

Язык JavaScript (Формат JSON)

JS

ФОРМАТ JSON

JSON - JavaScript Object Notation — текстовый формат обмена данными, основанный на JavaScript.

Это один из наиболее удобных форматов данных при взаимодействии с JavaScript. Если нужно с сервера взять объект с данными и передать его клиенту, то в качестве промежуточного формата – для передачи по сети, почти всегда используют именно его.

ФОРМАТ JSON

Данные в формате JSON (RFC 4627) представляют собой:

- JavaScript-объекты `{ ... }` или
- Массивы `[...]` или
- Значения одного из типов: строки в двойных кавычках, число, логическое значение `true/false`, `null`.

Почти все языки программирования имеют библиотеки для преобразования объектов в формат JSON.

Основные методы для работы с JSON в JavaScript – это:

`JSON.parse` – читает объекты из строки в формате JSON.

`JSON.stringify` – превращает объекты в строку в формате JSON, используется, когда нужно из JavaScript передать данные по сети.

ФОРМАТ JSON

Вызов `JSON.parse(str)` превратит строку с данными в формате JSON в JavaScript-объект/массив/значение.

```
var numbers = "[0, 1, 2, 3]";  
numbers = JSON.parse(numbers);  
alert( numbers[1] ); // 1
```

```
var user = '{ "name": "Вася", "age": 35, "isAdmin": false, "friends":  
[0,1,2,3] }';  
user = JSON.parse(user);  
alert( user.friends[1] ); // 1
```

Данные могут быть сколь угодно сложными, объекты и массивы могут включать в себя другие объекты и массивы. Главное, чтобы они соответствовали формату.

ФОРМАТ JSON

Умный разбор: `JSON.parse(str, reviver)`

// дата в строке - в формате UTC

```
var str = '{  
    "title": "Конференция",  
    "date": "2014-11-30T12:00:00.000Z"  
}';
```

```
var event = JSON.parse(str, function(key, value) {  
    if (key == 'date') return new Date(value);  
    return value;  
});
```

```
alert( event.date.getDate() );
```

ФОРМАТ JSON

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

```
var event = {  
    title: "Конференция",  
    date: "сегодня"  
};  
  
var str = JSON.stringify(event);  
alert( str ); // {"title":"Конференция","date":"сегодня"}  
  
// Обратное преобразование.  
event = JSON.parse(str);
```

!!!При сериализации объекта вызывается его метод `toJSON`. Если такого метода нет – перечисляются его свойства, кроме функций.

ФОРМАТ JSON

Во втором параметре `JSON.stringify(value, replacer)` можно указать массив свойств, которые подлежат сериализации.

```
var user = {  
    name: "Вася",  
    age: 25,  
    window: window  
};  
  
alert( JSON.stringify(user, ["name", "age"]) );  
// {"name":"Вася","age":25}
```

ФОРМАТ JSON

Для более сложных ситуаций вторым параметром можно передать функцию `function(key, value)`, которая возвращает сериализованное `value` либо `undefined`, если его не нужно включать в результат:

```
var user = {  
    name: "Вася",  
    age: 25,  
    window: window  
};  
  
var str = JSON.stringify(user, function(key, value) {  
    if (key == 'window') return undefined;  
    return value;  
});  
  
alert( str ); // {"name":"Вася","age":25 }
```


ФОРМАТ JSON

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.

```
var user = {  
    name: "Вася",  
    age: 25  
};  
  
var str = JSON.stringify(user, "", 4);  
alert( str );  
/* Результат -- красиво сериализованный объект:  
{  
    "name": "Вася",  
    "age": 25  
} */
```

ПОЛЕЗНЫЕ ССЫЛКИ

<https://learn.javascript.ru/json> - Формат JSON, метод toJSON