find方法用于遍历数组，将满足条件的第1个元素返回,如果没有满足条件 元素返回undefined

  // 语法

        数组.find( function(item,index,arr){

            return 条件

        } )

        \*/

        // filter方法作用，用于遍历数组，根据条件进行筛选，最终所有的满足条件的元素会以数组的形式返回  // 语法：

            数组.filter( function(item,index,arr){

                return 条件;

            } )

        \*/

 // reduce方法，用于遍历数组，并进行统计，常用于对数组求和，统计元素个数

        // 语法：

        /\*

            数组.reduce( function(){}, 初始值)

            数组.reduce( function(pre,item,index,arr){}, 初始值)

            函数function(pre,item,index,arr){},会被reduce调用多次(pre 与 初始值)

                首次调用时，reduce会将 初始值传递给pre

                其后各次调用时，reduce会将上一次调用函数的返回值传递给pre

   // 1. every方法

        //  遍历数组，判断数组中是否所有元素都满足指定的条件。

        /\*

        // 语法：

            数组.every(function(item,index,arr){

                return 条件

            })

// 2. some方法

        //  遍历数组，判断数组中是否有满足指定的条件的元素。

        /\*

        // 语法：

            数组.some(function(item,index,arr){

                return 条件

 1. 什么是构造函数:

        //  构造函数就是函数，它是函数的另一种调用形式

 术语：

            实例化      由构造函数得到对象

            实例(对象)  得到的对象就是实例(对象)

    /\*

        Object.keys()       获取键名

        Object.values()     获取键值

        Object.assign()     合并成员

         \*/

        构造命名：如果一个函数后续要使用new来调用，推荐将这个函数首字母大写

 // 字符串.substring(start,end)

      // 在字符串中以 start 为开始 end 为结尾之间的字符，含头不含尾

  // 1. str.startsWith(substr)

        // 判断字符串 str 是否以 substr 开头，是返回true，否则返回 false

        // let url = "http://www.xxx.com/admin/index.html";

        // console.log(url.startsWith('http://'));

        // 2. str.endsWith(substr)

        // 判断字符串 str 是否以 substr 结尾，是返回true，否则返回 false

        // let url = "http://www.xxx.com/admin/index.html";

        // console.log(url.startsWith('.html'));

        // 3. str.includes(substr)

        // 判断字符串str中是否包含substr