1、对象

将一堆信息组合起来，这一堆信息的组合 就称为一个对象

好处：方便用户查找数据，防止全局变量的污染。   对象可以动态的添加  属性 或  方法（功能）

2、构造函数

构造函数  ： 通过new 关键字创建出一个对象 ，  通过new的方式创建这个对象的函数就叫做构造函数。

3、数组

思考： 如何计算任意10个整数的平均数？

数组：  可以将数组理解成一个容器，这个容器中可以存放任意类型的数据。

4、数组定义

var  arr = [];//定义一个空数组

var  arr = new Array(); //定义一个空数组   通过构造函数的方式创建一个数组对象

var arr = new Array(3); // 如果参数是一个数值类型的数据，这个参数表示数组中可以存放数据的最大长度

var arr = new Array(2,3,5); // 多个参数表示数组中的数据

var arr = new Array("a"); // 表示数组中存放了一个 string类型的数据

5、数组赋值

第一种： 定义数组时直接给值

     var arr = [3,2,6,7,89];

     var arr = new Array(2,3,5);

第二种 ： 先定义数组 后赋值 (通过下标给数组赋值)

var arr = [];

arr[0] = 23;

6、js数组特点

1、数组长度不固定

2、可以动态的为数组添加数据

7、数组数据的读取

数组名[下标]

下标 从 0 开始 ，数组最大下标 ： 数组.length - 1

8、数组的遍历

for  循环

for ... in

for(var i in 数组或集合对象){

}

for...in 特点：

     没有循环条件

     自动取出下标

     最小下标 0  ，最大下标 数组.length-1

     下标类型是string

     for .. in 只能遍历 对象   json 对象 --- 只能用 for .. in  来遍历

9、测试一个变量是不是一个数组（扩展）

typeof  只能判断 一个变量（数组） 是不是一个object  不能测试一个变量是不是一个数组

instanceof  用来测试一个变量是不是一个数组

     变量名  instanceof  Array

10、数组的操作

push()   向数组的尾部添加一个（或多个）新的元素   多个数据之间通过逗号连接

语法 ： 数组名.push(数据)

pop()   将数组末尾的元素删除  并返回删除的元素

语法 ： 数组名.pop()

shift() 将数组中的第一个元素删除  并返回删除的元素

语法 :  数组名.shift()

unshift() : 向数组的首部添加一个或多个新的元素    返回新数组的新长度值

语法: 数组名.unshift(数据)

slice() ：  从开始位置截取到结束位置处的数据，截取的数据不包含结束位置处的数据

语法：  数组名.slice(start,end)    (不改变原数组)

splice() ： 从开始位置 截取指定长度的字符 （改变原数组）

语法  ：　　数组名．splice(start,length)

join()  : 通过指定的字符将数组连接成一个字符串（将数组转成字符串）

语法 ：　　数组名．join(字符)

concat() ： 将一个或多个数组连接到一起，形成一个数组

语法：  数组名.concat(数组名)

reverse() ：数组的翻转

语法 ：  数组名.reverse()

11、二维数组

var arr = [[1,2],[34,2],[5,6],[7,9]];

arr[0][1]

var arr = new Array( new Array(1,2,3), new Array(4,5,6)  );

取值：

数组名[行下标][列下标]

12 、栈 和 堆

栈的定义：

     在计算中，当定义一个变量时，计算机就会开辟一块存储空间，这个用来存放变量的一个值 ，管这个空间就叫做栈

     栈的特点 ： 后进先出

     栈中存放的数据类型 ： 基本的数据类型

指针（扩展）：在计算机中，有一种特殊的变量，这个变量中存放的是数据的地址   ，这样的变量 就成为 指针。

堆 ：堆中可以存放多个数据

     堆中存放的数据类型数据一般是  ： object

     一改全改

13、选择排序

选择排序：每一轮比较时，会用数组中的一个基准值和后面的数依次做比较，如果基准值大于后面某个数，就交换

每一轮比较后得到最小值

15、数组排序方法

arr.sort(函数名);

function fun(a,b){

     return a - b;//升序     return  b - a ---降序

}

16、ES5   数组新增的方法

1、严格模式  ： 在程序的开头 或  函数体开头  加一个 “use strict”  表示是一个严格模式

2、数组新增方法

indexOf()

