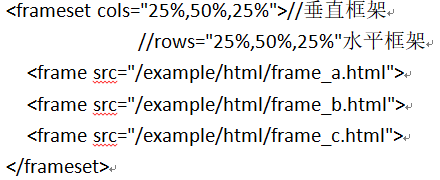
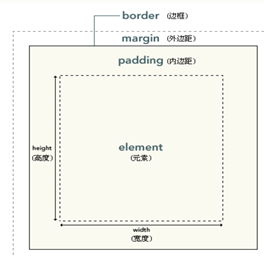
**1.框架集frameset**



**2.CSS规则主要由2个部分组成：选择器以及一条或多条声明**

**3.CSS盒模型**



outline轮廓 在边框外面 作用是突出元素

box-sizing:border-box/content-box; 标准是content-box ie6默认是border-box

border-box 元素的宽度=内容宽度+padding+border

content-box 元素宽度=内容宽度

**4.viewport**

视口 用户的可视区域

<meta name="viewport" content="width=device-width, initial-scale=1.0">

可设置的属性：

width/height

initial-scale/maximun-scale/minimum-scale

use-scalable 用户是否可以手动缩放

**5.网格布局Grid**

**6.媒体查询**

@media only screen and (max-width:500px){} 如果浏览器窗口小于500

@media only screen and (orientation:landscape){} 横屏landscape 竖屏portrait

**7.zoom**

zoom这个属性是ie专有属性，除了设置或者检索对象的缩放比例之外，它还有可以触发ie的haslayout属性，清除浮动，清除margin重叠等作用。 不过值得注意的一点就是火狐浏览器不支持zoom属性，但是在webkit内核浏览器中zoom这个属性也是可以被支持的

**8.图片后面有空白**

设置图片vertical-align:bottom 默认是基于基线对齐的

对图片使用float

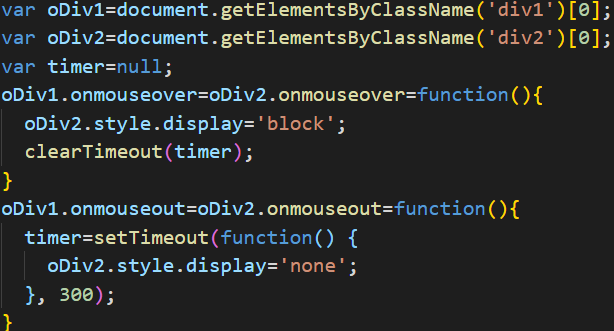
设置图片display:block

设置父元素font-size:0;

设置父元素line-hright:0;

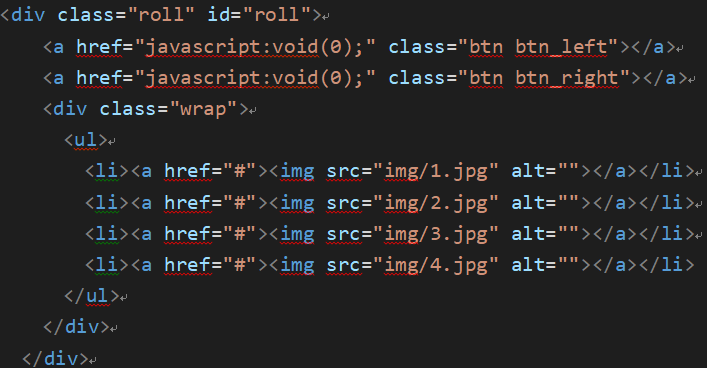
**9.延时提示框**

鼠标移入时清除定时器，鼠标移出设置定时器



**10.轮播图**

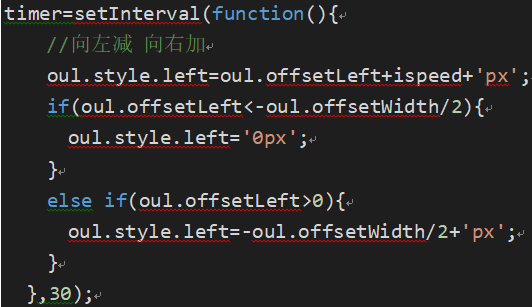
布局：div包裹ul div宽度固定且overflow:hidden，滚动是ul的位置在改变且ul绝对定位



