

Разработка Internet- и Web-приложений

Лекция 4. Изучаем JavaScript

Что такое JavaScript

JavaScript – это язык программирования для Веб

JavaScript является высокоуровневым, динамическим, нетипизированным и интерпретируемым языком программирования

Подключение JavaScript

```
<head>
```

```
  <title>Интеграция JavaScript</title>
```

```
  <!--Тег script-->
```

```
  <script type="text/javascript">
```

```
    alert("Hello world!");
```

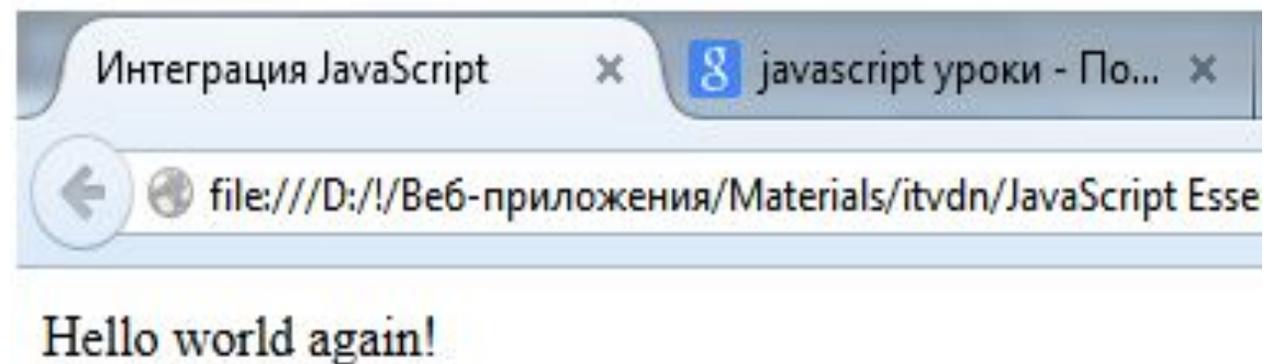
```
    document.write("Hello world again!");
```

```
  </script>
```

```
  <!--Завершение тега script-->
```

```
</head>
```

Подключение JavaScript



Подключение JavaScript

```
<head>
```

```
  <title>Интеграция JavaScript</title>
```

```
  <!--Тег script-->
```

```
  <script src="/js/script1.js"></script>
```

```
  <!--Завершение тега script-->
```

```
</head>
```

```
<body>
```

```
</body>
```

Что умеет JavaScript

- Создавать новые HTML-теги, удалять существующие, менять стили элементов, прятать, показывать элементы и т.п.
- Реагировать на действия посетителя, обрабатывать клики мыши, перемещение курсора, нажатие на клавиатуру и т.п.
- Посылать запросы на сервер и загружать данные без перезагрузки страницы(эта технология называется "AJAX").
- Получать и устанавливать cookie, запрашивать данные, выводить сообщения...и многое, многое другое!

Что НЕ умеет JavaScript

- JavaScript не может читать/записывать произвольные файлы на жесткий диск, копировать их или вызывать программы. Он не имеет прямого доступа к операционной системе.
- JavaScript, работающий в одной вкладке, не может общаться с другими вкладками и окнами, за исключением случая, когда он сам открыл это окно или несколько вкладок из одного источника (одинаковый домен, порт, протокол).
- Из JavaScript можно легко посылать запросы на сервер, с которого пришла страница. Запрос на другой домен тоже возможен, но менее удобен, т.к. и здесь есть ограничения безопасности.

Простая программа на JavaScript

```
var a, b;
```

```
var S , P;
```

```
a = parseFloat(prompt("Введите длину прямоугольника:", "0"));
```

```
b = parseFloat(prompt("Введите ширину прямоугольника:", "0"));
```

```
S = a * b;
```

```
P = 2 * (a + b);
```

```
alert("Площадь прямоугольника равна " + S);
```

```
alert("Периметр прямоугольника равен " + P);
```

Базовые типы: числа (Number)

