

# Создание веб-сайтов

§ 22. Веб-сайты и веб-страницы

§ 23. Текстовые веб-страницы

§ 24. Оформление документов

§ 25. Рисунки, звук, видео

§ 26. Таблицы

§ 27. Блоки

§ 28. XML § 28. XML и § 28. XML и XHTML

§ 29. Динамический § 29. Динамический HTML

§ 30. Размещение веб-сайтов

# Создание веб-сайтов

## § 22. Веб-сайты и веб-страницы

# Основные определения

---

**Гипертекст** (англ. *hypertext*) – это текст, содержащий гиперссылки.

**Гиперссылка** (англ. *hyper reference*) – это «активная» ссылка на другой объект (часть того же документа, другой документ, файл, каталог, программу и т.д.).

**Веб-страница** – это гипертекстовый документ в Интернете.

**Веб-сайт** – группа веб-страниц, которые объединены общей темой и оформлением, связаны гиперссылками.

**Веб-сервер** – это программа, которая принимает запросы по протоколу HTTP и отвечает на них – возвращает веб-страницы и дополнительные данные (рисунки, звуковые файлы, видеофайлы).

**Браузер** – это программа для просмотра веб-страниц на экране монитора.

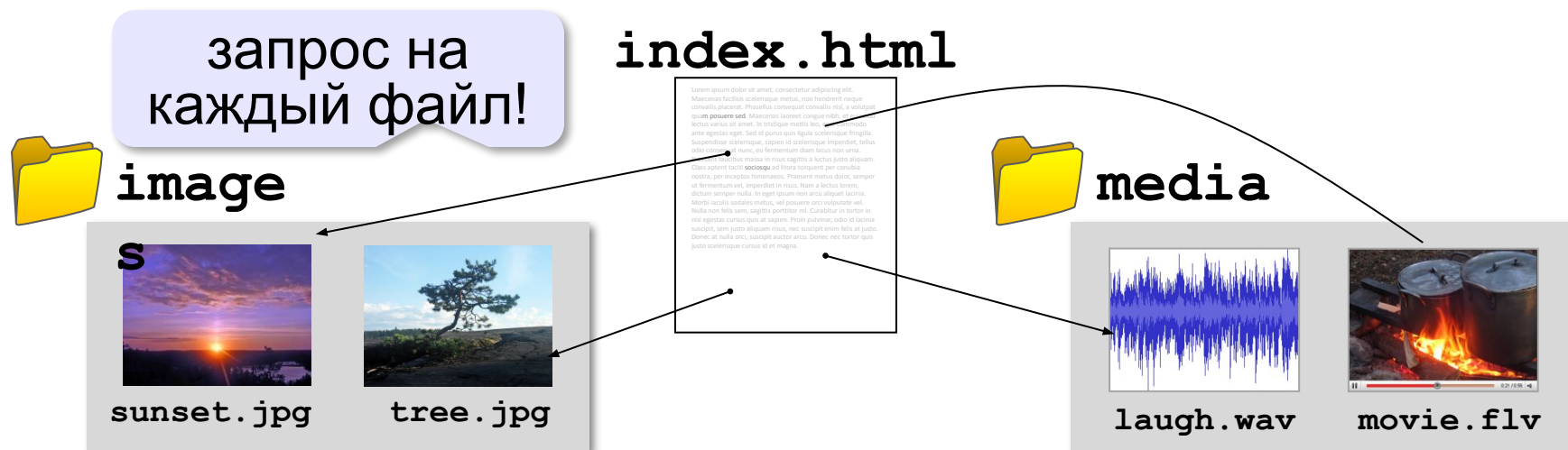
# Что такое веб-страница?

## HTML = *Hypertext Markup Language*

(язык разметки гипертекста)



HTML – это не язык программирования!



# Какие бывают веб-страницы?

---

- **статические** – готовые файлы `*.htm`, `*.html`



- быстро загружаются
- почти не нагружают сервер



- невозможна работа с базами данных (в т.ч. гостевые книги, комментарии, Веб 2.0)

- **динамические** – полностью или частично создаются на сервере в момент запроса

`*.php`, `*.asp`, `*.pl`, `*.cgi`, `*.shtml` ...



- работа с базами данных



- загружаются медленнее
- дополнительная нагрузка на сервер

# Веб-программирование

---

**Веб-программирование** – это программирование динамических сайтов в Интернете.

**Веб-приложения** – это программы, обеспечивающие работу сайта:

- **серверные** – выполняются на веб-сервере (*PHP, Python, ASP, Perl*)
- **клиентские** – выполняются в браузере пользователя

# Интерактивные страницы

**DHTML** = *Dynamic HTML*, динамический HTML.

**Скрипт** или сценарий (англ. *script*) – это программный код для автоматизации действий пользователя.

## Javascript

- замена текста, оформления, рисунков
- многоуровневые выпадающие меню
- скрывание и показ частей страницы
- проверка данных, введенные пользователем
- выполнение вычислений
- игры...

# Системы управления сайтами

**CMS** = *Content Management System*, система управления содержимым сайта.

## Функции:

- создание разделов сайта
- создание страниц
- база данных пользователей
- управление доступом
- обеспечение навигации и поиска



Не нужно знать HTML!

бесплатно!



**1С-БИТРИКС**

[1c-bitrix.ru](http://1c-bitrix.ru)



[ucoz.ru](http://ucoz.ru)



**Joomla!**

[joomla.org](http://joomla.org)



**Drupal**

[drupal.org](http://drupal.org)



**WORDPRESS**

[wordpress.org](http://wordpress.org)



# Создание веб-сайтов

## § 23. Текстовые веб-страницы

# Как создать веб-страницу?

Любой текстовый редактор (*Блокнот* и т.п.)



2×ЛКМ

**Текстовые редакторы** с подсветкой HTML-тэгов:

*Sublime Text* ([sublimetext.com](http://sublimetext.com))

*NotePad++* ([notepad-plus-plus.org](http://notepad-plus-plus.org))

*Bluefish* ([bluefish.openoffice.nl](http://bluefish.openoffice.nl))

*HEFS* ([kpolyakov.spb.ru/prog/hefs.htm](http://kpolyakov.spb.ru/prog/hefs.htm))

бесплатно!

**Редакторы WYSIWYG** = *What You See Is What You Get*

*TinyMCE* ([www.tinymce.com](http://www.tinymce.com))

*Kompozer* ([kompozer.net](http://kompozer.net))

*CKEditor* ([ckeditor.com](http://ckeditor.com))

*openWYSIWYG* ([www.openwebware.com](http://www.openwebware.com))

*BlueGriffon* ([bluegriffon.org](http://bluegriffon.org))

бесплатно!

# Первая веб-страница

**Тэг** – команда языка HTML

**контейнер**  
(парный тэг)

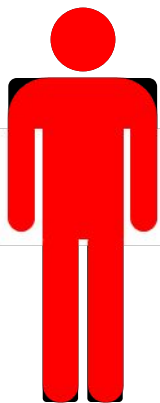
`<html>`

открывающий тэг



`</html>`

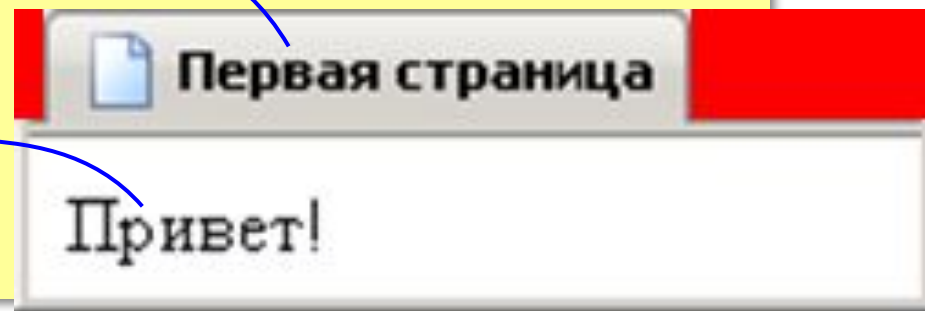
закрывающий тэг



```

<html>
  <head>
    <title>Первая страница</title>
  </head>
  <body>
    Привет!
  </body>
</html>

```



# Заголовки

заголовок  
документа



заголовок  
раздела

- <h1>** – заголовок документа
- <h2>** – заголовок раздела
- <h3>** – заголовок подраздела
- <h4>** – заголовок параграфа

```
<h1>Глава 1. Информация</h1>
```

```
<h2>1.1 Что такое информация?</h2>
```

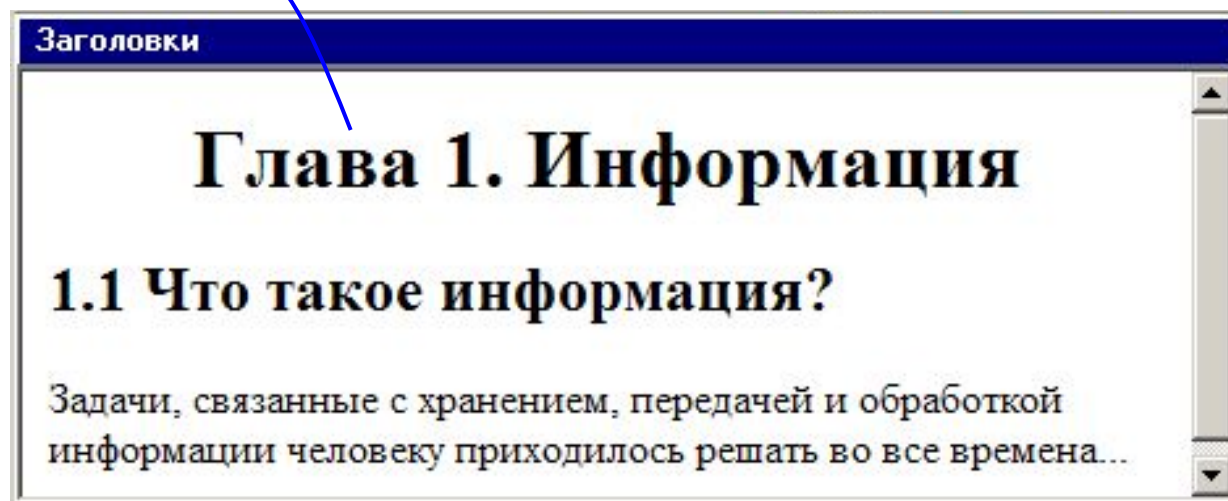
```
Задачи, связанные с хранением, передачей и  
обработкой информации человеку приходилось решать  
во все времена...
```

# Выравнивание заголовков

атрибут (свойство)

```
<h1 align="center">Глава 1. Информация</h1>
```

left,  
center,  
right



# Абзацы (параграфы)

---

И вечный бой! Покой нам только снится  
Сквозь кровь и пыль...  
Летит, летит степная кобылица  
И мнёт ковыль...

А. Блок «На поле Куликовом»

И вечный бой! Покой нам только снится Сквозь кровь и  
пыль... Летит, летит степная кобылица И мнет ковыль...



Браузер «проглатывает» переходы на новую строку!

# Абзацы (параграфы)

*paragraph* – абзац

`<p>И вечный бой! Покой нам только снится</p>`

`<p>Сквозь кровь и пыль...</p>`

`<p>Летит, летит степная кобылица </p>`

`<p>И мнёт ковыль...</p>`

интервал



# Выравнивание абзацев

*left, center, right, justify*

```
<p align="justify">
```

Молекула воды испарилась из кипящего чайника и, подлетая к потолку, лоб в лоб столкнулась с неизвестно как прокравшейся на кухню молекулой водорода. Кто быстрее отлетел?

```
</p>
```



Что плохо?

Решение:  
`align="left"`

Г. Остер «Задачник по физике»

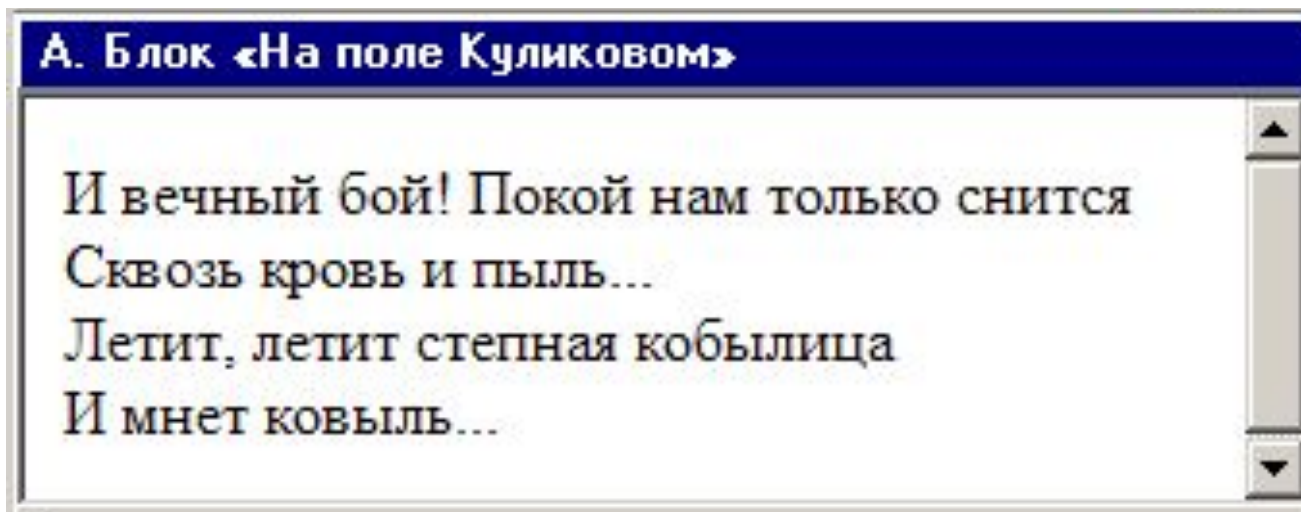
Молекула воды испарилась из кипящего чайника и, подлетая к потолку, лоб в лоб столкнулась с неизвестно как прокравшейся на кухню молекулой водорода. Кто быстрее отлетел?