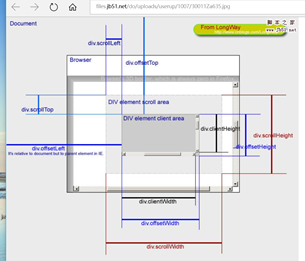
li的内容应该复制一倍，ul的长度在js中计算

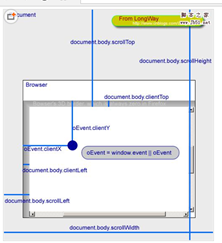
clipboard.png

向左减，如果offsetLeft小于-ul的宽度/2则将ul的left设置为0，向右加，如果ul的offsetLeft>0则将left设置为-ul的宽度/2



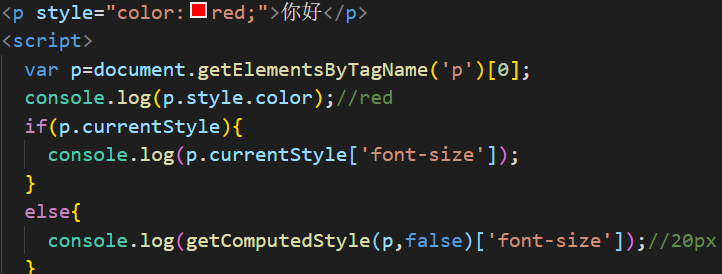
**11.元素大小和位置**





**12.获取元素样式**

obj.style只能获取行间样式



**13.input的内容从右到左**

clipboard.png

第一个光标在最右边，第二个光标在输入的字符的左边



**14.DOM节点**

nodeType 节点类型 元素1 属性2 文本3

childNodes子节点 包含空的文本节点 孙子节点不在子节点范围内

children 子节点 只包含元素节点 孙子节点不在子节点范围内

parentNode 父节点

offsetParent position是absolute的元素离最近的是relative的父元素的距离

firstChild 第一个子节点 包含空白子节点

firstElementChild 第一个元素子节点 ie678不支持 可以用children和nodeType结合

lastChild/lastElementChild 尾子节点

nextSibling 只有ie678支持 nextElementSibling 其他支持

previousSibling 只有ie678支持 previousElementSibling其他浏览器支持

firstChild lastChild nextSibling previousSibling 都会将空白或者换行当作文本节点

创建节点：

document.createElement('tagname') 返回一个element对象

创建文本节点

document.createTextNode('data') 返回文本节点

插入：

父节点.appendChild(newElement) 插入在末尾

父节点.insertBefore(newElement,someChild)在某子节点前插入

删除节点：

父节点.removeChild(element)

document.body.removeChild(element)

**15.元素属性操作**

获取/设置行间样式：

oDiv.style.display='block' oDiv.style['display']='block'

var bar=oDiv.style.display

获取非行间样式



获取和设置、删除非行间样式：

getAttribute(属性名)

setAttribute(属性名，属性值)

removeAttribute(属性名)

**16.BOM**

打开网页

window.open(url,[打开窗口]); 第二个参数：\_blank在空白标签页中打开，\_self在当前标签页中打开 返回值是新打开的页面

userAgent

window.navigator.userAgent 当前浏览器的版本

window.location 当前页面的路径

document的本质是最外面的一层标签 document.childNodes[0].tagName=html

**17.取消冒泡**

取消默认行为

return false; event.preventDefault

阻止冒泡

cancelBubble(ie) stopPropagation(w3c)

**18.右键菜单**

document.oncontextmenu=function(){return false;}不显示右键菜单

浏览器的自带行为都能用return false屏蔽：右键菜单、form提交、按键输入input

**19.拖拽事件**

按下的点与div左上角的距离不变

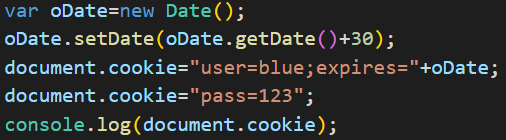
onmousedown 存储距离

onmousemove 根据距离计算div的最新位置

onmouseup 取消掉move事件

**20.cookie**

用来保存用户信息，可以实现记住用户名和自动登陆等，同一个网站所有页面共享一套cookie，没有加密不安全，如果不指定过期时间默认是浏览器关闭。



cookie和ajax都要在服务器环境下运行

**21.ajax**

第一步：创建XMLHttpRequest对象



第二步：设置请求的类型和url及是否异步(一般都为true)

