1. table

属性和方法：

<table>:tBodies

<tbody>:rows deleteRow(pos) insertRow(pos)

<tr>:cells deleteCell(pos) insertCell(pos)

创建一个2行2列的表格

var table=document.createElement("table");

table.border=1;

table.width="100%";

var tBody=document.createElement("tBody");

table.appendChild(tBody);

tBody.insertRow(0);

tBody.rows[0].insertCell(0);

tBody.rows[0].cells[0].appendChild(document.createTextNode("cell 1-1"));

tBody.rows[0].insertCell(1);

tBody.rows[0].cells[1].appendChild(document.createTextNode("cell 1-2"));

tBody.insertRow(1);

tBody.rows[1].insertCell(0);

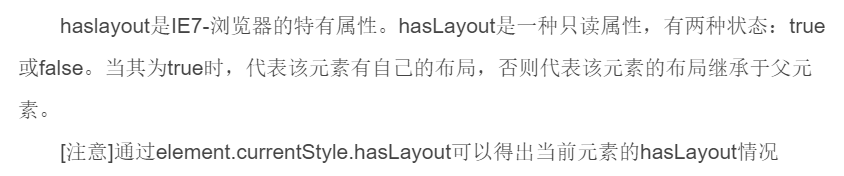
tBody.rows[1].cells[0].appendChild(document.createTextNode("cell -1"));

tBody.rows[1].insertCell(1);

tBody.rows[1].cells[1].appendChild(document.createTextNode("cell 2-2"));

document.body.appendChild(table);

1. hasLayout



1. margin
   1. 没有继承性
   2. ie6下双边距问题

margin双边距布局产生原因：block元素+浮动+margin



* 1. margin重叠问题

水平边距不会重合

边距宽度是相邻边距中较大的值

如果出现负边距，则是最大正边距减去绝对值最大的负边距

如果没有正边距，则从零中减去绝对值最大的负边距

* 1. margin不起作用的情况

多出现在firefox

浮动影响/margin重叠/padding与margin重叠

* 1. margin负值的应用
     1. 页面上实现css sprite背景定位效果

css sprite css精灵图/雪碧图

把页面上所有的图片集中到一张大图上

* + 1. 流动性布局
    2. 选项卡边框线

.title{border-right: 1px solid red;padding: 5px 15px;}

.menu-content li{display: none;margin-top: -1px;border-top: 1px solid black;}

* + 1. 左右边框

.box div{font-size: 14px;display: inline-block;border-left: 1px solid red;border-right: 1px solid red;margin-left: -1px;}

<div style="font-size:0px;" class="box">

<div>div1</div>

<div>div2</div>

<div>div3</div>

<div>div4</div>

</div>



* + 1. 图片与文字对齐问题

图片与文字默认是居底对齐

使用vertical-align:top对ie的兼容性不够好

可以使用margin

img{width: 50px;margin: 0 0 -40px 0;}

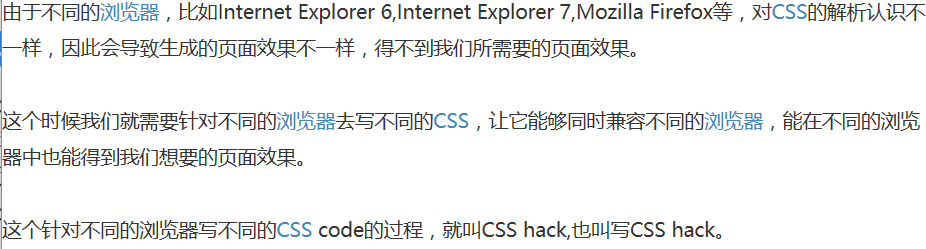
<div class="box">

aaaa

<img src="./img.PNG" alt="">

</div>

1. hack



1. 去除display:inline-block产生间隙的办法
   1. 在编辑器中去掉空格
   2. 不闭合标签
   3. html注释掉空格
   4. 给父元素设置font-size:0;

html{-webkit-text-size-adjust:none;/\* 使用webkit的私有属性，让字体大小不受设备终端的调整，可定义字体大小小于12px \*/}

div{font-size: 0px;\*word-spacing:-1px;/\* 使用word-spacing 修复 IE6、7 中始终存在的 1px 空隙，减少单词间的空白（即字间隔） \*/}

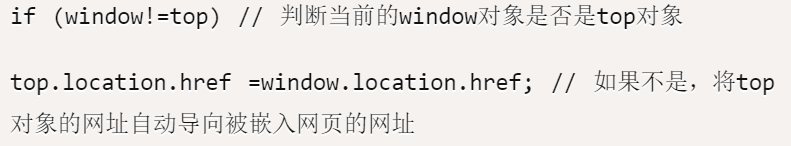
div span{font-size: 14px;letter-spacing: normal;/\* 设置字母、字间距为0 \*/ word-spacing: normal; /\* 设置单词、字段间距为0 \*/}

