# Основы JavaScript

## Роль клиентского JavaScript

1. Пользовательский интерфейс:

* Создание визуальных эффектов.
* Выполнение несложных вычислений.
* Проверка пользовательских данных после ввода.
* Манипуляция данными, которые пользователь ввел в формы.
* Поиск по данным, встроенным в загружаемую страницу.
* Сохранение данных в cookies.

1. Динамическое изменение содержимого страницы (DHTML).
2. Прямое взаимодействие с сервером без перезагрузки всей страницы (AJAX).

Для интеграции JavaScript в страницу необходимо использовать тег <script>…</script>. Тег рекомендуется разместить в теге <head>…</head>. Внутри элемента <script> размещается JavaScript, код которого будет выполняться при загрузке страницы.

<!DOCTYPE html>

<html>

<head>

<title>Интеграция JavaScript</title>

<!--Тег script-->

<script type ="text/javascript">

alert("Hello world!");

document.Write("Hello world again!");

</script>

<!--Завершение тега script-->

</head>

<body>

</body>

</html>

Лексическая структура JavaScript:

* Чувствителен к регистру.
* Инструкции должны заканчиваться точкой с запятой и начинаться с новой строки.
* Игнорирует пробелы и табуляции.

JavaScript – динамически типизированный, а не строго типизированный язык программирования. Основным ключевым словом при объявлении переменной здесь является **var**.

NaN – не число.

Number.MAX\_VALUE – максимально допустимое значение в JavaScript.

Number.MIN\_VALUE – минимально допустимое значение в JavaScript.

Number.NaN – не число.

Number.POSITIVE\_INFINITY - + бесконечность.

Number.NEGATIVE\_INFINITY - - бесконечность.

var string1 = "Hello"; эквивалентно var string1 = 'Hello';

var string1 = "10";

var string2 = 2;

var string3 = string1 + string2 – результатом будет отображение с автоприведением string2 в строковый тип, то есть: 10 + 2.

При вычитании, умножении и делении первый аргумент переводится в целочисленный тип и выводится результат операции с целочисленными типами.

</br> - вывод с новой строки.

</hr> - выводит линию.

var a = undefined – переменная не инициализирована.

Любые данные, выводимые в document превращаются в строковый тип.

Тег <pre></pre> - сохраняет форматирование, заложенное разработчиком.

<!DOCTYPE html>

<!--

Метод prompt(1, 2) - принимает два аргумента:

1 - выводится в качестве простой строки в модальном окне.

2 - значение по умолчанию в текстовом поле для ввода.

Метод prompt(1, 2) - выводит сообщение в окне с текстовым полем и двумя кнопками: "ОК" и "ОТМЕНА".

Метод prompt(1, 2) - возвращает введенное значение или null, если пользователь нажал на кнопку "ОТМЕНА".

Как и в alert, prompt - модальное окно

-->

<html>

<head>

<title>Метод prompt()</title>

<meta charset="utf-8" />

<script>

var a = prompt("Введите какое либо значение", "сюда"); // блокирует выполнение скрипта.

alert(a);

</script>

</head>

<body>

</body>

</html>

<!DOCTYPE html>

<!--

confirm() - действует похоже на alert, за исключением того, что добавляет кнопку "Отмена" в диалоговое окно.

Возвращает true, если нажата кнопка ОК, или false, если нажата кнопка Cancel.

-->

<html>

<head>

<title>Метод confirm()</title>

<meta charset="utf-8" />

<script>

var conf = confirm("Вы уверены?");

document.write(conf);

</script>

</head>

<body>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<title>Пример применения методов</title>

<meta charset="utf-8" />

<script>

var x = prompt("Input x value", "0"); // x будет string типа

var y = prompt("Input y value", "0"); // y будет string типа

var result = x + y;

document.write(result + "<hr/>");

// Конвертация строкового значения в числовое:

// parseFloat() конвертация в вещественное значение.

// parseInt() конвертация в целочисленное значение.

// Данные функции возвращают значение NaN, если конвертация не удалась.

var integerX = parseInt(x); // paseInt находит в переданном значении цифры и выполняет с ними математические операции,

var integerY = parseInt(y); // только если цифры идут вначале.

result = integerX + integerY;

document.write(result);

</script>

</head>

<body>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<title>Арифметические операторы</title>

<meta charset="utf-8" />

<script>

var a = 9;

var b = 5;

var result;

// Сложение

result = a + b; // в данном случае + - арифметическое сложение.

document.write("a + b = " + result + "<br/>");

// Вычитание

result = a - b; // в случае если а или b нечисловые, оператор пытается преобразовать их в числа.

document.write("a - b = " + result + "<br/>");

// Умножение

result = a \* b;

document.write("a \* b = " + result + "<br/>");

// Деление

result = a / b;

document.write("a / b = " + result + "<br/>"); // деление на 0 дает +/- бесконечность. 0/0 = NaN.

