

[←](#) 2-36 Desctructuring

Desctructuring

Desctructuring

在 ES6 中新增了变量赋值的方式：解构赋值。如果对这个概念不了解，我们可以快速展示一个小示例一睹风采：

```
let arr = ['Ilya', 'Kantor']
let firstName = arr[0]
let surname = arr[1]
```

想从数组中找出有意义的项要单独赋值给变量（——的写），在 ES6 中就可以这样写了：

```
let [firstName, surname] = ['Ilya', 'Kantor']
console.log(firstName) // Ilya
console.log(surname) // Kantor
```

[!TIP]

解构赋值重点是在赋值，赋值的元素是要拷贝出来赋值给变量，赋值的元素本身是不会被改变的。

是不是非常的刺激，在解构赋值里用的最多的就是 Object 和 Array，我们可以分别来看下两者的解构赋值是如何操作的。

Array Destructuring

上述的小示例已经在描述数组的解构赋值了，可是那只是开始。

- 可以跳过赋值元素

如果想忽略数组的某个元素对变量进行赋值，可以使用逗号来处理。

```
// second element is not needed
let [firstName, , title] = ["Julius", "Caesar", "Consul", "of the Roman Republic"];

console.log( title ); // Consul
```

- 赋值元素可以是任意可遍历的对象

赋值的元素不仅是数组，它可以是任意可遍历的对象

```
let [a, b, c] = "abc"; // ["a", "b", "c"]
let [one, two, three] = new Set([1, 2, 3]);
```

- 左边的变量

被赋值的变量还可以是对象的属性，不局限于单纯的变量。

```
let user = {};
[user.name, user.surname] = "Ilya Kantor".split(' ');

console.log(user.name); // Ilya
```

- 循环体

解构赋值在循环体中的应用，可以配合 entries 使用。

```
let user = {
  name: "John",
  age: 30
};
```