

Лабораторная работа №2 по JavaScript

JavaScript функции для работы с типами данных,
МЕТОДЫ ВВОДА И ВЫВОДА ДАННЫХ

Иногда в программе требуется преобразовать тип переменной.

Рассмотрим основные функции, связанные с преобразованием типов:

- **parseInt(строка, основание)**

преобразует указанную в параметре строку в целое число. Если есть второй параметр — система счисления — преобразование происходит по указанному основанию системы счисления (8, 10 или 16):

```
parseInt("3.14") // результат = 3
```

```
parseInt("-7.875") // результат = -7
```

```
parseInt("435") // результат = 435
```

```
parseInt("Вася") /* результат = NaN, то есть не является числом */
```

```
parseInt("15", 8) // результат = 13
```

- **parseFloat (строка)**

преобразует указанную строку в число с плавающей разделительной точкой (десятичной):

```
parseFloat("3.14") // результат = 3.14
```

```
parseFloat("-7.875") // результат = -7.875
```

```
parseFloat("435") // результат = 435
```

```
parseFloat("Вася") /* результат = NaN, то есть не является числом */
```

```
parseFloat("17.5") // результат = 17.5
```

Важно: Обе функции и `parseInt()` и `parseFloat()` используют в качестве параметра строку и преобразуют ее в число

- **Функция javascript isNaN(значение)**

служит для определения того, является ли значение выражения числом:

```
isNaN(123)           /* результат false (то есть это - число) */
```

```
isNaN("50 рублей") /* результат true (то есть это - не число) */
```

Иногда можно получить такую ошибку JavaScript — NaN — когда значение не является числом

(а ожидается число):

```
var a=8;
```

```
var b="pi";
```

```
var S=a*b; // S=NaN
```

В качестве значения переменной может быть **undefined** — не определено. Такое случается, когда переменная создана, но значение ей не присвоено:

```
var b;
```

```
document.write(b); // b=undefined
```

- Для того чтобы узнать текущий тип данных конкретной переменной в javascript используется функция **typeof**:

```
typeof 33 // возвращает строку "number",
```

```
typeof "A String" // возвращает строку "string",
```

```
typeof true // возвращает строку "boolean",
```

```
typeof null // возвращает строку "object"
```

Пример 2.1. Создайте сценарий:

- с двумя строковыми переменными со значениями "строка1" и "строка2";

с помощью операции конкатенации объедините строки;

- с двумя числовыми переменными со значениями 3.14 и FF; выведите их произведение, преобразованное в

целое число. Для вывода значений используйте метод `document.write()`.



```
конкатенация: строка1строка2
произведение: 800
```

Решение:

- Создайте веб-страницу с html-скелетом и тегом `script`:

```
<html><head></head>
```

```
<body>
```

```
<!-- Сценарий -->
```

```
<script>
```

```
    // для кода JavaScript
```

```
</script>
```

```
<!-- Конец сценария -->
```

```
</body></html>
```

- **В коде для javascript инициализируйте две строковые переменные:**

```
var c=3.14;
```

```
var d=0xFF; // 16-я система счисления
```

Вспомним, что для чисел 16-й системы используется префикс 0x.

- **Инициализируйте две числовые переменные:**

```
// Инициализация двух переменных:
```

```
var a="строка1";
```

```
var b="строка2";
```

- **Выведите результаты с помощью метода `document.write()` с несколькими параметрами, разделенными через запятую. Чтобы вывести в две строки следует применить тег `br`. Для преобразования в целый тип используйте функцию `parseInt()`:**

```
document.write("конкатенация: ",a+b,"<br/>произведение: ", parseInt(c*d));
```

- **Протестируйте сценарий в браузере.**

Задание Js 2.1: Даны переменные:

```
a="12";
```

```
b="7.15";
```

Найдите остаток от деления числовых значений переменной `a` на переменную `b`. Используйте функции преобразования типов. Результатом должно быть число 5.

Методы JavaScript вывода данных и ввода (диалоговые окна)

JavaScript может выдавать сообщения и запрашивать данные у пользователя с помощью трех диалоговых окон, вызываемых методами **alert**, **confirm** и **prompt**.