用法：

     arr.indexOf(num)   判断  num  在数组 arr中是否存在  如果存在，返回这个数num在arr中的下标，如果不存在 返回-1

forEach()  遍历

用法：

     arr.forEach(function( item,index,array ){

          item : 数组中的每一项

          index ：数组中每个数据下标

          array ：遍历的那个数组

     })

filter()   过滤   返回一个数组

用法：

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

     })

reduce()  归并

用法：

     arr.reduce(function(pre,next,item,array){    //  previous  上一个    next  下一个

          pre 上一个操作返回的结果

          next   下一个要操作的数据

     })

map()  映射    遍历数组 --  操作数组 ---返回数组

用法：

     arr.map(function(item,index,array){

     })

17、内置对象

Array   、  String   、  Math 、   Date  、   RegExp(正则)

18、String对象

字符串的定义：

      var str = "hello";

      var str = new String("hellooo");

获取字符串长度  ：  length

字符串的操作方法：

charAt    返回某个索引位置处的字符

用法：  str.charAt(index)

charCodeAt     返回某个索引位置处的字符对应的ASCII码值

用法：  str.charCodeAt(index)

String.fromCharCode  根据某个字符的 ASCII 值 找到对应的字符

用法：String.fromCharCode（ASCII值）

substr   截取从开始位置  指定长度的字符  如果没有指定长度值， 默认截取到字符串的结尾

用法: str.substr(start,length)

substring     截取从开始位置 到结束位置处的字符 （不包括结束位置）

用法：str.substring(start,end)

replace  字符串替换   默认替换一个   结果为替换后的新串

用法： str.replace(原字符，替换字符)

split  将一个字符串通过指定的字符分割成数组（将字符串转成数组）

用法： str.split(字符)

--------------------------------------

trim   去掉字符串两端的空白字符

用法:str.trim()

toLowerCase 将一个字符串转成小写字母

用法: str.toLowerCase()

toUpperCase  将一个字符串转成大写字母

用法：str.toUpperCase()

19、数学对象 Math

数学对象的方法 都是通过Math. 调用的    数学对象 Math  不需要定义

Math.ceil()    向上取整  Math.ceil(2.3) 3        Math.ceil(-2.3)  -2      大于这个参数的最小整数

Math.floor()   向下取整  Math.floor(2.3) 2       Math.floor(-2.3)   -3     小于这个参数的最大整数

Math.round()   四舍五入

Math.sqrt()    开方    Math.sqrt(16);

Math.pow(m,n)  计算m的n次方

Math.sin()     正弦

Math.cos()     余弦

Math.random()    获取随机数    [0,1)

Math.floor( Math.random()\*(max-min+1) + min )    获取任意区间值   max 表示区间的最大值  min表示区间的最小值

20、Date对象--日期时间对象

定义：

var now = new Date();   得到一个标准时间格式   系统时间

var self = new Date("2017-6-21");  定义一个自己的时间格式

21、获取日期时间方法

getFullYear()   得到年份

getMonth()    月份     实际月份  ( getMonth() + 1 )

getDate()   日期

getDay()    星期       周日 -- 0    周一--1

getHours()   小时

getMinutes()  分钟

getSeconds()   秒

22、日期格式化函数  dateToString

23、日期格式化函数   stringToDate

1、通过构造函数方式

     var str = "1998-10-09";

     var d = new Date(str);

2、通过构造函数   需要 通过Date.parse( str )  将字符串转成毫秒值

   Date.parse( 字符串  ) 从1970年1月1日零点  到 str时间的毫秒值

    var d = new Date( Date.parse(str) );

24、获取时间的毫秒值

var d = new Date();

d.getTime()   得到一个毫秒值   从1970年1月1日 午夜 到  当前时间的毫秒值

------------------

Date.parse("2017-09-08")  得到一个毫秒值

25、时间差方法

假设毕业时间是  2017年 9月 30号  ，计算还有 多少天毕业？

function diff(start,end){

     return Math.abs( start.getTime() - end.getTime() );