# Переход на новую строку

*break* –  
разрыв

И вечный бой! Покой нам только снится  
<br>Сквозь кровь и пыль...  
<br>Летит, летит степная кобылица  
<br>И мнёт ковыль...



# Специальные символы (*HTML entities*)

Символ	HTML-код	Название
	<code>&amp;nbsp;</code>	неразрывный пробел
–	<code>&amp;ndash;</code>	короткое тире
—	<code>&amp;mdash;</code>	(длинное) тире
§	<code>&amp;sect;</code>	параграф
«	<code>&amp;laquo;</code>	левая русская кавычка
»	<code>&amp;raquo;</code>	правая русская кавычка
<	<code>&amp;lt;</code>	левая угловая скобка
>	<code>&amp;gt;</code>	правая угловая скобка
©	<code>&amp;copy;</code>	символ авторского права
®	<code>&amp;reg;</code>	зарегистрированная торговая марка
°	<code>&amp;deg;</code>	градус
²	<code>&amp;sup2;</code>	квадрат
³	<code>&amp;sup3;</code>	куб
×	<code>&amp;times;</code>	знак умножения
÷	<code>&amp;divide;</code>	знак деления

# Применение специальных символов

Неразрывный пробел (`&nbsp;` ; )

Дом сдали в 2011 году .

А.С. Пушкин

Пёс весил 12 кг .

не отрывать:

год

инициалы

единицы  
измерения

Из дома вышел **А.С.  
Пушкин** – солнце  
русской поэзии.



# Применение специальных символов

## Длинное тире ( `&mdash;` )

Вышел А.С. `&nbsp;`Пушкин`&nbsp;`; `&mdash;`  
солнце русской поэзии.

не начинать  
строку с тире!

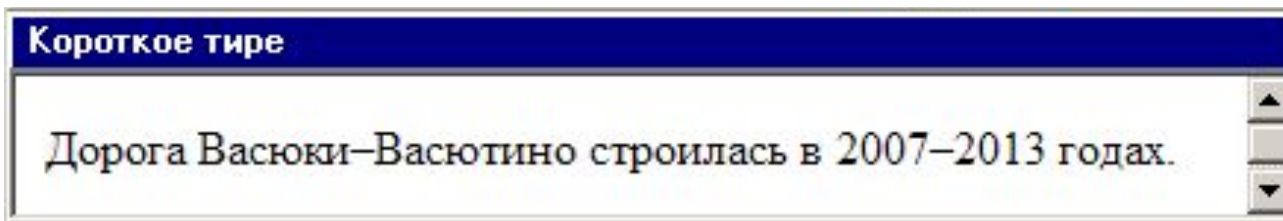
~~Вышел А.С. Пушкин  
– солнце русской  
поэзии.~~



Что плохо?

## Короткое тире ( `&ndash;` )

Дорога Васюки`&ndash;`Васютино строилась  
в 2007`&ndash;`2013`&nbsp;`годах.



# Применение специальных символов

---

Угловые скобки (`&lt;` `&gt;`)

`<p>Верно ли, что X < Y?</p>`



В чём проблема?

X `&lt;` Y

это начало  
тэга?

# Маркированные списки

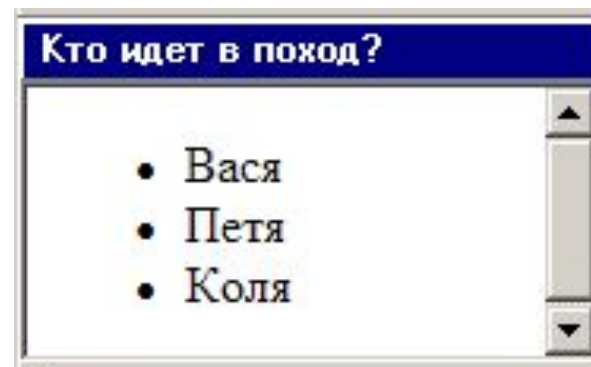
## unordered list

(неупорядоченный список)

## list item

(элемент списка)

```
<ul>
<li>Вася</li>
<li>Петя</li>
<li>Коля</li>
</ul>
```



## изменение маркера:

```
<ul type="disc">
...
</ul>
```

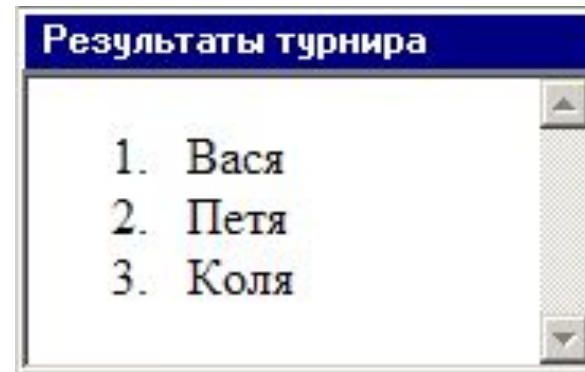
disc     ·  
circle    ○  
square    ■

# Нумерованные списки

## ordered list

(упорядоченный список)

```
<ol>  
<li>Вася</li>  
<li>Петя</li>  
<li>Коля</li>  
</ol>
```



## изменение нумерации:

1, i, I, a, A

```
<ol type="i" start="4">  
...  
</ol>
```



# Многоуровневые списки

```
<ol>
```

```
<li>Россия
```

```
<ul>
```

```
<li>Москва</li>
```

```
<li>Санкт-Петербург</li>
```

```
</ul>
```

```
</li>
```

```
<li>Украина
```

```
<ul>
```

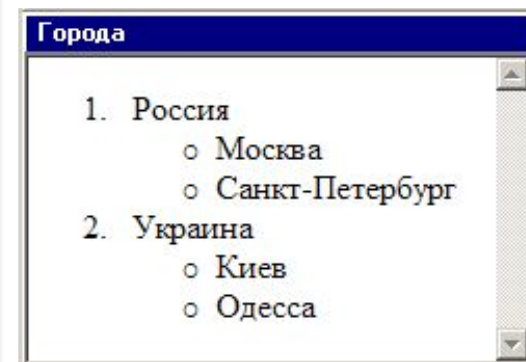
```
<li>Киев</li>
```

```
<li>Одесса</li>
```

```
</ul>
```

```
</li>
```

```
</ol>
```





# Гиперссылки (локальные)

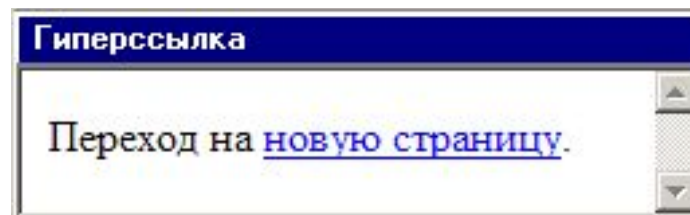
## страница в том же каталоге:

Переход на

```
<a href="newpage.html">новую страницу</a>.
```

anchor  
(якорь)

hyper reference  
(гиперссылка)



## во вложенном каталоге:

```
<a href="news/info.htm">Информация</a>.
```

## в родительском каталоге:

```
<a href=" ../info.htm">Информация</a>.
```

## в соседнем каталоге:

```
<a href=" ../news/info.htm">Информация</a>.
```

# Гиперссылки (внешние)

---

на URL:

```
<a href="http://example.net/news/info.htm">  
Информация</a>.
```

на главную страницу сайта:

```
<a href="http://example.net">Информация</a>.
```

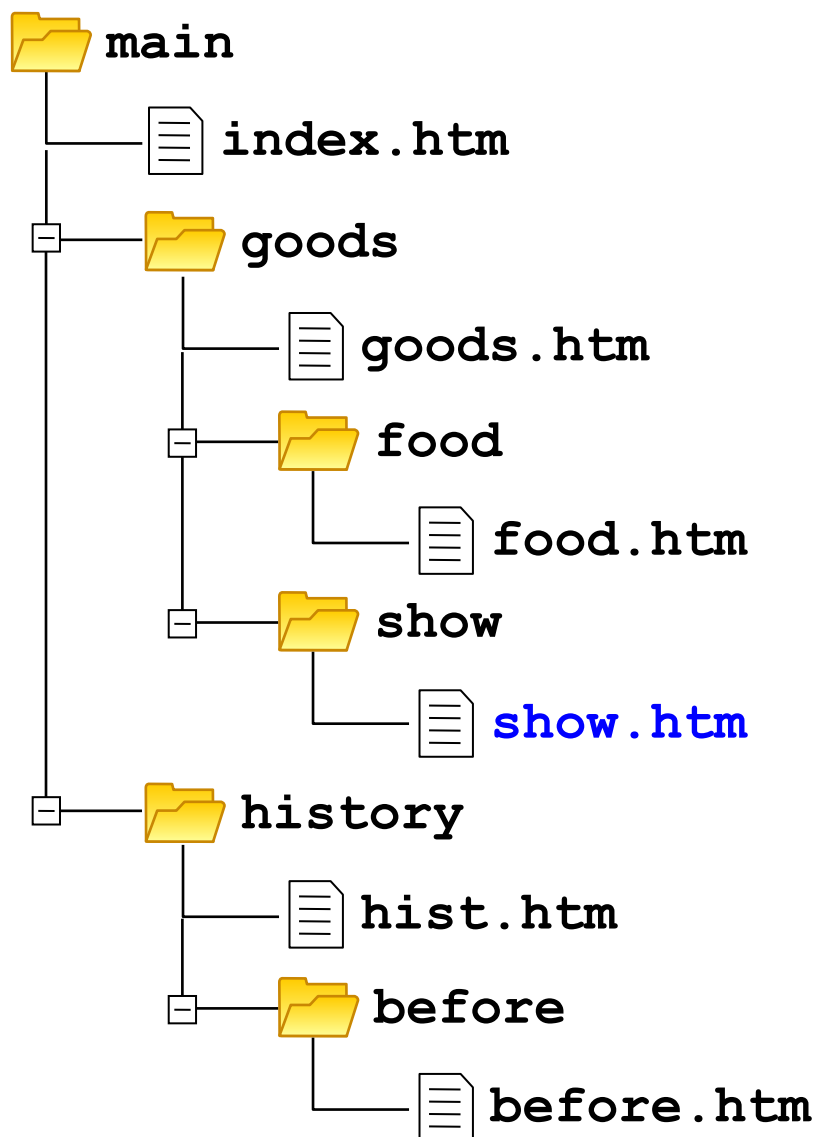
для скачивания архива:

```
<a href="http://example.net/game.zip">  
Скачать</a>.
```

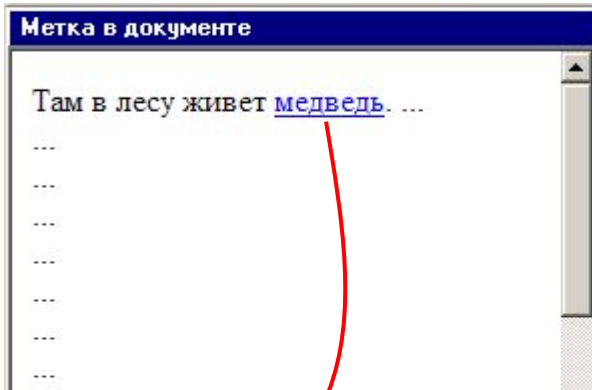
для запуска почтовой программы:

```
<a href="mailto:vasya@mail.ru">  
Напишите мне!</a>
```

# Как записать гиперссылки от `show.htm`?



# Метки внутри документов



Там в лесу живет

```
<a href="#bear">медведь</a>.
```

...

...

...

```
<a name="bear"></a>
```

```
<h2>Медведь</h2>
```

Медведь &nbsp; &mdash; хищное млекопитающее семейства медвежьих .

