**1、内联元素**，

就是不可设置宽高的元素,上下margin:设置无效，就是设置了也不生效，比如a、span等元素，但是可以通过display:block;浮动以及定位等一些方法将内联元素转换为块状元素。

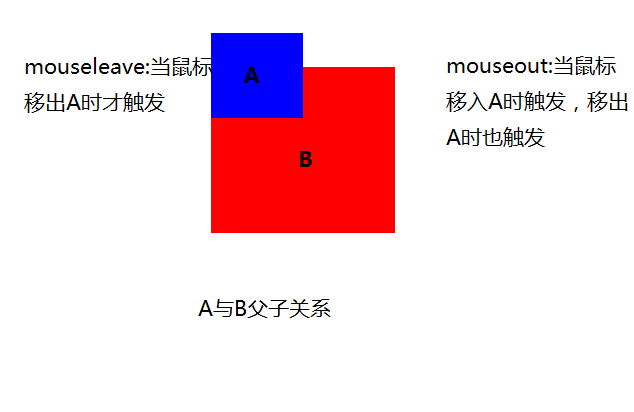
**2、块状元素**

默认情况下单独占一行，可以设置宽高。当然了，块状元素也可以转为内联元素，方法是是设置display:inline;

那么现在就可以解决楼主的问题了，你想让p元素和里面的内容宽度相同，那么你要知道p是块状元素，那么只需要设置p元素的display值为inline就可以了p{display:inline;},然后p就成了内联元素了，内联元素的尺寸是随着里面的内容而变化的。

**3、onmouseleave** 和**onmouseout** 的区别:

Leave 当鼠标移出元素时触发，元素上被子级覆盖，鼠标移到子级时不出发函数（mouseover）

out: 当鼠标移出元素时触发，元素上被子级覆盖，鼠标移到子级上时就触发函数(mouseenter)

# **4、[setTimeout与setTimeinterval的使用](http://www.cnblogs.com/yony/archive/2012/06/21/2557766.html)**

实际上，setTimeout和setInterval的语法相同。它们都有两个参数，一个是将要执行的代码字符串，还有一个是以毫秒为单位的时间间隔，当过了那个时间段之后就将执行那段代码。不过这两个函数还是有区别的，setInterval在执行完一次代码之后，经过了那个固定的时间间隔，它还会自动重复执行代码，而setTimeout只执行一次那段代码。

1. **innerHTML：** 就是元素里面所有的东西，包括文字 、换行、标签等。
2. **className:** 用来给标签添加类名：比如<div></div> 用js给div添加类名，oDiv.className=”类名”；
3. **在js的属性操作中不允许出项 ”-”,比如 oDiv.background-color是错误的，应该用驼峰式命名法：如oDiv.backgroundColor。**
4. **动画，animation**   
   :首先定义一套关键帧  
    方法：@keyframes 动画名称  
    {  
    0%{ 相应样式 }（可以从0%设置到100%）  
    100%{ 相应样式 }}  
   调用方法：animation:时间，延迟时间 ，动画名称，运动方式，动画执行的次数；
5. **设置透明度**  
   opacity:0.5; filter:Alpha(opacity=50);(兼容IE)
6. **perspective** 属性定义 3D 元素距视图的距离，以像素计。该属性允许您改变 3D 元素查看 3D 元素的视图。  
   这就是3D世界的坐标图。而perspective在Z轴上，而perspective是设置Z轴的长度。在css3中只有设置了perspective的值元素才会有Z轴，perspective为none元素则没有Z轴(就是2D元素)。  
   RotateZ:div围绕Z轴旋转，以div中心点为轴心旋转  
   rotateX:div围绕X轴旋转，以div横向中轴为轴心前后翻转  
   rotateY:div围绕Y旋转，以div纵向中轴为轴心左右翻转rotate的三个值找到了3D坐标，接下来找找perspective透视焦点。
7. **Clear:用来清浮动** 比如**clear:both,元素左右都不能有浮动元素；**
8. **3D** <http://www.zhangxinxu.com/wordpress/2012/09/css3-3d-transform-perspective-animate-transition/>
9. **元素居中**  
    <https://www.jianshu.com/p/af8c15dbc140>
10. **弹性盒模型**(Flexible Box Model)  
    dispaly:-webkit-box;与浮动有点像但比float功能强；  
    1、排列方式：--webki-box-orient:水平horizontal和垂直vertical  
    2、子级的排列位置，可以给父级加-webkit-box-pack:  
    方法：  
     -webkit-box-pack:start(左边)默认  
     -webkit-box-pack：justify(均分) center(居中)end(右边)  
    3、-webkit-box-flex：数字 子级根据数字比例来分父级
11. **Border-radius:horizontal/vertical(用/分开)**
12. **多背景  
    1、background：url() x y no-repeat,url() x y no-repeat  
    (x ,y 为图片显示位置)  
    2、background-size:设置图片大小，cover(铺满整个标签，保持原来比例，图片可能超出标签。导致图片显示不全) 、contain（保证图片全显示，不管是否铺满整个标签）  
    3、background-clip:(border-box padding-box content-box text)从边框显示（当边框宽度大是，且颜色有透明时，设置border-box时，图片会从边框开始显示。text需要加webkit）**
13. **渐变(重复渐变要加repeating-)**  
    linear-gradient(angle,rgba px,rgba px,rgba px,rgba px) angle也可以改为to  
    radial-gradient(at rgba px rgba px rgba px)  
    at 后可以接模糊值，如left center 为圆心位置，也可以接具体数值（圆的形状默认为椭圆，圆形时设置circle）
14. **倒影 -webkit-box-reflect**  
    第一个值：上above 下below 第二个值：中间距离 第三个值:可以加上渐变