}

26、设置 n天后的时间  （时间设置）

通过毫秒值计算10天后的时间：

var now = new Date();

var curr = now.getTime() + 10\*1000\*3600\*24;   用当前时间的毫秒值 加上十天后的毫秒值

now.setTime( curr );   将当前时间通过 setTime方式  设置成 10天后的标准时间格式  参数 curr 就是10天后的毫秒值

27、定时器

var timer = setInterval()  有两个参数  --- 连续执行   timer 表示定时器名称

第一个参数：要执行的任务 --- 函数

第二个参数：任务间隔时间  （毫秒）

用法一：

setInterval(function(){ .... },1000)

用法二、

setInterval("fun()",1000);

定时器任务：

     function fun(){

     }

用法三、

setInterval( fun , 1000 );

定时器任务：

     function fun(){

     }

clearInterval( 定时器名称 ) ---- 停止连续执行的定时器

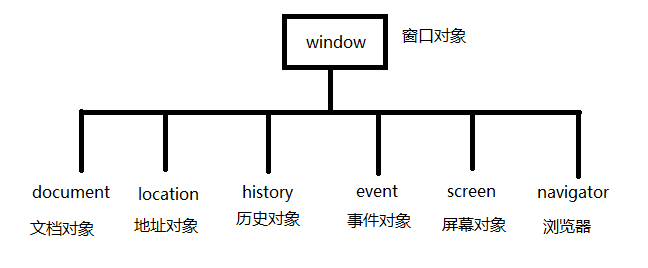
var timer = setTimeout();  参数和 用法 同setInterval()  ---  只执行一次 （一般用于延时）

clearTimeout()  ---  停止定时器

定时器特点：需要等待页面中所有程序都执行完毕后才执行定时器的内容

toLocaleString()  将日期时间对象转成 本地时间格式  ，字符串形式

10、BOM --- browser  object   model     浏览器对象模型

浏览器对象模型

document....navigator  叫做window对象的属性 或 window对象的内置对象

window.document.getElementById()

12、scroll  家族

scrollLeft  获取滚动条水平方向滚走的距离

scrollTop  （最火）  获取滚动条 垂直 方向滚走的距离

滚动条事件  获取页面向上滚走的距离

window.onscroll = function(){

     兼容火狐和谷歌浏览器的写法：

     获取页面滚走的距离：

     var sTop = document.body.scrollTop || document.documentElement.scrollTop;

}

设置页面滚走距离 ：

 document.body.scrollTop = document.documentElement.scrollTop = 值

BOM

window对象：窗口对象

说明：

     1、 window对象的属性和方法调用时可以省略 window.

     2、所有的window对象的属性和方法都属于全局的属性和全局方法

     3、全局变量和全局的方法（自定义函数）都属于window对象

window对象的属性 和 方法：

三个对话框：

     alert()  弹出一个对话框

     confirm()  弹出一个带有确定和取消按钮的对话框，点击确定返回true，点击取消返回false

     prompt()  输入框对话框   点击确定返回输入的内容，点击取消返回null

     这三个对话框都有阻塞浏览器其它功能的执行

两个定时器：

     setInterval()   clearInterval()

     setTimeout()    clearTimeout()

一个弹出窗口 open()

url--统一资源定位器

open( "弹出子窗口url","\_blank","弹出窗口的外观" );

第三个参数--外观：  width 宽   height  高    left  距离父窗口的左边距    top   距离父窗口的上边距

                  location地址栏、toolbar 工具条、scrollbars（滚动条）  等 这些属性一般不同的浏览器有不同的显示  （了解）  值 ：yes/1    no/0

open()方法的返回值  就是弹出的子窗口  。

close()  关闭窗口

focus()   获取焦点（让窗口处于最前端）

closed  判断窗口是否被关闭

opener   获取到父窗口

3、location  地址对象

页面跳转：

     location.href = "url"   设置路径      获取路径  ：  location.href

     location = "url"

     location.replace("url")   跳转后页面不可以实现后退效果    将当前页面内容覆盖了

     location.assign("url")