Рассмотрим работу с ними:

1. alert()

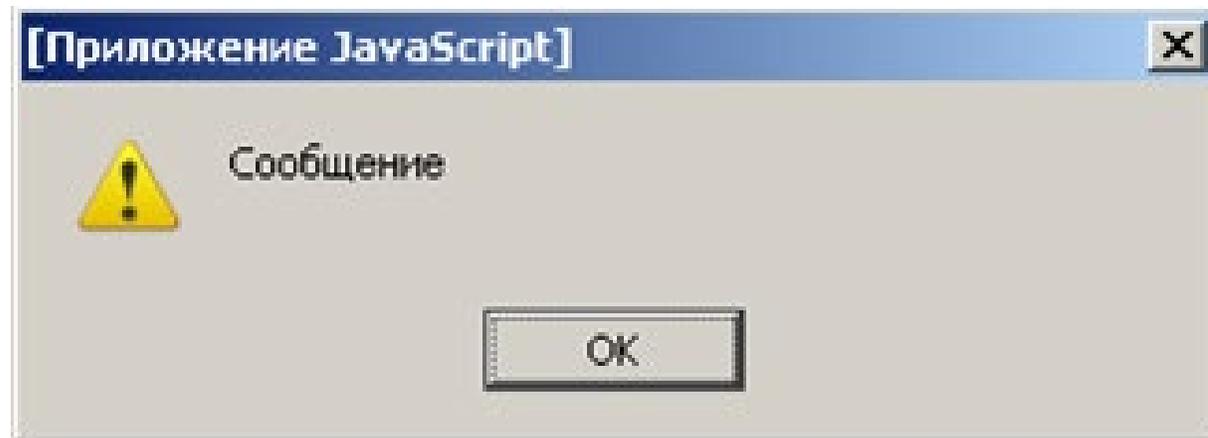
Синтаксис:

[window.]alert(сообщение)

С помощью уже знакомого нам модального окна с предупреждением на экран выводится информация для пользователя. Например:

alert("сообщение");

Пользователь читает сообщение и щелкает по кнопке ОК, чтобы закрыть это окно.



2. confirm()

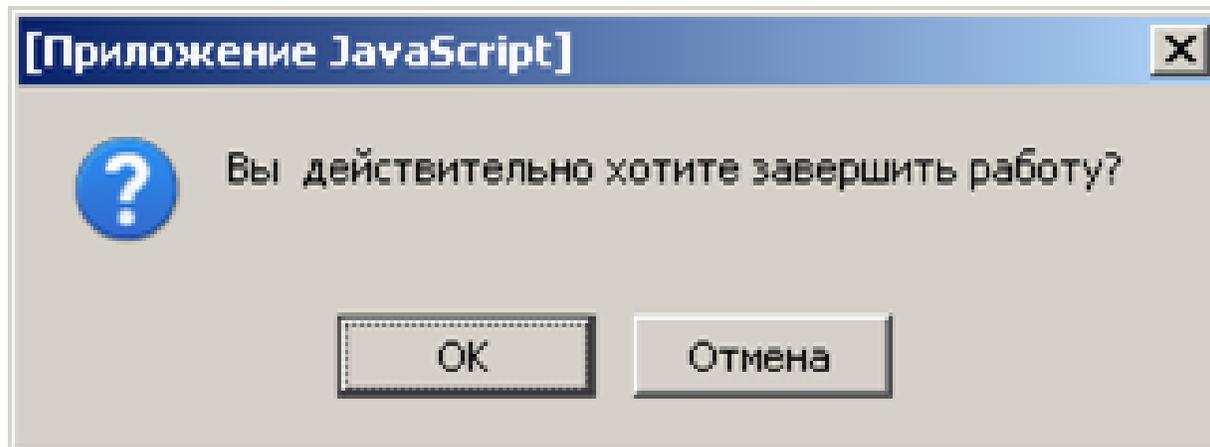
Синтаксис:

[window.]confirm(сообщение)

В отличие от предыдущего модального окна, в диалоговом окне **confirm** пользователь может выбрать один из двух вариантов: **ОК** или **Отмена**.

