

---

## *Лекция \_3*

# *Введение в программирование на языке гипертекстовой разметки HTML*

## *Расшифровка некоторых аббревиатур*

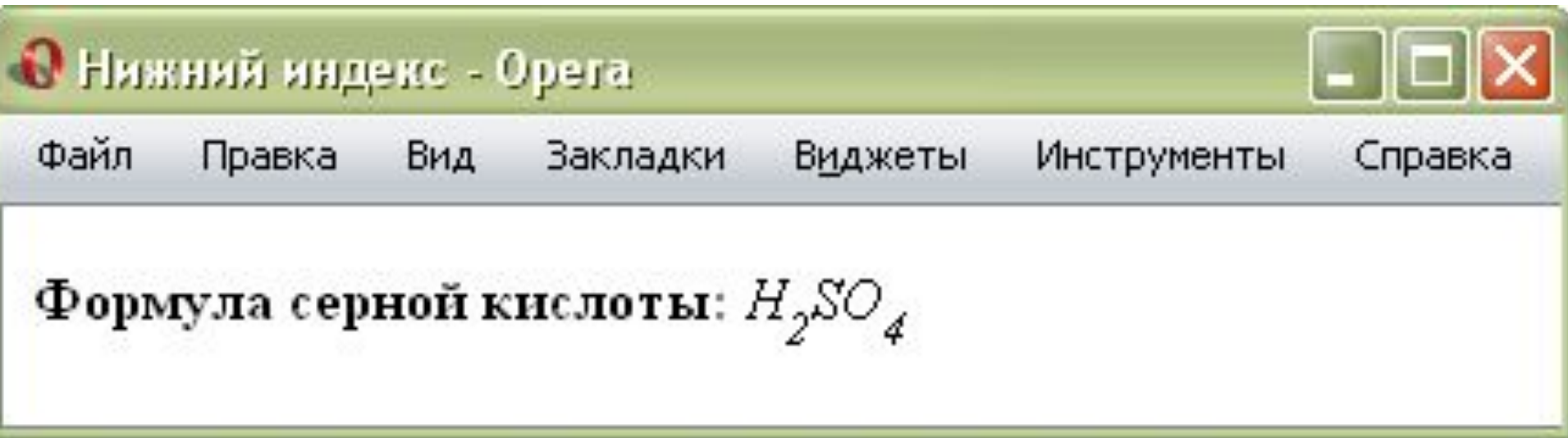
TBODY	table body
TD	table data cell
TH	table header cell
TR	table row
HR	horizontal rule
LI	list item
OL	ordered list
UL	unordered list
DD	definition description
DL	definition list
DT	definition term
FRAME	subwindow

## *Верхний и нижний индекс*

Индексом по отношению к тексту называется смещение символов относительно базовой линии вверх или вниз. В зависимости от положительного или отрицательного значения, индекс называется, соответственно, верхним или нижним. Они применяются в математике, физике, химии и для обозначения единиц измерения. HTML предлагает два тега для создания индекса: `<sup>` — верхний индекс и `<sub>` — индекс нижний. Текст, помещенный в один из этих контейнеров, обозначается меньшим размером, чем базовый текст и смещается относительно горизонтали.

## *Верхний и нижний индекс*

```
<body>  
  <p>  
    <b>Формула серной кислоты:</b>  
    <i>H<sub>2</sub>SO<sub>4</sub></i>  
  </p>  
</body>
```



## *Ссылки*

Являются основой гипертекстовых документов и позволяют переходить с одной Веб-страницы на другую.

Их особенность состоит в том, что сама ссылка может вести не только на HTML-файлы, но и на файл любого типа, причем этот файл может размещаться совсем на другом сайте.

Важно, чтобы к документу, на который делается ссылка, был доступ.

Если путь к файлу можно указать в адресной строке браузера, и файл при этом будет открыт, то на него можно сделать ссылку.

Для создания ссылки необходимо сообщить браузеру: что является ссылкой,

что является адресом документа, на который делают ссылку.

Оба действия выполняются с помощью тега <A>.

Синтаксис создания ссылок таков:

```
<a href="URL">текст ссылки</a>
```

## *Ссылки*

Параметр href определяет URL (Universal Resource Locator, универсальный указатель ресурса)/

Т.е. адрес документа, на который следует перейти, а содержимое контейнера <A> является ссылкой.

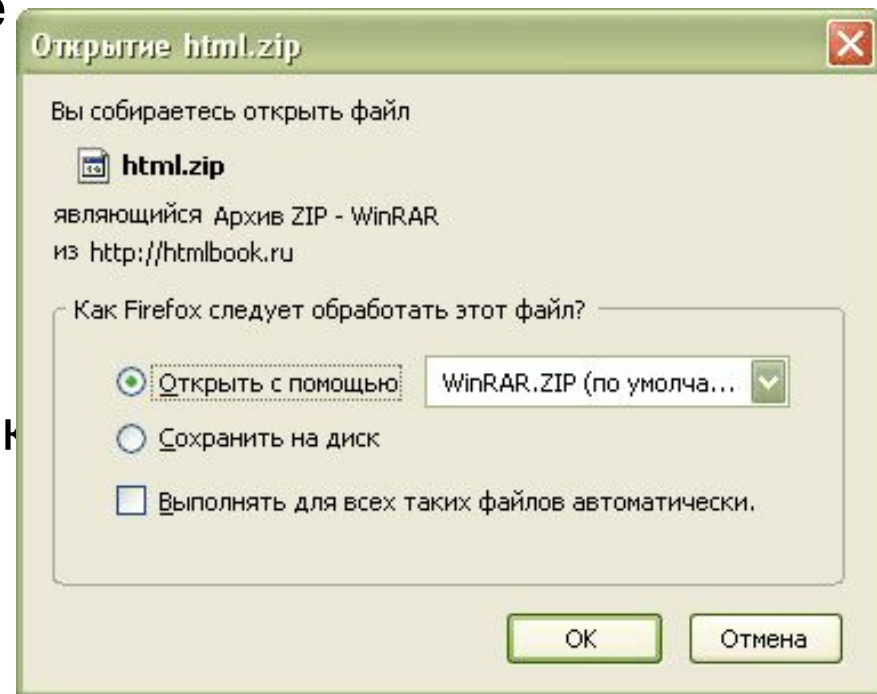
Текст, расположенный между тегами <A> и </A>, по умолчанию становится синего цвета и подчеркивается.

```
<head>
  <title>Ссылки на странице</title>
</head>
<body>
  <p><a href="dog.html">Собаки</a></p>
  <p><a href="cat.html">Кошки</a></p>
</body>
```

## Ссылки

Если указана ссылка на файл, которого не существует, его имя в параметре `href` набрано с ошибкой, то такая ссылка называется **битая**. Битых ссылок следует избегать, т.к. они вводят посетителей сайта в заблуждение.

Файл по ссылке открывается в окне браузера только тогда, когда браузер знает тип документа. Но если ссылку можно сделать на файл любого типа, то браузер не всегда может отобразить документ. При этом выводится сообщение, как следует обработать файл — открыть его или сохранить в указанную папку.



## *Абсолютные и относительные ссылки*

Абсолютные адреса должны начинаться с указания протокола (обычно http://) и содержать имя сайта.

Относительные ссылки ведут отсчет от корня сайта или текущего документа.

```
<title>Абсолютный адрес</title>
```

```
</head>
```

```
<body> <p>
```

```
    <a href="http://htmlbook.ru">Изучение HTML</a>
```

```
</p> </body>
```

Здесь ссылка вида `<a href="http://htmlbook.ru">Изучение HTML</a>` является абсолютной и ведет на главную страницу сайта `htmlbook.ru`.

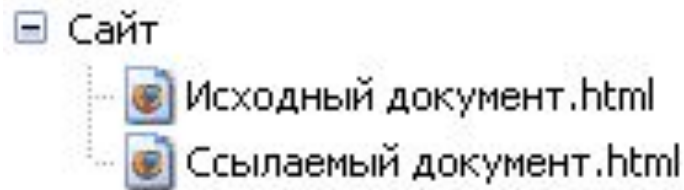
**Замечание** При указании в качестве ссылки каталога сайта (например, `http://htmlbook.ru/css/`), отображается индексный файл. Это файл, который загружается по умолчанию при обращении к каталогу без явного указания имени файла. В качестве индексного файла выступает документ с именем **`index.html`**.



## Абсолютные и относительные ссылки

При создании относительных ссылок надо понимать, какое значение для параметра href следует указывать, т.к. оно зависит от исходного расположения файлов. Рассмотрим варианты.