на метку в другом файле:

```
<a href="animals.htm#bear">медведь</a>
```

```
<a
```

```
href="http://qq.net/a.htm#bear">медведь</a>
```

# Куда переход?

---

```
<a href="example.html">...</a>
```

```
<a href=" ../example.html">...</a>
```

```
<a href=" ../../../../example.html">...</a>
```

```
<a href=" ../new/old/example.html">...</a>
```

```
<a href="example.html#resume">...</a>
```

```
<a href="download/example.zip">...</a>
```

```
<a href="mailto:john@yahoo.com">...</a>
```

# Создание веб-сайтов

## § 24. Оформление документов

# Содержание и оформление

Веб-страница:

оформление

содержание (контент)



содержание

```
<h1>Сборник задач по физике</h1>
<p class="author">Григорий Остер</p>
<h2>Задача 61</h2>
Петя ехал к бабушке на электричке, и всю
дорогу над ним издевались какие-то два
неведомые ему явления.
```

оформление



логическая разметка  
(\* .html)

физическая разметка  
(\* .css)

# Логическая разметка (HTML)

---

Выделение (*emphasize*):

```
<em>Вася</em>
```

Сильное выделение (*strong*):

```
<strong>Вася</strong>
```

Программный код (*code*):

```
<code>a := b + c ;</code>
```

Определение (*definition*):

```
<dfn>Информация</dfn> &mdash; это...
```

Цитата (*citation*):

```
<cite>Карету мне, карету!</cite>
```

Сокращение (*abbreviation*):

```
<abbr>НИИЧАВО</abbr>
```

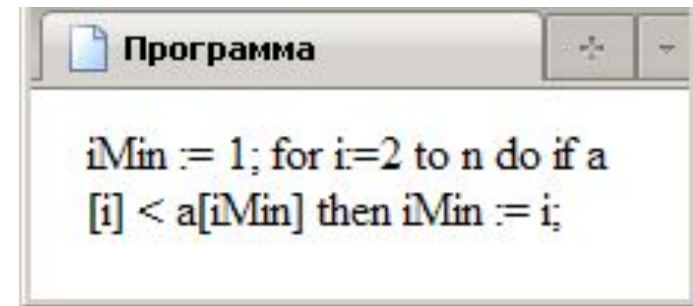


# Оформление текстов программ

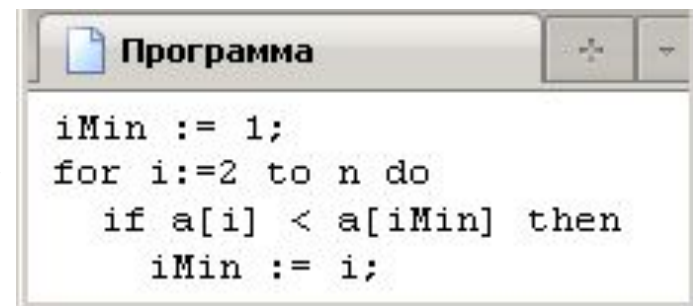
```
iMin := 1;
for i:=2 to n do
  if a[i] < a[iMin] then
    iMin := i;
```

*preformatted* (уже отформатированный)

```
<pre>
iMin := 1;
for i:=2 to n do
  if a[i] < a[iMin] then
    iMin := i;
</pre>
```



```
Программа
iMin := 1; for i:=2 to n do if a
[i] < a[iMin] then iMin := i;
```



```
Программа
iMin := 1;
for i:=2 to n do
  if a[i] < a[iMin] then
    iMin := i;
```

# Тэги физической разметки HTML

Курсив (*italic*):

```
<i>Вася</i>
```

Вася

Жирный (*bold*):

```
<b>Вася</b>
```

Вася

Подчёркивание (*underline*):

```
<u>Вася</u>
```

Вася

Зачёркивание (*strikeout*):

```
<s>Вася</s>
```

~~Вася~~

Верхний индекс (*superscript*):

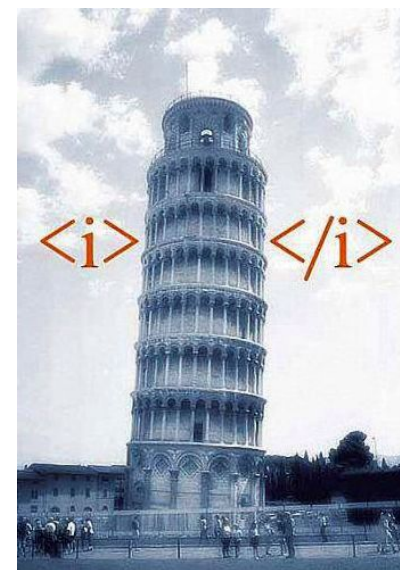
```
Вася<sup>2</sup>
```

Вася<sup>2</sup>

Нижний индекс (*subscript*):

```
Вася<sub>2</sub>
```

Вася<sub>2</sub>



# Стилевые файлы

## содержание

```
<h1>Рекурсия</h1>  
У попа была собака, он её любил,  
Она съела кусок мяса, он её убил,  
В землю закопал,  
Надпись написал:  
У попа была собака, он её любил,  
...
```

**CSS** (*Cascading Style Sheets*) – каскадные таблицы стилей.

## оформление



# Стилевые файлы

`test.css`

название тэга

`body {`

свойство

значение

селектор

`color: white;`

`background: #FF6600;`

`}`

`color` – цвет символов

`background` – цвет фона

$$R = FF_{16} = 255$$

$$G = 66_{16} = 102$$

$$B = 0$$

# Подключение стилевого файла

## qq.html

```
<html>
<head>
<title>Страница с файлом стилей</title>
<link rel="stylesheet" href="test.css">
</head>
<body>
Привет, Вася!
</body>
</html>
```

таблица стилей

Страница с файлом стилей

Привет, Вася!

## test.css

```
body {
  color: white;
  background: #0000E0;
}
```

# Наследование стилей

```
<body>
```

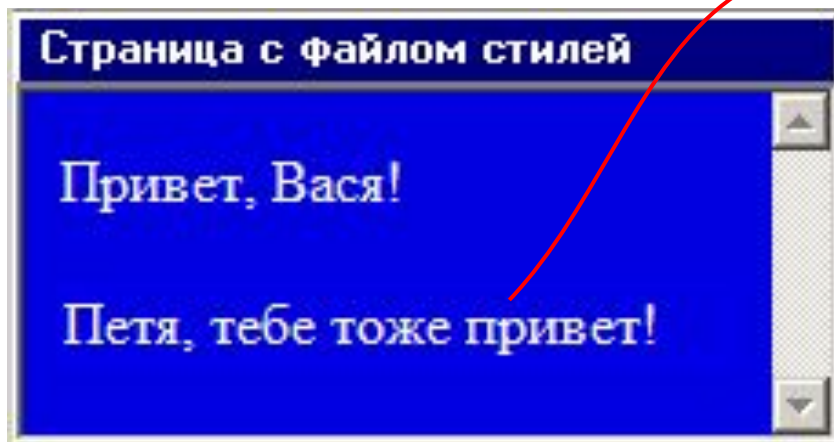
```
Привет, Вася!
```

```
<p>Петя, тебе тоже привет!</p>
```

```
</body>
```

наследуется

```
body {  
  color: white;  
  background: #0000E0;  
}
```



не наследуется  
(прозрачный фон)

# Стили для элементов: шрифты

для всех  
абзацев

семейство  
шрифтов

**serif** – с засечками  
**sans-serif** – без засечек  
**monospace** – моноширинный

```
p {  
  font-family: Arial, sans-serif;  
  font-size: 20px;  
  font-style: italic;  
  font-weight: bold;  
}
```

размер в пикселях

**normal** – обычный  
**italic** – курсив

**normal** – обычный  
**bold** – жирный

# Стили: размеры, выравнивание

```
p {  
  background: #E6E6FF;  
  width: 80%;  
  height: 100px;  
  text-align: left;  
  text-indent: 20px;  
}
```

цвет фона

ширина

высота

абзацный  
отступ

выравнивание:

left

center

right

justify

20px

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur felis velit, tempus sit amet cursus at, porta ut turpis. Sed augue tellus, tincidunt id ultricies at, tempor quis sem.

100px

80%



# Рамка, поля, отступы

**solid** – сплошная  
**dashed** – штриховая  
**dotted** – точечная

рамка

```

p {
  background: #ccffcc;
  border: 1px solid green;
  margin: 0 40px 20px 40px;
  padding: 5px;
}
  
