28：变量的解构赋值

定义：从数组和对象中提取值，对变量进行赋值，这就被称为结构赋值，只要等号两边的模式相同，左边的变量就会被赋予对应的值。

1：数组的结构赋值

可以直接赋值：var [a, b, c] = [1, 2, 3];

let支持解构赋值：let [foo,[[bar],baz]] = [1,[[2],3]];

let [x, , y] = [1, 2, 3];

如果解构不成功，变量的值就是underfined

当变量的值发生了不完全解构时，只解构与变量的数量相等的数量数组中的值

解构时可以带上默认值：

var [foo3 = true] = [];//foo3 为 true

[x3, y3 = 'b'] = ['a']; // x3='a', y3='b'

[x4, y4 = 'b'] = ['a',undefined]; // x4='a'y4='b

但是如果有默认值后边数组中又赋值，则值为 数组中的值

2：对象的解构赋值

var { foo1, bar1 } = { foo1: "aaa", bar1: "bbb" };

console.log(foo1,bar1);//aaa,bbb

3:数组解构赋值与对象解构的区别

数组赋值是是按照位置决定的，但是对象的属性没有次序，变量必须和属性同名，才能取到正确的值

4：字符串解构赋值

const [a, b, c, d, e] = 'hello';//相当于将'hello'转成了["h","e","l","l","o"]后解构

console.log(a); // "h"

console.log(b); // "e"

console.log(c); // "l"

console.log(d); // "l"

console.log(e); // "o"

5：数字的解构赋值

let {toString: s1} = 123;

console.log(s1);

解构赋值时，如果等号的右边是数值和布尔值，就会先转换成对象。

6：参数的解构赋值

function move1({x = 0, y = 0} = {}) {

return [x, y];

}

console.log(move1({x: 3, y: 4})); // [3, 4]

console.log(move1({x: 3})); // [3, 0]

console.log(move1({})); // [0, 0]

console.log(move1()); // [0, 0]

7：解构赋值的常见应用

交换变量

从函数中返回多个值

提取JSNO数据

给函数指定默认参数