Tween

创建TWEEN：var tween=TWEEN.Tween

设置缓动目标：缓动对象.to(设置目标对象,时间)

创建update回调函数：function updates(){}

缓动开始：缓动对象.start(time);

缓动停止：缓动对象.stop();

缓动停止：缓动对象.end();

时间函数：setinterval

设置时间间隔更新缓动：TWEEN.updata()

缓动特效调用：缓动对象.easing(缓动效果)

**缓动效果**

Linear：匀速

缓动方案：None

Quadratic：二次方的缓动

缓动方案：In：从0开始加速的缓动

Out：减速到0的缓动

InOut：前半段从0开始加速，后半段减速到0的缓动

Cubic：三次方的缓动

Quintic：五次方的缓动

Quartic：四次方的缓动

Sinusoidal：正弦曲线的缓动

Exponential：指数曲线的缓动

Circular：圆形曲线的缓动

Elastic：指数衰减的正弦曲线缓动

Back: 超过范围的三次方缓动

Bounce:指数衰减的反弹缓动

重复设置:缓动对象.repeat(次数) 次数设置为Infinity表示无线循环

Yoyo:缓动对象.yoyo(true) 可以逆向变化 使用yoyo时必须使用repeat

Chain:当执行一种缓动效果后可以继续使用另一个缓动效果

tweenA.Chain(tweenB);效果A-效果B

tweenB.Chain(tweenA)；效果B-效果A

缓动延迟：缓动对象.delay 参数是延迟时间

重复中的延迟时间：缓动对象.repeatDelay

事件

开始事件：onStart

完成事件：onComplete

停止事件：onStop

更新事件：onUpdate

6个步骤

1：创建TWEEN函数 初始对象

2：to 设置目标 终止对象 必须有缓动时间

3：时时更新回调函数

4：缓动效果

5：TWEEN.JS启动 start（）

6：TWEEN.update 设置在时间函数中时时更新缓动

DOM（Document Object Model）文档对象模型

创建新节点

document.createElement("标签名") 创建新元素

document.createTextNode("") 创建文本节点

插入、删除、替换、复制

appendChild() 放置到某元素最后

insertBefore() 插入到某元素前

removeChild() 删除父节点的某个子节点

remove() 删除当前节点

replaceChild() 替换节点

cloneNode（bol） 复制节点

bol为true 深层复制；

bol为false 浅层复制 默认

查找

document.getElementsByTagName() 通过标签名称

document.getElementsByName() 通过元素的Name属性值

document.getElementById() 通过元素Id，唯一性

document.getElementsByTagName() 通过标签名查找元素

document.querySelector() 查找单个元素

返回一个组:document.querySelectorAll

返回第一个子节点:firstChild

返回第一个标签节点:firstElementChild

返回最后一个子节点:lastChild

返回最后一个标签节点:lastElementChild

返回上一个兄弟节点:previousSibling

返回上一个兄弟元素:previousElementSibling

返回下一个兄弟节点:nextSibling

返回下一个兄弟元素:nextElementSibling

返回父级节点:parentNode

返回子节点:childNodes

返回子元素:children

获取属性:getAttribute(“属性名")

设置属性:setAttribute("属性名","属性值")

删除属性:removeAttribute("属性名“)

元素大小

获取元素距离左侧距离（距离最近的定位元素父级，若无定位，则相对窗口）：offsetLeft

获取元素距离顶部距离：offsetTop

包含边框宽度

获取元素外边宽度：offsetWidth

获取元素外边高度：offsetHeight

不包含边框宽度

获取元素宽度：clientWidth

获取元素高度：clientHeight

获取文档整体高度：scrollHeight

获取文档滚动高度：scrollTop

获取窗口高度/宽度：docutmentElement.clientHeight/clientWidth

新增事件

窗口滚动事件：onscroll

窗口大小监听事件：onresize

图片加载事件：onload

nodeName(节点名称)

元素节点的nodeName是标签名称

文本节点的nodeName永远是#text

注释节点的nodeName永远是#comment

nodeValue(节点值)

文本节点，nodeValue属性包含文本

元素节点nodeValue不可用

注释节点nodeValue包括注释内容

nodeType(节点类型)

Document对象常用属性

document.body:body元素

document.title:获取、设置文档的标题

document.URL:获取完整的URL

document.domain:获取域名

查找元素：

Document.getElementById() 根据id获取元素

Document.getElementsByTagName() 根据标签名获取元素

Document.getElementsByName() 根据name属性获取元素

Document.getElementsByClassName() 根据class标签获取元素

Document.querySelector() 根据标签名获取第一个元素

Document.querySelectorAll() 获取所有标签名的元素

节点遍历：

childNodes: 获取所有子节点（包括注释）

children: 所有是标签类型的子节点 获取所有子元素

parentNode: 获取已知节点的父节点

parentElementNode: 获取已知节点的父节点（元素）

firstChild: 第一个子节点

firstElementChild: 第一个子节点 （元素）

lastChild: 最后一个子节点

lastElementChild: 最后一个子节点（元素）

nextSibling: 下一个兄弟节点

nextElementSibling: 下一个兄弟节点(元素)

previousSibling: 上一个兄弟节点

previousElementSibling: 上一个兄弟节点(元素)

