```
数组的length属性是可读写的,如果一个长度为2的数组,设置length为3,那么第三项为undefined,如果设置为1,那么会删除第二项
                                      检测数组Array.isArray()和instanceof
                                       array to string ,通过toString()和valueOf(),join(),方法可以将数组转为字符串,当然join()不传参数的时候,默认是使用逗号隔开,如果使用参
                                      数的话,则把参数作为分隔转为字符串
                                      栈方法,LIFO (last in first out)后进先出,push推入pop弹出,push可以推入任意多个元素,返回的是新数组的长度,pop只能一个一个的删
                                      除数组的最后一个元素,返回的是删除项。
                                      队列方法,FIFO(first in first out) 先进先出,push推入和 shift转移,使用shift 转移数组第一项,并返回该项
                                      反队列方法, unshift和pop, unshift()向数组开头添加任意多个元素,并返回新数组的长度
                                      重排序方法,reverse()和sort(),reverse方法颠倒数组,sort()方法接受一个function,如果第一个参数应该在第二个参数的后面,那么返回的是
                                       一个正数,如果第一个参数应该在第二个参数的前面则返回一个负数。如果两个参数相等那么返回0
                                       操作方法concat(), slice()基于当前对象返回一个新数组,接收两个参数,左闭右开
                                      2. 地表最强array操作方法splice(), 可删除任意项, 可插入任意项, 可替换任意项, 可传递两个以上的参数, 第一个参数startIndex。第二个参数要
                           Array
                                       删除的项数deleteNumber, ,第三个及以后的参数为要从startIndex开始添加的参数
                                      位置方法, indexOf(), lastIndexOf()都可接受两个参数,第一个参数要查找的项,返回第一次出现的位置,第二个参数为查找的起始位置
                                      迭代方法, map, filter, forEach, some, every, 这五个方法都接收两个参数,第一个参数是回调函数,第二个表示该函数的作用域(可选的)--主要
                                      影响this, 其中回调函数接收三个参数 , item, index, arr
                                      every()对数组中的每项都执行回调函数,都返回true,那么every()返回true,否则返回false
                                      some()对数组中的每项都执行回调函数,有一个返回true,那么some()返回true,否则返回false
                                      filte()对数组中的每项都执行回调函数,返回该回调函数会返回true的数组成员组成的数组
                                      map() 对数组的每项都执行回调函数,返回每次回调函数执行的结果组成的结果
                                      forEach(), 对数组中每项都执行回调函数,没有返回值
                                      归并方法 reduce(减少,压缩), reduce Right,都是会迭代数组的每项,然后返回一个最终的返回值
                                      reduce方法从数组的第一项开始,逐个遍历到最后,reduceRight相反,都接受两个参数,一个是每项都执行的回调函数,第二个是作为归并基础
                                      值,其中回调函数接收四个参数,第一个prevent,current,index,arr,前一个值,当前值,索引,数组,当reduce第二个参数给了,那么就作
                                      为prevent , 否则prevent 为数组第一项 , current 为数组第二项
                                                            创建正则表达式有两种方式,一种是字面量方式,斜杠开始,let regexp = /abc/igm,斜杠结束,还有一种是用RegExp构造函数let regexp = ne
                                                            w RegExp('abc', 'igm')
                                                            表达式的属性分为两类,一类是igm修饰符属性, RegExp.prototype.ignoreCase忽略大小写, RegExp.prototype.global全局匹配, RegExp.protot
                                                            ype.mult iline多行模式,会修改^和$的匹配规则,都返回一个boolean表示设置了相应的igm属性,另一类的属性,RegExp.prototype.lastInd
                                                            ex只进行连续的搜索有意义,可读可写,表示接下来的搜索位置, RegExp.prototype.source, 正则表达式的字符串表示, 不包含反斜杠, 只读属
                                                            实例方法,RegExp.prototype.test()返回一个布尔值,表示当前模式能不能匹配参数字符串,RegExp.prototype.exec(),返回匹配数组结果,成
                                                            员是匹配成功的子字符串
                                                            字符串的实例方法, String.prototype.match() , String.prototype.search(), String.prototype.replace() , String.prototype.split () 方法
                                                                        字面量字符和元字符字面量匹配方式/a/匹配a,/b/匹配b,
                                                                        元字符 有 1 , 点字符(.), 匹配除了回车 , 换行 , 行分隔符 , 段分隔符 , 以外的所有字符。 2 ,位置字符 ( ^表示字符串的开始位置, $表示字符串的
                                                                        结束位置)3 ,选择符( | 表示或or的关系 )。 元字符还包括 \\, \*, +,?,(),[],{}
                                                                        转义符,如要匹配元字符,那么就要在元字符前面加上反斜杠\,\\+,那么匹配 +,正则表达式中需要转义的元字符有 ^,$,+,.,(,),{,[,|,*
                                                                        特殊字符, \n 匹配换行键。\r 匹配回车键。\t 匹配制表符 tab (U+0009)。\v 匹配垂直制表符 (U+000B)。\f 匹配换页符 (U+000C)。\0
                                                                        匹配null字符(U+0000)。
                                                                        字符类 class ,表示一系列的字符可供选择,只要匹配其中一个就可以,形式为 [xyz]表示x、y、z之中任选一个匹配 ,1,脱字符 ( ^ ) 表示除了
