1. **html概述**
2. 列表删除

parentNode 直接父级

<**div id="wrap"**>  
 <**ul class="list"**>  
 <**li**>xfz <**a href="javascript:;"**>删除</**a**></**li**>  
 <**li**>xfz2 <**a href="javascript:;"**>删除</**a**></**li**>  
 <**li**>xfz222 <**a href="javascript:;"**>删除</**a**></**li**>  
 <**li**>xfz2222 <**a href="javascript:;"**>删除</**a**></**li**>  
 <**li**>xfz22222 <**a href="javascript:;"**>删除</**a**></**li**>  
 <**li**>xfz222222 <**a href="javascript:;"**>删除</**a**></**li**>  
 </**ul**>  
</**div**>

**window**.**onload**=**function**(){  
 **var** liNodes=**document**.querySelectorAll(**"#wrap>ul>li>a"**);  
 **console**.log(liNodes);  
 **for**(**var** i=0;i<liNodes.**length**;i++){  
 liNodes[i].onclick=**function**(){  
 **this**.**parentNode**.style.**display**=**"none"**;  
 }  
 }  
}

1. offsetParent

与当前元素最近的经过定位(position)的父级元素：

offsetParent（如果body和html直接的margin被清掉）

本身定位为fixed

==> offsetParent:null（不是火狐）。

==> offsetParent:body（火狐）。

本身定位不为fixed

父级没有定位

==> offsetParent:body。

父级有定位

==> offsetParent:定位父级 。

2.haslayout

ie7以下,如果当前元素的某个父级触发了haslayout，

那么offsetParent就会被指向到这个触发了layout特性的父节点上。

3.注意点

1.分清parentNode和offsetParent的区别

parentNode：直接父级。

offsetParent：类似于css的包含块。

2.offsetParent的作用

offsetLeft 和 offsetTop 是参照于offsetParent的内边距边界的。

1. dom里所有的元素都是有offsetLeft 和 offsetTop的。

3.getBoundingClientRect

getBoundingClientRect：一个元素四个角！的相对位置

getBoundingClientRect + 滚动条滚动时元素滚动的距离---> 绝对位置

代表元素border-box的尺寸

height

width

元素左上角的相对位置

left

top

元素右下角的相对位置

right

bottom

4. 获取元素在页面中的位置

绝对位置:到body的距离

相对位置:到视口的距离

window.onload=function(){

var inner1 = document.querySelector("#inner1");

var point1 = getPointAb(inner1);

var point2 = getPointRe(inner1);

console.log(point1,point2);

//boder margin会影响这个函数的取值

function getPointRe(node){

//while循环叠加offsetParent的offsetTop和offsetLeft

var x =0;

var y = 0;

while(node){

x+=node.offsetLeft;

y+=node.offsetTop;

node = node.offsetParent;

}

var L = document.documentElement.scrollLeft||document.body.scrollLeft;

var T = document.documentElement.scrollTop||document.body.scrollTop; return {x:x-L,y:y-T};

}

//boder margin会影响这个函数的取值

function getPointAb(node){

//while循环叠加offsetParent的offsetTop和offsetLeft

var x =0;

var y = 0;

while(node){

x+=node.offsetLeft;

y+=node.offsetTop;

node = node.offsetParent;

}

return {x:x,y:y};

}

}

clientWidth : padding box(可视区域)： clientWidth = width+左右padding

offsetWidth : padding box(可视区域)+border：

offsetWidth = width + 左右padding + 左右boder

clientX、clientY

点击位置距离当前body可视区域的x，y坐标

**2.功能实现**

**2.1 实施目标**

**2.1.1 mac停靠功能**

window.onload=function(){

var r = 320;

var imgNodes = document.querySelectorAll("#wrap > img");

document.onmousemove=function(ev){

ev = ev||event;

for(var i=0;i<imgNodes.length;i++){

var a = imgNodes[i].getBoundingClientRect().left + imgNodes[i].offsetWidth/2 - ev.clientX;

var b = imgNodes[i].getBoundingClientRect().top + imgNodes[i].offsetHeight/2 - ev.clientY;

var c = Math.sqrt(a\*a+b\*b);

if(c>=r){

c=r;

}

imgNodes[i].style.width =128 - c\*0.2 +"px";