Важно: Если выбрана кнопка **ОК**, то в сценарий возвращается значение **true**, если **Отмена** — значение **false**.

```
confirm("Вы действительно хотите завершить работу?");
```



3. prompt()

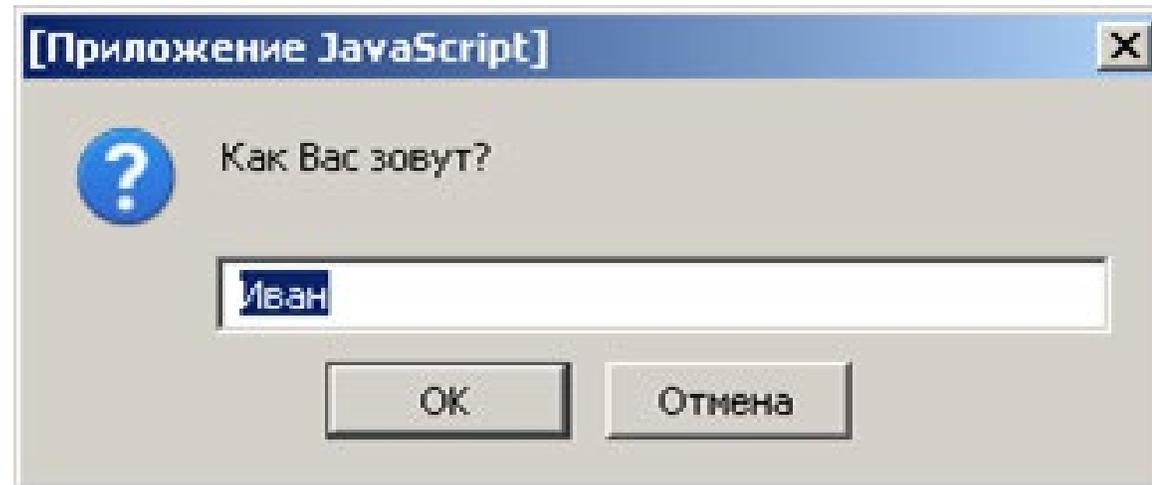
Синтаксис:

[window.]prompt(сообщение, значение_поля_ввода)

С помощью окна ввода, вызываемого методом **prompt**, пользователь может ввести данные, которые будут использоваться далее в сценарии.

prompt("Как Вас зовут?", "Иван");

Важно: Метод `prompt` принимает два аргумента: первый выводится в качестве простой строки в модальном окне; второй — значение по умолчанию в текстовом поле для ввода. Оба аргумента заключаются в кавычки.



Важно: Для решения заданий следует иметь в виду, что функция `prompt()` возвращает строковое значение, т.е. если требуется производить дальнейшие вычисления с введенным значением, то необходимо преобразовать тип данных

Важно: Во всех трех случаях использования методов ввода или вывода объект `window` можно опускать:

