**变量的解构赋值**

本质上就是一种匹配模式，只要等号两边的模式相同，等号左边的变量就可以被赋予对应的值

分：解构和赋值

如：

let [a, b, c] = [1, 2, 3];

console.log(a, b, c);

解构赋值分为：

1、数组的解构赋值

2、对象的解构赋值

用babel解析看看

<https://www.babeljs.cn/repl/>

var arr = [1,2,3];

var [a, b, c] = arr;

数组的解构赋值：数组解构赋值（由位置决定）

1. 按顺序解构 [a,b,c] = [1, 2, 3];
2. 不完全解构 [a] = [1, 2, 3];
3. 解构不成功 [a,b,c] = [1, , 3];
4. 默认值 [a,b=5,c] = [1, , 3];
5. 剩余值 [a,...r] = [1, 2, 3]

对象的解构赋值（属性没有次序，同名的属性才能取得值）

如：

let {a, b} = {b: 'bbb',a: 'aaa'};

console.log(a, b); //

先声明，再解构

let obj = {name: 'z3',age: 20};

let name, age; // 声明

({name,age} = obj); // 解构，此时要加小括号

console.log(name, age);

别名

如：n为a 的别名，从此，a 不能再访问了，而n可以访问

a为对象中的属性，但是这个a有可能同别的重名，所以配一个别名。

let {a: n} = {

b: 'bbb',

a: 'aaa'

};

console.log(n); // aaa

别名和默认值配合使用

let {c: n = 123} = {

b: 'bbb',

a: 'aaa'

};

console.log(n); // 123

let o = { a: 1, b: 2, c: 3 };

let { a, ...pzh } = o;

console.log(a, pzh); //

**用法：**

// 解构赋值的用法：从对象中取得函数，这样就省略了每次都写Math

let {random, abs} = Math;

console.log(random());

console.log(abs(-3));

// 解构赋值的第二种用法，从模块中取得返回对象

let o = (function() {

function add() {

console.log('add')

}

function getDom() {

console.log('getDom');

}

return {add, getDom}

})();

let {add, getDom} = o;

add();

getDom();