

Публикация на тему

# Базовый синтаксис JavaScript

*В публикации рассматриваются основы JavaScript*

## Автор

[Михалькевич Александр Викторович](#)

## Публикация

**Наименование** Базовый синтаксис JavaScript

**Автор** А.В.Михалькевич

**Специальность** В публикации рассматриваются основы JavaScript,

**Анотация**

**Anotation in English**

**Ключевые слова**

**Количество символов** 7060

## Содержание

[Введение](#)

1 [Подключение кода JavaScript](#)

2 [Строгий и не строгий режимы](#)

3 [Типы данных](#)

4 [Автоматическое определение типов данных](#)

[Заключение](#)

[Список использованных источников](#)

[Приложения](#)

## Введение

### 1 Подключение кода JavaScript

JavaScript подключается к HTML-странице и выполняет код на стороне клиента (браузера).

2 способа подключения JavaScript к странице HTML:

```
<script>//JavaScript код</script>
```

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

## 2 Строгий и не строгий режимы

JavaScript-среды поддерживают две версии языка: строгий режим и не строгий режим.

Строгий режим включается в программе путем добавления в начале кода строковой константы:

```
"use strict"
```

Кроме того строгий режим может быть включен в функцию:

```
function f(a){  
  "use strict";  
}
```

Необходимо придерживаться либо *только строгого режима*, либо *только не строгого режима*. Т.к. смешивание режимов может привести к конфликтам.

## 3 Типы данных

JavaScript работает со следующими типами данных: null, undefined, String, Number, Boolean, Object, Array, Date, RegExp, Error, Function.

Все типы данных, кроме null и undefined, имеют свои классы-конструкторы. Это специальные встроенные функции, которые вызываются для создания экземпляров своего типа данных.

Узнать, к какому типу данных принадлежит значение переменной a, можно узнать с помощью оператора typeof:

```
typeof a;
```

Кроме перечисленных типов данных, JavaScript поддерживает еще один важный тип - глобальный объект, который неявно присутствует в каждой переменной.

## 4 Автоматическое определение типов данных

JavaScript автоматически определяет тип данных. Иногда это приводит к непредвиденным последствиям:

Сложение числа с булевым значением:

```
1+true; //2
```

При сложении строки с числом, JavaScript отдает предпочтение строке:

Сложение числа со строкой:

```
"2"+3; // 23
```

Для сравнения значений на равенство либо не равенство между собой используются операторы == (двойное равно) или === (тройное равно).

Оператор `==` сравнивает два значения без учета типов значений. Оператор `===` сравнивает два значения с учетом типов каждого значения. Если два аргумента относятся к одному типу, между ними нет никакой разницы. Тем не менее, стоит избегать использования оператора со смешанными типами (`==`). Сравнение двойным равно может привести к неожиданным результатам. Например, код из следующего листинга вернет `true`:

Сравнение аргументов двойным равно, которое вернет `true`:

```
"1.0e0" == { valueOf: function(){return true;} }
```

Чтобы поведение программы было понятнее при сравнении значений разных типов необходимо использовать явное приведение типов данных и использовать оператор `===`.

Переменная, значением которой окажется `null`, не вызывает сбой при арифметическом вычислении, она будет преобразована в `0`.

Неопределенная переменная преобразуется в специальное значение с плавающей точкой `NaN`. Причем, `NaN` рассматривается как неравное самому себе.

Проверка значения на равенство `NaN` не работает:

```
var x = NaN
x === NaN; //false
```

Протестировать переменную на значение `NaN` можно путем проверки ее на неравенство самой себе:

Тестирование переменной на значение `NaN`:

```
var x = NaN
x !== x; // true
```

Можно создать специальную функцию:

Функция тестирования переменной на значение `NaN`:

```
function isNaN(x){
  return x !== x;
}
```

Еще одна разновидность приведения типов данных называется истинностью. Операторы `if`, `||`, `&&` работают с булевыми значениями, но на самом деле принимают любые. Большинство значений JavaScript являются истинными, т.е. неявным образом приводятся к `true`. Существует лишь 7 ложных значений:

`false`, `0`, `-0`, `""`, `NaN` и `undefined`

Все остальные значения истинны.

## Заключение

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. [Печатное издание] **JavaScript в Web-дизайне** В. Дронов. - М.: СПб: БХВ, 2014. - 880 с.
2. [Печатное издание] **Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript** Робин Никсон. - М.: Питер, 2013. - 496 с.
3. [Печатное издание] **JavaScript для всех** А.Г. Федоров. - М.: Машиностроение, 2012. - 384 с.

## **Приложения**