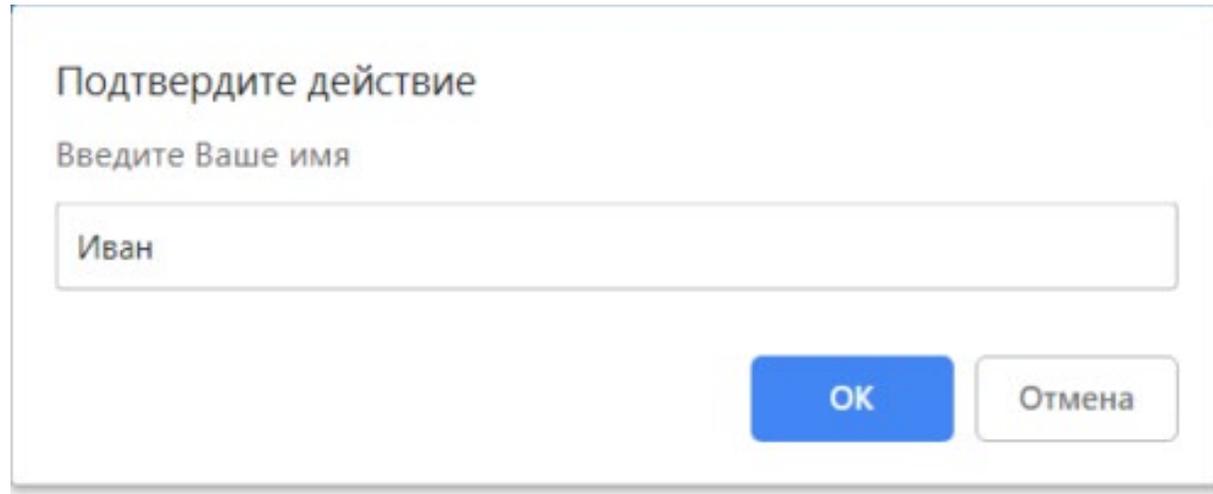
```
window.alert("сообщение"); // строгий синтаксис
```

```
alert("сообщение"); // нестрогий синтаксис
```

Этот объект означает «окно», и для краткости обращения к методам разрешено его не использовать, поэтому в синтаксисе он выделен квадратными скобками

Пример 2.2:

1. При загрузке страницы запросить у пользователя ввести имя (функция `prompt()`). С помощью метода `document.write` вывести на экран строку «Привет, введенное имя !».



Подтвердите действие
Введите Ваше имя

OK Отмена



Привет, Иван !

2. При помощи **функции prompt()** попросить пользователя ввести 2 числа. Выполнить произведение чисел и вывести результат на экран с помощью метода **document.write**.

Вывод должен иметь следующий вид $3 * 5 = 15$

Решение:

1 задание:

- Создайте веб-страницу с html-скелетом и тегом script:

```
<html><head></head>
```

```
<body>
```

```
<!-- Сценарий -->
```

```
<script>
```

```
    // для кода JavaScript
```

```
</script>
```

```
<!-- Конец сценария -->
```

```
</body></html>
```

- В коде для javascript инициализируйте переменную для присваивания ей значения — имени,

которое введет пользователь в модальное окно **prompt()**:

```
var name = prompt("Введите Ваше имя");
```

В результате данного присваивания в переменной name сохранится значение, которое введет пользователь в модальное окно.

- Выведите сохраненное значение с помощью метода **document.write()** с несколькими параметрами, разделенными через запятую:

```
document.write("Привет, ", name, " !");
```

- Протестируйте сценарий в браузере.

2 задание:

- Создайте веб-страницу с html-скелетом и тегом script.
- В коде для javascript инициализируйте две переменных для присваивания им значений — чисел, которое

введет пользователь в модальные окна **prompt()**. Для преобразования в целый тип используйте функцию **parseInt()**.

Вспомним, что метод **prompt()** возвращает строковое значение. Поэтому для дальнейшей работы с числами, значения необходимо преобразовать в числовой тип:

```
var a = parseInt(prompt("Введите первое число"));
```

```
var b = parseInt(prompt("Введите второе число"));
```

Преобразование также можно выполнить отдельным оператором:

```
var a = prompt("Введите первое число");
```

```
a = parseInt(a);
```

- Выполните произведение переменных, выдав результат с помощью метода **document.write()**.

Для «красивого вывода» используйте несколько параметров, перечисленных через запятую:

```
document.write(a, " * ", b, " = ", a*b);
```

- Протестируйте сценарий в браузере.

Задание Js 2.2. Допишите фрагмент кода для конвертации строкового значения в числовое. Для вывода результата используйте метод **document.write**:

```
var x = prompt("Введите значение x ", "0");  
var y = prompt("Введите значение y ", "0.0");  
var integerX = ...  
var floatY = ...  
var result = integerX + floatY;  
...;
```

Задание Js 2.3. Найдите ошибки в следующем фрагменте кода:

```
var firstName = prompt("Введите имя"; "сюда");  
var lastName = promt("Введите фамилию");  
int age = prompt("Введите возраст");  
alert(Age);
```

Задание Js 2.4.

