

[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。

用法不错

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

对象解构赋值

[TIP]

在这个结构赋值的过程中，左侧的“模板”结构要与右侧的 Object 一致，但是属性的顺序无需一致。

练习

1. 一个函数需要传入很多参数，是否可以利用解构赋值来简化操作呢？
2. 如何在业务开发中对接口数据进行解构赋值呢？

解构赋值的应用

```
// 1. 交换变量的值  
let x = 1;  
let y = 2;  
[x, y] = [y, x]  
  
// 2. 从函数返回多个值  
// 函数只能返回一个值 若要返回多个  
// 可以放在数组或对象里返回 还可以利用解构赋值  
function example() {  
    return [1, 2, 3];  
}  
  
let [a, b, c] = example();  
example() // 返回一个数组
```

```
function example() {
    return{
        foo:1,
        bar:2
    };
}
let {foo,bar} = example();
// 返回一个对象

// 3.函数参数的定义
// 参数是一组有序的值
function f([x,y,z]){...}
f([1,2,3]);

//参数是一组无序的值
function f({x,y,z}){...}
f({z:3,y:2,x:1})

// 4.提取json数据
let jsonData = {
    id:42,
    status:'ok',
    data:[867,5309]
};
let {id,status,data:number} = jsonData;
console.log(id,status,number)

// 5.函数参数的默认值
jQuery.ajax = function (url,{
    async = true,
    beforeSend = function() {},
    cache = true,
    complete = function() {},
    crossDomain = false,
    global = true,
} = {}){
}
```

// 6.遍历Map解构

```
const map = new Map();
map.set('first','hello')
map.set('last','world')
for(let [Key,value] of map){
    console.log(key + 'is' + value);
}
```



函数参数的解构赋值



```
function add([x,y]){
    return x + y;
}
add([1,2])
// add的参数表面是数组 但在传参时 数组参数被解构为x和y

[[1,2],[3,4]].map(([a,b]) => a +b)
// [3,7]
```



数值和布尔值的解构赋值



```
// 解构赋值时 等号右边是数值或布尔值时 则会先转为对象
let {toString:s} = 123;
s === Number.prototype.toString//true
let {toString:s} = true;
s === Boolean.prototype.toString;//true
//数值和布尔值的包装对象都有toString属性 因此变量s都能取到值

// 解构赋值原则 只要等号右边的值不是对象或数组
// 就先将其转换为对象 因为null和undefined无法转换为对象 所以解构赋值会报错
```