; COLOR:  
BLUE;"> Пример </P>
```

Пример

Обратите внимание: цвет размытия не задается, фильтр задает его сам; в данном примере определен только цвет текста.

Фильтры, применяемые к изображению

Для оформления изображений также используются различные фильтры.

- **Фильтр Alpha** (Осветление изображение);
- **Фильтр Blur** (Размытие изображения);
- **Фильтр Invert** (Инверсия изображения);
- **Фильтр Gray** (Черно-белое изображение);
- **Фильтр Chroma** (Определенный цвет делается прозрачным);
- **Фильтр XRay** (Негативное изображение);
- **Фильтр RevealTrans** (Появление изображения).



Исходное изображение

Фильтр Alpha

Осветляет изображение.

**STYLE=«FILTER: ALPHA (OPACITY=«Opacity», FINISHOPACITY=«Finishopacity»,
STYLE=«Style», STARTX=«X», STARTY=«и», FINISHX=«X», FINISHY=«Y»»»**

OPACITY — степень осветления (0—100. 0 светло, 100 темно);

FINISHOPACITY — конечная степень осветления (0—100);

STYLE — метод осветления (0 — без осветления, 1 — линия, 2 — круг, 3 —
прямоугольник);

STARTX, STARTY, FINISHX, FINISHY — координаты осветляемой области.

```
<IMG STYLE="filter:Alpha (Opacity='90', FinishOpacity='45',  
Style='2')" SRC="1.jpg" WIDTH="200" HEIGHT="200">
```



Фильтр Blur

Размывает изображение в определенную сторону.

**STYLE=«FILTER: BLUR (ADD=«Add», DIRECTION=«Direction», STRENGTH
=«Strength»»**

ADD — умеренное или сильное размытие (соответственно 1 или 0);

DIRECTION — в какую сторону произойдет размытие (от 0 до 315);

STRENGTH — смещение размытия.

```
<IMG STYLE="filter:Blur (Add='1', Direction='310',  
Strength='10')" SRC="1.jpg" WIDTH="200" HEIGHT="200">
```



Фильтр Invert

Инверсия изображения.

STYLE=«FILTER: INVERT»

```
<IMG STYLE="filter:Invert" SRC="1.jpg" WIDTH="200"  
HEIGHT="200">
```



Фильтр Gray

Делает изображение черно-белым.

STYLE=«FILTER: GRAY»

```
<IMG STYLE="filter:Gray" SRC="1.jpg" WIDTH="200"
HEIGHT="200">
```



Фильтр Chroma

Определенный цвет делается прозрачным.

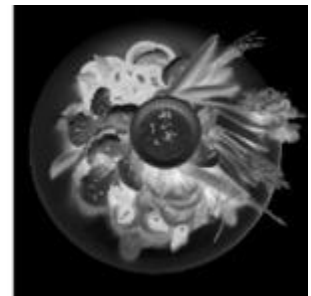
```
STYLE=«FILTER: CHROMA (COLOR=«Color'»)»
<IMG STYLE="filter:Chroma(Color='FFFFFF') SRC="1.jpg"
WIDTH="200" HEIGHT="200">
```



Фильтр Xray

Негативное изображение.

```
STYLE=«FILTER: XRAY»
<IMG STYLE="filter:Xray" SRC="1.jpg" WIDTH="200"
HEIGHT="200">
```



Фильтр RevealTrans

Появление изображения. Используется с применением скриптов.

```
STYLE=«FILTER: REVEALTRANS (DURATION=«Duration',
TRANSITION=«Transition'»)»
```

DURATION — время, за которое появится изображение (в секундах)

TRANSITION — эффект (совершенно разные эффекты от 0 до 23). Значение 23 меняет эффект появления изображения после каждой перезагрузки страницы.

```

<SCRIPT>
function Start() // Функция запуска фильтра. Кнопка
Показать.
{
RevealTransFilter.filters.item(0).Apply()
RevealTransFilter.filters.item(0).Play()
RevealTransFilter.style.visibility=""
}
function Clear() // Функция скрытия изображения. Кнопка
Сброс.
{
RevealTransFilter.style.visibility="hidden"
}
</SCRIPT>
<IMG id="RevealTransFilter"
STYLE="filter:revealTrans(Duration=1,
Transition=23); VISIBILITY: hidden;" SRC="1.jpg"
WIDTH="200" HEIGHT="200"><br>
<input type=button value=" Показать " OnClick="Start()">
<input type="reset" value=" Сброс " OnClick="Clear()">

```

