

# ОПЕРИРОВАНИЕ ДАННЫМИ, ОБРАБОТКА ИНФОРМАЦИИ И МАТЕМАТИЧЕСКИЕ РАСЧЕТЫ ПОСРЕДСТВОМ ЯЗЫКА JAVASCRIPT

**Пармон В.Д., Баркович А.А.**

Минский государственный лингвистический университет

**Аннотация.** Статья посвящена рассмотрению специфики оперирования данными, обработки информации и математических расчетов с помощью программных средств языка JavaScript. На примерах описаны возможности, функциональность и особенности задействования инструментария JavaScript в веб-разработке.

**Ключевые слова:** JavaScript, программирование, веб-разработка, функциональность, структура данных.

В JavaScript присутствует возможность обработки таких примитивных типов данных, как:

- boolean (булев тип);
- string (строка);
- number (число);
- BigInt (числа в формате длинной арифметики);
- symbol (символ).

Также присутствуют типы null/undefined (отсутствие данных) и object (простая структура данных в виде ключ-пара сущности, используемая не только для хранения данных, но и для создания других, более сложных структур – массивов, связанных списков, стеков, бинарных деревьев поиска и др.) [3].

Объект JavaScript под названием Math предоставляет ряд полезных свойств и методов для выполнения математических задач, таких как генерация случайных чисел, округление чисел, получение таких значений, как число пи и др. Он также включает методы

выполнения арифметических вычислений с помощью стандартных математических операторов (+, -, \* и /), тригонометрических операций (например, вычисление значений синуса или косинуса) посредством встроенных методов (sin(), cos() и др.) [1].

Наличие возможности обработки вышеупомянутых типов данных с помощью встроенных инструментов и функций (поиска, сортировки и др.), в том числе наличие возможности проведения практически любых математических операций делают этот язык мощным инструментом оперирования данными, обработки информации и проведения математических расчетов.

**API** (Application Programming Interface — «программный интерфейс приложения») — описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой [2].

**CRUD** — акроним, обозначающий четыре базовые функции, используемые при работе с базами данных: создание (create), чтение (read), модификация (update), удаление (delete). Это стандартная классификация функций по манипуляции данными [4].

Для работы с запросами и ответами HTTP в JavaScript существует специальный API под названием Fetch. В данном интерфейсе содержатся все необходимые методы для исполнения сетевых CRUD-операций:

- get (чтение)
- post (создание)
- patch\put (обновление)
- delete (удаление) [5].

```
const url = 'https://example.com/profile';
const data = { username: 'example' };

try {
  const response = await fetch(url, {
    method: 'POST', // или 'PUT'
    body: JSON.stringify(data), // данные могут быть 'строкой' или
    {объектом}!
    headers: {
      'Content-Type': 'application/json'
    }
  });
  const json = await response.json();
  console.log('Успех:', JSON.stringify(json));
} catch (error) {
  console.error('Ошибка:', error);
}
```

Рис. 1 – типичный пример обработки данных с помощью Fetch API

Любые данные отправляются и передаются в определенном формате. В веб-приложениях как для обмена данными между браузером и сервером, так и между серверами (программные HTTP-сопряжения) применяется формат JSON.

**JSON** (JavaScript Object Notation — текстовый формат обмена данными, основанный на JavaScript, но независимый от него).

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

- Набор пар «ключ: значение» (ассоциативный массив);
- Упорядоченный набор значений (массив, вектор, список).

Выбор конкретной структуры JSON зависит от языка программирования. В языке JavaScript для этого используется пара «ключ: значение» (абстрактный тип данных под названием ассоциативный массив), в которой ключом может быть только строка (упорядоченное множество из нуля или более символов юникода, заключённое в двойные кавычки).

Структуры данных, используемые JSON, поддерживаются любым современным языком программирования, что и позволяет применять JSON для обмена данными между различными языками программирования и программными системами [6].

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

Рис. 2 – пример информации, записанной в формате JSON

Таким образом, веб-разработка представляет собой сложноорганизованную совокупность тесно связанных между собой языков программирования, взаимодействие которых обеспечивается языком JavaScript. Среди типичных задач, выполняемых посредством

данного языка – оперирование данными, обработка информации и математические расчеты.

## ЛИТЕРАТУРА

1. Математические операторы JavaScript [Электронный ресурс]: Электронно-информационный ресурс Tutorial republic – Режим доступа: <https://www.tutorialrepublic.com/javascript-tutorial/javascript-math-operations.php> (дата обращения 01.06.2022).

2. Программный интерфейс приложения [Электронный ресурс]: Википедия. Свободная энциклопедия. – Режим доступа: <https://ru.wikipedia.org/wiki/API> (дата обращения: 01.06.2022).

3. Типы и структуры данных JS [Электронный ресурс]: Сайт веб-документации MDN. – Режим доступа: [https://developer.mozilla.org/ru/docs/Web/JavaScript/Data\\_structures](https://developer.mozilla.org/ru/docs/Web/JavaScript/Data_structures) (дата обращения 01.06.2022).

4. CRUD [Электронный ресурс]: Википедия. Свободная энциклопедия. – Режим доступа: <https://ru.wikipedia.org/wiki/CRUD> (дата обращения: 01.06.2022).

5. Fetch API [Электронный ресурс]: Сайт веб-документации MDN. – Режим доступа: <https://developer.mozilla.org> (дата обращения 01.06.2022).

6. JSON [Электронный ресурс]: Википедия. Свободная энциклопедия. – Режим доступа: <https://ru.wikipedia.org/wiki/JSON> (дата обращения: 01.06.2022).