var myVariable;

Примечание: Все инструкции в JavaScript должны заканчиваться точкой с запятой, чтобы указать, где заканчивается эта инструкция. Если вы не добавите их, вы можете получить неожиданные результаты.

| **Переменная** | **Пояснение** | **Пример** |
| --- | --- | --- |
| [**String**](https://developer.mozilla.org/ru/docs/%D0%A1%D0%BB%D0%BE%D0%B2%D0%B0%D1%80%D1%8C/%D0%A1%D1%82%D1%80%D0%BE%D0%BA%D0%B0) | Строка текста. Чтобы показать, что переменная является строкой, вы должны заключить ее в кавычки. | var myVariable = 'Bob'; |
| [**Number**](https://developer.mozilla.org/ru/docs/%D0%A1%D0%BB%D0%BE%D0%B2%D0%B0%D1%80%D1%8C/Number) | Числа. Числа не имеют кавычек вокруг них. | var myVariable = 10; |
| [**Boolean**](https://developer.mozilla.org/ru/docs/%D0%A1%D0%BB%D0%BE%D0%B2%D0%B0%D1%80%D1%8C/Boolean) | Значение True(Правда)/False(Ложь). Словаtrue и false специальные ключевые слова в JS, и не нуждаются в кавычках. | var myVariable = true; |
| [**Array**](https://developer.mozilla.org/ru/docs/Glossary/Array) | Структура, которая позволяет хранить несколько значений в одной ссылке. | var myVariable = [1,'Bob','Steve',10]; Обратиться к каждому элементу массива можно так: myVariable[0], myVariable[1], и т.д. |
| [**Object**](https://developer.mozilla.org/ru/docs/Glossary/Object) | В принципе что угодно. Все в JavaScript является объектом, и может храниться в переменной. Держите это в памяти, когда вы это узнаете. | var myVariable = document.querySelector('h1'); Все это из вышеприведенных примеров. |

Операторы

[operator](https://developer.mozilla.org/ru/docs/Glossary/operator) это математический символ, который производит результат, основанный на двух значениях (или переменных). В приведенной ниже таблице вы можете увидеть некоторые из наиболее простых операторов, наряду с некоторыми примерами, которые опробуйте в JavaScript консоли.

| **Оператор** | **Пояснение** | **Символ(ы)** | **Пример** |
| --- | --- | --- | --- |
| **сложение/конкатенация** | Используется для сложения двух чисел или склеивания двух строк вместе. | + | 6 + 9; "Hello " + "world!"; |
| **вычитание, умножение, деление** | Они делают то, что вы ожидаете от них в математике. | -, \*, / | 9 - 3; 8 \* 2; // умножение в JS это звездочка 9 / 3; |
| **оператор присваивания** | Вы уже видели это: он присваивает значение переменной. | = | var myVariable = 'Bob'; |
| **тождественный оператор** | Делает проверку, если увидит что два значения равны друг другу то возвращает true/false(Boolean) результат. | === | var myVariable = 3; myVariable === 4; |
| **отрицание, неравенство** | Возвращает логически противоположное значение, которое этому предшествует; превращает true в false, и т.д. Когда это используется вместе с оператором равенства, оператор отрицания проверяет, являются ли два значения *не* равными. | !, !== | Основное выражениеtrue, но сравнение возвращаетfalse, потому что мы отрицаем его:  var myVariable = 3; !(myVariable === 3);  Здесь мы проверяем "myVariableНЕ равно 3". Это возвращаетfalse, потому чтоmyVariableравно 3.  var myVariable = 3; myVariable !== 3; |