

js高级程序设计

数据类型

script元素的属性

文档类型doctype

- async只适合外部脚本，表示立刻下载脚本，但是不保证他们的执行顺序
- defer延迟到整个页面解析完成后再执行，告诉浏览器立即下载，但是延迟执行
- src表示包含要执行带的外部文件
- type表示编写代使用的脚本语言的内容类型(MIME),一般为type='text/javascript'
- charset，指定代码的字符集，utf-8

- 混杂模式quirks model
- 标准模式 standards model
- 两种模式主要影响css,但是所有浏览器默认开启的是混杂模式

- undefined, 一般而言不会把一个变量，显示的定义为undefined，let message // 默认取得了undefined值
- null, 表示一个空对象指针，所以typeof null 是一个 object, 一般如果我们定义一个变量接收一个对象时，初始化赋值一般给个null，这样做类型判断时null也是object
- Boolean类型，ecmascript 中所有类型数据都有对应的boolean值，任何数据类型的值都可调用Boolean()函数，总是会给你一个true或者false. 0, NaN, "", undefined, null 这五种数据转为 false

Number类型，采用IEEE754格式来表示

- 浮点数值需要保存整数内存的两倍
- 在默认情况下，ecmascript 会将小数点后面带有6位以上的浮点数值转为以e表示的数值0.0000004，默认表示为4e-7
- 所以在这种情况下，永远不要试图进行两个特定浮点数的比较，0.1 + 0.2 永远不等于0.3
- 这是因为ecmascript采用IEEE754格式来计算浮点值的缘故

NaN (Not an Number), 表示一个本来要返回一个数值的操作，而未返回数值的情况

- 任何数值除以非数值返回NaN
- 任何涉及到NaN的操作都会返回NaN
- NaN和任何值都不相等
- isNaN()方法，检测一个值是否“不是数值”

数值转换，Number(), parseInt(), parseFloat()

Number()可以将任何数据类型转为数值

- boolean值的话true1, false0
- 数值，什么传进，什么传出
- null 转为 0
- undefined 转为 NaN
- 空字符串转为 0
- 字符串只包含数字，那么会转成对应的数值
- 如果字符串中既包含数字又有字母，那么转为NaN

parseInt和parseFloat 专门用于将字符串转为数值

- parseInt, 会取第一个非空格字符，如果是数字或负号，则会转为对应的数字，会忽略非数字字符，但是如果第一个是空格字符，那么直接转为NaN不要给我面子
- parseInt可以将数值转为对应的二进制，八进制，十六进制等，parseInt('12', 2)
- parseFloat 直支持十进制，因此他没有第二个参数，其他基本和parseInt一样

String类型, 0个或多个16位Unicode字符组成的字符序列

- 将一个值转为字符串有两种方法，一个是调用几乎每个值都有的toString()方法，但是null, undefined，没有这两个方法
- 但是可以用String()函数，这个函数可以将任何值都可以转为字符串类型 String(null) // 'null', String(undefined) // 'undefined'

Object对象类型

Object实例的属性

- let obj = new Object()
- constructor 保存这创建当前对象的函数
- hasOwnProperty(propertyName) 检查当前的属性是实例中的而不是原型链中继承的，判断这个财产是自己挣的，还是他爹给的
- isPrototypeOf(obj) 用于检查传入的对象是不是当前对象的原型
- propertyIsEnumerable(propertyName) 用于检查给定的属性是不是能用for-in遍历，是否是可枚举的属性，判断当前对象是不是经得起瘦身，自己有没有鬼
- toLocaleString(), toString()返回对象的字符串表示, valueOf() 返回对象的字符串，数值，布尔值表示

Array类型

分支主题

Array类型

- 创建数组的两种方式，使用Array构造函数，let arr = new Array(3), 传递一个数值的话，则创建一个长度为数值的数组，如果let arr = new Array('username', 'iphone') // ['username', 'iphone']
- arr.length属性，可读写
- 数组检测，可以用 instanceof 和 Array.isArray()推荐
- 数组的toString()和valueOf(), join() 都是将数组转为字符串，toString()和valueOf() 都是默认将数组以逗号形式隔开以字符串形式返回，join() 允许传一个参数作为分隔的数组项，返回一个数组，如果不传递参数就默认为逗号形式分隔
- 栈方法LIFO(last in frist out) 后进先出，push()推入，pop()弹出，push方法可以传递任意多个参数，将参数添加到数组的末尾，返回的是修改后的数组长度，pop弹出数组的最后一项，返回的是弹出项
- 队列方法FIFO(frist in frist out) 先进先出，push() 推入，shift() 转移第一个，shift() 返回转移的项
- 反队列unshift()和pop, unshift 向数组第一项添加任意多个项，并返回新数组的长度，使用pop向数组末尾弹出最后一项
- 重排序方法，reverse, sort, reverse()方法翻转整个数组， 但是不够灵活，因此有了sort() 方法，升序排列数组项，最小值在最前面，最大值在最后面，sort() 的每一项都会调用toString()方法转型，然后在用得到的字符串比较确定怎么排序，但是这种方式也是比较僵硬不够灵活，因此此方法可以接受而一个函数这个函数接收两个参数，如果第一个参数应该位于第二个参数之前则返回一个负数
- 分支主题

变量，作用域和内存

- 函数传递参数的时候，如果传递引用类型的话，那么传递的是引用类型的指针，其实修改参数，就等于修改原始对象
- js 中的垃圾回收机制，标记清除法，当变量进入环境时，标记为进入环境，当变量离开环境时，将其标记为离开环境，可使用任何方式来标记变量，采用什么变量标记不是重点，重点是怎么采用什么策略标记，标记清除法是目前浏览器主要的内存回收机制。
- 采用引用计数策略，跟踪记录每一个值被引用的次数，当一个变量赋值，那么这个值的引用计数加1，那么当这个变量去另外一个值时，这个值的引用次数减1，直到这个值的引用次数为0, 那么没有办法就可以回收内存空间，目前浏览器已经放弃使用这种方式回收垃圾。
- 垃圾回收机制是周期性运行的，如果为变量分配的存储数量很可观，那回收工作量也是非常大的，因此垃圾回收的周期间隔是个非常重要的。一般如果一个变量已经没有使用，可以将其赋值null, 解除引用。