

Лабораторная работа № 1. Арифметические, логические операторы, операторы сравнения в JavaScript.

Цель: изучить возможности ввода и вывода данных; особенности использования арифметических, логических операторов, операторов сравнения, операторов условного перехода.

Теория

Ввод и вывод данных.

JavaScript был создан в 1995 году как инструмент, предоставляющий веб-дизайнерам возможности программирования. Все современные браузеры имеют поддержку JavaScript. JavaScript встраивается прямо в веб-страницы и исполняется браузером во время их загрузки/

JavaScript добавляется на веб-страницы с помощью тэга `<script>`. В следующем примере на страницу выводится текст 'Я изучаю JavaScript'.

```
<script type='text/javascript'>
document.write('Я изучаю JavaScript.');
```

JavaScript код также может храниться во внешнем текстовом файле с расширением .js и подключаться к страницам с помощью тэга `<script>` следующим образом:

```
<script type='text/javascript' src='ex.js'></script>
```

JavaScript чувствителен к регистру букв.

В JavaScript существует два вида комментариев: короткие и длинные. Все, что находится после символа `//` до конца строки, будет являться коротким комментарием. Многострочные комментарии начинаются с `/*` и заканчиваются `*/`.

В JavaScript предусмотрены три стандартных метода для ввода и вывода данных: **alert()**, **prompt()** и **confirm()**.

Метод **alert()** позволяет выводить диалоговое окно (окно оповещения) с заданным сообщением и кнопкой ОК. Используется в случаях, когда необходимо, чтобы пользователь обязательно обратил внимание на определенную информацию.

Метод **confirm()** позволяет вывести диалоговое окно (окно подтверждения) с сообщением и двумя кнопками - ОК и Отмена (Cancel). Используется в случаях, когда необходимо, чтобы пользователь подтвердил или отклонил что-либо. Если пользователь щелкнул на кнопке ОК, то возвращается значение `true`; если щелкнул на кнопке Отмена, то возвращается значение `false`.

Метод **prompt()** позволяет вывести на экран диалоговое окно (окно запроса) с сообщением, а также с текстовым полем, в которое пользователь

может ввести данные. Используются в случаях, когда от пользователя необходимо получить определенную информацию. В этом окне предусмотрены две кнопки: ОК и Отмена (Cancel). В отличие от методов alert() и confirm() данный метод принимает два параметра: сообщение и значение, которое должно появиться в текстовом поле ввода данных по умолчанию. `

Метод **Console.log()** выводит сообщения в веб-консоль, а не на саму веб-страницу. Синтаксис: Console.log(obj1, obj2, ... , objN).

Пример применения методов для ввода и вывода данных:

```
<html>
<head>
<script type='text/javascript'>
function fun1() {
    alert('Н окно оповещения');
}
function fun2() {
    confirm('Н окно подтверждения');
}
function fun3() {
    x=prompt('Введите Ваше имя:', 'Имя'); document.write('Здравствуйте,
    '+ x);
}
</script>
</head>
<body>
<input type='button' value='Окно оповещения' onclick='fun1()' />
<input type='button' value='Окно подтверждения' onclick='fun2()' />
<input type='button' value='Окно запроса' onclick='fun3()' />
</body>
</html>
```

Типы данных.

Типы данных представлены в таблице 1.

Типы данных в Java Script

Таблица 1.

Тип данных	Примеры	Описание значений
Строковый или символьный (string)	'Привет' 'д.т. 123-4567'	Последовательность символов, заключенная в кавычки, двойные или одинарные
Числовой (number)	3.14 -567 +2.5	Число, последовательность цифр, перед которой может быть указан знак числа (+ или перед положительными числами не обязательно ставить знак '+'; целая и дробная части чисел разделяются точкой. Число записывается без кавычек
Логический (булевский, boolean)	true false	true (истина, да) или false (ложь, нет); возможны только два значения
Null		Отсутствие какого бы то ни было значения

Объект (object)		Программный объект, определяемый своими свойствами. В частности, массив также является объектом
Функция (function)		Определение функции - программного кода, выполнение которого может возвращать некоторое значение
Symbol		Уникальный идентификатор. Например: <pre>let id1 = Symbol('id'); let id2 = Symbol('id'); alert(id1 == id2); // false</pre>

Арифметические операторы

Арифметические операторы используются для выполнения арифметических операций над переменными или значениями.

Арифметические операторы

Таблица 2.

Оператор	Название	Пример
+	Сложение	X + Y
-	Вычитание	X - Y
*	Умножение	X * Y
/	Деление	X / Y
%	Деление по модулю	X % Y
++	Увеличение на 1	X++
--	Уменьшение на 1	Y--
**	Возведение в степень	X**Y

Чтобы уменьшить размер кода, можно использовать сокращенную запись арифметических операций. Например, вместо `x = x + y` писать `x += y`.