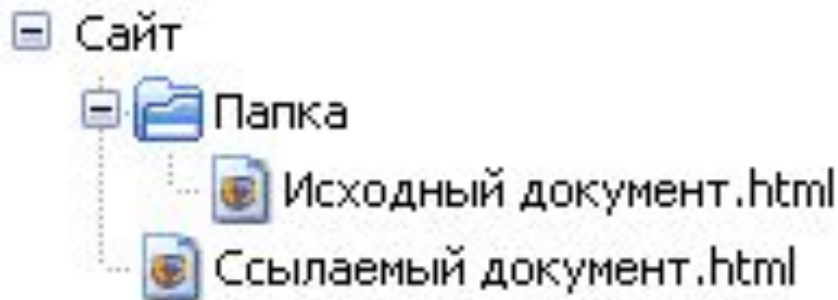
1. Файлы располагаются в одной папке.



Необходимо сделать ссылку из исходного документа на ссылаемый. В таком случае код будет следующий.

`<a href="Ссылаемый документ.html">Ссылка</a>`.

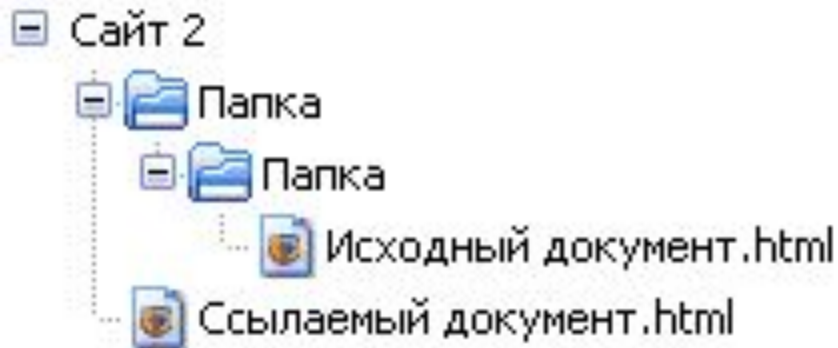
2. Файлы размещаются в разных папках.



`<a href=" ../Ссылаемый документ.html">Ссылка</a>`

## Абсолютные и относительные ссылки

3. Файлы размещаются в разных папках.



Исходный файл находится в двух вложенных Папках.

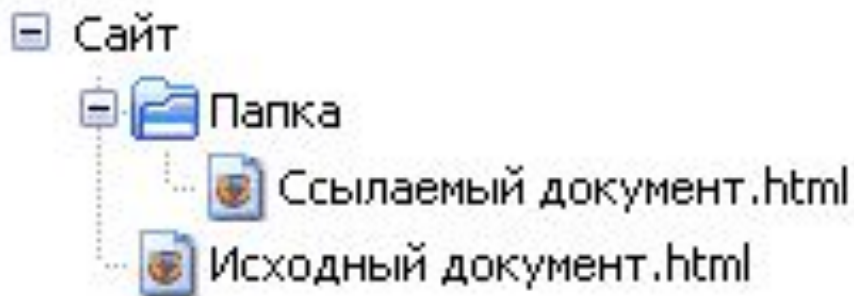
Чтобы сослаться на документ в корне сайта, требуется повторить написание предыдущего примера два раза.

```
<a href="../../../Ссылаемый документ.html">Ссылка</a>
```

Аналогично обстоит дело с любым числом вложенных папок.

# Абсолютные и относительные ссылки

4. Файлы размещаются в разных папках.



Исходный файл располагается в корне сайта, а файл, на который необходимо сделать ссылку - в папке. Тогда путь к файлу будет следующий.

`<a href="Папка/Ссылаемый документ.html">Ссылка</a>`

Здесь никаких дополнительных точек и слэшей перед именем папки нет. Если файл находится внутри не одной, а двух папок, то путь к нему записывается так.

`<a href="Папка 1/Папка 2/Ссылаемый документ.html">Сылка</a>`

**Ссылки относительно корня сайта.** Рассмотрим путь к файлу относительно корня сайта. `"/Папка/Имя файла"` со слэшем вначале. Запись `<a href="/course/">Курсы</a>` означает, что ссылка ведет в папку с именем `course`. Она располагается в корне сайта, а в ней необходимо загрузить индексный файл.

## *Абсолютные и относительные ссылки*

**Виды ссылок.** Любая ссылка на Веб-странице может находиться в одном из следующих состояний.

**Непосещенная ссылка.** Такое состояние характеризуется для ссылок, которые еще не открывали. По умолчанию непосещенные текстовые ссылки изображаются синего цвета и с подчеркиванием.

**Активная ссылка.** Ссылка помечается как активная в момент ее открытия. Поскольку время между нажатием на ссылку и началом загрузки нового документа достаточно мало, подобное состояние ссылки весьма кратковременно. Активной ссылка становится также, при ее выделении с помощью клавиатуры. Цвет такой ссылки по умолчанию красный.

**Посещенная ссылка.** Как только пользователь открывает документ, на который указывает ссылка, она помечается как посещенная и меняет свой цвет на фиолетовый, установленный по умолчанию.

**Правила вложений для тега <A>.** Любая ссылка является встроенным элементом, поэтому для нее действуют те же правила, что и для встроенных элементов. Нельзя размещать внутри тега <a> блочные элементы, но можно делать наоборот, и вкладывать ссылку в блочный контейнер.

## Параметры ссылок

Мы рассмотрели параметр `href` тега `<a>`. Рассмотрим еще другие необязательные атрибуты этого тега.

**Target.** По умолчанию, при переходе по ссылке документ открывается в текущем окне или фрейме. Это условие может быть изменено параметром `target` тега `<a>`. Его синтаксис: `<a target="имя окна">...</a>` В качестве аргумента используется имя окна или фрейма, заданное параметром `name`. Если установлено несуществующее имя, то будет открыто новое окно. В качестве зарезервированных имен применяются следующие аргументы.

`_blank` — загружает страницу в новое окно браузера.

`_self` — загружает страницу в текущее окно (это значение задается по умолчанию).

`_parent` — загружает страницу во фрейм-родитель, если фреймов нет, то этот параметр работает как `_self`.

`_top` — отменяет все фреймы и загружает страницу в полном окне браузера, если фреймов нет, то этот параметр работает как `_self`.

## *Параметры ссылок: title*

Добавляет поясняющий текст к ссылке в виде всплывающей подсказки. Подсказка отображается, когда курсор мыши задерживается на ссылке, после чего подсказка через некоторое время пропадает. Синтаксис следующий.

```
<a title="текст">...</a>
```

В качестве аргумента указывается любая текстовая строка. Строка должна заключаться в двойные или одинарные кавычки.

```
<head>
```

```
<title>Подсказка к ссылке</title>
```

```
</head>
```

```
<body> <p>
```

```
<a href="zoo.html" title="Рисунки различных животных и не только...">Рисунки</a></p> </body> </html>
```

**Замечание:** Посмотрите в разных браузерах – Opera, FireFox.

## *Ссылка на адрес электронной почты*

Создание ссылки на адрес электронной почты делается как и ссылка на Веб-страницу. Только вместо URL указывается **mailto:** адрес электронной почты.

В параметре href тега <a> вначале пишется ключевое слово mailto, затем через двоеточие желаемый почтовый адрес. Подобная ссылка ничем не отличается от ссылки на Веб-страницу, но при нажатии на нее запускается почтовая программа, установленная по умолчанию.

Поэтому в названии ссылки следует указывать, что она имеет отношение к электронной почте, чтобы читатели понимали, к чему приведет нажатие на нее.

Можно также автоматически добавить тему сообщения, присоединив к адресу электронной почты через символ вопроса (?) параметр subject="тема сообщения",

```
<body>
```

```
<p><a href="mailto:ask@htmlbook.ru?subject=Вопрос по HTML">Задавайте вопросы по электронной почте</a>
```

```
</body>
```

# Якоря

Якорем называется закладка с уникальным именем на определенном месте Веб-страницы.

Она позволяет создать переход к ней по ссылке.

Якоря удобно применять в документах большого объема, чтобы можно было быстро переходить к нужному разделу.

Для создания якоря следует вначале сделать закладку в соответствующем месте и дать ей имя при помощи параметра `name` тега `<a>`.

В качестве значения `href` для перехода к этому якорю используется имя закладки с символом решетки (`#`) впереди.





## *Создание якоря*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
<title>Быстрый переход внутри документа</title>
</head>
<body>
  <p><a name="top"></a></p>
  <p>...</p>
  <p>
    <a href="#top">Наверх</a>
  </p>
</body>
</html>
```

## Якоря

Между тегами `<a name="top">` и `</a>` текст не обязателен, так как требуется лишь указать местоположение перехода по ссылке, находящейся внизу страницы.

Имя ссылки на якорь начинается с символа `#`, после чего идет имя якоря. Оно выбирается любое, соответствующее тематике.

Главное, чтобы значения параметров `name` и `href` совпадали

(символ решетки не в счет). Ссылку можно также сделать на закладку, находящуюся в другой Веб-странице и даже другом сайте.

Для этого в параметре `href` тега `<a>` надо указать адрес документа и в конце добавить символ решетки `#` и имя закладки.

## *Ссылка на закладку из другой веб-страницы*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
<title>Якорь в другом документе</title>
</head>
<body>
  <p><a href="text.html#bottom">Перейти к нижней части текста</a>
</p> </body>
</html>
```

**Замечание.** Если на Веб-странице содержится ссылка на якорь, а самого якоря нет, то никакой ошибки не возникнет.

# Изображения

Добавление изображения происходит в два этапа: вначале готовится графический файл желаемого размера, затем он добавляется на страницу через тег `<IMG>`. Сам HTML предназначен только для того, чтобы отобразить требуемую картинку, при этом никак ее не меняя.