```

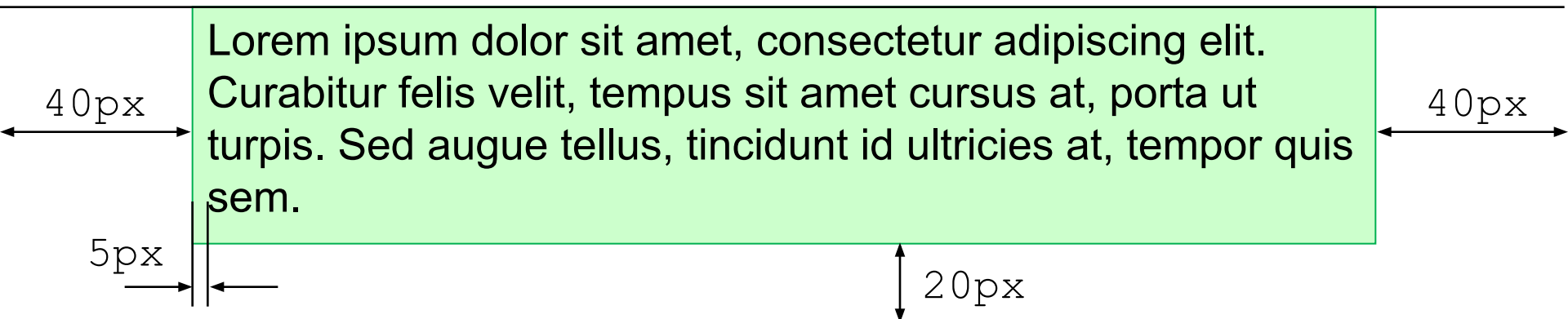
отступы  
снаружи

цвет

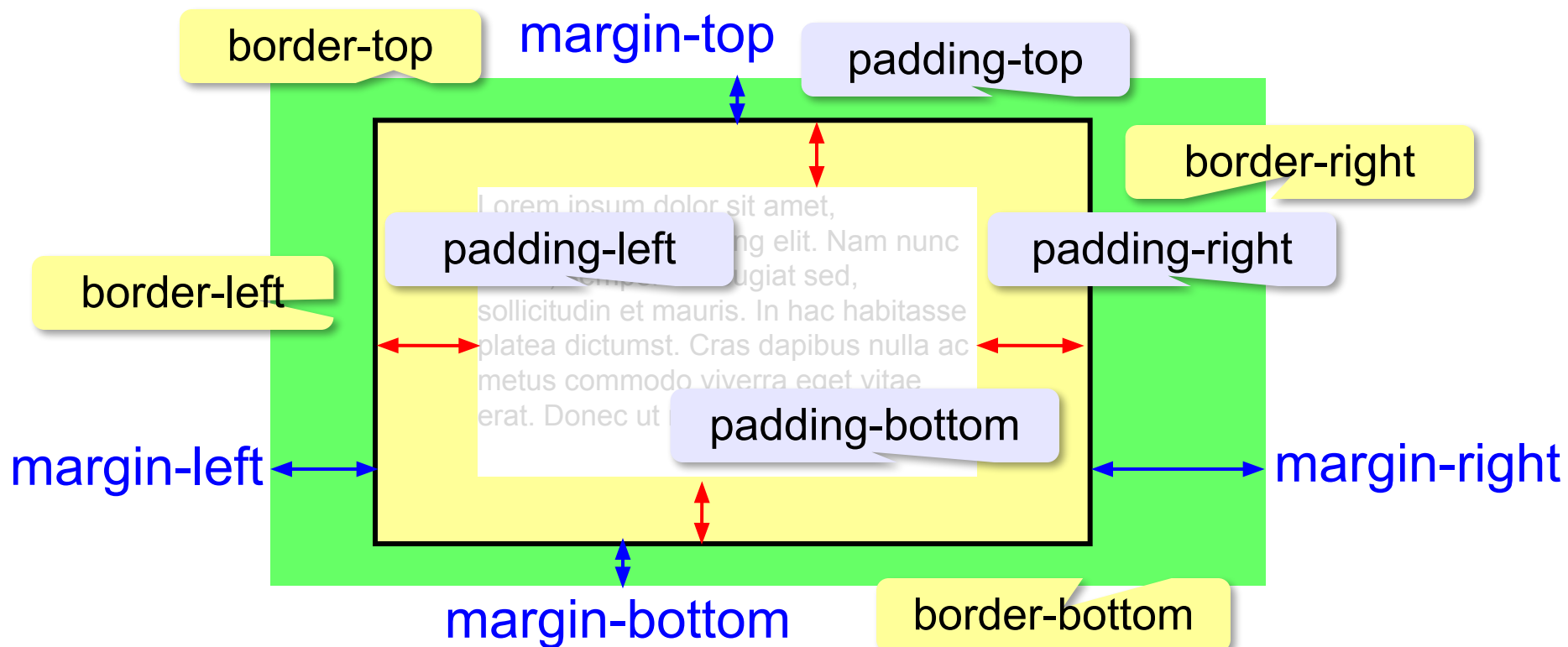
отступы  
внутри

со всех сторон

сверху, справа,  
снизу, слева



# Рамка, поля и отступы

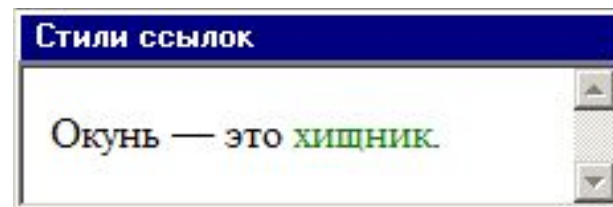


# Стили гиперссылок

## Все гиперссылки:

```
a {  
  color: green;  
  text-decoration: none;  
}
```

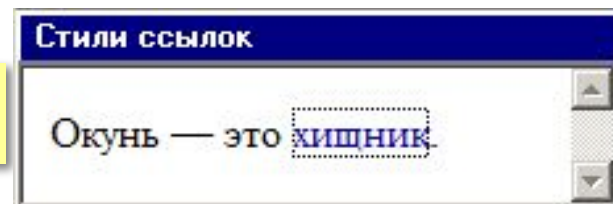
убрать  
подчёркивание



## Посещённые гиперссылки:

```
a:visited { color: #0000AA; }
```

псевдокласс

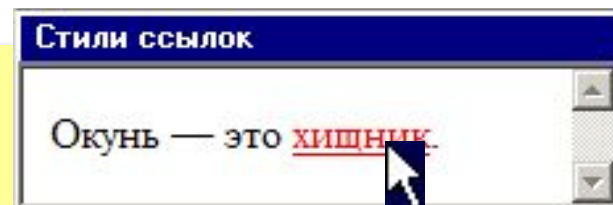


## Гиперссылки при наведении мыши:

```
a:hover {  
  color: red;  
  text-decoration: underline;  
}
```

псевдокласс

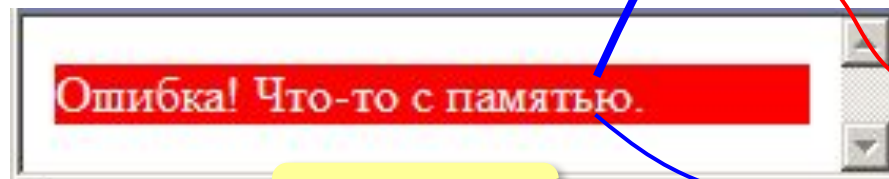
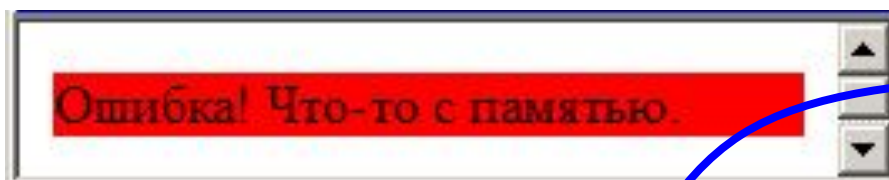
подчёркивание



# Классы

```
<p class="error">Ошибка!  
Что-то с памятью.  
</p>
```

для абзацев  
класса **error**



каскад!

```
p.error {  
  background:red;  
}
```

```
p {  
  background:blue;  
  color:white;  
}
```



Более конкретное указание отменяет более общее!

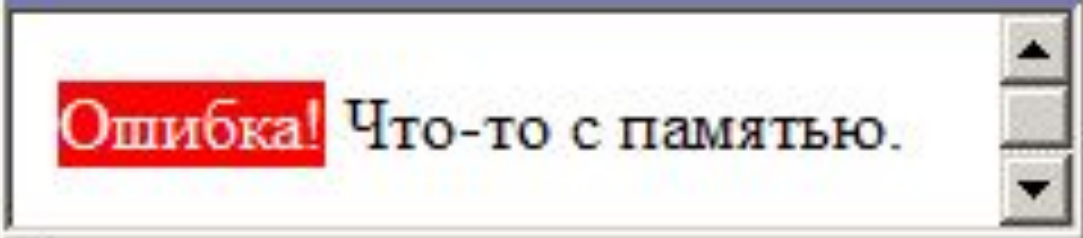
# Классы

для **всех** элементов  
класса `error`

```
.error {  
  background:red;  
  color:white;  
}
```

ЭЛЕМЕНТ В  
ТЕКСТЕ

```
<p>  
<span class="error">Ошибка!</span>  
Что-то с памятью.  
</p>
```



Ошибка! Что-то с памятью.

# Фоновый рисунок

```
p.hallo {  
  background: white url(images/grad.jpg);  
}
```

```
<p class="hallo">  
Привет, Вася!  
</p>
```

Привет, Вася!

```
p.hallo {  
  background: url(grad.jpg) repeat-y;  
}
```

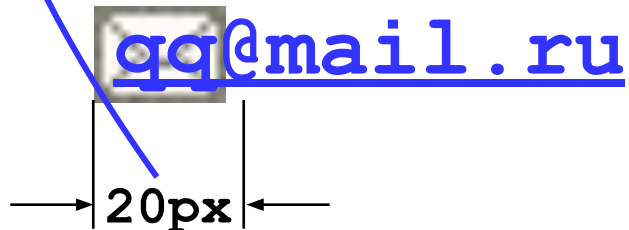
Привет, Вася!

# Фоновый рисунок без повторения



```
p.email {  
  background: url(letter.gif) no-repeat;  
  padding-left: 20px;  
}
```

```
<p class="email">  
<a href="mailto:qq@mail.ru">qq@mail.ru</a>  
</p>
```



# Составные селекторы

---

Общие свойства для нескольких тэгов:

```
code, pre {  
    font-weight: bold;  
    font-family: Courier New, monospace;  
}
```

Вложенные элементы:

**a** внутри **em**

```
em a {  
    font-style: normal;  
}
```



# Сложные селекторы

ссылки внутри абзаца класса email

```
p.email a {  
  color: green;  
  text-decoration:none;  
}
```

убрать подчеркивание

посещённые ссылки

```
p.email a:visited {  
  color: #F0F;  
}
```

то же, что #FF00FF

мышь над ссылкой

```
p.email a:hover {  
  color: #00F;  
  text-decoration:underline;  
}
```

подчеркнуть

# Создание веб-сайтов

## § 25. Рисунки, звук, видео

# Форматы рисунков

## **GIF** (*Graphic Interchange Format*)



- сжатие без потерь (LZW)
- прозрачные области
- анимация
- **только с палитрой** (2...256 цветов)

рисунки с четкими границами, мелкие рисунки

## **JPEG** (*Joint Photographer Expert Group*)

- сжатие с потерями
- только *True Color* (16,7 млн. цветов)
- **нет анимации и прозрачности**

рисунки с размытыми границами, фото



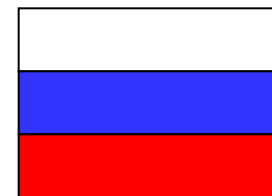
## **PNG** (*Portable Network Graphic*)

- сжатие без потерь
- с палитрой (PNG-8) и *True Color* (PNG-24)
- прозрачность и полупрозрачность (альфа-канал)
- нет анимации
- **плохо сжимает мелкие рисунки**

# Форматы рисунков

**SVG** (*Scalable Vector Graphics*, масштабируемые векторные изображения)

```
<svg>
<rect width="135" height="30"
      x="0" y="10"
      stroke-width="1" stroke="rgb(0,0,0)"
      fill="rgb(255,255,255)" />
<rect width="135" height="30" x="0" y="40"
      stroke-width="1" stroke="rgb(0,0,0)"
      fill="rgb(0,0,255)" />
<rect width="135" height="30" x="0" y="70"
      stroke-width="1" stroke="rgb(0,0,0)"
      fill="rgb(255,0,0)" />
</svg>
```



# Рисунки в документе

---

из той же папки:

*image*  
(изображение)

*source*  
(источник)

```

```

из другой папки:

```

```

```

```

с другого сервера:

```

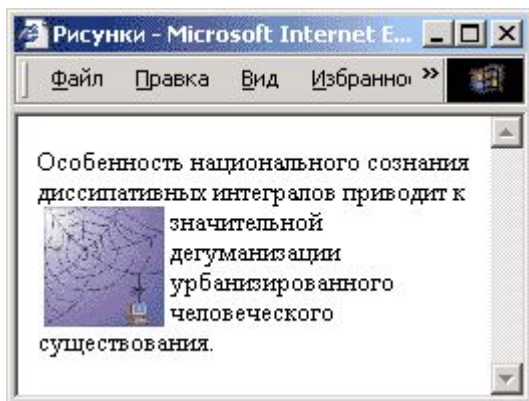
```

# Выравнивание

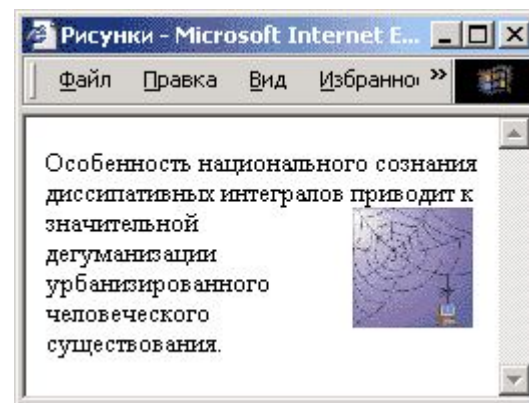
```

```

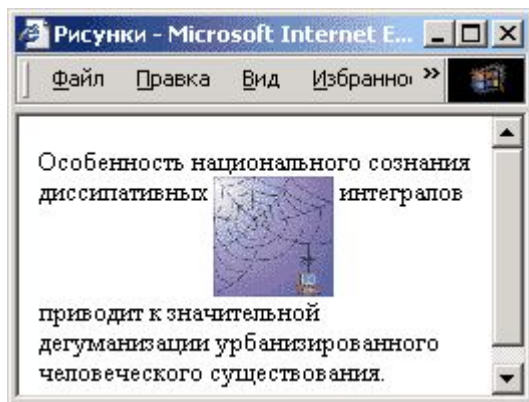
left



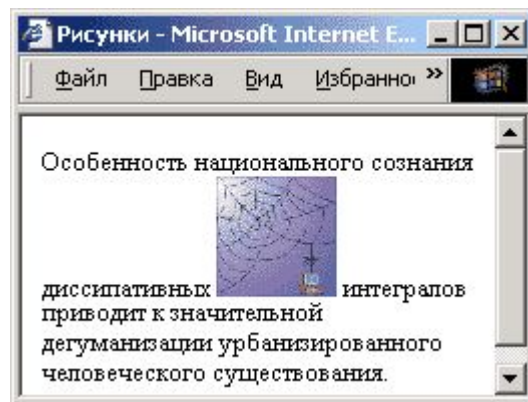
right



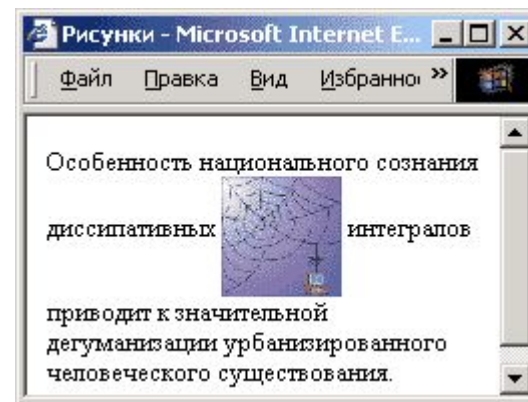
top



bottom  
(по умолчанию)



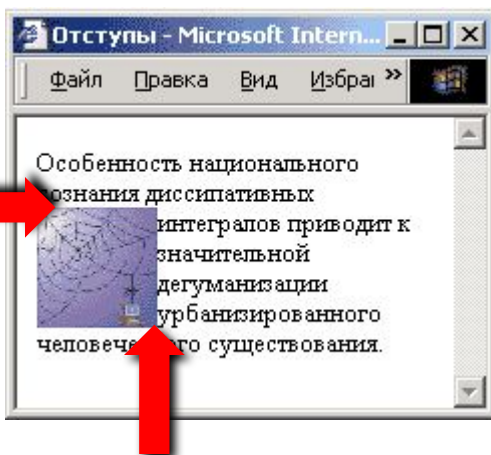
middle



# Отступы

```

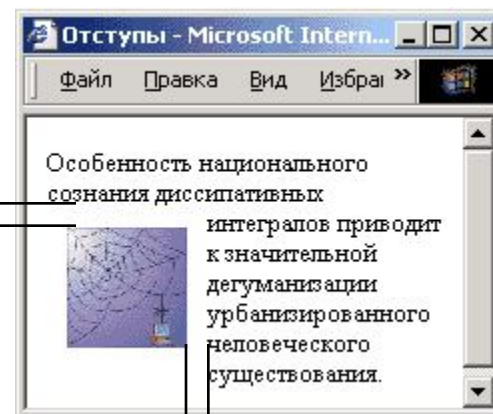
```



**vspace**  
(vertical space)

```

```



**hspace**  
(horizontal space)

# Другие атрибуты

```

```

**размеры** позволяют:

- растянуть – сжать
- не портить дизайн,  
если рисунка нет



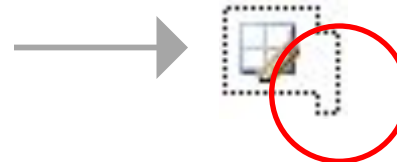
# Рисунок-гиперссылка

## локальная ссылка:

```
<a href="gallery.htm">  
  
</a>
```

иначе будет синяя  
рамка вокруг

если `</a>` не вплотную к  
`<img ...>`, будет «хвост»



