**ES6解构：es6允许按照一定模式，从数组和对象中提取值，对变量进行赋值，这被称之为解构**

**1.数组的解构赋值**

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

console.log(a);//1

console.log(b);//2

console.log(c);//3

//本质上这种写法属于‘模式匹配‘，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

let [foo,[bar]] = ["111",["abc"]];

console.log(bar); //abc

let [, ,third] = ["foo","bar","baz"];

console.log(third); //baz

let [x,y,z] = ['hah'];

console.log(y); //undefined

//如果解构不成功，变量的值就等于undefined

下面来看一下不完全解构的例子

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

console.log(x); //1

console.log(y); //2

**2. 默认值**

解构赋值允许指定默认值

let [x,y='b'] = ['a'];

console.log(y); //b

**3. 对象的解构赋值**

变量必须与属性同名，才能取到正确的值

//变量名与属性名一致的情况下

let {foo,bar} = {foo : "aaa",bar : "bbb"}

console.log(foo); //aaa

console.log(bar); //bbb

//变量名与属性名不一致的情况下，必须这样写

let {a : name, b : age} = {a : 'zhangsan', b : 33};

console.log(name); //zhangsan

console.log(age); //33

再看下面这个例子

const node = {

grand : {

father : {

line : 1,

column : 5

}

}

}

let {grand,grand : { father},grand : {father : {column}}} = node;

console.log(father); // {line : 1, column : 5}

console.log(column); // 5

**4. 字符串的解构赋值**

const [a,b,c,d,e] = 'hello';

console.log(a); //h

console.log(b); //e

console.log(c); //l

console.log(d); //l

console.log(e); //o

let { length : len} = 'yahooa';

console.log(len); //5,类似数组的对象都有一个length属性，还可以对这个属性解构赋值

**5. 数值和布尔值的解构赋值**

解构赋值时，如果等号右边是数值和布尔值，则会先转为对象，但是等号右边为undefined 和 null时无法转为对象，所以对他们进行解构赋值时，都会报错

let {prop : x } = undefined;

console.log(x); //报错：Uncaught TypeError: Cannot destructure property `prop` of 'undefined' or 'null'

**6.函数参数的解构赋值**

函数的参数也可以使用解构参数

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

return [x,y];

}

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

function move2({x,y} = {x : 1, y : 2 }){

return [x,y];

}

console.log(move2({x : 6,y : 8})); //[6,8]

**7. 解构用途**

交换变量的值

let x = 1;

let y = 2;

[x,y] = [y,x];

console.log(x); //2

console.log(y); //1

从函数返回多个值

函数只能返回一个值，如果要返回多个值的话，只能将它们放在数组或者对象里返回

function example(){

return {

foo : 'a',

bar : 'b'

}

}

let {foo,bar} = example();

console.log(foo); //a

console.log(bar); //b

函数参数的定义

//参数是一组有次序的值

function example([x,y,z]){

return x + y + z;

}

example([1,2,3])

console.log(example([1,2,3])); //6

//参数是一组无次序的值

function f({x,y,z}){

return x + y + z;

}

f({x : 'a', z : 'b', y : 'c' });

console.log(f({x : 'a', z : 'b', y : 'c' })); //acb