- 128
- 10000000
- -57

- 7.56
- 3.754e7
- 3.7540000

Операторы

- $2 + 2$

- Операторы бинарные и унарные

- $+$

- $-$

- $*$

- $/$

- $\%$

- Приоритет исполнения

- $()$

```
Math.pow(2, 53) // => 9007199254740992: 2 в степени 53
Math.round(.6) // => 1.0: округление до ближайшего целого
Math.ceil(.6) // => 1.0: округление вверх
Math.floor(.6) // => 0.0: округление вниз
Math.abs(-5) // => 5: абсолютное значение
Math.max(x, y, z) // Возвращает наибольший аргумент
Math.min(x, y, z) // Возвращает наименьший аргумент
Math.random() // Псевдослучайное число x, где 0 <= x < 1.0
Math.PI //  $\pi$ : длина окружности / диаметр
Math.E // e: Основание натурального логарифма
Math.sqrt(3) // Корень квадратный из 3
Math.pow(3, 1/3) // Корень кубический из 3
Math.sin(0) // Тригонометрия: имеются также Math.cos, Math.atan и другие.
Math.log(10) // Натуральный логарифм 10
Math.log(100)/Math.LN10 // Логарифм 100 по основанию 10 (десятичный)
Math.log(512)/Math.LN2 // Логарифм 512 по основанию 2
Math.exp(3) // Math.E в кубе
```

Базовые типы: строки (String)

- Кавычки и апострофы
 - "Hello, world!"
 - 'Hello, world!'
- Спецсимволы
 - 'Hello, \n\tworld!'
 - 'Hello \'my\' world!'
 - 'format disk c:\ on computer'
 - 'Это \u2014\x20 просто строка'
- Конкатенация
 - 'Hello ' + 'mv' + ' world' + '!'

Базовые типы: булев (Boolean)

- Константы **true** и **false**

- `3 < 4`

- `3 > 4`

- `5e2 == 500`

- Другие операторы сравнения

- `===`

- `<=` и `>=`

- `!=` и `!==`

- Сравнение строк

- `'abc' < 'xyz'`

`"100" == 100` `true`

`"100" === 100` `false`

Логические операторы

- При проверке значения *операндов* преобразуются в *логические*
- *Внимание, приоритеты выполнения!*
 - `!` [логическое NOT (НЕ)]
 - `&&` [логическое AND (И)]
 - `||` [логическое OR (ИЛИ)]
- Число `0` и пустая строка преобразуются в **false**
 - `3 && 4`
 - `3 && 0`
 - `'Hello' && 2 || '' && 5`

Переменные

- Объявление переменной
 - `var num;`
- Инициализация переменной
 - `var user = 'John';`
 - `var x = 'a', y = 1, z;`
- Использование переменной
 - `num = 25;`
 - `num = num + 10;`
 - `num += 10;`

Манипуляции с типами

- Проблемы
 - `2 + '2' // '22'`
 - `2 + true // 3`
 - `' ' + true // 'true'`
 - `'x' * 10 // NaN! Что это???`
- Оператор определения типа
 - `typeof`
- Функции проверки
 - `isNaN()`
 - `isFinite()`

Приведение типов

- В число
 - `x = '5' * 1;`
 - `x = +'5';`
 - `x = parseInt('5.3');`
 - `x = parseFloat('5.3');`
 - `x = Number(true);`
- В строку
 - `x = 5 + '';`
 - `x = String(true);`
- В логический тип
 - `x = !!5;`
 - `x = Boolean(true);`

Тривиальные типы

- Пробуем:
 - `var result;`
 - `result = x;`
 - `ReferenceError: x is not defined`
 - `typeof x;`
- `undefined`
- `null`

Управляющие конструкции

- Циклы
- Цикл `while`
- Операторы инкремента и декремента
- Цикл `for`
- Цикл `do while`
- Управляющие конструкции `if – else if – else`
- Прерывание и продолжение цикла
- Управляющая конструкция `switch`

ФУНКЦИИ

- `function sayHello() {
 console.log('Hello world!');
}`
- `sayHello();`

- `function sayHello(name) {
 console.log('Hello '+name+'!');`
- `sayHello('John');`

Возврат значений

- ```
function sum(x, y) {
 return x + y;
}
```
- ```
result = sum(10, 20);
```
- ```
console.log(result);
```
- ```
console.log(sum(2, 3));
```

Области видимости

- `var x = "x-global"; var y = "y-global";`
- ```
function outerFunction() {
 var x = "x-local";
 console.log(x); //x-local
 function innerFunction(z) {
 console.log(x); //x-local
 console.log(y); //y-global
 console.log(z); //z-local
 y = "y-local";
 }
 innerFunction('z-local');
}
outerFunction();
console.log(x); //x-global
console.log(y); //y-local
```

# Использование переменных

- Функция как значение переменной

- `function foo(val){  
 return val + val;  
}`

- `var sum = foo;`

- `sum(2);`

- Анонимная функция

- `var sum = function (val){  
 return val + val;  
}`

- `sum(2);`

# Замыкания

- ```
var x = "global";  
function outerFunction() {  
    var x = "local";  
  
    function innerFunction() {  
        console.log(x);  
    }  
    return innerFunction;  
}
```
- ```
var inner = outerFunction();
inner();
```

# Объектный тип: Объект (Object)

- `var user = {};`
- `user.name = 'John';`
- `user.age = 25;`
- `user.admin = true;`
- `console.log(user.name);`

# Свойства объекта

- `var user = {  
 name: 'Guest',  
 age: 0,  
 admin: false  
};`
- `user.name = 'John';`
- `user.age = 25;`

# Свойства объекта: варианты

- `var user = {  
 'user name': 'Guest',  
 age: 0,  
 'var': false  
};`
- `user['user name'] = 'John';`
- `console.log(user['var']);`

# Сравнение и передача значений

- `var obj1 = {x: 10};`
- `var obj2 = obj1;`
- `var obj3 = {x: 10};`
  
- `obj1 == obj2; // true`
- `obj1 == obj3; // false`
  
- `obj1.x = 20;`
- `console.log(obj2.x); // 20`

# Методы объекта

- `var o = {  
 method_1: function(){  
 console.log('Hello, world!');  
 },  
 method_2: function(x){  
 console.log('Hello, '+x+'!');  
 }  
};`
- `o.method_1();`
- `o.method_2('John');`

# Что такое this

```
■ var user = {
 name: 'Guest',
 getName: function(){
 console.log(this.name);
 }
};
```

```
■ user.name = 'John';
```

```
■ user.getName();
```

# Объектный тип: Массив (Array)

- `var a = [];`
- `var x = 'moon';`
- `var y = function(){  
                    console.log('Hello');  
                    };`
- `var a = [10, "sun", x, y, true];`
- `a[1];`
- `a[3]();`
- `a[9] = 100;`

# Длина массива

- `var a = [1, 5];`
- `console.log(a.length); // 2`
  
- `a[23] = 11;`
- `console.log(a.length); // 24`
  
- `var a = [1, 5];`
- `a.length = 3; // [1, 5, undefined]`

# Получение строки из массива

- `var a = [1, 5, 7];`
- `var s = a.toString(); // 1,5,7`
- `var s = a.join(); // 1,5,7`
- `var s = a.join('---'); // 1---5---7`

# Сложение массивов

- `var a = [1, 5];`
- `var b = [11, 8];`
  
- `var arr = a.concat(3, b);`
  - `// [1, 5, 3, 11, 8]`
  
- `var s = [1, 5] + [11, 8];`
  - `// 1,511,8`

# Сортировка массива

- `var a = [14, 51, 7, 2];`
- `a.reverse();`
  
- `var a = [14, 51, 7, 2];`
- `a.sort();`
  - `// [14, 2, 51, 7] !?`
  
- `function mySort(a, b) {return a-b;}`
- `a.sort(mySort);`
  - `// [2, 7, 14, 51]`

# Работа с концом массива

- Исходный массив
  - `var a = [5, 'abc', 73];`
- Извлечение элемента
  - `var v = a.pop();`
    - `// [5, 'abc'], v = 73`
- Добавление элементов
  - `var v = a.push(12, 3);`
    - `// [5, 'abc', 12, 3], v = 4`