                            RegExp regular expression
                                                                        字符类中的字符都可以匹配,[^xyz]表示除了x、y、z之外都可以匹配 2,连字符(-)表示字符的连续范围
                                                                                      \d 表示匹配0到9的数字[0-9]
                                                                                       \D 表示匹配除了0到9的数字[^0-9]
                                                                                       \w表示匹配字母, 数字, 下划线, [0-9a-zA-Z]
                                                            匹配规则
                                                                         预定义模式
                                                                                       \W 表示匹配除了字母,数字,下划线,以外的字符[^0-9a-zA-Z_]
                                                                                       \s 表示匹配空格符(换行符,制表符,空格符), [\r\n\t\v\f]
                                                                                       \S 表示匹配非空格符, [^\t\r\n\v\f]
                                                                         重复类精确匹配次数,{},{n}表示恰好重复n次,{n,}表示至少重复n次,{n,m}表示重复不少于n次,不多于m次。
                                                                         量词符 ?有或没有,问号表示某个模式出现0次或1次,等同于{0,1},*随便多少次, 星号表示某个模式出现0次或多次,等同于{0,}。 , + 表示
                                                                         一次或者多次,加号表示某个模式出现1次或多次,等同于{1,}。
                                                                         贪婪模式,默认情况下是最大可能的匹配,直到匹配不满足为止,这种方式为贪婪模式,默认是贪婪模式
                                                                        组匹配,括号表示分组匹配,括号中的模式可以用来匹配分组的内容。
                                                                        先行断言x(?=y), 只有在出现在y之前才匹配, y不会被计入返回结果。比如,要匹配后面跟着百分号的数字,可以写成(d+(?=%))。
                                                                        先行否定断言x(?!y), x只有不在y前面才匹配, y不会被计入返回结果。比如, 要匹配后面跟的不是百分号的数字, 就要写成/\d+(?!%)/。
                                         toString() 用字符串形式表示和 value Of() 返回字的原始值也是用字符串表示i
                           Number
数据类型
                                         toFixed()方法保留几位小数,返回的是字符串的形式
                                                               URI 统一资源表示符的编码方法 , encodeURI() 和 encodeURIComponent () 对URI 进行编码
                           global 对象,最特别的一个对象
                                                               encodeURI不会对特殊字符进行编码如冒号,正斜杠,问号,并号等,但是encodeURIComponent会对任何非标准字符串字符进行编码
                                                               与上两个方法对应的是decodeURI 和 decodeURIComponent 进行解码
                                       string 中的空格也是占一个字符的
                                       访问字符串特定位置的字符, chart At (index)
                                       串联多个字符串, concat(string1, string2, ...), 也是对于原字符串没有影响
                                       slice, substring, substr, 基于当前字符串创建子字符串, slice 和 substring substr 都接受一个或两个参数,第一个参数为开始的位置,第二个为
                                       到哪里结束, slice 和 substring, 第二个参数为结束的位置, 而substr第二个参数为返回字符串的个数。这三个方法都对原字符串没有影响
                                       indexOf和lastIndexOf查找给定字符的位置, indeXOf('stirng', index)
                                       trim() 删除前置和后缀的空格
                           String
                                       字符串大小写转换, toLowerCase(), toUpperCase()
                                       字符串的模式匹配方法,match()只接受一个参数,要么是正则表达式,要么是RegExp对象, match() 返回一个数组,数组的第一项是与整个模式
                                       匹配的字符串,之后的每项保存着正则表达式中捕获组匹配的字符串
                                       search() 这个方法,也是接收一个正则表达式,返回值是第一个匹配项的索引值
                                       replace (更换,替换)用于替换匹配的字符串 string.replace(/a/g, 'b')
                                       split (分裂,细胞分裂)基于指定的分隔符将字符串分割为多个字符串,并将结果放在一个数组中。分隔符可以是字符串,也可以是正则表达式,s
                                       plit 接收第二个参数,用于指定数组的大小。
                                       通过let str = new String('username') // 方式创建的str 是一个对象Object类型, 这是包装对象, 通过let str = String('usern
                                       ame') // 创建的是一个字符串 , 类似的有 new Number(), new Boolean() ,这三个都是 , 如果添加new 方式创建的是一个对象
                                               函数的属性,有name, length, name 表示函数的名称, length表示函数参数的个数
                                               函数的参数类型有arguments对象,他有callee属性指向的原函数,严格模式下访问arguments.callee,访问错误
                           Function 函数