**19、阻止冒泡：stopPropagation(IE:event.cancelBubble)  
20、事件委托：利用事件冒泡，和事件源对象进行处理**  
<ul>  
<li>1</li>  
<li>2</li>  
<li>3</li>  
<li>4</li>  
</ul>  
点击li标签弹出标签内容：  
 ul.onclick=function(e){  
 var event=e||window.event;  
 var target=event.target||event.srcElement;  
 console.log(target.innerText);  
   
 }

**21、添加事件：**  
1、addEventListener( “click” ,function(),flase )  
 不能能重复绑定同一事件函数  
2、attachEvent(“onclick”,function())

**22、input 消除点击后出现边框：伪类：focus{ outline:none}**

**23、事件对象兼容形参写法** var event= e||window.event;

Var target=event.target||event.srcElement;(对父级绑定事件，target为子级元素)

1. **盒模型box-sizing** （1）标准盒模型大小：border+padding+width/height;(border-box)  
    （2）怪异盒模型大小：width/height;(content-box)
2. **绑定事件** （W3C）obj.addEventlistener(“click”,function,false)  
    （IE） obj.attachEvent(on+”click”,function);

**26、事件处理模型** （1）事件冒泡：子级触发事件，事件会一级一级向祖辈传递。  
 （2）事件捕获：子级触发事件，祖辈捕获事件并触发，接着一级 一级向子辈传递。（focus，blur,change,submit,reset,select等事件不冒泡）

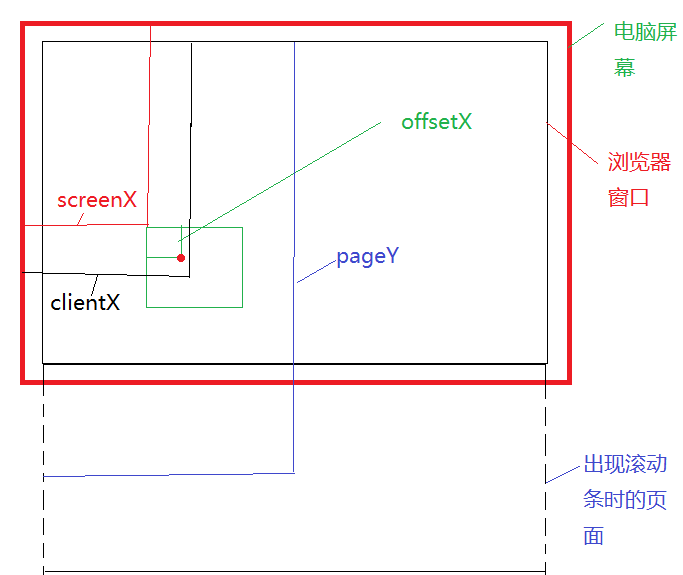
**27、parseInt、number** parseInt:可以转化非纯数字类型字符串：“123add”->123  
 number:自能转化纯数字类型字符串：“123add”->NaN