## ссылка на другой сервер:

```
<a href="http://www.mail.ru">  
</a>
```

не будет  
«хвоста»

# Вставка векторных рисунков

---

тип нестандартных  
данных

```
<object type="image/svg+xml"  
  data="test.svg"  
  width="48" height="48"  
  align="left">  
</object>
```

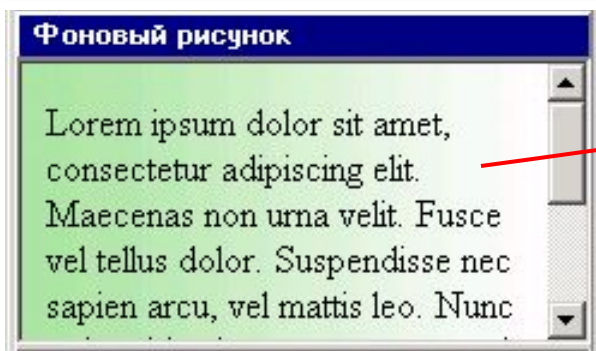
имя файла

размеры

выравнивание

# ФОНОВЫЙ РИСУНОК

```
body {  
    background: url(grad.jpg);  
}
```



если рисунка нет...

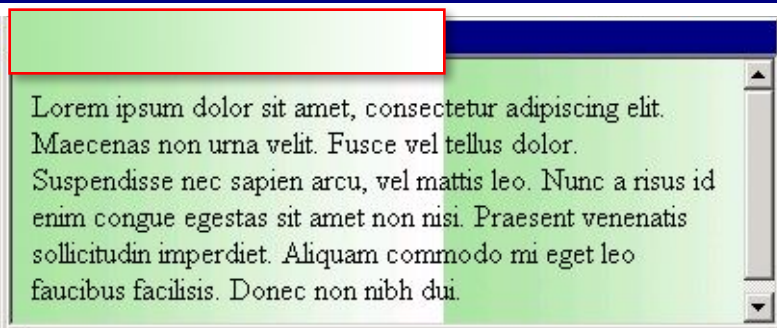
```
body {  
    background: url(grad.jpg) #EEE;  
}
```

'images/grad.jpg'

'../images/grad.jpg'

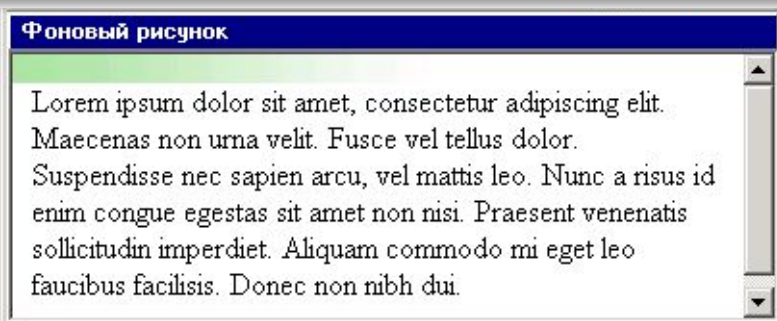
'http://www.vasya.ru/images/grad.jpg'

# Фоновый рисунок



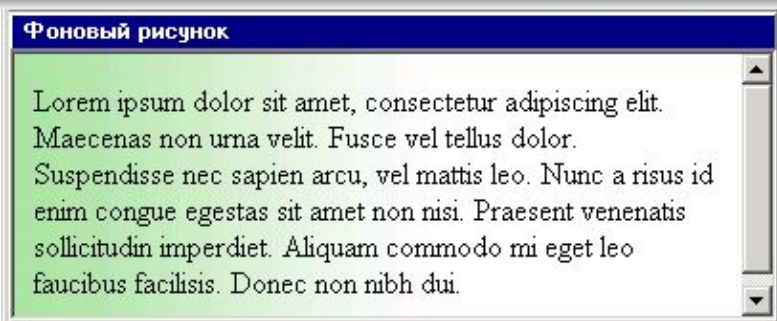
если рисунок  
меньше, он  
повторяется

```
body{ background: url (grad.jpg) no-repeat; }
```



не повторять

```
body{ background: url (grad.jpg) repeat-y; }
```

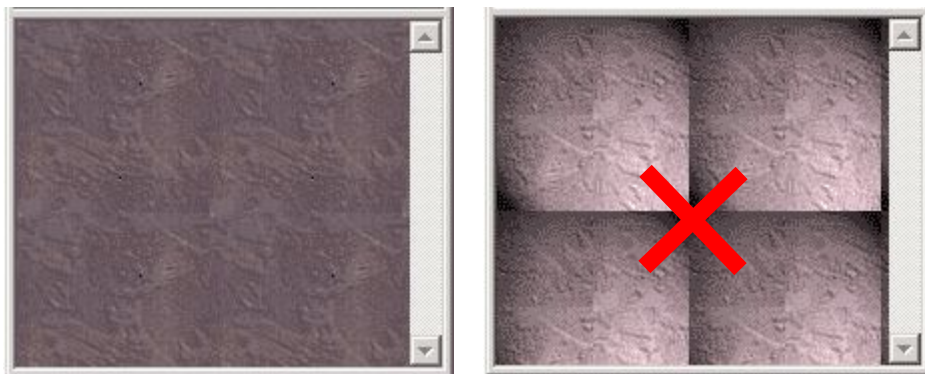


повторять по  
вертикали

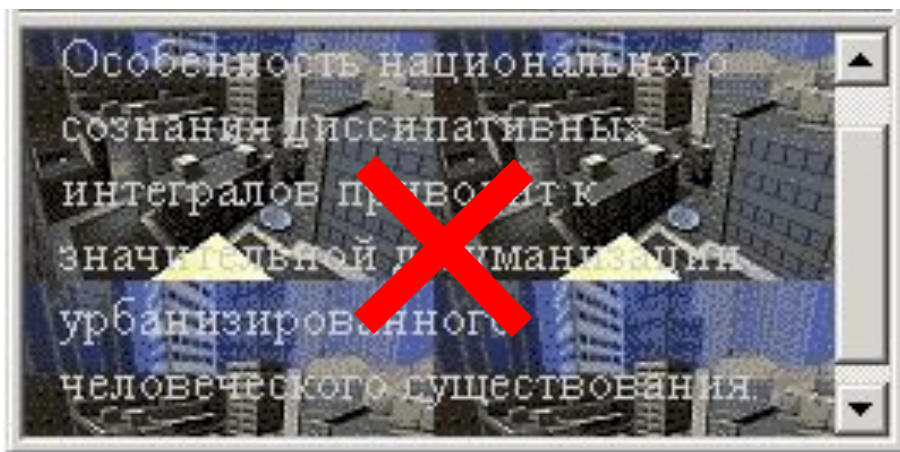
# Фоновый рисунок



Не должно быть «швов»!



Фон не должен мешать чтению!



# Вставка мультимедийных файлов

Звуковой файл:

имя файла

```
<embed src="myaw.wav"  
autostart="false"></embed>
```

автозапуск



Флэш-анимация:

имя файла

```
<embed src="cube.swf"  
width="275" height="200">  
</embed>
```

размеры

# Вставка мультимедийных файлов

---

## Видео:

адрес ролика

```
<embed  
src="http://www.youtube.com/v/YvLNA5OW6xZ"  
width="425" height="350">  
</embed>
```

размеры

# HTML5 – тэг `audio`

```
<audio  
  src="nature.ogg"  
  autostart="false"  
  controls="controls"  
  width="425"  
  height="350">  
</audio>
```

имя файла

показывать элементы  
управления

размеры





## HTML5 – тэг video

```
<video src="sail.ogv"  
  controls="controls"  
  autostart="false"  
  width="425" height="350">
```

Ваш браузер не поддерживает элемент video.

```
</video>
```

```
<video width="425" height="350"  
  controls="controls">  
<source src="start.mp4"  
  type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'>  
</video>
```

```
<video width="425" height="350"  
  controls="controls">  
<source src="finish.webm"  
  type='video/webm; codecs="vp8, vorbis"'>  
</video>
```

# Создание веб-сайтов

## § 26. Таблицы

# Простые таблицы

```
<table border="1">
```

```
. . .
```

```
</table>
```

толщина рамки

`table row`  
(строка)

`table data`  
(ячейка с  
данными)

```
<table border="1">
```

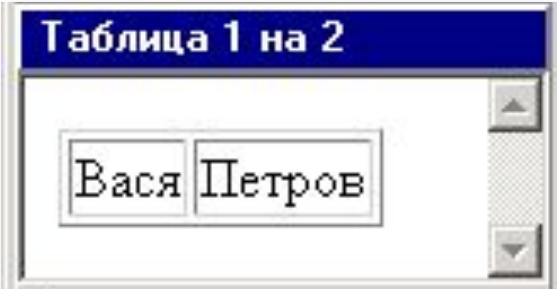
```
<tr>
```

```
<td>Вася</td>
```

```
<td>Петров</td>
```

```
</tr>
```

```
</table>
```



Вася	Петров
------	--------

# Заголовки таблиц

```
<table border="1">
<tr>
  <th>месяц</th>
</tr>
<tr>
  <td>январь</td>
</tr>
<tr>
  <td>февраль</td>
</tr>
<tr>
  <td>март</td>
</tr>
</table>
```

**table header**  
(заголовков)

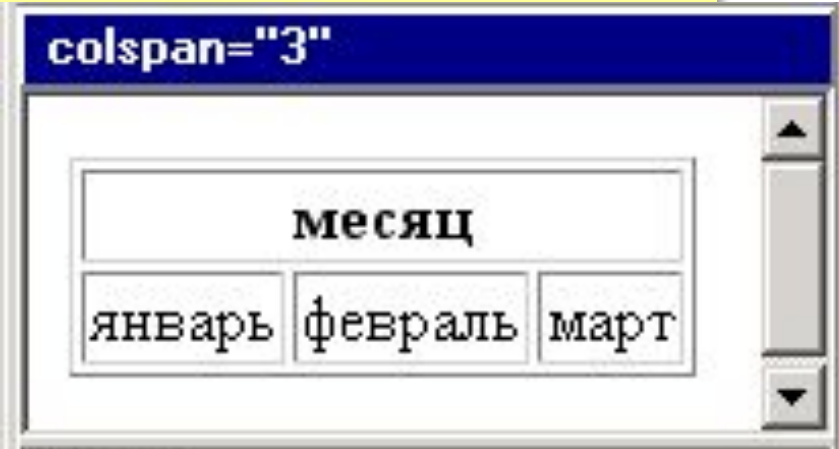
жирный, по  
центру



# Объединение ячеек – столбцы

```
<table border="1">
<tr>
  <th colspan="3">месяц</th>
</tr>
<tr>
  <td>январь</td>
  <td>февраль</td>
  <td>март</td>
</tr>
</table>
```

**column span**  
(ОХВАТ СТОЛБЦОВ)



месяц		
январь	февраль	март

# Объединение ячеек – строки

```
<table border="1">
<tr>
  <th rowspan="3">Привет, </th>
  <td>Вася!</td>
</tr>
<tr>
  <td>Петя!</td>
</tr>
<tr>
  <td>Коля!</td>
</tr>
</table>
```

**row span**  
(охват строк)

rowspan="3"	
Привет,	Вася!
	Петя!
	Коля!

# Табличная вёрстка

```
<table>
```

```
<tr>
```

```
<td>
```

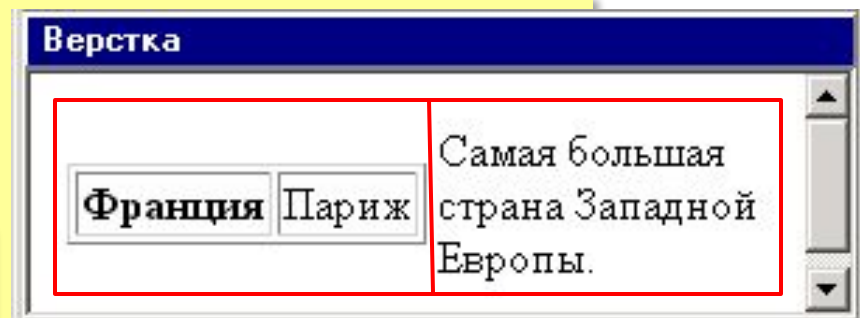
```
<table border="1">  
  <tr>  
    <th>Франция</th>  
    <td>Париж</td>  
  </tr>  
</table>
```

```
</td>
```

```
<td>Самая большая страна  
  Западной Европы.</td>
```

```
</tr>
```

```
</table>
```



скрытая таблица!