4、history  历史对象

前进 ：　history.go(1) /  history.forward()

后退 ：　history.go(-1)  / history.back()

刷新 ： history.go(0)

navigator浏览器对象

appName ： NetScape  /   浏览器全称

appVersion  浏览器版本

screen  屏幕对象

screen.width / availWidth

          .height / availHeight  后者获取的高度不包括 工具栏的高度

5、document 文档对象

要做事，先找人（页面元素）

write()  浏览器打印信息

----------------------

查找元素的方法：

getElementById()  通过给定的id查找元素   结果是唯一

getElementsByTagName() 根据给定的标签名查找元素  得到是一个集合 （伪数组）  即便找到一个元素，也要加一个下标  0

getElementsByName()    根据给定的name值找元素   （用于表单）

getElementsByClassName()  根据给定的类名 查找元素  （ie6不可用）

querySelector();     根据给定的选择器查找元素   结果唯一    参数是一个选择器

querySelectorAll() ;  得到是一个集合 （伪数组）  即便找到一个元素，也要加一个下标  0   参数是一个选择器

document.body

document.title

document.head

document.documentElement   (指的是html元素)

6、操作元素

操作元素属性（标签上的属性）

     获取元素的属性 （读）

          元素.属性名

     设置元素的属性  （写）

          元素.属性名 = 值

操作元素样式（style内的样式）

     获取元素样式：

          元素.style.样式名

     设置元素样式：

          元素.style.样式名 = 值

          元素.style.cssText = "值"

          元素.className = "class名"

操作元素内容

     普通标签 ：

          获取元素的内容：

               元素.innerHTML      得到元素内部所有内容（包括内部标签）

               元素.innerText  (火狐不兼容    元素.textContent)     得到纯文本数据

               元素.outerHTML      得到元素内部及其该元素本身的所有内容

          设置元素内容:

               元素.innerHTML = 值      识别网页标记

               元素.innerText = 值  (火狐不兼容    元素.textContent)    将值中的标签作为纯文本显示

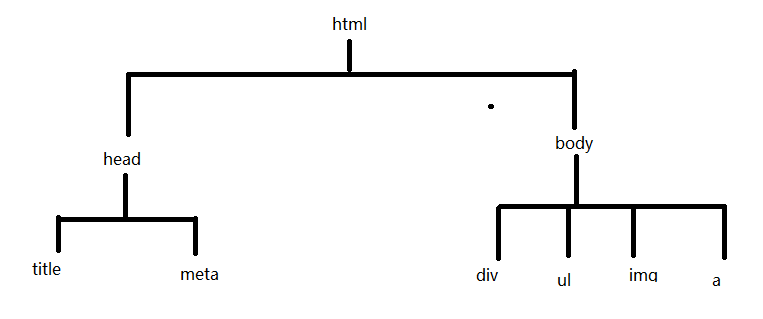
               元素.outerHTML = 值      改变元素本身

     表单标签:

           获取  :  元素.value

           设置  :  元素.value = "值"

2、DOM   document  object  model    -- 文档对象模型

dom模型

3、节点种类

html元素是dom的根节点

所有的html元素 都叫做  元素节点

所有的HTML元素的内容   文本节点

所有的html元素的属性   属性节点

一切都是节点

4、节点关系

父节点：parentNode

下一个兄弟节点：  nextElementSibling （高版本浏览器）               nextSibling（低版本浏览器）

前一个兄弟节点：  previousElementSibling

第一个孩子节点:   firstElementChild

最后一个孩子节点：  lastElementChild

所有孩子  ：  childNodes

             children

访问节点 ：  getElmentBy....

5、children 和 childNodes区别

children 得到所有的元素节点

childNodes  会得到元素节点和文本节点的集合 （会将页面中的回车作为文本节点）

通过 nodeType 值确定节点的类型：

值为  1   元素节点

值为  2   属性节点

值为  3   文本节点

nodeName  或 tagName  得到标签的名称