属性操作：

设置节点上的属性：setAttribute(属性，值)

获取节点上属性的值：getAttribute(属性)

删除节点上属性：removeAttribute(属性)

获取计算后样式：

currentStyle

ie所支持的获取非行间样式的方法

用法：对象.currentStyle.样式名

getComputedStyle

非ie所支持的获取非行间样式的方法

用法：getComputedStyle(对象,伪类).样式名

onscroll窗口滚动事件

onresize 窗口大小监听事件

onload 图片加载事件

offsetLeft 获取元素距离左侧的距离（距离父级，若无定位，则相对窗口的距离）

offsetTop 获取元素距离顶部的距离

offsetWidth 获取元素自身宽度（包括边框宽度width+border+padding）

offsetHeigh获取元素自身高度（包含边框高度height+border+padding）

clientWidth 获取元素宽度（不包含边框宽度width+padding）

clientHeight 获取元素的高度（不包括边框高度height+padding）

scrollHeight ,scrolWidth返回的是元素的实际内容高度，值=子元素的height值+元素的padding-top+元素的padding-top+父元素padding=bottom.

scrollTop, scrollLeft 返回已经滚动到元素的左边界或上边界的像素数。只有在元素有滚动条的时候，这些像素才有用。

兼容写法

Chrome浏览器

document.body.scrollTop

非chrome浏览器

document.documentElement.scrollTop

创建一个图片对象 var img = new Image();

图片对象的属性 img.src,img.alt , img.width,img.height

图片对象的加载事件 img.onload

预加载的实现

创建一个数组 var imgArr = [];

创建图片对象 var img = new Image();

赋予图片对象路径 img.src =“…...”;

操作获取属性、

get Attribute(“属性名”) 获取 属性

setAttribute（“属性名”，属性值） 设置过属性

removeAttribute(“属性名”) 删除属性

BOM(Browser Object Model) 浏览器对象模型

console.log() 输出到控制台查看

alert() 弹出框（警告框）

confirm() 输入框

prompt() 提示框

open() 打开新页面

close() 关闭当前页面

窗口尺寸

window.innerHeight 浏览器窗口内部高度

window.innerWidth 浏览器窗口内部宽度

window.outerHeight 浏览器的整个高度

window.outerWIdth 浏览器的整个宽度

location对象

location.href 设置或返回完整的URL

location.hash 设置或返回锚

location.hostname 返回主机的域名

location.pathname 返回路径和文件名

location.port 返回主机端口

location.protocol 返回web协议

location.reload() 刷新页面（重新加载）

location.href 打开新页面 生成历史记录 （ 最常用）

location.assign 打开新页面 生成历史记录

window.location 打开新页面 生成历史记录

location.replace() 不会产生历史记录，不能后退到前一个页面

浏览器嗅探

navigator.userAgent （浏览器嗅探写法）

history对象

history.back() 方法加载历史列表中的前一个 URL。

history.forward() 方法加载历史列表中的下一个 URL。

事件

PageX 光标相对于该网页的水平位置

pageY 光标相对于该网页的垂直位置

screenX 光标相对于该屏幕的水平位置

screenY 光标相对于该屏幕的垂直位置

clientX 光标相对于该网页的水平位置 （当前可见区域）

clientY 光标相对于该网页的垂直位置 （当前可见区域）

鼠标事件：

onclick 点击事件

ondblclick 双击事件

oncontextmenu 右击事件

onmousedown 按下事件

onmouseup 抬起事件

onmouseover 移动到元素上

onmouseout 从元素上移开

onmousemove 移动鼠标

键盘事件：

onkeydown 按下按键

Onkeyup 按下按键抬起

keyCode 按键编码

表单事件：

onfocus 当输入框聚焦的时候触发

onblur 当输入框失焦的时候触发

oninput 当输入的时候实时触发

onchange 内容发生改变，并失焦后才触发该事件

添加侦听器：element.addEventListener(‘事件名’,事件函数名,事件冒泡/捕获)

取消侦听器：element.removeEventListener(‘事件名’,事件函数名,事件冒泡/捕获)1

如何阻止冒泡函数，两种写法

e.cancelBubble = true

e.stopPropagation();

阻止默认事件的两种方式

returnValue=false 返回一个值false

ev.preventDefault() prevent阻止 Default不执行

Onmousewheel 在其他浏览器的滚轮事件

DOMMouseScroll 在火狐浏览器中的滚轮事件

window.navigator.userAgent 当前浏览器是哪种浏览器

maxMove=scroll.offsetHeight-moves.offsetHeight; 小块儿可移动的最大距离=滚动框高度-小块儿高度

maxTxt=txt.offsetHeight-warp.clientHeight; 文本可移动的最大距离=文本高度-外边框高度

txtT = mt/maxMove\*maxTxt; 文本的top值/ 文本可移动的最大距离= 小块儿的top值/小块儿可移动的最大距离

非火狐：event.wheelDelta

火狐：event.detail

textContent 文本添加文字

offsetHeight 元素可见高度