Метод call() в JavaScript

Основные концепции

call() — метод, позволяющий явно установить контекст (this) для вызова функции.

Синтаксис:

javascript

Copy

Download

func.call(context, arg1, arg2, ...)

Практические примеры

1. Базовое использование

javascript

Copy

Download

function showFullName() {

console.log(`${this.firstName} ${this.lastName}`);

}

const user = {

firstName: 'Иван',

lastName: 'Петров'

};

showFullName.call(user); // "Иван Петров"

2. Передача аргументов

javascript

Copy

Download

function greet(greeting, punctuation) {

console.log(`${greeting}, ${this.name}${punctuation}`);

}

const person = { name: 'Мария' };

greet.call(person, 'Привет', '!'); // "Привет, Мария!"

3. Заимствование методов (method borrowing)

javascript

Copy

Download

// Преобразование arguments в массив

function listArgs() {

const args = Array.prototype.slice.call(arguments);

console.log(args);

}

listArgs(1, 'text', true); // [1, "text", true]

Особенности работы

1. В нестрогом режиме:
   * null и undefined заменяются на глобальный объект
   * Примитивы преобразуются в объекты
2. В строгом режиме:
   * Значения передаются как есть

Сравнение с другими методами

| **Метод** | **Вызов функции** | **Аргументы** | **Возвращает** |
| --- | --- | --- | --- |
| call | Немедленно | Через запятую | Результат функции |
| apply | Немедленно | Массив | Результат функции |
| bind | Позже | Через запятую | Новую функцию |

Полезные применения

1. **Вызов родительского конструктора**:

javascript

Copy

Download

function Product(name, price) {

this.name = name;

this.price = price;

}

function Food(name, price) {

Product.call(this, name, price);

this.category = 'food';

}

1. **Работа с псевдомассивами**:

javascript

Copy

Download

function getMax() {

return Math.max.apply(null, arguments);

}

console.log(getMax(1, 3, 2)); // 3

Метод call() — мощный инструмент для управления контекстом выполнения функций в JavaScript.