

Деструктуризация. Spread. Rest

Деструктуризация позволяет быстро создать переменные на основе массива или объекта

Полный синтаксис для массива:

```
const [item1 = default, item2, ...rest] = array
```

Первый элемент отправляется в **item1**; второй отправляется в **item2**, все остальные элементы попадают в массив **rest**.

Полный синтаксис для объекта:

```
const { objectPropertyName: newVariableName = default, ...rest } = object
```

Свойства, которые не были упомянуты, копируются в объект **rest**.

Обратить внимание, что **let** или **const** - обязательно.

Для работы с массивом используем квадратные скобки [], для работы с объектом фигурные {}

При работе с объектом по умолчанию используем те же названия свойств что и в объекте

Если из объекта надо создать переменные с другими именами, то задаем их в виде { **objectPropertyName: newVariableName** }

Примеры:

1) **Массив**

```
let [firstName, surname] = ["Ilya", "Kantor"];
alert(firstName); // Ilya
alert(surname); // Kantor
```

2) **Массив**

```
let [firstName, surname] = "Ilya Kantor".split(' ');
```

3) **Объект**

```
const obj = {
  firstName: "John",
  secondName: "Smith",
};
const { firstName, secondName } = obj;
console.log(firstName, secondName); // John Smith
```

```
const { firstName: name, secondName: second } = obj;
console.log(name, second); // John Smith
```

4) **Возвращаемое значение для функции**

```
const obj = {
  function isFullAgeFunc(yearOfBirth) {
    const age = 2020 - yearOfBirth; // 22
    const isFullAge = age >= 18; // true
    return [age, isFullAge];
  }
}
const [age, isFullAge] = isFullAgeFunc(1992);
console.log(age, isFullAge); // 28 true
```

ES6. Spread оператор.

Преобразует массив с значениями в **аргументы**, в вызове функции.

//ES5

```
var numbers = [18, 30, 12, 21];
function calcSum(a, b, c, d) {
  return a + b + c + d;
}
var sumES5 = calcSum.apply(null, numbers);
console.log(sumES5); // 81
```

//ES6 с помощью оператора **spread**

```
var numbers = [18, 30, 12, 21];
function calcSum(a, b, c, d) {
  return a + b + c + d;
}
const sumES6 = calcSum(...numbers);
console.log(sumES6); // 81
```

Другой пример - объединение массивов:

```
const familySmith = ["John", "Jane", "Mark"];
const familyMiller = ["Mary", "Bob", "Ann"];
const bigFamily = [...familySmith, "Emily", ...familyMiller];
console.log(bigFamily); // ['John', 'Jane', 'Mark', 'Emily', 'Mary', 'Bob', 'Ann']
```

Пример с нодами:

```
const header = document.querySelector("h1");
const boxes = document.querySelectorAll(".box");
const allElements = [header, ...boxes];
console.log(allElements); // [h1.title, div.box.green, div.box.blue, div.box.orange]
```

ES6. Rest параметры.

Принимает одиночные аргументы функции и трансформирует их в массив

// ES5

```
function isFullAgeES5() {
  console.log(arguments);
  var argsArray = Array.prototype.slice.call(arguments);
  argsArray.forEach(function (item) {
    console.log(2020 - item >= 18);
  });
}
isFullAgeES5(1990, 1999, 1965); // true, true, true
```

// ES6 с помощью оператора **rest**

```
function isFullAgeES6(...years) {
  console.log(years);
  years.forEach((item) => console.log(2020 - item >= 18));
}
```

isFullAgeES6(1990, 1999, 1965); // true, true, true

Пример 02. Массив и отдельный первый аргумент

// ES5

```
function isFullAgeES5(limit) {
  console.log(arguments);
  var argsArray = Array.prototype.slice.call(arguments, 1);
  console.log(argsArray);
```

```
  argsArray.forEach(function (item) {
    console.log(2020 - item >= limit);
  });
}
isFullAgeES5(18, 1990, 1999, 1965); // true, true, true
```

// ES6

```
function isFullAgeES6(limit, ...years) {
  console.log(years);
  years.forEach((item) => console.log(2020 - item >= limit));
}
```

isFullAgeES6(24, 1990, 1999, 1965); // true, false, true

Параметры по умолчанию

// ES5

```
function SmithPersonES5(firstName, yearOfBirth, lastName, nationality) {  
  if (lastName === undefined) {  
    lastName = "Smith";  
  }  
  
  if (nationality === undefined) {  
    nationality = "american";  
  }  
  
  this.firstName = firstName;  
  this.yearOfBirth = yearOfBirth;  
  this.lastName = lastName;  
  this.nationality = nationality;  
}  
  
var john = new SmithPersonES5("John", 1990);  
var emily = new SmithPersonES5("Emily", 1990, "Diaz", "spanish");  
console.log(john);  
console.log(emily);
```

// ES6

```
function SmithPersonES6(firstName, yearOfBirth, lastName = "Smith", nationality = "american") {  
  this.firstName = firstName;  
  this.yearOfBirth = yearOfBirth;  
  this.lastName = lastName;  
  this.nationality = nationality;  
}  
  
var johnES6 = new SmithPersonES6("John", 1990);  
var emilyES6 = new SmithPersonES6("Emily", 1990, "Diaz", "spanish");  
console.log(johnES6);  
console.log(emilyES6);
```