Метод map() в JavaScript

Основные принципы работы

Метод map() создает новый массив, применяя функцию-колбэк к каждому элементу исходного массива.

Ключевые особенности:

* Не изменяет исходный массив
* Возвращает новый массив той же длины
* Принимает функцию-колбэк с тремя параметрами:
  1. Текущий элемент
  2. Индекс элемента
  3. Сам массив

Примеры использования

1. Преобразование числового массива

javascript

Copy

Download

const numbers = [1, 2, 3, 4, 5];

// Возведение в квадрат

const squares = numbers.map(num => num \*\* 2);

// [1, 4, 9, 16, 25]

// Упрощенная запись без return

const doubled = numbers.map(num => num \* 2);

// [2, 4, 6, 8, 10]

2. Работа с объектами

javascript

Copy

Download

const users = [

{ id: 1, name: 'Анна', email: 'anna@example.com', age: 25 },

{ id: 2, name: 'Иван', email: 'ivan@example.com', age: 30 }

];

// Извлечение нужных полей

const userContacts = users.map(user => ({

name: user.name,

email: user.email

}));

/\*

[

{name: 'Анна', email: 'anna@example.com'},

{name: 'Иван', email: 'ivan@example.com'}

]

\*/

3. Использование индекса

javascript

Copy

Download

const items = ['a', 'b', 'c'];

const indexedItems = items.map((item, index) => ({

value: item,

position: index

}));

/\*

[

{value: 'a', position: 0},

{value: 'b', position: 1},

{value: 'c', position: 2}

]

\*/

Сравнение с forEach()

| **Характеристика** | map() | forEach() |
| --- | --- | --- |
| Возвращаемое значение | Новый массив | undefined |
| Изменение исходного | Нет | Возможно (мутация) |
| Цепочка методов | Да | Нет |
| Производительность | Медленнее | Быстрее |

Практические советы

1. **Не используйте**map**если:**
   * Нужен только перебор элементов
   * Не используете возвращаемый массив
   * Требуется максимальная производительность
2. **Лучшие практики:**
   * Избегайте мутаций исходных данных
   * Делайте колбэки чистыми функциями
   * Используйте деструктуризацию для сложных объектов

javascript

Copy

Download

const names = users.map(({ name }) => name);

Метод map() — идеальный выбор для преобразования массивов без побочных эффектов.