

Практическое занятие № 9

Тема: Формат JSON. Понятие и основные приемы работы.

Цель: Выработать первичные навыки работы с форматом данных JSON в языке JavaScript.

Теоретическая часть

JSON (JavaScript Object Notation) — текстовый формат обмена данными, основанный на JavaScript. Как и многие другие текстовые форматы, JSON легко читается людьми.

Несмотря на происхождение от JavaScript (точнее, от подмножества языка стандарта ECMA-262 1999 года), формат считается независимым от языка и может использоваться практически с любым языком программирования. Для многих языков существует готовый код для создания и обработки данных в формате JSON.

JSON обладает двумя основными функциональными возможностями:

- является форматом данных, передаваемых между клиентом и сервером;
- используется для определения конфигураций.

Синтаксис JSON прост и обладает ограниченной поддержкой типов данных, к которым относятся object ({}), array ([]), number, string, boolean и null. Однако функции, NaN, Infinity, undefined и Symbol не являются допустимыми значениями JSON. JSON не имеет поддержки пространств имен, комментариев или атрибутов. Он не может поддерживать сложные конфигурации. Эти ограничения делают JSON простым и доступным, и поэтому он быстро усваивается и легко интерпретируется.

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 предлагает два статических метода — JSON.parse() и JSON.stringify().

JSON.parse()

JSON.parse(text) парсит строку JSON для создания значения или объекта JavaScript. Для объектов имена свойств JSON должны быть строками с двойными кавычками, а запятые в конце строки запрещены. Для примитивных типов JSON.parse() возвращает примитивные значения. Для чисел запрещены начальные нули, а за десятичной точкой должна следовать хотя бы одна цифра. Любые нарушения синтаксиса JSON приводят к ошибке SyntaxError.

JSON.stringify()

JSON.stringify(value) возвращает JSON-строку, соответствующую указанному value. boolean, number и string преобразуются в соответствующие примитивные значения. Функции, undefined и Symbol являются недопустимыми значениями JSON, которые опускаются в объекте или заменяются на null в массиве. NaN, Infinity и null заменяются на null. Полученная строка json называется JSON-форматированным или сериализованным объектом. Его можно отправить по сети или поместить в обычное хранилище данных.

```
<script>
  let json1 = '[12, 45, 20, 135, 7]'; // массив
  console.log(json1);

  let json2 = '{"firstName":"Иванов","lastName":"Иван","age": 20}'; // простой
  объект

  let data2 = JSON.parse(json2); //метод parse объекта JSON распарсил строку
  json2 в объект data2 с набором атрибутов
  console.log(data2);
  console.log(data2.age);
  console.log(data2.firstName);
  console.log(data2.lastName);

  let obj = {
    firstName:"Петров",
    lastName:"Петр",
    age: 21
  }
  console.log(obj);

  let json3 = JSON.stringify(obj); //метод stringify объекта JSON вернул объект
  obj в строку json3
  console.log(json3);

  // объект со сложной структурой
  let json4 = `{
    "book": {
      "number1": {
        "author": "Достоевский",
        "title": "Идиот"
      },
      "number2": {
        "author": "Чехов",
        "title": "Палата № 6"
      }
    }
  }`;
  let data4 = JSON.parse(json4);
  console.log(data4);
  console.log(data4.book.number1.title);

</script>
</body>
</html>
```

Ход работы

Имеется строка

```
{
  "firstName": "Иванов",
  "lastName": "Иван",
  "age": 20,
  "address": {
    "streetAddress": "пл. Гагарина, 1",
    "city": "Ростов-на-Дону",
    "postalCode": 344000
  },
  "phoneNumbers": [
    {
      "type1": "home",
      "number1": "634-5625-45-63"
    },
    {
      "type2": "fax",
      "number2": "634-5625-45-64"
    }
  ]
}
```

Выполнить следующие действия:

1. Распарсить строку в объект.
2. Вывести полученный объект в консоль.
3. Вывести в консоль фамилию, возраст, город.
4. Вывести в консоль все номера.
5. Вывести в консоль номер 634-5625-45-64.
6. Из имеющейся строки создать объект. Затем полученный объект преобразовать в строку JSON
7. Сделать выводы по результатам работы.