javascript笔记

defer属性告诉浏览器延迟执行脚本

属性async异步执行脚本立即下载脚本，但不能妨碍其他下载

NAN（not a number）当被除数不是数字是返回NAN，他属于数字，表示非数字值

infinity 当**计算数超出最大范围返回的数，属于数**

**typeof 返回数据类型的字符串值 可能的值：str**ing number b array object undefined function

search 搜索并返回首字母位置

正则表达式语法： /pattern 模式/modifiers修饰符

修饰符 i执行对大小写敏感的匹配 g全局匹配 m 多行匹配

表达式模式： [abc]查找方括号之间的任何字符 [0-9]查找0到9之间的任何数字 (x|y) 查找 | 两端的任何选项 括号用于查找一定范围的字符串

严格模式 在脚本开头添加 use strict

arguments对象与数组类似可用 [ ] 来访问函数的参数，且可以与命名参数一起使用

显式类型转换 Number转换为数

parseInt 转换为整型 两个参数 被解析的字符串和基底（参数字符串进制）

parseFloat 转换为浮点数 两个函数都是字符串开头解析直到非数字或解析完

String 转换为字符串 Boolearn转换为布尔类型

toString 转换为字符串 用点方法 demo.toString(radix) 转换为目标进制

隐式类型转换

isNaN() /++ /-- /- / % 调用Number +号两侧有一个字符串就把两个都转换成字符串拼接 调用String , && || == != 调用Boolearn

数组

引用类型即对象 对象的属性名会自动转换成字符串

数组属性length---不是只读的，可以通过修改他从数组的末尾移除或添加项

数组方法 toString 返回数组每个值以逗号分隔开形成的字符串

join 方法则可以定义分隔的符号

而pop 和 pusing 方法则可以让数组表现的像栈一样， pop用于删除最后一个元素，push 用于向数组结尾处添加元素 ， 而shift方法能移除数组中得第一个项并返回该项

reverse 方法会反转数组元素顺序 sort方法会对数组每一项转成字符串并比较然后从大到小排列由于字符串比较方法很蠢，所以需要一个比较函数

concat 方法会复制当前数组并且可以添加任一项然后返回一个新数组

slice 方法能够剪切数组 可以有两个参数分别为起始位置和删除多少个

splice 可以用来删除、插入和替换数组中的项， 他有3个参数 第一个起始位置 第二个项数 第三个参数为要插入或替换的项可以有多个

数组的迭代方法有五种 every filter forEach map some 都有两个参数，对于每一项要运行的参数，运行该函数的作用域对象（可选）

其中every要传入的函数对每一项都返回 true , some 传入的函数对于任一项返回true即可返回true，filter对每项运行函数返回会返回true的项，map返回每次函数调用结果组成的数组，forEach对每一项运行函数但不返回

数组的归并方法有 reduce 和 reduceRight reduce会从头遍历数组，reduceRight从末尾遍历数组，方法都拥有两个参数即每一项调用的函数和归并初始值，函数有四个参数 前一个值 当前值 项的索引 数组对象