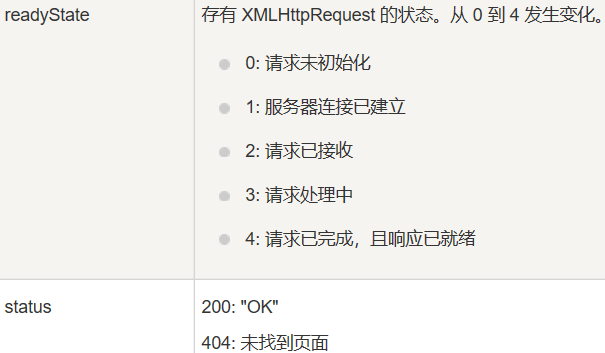
clipboard.png

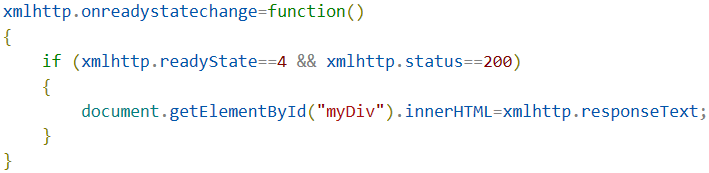
第三步：发送请求

clipboard.png

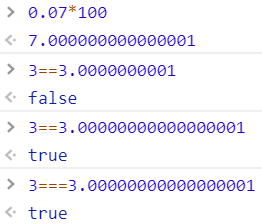
第四步：回调函数

每当 readyState 改变时，就会触发 onreadystatechange 事件





**22.js精度误差**



**23.正则表达式**

常用方法：

obj.replace(rgExp,replaceText)替换

obj.test(exp) obj中是否有匹配的字符串 返回布尔值

obj.search(rgExp) 返回查找结果或-1

obj.exec(str) 返回匹配结果

str.match()返回匹配结果 是String对象的方法

方括号代表有其中一个

[0-9] 数字

[a-z]字母

[0-9a-z]数字或字母

^a 除了a

[^0-9a-z]除了数字字母

+代表多个

.代表任意(个)字符

\d数字

\w === [a-z0-9\_] word

\s space

\D [^0-9]

\W [^0-9a-z\_]

\S 除了空白字符

{n,m} 最少n次，最多m次

{n,} 最少n次，最多不限 i

{,m}最少不限，最多m次

{n} 正好n次

\* 0或任意次

? 0次或1次

邮箱 /^\w+@[a-z0-9]+\.[a-z]{2,4}$/

中文 [\u4e00-\u9fa5]

**24.事件**

绑定事件：

oBtn.onclick=fn(){} 只能绑定一个函数，后面的会覆盖掉前面的

oBtn.attachEvent('onclick',fn) ie支持 可以绑定多个事件 解绑为detachEvent

oBtn.addEventListener('click',fn,false) 第三个参数为是否捕获 解绑为removeEventListener()

setCapture捕获事件 releaseCapture释放捕获

事件冒泡与事件捕获：

IE678只支持事件冒泡，其他两个都支持，所以最好用事件冒泡

事件流分为3个阶段：捕获阶段 目标阶段 冒泡阶段

事件委托：为同一类型事件指定一个事件处理程序，新加入的元素也可以绑定事件

**25.单位**

px像素 em相对于父元素 rem相对于根元素html

vh-viewportHeight vw-viewportWidth 100vh高度和可视区高度相同

**26.沙盒机制**

每一个ios应用程序都会为自己创建一个文件系统目录(文件夹)，这个独立、封闭、安全的空间，叫沙盒。每一个应用程序都会拥有一个应用程序沙盒。应用程序沙盒就是一个文件系统的目录。

沙盒是一种安全体系，它规定了应用程序只能在为该应用程序创建的文件夹(沙盒)内访问文件，不可以访问其他沙盒内的内容。(ios8已部分开放访问)。所有的非代码文件都保存在这个地方，比如图片、声音、属性列表(plist)、splite数据库和文本文件等。

