JS笔记

JavaScript组成

各组成部分的兼容性，兼容性问题由来

ECMAScript：翻译 核心 解释器

DOM Document Object Model HTML document

操作HTML的能力

BOM Browser Object Model 浏览器 window

ECMA 几乎没有兼容性问题

DOM 有一些操作不兼容

BOM 没有兼容问题（完全不兼容）

命名规范

可读性

（给变量命名需要前缀，给函数命名不需要前缀）匈牙利命名法：类型前缀，首字母大写

+ 1.字符串连接 2.数字相加

- 1.数字相减（隐式类型转换）

索引值的使用

什么时候使用索引值，需要知道第几个的时候

Html中添加index，会被过滤，js中添加不会过滤。

Alert(‘abc’+(12+5)+’def’);输出abc17def

Alert(‘abc’+12+5+’def’);输出abc125def

网页换肤，两套css链接<link>中的href link可以加id

function skin1(){

var ol = document.getElementById("");

ol.href = 'css1.css';

}

function skin2(){

var ol = document.getElementById("");

ol.href = 'css2.css';

}

再调用函数

任何标签都可以加id

任何标签的任何属性js怎么写

1. document.getElementById在任何浏览器下均可使用
2. get Element By Id

获取 元素 用 ID 通过ID获取元素

1. function 函数名()

{

代码

}

函数名()

4.当点击的时候

如果div是显示的，隐藏掉

或者，显示出来

如果div是显示的，oDiv.style.display='none'

或者，oDiv.style.display='block'

if(div是显示的)

{

oDiv.style.display='none';

}

else

{

oDiv.style.display='block';

}

1. = 赋值（改变） == 判断
2. 函数传参：参数就是占位符

什么时候用传参——函数里定不下来的东西

1. style与className

元素.style.属性=xxx 是修改行间样式

之后再修改className不会有效果

1. 获取一组元素

getElementsByTagName

1. 用for循环为一组元素添加事件

什么时候用循环——一组元素

例子：

全选——checked属性

反选——for循环配合if判断

Break整个循环中断

Continue 本次循环中断，继续下次循环

真:true 非零数字 非空字符串 非空对象

假:false 数字零 空字符串 空对象 undefined

Json没有length

Arr有length属性，数组循环一般用length循环

Json循环

1. 选项卡

按钮的实现

添加事件

this的使用

先清空所有按钮，再选中当前按钮

内容的实现(div)

先隐藏所有Div，再显示”当前”Div

索引值的使用

什么时候用索引值——需要知道“第几个”的时候

html添加index——会被FF过滤

js添加index

11.字面量（常量）：看到东西，自然之道是什么

12.样式优先级：

\*<标签<class<ID<行间

13.function 名字()

{

...

}

oBtn.onclick=名字

oBtn.onclick=function ()

{

....

};

1. window.onload 页面加载完成的时候发生
2. 行为、样式、结构三者分离

JS CSS HTML

16.类型：typeof运算符

用法、返回值

常见类型：

number、string、boolean、undefined、object、function

一个变量应该只存放一种类型的数据

17.字符串转数字parseInt()：输出为整数

parseFloat();可输出为小数，也可以是小数

从左到右碰到不是数字，则跳出。

NaN: Not a Number非数字

任何数和NaN相加都为NaN

NaN和NaN不相等

isNaN();判断是否是NaN

18.== 双等号，先转换类型，然后比较

=== 三等号，不转换类型，直接比较

19.什么是闭包

子函数可以使用父函数中的局部变量

20.算术：+ 加、- 减、\* 乘、/ 除、% 取模

实例：隔行变色、秒转时间

赋值：=、+=、-=、\*=、/=、%=

关系：<、>、<=、>=、==、===、!=、!==

逻辑：&& 与、|| 或、! 否

实例：全选与反选

运算符优先级：括号

21.一个函数应该只返回一种类型的值

22.可变参（不定参）：arguments

参数的个数可变，参数数组

例子1：求和 求所有参数的和

例子2：CSS函数

判断arguments.length

给参数取名，增强可读性

取非行间样式(不能用来设置)：

obj.currentStyle[attr]获取非行间样式 currentStyle 只兼容IE或者高的谷歌

getComputedStyle(obj, false)[attr]可兼容获取元素，第二个参数随意值（false）（IE9之前不可用）

例：css(oDiv, 'width') 获取样式

css(oDiv, 'width', '200px') 设置样式

复合样式：background、border

单一样式：width、height、position

23.数组的使用

定义：

var arr=[12, 5, 8, 9];

var arr=new Array(12, 5, 8, 9);