# Вложенные таблицы

```
<table border="0" cellspacing=10>
```

```
<tr><td>
```

```
<table border="1">
```

```
<tr><td>Вася</td><td>Петя</td></tr>
```

```
<tr><td>Маша</td><td>Даша</td></tr>
```

```
</table>
```

```
</td>
```

```
<td>
```

```
<table border="1">
```

```
<tr><td>1</td><td>22</td></tr>
```

```
<tr><td>333</td><td>4444</td></tr>
```

```
</table>
```

```
</td></tr>
```

```
</table>
```

Вася	Петя	1	22
Маша	Даша	333	4444



# Размеры

всей таблицы:

высота в пикселях

```
<table width="60%" height="300">  
...  
</table>
```

ширина в пикселях  
или в % от ширины  
окна браузера

строки:

```
<tr height="50">  
...  
</tr>
```

ширина в пикселях  
или в % от ширины  
таблицы

ячейки:

```
<td width="25%" height="50">  
...  
</td>
```

# Размеры (через CSS)

---

всей таблицы:

```
table.spec {  
  width: 60%;  
  height: 300;  
}
```

```
<table class="spec">  
  ...  
</table>
```

строки:

```
table.spec tr {  
  height: 50px;  
}
```

ячейки:

```
table.spec td.qq {  
  width: 25%;  
  height: 50px;  
}
```

# Выравнивание

## всей таблицы:

```
<table align="center">
...
</table>
```

left,  
center,  
right

## информации в ячейках:

```
<table border="1">
<tr align="center"
    valign="top">
  <td width="120">по центру,
    по верхней границе</td>
  <td align="right" valign="middle"
    width="200">по правой границе,
    по середине</td>
</tr>
</table>
```

left,  
center,  
right

top,  
middle,  
bottom

По центру, по  
верхней  
границе

По правой границе, по  
середине

# Выравнивание (через CSS)

```
<table class="ex" border="1">
<tr class="centop">
  <td>По центру,
    по верхней границе</td>
  <td class="rmid">По правой
    границе, по середине</td>
</tr>
</table>
```

По центру, по  
верхней  
границе

По правой границе, по  
середине

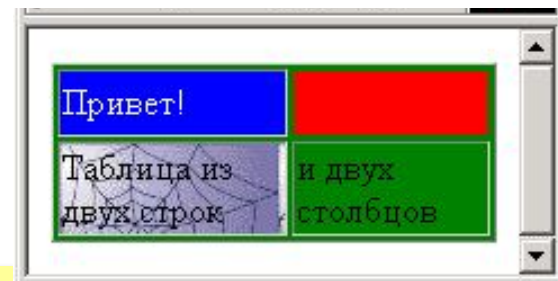
```
table.ex tr.centop {
  text-align: center;
  vertical-align: top;
  height: 100px;
}
table.ex td.rmid {
  text-align: right;
  vertical-align: middle;
  width: 200px;
}
```

left,  
center,  
right

top,  
middle,  
bottom

# Фон и цвет текста (через CSS)

```
<table border="1" class="qq">
<tr class="spec">
  <td>Привет!</td>
  <td class="r"></td>
</tr>
<tr>
  <td class="pic">
    Таблица
    из двух строк</td>
  <td>и двух столбцов</td>
</tr>
</table>
```



```
table.qq td {
  background: green;
}
table.qq tr.spec td {
  background: blue;
  color: white;
}
table.qq tr.spec td.r {
  background: red;
}
table.qq td.pic {
  background: url("web.jpg");
}
```

цвет фона

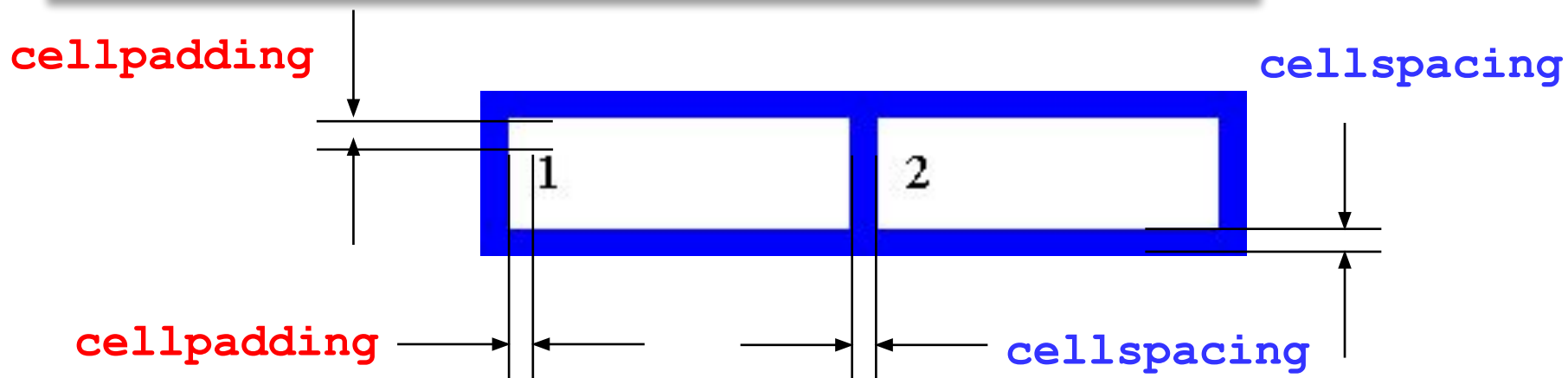
рисунок

# Отступы

интервал между  
ячейками

отступ внутри  
ячеек

```
<table cellpadding = "10" cellspacing = "10"  
  bgcolor = blue>  
<tr bgcolor = white>  
  <td width = "100">1</td>  
  <td width = "100">2</td>  
</tr>  
</table>
```



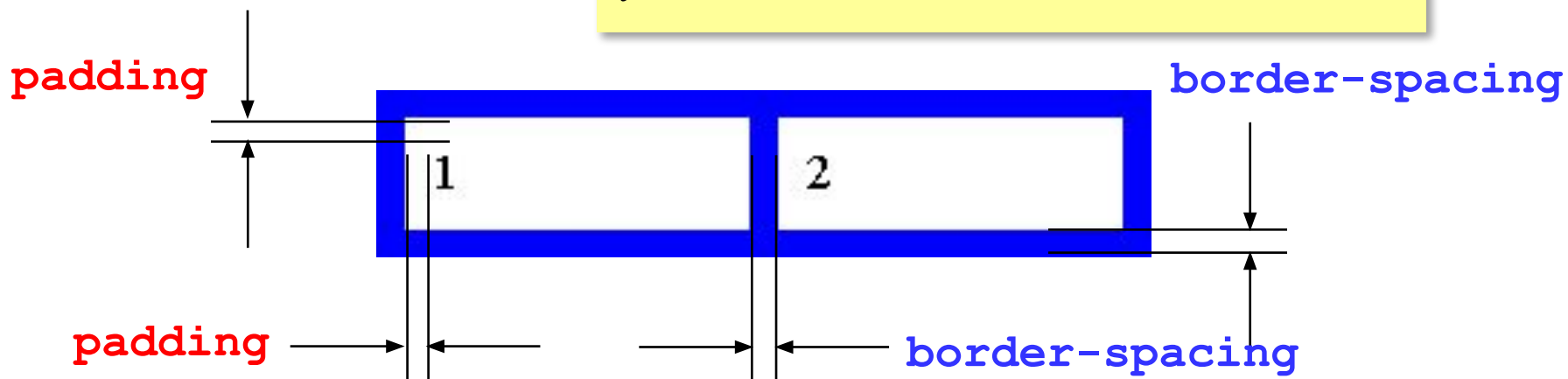
# Отступы (через CSS)

**идентификатор** – уникальное имя

```
<table id="qq">
<tr>
  <td> 1 </td>
  <td> 2 </td>
</tr>
</table>
```

```
table#qq {
  background: blue;
  border-collapse: separate;
  border-spacing: 10px;
  padding: 10px;
}
#qq tr {
  background: white;
}
```