1. a标签

<a href="#">链接1</a><!--点击时url后面会加个#-->

<a href="javascript:void(0)">链接2</a><!--推荐这种-->

1. 防止网页被嵌入框架的代码



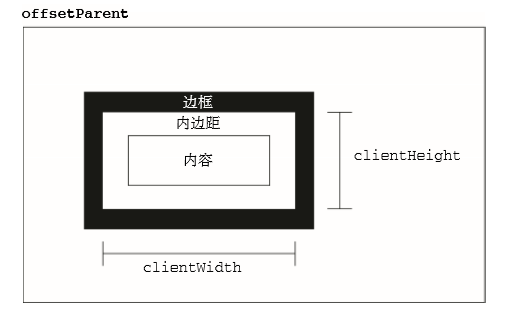
1. 闭包
2. 用js获取页面元素的位置
   1. 网页大小和浏览器窗口大小

一张网页的全面积就是它的大小

浏览器窗口的大小，指的是浏览器窗口中看到的那部分的面积，又叫viewport(视口)

* 1. 获取网页大小-clientWidth clientHeight

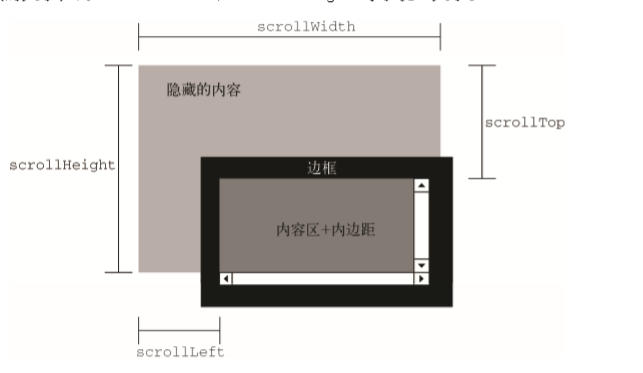
clientWidth和clientHeight不包括border和滚动条



var width=document.body.clientWidth||document.documentElement.clientWidth;

var height=document.body.clientHeight||document.documentElement.clientHeight;

* 1. 获取网页大小-scrollHeight scrollWidth



function getPageArea(){

　　　　if (document.compatMode == "BackCompat"){

　　　　　　return {

　　　　　　　　width: Math.max(document.body.scrollWidth,

　　　　　　　　　　　　　　　　document.body.clientWidth),

　　　　　　　　height: Math.max(document.body.scrollHeight,

　　　　　　　　　　　　　　　　document.body.clientHeight)

　　　　　　}

　　　　} else {

　　　　　　return {

　　　　　　　　width: Math.max(document.documentElement.scrollWidth,

　　　　　　　　　　　　　　　　document.documentElement.clientWidth),

　　　　　　　　height: Math.max(document.documentElement.scrollHeight,

　　　　　　　　　　　　　　　　document.documentElement.clientHeight)

　　　　　　}

　　　　}

　　}

console.log(getPageArea());//Object{width:100px,height:100px}

* 1. 获取网页元素的绝对位置

该元素的左上角相对于整个网页左上角的坐标

function getElementLeft(element){

　　　　var actualLeft = element.offsetLeft;

　　　　var current = element.offsetParent;

　　　　while (current !== null){

　　　　　　actualLeft += current.offsetLeft;

　　　　　　current = current.offsetParent;

　　　　}

　　　　return actualLeft;

　　}

　　function getElementTop(element){

　　　　var actualTop = element.offsetTop;

　　　　var current = element.offsetParent;

　　　　while (current !== null){

　　　　　　actualTop += current.offsetTop;

　　　　　　current = current.offsetParent;

　　　　}

　　　　return actualTop;

　　}

* 1. 获取网页元素的相对位置

该元素相对于浏览器窗口左上角的坐标

function getElementViewLeft(element){

　　　　var actualLeft = element.offsetLeft;

　　　　var current = element.offsetParent;

　　　　while (current !== null){

　　　　　　actualLeft += current.offsetLeft;

　　　　　　current = current.offsetParent;

　　　　}

　　　　if (document.compatMode == "BackCompat"){

　　　　　　var elementScrollLeft=document.body.scrollLeft;

　　　　} else {

　　　　　　var elementScrollLeft=document.documentElement.scrollLeft;

　　　　}

　　　　return actualLeft-elementScrollLeft;

　　}

　　function getElementViewTop(element){

　　　　var actualTop = element.offsetTop;

　　　　var current = element.offsetParent;

　　　　while (current !== null){

　　　　　　actualTop += current. offsetTop;

　　　　　　current = current.offsetParent;

　　　　}

　　　　 if (document.compatMode == "BackCompat"){

　　　　　　var elementScrollTop=document.body.scrollTop;

　　　　} else {

　　　　　　var elementScrollTop=document.documentElement.scrollTop;

　　　　}

　　　　return actualTop-elementScrollTop;

　　}