没有任何差别，[]的性能略高，因为代码短

数组的属性：

length

既可以获取，又可以设置

例子：快速清空数组

**数组使用原则：数组中应该只存一种类型的变量**

24.数组的方法

添加

push(元素)，从尾部添加

unshift(元素)，从头部添加

删除

pop()，从尾部弹出

shift()，从头部弹出

25.排序：

sort([比较函数])，排序一个数组

排序一个字符串数组

排序一个数字数组

转换类：

concat(数组2)

连接两个数组

join(分隔符)

用分隔符，组合数组元素，生成字符串

字符串split

splice

splice(开始, 长度,元素…)

先删除，后插入

删除

splice(开始,长度)

插入

splice(开始, 0, 元素…)

替换

25.开启定时器

setInterval 间隔型

setTimeout 延时型

两种定时器的区别

停止定时器

clearInterval

clearTimeout

1. 数码时钟

获取系统时间

Date对象

getHours、getMinutes、getSeconds

显示系统时间

年getFullYear()

月getMonth()

日getDate()

星期getDay()

字符串连接

空位补零

设置图片路径

charAt方法

27.延时提示框

原来的方法

移入显示，移出隐藏

移出延时隐藏

移入下面的Div后，还是隐藏了

简化代码

合并两个相同的mouseover和mouseout

28.无缝滚动——基础

物体运动基础

让Div移动起来

offsetLeft的作用offsetLeft获取左边距，综合考虑能影响元素所有左边距属性。

用定时器让物体连续移动

效果原理

让ul一直向左移动

复制li

innerHTML和+=

修改ul的width

滚动过界后，重设位置

判断过界

改变滚动方向

修改speed

修改判断条件

鼠标移入暂停

移入关闭定时器

移出重新开启定时器

29.运动基础

让Div运动起来

速度——物体运动的快慢

运动中的Bug

不会停止

速度取某些值会无法停止

到达位置后再点击还会运动

重复点击速度加快

匀速运动

速度不变

30.运动框架

在开始运动时，关闭已有定时器

把运动和停止隔开(if/else)

运动框架实例

例子1：“分享到”侧边栏

通过目标点，计算速度值

例子2：淡入淡出的图片

用变量存储透明度

30.缓冲运动

逐渐变慢，最后停止

距离越远速度越大

速度由距离决定

速度=(目标值-当前值)/缩放系数

例子：缓冲菜单

Bug：速度取整

跟随页面滚动的缓冲侧边栏

潜在问题：目标值不是整数时

31.运动终止条件

匀速运动：距离足够近

缓冲运动：两点重合

32.多个物体同时运动

例子：多个Div，鼠标移入变宽

单定时器，存在问题

每个Div一个定时器

多物体运动框架

定时器作为物体的属性

参数的传递：物体、目标值

例子：多个Div淡入淡出

所有东西都不能公用

属性与运动对象绑定：速度、其他属性值（如透明度等）

33.offset属性的Bug

有边框的Div变宽

用currentStyle代替offset

原有运动框架的问题

只能让某个值运动起来

如果想让其他值运动起来，要修改程序

扩展的运动框架

运动属性作为参数

封装opacity

小数的问题

34.效果思路

两边的按钮——淡入淡出

大图下拉——层级、高度变化

下方的li——多物体淡入淡出

下方的Ul——位置计算

左右按钮

淡入淡出

鼠标移到按钮上，按钮会消失

层级问题

按钮和遮罩上都得加上事件

下方Li效果

点击切换大图——选项卡

Li淡入淡出——移入移出

Ul移动——位置计算

大图片切换

图片层级——zIndex一直加1

图片下拉效果(运动框架)

可以改为淡入淡出

加入自动播放

和选项卡一样

35.什么是event对象

用来获取事件的详细信息：鼠标位置、键盘按键

例子：获取鼠标位置：clientX

document的本质：document.childNodes[0].tagName

获取event对象(兼容性写法)

var oEvent=ev||event;

事件流

事件冒泡

取消冒泡：oEvent.cancelBubble=true

例子：仿select控件

DOM事件流

1. 鼠标位置

用clientX,Y一定要用scrollTop

clientY 鼠标可视区位置

可视区位置：clientX、clientY

例子1：跟随鼠标的Div

消除滚动条的影响

滚动条的意义——可视区与页面顶部的距离

获取鼠标在页面的绝对位置

封装函数

例子2：一串跟随鼠标的Div

37.keyCode

获取用户按下键盘的哪个按键

例子：键盘控制Div移动

其他属性

ctrlKey、shiftKey、altKey

例子：提交留言

回车 提交

ctrl+回车 提交

38.阻止默认行为

普通写法：return false;