кроме IE 6



# Создание веб-сайтов

## § 27. Блоки

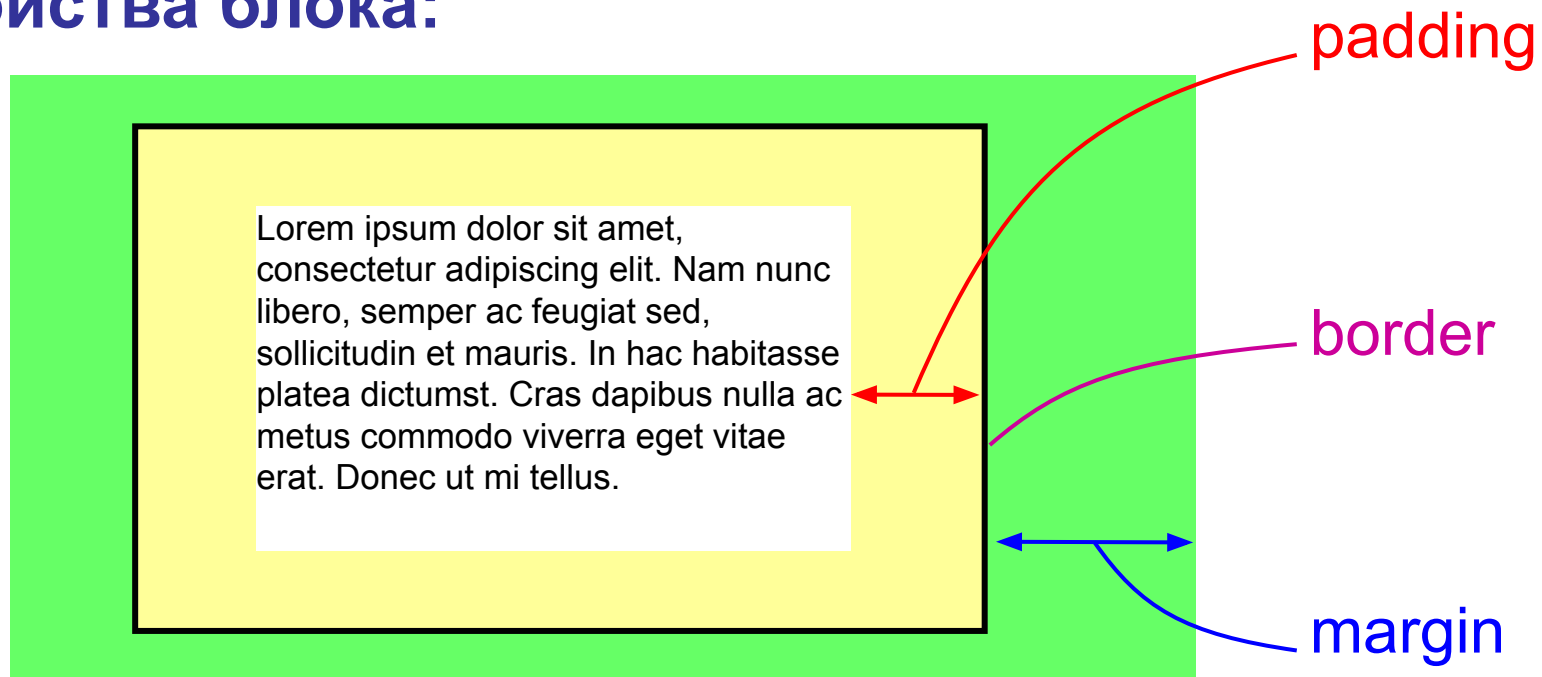


# Что такое блок (**div**)?

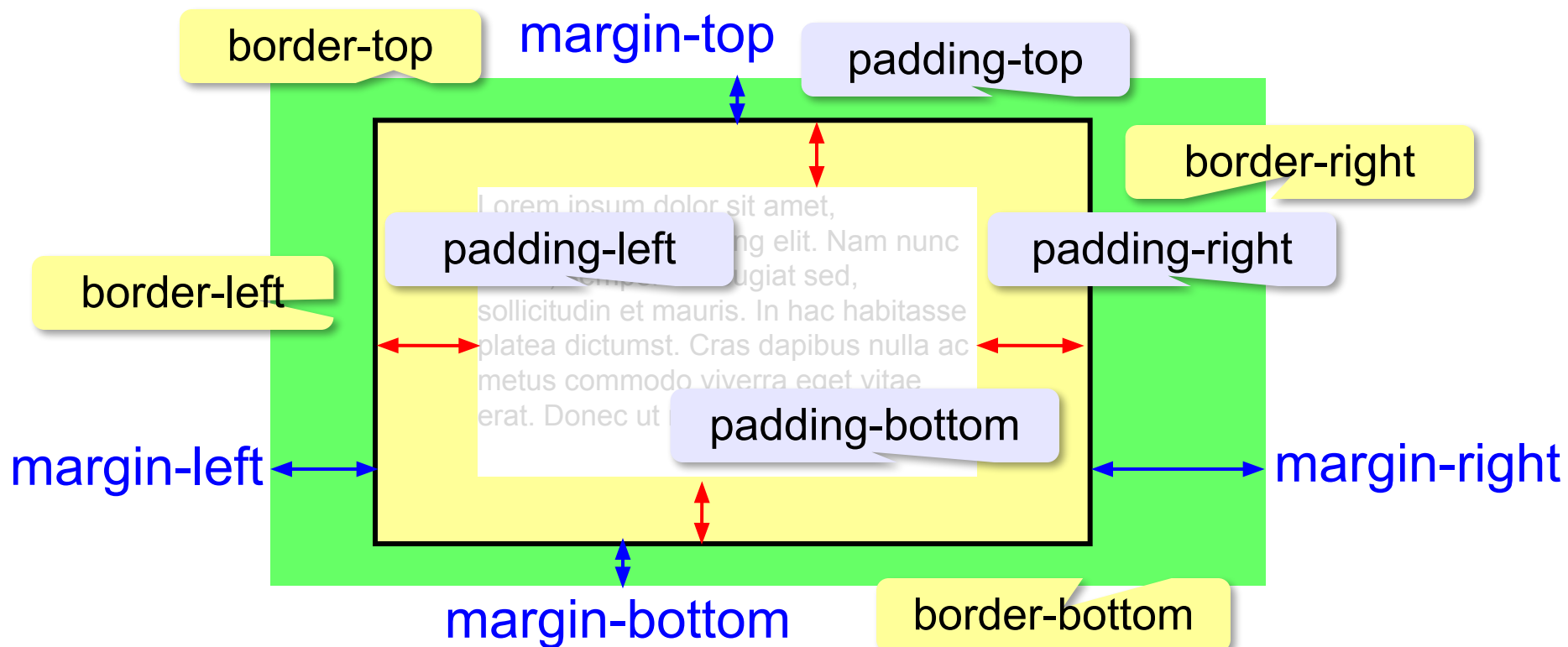
```
<div>  
...  
</div>
```

```
<p>  
<table>  
<a>  
<img>  
...
```

## свойства блока:



# Рамка, поля и отступы

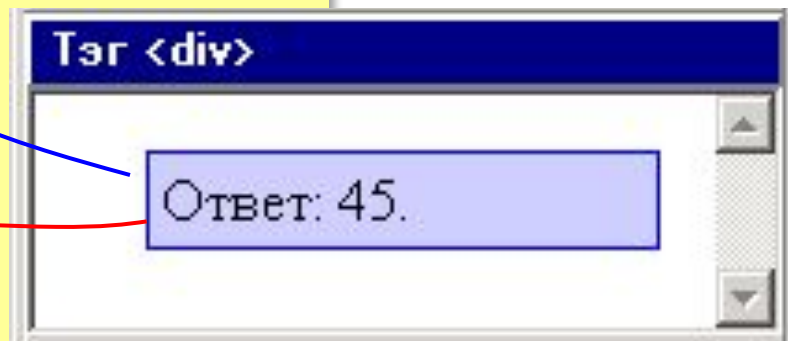


# Стили для блоков

```
<div class="info" id="result">  
Ответ: 45.  
</div>
```

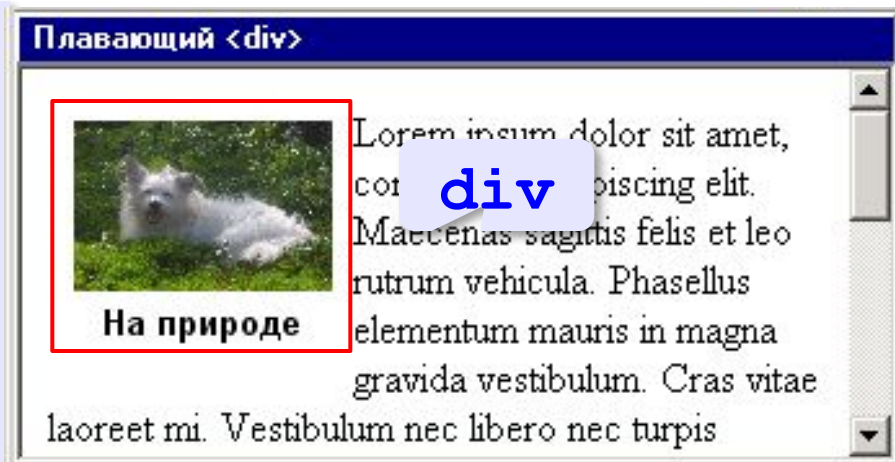
идентификатор  
(уникальный!)

```
.info {  
  margin: 5px 5px 10px 20px;  
  padding: 3px;  
}  
#result {  
  background: #CCCCFF;  
  border: 1px solid blue;  
}
```



## «Плавающий» блок

```
<div class="picture">
  
  <p>На природе</p>
</div>
```



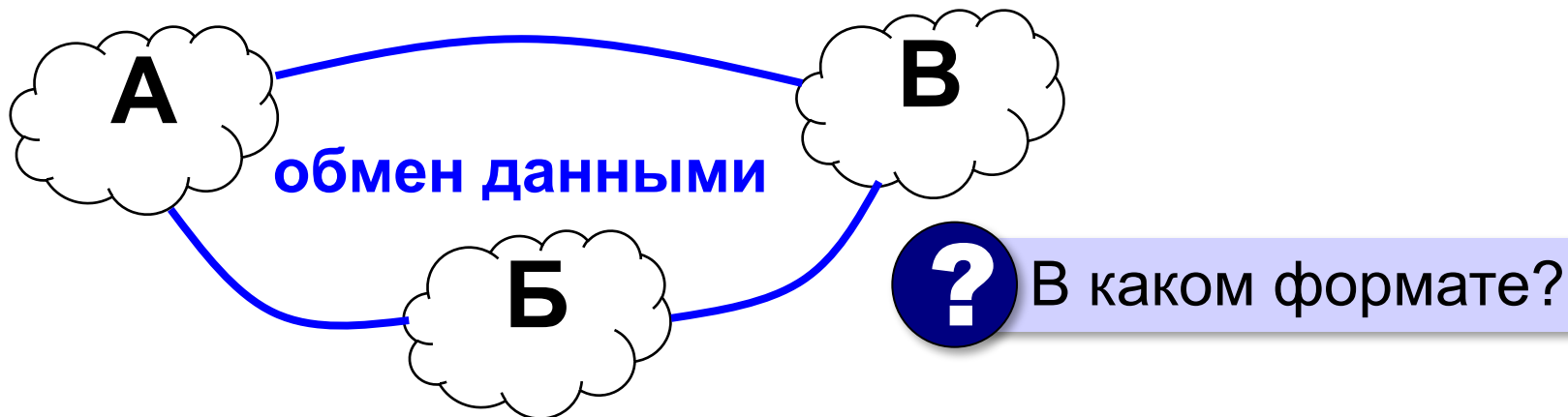
```
.picture { float:left; margin: 5px; }
.picture p {
  margin: 0;
  text-align: center;
  font-family: sans-serif;
  font-size: 80%;
  font-weight: bold;
}
```

# Создание веб-сайтов



## § 28. XML и XHTML

# В чём проблема?

Интеграция (объединение) информационных систем



## Двоичные файлы:

-  ■ небольшой объём
-  ■ множество форматов, приёмник должен суметь прочитать файл

# В чём проблема?

---

## Задачи:

- универсальный формат
- текстовая форма
- понятен человеку при просмотре
- только содержание (без оформления)
- сохранение структуры (главы, разделы)
- автоматическая обработка

## HTML:



- содержит тэги физической разметки
- нельзя вводить новые тэги
- нестрогий синтаксис (можно не закрывать `<p>`)

# Что такое XML?

```
<?xml version="1.0"?>
<компьютер>
  <процессор частота="2 ГГц">Intel Celeron
</процессор>
  <память фирма="Kingston">2048 Мб</память>
  <винчестер фирма="Seagate">320 Гб
</винчестер>
  <периферия>
    <монитор>Philips 190C1SB</монитор>
    <клавиатура>Logitech Classic 200
  </клавиатура>
    <мышь>Genius Navigator 600</мышь>
  </периферия>
</компьютер>
```



# Что такое XML?

---

**XML** = *eXtensible Markup Language*  
(расширяемый язык разметки)

- тэги-контейнеры в угловых скобках

`<тэг>... </тэг>`

- атрибуты тэгов (дополнительные данные)

`<тэг атрибут="значение">... </тэг>`

- МОЖНО ВВОДИТЬ **НОВЫЕ ТЭГИ**
- нет никаких тэгов оформления, **только данные**
- в контейнер могут быть вложены другие контейнеры  
иерархия – **дерево!**



Это иерархическая база данных!

# Использование XML

---

- документы *Microsoft Office* и *OpenOffice.org*:  
набор XML-файлов → **zip**-архив

 RSS (ленты новостей на сайтах и в блогах)

- *MathML* (описание математических формул)
- *SVG* (векторная графика на веб-страницах)
- файлы настроек (конфигурации) программ
- электронные книги **.fb2**
- XAML – язык описания интерфейсов в *Windows 8*

# XML: «за» и «против»

---




- открытый текстовый формат
- не зависит от ОС и ПО
- строгие правила, стандарты ⇒ легко обрабатывать в программах
- удобен для многоуровневых списков и иерархических БД



- сложно описать структуры, отличающиеся от иерархии (графы)
- не различаются типы данных (число, текст, дата, время, логическое значение)
- большой объем файлов

# XHTML

Задача: автоматическая обработка веб-страниц

**HTML:**  нестрогий синтаксис (можно не закрывать `<p>`)  
 не различаются заглавные и строчные буквы

**XHTML** = *eXtensible Hypertext Markup Language*

- имена тегов и атрибутов – только строчные буквы
- все тэги должны быть закрыты (например, `<br />`)
- правильная вложенность тэгов

```
<p><strong>Вася</p></strong>
```

- значения атрибутов в кавычках

```

```

- замена служебных символов:

«`<`» → «`&lt;`»

«`&`» → «`&amp;`»

HTML



XHTML



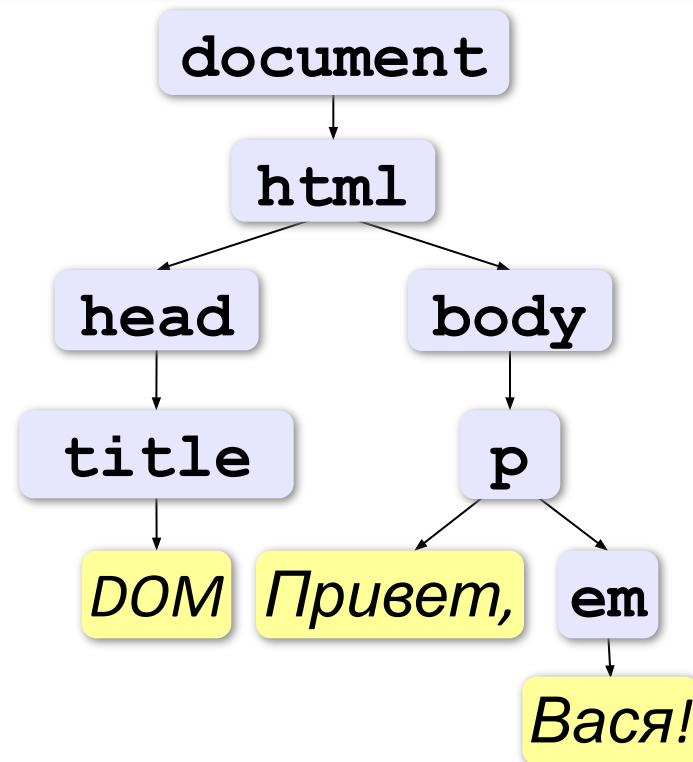
# Создание веб-сайтов

## § 29. Динамический HTML

# Что такое DHTML?

**Динамический HTML (DHTML)** – это технология создания интерактивных сайтов, использующая *HTML*, *CSS*, язык программирования (чаще всего *JavaScript*) и объектную модель документа (англ. *DOM = Document Object Model*).