При подготовке изображений следует учесть несколько моментов.

Поскольку Веб-страница загружается по сети, существенным фактором выступает объем графического файла, встроенного в документ. Чем он меньше, тем быстрее отобразится изображение.

Размер картинки необходимо ограничить по ширине, например, установить не более 800 пикселей. Иначе изображение целиком не поместится в окне браузера, и появятся полосы прокрутки.

# *Изображения*

Широкое распространение для веб-графики получили два формата — GIF и JPEG.

Их многофункциональность, универсальность, небольшой объем исходных файлов при достаточном для сайта качестве, сослужили им положительную службу, фактически определив их как стандарт Веб-изображений.

Есть еще формат PNG, который также поддерживается браузерами при добавлении изображений и существует в двух ипостасях — PNG-8 и PNG-24.

Однако популярность PNG сильно уступает признанию форматам GIF и JPEG.

# Формат GIF

Graphics Interchange Format — формат графических файлов, широко применяемый при создании сайтов. GIF использует 8-битовый цвет и эффективно сжимает сплошные цветные области, при этом сохраняя детали изображения.

## Особенности

- Количество цветов в изображении может быть от 2 до 256, но это могут быть любые цвета из 24-битной палитры.
- Файл в формате GIF может содержать прозрачные участки. Если используется отличный от белого цвета фон, он будет проглядывать сквозь «дыры» в изображении.
- Поддерживает покадровую смену изображений, что делает формат популярным для создания баннеров и простой анимации.
- Использует свободный от потерь метод сжатия

## Область применения

Текст, логотипы, иллюстрации с четкими краями, анимированные рисунки, изображения с прозрачными участками, баннеры.

# Формат JPEG

Joint Photographic Experts Group — популярный формат графических файлов. Используется при создании сайтов и хранения изображений. Поддерживает 24-битовый цвет и сохраняет яркость и оттенки цветов в фотографиях неизменными. Данный формат называют сжатием с потерями, поскольку алгоритм JPEG выборочно отвергает данные. Метод сжатия может внести искажения в рисунок, особенно содержащий текст, мелкие детали или четкие края. Формат JPEG не поддерживает прозрачность. Когда вы сохраняете фотографию в этом формате, прозрачные пиксели заполняются определенным цветом.

## Особенности

- Количество цветов в изображении — около 16 миллионов, что вполне достаточно для сохранения фотографического качества изображения.
- Основная характеристика формата — качество, позволяющее управлять конечным размером файла.
- Поддерживает технологию, так называемый прогрессивный JPEG, в котором версия рисунка с низким разрешением появляется в окне просмотра до полной загрузки самого изображения.

**Область применения.** Используется для фотографий. Мало подходит для рисунков с прозрачными участками, мелкими деталями или текст.

## *Формат PNG-8*

Portable Network Graphics — формат по своему действию аналогичен GIF. По заверению разработчиков использует улучшенный формат сжатия данных, но как показывает практика, это не всегда так.

### **Особенности**

Использует 8-битную палитру (256 цветов) в изображении, за что и получил в своем названии цифру восемь. При этом можно выбирать, сколько цветов будет сохраняться в файле — от 2 до 256.

В отличие от GIF, не отображает анимацию ни в каком виде.

### **Область применения**

Текст, логотипы, иллюстрации с четкими краями, изображения с градиентной прозрачностью.



## *Формат PNG-24*

Аналогичен PNG-8, но использует 24-битную палитру цвета. Как и формат JPEG, сохраняет яркость и оттенки цветов в фотографиях. Подобно GIF и формату PNG-8, сохраняет детали изображения, в линейных рисунках, логотипах, или иллюстрациях

### **Особенности**

Использует примерно 16,7 млн. цветов в файле, из-за чего этот формат применяется для полноцветных изображений.

Поддерживает многоуровневую прозрачность. Что позволяет создавать плавный переход от прозрачной области изображения к цветной, так называемый градиент.

Из-за того, что используемый алгоритм сжатия сохраняет все цвета и пиксели в изображении неизменными, если сравнивать с другими форматами, то у PNG-24 конечный объем графического файла получается наибольшим.

### **Область применения**

Фотографии, рисунки, содержащие прозрачные участки, рисунки с большим количеством цветов и четкими краями изображений.

## *Добавление рисунка*

Для добавления изображения на веб-страницу используется тег `<img>`, параметр `src` которого определяет адрес графического файла. Общий синтаксис добавления изображения будет следующий.

```

```

URL (Universal Resource Locator, универсальный указатель ресурсов) представляет собой путь к графическому файлу. Для его указания можно использовать как абсолютный, так и относительный адрес.

Рассмотрим несколько разных путей к графическому файлу для размещения его на Веб-странице.

Возьмем файл с рисунком, который называется `sample.gif` и хранится в папке `images` корня сайта.

## *Добавление рисунка*

- Если в начале адреса стоит слэш (символ /), это значит, что отсчет идет от корня сайта. Например, адрес сайта — `http://baklan.narod.ru`, значит, написав путь к изображению как `/images/bird.jpg`, мы говорим серверу, что показать следует файл <http://baklan.narod.ru/images/bird.jpg>. Подобные ссылки со слэшем впереди работают только на веб-сервере. На локальном компьютере они действовать не будут.
- Если перед адресом добавляется упоминание протокола `http` (`http://`), то речь идет об абсолютной ссылке. Изображение всегда будет загружаться с указанного адреса в Интернете, даже при сохранении Веб-страницы на локальный компьютер.
- Двоеточие со слэшем (`../`) в начале адреса говорит о том, что и рисунок и веб-страница находятся в разных папках, которые размещены на одном уровне. Ниже показан файл `index.html`, в который требуется поместить рисунок `pic.gif`. Относительный путь к изображению из `index.html` будет `../images/pic.gif`.

## Добавление рисунка

Возможны случаи хранения файлов, что обращение из одного файла к другому превращается в набор двоеточий, например:  
../../../../images/pic.gif.

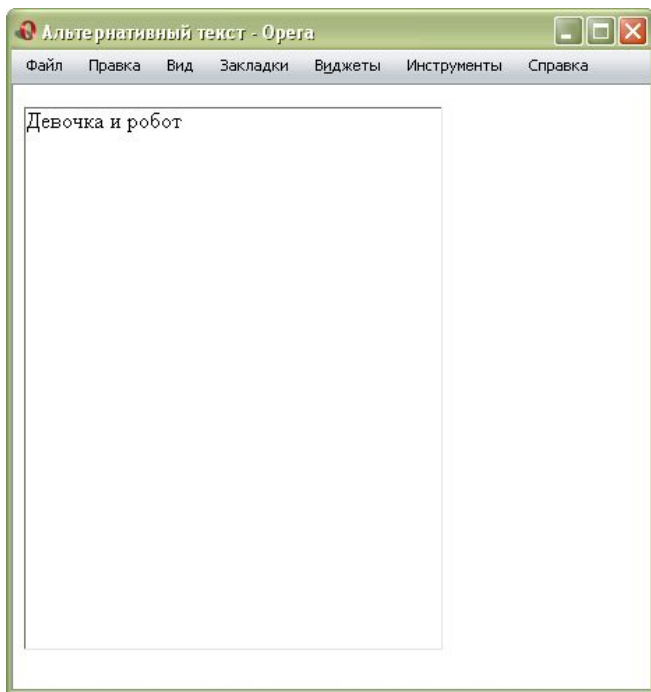


- Имя папки в начале пути, без всяких слэшей и двоеточий, сообщает, что и текущий файл и папка с изображением находятся на одном уровне. Как показано на рисунке относительный путь к рисунку pic.gif из файла index.html будет images/pic.gif.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
  <title>Добавление рисунков</title>
</head>
<body>
  <p></p>
  <p></p>
  <p></p>
</body>
</html>
```

## *Альтернативный текст*

Альтернативный текст позволяет получить текстовую информацию о рисунке при отключенном в браузере показе картинок или во время их загрузки. Такой текст появляется раньше самого изображения и дает представление об его содержании. Затем зарезервированное пустое поле заменяется картинкой.



## *Добавление рисунка*

Некоторые браузеры также отображают альтернативный текст в виде подсказки, появляющейся при наведении курсора мыши на изображение.



Для создания альтернативного текста используется параметр alt тега `<img>`

```
<a href="index.html">
```

```
  
```

```
</a>
```

## *Добавление альтернативного текста*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
    <title>Альтернативный текст</title>
  </head>
  <body>
    <p><a href="index.html"></a></p>
  </body>
</html>
```

Текст в параметре alt обязательно должен быть взят в кавычки.