例子1. 屏蔽右键菜单

弹出自定义右键菜单

例子2. 只能输入数字的输入框

keydown、keyup事件的区别

39.简易拖拽

拖拽原理

距离不变

三个事件

靠谱拖拽

原有拖拽的问题

直接给document加事件

FF下，空Div拖拽Bug

阻止默认事件

防止拖出页面

修正位置

1. 事件绑定

setCapture 不会选中文字，事件捕获只有IE兼容，整个网页的所有事件都会聚集在一个div上

releaseCapture();取消事件捕获

oDiv.setCapture();集中在oDiv上。

Document改为oDiv（IE下）

IE方式

attachEvent(事件名称, 函数)，绑定事件处理函数

detachEvent(事件名称, 函数)，解除绑定

DOM方式

addEventListener(事件名称,函数, 捕获)

removeEventListener(事件名称, 函数, 捕获)

为什么给document加事件，因为div太小鼠标就跑了，div被甩掉。

何时使用事件绑定

绑定事件和this

绑定匿名函数，会无法删除

41.复习拖拽原理

距离不变

三个事件：down、move、up

限制范围

对位置进行判断

例子1：不能拖出窗口的Div

例子2：不能拖出指定对象的Div

磁性吸附

42.图片拖拽

阻止默认事件

文字选中

阻止默认事件

IE下拖动有问题

事件捕获

与DOM配合

带框的拖拽

保留原有位置的拖拽

43.拖拽

只有横向拖拽

限制范围——范围的大小

计算比例——当前值/最大值

控制其他对象

例子1：控制物体的大小

例子2：控制物体的透明度

例子3：控制文字滚动

44.什么是对象

什么是收音机

对象是一个整体，对外提供一些操作

什么是面向对象

使用对象时，只关注对象提供的功能，不关注其内部细节

比如JQuery

面向对象（）

不了解原理的情况下，使用它内部的功能。（使用对象时，只关注对象提供的功能，不关注其内部细节。比如：jQuery）

对象：

黑盒子，不了解内部结构，知道表面的各种操作（按钮）duixian

对象是一个整体，对外提供一些操作。

面向对象是一种通用思想，并非只有编程中能用，任何事情都可以用

1. js面向对象

对象：属性+方法

继承：父级的属性+方法

面向对象编程(OOP)的特点

抽象：抓住核心问题

封装：不考虑内部实现，只考虑功能使用

继承：从已有对象上，继承出新的对象

多重继承

多态

对象的组成

方法——函数：过程、动态的

属性——变量：状态、静态的

变量能做的事情，属性也能做。

变量：自由的，不属于任何人

属性：属于一个对象的

函数：自由

方法：属于对象

46.为对象添加方法和属性

this详解，事件处理中this的本质

window

this——函数属于谁

This:当前发生事件的对象（之前）

当前的方法属于谁，this就是谁

不能在系统对象中随意附加方法、属性，否则会覆盖已有方法、属性

object对象

47.什么是工厂

原料 加工 出厂

工厂方式

用构造函数创建一个类

什么是类、对象（实例）：模具和零件

48.问题

没有new

函数重复定义

加上new

偷偷做了两件事

替你创建了一个空白对象

替你返回了这个对象

new和this

49.什么是原型prototype

原型是class，修改他可以影响一类元素

在已有对象中加入自己的属性、方法

原型修改对已有对象的影响

为Array添加sum方法

给对象添加方法，类似于行间样式

给原型添加方法，类似于class

原型的小缺陷

无法限制覆盖

50.流行的面向对象编写方式

用混合方式构造对象

混合的的构造函数/原型方式

Mixed Constructor Function/Prototype Method

原则

构造函数：加属性

原型：加方法

对象命名规范

类名首字母大写

51实例：面向对象的选项卡.