Date 对象 Date.parse方法接收一个表示日期的字符串参数

Date.UTC方法也返回表示日期的毫秒数 参数依次 年，月，日，时，分，秒，毫秒 （只有前两个是必须的）

闭包

window.alert() 写入警示框 document.write() 写入HTML输出（该方法仅用于测试在代码加载完毕后会删除所有HTML元素）

innerHTML 写入HTML元素 console.log() 写入浏览器控制台

function 用于声明函数

预编译：发生在编译的前一刻

函数声明会整体提升到程序头部，变量声明只会把声明提升到头部

暗示全局变量：任何变量未经声明即归全局对象所有，即是window的属性

1、创建AO(Activation object) GO对象

2、找形参和变量声明作为AO对象的属性名，值为undefined

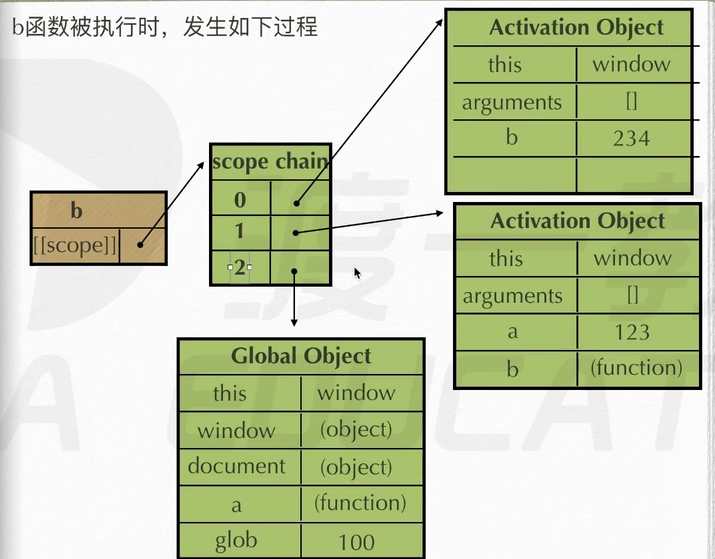
3、将实参和形参统一

4、在函数体内找函数声明，值赋予函数体

后面执行的会覆盖前面的值，如变量、参数和函数名相同的话

函数创建时会创建 [[scope]] 对象，函数定义时便产生GO对象在第0位，之后再执行时产生AO对象在第0位，GO就到第1位以压栈的形式存入形成作用域链，[[scope]] 就是作用域，里面存储的就是执行期上下文对象的集合，也叫作用域链，

查找变量从作用域链顶端依次向下



二级函数b被执行时产生自己的AO到第0位形成作用域链

闭包：内部函数被保存到外部就会生成闭包，就可以在外部访问内部变量，因为被保存到外部的函数的作用于链该是在内部时的作用域链，闭包指的是有权访问父作用域的函数，即使在父函数关闭之后。

立即执行函数: 针对初始化作用的函数只执行一次，

var num = (function abc (a,b,c) {

这个函数被执行完就立即释放销毁

var d = a + b + c;

} (1,2,3) )

函数的执行符号为 ( ) 能被执行符号直接 执行的函数为立即执行函数

对象

创建方法 : 1、对象直接量 var obj = {} 直接定义

2、构造函数 系统自带函数 new object()

自定义函数名字 用new关键字构造新的对象

命名最好按照大驼峰方式，父对象 子对象？？

可以通过参数传递属性变量 构造函数原理 1、在函数体前隐式的加 this = {}

2、执行 this.xxx = xxx; 3、隐式的 return this

对象添加属性方法直接赋值即可，在方法里的this指拥有这个方法的对象

删除对象属性用 delate操作符

原始值不能操纵属性，如数值、字符串、布尔值，要这些对象才行

原型( prototype)是function对象的一个属性，是构造函数构造的对象的公共祖先 他是一个对象，构造的对象可以继承原型对象的属性和方法，如果function上有同名属性原型上的就会被覆盖

把固定的属性和方法放入原型里可以减小代码的耦合，

原型的修改与删除不能通过构造的对象修改，那样只会在构造的对象内创建新属性，只有通过原型属性才能做到

\_proto\_ 系统隐式定义的属性。里面存对象的原型用于构造的对象链接原型，可以更改\_proto\_的指向为其他对象

原型链的访问顺序跟作用域链一样访问近的，一级一级访问，object.prototype是原型链的终端

creat : var obj = object.creat(原型); 创建对象但要指定原型

call方法 函数执行时 test.call(obj) 能改变函数的this指向

apply跟call差不多 不过传参数用数组形式

共享原型：直接把复函数的原型赋给子函数形成公有原型，可以写成inherit函数包装传参（函数）进行使用。但这样会导致两个函数原型指向同一空间，是的子函数没有独立性，

圣杯模式：用一个中间构造函数传递原型这样就不会两个函数共同指向了一个位置了

function inherit(Target, Origin){

function F(){};

F.prototype = Origin.prototype;

Target.prototype = new F();

Target.prototype.constructor = Target; //指明构造函数

Target.prototype.uber = Origin.prototype; //指明超类，最终继承的信息

}

var inherit = (function(){

var F = function(){};

return function (Target, Origin){

F.prototype = Origin.prototype;

Target.prototype = new F();

Target.prototype.constructor = Target; //指明构造函数

Target.prototype.uber = Origin.prototype; //指明超类，最终继承的信息

}

}());

