Число “Number”

var n = 123;

n = 12.345;

Единий тип числ каторий используется как для целых так и для дробных чисел

Существуют специальные числовые значения Infinity (бесконечность) и NaN (ошибка вычислений)

alert( 1 / 0 ); // Infinity

Ошибка вичислений NaN будет результатом некоректной математической операции, например:

alert( "нечисло" \* 2 ); // NaN, ошибка

Строка “string”

var str = "Мама мыла раму";

str = 'Одинарные кавычки тоже подойдут';

**[Булевый (логический) тип «boolean»](http://learn.javascript.ru/types-intro" \l "%D0%B1%D1%83%D0%BB%D0%B5%D0%B2%D1%8B%D0%B9-%D0%BB%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9-%D1%82%D0%B8%D0%BF-boolean)**

У него два значения true (истина) и false (ложь)

var checked = true; // поле формы помечено галочкой

checked = false; // поле формы не содержит галочки

## [Специальное значение «null»](http://learn.javascript.ru/types-intro" \l "%D1%81%D0%BF%D0%B5%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5-%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%B8%D0%B5-null)

Значение null не относится ни к одному из типов выше, а образует свой отдельный тип, состоящий из единственного значения null:

var age = null;

## [Специальное значение «undefined»](http://learn.javascript.ru/types-intro" \l "%D1%81%D0%BF%D0%B5%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5-%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%B8%D0%B5-undefined)

Значеник “undefined” как и null образует собствений тип состоящего значения

Если переменная объявлена, но в неё ничего не записано, то её значение как раз и есть undefined

var x;

alert( x ); // выведет "undefined"

Можно присвоить undefined и в явном виде, хотя это делается редко

var x = 123;

x = undefined;

alert( x ); // "undefined"

## [Объекты «object»](http://learn.javascript.ru/types-intro" \l "%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D1%8B-object)

Особняком стоит шестой тип: «объекты».

Он используется для коллекций данных и для объявления более сложных сущностей.

Объявляются объекты при помощи фигурных скобок {...}, например:

var user = { name: "Вася" };

## [Оператор typeof](http://learn.javascript.ru/types-intro" \l "type-typeof)

Оператор typeof возвращает тип аргумента.

У него есть два синтаксиса: со скобками и без:

Синтаксис оператора: typeof x.

Синтаксис функции: typeof(x).

typeof undefined // "undefined"

typeof 0 // "number"

typeof "foo" // "string"

typeof {} // "object"

typeof null // "object" (1)

typeof function(){} // "function" (2)

Последние две строки помечены, потому что typeof ведет себя в них по-особому.

1. Результат typeof null == "object" – это официально признанная ошибка в языке, которая сохраняется для совместимости. На самом деле null – это не объект, а отдельный тип данных.
2. Функции мы пройдём чуть позже. Пока лишь заметим, что функции не являются отдельным базовым типом в JavaScript, а подвидом объектов. Но typeof выделяет функции отдельно, возвращая для них "function". На практике это весьма удобно, так как позволяет легко определить функцию.