把面向过程的程序，改写成面向对象的形式

原则：

不能有函数套函数、但可以有全局变量

过程

onload → 构造函数

全局变量 → 属性

函数 → 方法

改错

this、事件、闭包、传参

对象与闭包

通过闭包传递this

52.Json方式的面向对象

把方法包在一个Json里

有人管他叫——命名空间

在公司里，把同一类方法，包在一起

53.面向对象的拖拽

改写原有拖拽

对象的继承

什么是继承

在原有类的基础上，略作修改，得到一个新的类

不影响原有类的功能

instanceof运算符

查看对象是否是某个类的实例

53.限制范围的拖拽类

构造函数伪装

属性的继承

原理：欺骗构造函数

call的使用

原型链

方法的继承

原理：复制方法

覆盖原型和方法复制

54.本地对象（非静态对象）

什么是本地对象

常用对象

Object、Function、Array、String、Boolean、Number、Date、RegExp、Error

内置对象（静态对象）

什么是本地对象

Global、Math

宿主对象（由浏览器提供的对象）

DOM、BOM

宿主：运行环境（浏览器）

本地对象+内置对象（JS语言本身具备的，跟环境无关）

静态对象：不能new没直接用的 代表 Math

非静态对象：必须new,实例化才能用的对象

1. DOM基础

DOM document 操作浏览器上的元素

childNodes（子节点）算空的文本节点 在IE6—8不会出问题。

Children 只取第一层

Asdasd文本节点

<span>.fasdf元素节点</span>

nodeType==3文本节点

nodeType==1元素节点

什么是DOM

浏览器支持情况

DOM节点

childNodes nodeType

获取子节点

children

parentNode

例子：点击链接，隐藏整个li

offsetParent获取有定位的父级，根据样式会改变

例子：获取元素在页面上的实际位置

56.DOM节点(2)

首尾子节点

有兼容性问题

firstChild、firstElementChild

lastChild 、lastElementChild

兄弟节点

有兼容性问题

nextSibling、nextElementSibling

previousSibling、previousElementSibling

57.元素属性操作

第一种：oDiv.style.display=“block”;

第二种：oDiv.style[“display”]=“block”;

第三种：Dom方式

DOM方式操作元素属性

获取：getAttribute(名称)

设置：setAttribute(名称, 值)

删除：removeAttribute(名称)

58.用className选择元素

如何用className选择元素

选出所有元素

通过className条件筛选

封装成函数

59.创建DOM元素

createElement(标签名) 创建一个节点

appendChild(节点) 追加一个节点

例子：为ul插入li

插入元素

insertBefore(节点, 原有节点) 在已有元素前插入

父级.insertBefore（子节点，哪个位置前）

例子：倒序插入li

删除DOM元素

removeChild(节点) 删除一个节点

例子：删除li

60.文档碎片

文档碎片可以提高DOM操作性能(理论上)

文档碎片原理

document.createDocumentFragment() 创建文档碎片

文档碎片理论上可以提高效率（IE6-7才会有些效果，高级浏览器不会有很多影响，基本没有用了，甚至可能会降低性能）

1. 表格应用

Tbody浏览器会自动加上，最好主动写上。

tBodies获取tbody

Rows行

Cells取行中单元格

只能有一个标头，可以有很多tbody

获取

tBodies、tHead、tFoot、rows、cells

隔行变色

鼠标移入高亮

添加、删除一行

DOM方法的使用

62.搜索

版本1：基础版本——字符串比较

版本2：忽略大小写——大小写转换

版本3：模糊搜索——search的使用

版本4：多关键词——split

高亮显示、筛选

排序

移动Li

元素排序：转换——排序——插入

63.表单基础知识

什么是表单

向服务器提交数据，比如：用户注册

action 提交到哪里

表单事件

onsubmit 提交时发生

onreset 重置时发生

64.表单内容验证

阻止用户输入非法字符 阻止事件

输入时、失去焦点时验证 onkeyup、onblur

提交时检查 onsubmit

\*后台数据检查

1. BOM

Document.write();先清空，再写

打开、关闭窗口

open

蓝色理想运行代码功能

close

关闭时提示问题

常用属性

window.navigator.userAgent判断当前浏览器类型，操作系统类型

Window.location 当前网页地址

窗口尺寸、工作区尺寸

srocllTop可视区距离顶部的距离

可视区尺寸

document.documentElement.clientWidth

document.documentElement.clientHeight

滚动距离

document.body.scrollTop

document.documentElement.scrollTop

66.系统对话框

警告框：alert(“内容”)，没有返回值

选择框：confirm(“提问的内容”)，返回boolean

输入框：prompt()，返回字符串或null

window对象常用事件

onload

onscroll

Onresize

Window.onresize当窗口改变大小是发生

userAgent>IE6 fixed

IE6 运动

例子：回到顶部按钮、侧边栏广告

闪烁问题

Ajax

让js可以读取或者发送服务器上的数据（不用刷新页面）

什么是服务器

PC，手机也能作为服务器

Locallhost 本地主机

自己配一个服务器，把文件存在www下

Function alax(url,fnSucc,fnFild){}

第一个参数，从哪去取

第二个参数，读成功，回来的时候，会主动访问自己写的一个函数

不成功返回fnFile

放在服务器下的文件，不能用中文命名

统一存储编码都是UTF-8或者其他。

缓存

根据url来缓存的

域名后的?后 get数据

全局变量是window属性

Js中没有定义的的变量--报错

没有定义的属性--不报错

Open(方法,文件名,异步传输)

1.Get？Post? 2.读哪个文件 3.

