**遍历数组的方法:**

**for-in**

var arr=["张三","李四","王五","赵六"];

for (var i in arr){

console.log(i,":",arr[i]);

}

**需要分析出array 的每个属性，这个操作性能开销很大**。用在 key 已知的数组上是非常不划算的。所以尽量不要用 for-in，除非你不清楚要处理哪些属性，例如 JSON 对象这样的情况。使用 break，continue 跳出循环如果使用for in语句遍历数组，可能会出现以下问题：

　　1. 索引为字符串类型，不能直接进行几何运算。

　　2. 遍历顺序有可能不是按照实际数组的内部顺序。

　　3. 使用for in会遍历数组所有的**可枚举属性，包括原型。**

**for of**

语句和for in 语句相比使用更方便，它是ES6新提出的遍历数组的方式，for in遍历的是数组的索引（即键名），而for of**遍历的是数组元素值**。for of遍历的只是数组内的元素，而不包括数组的原型属性method和索引name

var arr=["a","b","c","d"]; for(var a in arr){ console.log(a); } //打印值 0 1 2 3

for(var a of arr){ console.log(a); } // 打印值 a b c d

for循环每进行一次，就要检查一下数组长度。读取属性（数组长度）要比读局部变量慢，尤其是当 array 里存放的都是 DOM 元素，因为每次读取都会扫描一遍页面上的选择器相关元素，速度会大大降低。var arr = [1,2,3,4,5];使用 break，continue 跳出循环

var length =arr.length;

for(var i=0; i<length; i++){

　　fn(arr[i]);

}

2.避免了for-in，forEach的所有缺陷

     3.可以用在 类数组，字符串，set和map数据 上

     4.不支持对象的遍历（但是它提供了其他的三种方法）

            ①Object.keys()

            ②Object.values()

            ③Object.entries()

1）Object.keys()

var obj = { name:"hello", age:"18" }; for(var i of Object.keys(obj)){ console.log(i); }

**foreach** forEach() 方法用于调用数组的每个元素，并将元素传递给回调函数。使用比较复杂。

var arr=[1,2,3,4]; arr.forEach(function(val, index) { console.log(val, index); });

使用 return 是跳过本次循环；

arr.map

**主要使用for in遍历对象**，Object.keys(obj)遍历对象 返回一个数组,包括对象自身的(不含继承的)所有可枚举属性(不含Symbol属性).

var obj = {'0':'a','1':'b','2':'c'}; Object.keys(obj).forEach(function(key){ console.log(key,obj[key]); });

var obj = {'0':'a','1':'b','2':'c'}; for(var i in obj) { console.log(i,":",obj[i]); }

**遍历字符串**

let str = 'love'

for (let o of str) { console.log(o) // l,o,v,e }