W1D1

下载并注册Xmind, 学习使用Xmind.

ES6 入门

**1：Let and Const**

* **Let命令**

let声明的变量只在它所在的代码块有效。

let 只能声明一次 var 可以声明多次，不能重复声明

Let适合用于计数器：

for (let j = 0; j < 10; j++) {

setTimeout(function() {

console.log(j);

})

}

这个for循环输出123456789 而不是10个10， 每次循环的j都是一个全新的变量，所以定时器里的j是不同的变量。

let 不存在变量提升，var 会变量提升。

块级作用域：

function f1() {

let n = 5;

if (true) {

let n = 10;

}

console.log(n); // 5}

外层代码块不受内层代码块的影响。

* **Const 命令**

const 声明一个只读变量，声明之后不允许改变。意味着，一旦声明必须初始化，否则会报错。

**2: 变量的解构赋值**

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

可以从数组中提取值，按照对应位置，对变量赋值

let [foo, [[bar], baz]] = [1, [[2], 3]];

foo // 1

bar // 2

baz // 3

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

third // "baz"

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

x // 1

y // 3

let [head, ...tail] = [1, 2, 3, 4];

head // 1

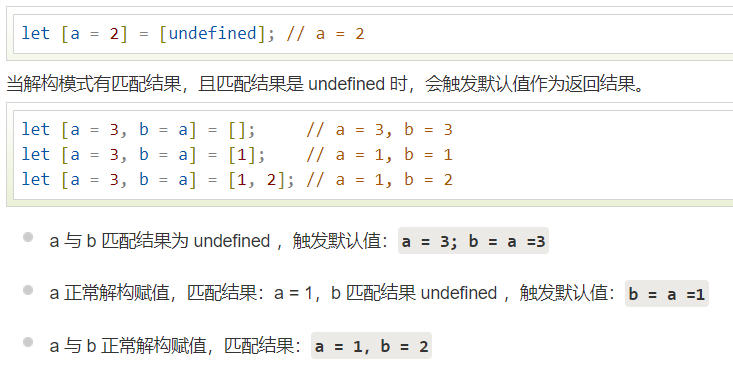
tail // [2, 3, 4]

let [x, y, ...z] = ['a'];

x // "a"

y // undefined

z // []



如果等号右边的不可遍历，那么将会报错。

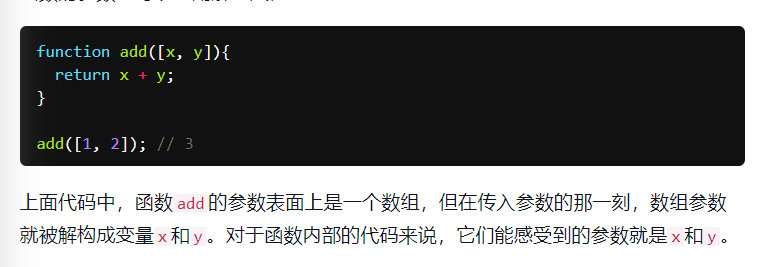
let [x = 1, y = x] = []; // x=1; y=1

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

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

let [x = y, y = 1] = []; // ReferenceError: y is not defined

x用y做默认值时，y还没有声明



* 用途：
* 交换变量的值

[x,y] = [y,x]

* + 从函数中返回多个值

将多个值放在数组或者对象里返回

// 返回一个数组function example() {

return [1, 2, 3];}

let [a, b, c] = example();

// 返回一个对象function example() {

return {

foo: 1,

bar: 2

};}

let { foo, bar } = example();

* ****函数参数的定义****

解构赋值可以方便地将一组参数与变量名对应起来。

* **提取 JSON 数据**
* **函数参数的默认值**
* **遍历 Map 结构**
* **输入模块的指定方法**

****3: 字符串****

"\u{20BB7}"

// "

"\u{41}\u{42}\u{43}"

// "ABC"

let hello = 123;

hell\u{6F} // 123'

\u{1F680}' === '\uD83D\uDE80'

// true

let text = String.fromCodePoint(0x20BB7);

for (let i = 0; i < text.length; i++) {

console.log(text[i]);}

// " "

// " "

for (let i of text) {

console.log(i);}

// "