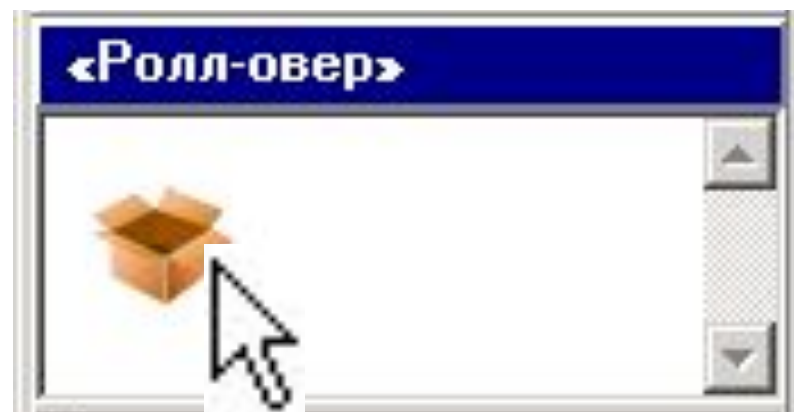
```
<html>
  <head>
    <title>DOM</title>
  </head>
  <body>
    <p>Привет,
      <em>Вася!</em></p>
  </body>
</html>
```



# «Живой» рисунок



box\_closed.gif



box\_opened.gif

```

```

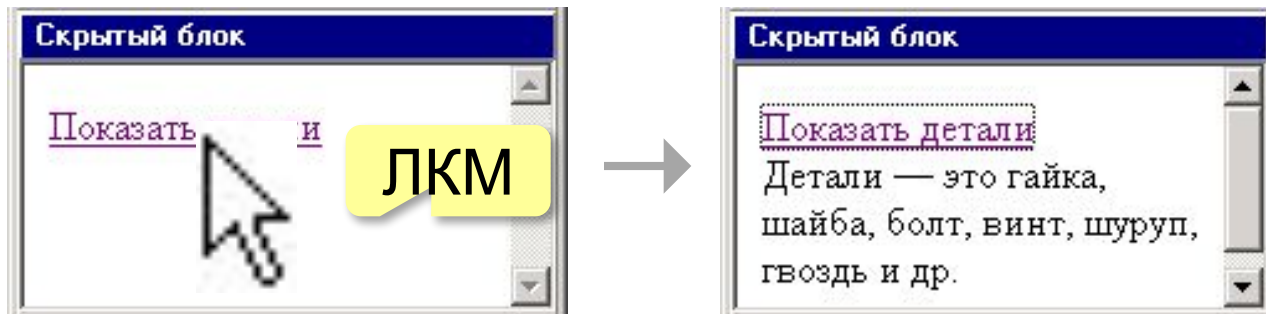
СВОЙСТВО **src**  
этого объекта

обработчик события  
(*Javascript*)

ВЛОЖЕННЫЕ  
КАВЫЧКИ

```
this.src='box_opened.gif'
```

# Скрытый блок



```
<div id="details" class="hidden">
Детали &#151; это гайка, шайба, болт, винт,
шуруп, гвоздь и др.
</div>
```

```
.hidden {
display:none;
}
```

не показывать



Зачем нужен **id**?

**block** — во всю ширину  
**inline** — в тексте



# Javascript-файл

Файл `test.js`

функция

имя функции

```
function show ( name )  
{  
    var elem = document.getElementById ( name );  
    if ( elem )  
        elem.style.display = "block";  
}
```

найти элемент по `id`

переменная

если нашли

ИЗМЕНИТЬ СВОЙСТВО  
`display`

`block` – во всю ширину  
`inline` – в тексте  
`none` – не показывать

# Подключение Javascript-файла

---

```
<head>  
  <script src="test.js"></script>  
  ...  
</head>
```

имя файла

# Вызов функции

---

ссылка на эту  
страницу

по щелчку вызвать  
функцию

```
<a href="#"  
  onClick="show('details');return false;">
```

Показать детали

```
</a>
```

дальнейшая  
обработка (переход  
по ссылке) не нужна

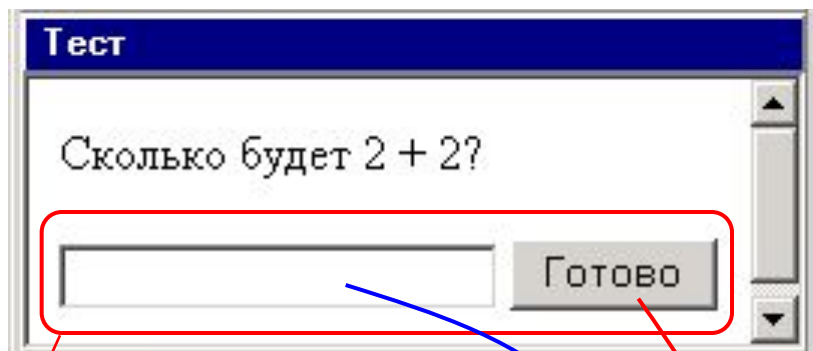
# HTML-файл

```
<html>
<head>
  <title>Скрытый блок</title>
  <script src="test.js"></script>
  <link rel="stylesheet" href="test.css">
</head>
<body>
  <a href=# onClick="show('details'); return false;">
    Показать детали</a>
  <div id="details" class="hidden">
    Детали &#151; это гайка, шайба, болт, винт,
    шуруп, гвоздь и др.
  </div>
</body>
</html>
```



Как сделать два скрытых блока?

# Формы



форма – набор элементов диалога

```
<form name="calc">
```

```
<input name="answer">
```

```
<input type="button"
```

```
value="Готово"
```

```
onClick="check();">
```

```
</form>
```

поле ввода

кнопка

текст на кнопке

что делать при щелчке

# Формы

```
<form name="calc">
  <input name="answer">
  <input type="button"
    value="Готово"
    onClick="check();">
</form>
```

Обработка события «щелчок мышью»:

```
function check()
{
  if ( calc.answer.value == "4" )
    alert("Правильно!");
  else alert("Неправильно!");
}
```

ВВЕДЁННЫЙ ТЕКСТ

В файл  
calc.js

# Создание веб-сайтов

## § 30. Размещение веб-сайтов

# Как разместить сайт?

---

## На своём компьютере:

- постоянно включенный компьютер
- стабильный скоростной канал связи
- «белый» IP-адрес;
- установка и настройка веб-сервера
- защита сайта от взломщиков, вредоносных программ и сетевых атак – самостоятельно



# Что такое хостинг?

**Хостинг** — услуга по размещению сайта (данных) на постоянно работающем сервере.

**Тарифный план** ⇒ стоимость

- ограничение объема **данных** (файлов и почты)
- ограничение **трафика**

ежедневно: 100 пользователей  
10 страниц по 100 Кбайт

≈ 100 Мбайт

в месяц: ≈ 3 Гбайт

- поддержка **баз данных** (да/нет/количество)
- поддержка серверных скриптов (**PHP**)

**Бесплатные хостинги**

оплата – реклама

- [ucoz.ru](http://ucoz.ru)
- [webservis.ru](http://webservis.ru)
- [www.hostinger.ru](http://www.hostinger.ru)

# Типы хостинга

## Виртуальный хостинг

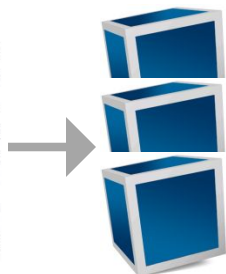


*garage.ru*  
*toy.com*  
*vobla.net*  
...

до 1000 сайтов

- копирование файлов в каталог
- ПО хостера

## Виртуальный частный сервер



*garage.ru*  
*toy.com*  
*vobla.net*

- виртуальная машина для каждого сайта
- ПО хозяина сайта

## Выделенный сервер



*vobla.net*

- отдельный компьютер для сайта
- ПО хозяина сайта

# Доменные имена

IP-адреса: 94.100.101.202

Доменные имена: www.mail.ru

DNS-сервер



Как зарегистрировать?

nic.ru  
(RU-CENTER)

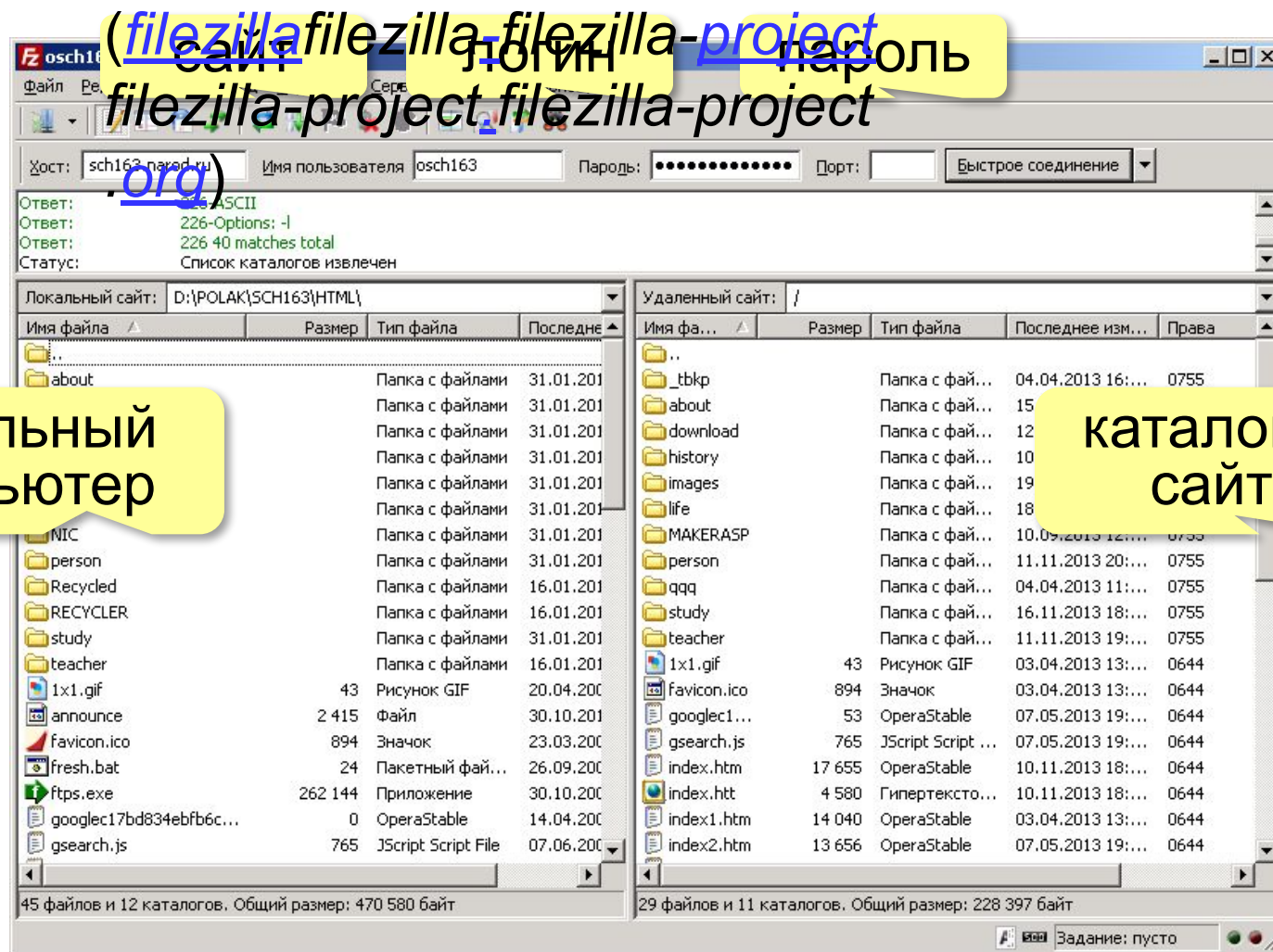
Домен		Цена
vasyapupkin.ru	занят	
vasyapupkin.com	занят	
vasyapupkin.net	занят	
vasyapupkin.biz	свободен	465 руб./год
vasyapupkin.me	свободен	650 руб./год
vasyapupkin.pro	свободен	570 руб./год
vasyapupkin.su	свободен	600 руб./год

Бесплатно (3-й уровень): vasya.**ucoz.ru**

# Загрузка файлов на сайт

- создание страниц в **конструкторе**: \*.ucoz.ru

- FTP  FileZilla



локальный компьютер

каталог на сайте

# Конец фильма

---

**ПОЛЯКОВ Константин Юрьевич**

д.т.н., учитель информатики

ГБОУ СОШ № 163, г. Санкт-Петербург

[kpolyakov@mail.ru](mailto:kpolyakov@mail.ru)

**ЕРЕМИН Евгений Александрович**

к.ф.-м.н., доцент кафедры мультимедийной

дидактики и ИТО ПГГПУ, г. Пермь

[eremin@pspu.ac.ru](mailto:eremin@pspu.ac.ru)

# Источники иллюстраций

---

1. [www.kkfreight.ru](http://www.kkfreight.ru)
2. [www.revelation-of-silence.com](http://www.revelation-of-silence.com)
3. [www.dejurka.ru](http://www.dejurka.ru)
4. [www.foru.ru](http://www.foru.ru)
5. [www.ibizsys.com](http://www.ibizsys.com)
6. [ru.wikipedia.org](http://ru.wikipedia.org)
7. иллюстрации художников издательства «Бином»
8. авторские материалы