## *Всплывающая подсказка*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
    <title>Параметр title </title>
  </head>
  <body>
    <p><a href="index.html"></a></p>
  </body>
</html>
```

## *Изменение размеров рисунка*

Для изменения размеров рисунка средствами HTML у тега <img> предусмотрены параметры width (ширина) и height (высота). В качестве значения используются пикселы, при этом аргументы должны совпадать с физическими размерами картинка. Так, ниже показано изображение, которое имеет размеры 100x111 пикселей.

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
  <title>Размеры изображения</title>
</head>
<body>
  <p></p>
</body>
</html>
```



## *Таблицы*

Благодаря универсальности таблиц, большому числу параметров, управляющих их видом, таблицы стали определенным стандартом для верстки Веб-страниц. Таблица с невидимой границей представляет собой словно модульную сетку. В ее блоках которой удобно размещать элементы Веб-страницы.

В настоящее время на смену таблицам при верстке сайтов пришли слои. Пока слои используются редко, но наметилась тенденция .

**Таблицы** применяются для размещения табличных данных, а **слои** — для верстки и оформления.

## *Создание таблиц*

Таблица состоит из строк и столбцов ячеек. Они могут содержать текст и рисунки.

Таблицы используются для упорядочения и представления данных.

С их помощью удобно верстать макеты страниц, располагая нужным образом фрагменты текста и изображений.

Для добавления таблицы на Веб-страницу используется тег `<table>`. Он служит контейнером для элементов, определяющих содержимое таблицы.

Таблица состоит из строк и ячеек. Они задаются с помощью тегов `<tr>` и `<td>`. В таблице должна быть хотя бы одна ячейка. Можно вместо тега `<td>` использовать тег `<th>`. Текст в ячейке, оформленной с помощью тега `<th>`, отображается браузером шрифтом жирного начертания и выравнивается по центру ячейки. В остальном, различия между ячейками, созданными через теги `<td>` и `<th>` нет.

## *Создание таблиц*

Таблица состоит из строк и столбцов ячеек. Они могут содержать текст и рисунки.

Таблицы используются для упорядочения и представления данных.

С их помощью удобно верстать макеты страниц, располагая нужным образом фрагменты текста и изображений.

Для добавления таблицы на Веб-страницу используется тег `<table>`. Он служит контейнером для элементов, определяющих содержимое таблицы.

Таблица состоит из строк и ячеек. Они задаются с помощью тегов `<tr>` и `<td>`. В таблице должна быть хотя бы одна ячейка. Можно вместо тега `<td>` использовать тег `<th>`. Текст в ячейке, оформленной с помощью тега `<th>`, отображается браузером шрифтом жирного начертания и выравнивается по центру ячейки. В остальном, различия между ячейками, созданными через теги `<td>` и `<th>` нет. (12.1.htm).

## *Параметры тега <TABLE>*

Гибкость и универсальность таблиц обеспечивается обилием параметров тегов `<table>`, `<tr>` и `<td>`.

### **ALIGN**

Задаёт выравнивание таблицы по краю окна браузера.

Допустимые значения:

**left** — выравнивание таблицы по левому краю,

**center** — по центру и

**right** — по правому краю.

Когда используются аргументы **left** и **right**, текст начинает обтекать таблицу сбоку и снизу.

### **BGCOLOR**

Устанавливает цвет фона таблицы.

### **BORDER**

Устанавливает толщину границ в пикселах. Границы отображаются вокруг таблицы и между ячейками.

## *Параметры тега <TABLE>*

### **CELLPADDING**

Определяет расстояние между границей ячейки и ее содержимым. Он добавляет пустое пространство к ячейке, увеличивая ее размеры. Без **cellpadding** текст в таблице «налипает» на рамку и снижает восприятие. Добавление **cellpadding** улучшает читабельность текста. Этот параметр не имеет особого значения, но помогает, когда требуется установить пустой промежуток между ячейками.

### **CELLSPACING**

Задаёт расстояние между внешними границами ячеек. Если установлен параметр **border**, толщина границы принимается в расчет и входит в общее значение.

Параметр **COLS** указывает количество столбцов в таблице. Он помогает браузеру в подготовке к ее отображению. Без него таблица будет показана тогда, когда все ее содержимое будет загружено в браузер и проанализировано.

## *Параметры тега <TABLE>*

**RULES.** Сообщает браузеру, где отображать границы между ячейками. По умолчанию рамка рисуется вокруг каждой ячейки, образуя тем самым сетку. В дополнение можно указать отображать линии между колонками (значение **cols**), строками (**rows**) или группами (**groups**), которые определяются наличием тегов **<thead>**, **<tfoot>**, **<tbody>**, **<colgroup>** или **<col>**. Толщина границы указывается с помощью параметра **border**.

**WIDTH.** Задаёт ширину таблицы. Если общая ширина содержимого превышает указанную ширину таблицы, то браузер будет пытаться «втиснуться» в заданные размеры за счёт форматирования текста. Если это невозможно (в таблице находятся изображения), параметр **width** будет проигнорирован, и новая ширина таблицы будет вычислена на основе её содержимого. Как и высота, если ширина явно не указана, то она будет вычисляться на основе содержимого таблицы.



## *Параметры тега <TABLE>*

### **Параметры тега <td>**

Каждая ячейка таблицы, задаваемая через тег <td>, тоже имеет свои параметры. Некоторые совпадают с параметрами тега <table>.

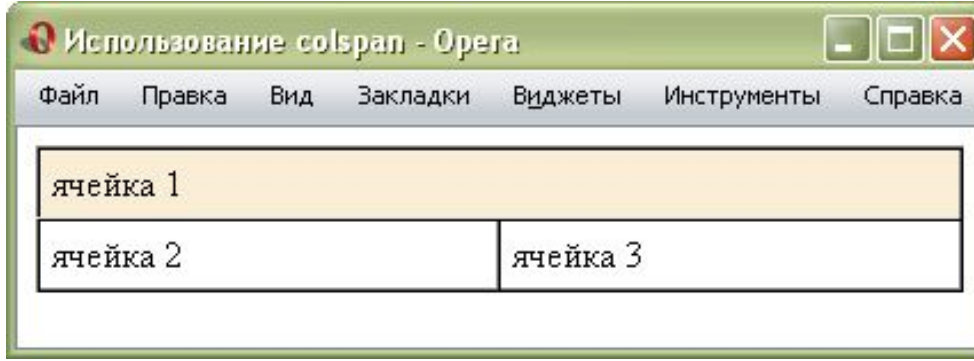
**ALIGN** задает выравнивание содержимого ячейки по горизонтали. Возможные значения: **left** — выравнивание по левому краю, **center** — по центру и **right** — по правому краю ячейки.

**BGCOLOR** Устанавливает цвет фона ячейки. Используя этот параметр совместно с атрибутом **bgcolor** тега <table> можно получить разнообразные цветовые эффекты в таблице.

### **COLSPAN**

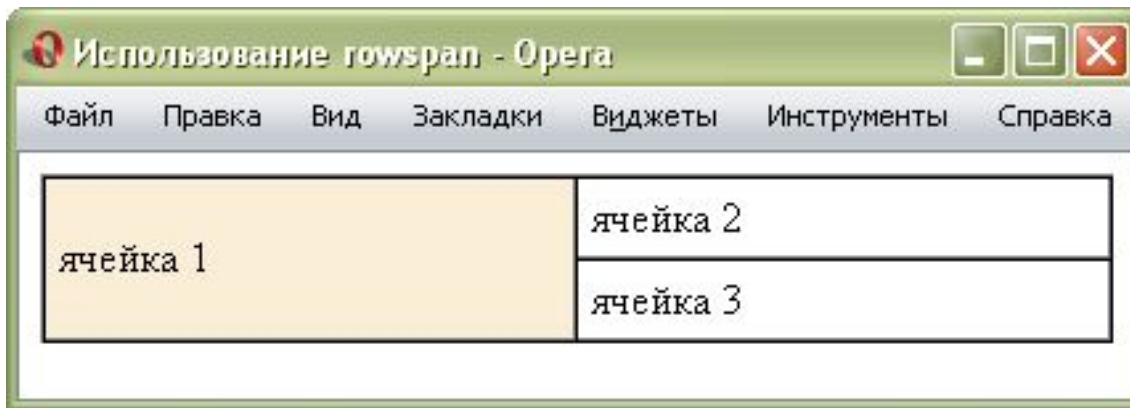
Устанавливает число ячеек, которые должны быть объединены по горизонтали. Этот параметр имеет смысл для таблиц, состоящих из нескольких строк.

# Параметры тега <TABLE>



Здесь содержатся две строки и две колонки. верхние горизонтальные ячейки объединены с помощью параметра `colspan`.

