

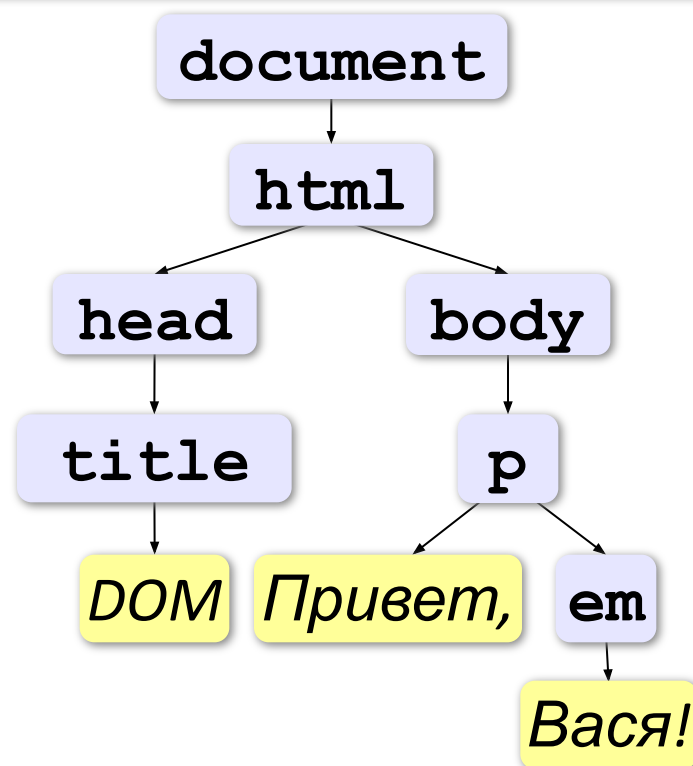
# Создание веб-сайтов

## § 31. Динамический HTML

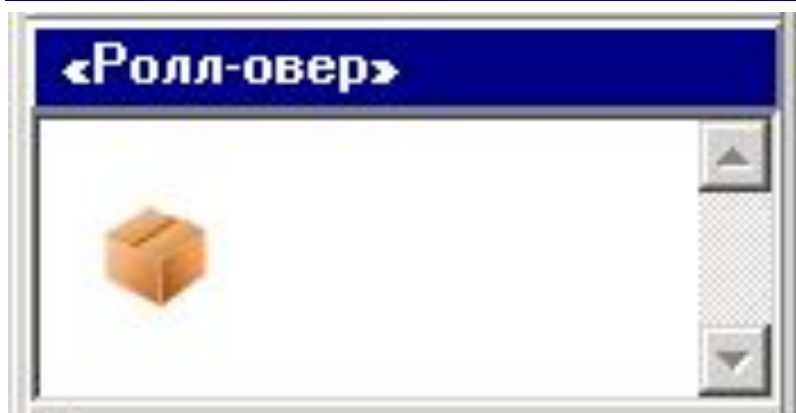
# Что такое DHTML?

**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*).

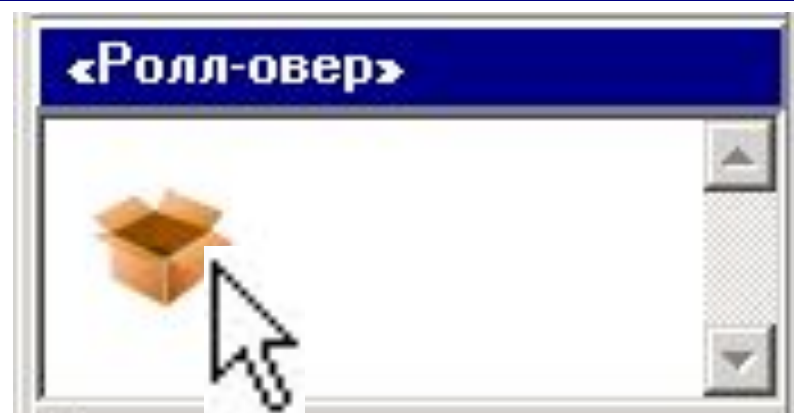
```
<html>
  <head>
    <title>DOM</title>
  </head>
  <body>
    <p>Привет,
      <em>Вася!</em></p>
  </body>
</html>
```



