

JSON

JSON

JSON (JavaScript Object Notation) - простой формат обмена данными, удобный для чтения и написания как человеком, так и компьютером. Он основан на подмножестве [языка программирования JavaScript](#), определенного в [стандарте ECMA-262 3rd Edition - December 1999](#).

JSON

JSON-текст представляет собой (в закодированном виде) одну из двух структур:

- Набор пар *ключ: значение*. В различных языках это реализовано как *объект*, *запись*, *структура*, *словарь*, *хэш-таблица*, *список* с ключом или *ассоциативный массив*. Ключом может быть только строка, значением — любая форма.
- Упорядоченный набор *значений*. Во многих языках это реализовано как *массив*, *вектор*, список или *последовательность*.

JSON

В качестве значений в JSON используются структуры:

- **Объект** — это неупорядоченное множество пар **ключ:значение**, заключённое в фигурные скобки «{ }». Ключ описывается **строкой**, между ним и значением стоит символ «:». Пары *ключ-значение* отделяются друг от друга запятыми.
- **Массив** (одномерный) — это упорядоченное множество **значений**. Массив заключается в квадратные скобки «[]». Значения разделяются запятыми.
- **Значение** может быть **строкой** в двойных кавычках, **числом**, **объектом**, **массивом**, одним из **литералов**: *true*, *false* или *null*. Т.о. структуры могут быть вложены друг в друга.
- **Строка** — это упорядоченное множество из нуля или более символов **юникода**, заключённое в двойные кавычки. Символы могут быть указаны с использованием **escape-последовательностей**, начинающихся с **обратной косой черты** «\».

JSON

```
{
  "firstName": "Иван",
  "lastName": "Иванов",
  "address": {
    "streetAddress": "Московское ш., 101, кв.101",
    "city": "Ленинград",
    "postalCode": 101101
  },
  "phoneNumbers": [
    "812 123-1234",
    "916 123-4567"
  ]
}
```

JSON

Объекты JSON отличаются от обычных JavaScript-объектов более строгими требованиями к строкам — они должны быть именно в двойных кавычках.

В частности, первые два свойства объекта ниже — некорректны:

```
1 {
2   name: "Вася",      // ошибка: ключ name без кавычек!
3   "surname": 'Петров', // ошибка: одинарные кавычки у значения!
4   "age": 35          // .. а тут всё в порядке.
5   "isAdmin": false   // и тут тоже всё ок
6 }
```

JSON

Последние версии веб-браузеров имеют встроенную поддержку JSON и способны его обрабатывать без вызова функции `eval()`, приводящей к описанной проблеме. Обработка JSON в таком случае обычно осуществляется быстрее.

- Mozilla Firefox 3.5+
- Microsoft Internet Explorer 8
- Opera 10.5+
- Браузеры, основанные на WebKit (например, Google Chrome, Apple Safari)^[1]

JSON.stringify JSON.parse

⇒ Метод `JSON.stringify(value, replacer, space)` преобразует («сериализует») значение в JSON-строку.

Он поддерживается во всех браузерах, включая IE8+. Для более старых IE рекомендуется библиотека [JSON-js](#), которая добавляет аналогичную функциональность.

⇒ Метод `JSON.parse(str, reviver)` читает JavaScript-значение из строки.

Пример использования:


```
▶ Запустить
01 var event = {
02   title: "Конференция",
03   date: "сегодня"
04 };
05
06 var str = JSON.stringify(event);
07 alert( str ); // {"title":"Конференция","date":"сегодня"}
08
09 // Обратное преобразование.
10 event = JSON.parse(str);
```

JSON.stringify

В методе `JSON.stringify(value, replacer, space)` есть ещё третий параметр `space`.

Если он является числом — то уровни вложенности в JSON оформляются указанным количеством пробелов, если строкой — вставляется эта строка.

Например:

```
▶ Запустить 
01 var user = {
02   name: "Бася",
03   age: 25,
04   roles: {isAdmin: false, isEditor: true}
05 };
06
07 var str = JSON.stringify(user, "", 4);
08
09 alert(str);
10 /* Результат -- красиво сериализованный объект:
11 {
12   "name": "Бася",
13   "age": 25,
14   "roles": {
15     "isAdmin": false,
16     "isEditor": true
17   }
18 }
19 */
```