解构赋值语法是一个 Javascript 表达式，这使得可以将值从数组或属性从对象提取到不同的变量中。

[a, b, ...rest] = [10, 20, 30, 40, 50];

({ a, b } = { a: 10, b: 20 });

({a, b, ...rest} = {a: 10, b: 20, c: 30, d: 40});

console.log(rest); //{c: 30, d: 40}

数组形式的解构，按索引顺序依次赋值，当使用剩余模式赋值时，(…variable)一定是变量数组的最后一位。

在数组形式中，也可以通过省略变量来跳过不需要的赋值，如：

var [，a, ...b] = [1, 2, 3，4];

console.log(a); // 2

console.log(b); // [3, 4]

也可以直接交换变量

var a = 1;

var b = 3;

[a, b] = [b, a];

console.log(a); // 3

console.log(b); // 1

对于对象的解构可以声明后赋值

var o = {p: 42, q: true};

var {p, q} = o;

console.log(p); // 42

console.log(q); // true

也可以不声明直接赋值

var a, b;

({a, b} = {a: 1, b: 2});

可以从一个对象中提取变量并赋值给和对象属性名不同的新的变量名。

var o = {p: 42, q: true};

var {p: foo, q: bar} = o;

console.log(foo); // 42

console.log(bar); // true

一个属性可以是

1）从一个对象解构，并分配给一个不同名称的变量，

2）分配一个默认值，以防未解构的值是undefined。

var {a:aa = 10, b:bb = 5} = {a: 3};

console.log(aa); // 3

console.log(bb); // 5

需要注意的是，在对象解构中｛a:b｝中，a是作为键值(key)存在b是作为属性变量名存在，实际赋值赋予的是b = variable，***当给解构对象赋予默认值时，格式是{a:b=1}，而不是{a:1}这种错误表示形式***。若不用键值对的形式表示，那么只需要展示出键值就行，如{a,b} ={a:1,b:2}，赋值部分一定是一个完整的object结构。