# «Живой» рисунок



box\_closed.gif



box\_opened.gif

```

```

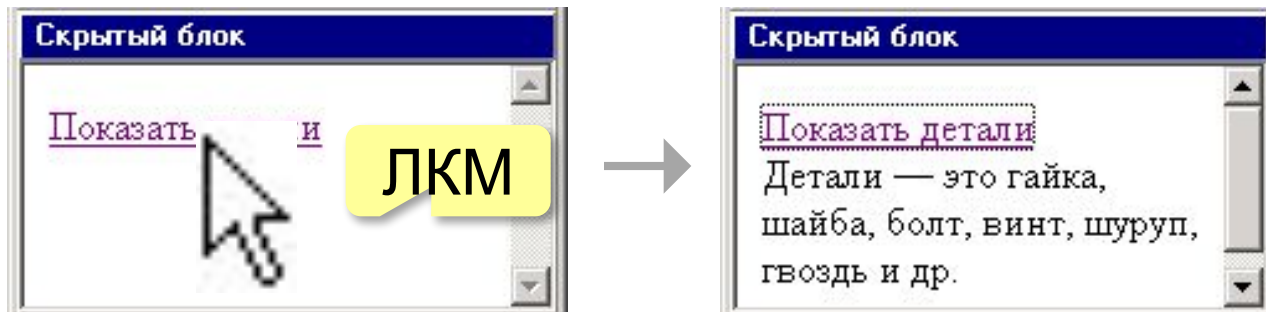
свойство **src**  
этого объекта

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 )
```

найти элемент по `id`

```
{  
  переменная
```

```
    var elem = document.getElementById(name) ;
```

```
    if ( elem )
```

если нашли

```
        elem.style.display = "block" ;
```

```
}
```

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

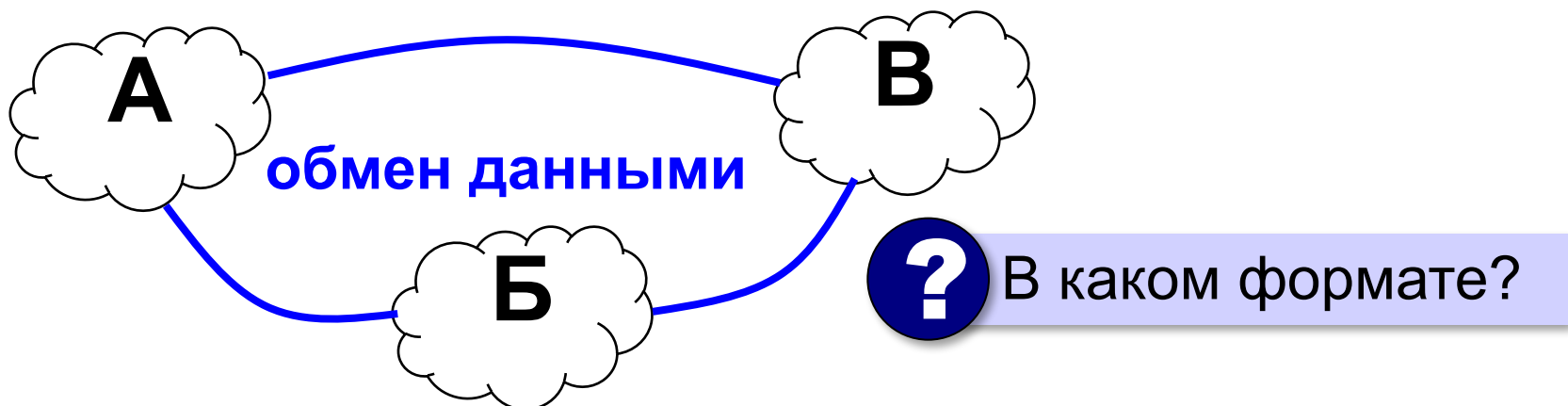
В файл  
calc.js

# Создание веб-сайтов

## § 32. XML и XHTML

# В чём проблема?

Интеграция (объединение) информационных систем



## Двоичные файлы:

- ⊕ ■ небольшой объём
- ⊖ ■ множество форматов, приёмник должен суметь прочесть 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  $\Rightarrow$  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** = e**X**tensible **H**ypertext **M**arkup **L**anguage

- 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

