Прошлое, настоящее и будущее

JavaScript

Дж. Дрансфилд, Р. Титов
Группа е-бизнеса отдела ИТ
ЦЕРН – Женева, Швейцария

Что такое JavaScript?

```
<html>
<script language="JavaScript" type="text/javascript">
  function runMe()
  {
    d = new Date();
    window.alert("Точное время: " + d.getHours() + ":" + d.getMinutes());
  }
</script>
<body onload="runMe()">
</body>
</html>
```

- **Применяется при разработке веб-страниц**
- **Выполняется на клиенте (внутри браузера)**
- **Интерпретируемый**

Немного истории

- 1995: создан Netscape
- 1996: перенесен на Internet Explorer (JScript)
- 1997: Стандартизация - ECMAScript

JavaScript

«Язык, чаще других понимаемый неправильно»

- **JavaScript похож на Java?**
- **Проблемы переносимости?**
- **Язык для непрофессионалов?**
- **JavaScript – язык ООП?**

JavaScript – ООП?

- **Классы/объекты**
- **Наследование**
- **Перегрузка методов**

JavaScript – классы/объекты

```
function Person(name, age, weight)
 {
   var m_name = name;
   var m_age  = age;
   var m_weight = weight;

   this.getName = function() { return m_name };
 }

 slava = new Person("P. Титов", 32, 95);

 alert(slava.getName());
```

this. – Описание и инициализация полей объекта

Описание методов

```
function MyObject(arg) {

    function method1(param) {
       alert(param);
    }

   method2 = function(param) { alert(param); }

   method3 = new Function("param", "alert(param)");

}
```

Видимость методов и переменных

Привилегированные (Priveleged)

```
function Foo(arg) {
        var test = 10;
        this.foo2 = function() {
                return test;
        }
}
var foo = new Foo(10);
foo.foo2();
```

- Переменные с префиксом this. видны извне (глобальные)
- В противном случае видимость ограничена скобками { }

Наследование и перегрузка методов

```
function Person(name, age) {
  var m_name = name;
  var m_age  = age;

  this.getName = function() { return m_name; }
  this.getAge  = function() { return m_age; }
}

function Student(name, age, group) {
  var m_group     = group;
  var m_prototype = new Person(name, age);

  this.getName = function() { return m_prototype.getName() + ", "  +m_group;
  this.getAge  = m_prototype.getAge;

  this.getGroup = function() { return m_group; }
}
```

Прототипирование

```
Student.prototype.payStipend = new function() { … };
```

- **Наследование**
- **Добавление новых методов**
- **Перегрузка существующих методов**
- **Работает и для стандартных объектов!**

Прототипирование - 2

```
d = new Date();

document.write(d.toString());   // Tue Oct 18 16:40:32 UTC+0200 2005

Date.prototype.toString = function() { return this.getHours() + "ч."; };

document.write(d.toString());   // 16ч.

// Object.prototype.doCoolStuff = function() { … };
```

- **Любому стандартному и нестандартному объекту можно добавить новые методы, либо заменить существующие**

Наследование и перегрузка - 2

```
function Person()                    // Объект Person
{
  this.getName = function() { return "A person" };
  this.getAge  = function() { return 20; };
}

function Student()                   // Объект Student
{
  this.getGroup = function() { return “K12-221”; }
}

Student.prototype = new Person();   // Student имеет все свойства Person
Student.prototype.getName = function() { return "A student"; }

s = new Student();
alert(s.getName());          // "A student"
alert(s.getGroup());         // "K12-221"
alert(s.getAge());           // "20"
```