На нижнем рисунке применяется вертикальное объединение ячеек



## *Параметры тега <TABLE>*

### **HEIGHT**

Браузер сам устанавливает высоту таблицы и ее ячеек исходя из их содержимого. Однако при использовании параметра **height** высота ячеек будет изменена. Здесь возможны два варианта. Если значение **height** меньше, чем содержимое ячейки, то этот параметр будет проигнорирован. Если высота ячейки, превышает ее содержимое, добавляется пустое пространство по вертикали.

### **ROWSPAN**

Устанавливает число ячеек, которые должны быть объединены по вертикали. Этот параметр имеет смысл для таблиц, состоящих из нескольких строк.

### **WIDTH**

Задаёт ширину ячейки. Суммарное значение ширины всех ячеек может превышать общую ширину таблицы только в том случае, если содержимое ячейки превышает указанную ширину.

## *Параметры тега <TABLE>*

### **VALIGN**

Устанавливает вертикальное выравнивание содержимого ячейки. По умолчанию содержимое ячейки располагается по ее вертикали в центре. Возможные значения:

**top** — выравнивание по верхнему краю строки,

**middle** — выравнивание по середине,

**bottom** — выравнивание по нижнему краю,

**baseline** — выравнивание по базовой линии, при этом происходит привязка содержимого ячейки к одной линии.

# Параметры тега <TABLE>

## Особенности таблиц

У каждого параметра таблицы есть свое значение установленное по умолчанию. Это означает, что если какой-то атрибут пропущен, то неявно он все равно присутствует, и с некоторым значением. Из-за этого вид таблицы может оказаться совсем другим, нежели предполагал разработчик. Чтобы знать, что можно ожидать от таблиц, нужно знать их явные и неявные особенности.

- Можно одну таблицу помещать внутрь ячейки другой таблицы. Это требуется для представления сложных данных или тогда, когда одна таблица выступает в роли модульной сетки, а вторая, внутри нее, как обычная таблица.
- Размеры таблицы изначально не установлены и вычисляются на основе содержимого ячеек. Например, общая ширина определяется автоматически исходя из суммарной ширины содержимого ячеек плюс ширина границ между ячейками.

# Параметры тега <TABLE>

## Особенности таблиц

Поля вокруг содержимого, устанавливаемые через параметр `cellpadding` и расстояние между ячейками, которые определяются значением `cellspacing`.

- Если для таблицы задана ее ширина в процентах или пикселах, то содержимое таблицы подстраивается под указанные размеры. Браузер автоматически добавляет переносы строк в текст, чтобы он полностью поместился в ячейку, и при этом ширина таблицы осталась без изменений. Иногда ширину содержимого ячейки невозможно изменить, когда работают с рисунками. Тогда ширина таблицы увеличивается, несмотря на указанные размеры.
- Пока таблица не загрузится полностью, ее содержимое не начнет отображаться. Браузер, прежде чем показать содержимое таблицы, должен вычислить необходимые размеры ячеек, их ширину и высоту. Но для этого необходимо знать, что в этих ячейках находится. Поэтому браузер и ожидает, пока загрузится все, что находится в ячейках, и только потом отображает таблицу.

# Параметры тега <TABLE>

## Выравнивание таблиц

Чтобы выровнять таблицы по центру веб-страницы или по одному из ее краев используют параметр `align` тега `<table>`. Результат будет заметен только в том случае, если ширина таблицы не занимает всю доступную область, т.е. меньше, чем 100%. На самом деле `align` не только устанавливает выравнивание, но и заставляет текст обтекать таблицу с других сторон аналогично поведению тега `<img>`.

Выравнивание таблицы

Для задания выравнивания таблицы по центру веб-страницы или по одному из ее краев предназначен параметр `align` тега `table`. Результат будет заметен только в том случае, если ширина таблицы не занимает всю доступную область, другими словами, меньше, чем 100%. На самом деле `align` не только устанавливает выравнивание, но и заставляет текст обтекать таблицу с других сторон аналогично поведению соответствующего тега. По умолчанию таблица формируется в виде сетки. При этом в каждой строке таблицы содержится одинаковое количество ячеек. Такой вариант вполне подходит для формирования простых таблиц, но совершенно не годится для тех случаев, когда предстоит сделать сложную таблицу. В подобных ситуациях применяются два основных метода: объединение ячеек и вложенные таблицы.

Содержимое  
таблицы

# Параметры тега <TABLE>

## Выравнивание таблиц

По умолчанию таблица формируется в виде сетки, при этом в каждой строке таблицы содержится одинаковое количество ячеек. Такой вариант вполне подходит для формирования простых таблиц, но совершенно не годится для тех случаев, когда предстоит сделать сложную таблицу. В подобных ситуациях применяются два основных метода:

***объединение ячеек и  
вложенные таблицы.***



## Параметры тега <TABLE>

### Объединение ячеек

Для объединения двух и более ячеек в одну используется параметр `colspan` тега `<td>`. Он устанавливает, сколько ячеек следует объединить по горизонтали. Аналогично работает и параметр `rowspan`, с тем лишь отличием, что объединяет ячейки по вертикали. Перед добавлением атрибутов проверьте число ячеек в каждой строке, чтобы не возникло ошибок. Так, `<td colspan="3">` заменяет три ячейки, поэтому в следующей строке должно быть три тега `<td>` или конструкция вида `<td colspan="2">...</td><td>...</td>`. Если число ячеек в каждой строке не будет совпадать, появятся пустые фантомные ячейки. В листинге приведен хотя и валидный, но неверный код, в котором как раз проявляется подобная ошибка.

## *Параметры тега <TABLE>*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html> <head>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  <title>Неправильное использование параметра colspan</title>
</head>
<body>
  <table border="1" cellpadding="5">
    <tr>
      <td colspan="2">Ячейка 1</td>
      <td>Ячейка 2</td>
    </tr>
    <tr>
      <td>Ячейка 3</td>
      <td>Ячейка 4</td>
    </tr>
  </table>
</body>
</html>
```

## Параметры тега <TABLE>

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html> <head>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  <title>Неправильное использование параметра colspan</title>
</head>
<body>
  <table border="1" cellpadding="5">
    <tr>
      <td colspan="2">Ячейка 1</td>
      <td>Ячейка 2</td>
    </tr>
    <tr>
      <td>Ячейка 3</td>
      <td>Ячейка 4</td>
    </tr>
  </table>
</body>
</html>
```

В первой строке примера задано три ячейки, две из них объединены с помощью параметра `colspan`, а во второй строке добавлено только две ячейки. Из-за этого возникает дополнительная ячейка, которая отображается в браузере.