Вычислить значение выражения по формуле (все переменные принимают вещественные значения):

$$\frac{x^2 - 7x + 10}{x^2 - 8x + 12}$$

Предлагаемый алгоритм:

- Запрос значения x
- Вычисление выражения
- Вывод результата при помощи метода `alert`

Условные операторы языка JavaScript

В JavaScript условие осуществляет **оператор if**. Рассмотрим синтаксис условного оператора:

if (условие)

```
{  
    // если истина  
    operator1;  
    operator2;  
}  
else  
{  
    // если ложь  
    operator3;  
    operator4;  
}
```

В качестве условия может использоваться логическое выражение или переменная.

Важно: Блок операторов заключенный в `{ }` — это составной оператор, объединяющий несколько операторов в конструкции. Скобки можно опустить, если внутри конструкции только один оператор.

Часть конструкции после служебного слова `else` — необязательна.

В условии используются следующие операции отношений:

`<` **Меньше**

`>` **Больше**

`<=` **Меньше либо равно**

`>=` **В javascript больше или равно**

`==` **Равно (сравнение)**

`!=` **в javascript не равно**

`===` **Сравнение с учетом типа (идентичность)**

Важно: Идентичность (`===`) требует совпадения типа и значения переменной, тогда как для равенства (`==`) достаточно только совпадающих значений

1. если значения имеют разные типы, то они не идентичны
2. если значения являются числами, имеют одинаковые значения и не являются значениями NaN

— они идентичны.

Для оператора равенства стоит использовать символ `==`

```
//num1 == num2
```

Для оператора идентичности стоит использовать `===`

```
//num1 === num2
```

Пример: выводить в модальное окно «а больше 1», если переменная $a > 1$, иначе выводить «а не больше 1»

```
var a=1;
```

```
if (a>1)
```

```
    alert("а больше 1")
```

```
else
```

```
    alert("а не больше 1");
```

Теперь рассмотрим используемые в javascript логические операторы для построения сложных (например, двойных) условий:

!

в javascript отрицание НЕ

if (!x)

&&

в javascript И

if (x>1) && (x<5)

||

в javascript ИЛИ

if (x>1) || (y>1)

Пример: Запрашивать у пользователя возраст. В зависимости от введенного возраста выдавать сообщение "Вы слишком молоды" или "Вы нам подходите". Показать решение

Задание Js 2.5.

Написать программу, которая по паролю определяет уровень доступа сотрудника к секретной информации в базе данных.

Доступ к базе данных имеют сотрудники, разбитые на три группы по уровням доступа. Они имеют следующие пароли:

- **9583, 1747 – доступны модули баз А, В и С;**
- **3331, 7922 - доступны модули баз В и С;**
- **9455, 8997 – доступен модуль базы С.**

Выводить доступные модули на запрос.

Задание Js 2.6. Найдите ошибки в следующем фрагменте кода:

```
var a = prompt("Введите число 11", "0");  
if (a = 11)  
    document.write("Введенное значение верно.");  
}
```

В большинстве языков программирования также как и в javascript используется так называемое "правило лжи".

Рассмотрим пример:

```
var a=1;  
if (a)  
    alert(a);
```

Как в данном примере поступит интерпретатор? как будет проверять условие, состоящее просто из одной переменной? - По правилу лжи:

ЛОЖНО:

- Null значения
- Undefined
- NaN
- 0
- Пустая строка ("")
- false

Пример со строковой переменной:

```
var s="";
```

```
if (s)
```

```
    alert("строка не пуста");
```

```
else
```

```
    alert("строка пуста");
```

Задание Js 2.7.

Запрашивать у пользователя имя (метод `javascript prompt`). Если имя введено, то выводить **"Привет, имя!"**. Если пользователь не ввел имя или щелкнул **cancel** (значение `null`) выводить **"Привет, незнакомец!"**

Задание Js 2.8.