特点：

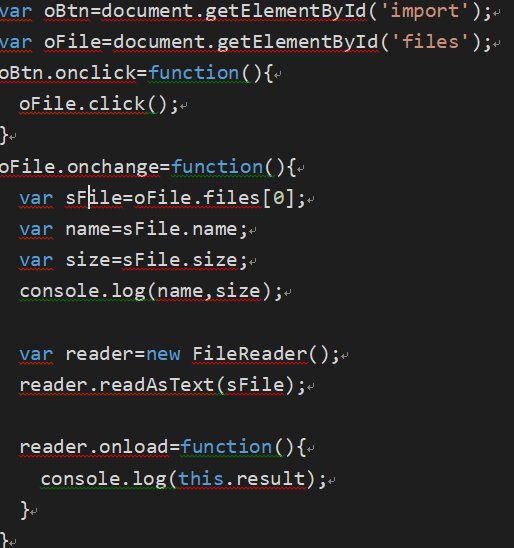
          每个应用程序的活动范围都限制在自己的沙盒内

          不能随意跨越自己 沙盒去访问别的应用程序沙盒中的内容

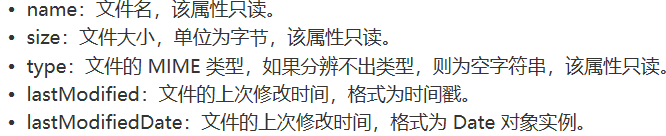
          应用程序向外请求或接收数据都需要经过权限认证

**27.H5实现文件读写**

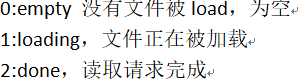
clipboard.png



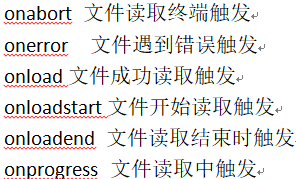
文件的属性：



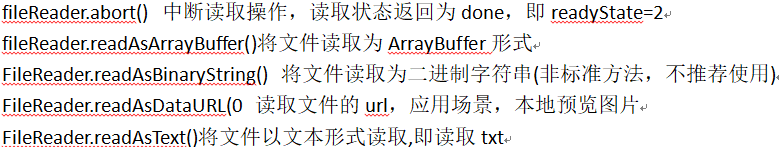
fileReader读取文件内容 fileReader.readyState有3种状态



fileReader事件处理：

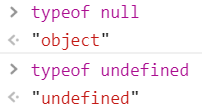


fileReader的5种读取方法：

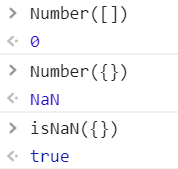


写入：FileWriter 只被Chrome支持

**28.使用typeof判断是否为对象前要判断不等于null**



**29.isNaN在判断的时候会先将值转换为数字**



**30.慎用||填充默认值**

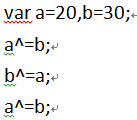
expr1||expr2的意思是如果expr1能转换成true，则返回expr1，否则返回expr2

不能转换为true的：null undefined NaN 0 ''

**31.令placeholder居中**

clipboard.png

**32.两个整数交换值**



**33.用解构赋值获取map的键名和键值**

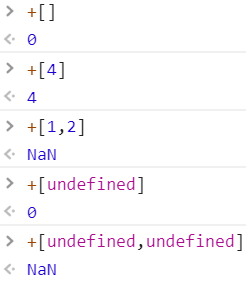


**34.var**

var a=b=3;相当于var a=3;b=3;b是全局的

**35.数组转换为数字**

当数组长度大于等于2时，转换为数字为NaN



**36.自适应内部元素**

width的新属性：

min-content/fit-content/fill-avaliable/max-content

如果一个元素已经设置过width宽度，再设置fit-content会冲突，这时会向内自适应。比如一个div宽为200，再设置fit-content，div里放一个100\*100的图片，这个div的宽为100

fill-avaliable为撑满空闲空间

min-content表示采用内部元素最小宽度值最大的那个元素宽度作为最终容器宽度

max-content表示采用内部元素宽度值最大的那个元素的宽度作为最终容器的宽度，如果出现文本，则相对于文本不换行