# Параметры тега <TABLE>

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<title>Объединение ячеек</title>
</head>
<body>
<table border="1" cellpadding="4" cellspacing="0">
<tr>
<td rowspan="2">Браузер</td>
<th colspan="2">Internet Explorer</th>
<th colspan="3">Opera</th>
<th colspan="2">Firefox</th>
</tr>
<tr>
<th>6.0</th><th>7.0</th><th>7.0</th><th>8.0</th><th>9.0</th><th>1.0</th><th>2.0</th>
</tr>
<tr align="center">
<td>Поддерживается</td>
<td>Нет</td><td>Да</td><td>Нет</td><td>Да</td><td>Да</td><td>Да</td>
</tr>
</table>
</body>
</html>
```

Здесь показано  
правильное использова-  
ние параметров **colspan** и  
**rowspan**

## Параметры тега <TABLE>

В этой таблице установлено восемь колонок и три строки. Часть ячеек с надписями «Internet Explorer», «Opera» и «Firefox» объединены где по две, а где и по три ячейки.

В ячейке с надписью «Браузер» применено объединение по вертикали.



The screenshot shows a browser window titled "Объединение ячеек - Опера". The menu bar includes "Файл", "Правка", "Вид", "Закладки", "Виджеты", "Инструменты", and "Справка". The table below displays support for various browser versions.

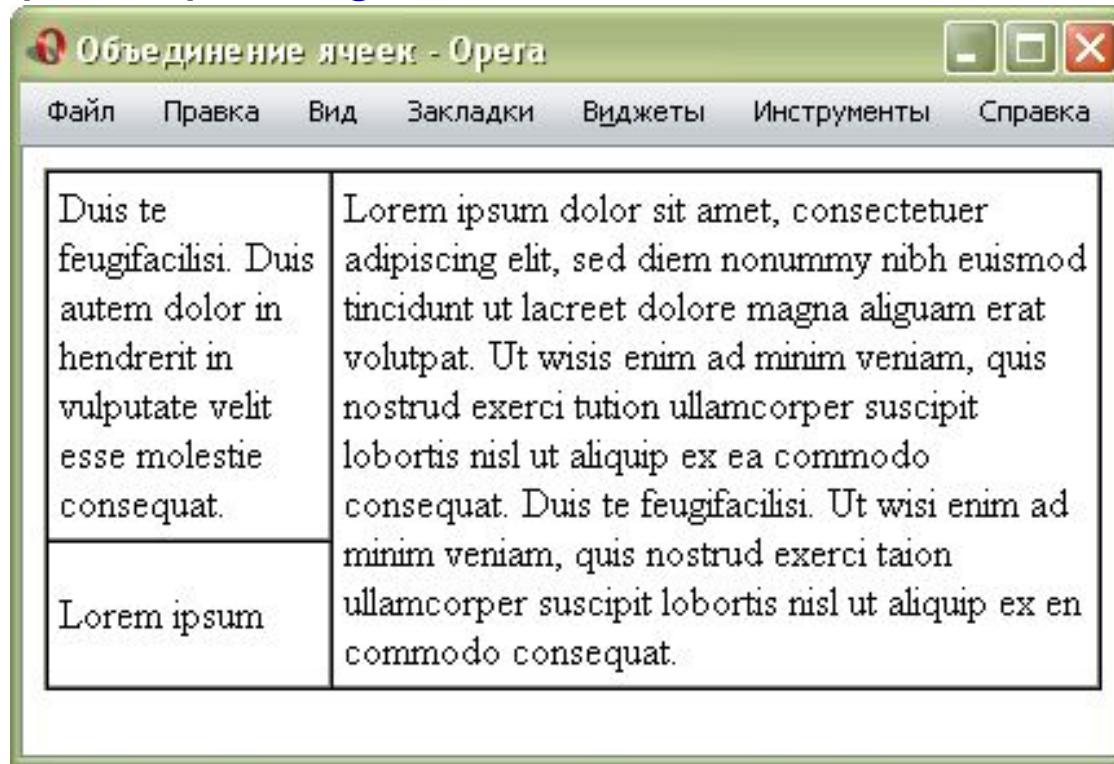
Браузер	Internet Explorer		Opera			Firefox	
	6.0	7.0	7.0	8.0	9.0	1.0	2.0
Поддерживается	Нет	Да	Нет	Да	Да	Да	Да

# Параметры тега <TABLE>

## Вложенные таблицы

Объединение ячеек имеет некоторые недостатки, поэтому этот метод создания таблиц нельзя использовать повсеместно.

Рассмотрим следующий листинг, где задается высота ячейки с помощью параметра `height`.



Объединение ячеек - Opera

Файл Правка Вид Закладки Виджеты Инструменты Справка

Duis te feugifacilisi. Duis autem dolor in hendrerit in vulputate velit esse molestie consequat.	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tution ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis te feugifacilisi. Ut wisi enim ad
Lorem ipsum	minim veniam, quis nostrud exerci taion ullamcorper suscipit lobortis nisl ut aliquip ex en commodo consequat.

## *Параметры тега <TABLE>*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html> <head>
  <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
  <title>Выравнивание таблицы</title> </head>
<body>
  <table width="50" bgcolor="#c0c0c0" cellspacing="0" cellpadding="5"
border="1" align="right">
  <tr><td>Содержимое таблицы</td></tr>
</table>
  <p>Для задания выравнивания таблицы по центру веб-страницы
или по одному из ее краев предназначен параметр align тега table. .
. . таблицы.
</p>
</body>
</html>
```

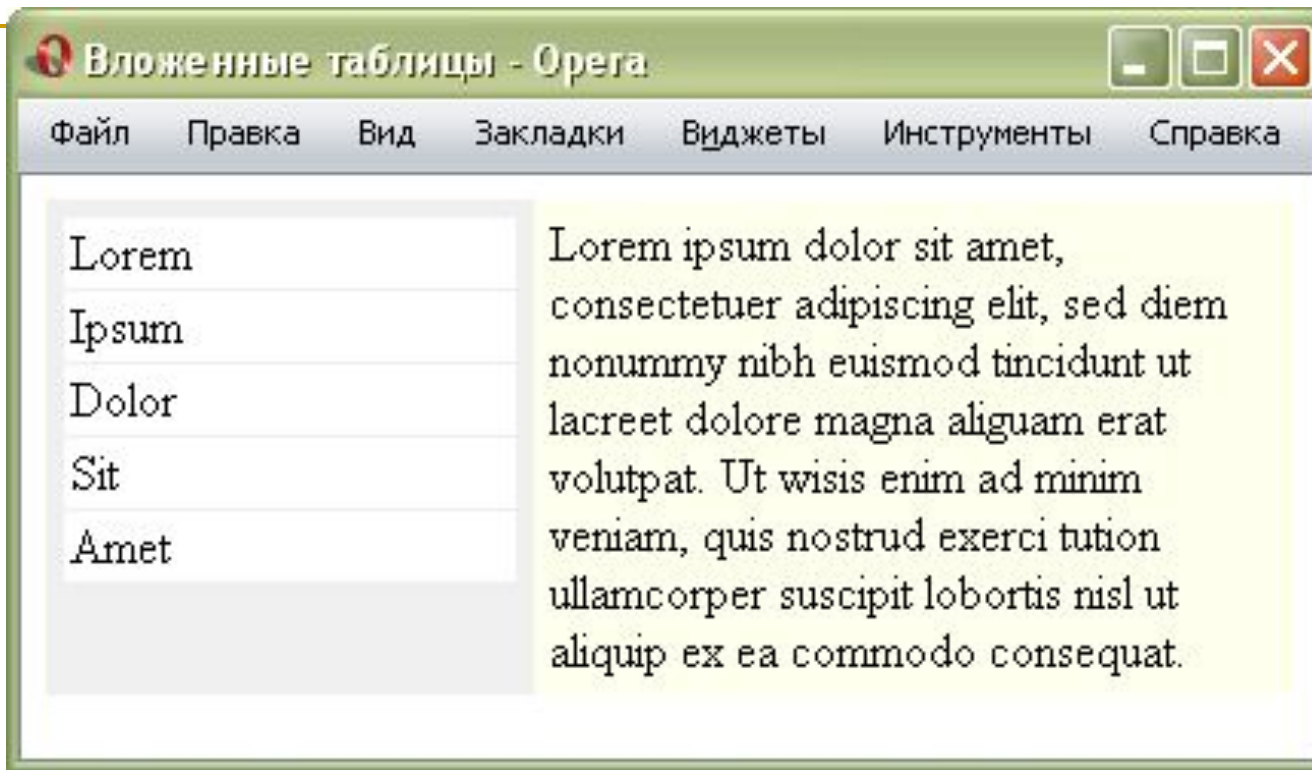
## *Параметры тега <TABLE>*

### **Вложенные таблицы**

Левая нижняя ячейка согласно коду HTML имеет высоту 40 пикселей, но поскольку высота содержимого правой колонки больше, чем содержимое левой колонки, то высота ячейки меняется. Получается, что параметр `height` в данном случае игнорируется. Некоторые браузеры могут отображать сложные таблицы с ошибками. Это может выражаться в тех таблицах, где явно устанавливается высота ячеек и их объединение по вертикали. Для упрощения верстки применяется прием с вложенными таблицами.

Идея проста — в ячейку вкладывается еще одна таблица со своими параметрами. Поскольку эти таблицы в каком-то смысле независимы, то можно создавать довольно различные конструкции. Чтобы вложенная таблица занимала всю ширину ячейки, таблице надо задать ширину 100%.





В данном макете с помощью таблицы создается две колонки, причем левая колонка имеет фиксированную ширину 150 пикселей. Чтобы создать подобие навигации, внутрь ячейки добавлена еще одна таблица с шириной 100%. Как видно из рисунка, если не задавать границы, то определить наличие таблиц по виду Веб-страницы довольно сложно. Поэтому таблицы активно применяются для верстки сложных макетов.

## *Параметры тега <TABLE>*

### **Заголовок таблицы**

При большом количестве таблиц на странице каждой из них удобно задать заголовок, содержащий название таблицы и ее описание. Для этой цели в HTML существует специальный тег `<caption>`, который устанавливает текст и его положение относительно таблицы. Проще всего размещать текст по центру таблицы сверху или снизу от нее, в остальных случаях браузеры по-разному интерпретируют параметры тега `<caption>`, из-за чего результат получается неодинаковый. По умолчанию заголовок помещается сверху таблицы по центру, его ширина не превышает ширины таблицы и в случае длинного текста он автоматически переносится на новую строку. Для изменения положения заголовка у тега `<caption>` существует параметр `align`, который может принимать следующие значения.

## Параметры тега <TABLE>

### Заголовок таблицы

**left** — выравнивает заголовок по левому краю таблицы. Текст выравнивается по левому краю таблицы.

**right** — заголовок располагается сверху таблицы и выравнивается по правому краю таблицы. Firefox отображает заголовок справа от таблицы.

**center** — заголовок располагается сверху таблицы по ее центру. Такое расположение задано в браузерах по умолчанию.

**top** — результат аналогичен действию параметра **center**, но в отличие от него входит в спецификацию HTML 4 и понимается всеми браузерами.

**bottom** — заголовок размещается внизу таблицы по ее центру. В листинге 12.7 показано, как установить заголовок сверху таблицы. Обратите внимание, что тег `<caption>` находится внутри контейнера `<table>`, это его стандартное местоположение.

