用css实现一个三角形

.con{

width: 0;

height: 0;

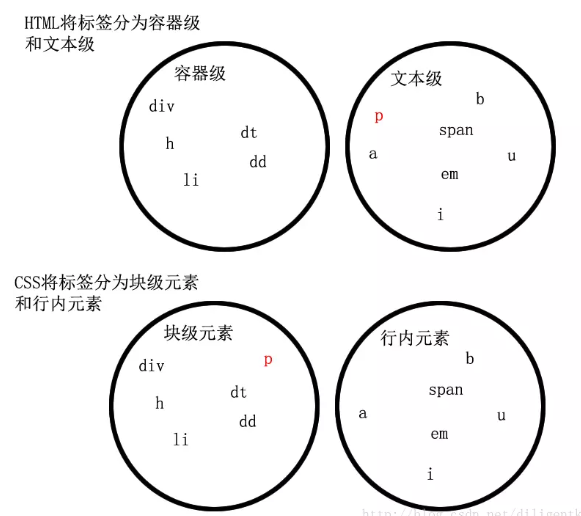
border-width: 40px;

border-style: solid;

border-color: red transparent transparent transparent;

}

所谓文档流，指元素在排版布局的过程中，元素会自动从左到右，从上到下的流式排列。脱离文档流呢，就是元素打乱了这个排列，或是从排版中拿走。

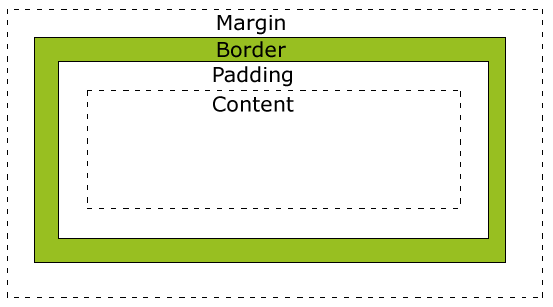


## CSS 盒子模型(Box Model)

所有HTML元素可以看作盒子，封装周围的HTML元素，它包括：边距，边框，填充，和实际内容。

盒模型允许我们在其它元素和周围元素边框之间的空间放置元素。

下面的图片说明了盒子模型(Box Model)：



****文档流****指的是元素排版布局过程中，元素会****默认****自动从左往右，从上往下的****流式排列方式****。并最终窗体自上而下分成一行行，并在每行中从左至右的顺序排放元素。

因为float的元素不占用文档流空间，为了解决上面的问题，就需要在float之后的元素上面使用clear，在此例中，我们在div3后面加入一个空段落，并设置其为clear，   
<div id="div1">div1  
<div id="div2">div2</div>  
<div id="div3">div3</div>  
<p class="clear"></p>  
</div>  
  
clear 属性定义了元素的哪边上不允许出现浮动元素。下面是新增加的空段落的CSS代码：

.clear{  
  clear:both;  
}  
这时div1有了p这块内容（尽管p里面是空的）,并且clear:为left, 使得p的上外边框边界刚好在其上浮动元素的下外边距边界之下

Position：

|  |  |
| --- | --- |
| absolute | 生成绝对定位的元素，相对于 static 定位以外的第一个父元素进行定位。  元素的位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。 |
| fixed | 生成绝对定位的元素，相对于浏览器窗口进行定位。固定定位  元素的位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。 |
| relative | 生成相对定位的元素，相对于其正常位置进行定位。  因此，"left:20" 会向元素的 LEFT 位置添加 20 像素。 |
| static | 默认值。没有定位，元素出现在正常的流中（忽略 top, bottom, left, right 或者 z-index 声明）。 |
| inherit | 规定应该从父元素继承 position 属性的值。 |

1、float

2、position:absolute;[绝对定位](https://www.baidu.com/s?wd=%E7%BB%9D%E5%AF%B9%E5%AE%9A%E4%BD%8D&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "https://zhidao.baidu.com/question/_blank)

3、position:fixed;固定定位 //IE6不兼容

Display：

none此元素不会被显示。

block此元素将显示为块级元素，此元素前后会带有换行符。

inline默认。此元素会被显示为内联元素，元素前后没有换行符。

inline-block行内块元素。（CSS2.1 新增的值）

常见布局方式：

a、普通布局：前端中用到的标签主要分为块级标签（display：blcok）和行级标签(display:inline)。块级标签为必须占一行的标签元素，后面的元素必须在下一行显示；行级标签是显示时后面的行级元素紧跟着前一个行级标签显示。普通布局是在按照这些元素的特性进行界面的布局，从左到右，从上到下的顺序进行。另：普通元素的position为static。

b、浮动布局：浮动布局包括左右浮动，即相对于普通布局，该布局元素会脱离普通布局也是流布局，相当于浮动元素会在普通布局之上进行界面的布局，会导致的问题就是会有普通布局的部分被浮动布局元素给遮挡，解决办法为清除浮动，即在被遮挡的普通元素上添加clear：both进行浮动的清除，使得普通布局的元素不会跟浮动元素在一行显示并另起一行进行dom文档的渲染，清除浮动还有其他办法请自行查询。另：浮动布局中浮动的范围是根据父元素的位置进行浮动位置定位。

c、绝对布局：常用的position为absolute时的元素会进行绝对布局。绝对定位的参考位置为其祖先元素中position为非static的第一个元素。所以一般将参考位置元素设置为position：relative。

[display](https://developer.mozilla.org/en-US/docs/Web/CSS/display" \t "http://www.ruanyifeng.com/blog/2015/07/_blank) 属性 + [position](https://developer.mozilla.org/en-US/docs/Web/CSS/position" \t "http://www.ruanyifeng.com/blog/2015/07/_blank)属性 + [float](https://developer.mozilla.org/en-US/docs/Web/CSS/float" \t "http://www.ruanyifeng.com/blog/2015/07/_blank)属性

1. 任意元素垂直水平居中（多种方法）

<div class="container">Hello World！</div>

.container {

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%); // 自身宽度和高度的一半

border: 1px solid red;

}

.container {

width: 300px;

margin: 50vh auto;

transform: translateY(-50%);

border: 1px solid red;

}

动画：

当您在 @keyframes 中创建动画时，请把它捆绑到某个元素。

通过规定至少以下两项 CSS3 动画属性，即可将动画绑定到选择器：

* 规定动画的名称
* 规定动画的时长

把 "myfirst" 动画捆绑到 div 元素，同时可以设置animation的属性（名称，时长，次数，速度曲线）。时长：5 秒：

div

{

animation: myfirst 5s;

-moz-animation: myfirst 5s; /\* Firefox \*/

-webkit-animation: myfirst 5s; /\* Safari 和 Chrome \*/

-o-animation: myfirst 5s; /\* Opera \*/

}

@keyframes myfirst

{

0% {background: red;}

25% {background: yellow;}

50% {background: blue;}

100% {background: green;}

}

@keyframes myfirst

{

from {background: red;}

to {background: yellow;}

}

可以设置from-to，也可以按100分比的阶段开设置阶段效果。