可以利用闭包实现变量的私有化，防止污染全局变量（可以重名而不发生冲突）

枚举 ：for …. in 循环用来遍历对象

hasOwnProperty( prop ) 验证对象的属性是不是自己的属性

in 操作符用于判断对象能不能访问这个属性 例： ‘name’ in obj

instanceof 用于判断A对象是不是B构造函数构造出来的对象 A instanceof B

他是看A的原型链上 有没有 B的原型

this指向：

1、函数预编译过程中 this 🡪 window

2、全局作用域里 this-🡪 window

3、call/apply 可以改变this指向

4、obj.func(); func() 里面的this指向obj

5、事件中指接收事件的元素

属性 arguments.callee 指向函数自身引用

属性 caller 指向函数被调用的环境

数字和字符串相加会返回字符串getElementById() 返回一个元素对象 getElementsTagName() 返回的是一个数组

属性： innerHTML 元素的文本值

parentNode 元素的父节点

childNodes 元素的子节点

attributes 元素的属性节点

nodename 规定节点名称

nodevalue 规定与查询节点值

nodeType 查询节点类型 元素1 属性2 文本3

fistchild 第一个子元素

lastchild 最后一个子元素

className 用于为元素设置class属性

alert( ); 通过提示框显示变量值

创建数组 var man=arry(); 赋值 man=[“john”, 1940, false ]

创建对象 var man=object(); 赋值

var man={name:”john”, year:”1940”, living:”false”}

函数 founction name () { }

DOM 文档对象模型

查询属性值 getAttribute(“属性名”)

修改属性值 setAttribute(“属性值” ”修改后的属性值”) 只用于元素节点

事件处理函数：在特定事件发生时调用特定JavaScript代码

onclick 用于按钮、链接被点击是调用函数

onload 用于网页加载时调用函数

在a标签里的onclick事件里添加 return false; 就可以让原本的链接不被触发

完成在一个网页内切换图片 需要定义函数 提取图片链接的href属性和原本网页显示图片位置的src属性 用setAttribute 改变src属性值为点击链接的href属性值 并在html的a属性里用事件处理函数onclick=”show(this); return false;”

用if语句检测方法浏览器支不支持方法

if(!document.getElementById) return false;

//当网页被加载完毕时运行当前函数，这样可以做到让事件从文档中分离,当有多个函数需要这样做时，可以将他们都放到一个匿名函数里

window.onload =function(){

window.onload = prepareLinks();

window.onload = prepackages();

}

另一种方法

//也可以使用函数addLoadEvent譬如这样把想要页面加载好后就运行的函数放入列表就行

function addLoadEvent(func){

var oldonload = window.onload;

if (typeof window.onload != 'function'){

window.onload = func;

}

else {

window.onload = function(){

oldonload();

func();

}

}

}

addLoadEvent(prepareLinks);

addLoadEvent(prepackages);

//link[i].onkeydown = link[i].onclick; 把onclick事件的功能传给键盘处理事件onkeydownd 但最好不要使用它

jQuery:

可以使用多个方法进行连缀写法

可以用函数 $(‘ ’) 来使用css的选择器更加便捷的选择元素

$函数其实就是对专有的jQuery对象的简写别名

如 $(‘p .classname’) 可用于选择p元素后代中类名是classname的元素

css选择器大多都能用同时还有一些独有的选择器

$(‘tag:even’) 选择匹配元素集中偶数序号的的元素

$(‘tag:odd’) 奇数元素

$(‘tag:parent’) 选择匹配元素中包含子元素的元素，文本也算

还有一些专门用于访问表单的 :input :text ：password :radio 等等

一些JQuery事件：

jQuery方法：

hide()隐藏 show()显示 括号里填写数字为动作速度单位：毫秒

toggle() 用于切换hide()和show() 两种方法

fadeIn() 用于淡入已隐藏的元素

fadeOut() 方法用于淡出可见元素。

fadeToggle() 方法用于切换fadeIn()和fadeOut()两种方法

fadeTo() 方法允许渐变为给定的不透明度（值介于 0 与 1 之间） 效果时长和不透明度两个参数必填

slideDown() 方法用于向下滑动元素。展开隐藏

slideUp() 方法用于向上滑动元素，收起隐藏

slidToggle() 方法用于切换 slidDown()和slidUp() 两种方法

animate({params},speed,callback) 方法用于创建自定义动画 params为形成动画的css属性必填

stop()可以用于停止动画效果

callback参数在一个动画或效果完成后执行的函数