}

}

}

1.A>B 表示选择A元素的所有子B元素（只选择第一代）

2.A B   表示选择A元素的所有子B元素（选择在A标签里面所有的B元素，不止第一代）

2.A+B表示HTML中紧随A的B元素：用一个结合符只能选择两个相邻兄弟中的第二个元素

**2.2 实施内容**

**2.2.1 鼠标滚轮事件**

###鼠标滚轮事件

ie/chrome : onmousewheel(dom0)

event.wheelDelta

上：120

下：-120

firefox : DOMMouseScroll 必须用(dom2的标准模式)

event.detail

上：-3

下：3

###阻止dom2的默认行为

if(e.preventDefault){

e.preventDefault();

}

###阻止dom0的默认行为

return false;

window.onload=function(){

var testNode = document.querySelector("#test");

if(testNode.addEventListener){

testNode.addEventListener("DOMMouseScroll",fn);

}

testNode.onmousewheel=fn;

function fn(ev){

ev=ev||event;

var dir="";

if(ev.wheelDelta){

dir = ev.wheelDelta>0?"up":"down";

}

if(ev.detail){

dir = ev.detail<0?"up":"down";

}

/\*

对样式进行设置(特殊性最高)

node.style.height

对样式进行读取

node.style.height (读不到css样式表中属性的)

读取一般通过api来进行读取

testNode.getComputedStyle()

\*/

switch (dir){

case "up":

testNode.style.height = testNode.offsetHeight -10+"px";

//...

break;

case "down":

testNode.style.height = testNode.offsetHeight +10+"px";

//....

break;

}

//禁止事件的默认行为 dom2

if(ev.preventDefault){

ev.preventDefault();

}

//禁止事件的默认行为 dom0

return false;

}

}

offsetWidth       //返回元素的宽度（包括元素宽度、内边距和边框，不包括外边距）

offsetHeight      //返回元素的高度（包括元素高度、内边距和边框，不包括外边距）

clientWidth        //返回元素的宽度（包括元素宽度、内边距，不包括边框和外边距）

clientHeight       //返回元素的高度（包括元素高度、内边距，不包括边框和外边距）

style.width         //返回元素的宽度（包括元素宽度，不包括内边距、边框和外边距）

style.height       //返回元素的高度（包括元素高度，不包括内边距、边框和外边距）

scrollWidth       //返回元素的宽度（包括元素宽度、内边距和溢出尺寸，不包括边框和外边距），无溢出的情况，与clientWidth相同

scrollHeigh       //返回元素的高度（包括元素高度、内边距和溢出尺寸，不包括边框和外边距），无溢出的情况，与clientHeight相同

1. style.width 返回的是字符串，如28px，offsetWidth返回的是数值28；

2. style.width/style.height与scrollWidth/scrollHeight是可读写的属性，clientWidth/clientHeight与offsetWidth/offsetHeight是只读属性

3. style.width的值需要事先定义，否则取到的值为空。而且必须要定义在html里(内联样式)，如果定义在css里，style.height的值仍然为空，但元素偏移有效；而offsetWidth则仍能取到。

offsetTop    //返回元素的上外缘距离最近采用定位父元素内壁的距离，如果父元素中没有采用定位的，则是获取上外边缘距离文档内壁的距离。

             所谓的定位就是position属性值为relative、absolute或者fixed。返回值是一个整数，单位是像素。此属性是只读的。

offsetLeft       //此属性和offsetTop的原理是一样的，只不过方位不同，这里就不多介绍了。

scrollLeft        //此属性可以获取或者设置对象的最左边到对象在当前窗口显示的范围内的左边的距离，也就是元素被滚动条向左拉动的距离。

             返回值是一个整数，单位是像素。此属性是可读写的。

scrollTop   //此属性可以获取或者设置对象的最顶部到对象在当前窗口显示的范围内的顶边的距离，也就是元素滚动条被向下拉动的距离。

             返回值是一个整数，单位是像素。此属性是可读写的。

当鼠标事件发生时（不管是onclick，还是omousemove，onmouseover等）

clientX        鼠标相对于浏览器（这里说的是浏览器的有效区域）左上角x轴的坐标；  不随滚动条滚动而改变；

clientY        鼠标相对于浏览器（这里说的是浏览器的有效区域）左上角y轴的坐标；  不随滚动条滚动而改变；

pageX        鼠标相对于浏览器（这里说的是浏览器的有效区域）左上角x轴的坐标；  随滚动条滚动而改变；

pageY        鼠标相对于浏览器（这里说的是浏览器的有效区域）左上角y轴的坐标；  随滚动条滚动而改变；

screenX     鼠标相对于显示器屏幕左上角x轴的坐标；

screenY      鼠标相对于显示器屏幕左上角y轴的坐标；

offsetX        鼠标相对于事件源左上角X轴的坐标

offsetY        鼠标相对于事件源左上角Y轴的坐标