前端的技术栈类比



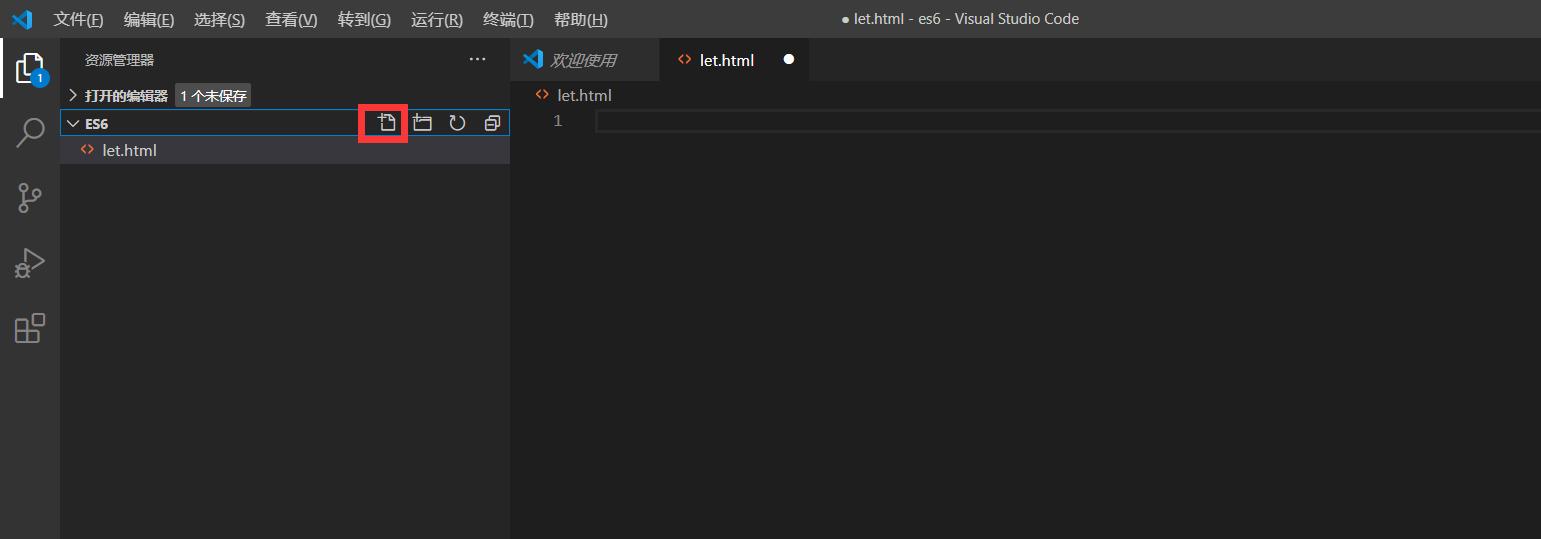
1. ES6
   1. 简介

ECMAScript 6.0（以下简称 ES6，ECMAScript是一种由Ecma,国际(前身为欧洲计算机制造商协会,英文名称是European Computer Manufacturers Association)通过ECMA-262标准化的脚本程序设计语言）是JavaScript语言的下一代标准，已经在2015年6月正式发布了，并且从 ECMAScript 6开始，开始采用年号来做版本。即 ECMAScript 2015，就是 ECMAScript6。它的目标,是使得JavaScript语言可以用来编写复杂的大型应用程序,成为企业级开发语言。每年一个新版本。

所以，ECMAScript是浏览器脚本语言的规范，而各种我们熟知的js.,语言,如JavaScript则是规范的具体实现。（类似于jdbc是规范，mysql是实现。

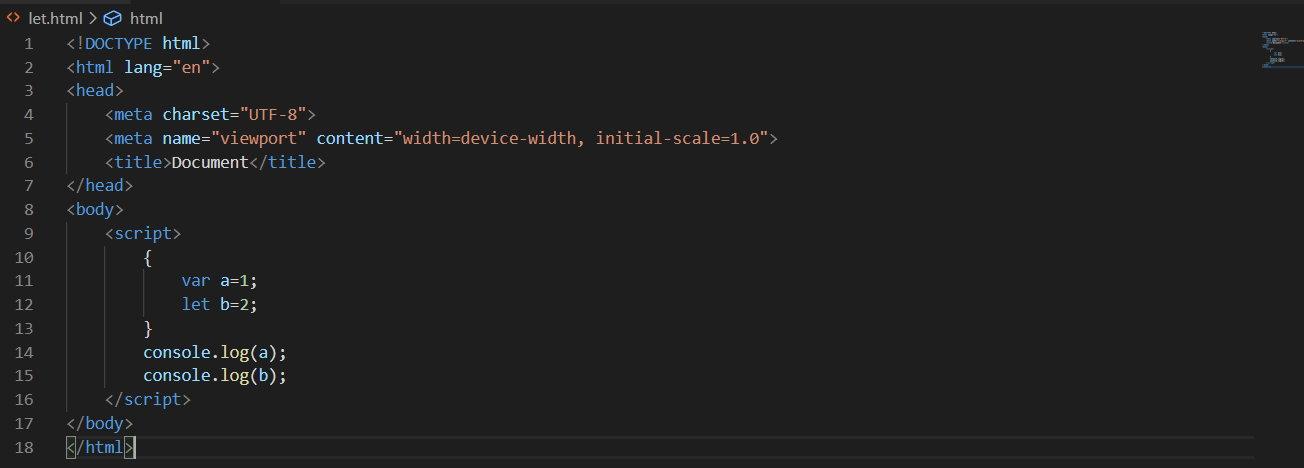
* 1. 语法新特性

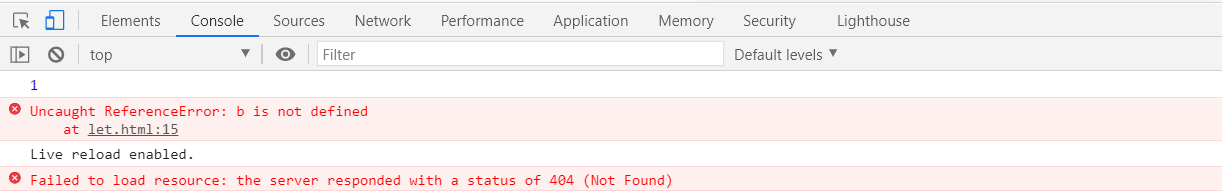
选择新建文件夹，新建一个文件夹然后打开，点击添加一个新文件



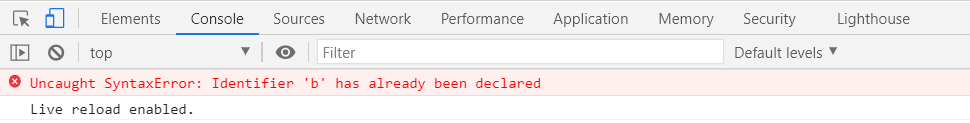
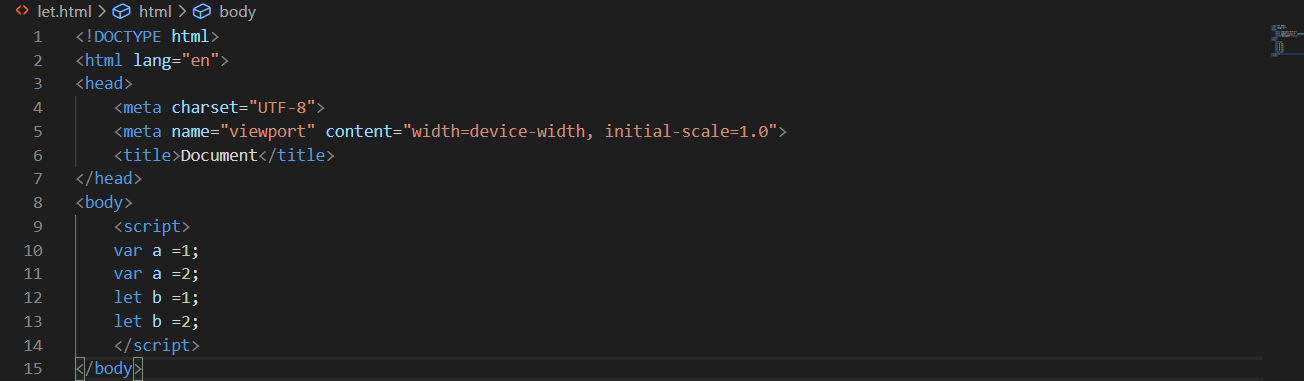
·新特性let

1.let不会越域





2.var可以声明多次，但是let声明的变量只能声明一次

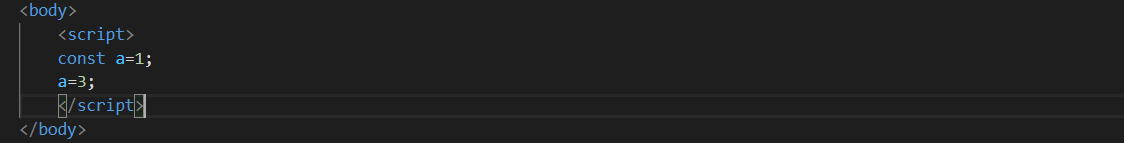


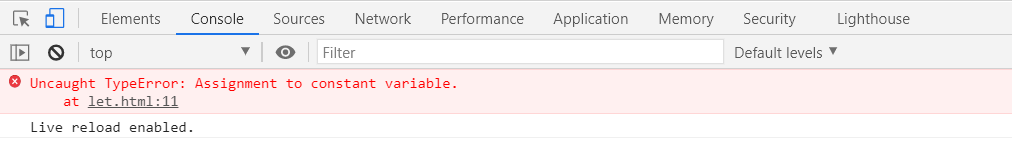
1. var存在变量提升，let没有



·新特性const（常量）

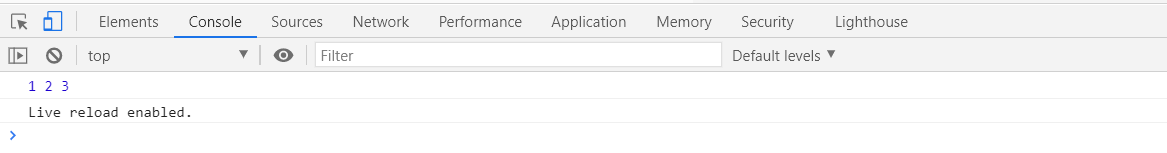
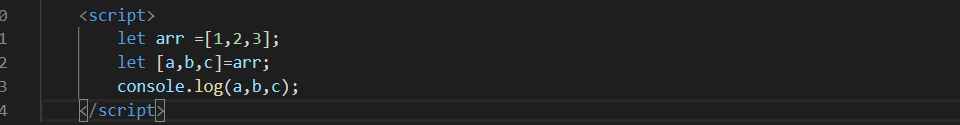
1. 一经定义就无法在改变值



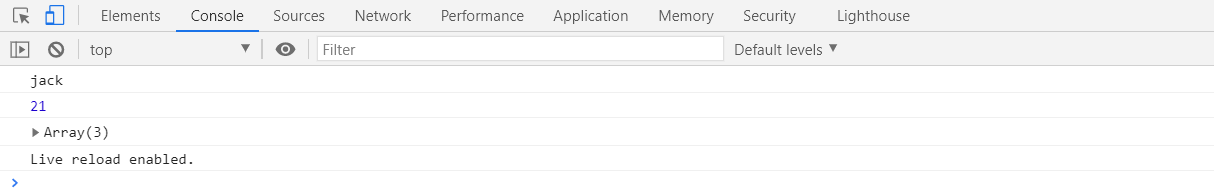
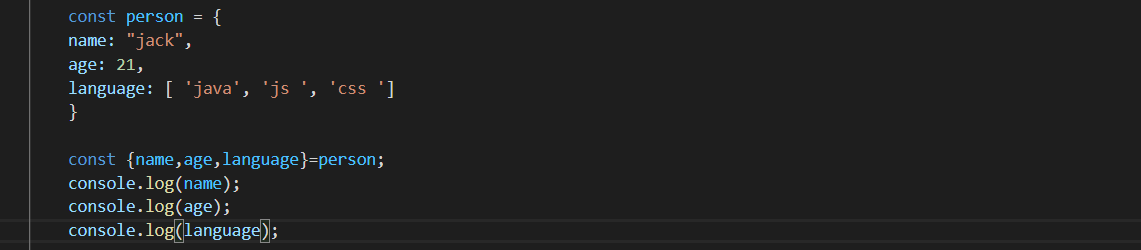


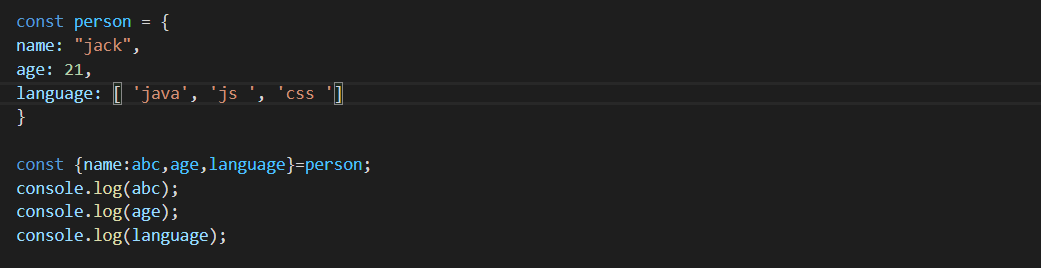
·新特性：结构表达式

1. 数组结构

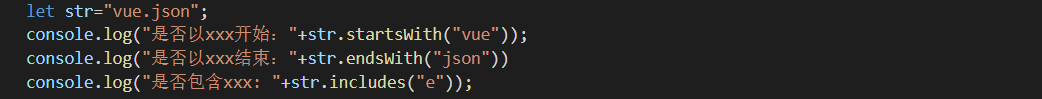


1. 对象结构：这里属性必须和对象的属性一一对应

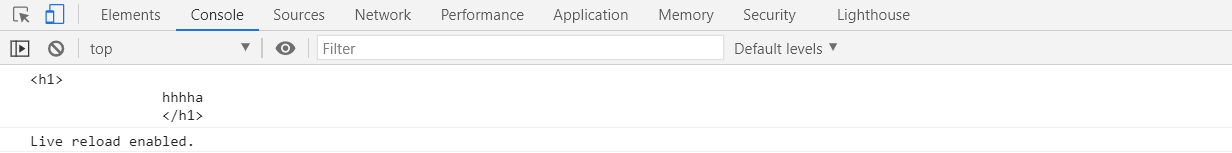
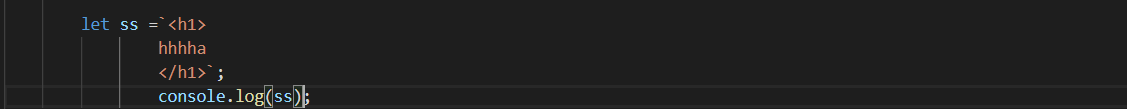
也可以和对象属性不一样，只需要在属性后加上：



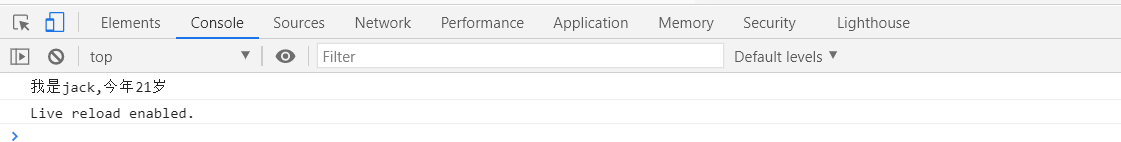
1. 字符串扩展

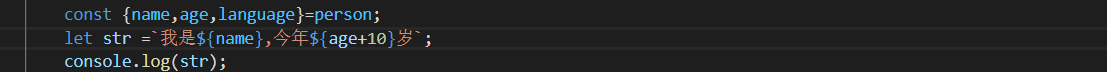


1. 字符串模板：不用再拼字符串了

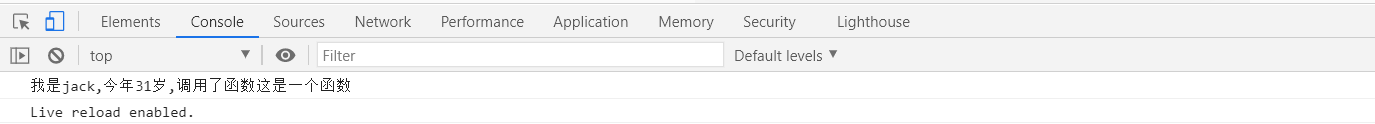
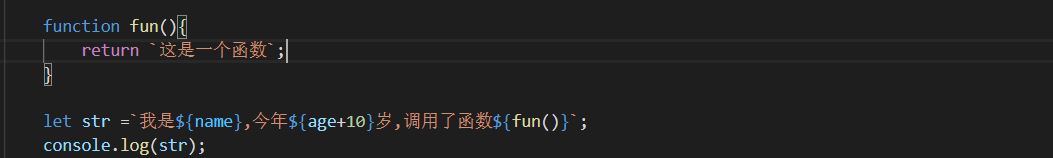


1. 字符串插入变量

#还能加入表达式



#甚至加入函数

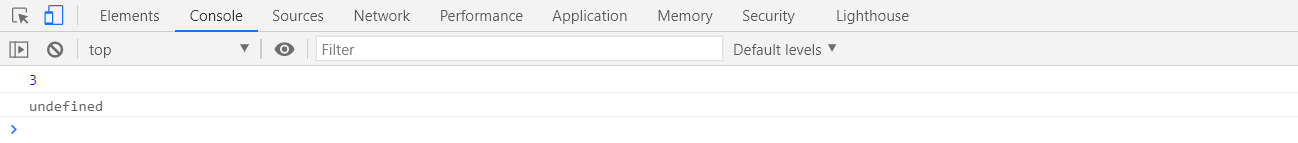


·新特性：函数优化。

1. 函数参数默认值：如果不传值，就启动默认值

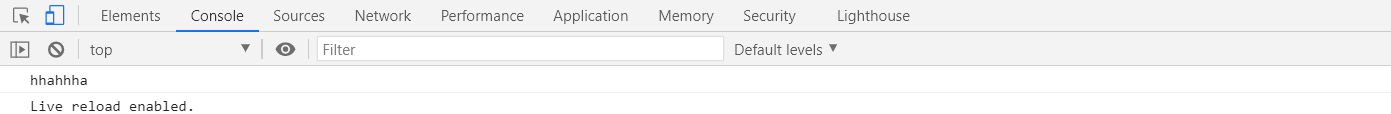


1. 不定参数



1. 箭头函数：兰姆达表达式（var 方法名 =(参数a，参数b)=>参数a+参数b）

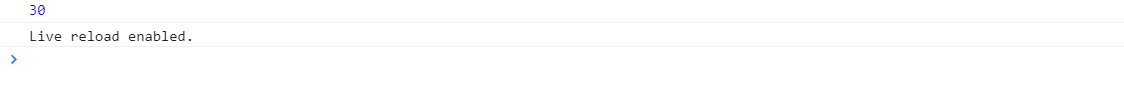
·无法使用this



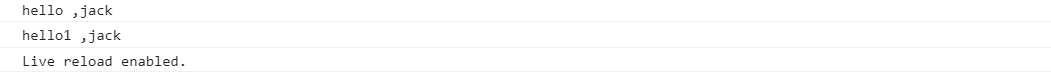
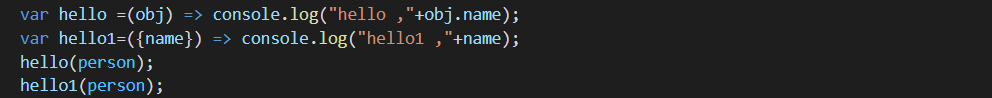
两个参数的箭头函数



多行计算也可以使用



也可以提取出对象的属性用来输出



·对象优化

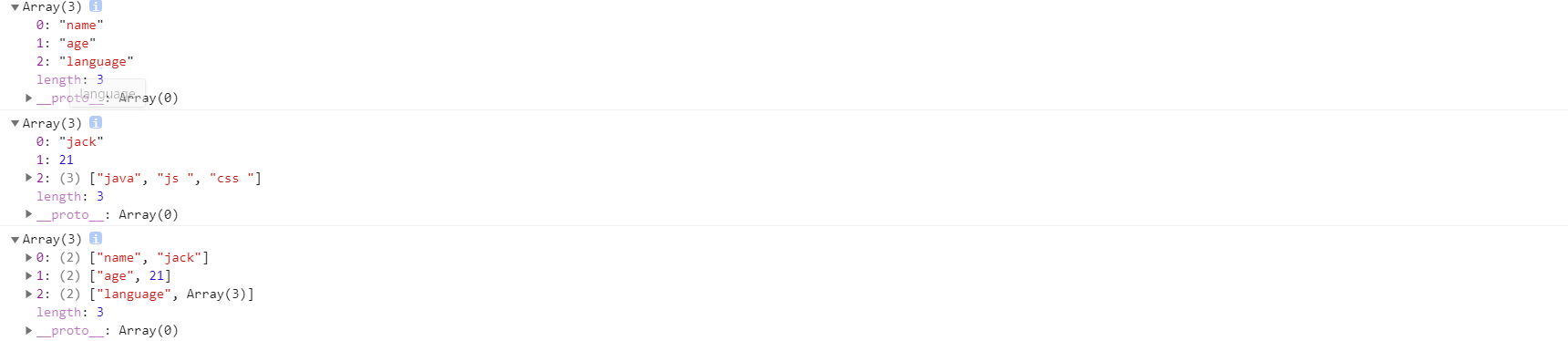
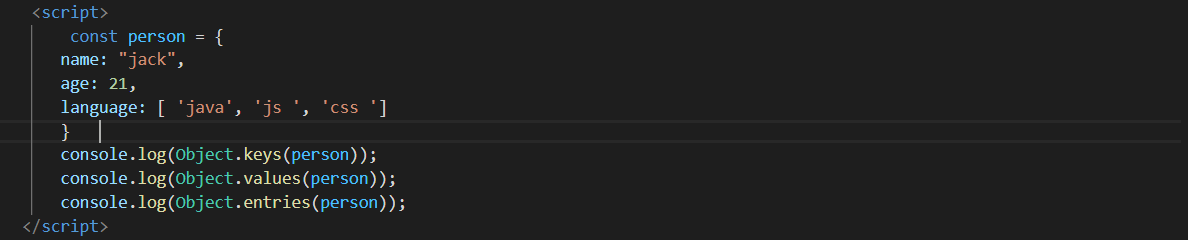
1. 获取对象的键，值，键值

ES6给Qbject拓展了许多新的万法，如:

- keys(obj):获取对象的所有key形成的数组

- values(obj):获取对象的所有value形成的数组

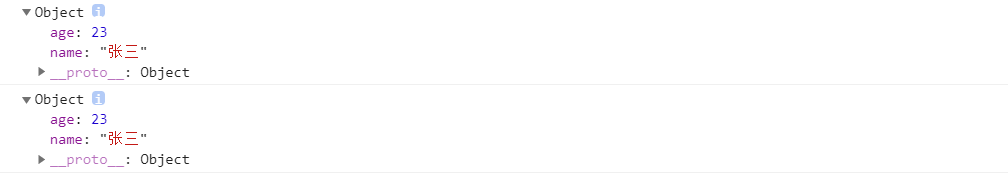
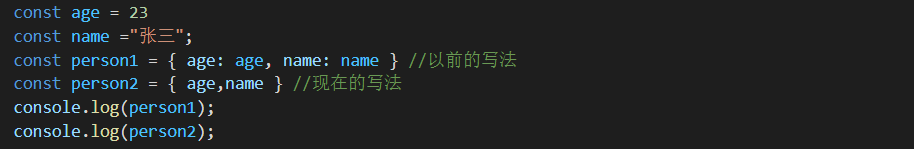
- entries(obj):获取对象的所有key和 value形成的二维数组。



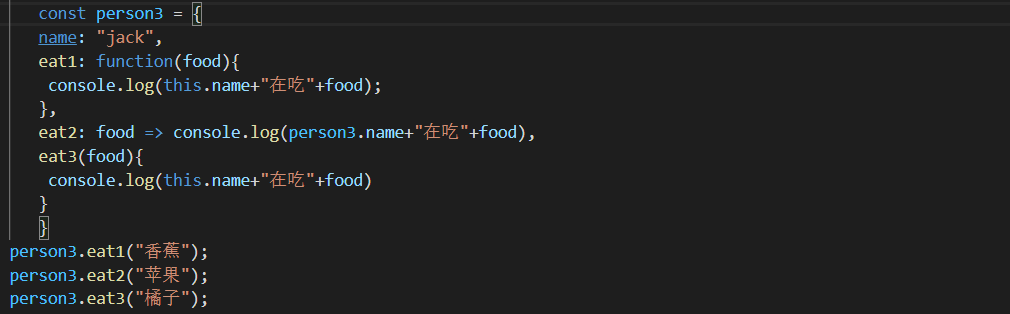
1. 对象合成，Object.assign();



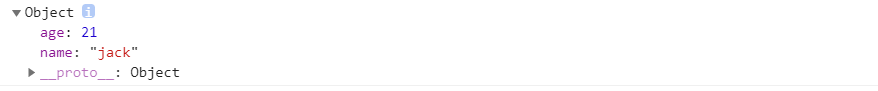
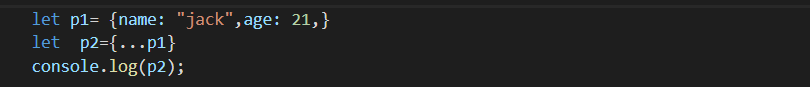
1. 声明对象简写



1. 对象函数简写：记得这里的箭头函数不能用this



1. 深度拷贝，就是连同属性全拷贝

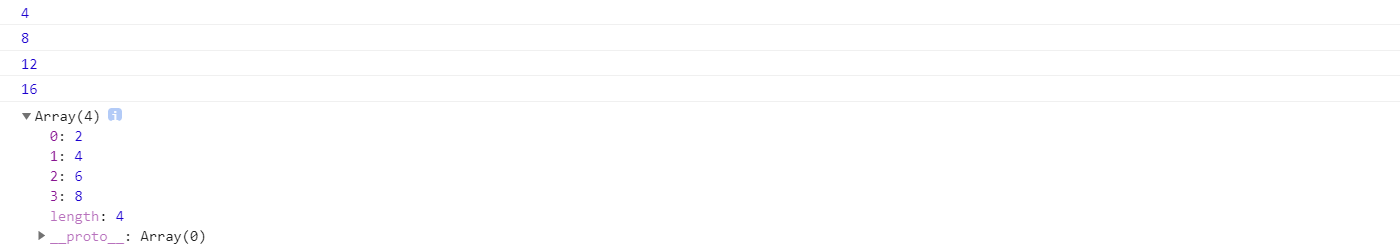


多个深度拷贝，如果这个属性原本的对象里面也有，那么这个属性会被覆盖

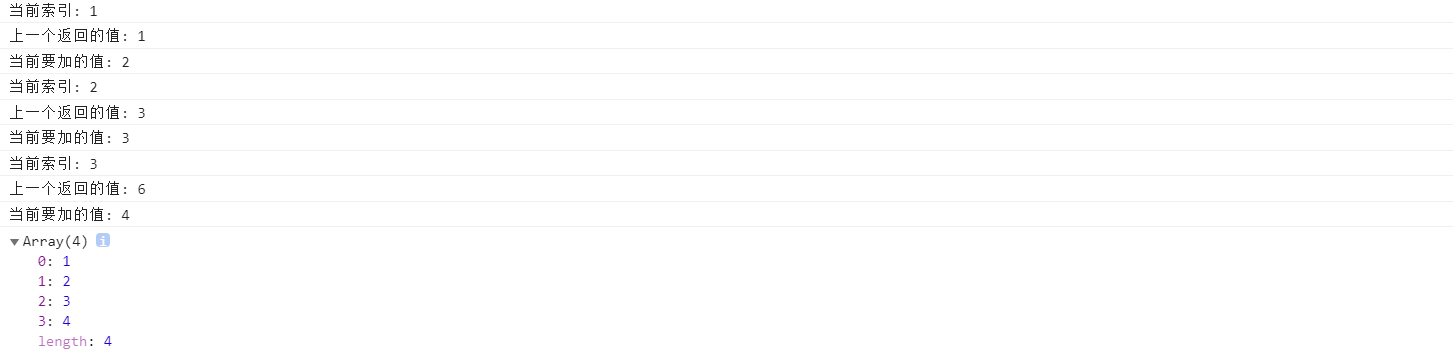
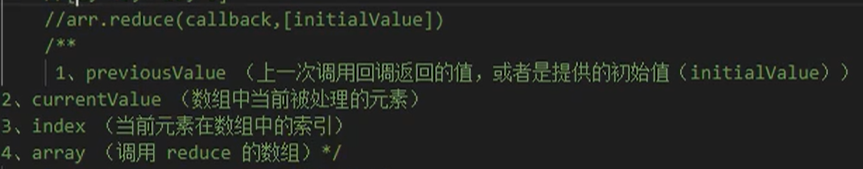


·数组优化

1. map：接收一个函数，将原数组中的所有元素用这个函数处理后放入新数组返回。



1. reduce：为数组中的每一个元素依次执行回调函数，不包括数组中被删除或从未被赋值的元素。



·Promise：将原本繁琐的连锁查询变得不那么繁琐。。。

1. 首先先引入jQuery依赖，然后写原本的ajax请求（旧版）

<script src="https://cdn.bootcss.com/jquery/3.4.1/jquery.min.js"></script>



1. 采用promise查询结果

function get(url,data){

           return new Promise((reso,reject)=>{

               $.ajax({

                   url:url,

                   data:data,

                   success:function(data){

                    reso(data);

                   },

                   error:function(err){

                       reject(err);

                   }

               })

           })

       }

       get(`mock/user.json`)

            .then((data)=>{ //这个代表上一次的查询结果，即学生的json

                console.log("用户查询成功: ",data);

                return get(`mock/user\_corse\_${data.id}.json`);

                //这里返回的是下一个查询结果的promise

            })

            .then((data)=>{//这个data是课表的json

                console.log("对应的课表是:",data);

                return get(`mock/corse\_score\_${data.id}.json`);

                //这里返回的是下一个查询结果的promise

            })

            .then((data)=>console.log(`最终查询结果： `,data));

·模块化：

模块化就是把代码进行拆分，方便重复利用。类似java中的导包:要使用一个包，必须先导包。而s中没有包的概念，换来的是模块。

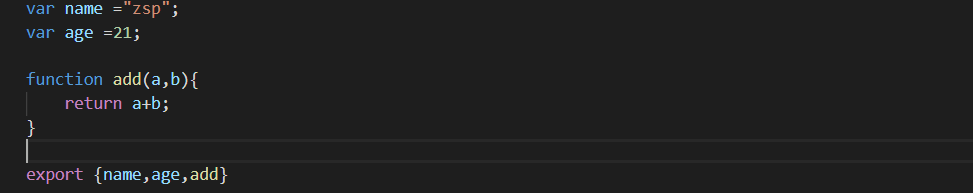
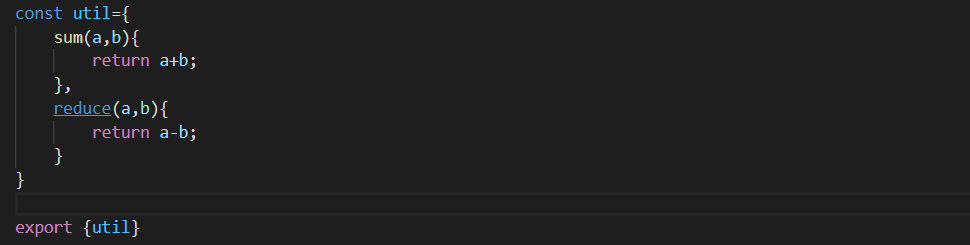
模块功能主要由两个命令构成:'export`和`'import'。

'export`命令用于规定模块的对外接口。

`import`命令用于导入其他模块提供的功能。

~导出可以导出变量，方法，工具类

export {name,age,add}



导入在引入的地方引入即可

import util from `./hello`

import {name,age,add} from `./add`