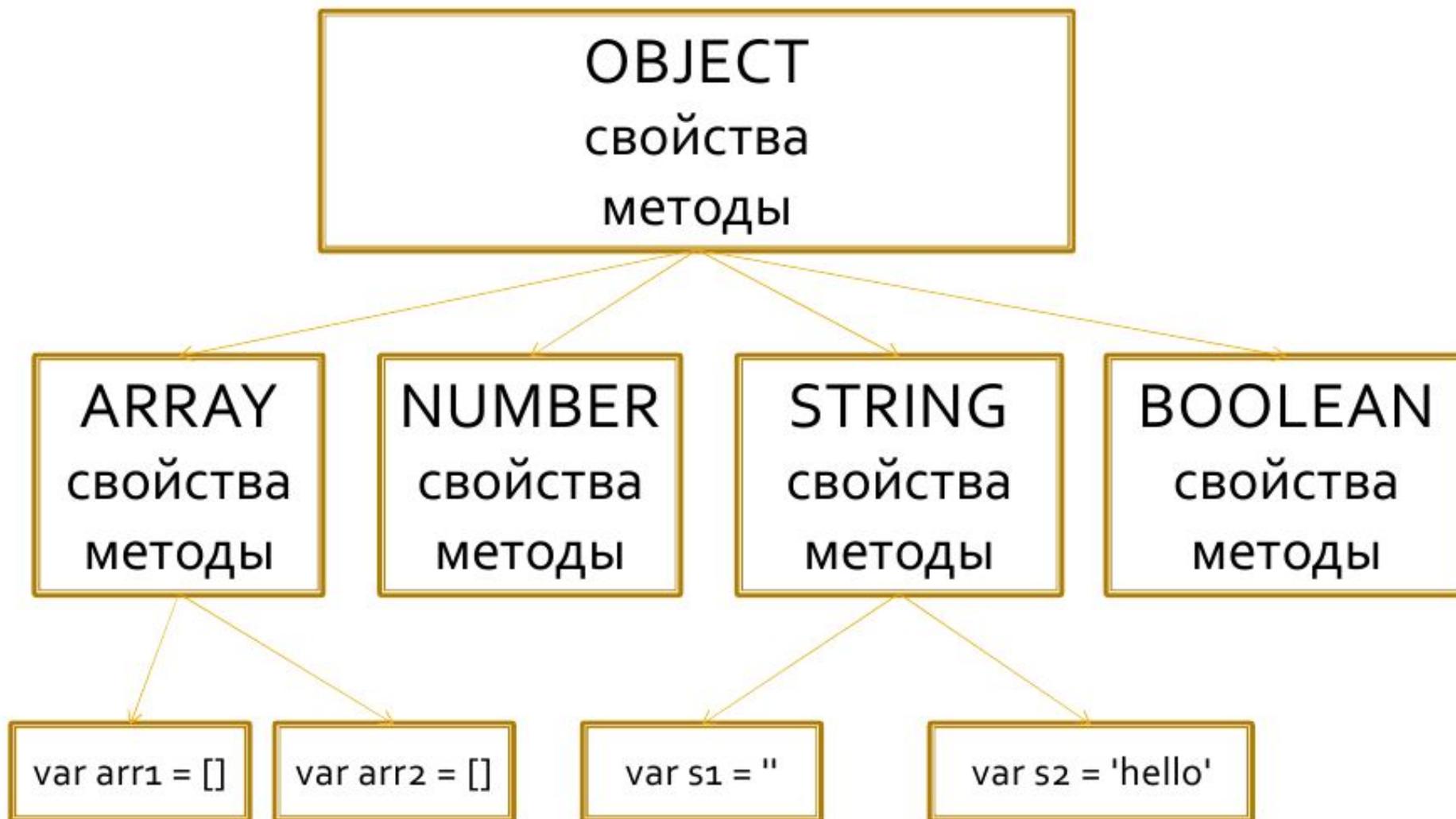
# Работа с началом массива

- Исходный массив
  - `var a = [5, 'abc', 73];`
- Извлечение элемента
  - `var v = a.shift();`
    - `// ['abc', 73], v = 5`
- Добавление элементов
  - `var v = a.unshift(12, 3);`
    - `// [12, 3, 'abc', 73], v = 4`

# Вставка и удаление в любом месте

- Исходный массив
  - `var a = [5, 'abc', 73, 12, 8];`
- `var arr = a.splice(1, 2);`
  - `// [5, 12, 8]`
  - `// arr = ['abc', 73]`
- `var arr = a.splice(1, 0, 3);`
  - `// [5, 3, 'abc', 73, 12, 8]`
  - `// arr = []`

# Что мы имеем?



# Преобразование числа в строку

- `toString()`
- `var n = 12345.6789;`
- `n.toFixed(2); // '12345.68'`
- `n.toExponential(2); // '1.23e+4'`

# Преобразование строки в число

- `var s = '37.5 км';`

- `var n = parseFloat(s); // 37.5`

- `var n = parseInt(s); // 37`

- `var s = '$99.9';`

- `var n = parseFloat(s); // NaN`

- `var n = parseInt(s); // NaN`

# Работа со строками

- `var s = 'просто пример';`
- `var x = s.charAt(4); // 'т'`
- `var x = s.slice(3, 6); // сто`
- `var x = s.substring(3, 6); // сто`
- `var x = s.slice(6); // пример`
- `var x = s.substring(6); // пример`