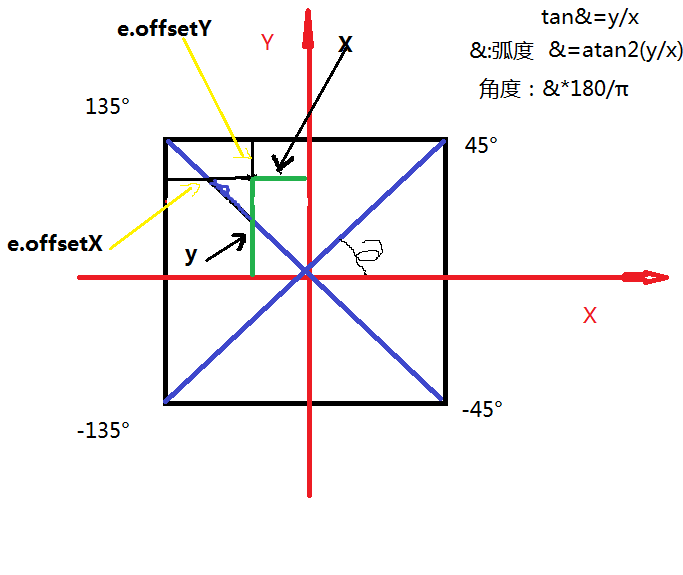
Number(null)->0、Number(true/false)->1/0;

number(undefined)->NaN

parseInt(null),parseInt(undefined)->NaN

parseInt(true/false)->NaN

**28、clientX、offsetX、pageX、screenX区别  
 （1）clientX：相对浏览器可视区域坐标  
 （2）offsetX：相对于元素自身坐标  
 （3）pageX：相对浏览器整体大小坐标（出现滚动条时与clientX有区别）  
 （4）screenX：相对于整个屏幕坐标**

1. **求鼠标从div哪个方向移入移出（哪个边）  
     
   正方形：**

X=e.offsetX-this.offsetWidth/2  
 y=this.offsetHeight/2-e.offsetY

**长方形：**

X=(e.offsetX-this.offsetWidth/2)\*(Width>height?(heiht/width):1)  
 y=(this.offsetHeight/2-e.offsetY)\*(height>Width?(width/heiht):1)

点（x,y）对应的弧度值：rad=Math.atan2(y,x)  
点（x,y）对应的角度值：angle=rad\*180/π

**此时x,y相应坐标对应上方坐标系，也就是以div中心点为坐标原点。**

1. **+180°使得角度范围为0-360**
2. **除90°**
3. **Math.round()取整**
4. **加3**
5. **模4**

**div.top： 135°45°315-225**

**div.right：-45°45°135-225**

**div.bottom：-45°-135°135-45**

**div.left：135°-315°45-135  
30、脱离文档流** （1）当元素设置float或position:absolute(relative不会产生)时，元素会脱离文档流。  
 （2）脱离文档流的元素具有inline-block属性：（除代码换行外）  
 相同元素排在同一行  
 内容撑开宽度（当不设宽时）  
 **解决方法**：（1）父级设置overflow:hidden;(当子级因为定位脱离文档流时，overflow无效)

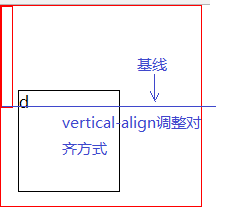
**31、BFC(blcok Formatting context)（清浮动：clear：both;）**

(1) 父级浮动 弊端：margin:0 auto;失效

(2) 父级加inline-block; 弊端：margin:0 auto;失效

(3) 父级overflow:hidden;

