

JavaScript

История

- 1992 г. – Сmm (Си минус-минус)
 - Автоматическая работа с памятью
 - Слабая типизация
- ScriptEase
- Cenvi - Espresso Pages в Netscape Navigator
- LiveScript - внедрение Java-апплетов
- 4 декабря 1995 – JavaScript
 - Стандарт ECMAScript

JavaScript

- Состоит из 3-х частей
 - ядро (ECMAScript) – описание языка – синтаксис, семантика
 - объектная модель браузера – управление окнами, фреймами и т.п.
 - объектная модель документа – управление HTML и XML элементами документа

JavaScript – ядро

- Типы данных
 - Примитивные
 - числовой (англ. Number),
 - строковый (англ. String),
 - логический (англ. Boolean),
 - нулевой (англ. Null),
 - неопределённый (англ. Undefined).
 - Составной
 - Объектный (англ. Object)

JavaScript – ядро

- Очень похож на C
 - Имена: буквы, цифры, _, \$.
Имя не может начинаться с цифры.
 - Блок

```
{  
    ...  
}
```
 - ;
 - Описание переменных

```
var x = 2, y, z;
```
 - `if (<условие>) <выражение>`
[`else <выражение>`]

JavaScript – ядро

- Очень похож на C
 - Циклы
 - `while (<усл>)`
 <выражение>
 - `for (<иниц-р счет>; <усл>; <переход>)`
 <выражение>
 - `do`
 <выражение>
 `while (<усл>);`
 - `break;`
 - `return;`
 - `continue;`

JavaScript – ядро

- **Функции**

```
function ИМЯ([<парам1>, [<парам2>, ...]])  
{  
    ...  
    [return [<значение>]];  
}
```

- **Вызов**

```
var  
    bar = 2,  
    x = foo(bar);
```

JavaScript - ядро

- ```
function foo()
{
 var sum = 0;
 for (var i = 0; i < 42; i += 2)
 {
 var tmp = i + 2;
 sum += i * tmp;
 }
 alert(tmp);
 return sum;
}
foo();
```
- Переменная объявленная вне функции – глобальная область видимости

# JavaScript - ядро

- ```
function sum1(x1, x2)
{
  return x1 + x2;
}

var sum2 = function(x1, x2)
{
  return x1 + x2;
}

var sum3 = new Function(
  "x1",
  "x2",
  "return x1+x2;");
```

JavaScript - ядро

- Объявление переменной происходит в начале скрипта, присваивание значения – только при достижении выполнением `var`. До `var` она будет инициализирована значением *undefined*

`alert(sum(3, 4));` // 7: переменная `sum` к моменту выполнения этой строки уже создана и в качестве значения ей присвоена функция

```
function sum(arg1, arg2)
{
  return arg1 + arg2;
}
```

`alert(sum2(3, 4));` // ошибка: переменная `sum2` к моменту выполнения этой строки уже создана, но в качестве значения ей присвоено *undefined*

```
var sum2 = function(arg1, arg2)
{
  return arg1 + arg2;
};
alert(sum2(3, 4));
```

JavaScript – ядро

- Объекты (словари), ассоциированные массивы

```
var x = {  
  "foo": 5,  
};
```

- Динамическое определение свойств

```
x.foo = 2;  
x["foo"] = 7;
```

```
x.bar = 13;  
x.hoho = x["bar"] + x.foo;
```

JavaScript + HTML

- Внутри страницы:

```
<script type="text/javascript">  
    alert('Hello, World!');  
</script>
```

- Внутри элемента

```
<a href="delete.php"  
    onclick="return confirm('Вы уверены?');">  
    Удалить  
</a>
```

- В виде отдельного скрипта

JavaScript + HTML

- В виде отдельного скрипта, подключаемого к странице

```
<head>  
  <script  
    type="text/javascript"  
    src="http://Путь_к_файлу_со_скриптом">  
  </script>  
</head>
```

JavaScript - DOM

- `document` – главный элемент доступа к документу
- `document.write('МАМА!');`
- `alert("МАМА!");`

JavaScript - DOM

- ```
var p = document.createElement('p');
p.innerHTML = "МАМА!!!";
document.body.appendChild(p);
```
- В секции `body` документа будет добавлен новый параграф, содержимое которого **МАМА!!!**

# JavaScript - DOM

- document
  - .body – устанавливает или возвращает элемент body
  - .createElement() – создает новый элемент
  - .createAttribute() – создает новый атрибут
  - .getElementById() – возвращает элемент по значению атрибута id
  - .getElementsByTagName() – возвращает массив (список) элементов по тэгу
  - .getElementsByTagName() – возвращает массив (список) элементов по атрибуту name

# JavaScript - DOM

- `element`
  - `.id` – устанавливает или возвращает атрибут `id`
  - `.innerHTML` – устанавливает или возвращает содержимое элемента
  - `.textContent` – устанавливает или возвращает текстовое содержимое элемента (и его детей)
  - `.getAttribute()` – возвращает атрибут элемента

# JavaScript - DOM

- События
  - В HTML:
    - `<element onclick="myScript">`
  - В Javascript:
    - `element.onclick=function(){  
    myScript;  
};`
  - В Javascript, используя EventHandlers:
    - `element.addEventListener("click", function() {  
    myScript;  
});`