正课:

1. ES5

2. ES6

1. ES5

数组API:

遍历API:

.forEach(): 对原数组中每个元素执行相同的操作

何时: 直接修改原数组时

如何:

arr.forEach(function(val,i,arr){

arr[i]=新值

})

.map(): 取出原数组中每个元素，执行相同操作后，放入新数组返回。

何时: 保护原数组不被修改，而返回新数组时

如何:

var newArr=arr.map(function(val,i,arr){

return 新值

})

过滤和汇总:

过滤: 选择出数组中符合条件的元素组成新数组

强调: 原数组不变

如何: var sub=arr.filter(function(val,i,arr){

return 判断条件

})

汇总: 将数组中所有元素统计得出一个最终结果

如何: var result=arr.reduce(function(prev,val,i,arr){

//prev: 截止当前元素之前的临时汇总值

return prev+val;

}, base )

2. ES6:

不改变原理的基础上，极简化了代码

学ES6，除了学简化外，必须会替代方法

包括:

let: 代替var声明变量

var的问题: 2个:

1. 声明提前, 2. 没有块级作用域

let的改进: 1. 阻止声明提前，2. 添加了块级作用域

原理: 匿名函数自调！

const: 常量, 一旦创建，值不能改变的变量

ES6中的const也具有let的特征

参数增强: 3项:

1. 默认值:

function fun(形参1,...,形参n=默认值)

调用时，如果没有给形参n传参，则用默认值代替

兼容: if(形参n===undefined)

形参n=默认值

强调: 带默认值的参数，必须放在参数列表的末尾

2. 剩余参数: rest

代替arguments！

arguments问题:

1. 不是数组类型，数组家api不能用！

2. 只能获取全部参数值，不能选择部分

如何使用rest: 2步

1. 定义函数时: function fun(形参1,..., ...数组名)

2. 函数内: 数组名 只接到除前几个参数外，剩余的参数值

兼容: 用回arguments

将类数组对象转为数组:

原理: 用slice复制整个类数组对象中所有元素

[].slice.call(arguments[,starti])

3. 散播:spread

代替apply，打散数组参数

apply的问题: 其实apply不是专职打散数组参数

是专职替换函数中的this，顺带能打散数组

如何: 调用时: fun(实参1,..., ...数组)

Math.max (...arr)

兼容: fun.apply(obj,数组)

箭头函数:

代替回调函数

如何: 口诀三句话:

1. 去掉function，换=>

2. 如果只有一个参数，可省略()

3. 如果函数体只有一句话，可省略{}

如果仅有的一句话还是return，则必须省略return

特例: 内外this通用！

事件处理函数不能改为箭头函数！

兼容: 就是用回function

模板字符串:

代替字符串拼接

ESLint中规定不允许使用+拼接字符串，必须用模板字符串

如何:

1. ``包裹字符串模板

2. 内部支持换行

3. 只是动态生成内容: ${js表达式}

解构:

简化变量的批量赋值

3种情况:

1. 数组解构: 将数组中的元素，提取出来，分别保存到多个变量中

如何: 下标对下标

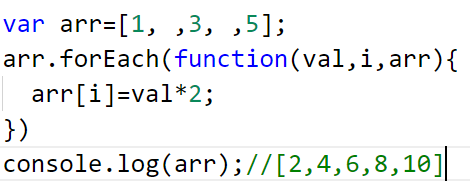
课后任务:

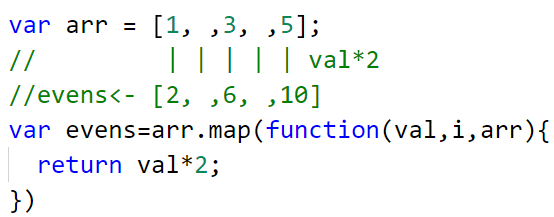
(1)复习: 收集整理js笔试题

(2)作业: 完成课后练习:  
 题目要求:

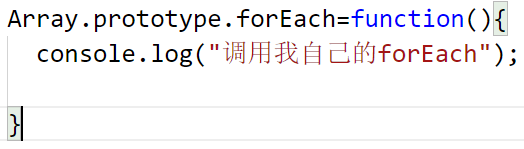
1. 让旧浏览器也支持forEach()和map()

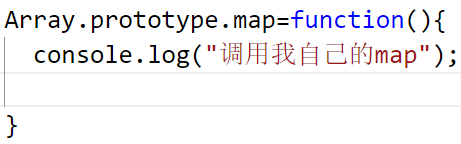
运行效果:





提示:





(3)项目:

题目要求:

运行效果: