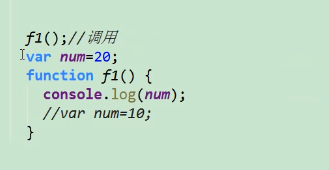
1. 如果for循环不写var就是一个隐式全局变量-->

for(i = 0; i < 10; i ++) i是一个隐式全局变量

1. 预解析

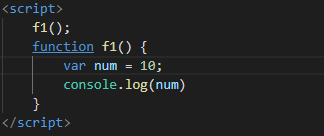
1>.在当前作用域，javascript代码执行之前，浏览器首先会把所有带var和function声明的变量进行提前声明或者定义-->



打印出的num为undefined。

1. .var和function预解析的区别：

var声明的变量和function声明的函数在预解析的时候有区别，var声明的变量在预解析的时候只是提前的声明，function声明的函数在预解析的时候会提前声明并且会同时定义。也就是说var声明的变量和function声明的函数的区别是在声明的同时有没同时进行定义。-->



打印出的结果为10。

1. 创建对象的三种方式(可以多次使用) （instanceof判断类型）
2. .调用系统构造函数创建对象(不能识别什么类型)

var 变量名 = new Object() //用个变量接收叫实例化对象-->



1. .工厂模式创建对象(能批量生产对象,且能识别类型，但繁琐)-->

function 变量() { var obj = new Object(); 添加属性方法在此添加 return obj;}

var 变量接收 = 变量() (函数执行); (接收变量和变量可以不同)-->



注意:return obj一定要写不然打印出的对象是undefined

1. .自定义构造函数创建对象(解决了前面所有毛病)-->

function 大写变量() { 添加的属性方法在此添加 } var 变量(小写) = new 大写变量(函数执行);-->



4>．new对象时候详细说明



1. 通过字面量创建对象(确定只可以使用一次)-->

var obj = {}

1. 内置对象Math(大部分用来处理数字) --> 查看js文档百度搜索MDN--》

<https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Math>

注意：Math不是一个构造函数不像Object -->

不能使用new Math实例化一个方法(报错)

调用里面的方法Math.方法，new Object是使用实例化变量.方法来调用

Math.PI();(PI方法在里面已经定义好了) obj.see();（需在里面定义了）

封装一个Math.max()方法-->



1. Date的基本使用
2. .new Date() 获取当前时间--->





1. .new Date(string) 传入字符串参数的时间 8:00 -->





注意:需要是字符串，获取的是时间类型的值

1. .计算时间-->





注意：是数字类型 为了计算，时间值和时间值不可计算可以用其计算时间



1. Data常用的方法
2. .获取年的方法用getFullYear()，.getYear();方法不常用(1900到现在的年数);-->





1. .获取月份方法getMonth()

注意：获取的月份从0开始要 +1才是当前的月份 例子中真实是7月-->





1. .获取星期几getDay-->





1. .获取日期getDate() -->





1. .获取小时,分钟，秒



1. .其他方法

+日期









+小时分钟秒









+毫秒数





和Date.now()区别

-- Date.now和new Date是同级， 而value需先new Date在使用

1. 字符串string和字符串对象(new String())

-->

 输出还是hello 无法改变(只可以读，不可写)

-->

栈指向了新的堆

输出的结果是”小苏好邪恶”

1. 字符串常用的方法(都是静态方法)
2. .charAt(index) 返回对应字符串的第index位字符-->





注意: 超过字符串长度(length-1)返回空字符串,空格也占一个字符索引位。

1. String.fromCharCode(ussc码, ussc码, ...)返回参数对应的USSC码对应的字符串





1. .concat(string, string, ....)字符串拼接





1. .indexOf(要找的字符串, [从某个位置开始找(索引值)]) 找参数字符串的索引位(没找到返回-1) lastIndexOf()从后先前找，获取的索引仍然是从左往右数





1. .replace(“old string”, “nwe string”)方法 把old string替换成nwe string





注意：不用正则只替换一次

1. .slice(“从第几位”, “到第几位”)返回提取后的字符串(第几位都是索引)





1. .split(“删除什么字符串”, [“切割后留下几个”] )按什么字符串拆分 返回多个数组





1. .substr(“开始的位置”, [“个数”])返回截取后的新字符串





1. .substring(“开始的索引”, “结束的索引”), 返回截取后的字符串，不包含结束的索引字符串





1. .toLocaleLowerCase(); 把字符串转为小写的 和toLocalecase();一模一样





1. .toUpperCase(); 把字符转为大写





1. .trim(); 去除字符串两端的空格





1. 静态方法(Math.radaio())和实例方法(new Date().方法)



1. 常用的数组方法
2. . A instanceof B(为Array或者Object) 检测A是否属于B -->





注意：数组是特殊的对象上例子中写 [] instanceof Object 返回值为true

1. Array.isArray(数组)判断是否是数组





1. .concat() 链接两个数组





1. .every(function () {}) 可用于数组判断 数组中的每个元素都会执行这个函数 函数的第一个参数代表元素的值, 第二个代表元素的索引, 第三个是原数组(谁调用的就是谁一般用不到第三个参数) 返回的是布尔值, 所有都满足条件才返回true





1. .filter(function () {})筛选 返回符合设定条件的值成数组(新数组)形式 数组中的每个元素都会执行这个函数 函数的第一个参数为元素的值, 第二个元素的索引, 第三个参数为遍历的数组(原数组)





1. .push(值); 把值追加贷数组中(最后面), 返回的是追加后数组的长度



arr为 返回值为

1. .pop(); 把删除数组中最后一个元素的值, 返回值为删除的这个值



arr为 返回值为

1. .shift();删除数组中第一个元素, 返回值为删除的这个值



arr为 返回值为

1. .unshift(值); 向数组中的第一个元素前插入值, 返回的是插入后数组的长度



arr为 返回值为

1. .forEach(function () {}) 循环数组(相当于一个for循环) 第一个参数为元素的值, 第二个为元素的索引, 第三个参数为原数组
2. indexOf(元素值) 返回的是元素的索引, 没有这个元素值则返回-1





1. join(“字符串”) 返回数组每一位按字符串连接的字符串





注意区分字符串拆分成数组split方法-->





1. .map(function () {}) 数组中每个元素都会执行这个函数，执行后的结构重新放到一个新数组中返回出来





1. .reverse() 返回数组翻转之后的数组





15>.sort(function () {})排序不稳定, 可传入一个比较函数可以进行排序

16>.slice(“开始”, “结束“);截取数组从开始到结束(不包括结束那个元素)





17>.splice(“开始”, “要删除的个数”, [“截取后添加的元素”])截取切在截取位置添加元素



arr为  返回值为

1. 基本包装类

普通变量不能直接调用属性和方法, 而对象可以直接调用属性和方法

基本包装类型：数字(number), 字符串(string), 布尔(boolean)

这些普通类型的变量会默认会调用包装

包装类: 普通类型的变量经过包装可以直接调用属性和方法 -->