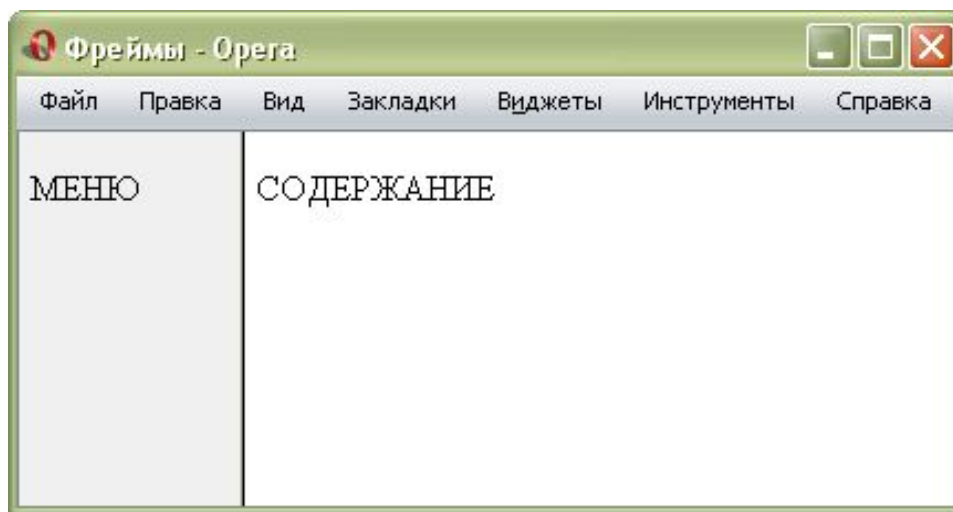
---

# *Параметры тега <TABLE>*



## Создание фреймов

Для создания фрейма используется тег `<frameset>`, который заменяет тег `<body>` в документе и применяется для разделения экрана на области. Внутри данного тега находятся теги `<frame>`, которые указывают на HTML-документ, предназначенный для загрузки в область.



Пример разделения окна браузера на два фрейма

## Создание фреймов

При использовании фреймов необходимо как минимум три HTML-файла:

первый определяет фреймовую структуру и делит окно браузера на две части, а оставшиеся два документа загружаются в заданные окна. Количество фреймов не обязательно равно двум, может быть и больше, но никак не меньше двух, иначе вообще теряется смысл применения фреймов.

[index.html](#) — определяет структуру документа,

[menu.html](#) — загружается в левый фрейм и

[content.html](#) — загружается в правый фрейм.

## Создание фреймов

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd"> <html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
<title>Фреймы</title>
</head>
<frameset cols="100,*">
<frame src="menu.html" name="MENU">
<frame src="content.html" name="CONTENT">
</frameset>
</html>
```

В первой строке кода **<!DOCTYPE>** указывают браузеру, что он имеет дело с фреймами, эта строка кода является обязательной. Контейнер **<head>** содержит типовую информацию вроде кодировки страницы и заголовка документа.



## Создание фреймов

Здесь окно браузера разбивается на две колонки с помощью параметра `cols`, левая колонка занимает 100 пикселей, а правая — оставшееся пространство, заданное символом звездочки.

Ширину или высоту фреймов можно также задавать в процентном отношении, наподобие таблиц.

В теге `<frame>` задается имя HTML-файла, загружаемого в указанную область с помощью параметра `src`.

В левое окно будет загружен файл, названный `menu.html`, а в правое — `content.html`.

Каждому фрейму желательно задать его уникальное имя, чтобы документы можно было загружать в указанное окно с помощью параметра `name`.

## *Файл меню.html*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
    <title>Навигация по сайту</title>
  </head>
  <body style="background: #f0f0f0">
    <p>МЕНЮ</p>
  </body>
</html>
```

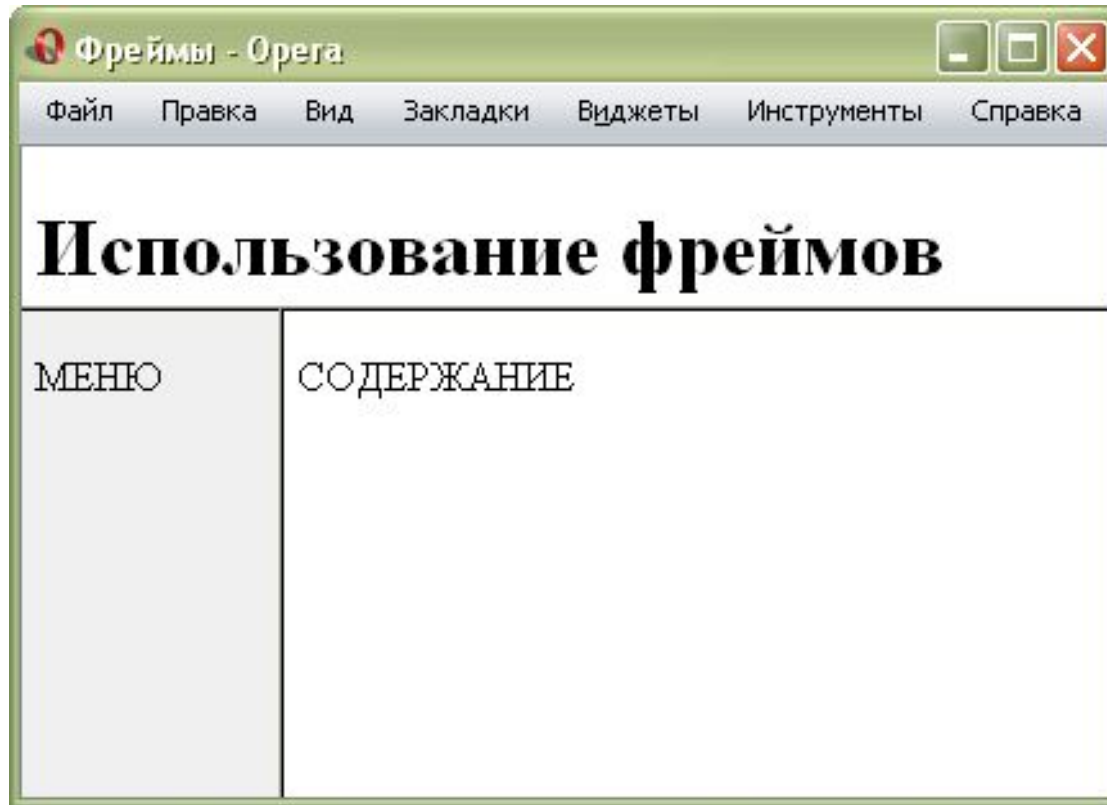
## *Файл content.html*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
    <title>Содержание сайта</title>
  </head>
  <body>
    <p>СОДЕРЖАНИЕ</p>
  </body>
</html>
```

## Файл с тремя фреймами

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
    <title>Фреймы</title>
  </head>
  <frameset rows="25%,75%">
    <frame src="top.html" name="TOP" scrolling="no" noresize>
    <frameset cols="100,*">
      <frame src="menu.html" name="MENU">
      <frame src="content.html" name="CONTENT">
    </frameset>
  </frameset>
</html>
```

## Файл с тремя фреймами



Здесь снова используется тег `<frameset>`, но два раза, причем один тег вкладывается в другой.

Горизонтальное разбиение создается через параметр `rows`, где для разнообразия применяется процентная запись.

## Файл с тремя фреймами

Из примера видно что, контейнер `<frameset>` с параметром `rows` вначале создает два горизонтальных фрейма, но вместо второго фрейма подставляется еще один `<frameset>`.

Чтобы не появилась вертикальная полоса прокрутки, и пользователь не мог самостоятельно изменить размер верхнего фрейма, добавлены параметры `scrolling="no"` и `noresize`.

### Ссылки во фреймах

В обычном HTML-документе при переходе по ссылке, в окне браузера текущий документ заменяется новым.

При использовании фреймов схема загрузки документов отличается от стандартной. Основное отличие — возможность загружать документ в выбранный фрейм из другого. Для этой цели используется параметр `target` тега `<a>`. В качестве значения используется имя фрейма, в который будет загружаться документ, указанный параметром `name`.

## *Ссылка на другой фрейм*

Файл index.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
    <title>Фреймы</title>
  </head>
  <frameset cols="100,*">
    <frame src="menu.html" name="MENU">
    <frame src="content.html" name="CONTENT">
  </frameset>
</html>
```

## *Ссылка на другой фрейм*

### Файл menu.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  
<html>  
  <head>  
    <meta http-equiv="Content-Type" content="text/html;  
charset=windows-1251">  
    <title>Навигация по сайту</title>  
  </head>  
  <body>  
    <p><a href="text.html" target="CONTENT">Содержание</a></p>  
  </body>  
</html>
```



## *Ссылка на другой фрейм*

Здесь фрейму присваивается имя CONTENT.