                                              立即执行函数 (function() {})(); 或者(function(){}()); 不能直接function(){}()这么写的原因是function关键字作为开头被解析为语句,不能以()结
                                               尾,因此会报错,因此用(function(){})(); (function(){}()); 来表示,且不能省去;分号
                                        描述性对象,有六个属性,value, writable默认为true可写属性 configurable可配置属性默认为true, enumerable 可枚举属性默认为true
                                        configurable 可配置性,默认为true,若设置为false,则无法通过在改变该属性的描述对象,也不能删除该属性
                                        enumerable可枚举属性,默认为true,如果设置为false,那么for...in和 Object.keys不会遍历该属性
                                        get 属性, get是一个函数, 表示该属性的取值函数 (getter)
                                        set 属性 , set是一个函数 , 表示该属性的存值函数 ( setter )
                                        Object.getOwnPropertyDescriptor()接收两个参数,第一个是对象,第二个是对象属性,表示获取当前对象属性的描述性对象,只对对象自身
                                        属性有效,继承的属性无效
                                        Object.getOwnPropertyNames()返回一个数组,成员是参数对象自身的全部属性的属性名,不管该属性是否可遍历。这跟Object.keys的行为不
                                        同,Object.keys只返回对象自身的可遍历属性的全部属性名。
                                        Object.defineProperty(), Object.defineProperties()方法允许通过属性描述对象,定义或修改一个属性,然后返回修改后的对象,writable、con
                                        figurable、enumerable这三个属性的默认值都为false。但是通过字面量定义的对象,三个属性都是默认为 true
                                        对象的拷贝,通过Object.defineProperty()和 Object.getOwnPropertyDescriptor()和 obj.hasOwnProperty, 先for in遍历对象,在通过对象
                                        实例 hasOwnProperty() 过滤不是对象自身的属性,然后通过Object.defineProperty() 方法定义新的对象,然后通过Object.getOwnPropertyD
                                        escript or() 方法来获取原对象的描述性属性来给新的对象定义相关的属性
                                        有时候需要定义一些冻结对象,防止对象被篡改,使用的是Object.prevent Extensions()阻止扩展,使得一个对象无法再添加新的属性。Object.se
                                        al() 密封对象,使得一个对象既无法添加新属性,也无法删除旧属性,并不影响修改某个属性的值。 Object.freeze() 冻结对象, 可以使得一个对象
                                        无法添加新属性、无法删除旧属性、也无法改变属性的值,使得这个对象实际上变成了常量。prevent Extensions < seal < freeze
                                                     创建一个空对象,将构造函数的原型赋值给空对象隐式原型,然后将空对象用apply方式改变构造函数中的this。function newFactory() { let obj
                            Object
                                                     = new Object (); let context = [].shift.call(arguments); obj.__proto__ = context.prototype; let result = context.apply(obj, arguments); if the context is context is context.apply(obj, arguments); if the context is context is context is context is context is context in the context in the context is context in the context in th
                                                     ypeof result === 'Object' && result !== null ) return result return obj} function PersonList (name, zone) { this.name = na
                                                     me this.zone = zone } let obj = newFactory(PersonList, 'travel_xcc', 1) 如果 构造函数 有return 一个对象或者return null , 那
                                        new 原理
                                                     么new 一个实例时 会直接返回这个对象或者null
                                                     Object.create() 创建一个对象,可以通过new一个实例,但是很多时候我们需要继承一些对象的方法,又没有找到对象的实例,因此可以通过 O
                                                     bject.create() 方法创建。 let b = { userlist:[] } let a = Object.create(b), 那么 a 继承了b 其实就是将 a.__proto__ = b
                                        prototype 原型对象,每个函数function都有这个属性,prototype对象有一个constructor属性,默认指向prototype对象所在的构造函数
                                        Object.prototype的原型是null。null没有任何属性和方法,也没有自己的原型。因此,原型链的尽头就是null。
                                        instanceof判断一个实例是不是某个构造函数的实例,运算符只能用于对象,不适用原始类型的值。
                                        Object.getPrototypeOf方法返回参数对象的原型。这是获取原型对象的标准方法
                                        Object.set Prototype Of方法为参数对象设置原型,返回该参数对象。它接受两个参数,第一个是现有对象,第二个是原型对象
                                                            construct or 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
                                        Object实例的属性
                                                            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搜身,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
```

创建数组的方式两种,一种是let arr = new Array(2), 创建一个长度为2的数组,let arr = new Array('username', 'loginna

me') // ['username', 'loginname'] , 另一种是通过字面量方式