同步 多件事一起 js中事件一件一件来

异步 一件一件来 js中多个事情一起来（Ajax）

如果同步的话，网速很慢，其他标签什么的都不会显

onreadystatechange和服务器交互的时候发生的事件

什么是服务器

网页浏览过程分析

如何配置自己的服务器程序（AMP）

什么是Ajax

无刷新数据读取

用户注册、在线聊天室

异步、同步

68.使用Ajax

基础：请求并显示静态TXT文件

字符集编码

缓存、阻止缓存

动态数据：请求JS（或json）文件

eval的使用Eval()；解析成js

DOM创建元素

局部刷新：请求并显示部分网页文件

69.HTTP请求方法

GET——用于获取数据（如：浏览帖子）

POST——用于上传数据（如：用户注册）

GET、POST的区别

get是在url里传数据：安全性、容量

缓存

（网址长度有限）

Get:通过网址

容量小

安全性非常差

有缓存

用于获取数据

Post:不通过网址，http

容量大（2G）

安全性好一点

没有缓存

向服务器传递信息

Ajax原理(id 给前台用，name给后台用)

Get请求

Get方式：放入url

1. 编写Ajax

创建Ajax

ActiveXobject();针对IE6兼容

XMLHttpRequest();针对非IE6

创建Ajax对象

ActiveXObject("Microsoft.XMLHTTP")

XMLHttpRequest()

连接服务器

open(方法, 文件名, 异步传输)

同步和异步

发送请求

send()

71.请求状态监控

onreadystatechange事件

readyState属性：请求状态

0 （未初始化）还没有调用open()方法

1 （载入）已调用send()方法，正在发送请求

2 （载入完成）send()方法完成，已收到全部响应内容

3 （解析）正在解析响应内容

4 （完成）响应内容解析完成，可以在客户端调用了

status属性：请求结果

responseText

字符集

所有文件字符集相同

72.什么是cookie

页面用来保存信息

比如：自动登录、记住用户名

cookie的特性

同一个网站中所有页面共享一套cookie

数量、大小有限

过期时间

JS中使用cookie

document.cookie

73.cookie的使用

设置cookie

格式：名字=值

不会覆盖

过期时间：expires=时间

日期对象的使用

封装函数

读取cookie

字符串分割

删除cookie

已经过期

74.cookie的使用

例子

用cookie记录上次登录的用户名

提交时——记录用户名

window.onload——读取用户名

1. 正则表达式

复习字符串操作

search 查找

substring 获取子字符串

charAt 获取某个字符

split 分割字符串，获得数组

找出字符串中的所有数字

用传统字符串操作完成

用正则表达式完成

76.什么是正则表达式

什么叫“正则”

规则、模式

强大的字符串匹配工具

是一种正常人类很难读懂的文字

RegExp对象

JS风格——new RegExp(“a”, “i”)

第二个参数i，忽略大小写,ignore

perl风格——/a/i

77.字符串与正则配合 - 1

search

字符串搜索

返回出现的位置

忽略大小写：i——ignore

判断浏览器类型

match

获取匹配的项目 把所有匹配的东西都找出来

量词：+

量词变化：\d、\d\d和\d+

全局匹配：g——global找到全部，全局匹配

例子：找出所有数字

replace

替换所有匹配

返回替换后的字符串

例子：敏感词过滤

任意字符

[abc]

例子：o[usb]t——obt、ost、out

范围

[a-z]、[0-9]

例子：id[0-9]——id0、id5

排除

[^a]

例子：o[^0-9]t——oat、o?t、o t

78.组合

[a-z0-9A-Z]

实例：偷小说

过滤HTML标签

自定义innerText方法

转义字符

.（点）——任意字符

\d、\w、\s

\D、\W、\S

79.什么是量词

出现的次数

{n,m}，至少出现n次，最多m次

例子：查找QQ号

常用量词

{n,} 至少n次

\* 任意次 {0,}

？ 零次或一次 {0,1}

+ 一次或任意次{1,}

{n} 正好n次

80.表单校验实例

校验邮箱

行首行尾

81.Text 字符串一部分符合要求，就返回true

^行首 $行尾

/d == [0-9]

<.+> 任意个任意字符

82.

手机号：[1-9]\d{7}

Qq号：[1-9]\d{4-10}

固定号码:(0\d{2,3}-)?[1-9]\d{7}

83.过滤html标签

Var re = /<[^<>]+>/g; 除了两个<>以外的所有字符

\d 相当于[0-9]

[^a-z0-9]匹配除了英文和数字以外的

Math.round四舍五入