Дописать код: объявлено 3 локальные переменные с использованием ключевого слова `var`. Нужно в переменную `max` присвоить значение следующего тернарного оператора: если `a` больше `b`, то возвращаем `a`, иначе возвращаем `b`

```
var a = 1, b = 2, max = 0;
```

```
...
```

```
document.write(max);
```

Задание Js 2.9. Запрашивать у пользователя имя (метод `javascript prompt`). Если имя введено, то выводить **"Привет, имя!"**. Если пользователь не ввел имя или щелкнул **cancel** (значение `null`) выводить **"Привет, незнакомец!"**

Тернарный оператор `JavaScript`

Важно: Тернарный оператор языка `javascript` - это сокращенная форма конструкции **`if ... else`**.

Данный оператор включает в себя три операнда:

(условие ? Если условие ИСТИННО : Если условие ЛОЖНО)

Т.е. синтаксис: **(логическое выражение) ? выражение 1 : выражение 2**

Пример: Переменной `b` присваивать значение `100` в случае истинности условия `a>1` и присваивать `200` в случае ложного условия

Показать решение:

Т.е. в случае истинности условия, выполняется то, что стоит до знака `:`, в случае ложности условия - выполняется то, что стоит после знака `:`.

Задание Js 2.10. Найдите и исправьте ошибки в следующем фрагменте кода:

```
var number = prompt("Введите число 1 или 2:");
switch (number) {
case "1"
{
    document.write("Один");
};
    break;
case "2"
{
    document.write("Два");
};
    break;
default
{
    document.write("Вы ввели значение, отличное от 1 и 2");
};
}
```

Задание Js 2.11.

Дописать код: объявлено 3 локальные переменные с использованием ключевого слова **var**.

Нужно в переменную **max** присвоить значение следующего тернарного оператора: **если a больше b, то возвращаем a, иначе возвращаем b**

```
var a = 1, b = 2, max = 0;
```

```
...
```

```
document.write(max);
```

Оператор переключения в JavaScript - switch

Оператор switch - служит для проверки переменной на множество значений:

```
switch (выражение) {
```

```
case вариант1:
```

```
    //..блок операторов..
```

```
    break
```

```
case вариант2:
```

```
    //..блок операторов..
```

```
    break
```

```
    [default:
```

```
        //..блок операторов..]
```

```
}
```

Блок, начинающийся со служебного слова **default** можно опустить. Операторы блока будут выполнены в случае, если ни одно из перечисленных значений не подходит.

Важно: Оператор **break** обязателен после каждого рассмотренного значения, если его не использовать, то будут выведены все, расположенные ниже операторы

Пример:

```
var a =2;
```

```
switch(a)
```

```
{
```

```
    case 0:
```

```
    case 1:
```

```
        alert("Ноль или один");
```

```
        break;
```

```
    case 2:
```

```
        alert("Два");
```

```
        break;
```

```
    default:
```

```
        alert("Много");
```

```
}
```

Пример:

Запрашивать у пользователя ввести цвет. Выводить перевод на английский язык введенного цвета

Показать решение:

Задание Js 2.12. У пользователя запрашивать число – количество ворон на ветке. В зависимости от введенного числа (не более 10), выводить сообщение. Для проверки использовать оператор Switch javascript:

- На ветке сидит 1 ворона
- На ветке сидит 4 вороны
- На ветке сидит 10 ворон

Задание Js 2.13. Найдите и исправьте ошибки в следующем фрагменте кода:

```
var number = prompt("Введите число 1 или 2:");
switch (number) {
case "1"
{
    document.write("Один");
};
    break;
case "2"
{
    document.write("Два");
};
    break;
default
{
    document.write("Вы ввели значение, отличное от 1 и 2");
};
}
```

Задание Js 2.14. Что выведется на экране при выполнении следующего кода?:

```
var value = "2";  
switch (value) {  
  case "1":  
  case "2":  
  case "3":  
    document.write("Hello");  
    break;  
  case "4":  
  case "5":  
    document.write("World");  
  default:  
    document.write("Error");  
}
```