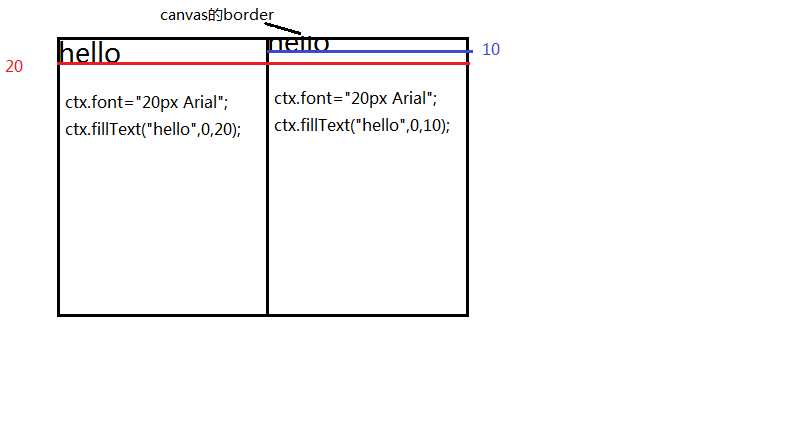
(4) 父级 position:absolute;

(5) 伪类清浮动：:after{ content=””;display:block;clear:both;} **32、input事件** onchange：焦点失去时触发  
 oninput：当input框中数据发生变化时触发  
**33、当inline-block元素后有其他文本时对齐方式  
vertical-align:top:字母d会top对齐，对应的盒子也会跑上去。**

**字母d垂直对齐:vertical-aling:top**

**Line-height:**

1. **canvas中fillText(string,x,y)  
    其中x是开始绘制的起点  
    y绘制位置比较特殊如下图：（y坐标为字体底部开始坐标）**

****

1. **Margin**

inline和inline-block：不支持 margin：0 auto;

Inline元素不支持上下margin

1. **居中**

**水平居中：** 块级元素：margin：0 auto；

Left:50%;transform：translateX(-50%);

行级元素：text-align:center;

**垂直居中：**

块级元素：（1）position:absolute:

Top:50%;left:50%

(Mrgin-top/margin-left:负的自身高度 宽度 一半)或(transform:translate(-50%,-50%))

1. **数组（对象）遍历操作  
    (1)for(proto in arr):对于数组 proto为index、对于对象proto为属性**
2. **数组for( value of arr) value为数组每位值，  
    对象for(key of Object.keys(obj) key为对象属性  
    对象for(value of Object.values(obj) value为对象值**

**(3)arr.forEach(function(ele, index){}):ele对应每位的值（对象没有此 方法）  
(4)arr.filter(function(ele,index,self){}):self数组自己return true:时会返回对应的值；  
(5)reduce(total,currentValue, index,arr)**

|  |  |
| --- | --- |
| total | 必需。初始值, 或者计算结束后的返回值。 |
| currentValue | 必需。当前元素 |
| currentIndex | 可选。当前元素的索引 |
| arr | 可选。当前元素所属的数组对象。 |

**38、跨源通信postMessage**

（发送）Otherwindow.postMessage(“data”,url()) url指定哪个地址的窗口 可以接受消息：

（接收）Window.addEventlistener(‘message’,function(){  
 event.data;(接收的数据)  
 event.origin(消息来源)

},false)

1. **jquery:  
    （1）**css:(属性可以不用加引号)  
    (1)添加或更改一个属性时：css(“color”:”red”)  
    (2)添加多个时(传入对象):css({“color”:”red”，”background”:”blue”}) **（2）**$(“div”)[0]:返回的是BOM  
    **（3）**>匹配所有父元素下面的所有子元素（不包含孙元素）  
    **（4）**prev+next匹配prev紧接着相邻的next元素  
    **（5）**:gt(index)：匹配序号大于index的同类元素。  
    **（6）**:lt(index)：匹配从0开始到index同类元素  
    **（7）**div:has(p)：匹配含有p元素的div.
2. ul:parent：匹配含有子元素或文本的ul元素。  
    **（9）**on(EventType,EventTarget,parameter,function)
3. 浏览器常驻线程  
    （1）js引擎线程  
    （2）GUI线程  
    （3）http网络请求线程  
    （4）定时器触发器线程  
    （5）浏览器事件处理线程  
   40、p:nth-child(n) 找p所在的父级中的第n个是p的子元素

41、设置opacity元素的所有后代元素会随着一起具有透明性，包括其中的文字。设置透明在background：中使用rgba;

