正课:

ES6

Set 集合:

什么是: 值不允许重复的集合

何时: 快速去重复

通过元素值，快速判断是否包含元素时

如何: var set=new Set(凡是可遍历的，都可打散转为Set)

将set转回数组: var arr=[...set];

Map 集合:

什么是: 一组键值对儿的集合

何时: 代替关联数组使用

如何: var hash=new Map(); //{}

hash.set(key,value); //hash[key]=value

var bool=hash.has(key); //hash[key]!==undefined

for of : 代替for循环遍历

什么是: 依次获得数组/类数组对象/集合中每个值

何时: 只要不关心位置，只关心每个元素值时

如何: for(var value of 数组|类数组对象|集合){

value: 自动获得当前元素值

}

问题: 1. 无法获得位置

2. 必须从头到尾，逐个执行，不能跳跃或遍历部分

3. 只能从头到尾，顺序遍历，不能反向遍历

\*\*\*\*Promises

什么是: 解决"callback hell"

为什么: 回调函数，层级很可能很深，可读性极差

何时: 今后，只要多个回调函数，必须保证顺序执行时

如何: 2步:

1. 在前一个函数内

return new Promise((callback)=>{

... callback() ...

})

2. 在调用前一个函数时,

前一个函数.then(()=>{ 后一个函数(); })

错误处理:

1. 在前一个函数内

return new Promise(resolve,reject)=>{

如果没出错，执行resolve

如果出错，执行reject("错误提示")

})

2. 在调用前一个函数时,

前一个函数.then(()=>{ 下一个函数(); })

.catch((err)=>{ 输出err })