// Деление по модулю

result = a % b;

document.write("a % b = " + result + "<br/>");

</script>

</head>

<body>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<title>Деление по модулю</title>

<meta charset="utf-8" />

<script>

var a = -8;

var string = "10";

// Унарный минус (-)

var result = -a; // Операция смены знака на противоположный.

document.write("-a = " + result + "<br/>");

// Унарный плюс (+)

// не изменяет знака при применении к числовому типу (ничего не происходит).

result = +a;

document.write("+a = " + result + "<br/>");

// Унарный плюс (+)

// при применении к строковому типу преобразовывает его в числовой.

result = +string + a;

document.write("+string +a = " + result + "<br/>");

</script>

</head>

<body>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<title>Инкремент</title>

<meta charset="utf-8" />

<script>

var a = 9;

a++;

document.write(a + "<br/>"); // выведет 10

document.write(a++ + "<br/>"); // выведет 10

document.write(a + "<br/>"); // выведет 11

document.write("<hr/>");

document.write(a + "<br/>"); // выведет 11

++a;

document.write(a + "<br/>"); // выведет 12

document.write(++a + "<br/>"); // выведет 13

document.write(a + "<br/>"); // выведет 13

document.write("<hr/>");

document.write(a + "<br/>"); // выведет 13

a--;

document.write(a + "<br/>"); // выведет 12

document.write(a-- + "<br/>"); // выведет 12

document.write(a + "<br/>"); // выведет 11

document.write("<hr/>");

document.write(a + "<br/>"); // выведет 11

--a;

document.write(a + "<br/>"); // выведет 10

document.write(--a + "<br/>"); // выведет 9

document.write(a + "<br/>"); // выведет 9

</script>

</head>

<body>

</body>

</html>

<!DOCTYPE html>

<!--

Оператор равенства "==" предназначен для сравнения значений двух переменных. При этом необходимо помнить

о следующих правилах:

1. Если одно значение null, а другое undefined - значения равны.

2. Если одно значение число, а другое строка, то строка преобразовывается в число и сравнивается.

3. Если одно значение true, оно преобразовывается в 1. Если false - в 0.

-->

<html>

<head>

<title>Оператор сравнения</title>

<meta charset="utf-8" />

<script>

var a = 1;

var b = "1";

var c = "some";

var d = "som";

var e = true;

var res;

res = a == b;

document.write("a равно b : " + res + "<br/>");

res = c == d;

document.write("c равно d : " + res + "<br/>");

res = a == e;

document.write("a равно e : " + res + "<br/>");

</script>

</head>

<body>

</body>

</html>

<!DOCTYPE html>

<!--

Идентичность - требует совпадения типа и значения переменной, когда для равенства достаточно

только совпадающих значений.

Оператор идентичности "===" предназначен для проверки значений двух переменных на идентичность.

При этом нужно помнить о следующих правилах:

1. Если значения имеют разные типы, они не идентичны.

2. Если значения являются числами, имеют одинаковые значения и не NaN - они идентичны.

3. Если две строки имеют одинаковую длину и содержат одни и те же символы - они идентичны.

4. Два одинаковых логических значения идентичны.

-->

<html>

<head>

<title>Операторы равенства</title>

<meta charset="utf-8" />

<script>

var a = 1;

var b = "1";

var c = "some";

var d = "some";

var e = true;

var f = 1;

var g = true;

var res;

res = a == b;

document.write("a равно b : " + res + "<br/>");

res = a === b;

document.write("a идентично b : " + res + "<br/>");

document.write("<hr/>");

res = a === f;

document.write("a идентично f : " + res + "<br/>");

document.write("<hr/>");

res = c === d;

document.write("c идентично d : " + res + "<br/>");

document.write("<hr/>");

res = f === g;

document.write("f идентично g : " + res + "<br/>");

document.write("<hr/>");

res = a != b;

document.write("a не равно b : " + res + "<br/>");

res = a !== b;

document.write("a не идентично b : " + res + "<br/>");

</script>

</head>

<body>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<title>Операторы</title>

<meta charset="utf-8" />

<script>

var a = +prompt("Введите первое число:", "сюда"); // +prompt преобразует а и b в число.

var b = +prompt("Введите второе число:", "сюда");

var res;

// Меньше

res = a < b;

document.write("a < b" + res + "<br/>");

// Больше

res = a > b;

document.write("a > b" + res + "<br/>");

// Меньше или равно

res = a <= b;

document.write("a <= b" + res + "<br/>");

// Больше или равно

res = a >= b;

document.write("a >= b" + res + "<br/>");

</script>

</head>

<body>

</body>

</html>

<!DOCTYPE html>

<!--

Сравнение строк производится посимвольно для числовых значений каждого символа из кодировки Unicode.