42、onkeydown

这个事件在用户按下任何键盘键（包括系统按钮，如箭头键和功能键）时发生。

onkeypress

这个事件在用户按下并放开任何字母数字键时发生。系统按钮（例如，箭头键和功能键）无法得到识别。

1. 单行文本越界呈现...  
    父级：overflow：hidden; text-overflow:ellipsis; white-space:nowrap;
2. 网站logo做法： CSS不能加载，照样显示连接
3. 将文字移出盒子，用text-indent:px(首行缩进，缩进大小为盒子的宽度)，white-space:nowrap; overfow:hidden;  
    2、将盒子高度设为0，设置padding-top高度，让后让padding-top显示图片背景，将文字挤出盒子，然后设置overflow：hidden；
4. p标签里不能套块级元素，a标签里不能套a标签；
5. offsetTop/offsetLeft:  
    相对于有定位父级的元素的上边界/左边界的距离；
6. Canvas 中的globalCompositeOperation：  
    https://blog.csdn.net/laijieyao/article/details/41862473  
    1、source-over  
    这是默认值，他表示绘制的图形将画在现有画布之上

2、destination-over  
 这个操作的值与前一个值相反，所以现在目标绘制在源之上  
 3、source-atop

这个操作会将源绘制在目标之上，但是在重叠区域上两者都是不透明的。绘制在其他位置的目标是不透明的，但源是透明的。  
 4、destination-atop

这个操作与source-atop相反，目标绘制在源之上

5、source-in

在源于目标重叠的区域只绘制源，而不重叠的部分编程透明的。  
 6、destination-in

这个操作与source-in相反，在源于目标重叠的区域保留目标。而不重叠的部分都变成透明的。  
 7、source-out

在与目标不重叠的区域上绘制源，其他部分都变成透明的。  
 8、destination-out

在与源不重叠的区域上保留目标。其他部分都变成透明的。  
 9、lighter

这个值与顺序无关，如果源与目标重叠，就将两者的颜色值想家。得到的颜色值的最大取值为255，结果就为白色。  
 10、copy

这个值与顺序无关，只绘制源，覆盖掉目标。  
 11、xor

这个值与顺序无关，只绘制出不重叠的源与目标区域。所有重叠的部分都变成透明的

1. BOM对象包括：  
    1、Navigator 包括客户端浏览器的信息；  
    2、History 包括了浏览器窗口访问过的URL；
2. Location 包括了当前URL的信息；  
    4、Screen 包含客户端显示屏的信息；

5、Window

49、js事件线  
 1、创建Document对象，开始解析web页面，解析HTML元素和他们的文本内容后添加Element对象和Text节点到文档中，这个阶段document.readyState=”loading”.

2、遇到link外部css。创建线程加载，并继续解析文档。

3、遇到script外部js,并且没有设置async,defer,浏览器加载、并阻塞、等待js加载完成并执行脚本。然后继续解析文档。

4、遇到script外部js，并且设置async，defer，浏览器创建线程加载，并继续解析文档。对于async属性的脚本，脚本加载完成后立即执行，(异步禁止使用document。Write（）)

5、遇到img等，先正常解析dom结构，然后浏览器异步加载src，并继续解析文档。

6、当文档解析完成，document.readyState=”interative”

7、文档解析完成后，所有设置有defer的脚本会按照顺序执行。

8、document对象触发DOMContentLoaded事件，这也标志着程序执行从同步脚本执行阶段，转化为事件驱动阶段。

9、当所有async的脚本加载完成并执行后，img等加载完成后，documen.readyState=”complete”,window对象触发load事件。

50、取消页面默认可以选中东西事件  
 document.body.style.userSelect="none";  
 可以通过设置属性：user-select:none;  
 ::selection （伪类选择器，表示鼠标选中的）

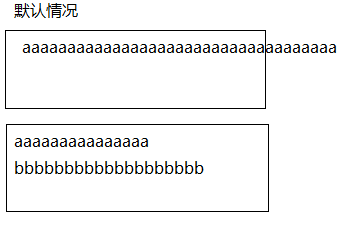
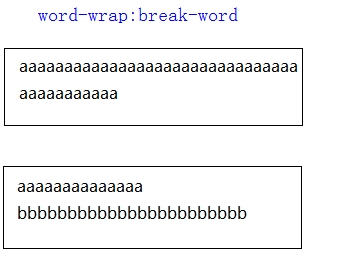
51、取消图片可以拖动事件  
 取消onmousedown默认事件 e.preventDefault();