Чтобы документ загружался в указанный фрейм, используется параметр `target="CONTENT"`.

Имя фрейма должно начинаться на цифру или латинскую букву.

В качестве зарезервированных имен используются следующие:

`_blank` — загружает документ в новое окно;

`_self` — загружает документ в текущий фрейм;

`_parent` — загружает документ во фрейм, занимаемый родителем, если фрейма-родителя нет параметр действует также, как `_top`;

`_top` — отменяет все фреймы и загружает документ в полное окно браузера.

## Границы между фреймами

Граница между фреймами отображается по умолчанию и, как правило, в виде трехмерной линии.

Чтобы ее скрыть используется параметр `frameborder` тега `<frameset>` со значением `0`.

Однако в браузере Opera граница хоть и становится в этом случае бледной, все же остается.

Для этого браузера требуется добавить параметр `framespacing="0"`.

Таким образом, комбинируя разные параметры тега `<frameset>`, получим универсальный код, который работает во всех браузерах.

Если граница между фреймами нужна, в браузере она рисуется по умолчанию, без задания каких-либо параметров. Можно, также, задать цвет рамки с помощью параметра `bordercolor`, который может применяться в тегах `<frameset>` и `<frame>`. Цвет указывается по его названию или шестнадцатеричному значению, а толщина линии управляется параметром `border`.

## *Убираем границу между фреймами*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
    <title>Фреймы</title>
  </head>
  <frameset cols="100,*" frameborder="0" framespacing="0">
    <frame src="menu.html" name="MENU">
    <frame src="content.html" name="CONTENT">
  </frameset>
</html>
```

## *Изменение цвета границ*

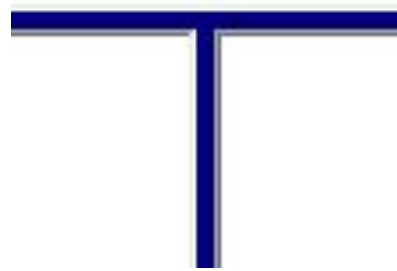
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
    <title>Фреймы</title>
  </head>
  <frameset cols="100,*" bordercolor="#000080" border="5">
    <frame src="menu.html" name="MENU">
    <frame src="content.html" name="CONTENT">
  </frameset>
</html>
```

## Границы между фреймами

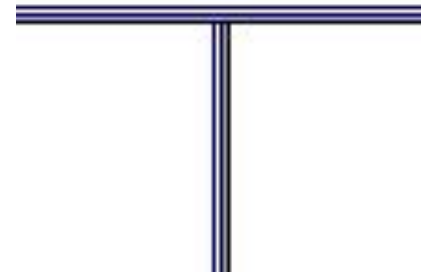
Параметры `bordercolor` и `border` тега `<frameset>` также не являются валидными и не признаются спецификацией HTML. В данном примере линия между фреймами задается синего цвета толщиной пять пикселей. Линии различаются по своему виду в разных браузерах, несмотря на одинаковые параметры



Opera



Internet Explorer



Firefox

Вид границы между фреймами в разных браузерах

## Изменение размеров фреймов

По умолчанию размеры фреймов можно изменять с помощью курсора мыши, наведя его на границу между фреймами. Для блокировки возможности изменения пользователем размера фреймов следует воспользоваться параметром `noresize` тега `<frame>`

Параметр `noresize` не требует никаких значений и используется сам по себе. Для случая двух фреймов этот параметр можно указать лишь в одном месте. Естественно, если у одного фрейма нельзя изменять размеры, то у близлежащего к нему размеры тоже меняться не будут.

Параметр `noresize` не требует никаких значений и используется сам по себе. Для случая двух фреймов этот параметр можно указать лишь в одном месте.

Поскольку у одного фрейма нельзя изменять размеры, то у близлежащего к нему размеры тоже меняться не будут.

## *Запрет на изменение размера фреймов*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
    <title>Фреймы</title>
  </head>
  <frameset cols="100,*">
    <frame src="menu.html" name="MENU" noresize>
    <frame src="content.html" name="CONTENT">
  </frameset>
</html>
```

## *Полосы прокрутки*

Если содержимое фрейма не помещается в отведенное окно, автоматически появляются полосы прокрутки для просмотра информации. В некоторых случаях полосы прокрутки нарушают дизайн Веб-страницы, поэтому от них можно отказаться.

Для управления отображением полос прокрутки используется параметр **scrolling** тега `<frame>`.

Он может принимать два основных значения:

**yes** — всегда вызывает появление полос прокрутки, независимо от объема информации и **no** — запрещает их появление.

Если параметр **scrolling** не указан, то полосы прокрутки добавляются браузером только по необходимости, в том случае, когда содержимое фрейма превышает его видимую часть.

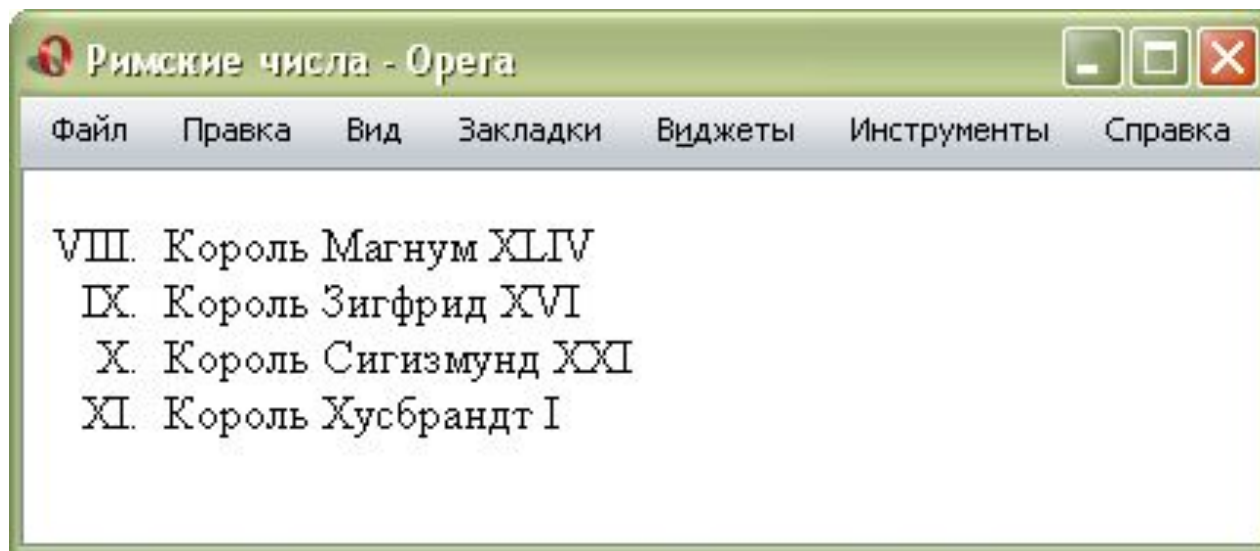
При выключенных полосах прокрутки, если информация не помещается в окно фрейма, просмотреть ее будет сложно.

Поэтому параметр **scrolling="no"** следует использовать осторожно.



## Плавающие фреймы

Это - фрейм, который можно добавлять в любое место Веб-страницы. Другое его название — встроенный фрейм. Он называется так из-за своей особенности встраиваться прямо в тело Веб-страницы.



Плавающий фрейм на Веб-странице.

В него можно загружать HTML-документ и прокручивать его содержимое независимо от остального материала на странице. Размеры фрейма устанавливаются самостоятельно согласно дизайну сайта или собственным предпочтениям.

## *Плавающие фреймы*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  
<html>  
  <head>  
    <meta http-equiv="content-type" content="text/html;  
charset=windows-1251">  
    <title>Плавающий фрейм</title>  
  </head>  
  <body>  
    <p><a href="rgb.html" target="color">RGB</a> |  
      <a href="cmyk.html" target="color">CMYK</a> |  
      <a href="hsb.html" target="color">HSB</a></p>  
    <p><iframe src="model.html" name="color" width="100%"  
height="300"></iframe></p>  
  </body>  
</html>
```

## Плавающие фреймы

Здесь ширина и высота фрейма устанавливается через параметры `width` и `height`. Сам загружаемый во фрейм файл называется `rgb.html`.

Если содержимое не помещается целиком в отведенную область, появляются полосы прокрутки.

Другое удобство плавающих фреймов состоит в том, что в него можно загружать документы по ссылке.

Для этого требуется задать имя фрейма через параметр `name`, а в теге `<a>` указать это же имя в параметре `target`.

Мы добавили несколько ссылок. Они открываются во фрейме с именем `color`.





---

*Лекция окончена!*

*Благодарю за внимание!*

*Ваши вопросы.*

*Молородов Юрий Иванович*  
*yuto@ict.nsc.ru*

*[ftp://pine.ict.nsc.ru/pub/tmp/SibGUTI\\_Lec](ftp://pine.ict.nsc.ru/pub/tmp/SibGUTI_Lec)*