Операторы сравнения

Операторы сравнения представлены в таблице

Операторы сравнения

Таблица 3.

Оператор	Название	Пример
==	Равно	X==Y
!=	Не равно	X!=Y
>	Больше, чем	X>Y
>=	Больше или равно	X>=Y
<	Меньше, чем	X<Y
<=	Меньше или равно	X<= Y

Оператор 'равно' записывается с помощью двух символов без пробелов между ними.

Логические операторы

Логические данные, обычно получаемые с помощью элементарных выражений, содержащих операторы сравнения, можно объединять в более сложные выражения. Для этого используются логические (булевские) операторы: И (&&), ИЛИ (||), НЕ (!). Выражения с логическими операторами возвращают значение true или false.

Операторы условного перехода

Оператор условного перехода **if** позволяет реализовать структуру условного выражения если ..., то ..., иначе ...

Пример оператора if перехода:

```
if (sr_ball>=5)
{ document.write ('Стипендия есть');
}
else
{ document.write ('Стипендии нету');
}
```

В фигурных скобках располагается блок кода - несколько выражений. Если в блоке используется одна команда, то фигурные скобки можно не писать.

Оператор **Switch** (переключатель) удобно использовать, если требуется проверить несколько условий, которые не являются взаимоисключающими.

Пример оператора switch:

```
switch (sr_ball) {
case 5: document.write('Стипендия минимальная')
case 6: document.write('Стипендия с коэффициентом 1,2')
case 9: document.write('Стипендия с коэффициентом 1,4')
}
```

Числа в различных системах счисления

Шестнадцатеричное число начинается с **0x** или **0X**. Используются шестнадцатеричные цифры 0...9 a...f A...F. Перед числом допустимы знаки + или -:

```
var x = 0x123abc;
var y = -0XFFF;
```

Целые двоичные литералы: начинаем с **0b** или **0B** и используем цифры 0 и 1. Впереди допускается + или -.

Целые восьмеричные литералы: начинаем с **0o** или **0O** и используем цифры от 0 до 7. Впереди можно + или -.

Задания к лабораторной работе № 1

Задание 1. Опробовать работу с модальными окнами **alert**, **prompt**, **confirm**.

Разработать свой диалог с использованием этих команд. Пример диалога:

- вывести приветствие в окне alert “Вас приветствует учебный центр”;
- затем в окне prompt вывести сообщение ‘Введите имя’, а в окне alert вывести сообщение ‘Добро пожаловать на наши курсы, *имя*’;
- в окне confirm вывести сообщение ‘Хотите стать Web-дизайнером?’ с альтернативными ответами в окнах alert, если TRUE, то ‘Учите стили CSS и JavaScript!’ , если FALSE то ‘Упускаете время!’ .

Задание 2. Использование оператора + с числовыми и строковыми переменными.

1) сложить два любых числа (например, 10 и 5). Результат вывести в консоль;

2) сложить две строки (например, “10” и “5”). Результат вывести в консоль;

3) сложить число и строку (например, 22 и “5”). Результат вывести на страницу;

4) сложить строку и число (например, “22” и 5). Результат вывести на страницу;

5) вывод - *Результатом сложения строки и числа всегда будет ...* - вывести в окно.

Задание 3. Найти значение 2-х любых арифметических выражений, записанных на JS с использованием арифметических операций (например, $35y - 25x / 5 + 232$ и $8 * y / x + 5 * x / y - 43$)*6). Значения *x* и *y* задать произвольно, но подобрать такие, чтоб значение одного выражения было намного больше второго. Найти остаток от деления значения одного выражения на значение другого. Вывести результаты на страницу, а также в отдельное окно.

Задание 4. Продемонстрировать использование операторов сравнения (==, !=, >, >=, <, <=) и логических операторов (!, &&, ||).

Введите любое число. Если оно меньше 20 или больше 40 и не равно 15 и делится без остатка на 5, то вывести сообщение ‘Правильное значение’, иначе ‘Не правильное значение’ .

Придумайте свой пример с логическими операторами.

Задание 5. Конструкция if..elseif..else.

Ввести 2 числа с клавиатуры. Сравнить их и вывести сообщение на страницу, например “*A* больше *B*”, используя конструкцию if..elseif..else.

Написать свой пример с использованием альтернативного синтаксиса конструкции if..else (условие)?(команды1)!(команды2)

Задание 6. Определить, какой сегодня день недели (использовать конструкцию switch case).

Задание 7. Объявите переменные и присвойте им значения любых чисел в двоичной, восьмеричной и шестнадцатеричной системах счисления.

Задание 8*. Придумать пример, демонстрирующий обработку исключений с использованием конструкции try ... catch.