52、文档碎片：

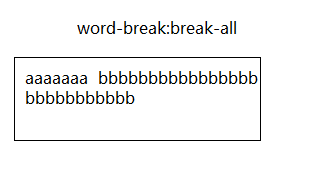
Doucment.createDocumentFragment();

53、获取frame标签中的document  
 oframe.contentDocument;

54、如果单词能宽度小于父级宽度，默认超出的哪个单词会另起一行，  
 如果单词宽度大于父级宽度，默认会超出父级。

  
 word-wrap:break-word;单词长度超过父级宽度截断单词另起一行

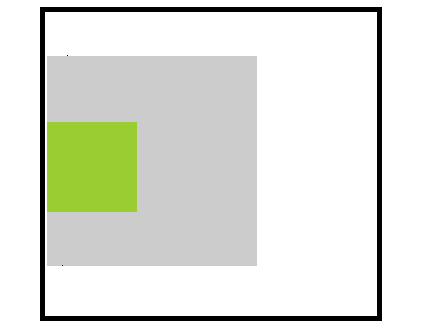
word-break:break-all;



white-space:nowrap; 不换行

1. vertical-align  
    当两个行级元素放在同一个标签内，为了防止其中一个因设置了margin导致其他元素向上或下移动，可以设置vertical-align
2. **Z-index 仅能在定位元素上奏效**
3. **伪元素定位以自身元素为父级进行定位  
    li:hover::after{} 可以使用；**
4. **子级height：为0时，用padding用%撑起高度时，不管父级是否设置了高度。Padding高度相对于父级宽度的%；和父级高度无关，与自己宽度也无关；**
5. **Btoa: 将字符串进行base64编码**

**Atob: 将base64编码转化成字符串(AsciII)**

1. **inline-block垂直居中问题：  
    **

**Div 中间两个inline-block元素同时居中，  
 1、父级div css: 设置行高; font-size:0px;  
 2、子级 css:vertical-align:middle;**

1. **img 加载失败触发onerror事件**
2. **addEventListener  
    第二个参数可以是一个对象，对象中必须有handleEvent属性对应的函数。事件触发时系统自动调用对象中的handleEvent.**

**63、clientLeft:求的时边框宽度**

**64、***window.onbeforeunload， window.onunload  
 浏览器关闭时触发，chrome 没反应*

**65、自定义事件：**

// Create the event.var event = document.createEvent('Event');

// Define that the event name is 'build'.

event.initEvent('build', true, true);

// Listen for the event.

elem.addEventListener('build', function (e) {

// e.target matches elem}, false);

// target can be any Element or other EventTarget.

elem.dispatchEvent(event);

**66、transition 结束事件：transitionend；注意：当离开当前页面时，会发生不出发现象。**

**67、visibilitychange：事件用来监听页面是否显示在浏览器中**

// 页面的 visibility 属性可能返回三种状态

// prerender，visible 和 hidden

let pageVisibility = document.visibilityState;

// 监听 visibility change 事件

document.addEventListener('visibilitychange', function() {

// 页面变为不可见时触发

if (document.visibilityState == 'hidden') { ... }

// 页面变为可见时触发

if (document.visibilityState == 'visible') { ... }

}

);

1. **window.VBArray 可以判断是否是IE浏览器， document.documentMode 返回当前IE版本号**
2. **元素清浮动导致高度改变：解决方案：轻浮动元素设置 overflow：hidden**
3. **Document.documentElement 代表当前html   
    清当前clientHeight/clientWidth:可以用document.doucmentElement.clientHeight/clientWidth**
4. **Rem 相对于文档根节点font-size大小。浏览器默认为16px.**
5. **Rm 相对于元素父节点font-size大小**
6. **Rem 布局：  
    1.页面加载设置根节点字体大小为 宽度的1/100px;(结果小于12px时，使用1/10)**
7. **根元素下的dom宽度以rem为单位。计算方法为当前dom px尺寸/设计稿尺寸\*100（意思是，当前元素占总宽度的半分数）**
8. **html5资源预加载：**

<link rel="prefetch" href="http://www.example.com/">