Синтаксис **деструктурирующего присваивания** в выражениях JavaScript позволяет извлекать данные из массивов или объектов при помощи синтаксиса, подобного объявлению массива или литералов в объекте.

## [Синтаксис](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Operators/Destructuring_assignment#%D1%81%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81)

|  |
| --- |
| var a, b, rest;  [a, b] = [1, 2];  console.log(a); // 1  console.log(b); // 2  [a, b, ...rest] = [1, 2, 3, 4, 5];  console.log(a); // 1  console.log(b); // 2  console.log(rest); // [3, 4, 5]  ({ a, b } = { a: 1, b: 2 });  console.log(a); // 1  console.log(b); // 2  ({ a, b, ...rest } = { a: 1, b: 2, c: 3, d: 4 });  console.log(a); // 1  console.log(b); // 2  console.log(rest); // { c:3, d:4 } |

Пример:

|  |
| --- |
| var foo = ["one", "two", "three"];  // без деструктурирования  var one = foo[0];  var two = foo[1];  var three = foo[2];  // с деструктурированием  var [one, two, three] = foo; |

**Spread syntax** позволяет расширить доступные для итерации элементы (например, массивы или строки) в местах

* для функций: где ожидаемое количество аргументов для вызовов функций равно нулю или больше нуля
* для элементов (литералов массива)
* для выражений объектов: в местах, где количество пар "ключ-значение" должно быть равно нулю или больше (для объектных литералов)