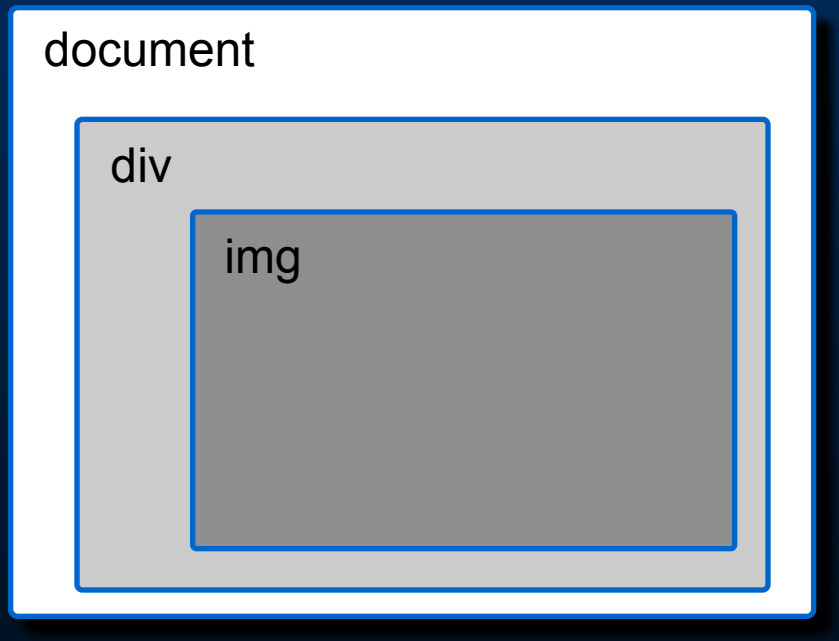
Малоизвестные возможности

- **try { … } catch (e) { … }**
- **X instanceOf Y**

```
try {
  x = new String();
  if (x instanceof String)
    throw "x is a String";
}
catch (e) {
  alert(e);
}
```

Объектная модель документа (DOM)

- **Разбор HTML**
  - **В памяти создается объектная модель**
  - **Доступна через JavaScript**



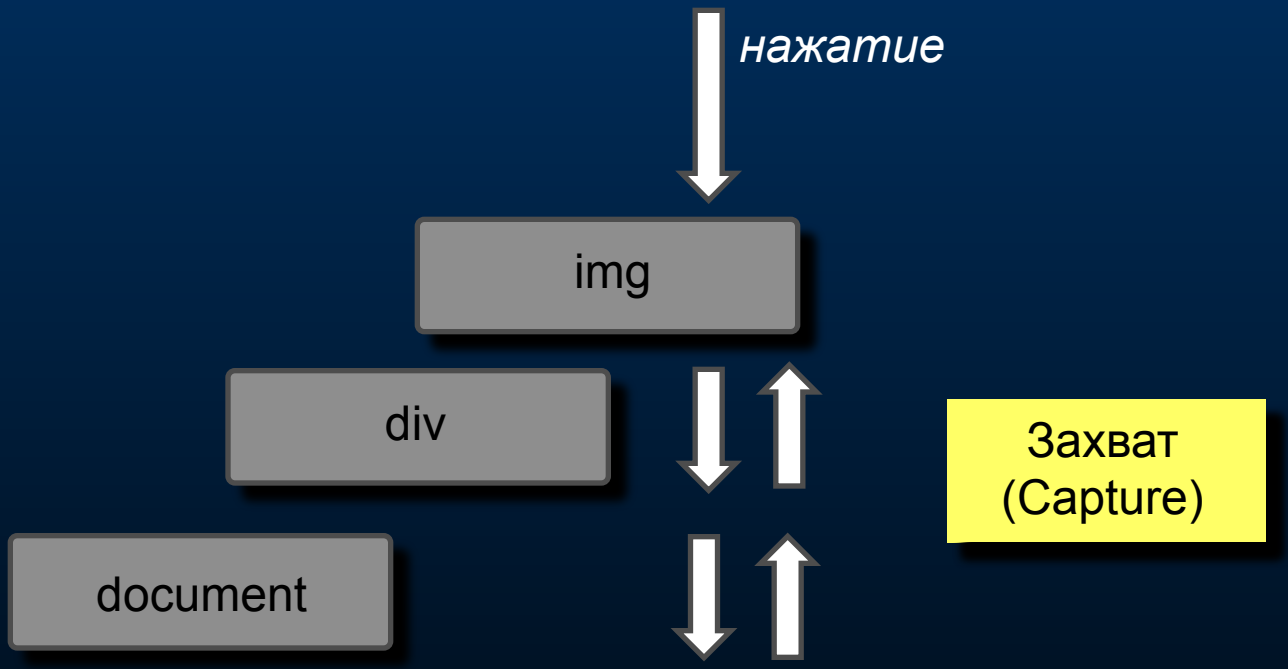
Методы работы с DOM

- element :: document.getElementById(id)
- element[] :: document.getElementByTagname(tagname)
- element :: document.createElement(elementName)
- void :: element.appendChild(element)
- void :: element.removeChild(element)