Символ в верхнем регистре считается меньше символа в нижнем регистре.

Например: A - code (41 Hex)

a - code (61 Hex)

-->

<html>

<head>

<title>Сравнение строк</title>

<meta charset="utf-8" />

<script>

var res = "A" < "a";

document.write("A < a = " + res + "<br/><hr/>");

var c = "hello";

var d = "world";

res = c < d;

document.write("hello < world = " + res + "<br/><hr/>");

</script>

</head>

<body>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<title>Логические операторы</title>

<meta charset="utf-8" />

<script>

var a = true;

var b = false;

var res = a && b;

document.write("a && b = " + res + "<br/>");

</script>

</head>

<body>

</body>

</html>

## Условные и циклические конструкции

При использовании **if** во всех случаях необходимо ставить не одинарное =, а двойное (==).

<!DOCTYPE html>

<html>

<head>

<title>Coffee Maker</title>

<meta charset="utf-8" />

<script>

var coffeesize = prompt("Размеры стаканчиков: 1 = Маленький, 2 = Средний, 3 = Большой.");

var cost = 0;

switch (coffeesize) {

case "1":

{

cost += 25;

document.write("<img src='coffee.jpg' width='100' height='100'/>");

}

break;

case "2":

{

cost += 50;

document.write("<img src='coffee.jpg' width='150' height='150'/>");

}

break;

case "3":

{

cost += 75;

document.write("<img src='coffee.jpg' width='256' height='256'/>");

}

break;

}

if (cost != 0) {

document.write("<p>Внесите " + cost + " копеек.<p>Спасибо!");

}

</script>

</head>

<body>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<title>Циклическая конструкция Do While</title>

<meta charset="utf-8" />

<script>

// Ctrl + K, C Ctrl + K,U - убрать/снять комментарий.

var input;

do {

input = prompt("Введите значение.");

document.write("<p>" + input);

}

while (input != null);

</script>

</head>

<body>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<title>Циклическая конструкция for</title>

<meta charset="utf-8" />

<script>

for (var i = 0; i < 3; i++) {

document.write("<p>Counter = " + i);

}

// i доступно за пределами цикла

document.write("<p>" + i);

</script>

</head>

<body>

</body>

</html>

## Массивы

<!DOCTYPE html>

<html>

<head>

<title>Массивы</title>

<meta charset="utf-8" />

<script>

var a = [1, 2, 3, 4, 5];

document.write(a); // Вывод элементов массива, при этом массив преобразуется в строку

var b = ["Jan", "Feb", "Mar", "Apr", "May"];

document.write("<hr/>");

document.write(a);

var c = [, , , ];

document.write("<hr/>");

document.write(c);

var d = ["Hello", 34, true];

document.write("<hr/>");

document.write(d);

</script>

</head>

<body>

</body>

</html>

<!DOCTYPE html>

<!--

1. Конструктор без аргументов создает пустой массив.

2. Конструктор с одним целочисленным аргументом создаст массив размерностью в количество элементов.

3. Конструктор с двумя и более аргументами - создает массив из передаваемых аргументов, при этом все аргументы

будут расположены в массиве, так как заданы в аргументах.

Такой подход использования массива не рекомендуется!

-->

<html>

<head>

<title>Конструктор Array</title>

<meta charset="utf-8" />

<script>

var a = Array(); // empty array

var b = Array(1, 2, 3, 4); // массив размерностью четыре элемента

document.write(b);

document.write("<hr/>");

var c = Array(10); // массив размерностью в десять элементов

c[0] = 1;

c[1] = 2;

c[2] = 3;

c[3] = 4;

c[4] = 5;

c[5] = 6;

c[6] = 7;

c[7] = 8;

c[8] = 9;

c[9] = 10;

c[15] = "Десять"; // происходит изменение размера массива

document.write(c + " " + c.length);

c.length = 20; // изменяется длина массива

</script>

</head>

<body>

</body>

</html>