1. 数组的解构赋值 先解除结构，然后赋值。
2. 模式对应

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

**console.log(a);//1**

**console.log(b);//2**

**console.log(c);//3**

2、例子：

**let [a,[b,c],d] = [1,[2,3],4];**

**console.log(a);//1**

**console.log(b);//2**

**console.log(c);//3**

**console.log(d);//4**

1. 不完全对称

**let [a,[b,c],d] = [1,[2],4];**

**console.log(a);//1**

**console.log(b);//2**

**console.log(c);//undefined**

**console.log(d);//4**

1. 默认值

**let [a = 1] =[];**

在这里，a的默认值为1，

如果右边的值为undefined与之对应，那么就采用默认值

5、特殊：

当解构赋值的默认值是一个表达式，而且没有必要用到默认值时，表达式不会运行

**let x = function(){**

**console.log(2);**

**return 1;**

**}**

**let [a = x() ] = [3];**

**console.log(a);//3**

6、默认值 相互引用

**let [a,b=a] = [1];**

**console.log( a , b );//1 1**

这个的引用过程是：先let a = 1， 然后再let b = a = 1；

所以a和b的值都是1.

**let [b=a,a] = [,1];**

**console.log( a , b );//error**

这个的引用过程是：先let b = a，因为a没有定义，在es6里面就会报错。

所以在es6中，不允许前面的变量引用后面的变量。

而且也看出解构赋值是有顺序的，一个一个赋值。

1. 若当前的数据拥有Iterator接口，那么可以进行解构，

这个只针对等号右边的数据。

属性里面有Symbol(Symbol.iterator)

以下几个有该接口

Array,String,NodeList,arguments

1. 对象的解构赋值，
2. 模式匹配：

**let {a:x,b:y} = {a:1,b:3};**

两个对象相对应，属性相对应，然后值就赋过去 x = 1， y = 3

这里的属性只有匹配的作用。

2、特点：

当属性名和变量名一样时，可以简写：

**let {a:a,b:b} = {a:1,b:3};**

可以简写为：

**let {a,b} = {a:1,b:3};**

两者拥有同样的效果。

1. 几个比较复杂的例子：

**let {a,b:{x,y:z}} = {a:1,b:{x:1,y:2}};**

**console.log( a );//1**

**console.log( x );//1**

**console.log( z );//2**

**let {g:k,j,l:[a,b]} = {g:1,j:2,l:'tt'};**

**console.log( k,j,a,b);//1 2 "t" "t"**

三、数组的拓展运算符

1、作用，将数组拆解成以逗号隔开的参数列表。

2、例子：

**let arr = [1,2,3];**

**let fn = function(a,b,c){**

**console.log(a,b,c);**

**}**

**fn(arr);//[1, 2, 3] undefined undefined**

**fn(...arr);//1 2 3**

从上可以看出，...arr能够将数组的内容分解成一项一项的。

1. 作用：

数组拼接：

**let arr = [1,2,3];**

**let arr2 = [4,5,6];**

**let arr3 = [...arr,...arr2];**

**console.log( arr3 );//[1, 2, 3, 4, 5, 6]**

深拷贝数组：

**let arr = [1,2,3];**

**let x = [...arr];**

**console.log( x === arr );//false**

1. 原理：有iterator接口的都能够使用...拆解

例如：Array，String，Nodelist

nodeList要是用forEach来遍历的话，得：

**[].slice.call(node).forEach() | [].forEach.call( node ,)**

如果在这儿只需要：

**[...arr].forEach();**

求数组的最大值：

Ea5： **Math.max.apply( Math,arr)**

Es6： **Math.max(...arr);**

1. 剩余变量形式：

**let [a,...b] = [1,2,3,4];**

**console.log(b);//[2, 3, 4]**

如果没有内容：

**let [a,...b] = [1];**

**console.log(b);//[]**

6、对象里面也可以进行...拓展