DHTML

- Через JavaScript можно управлять стилями CSS
  - element.style.height = “50px”;
  - background-color  backgroundColor

События



Регистрация обработчиков событий

- **Традиционный (кросс-платформенный)**
  - **<img onclick="doSomething()";**

  Надежно, работает везде, но требует написания кода в HTML
  - **element.onclick = doSomething;**

  Наиболее надежный метод, работает везде, лучший выбор сегодня
- **Новый стиль:**
  - **element.addEventListener(“click”, “doIt”, false);**

  рекомендация W3C, но не работает в IE 5 и 6
  - **element.attachEvent(“onclick”, “doSomething”);**

  стиль Microsoft, но мало где работает, лучше не использовать

Кросс-платформенность

- Проверка версии браузера
  - объект navigator:
    - appName
    - appVersion
- Проверка функциональности
  - например:
    - if (document.all) document.all.element_id;

Кросс-платформенность: события

- **Таблица совместимости для событий:**

http://www.quirksmode.org/js/events_compinfo.html

Пример:

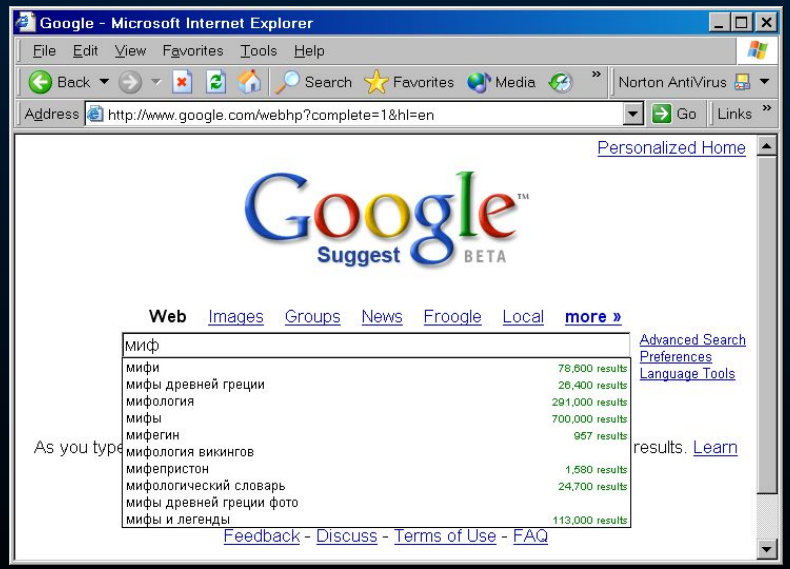
Question	Explorer 5 Windows	Explorer 6 Windows	Explorer 5.2 Mac	Mozilla 1.75	Safari 1.3	Opera 8	Netscape 4
Event accessing models							
How do I access the event?							
W3C/Netscape							Event accessing
	No	No	No	Yes	Yes	Yes	Yes
function doSomething(e) {code} e contains the event							
Microsoft							Event accessing
	Yes	Yes	Yes	No	Yes	Yes	No
window.event contains the event							
Cross browser							Event accessing
	Yes	Yes	Yes	Yes	Yes	Yes	Yes

```
function doSomething (e) {
        if (!e) var e = window.event;
}
```

Новые технологии

- **AJAX** (Asynchronous JavaScript and XML)
- Demo:

http://www.google.com/webhp?complete=1&hl=en



Новые технологии

- **Пользовательский интерфейс на XML**
- **Сложные элементы форм**
  - XUL, Mozilla Foundation
  - XSML, Microsoft
  - XForms, W3C
- Demo:

http://www.hevanet.com/acorbin/xul/top.xul

Контактная информация

Эта презентация:

http://sbnt.jinr.ru/iris

Для связи:

Rostislav_Titov@mail.ru