

Лабораторная работа № 23

ТЕМА: «ФОРМАТ JSON И СПОСОБЫ ОБРАБОТКИ ДАННЫХ В КЛИЕНТСКИХ ПРИЛОЖЕНИЯХ»

Цель работы: изучить структуру формата JSON и работу с этим форматом в двух направлениях: конвертирование объекта JavaScript в строку JSON и обратно, рассмотреть примеры работы с методами JSON.stringify() и JSON.parse() .

Повторить учебный материал по темк «Основы FRONT-END разработки»

ФОРМАТ JSON И ЕГО СТРУКТУРА [1, 349-350]
КОНВЕРТАЦИЯ ОБЪЕКТА JAVASCRIPT В СТРОКУ JSON [1, 350-351]
ПАРСИНГ СТРОКИ JSON В ОБЪЕКТ JAVASCRIPT [1, 351]

1. Изучить теоретический материал по перечисленным выше темам по учебному пособию [1, 349-351]. Записать в тетрадь определение формата JSON и сравнение его с форматом XML. Описать процессы сериализации и десериализации.
2. Отладить пример по конвертации объекта JavaScript в строку JSON.
3. Отладить пример парсинга строки JSON в объект JavaScript.

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

JavaScript предоставляет методы:

- JSON.stringify для преобразования объектов в JSON.
 - JSON.parse для преобразования JSON обратно в объект.
4. Отладить примеры и рассмотреть особенности использование метода JSON.stringify

Например, здесь мы преобразуем через JSON.stringify данные студента:

```
let student = {
  name: 'John',
  age: 30,
  isAdmin: false,
  courses: ['html', 'css', 'js'],
  wife: null
};

let json = JSON.stringify(student);

alert(typeof json); // мы получили строку!

alert(json);
/* выведет объект в формате JSON:
{
  "name": "John",
  "age": 30,
  "isAdmin": false,
  "courses": ["html", "css", "js"],
  "wife": null
}
*/
```

Метод JSON.stringify(student) берёт объект и преобразует его в строку.

Полученная строка json называется *JSONформатированным* или *сериализованным* объектом. Строку можно отправить по сети или поместить в обычное хранилище данных.

Обратите внимание, что объект в формате JSON имеет несколько важных отличий от объектного литерала:

- Строки используют двойные кавычки. Никаких одинарных кавычек или обратных кавычек в JSON. Так 'John' становится "John".

- Имена свойств объекта также заключаются в двойные кавычки. Это обязательно. Так age:30 становится "age":30.

JSON.stringify может быть применён и к примитивам.

JSON поддерживает следующие типы данных:

- Объекты { ... }
- Массивы [...]
- Примитивы:
 - строки,
 - числа,
 - логические значения true/false,
 - null.

Например:

```
// число в JSON остаётся числом
alert( JSON.stringify(1) ) // 1
```

```
// строка в JSON по-прежнему остаётся строкой, но в двойных кавычках
alert( JSON.stringify('test') ) // "test"
```

```
alert( JSON.stringify(true) ); // true
```

```
alert( JSON.stringify([1, 2, 3]) ); // [1,2,3]
```

5. Отладить примеры и рассмотреть особенности использование метода JSON.parse

Чтобы декодировать JSON-строку, нам нужен другой метод с именем [JSON.parse](#).

Синтаксис:

```
let value = JSON.parse(str[, reviver]);
```

str

JSON для преобразования в объект.

reviver

Необязательная функция, которая будет вызываться для каждой пары (ключ, значение) и может преобразовывать значение.

Например:

```
// строковый массив
let numbers = "[0, 1, 2, 3]";

numbers = JSON.parse(numbers);
```

```
alert( numbers[1] ); // 1
```

Или для вложенных объектов:

```
let user = '{ "name": "John", "age": 35, "isAdmin": false, "friends": [0,1,2,3] }';
```

```
user = JSON.parse(user);
```

```
alert( user.friends[1] ); // 1
```

JSON может быть настолько сложным, насколько это необходимо, объекты и массивы могут включать другие объекты и массивы. Но они должны быть в том же JSON-формате.

Вот типичные ошибки в написанном от руки JSON (иногда приходится писать его для отладки):

```
let json = `{
  name: "John",                // Ошибка: имя свойства без кавычек
  "surname": 'Smith',          // Ошибка: одинарные кавычки в значении
(должны быть двойными)
  'isAdmin': false,            // Ошибка: одинарные кавычки в ключе
(должны быть двойными)
  "birthday": new Date(2000, 2, 3), // Ошибка: не допускается конструктор
"new", только значения
  "gender": "male"             // Ошибка: отсутствует запятая после
непоследнего свойства
  "friends": [0,1,2,3],        // Ошибка: не должно быть запятой после
последнего свойства
}`;
```

Кроме того, JSON не поддерживает комментарии. Добавление комментария в JSON делает его недействительным.

6. Задача. Преобразуйте объект в JSON, а затем обратно в обычный объект
Преобразуйте user в JSON, затем прочитайте этот JSON в другую переменную.

```
let user = {
  name: "Василий Иванович",
  age: 35
};
```

Решение записать в тетрадь

7. Оформить отчет по лабораторной работе с рассуждениями о выполнении заданий.
Отлаженные скрипты должны быть подключены к сайту.
8. Защитить лабораторную работу.

ДОМАШНЕЕ ЗАДАНИЕ

9. Ответить на контрольные вопросы [1, 352]
10. Индивидуальное задание. Оформить сайт по теме «Основы FRONT-END разработки».
Содержание сайта должно включать **скрипты лабораторных работ по JavaScript.**

ЛИТЕРАТУРА

1. Брылёва, А. А. Программные средства создания интернет-приложений : учеб. Пособие / А. А. Брылёва. – Минск : РИПО, 2022. – 483 с, : ил.

*Лабораторная работа составлена
преподавателем спецдисциплин
Еленой Иосифовной Шутько
февраль 2025 год*