[HTML 1](#_Toc38459892)

[1. CSS三角形 1](#_Toc38459893)

[2. 类似QQ气泡的 5](#_Toc38459894)

[3. 清除浮动的方式 8](#_Toc38459895)

[4. 定位（定位类型和边偏移） 9](#_Toc38459896)

[5. 如何让绝对定位的盒子居中显示 10](#_Toc38459897)

[6. 定位模式转换 11](#_Toc38459898)

[7. 提高显示层级（比较有趣） 12](#_Toc38459899)

[8. Display和visibility 13](#_Toc38459900)

[9. 图片和文字是基线对齐的 14](#_Toc38459901)

[10. 外边线和侧边线需要删除的 15](#_Toc38459902)

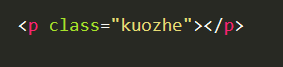
[11. 定义重置按钮 15](#_Toc38459903)

[12. 覆盖element-ui的本身样式的方法 16](#_Toc38459904)

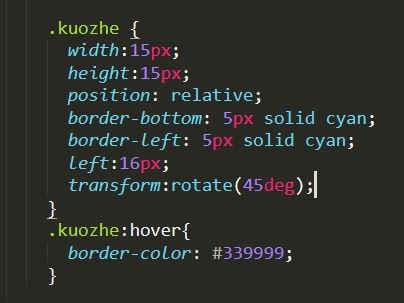
# HTML

## CSS三角形

HTML：



CSS：



.kuozhe {

width:15px;

height:15px;

position: relative;

border-bottom: 5px solid cyan;

border-left: 5px solid cyan;

left:16px;

transform:rotate(45deg);

border-radius: 2px;

}

.kuozhe:hover{

border-color: #339999;

}

效果：

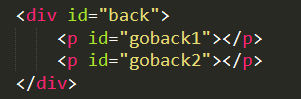


改进版：

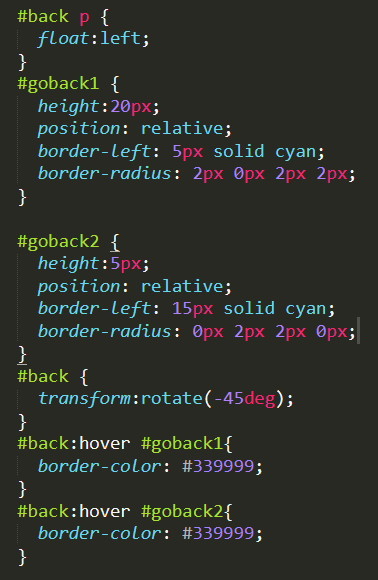


描述：四周都变成弧形了。

HTML：

、

CSS：



#back p {

float:left;

}

#goback1 {

height:20px;

position: relative;

border-left: 5px solid cyan;

border-radius: 2px 0px 2px 2px;

}

#goback2 {

height:5px;

position: relative;

border-left: 15px solid cyan; //不能用border-bottom，这个改变不了右上角的弧度

border-radius: 0px 2px 2px 0px;

}

#back {

transform:rotate(-45deg);

}

#back:hover #goback1{

border-color: #339999;

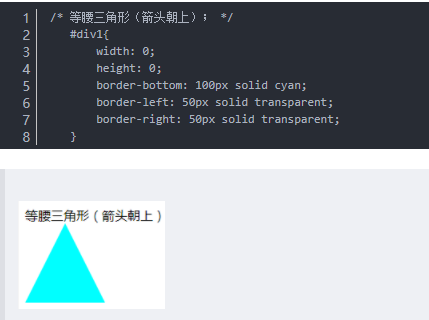
}

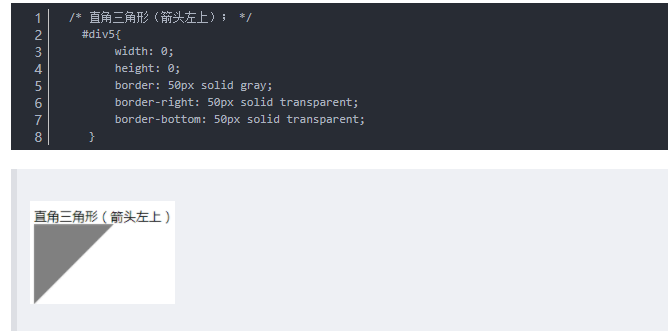
#back:hover #goback2{

border-color: #339999;

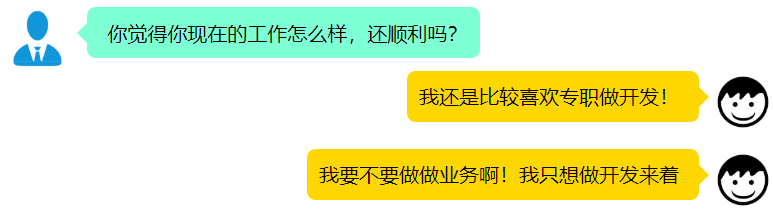
}

CSS：





## 类似QQ气泡的



参考：https://www.jb51.net/html5/562204.html

<html>

<head>

<style>

/\* bubble style \*/

.sender{

clear:both;

}

.sender div:nth-of-type(1){

float: left;

}

.sender div:nth-of-type(2){

background-color: aquamarine;

float: left;

margin: 0 20px 10px 15px;

padding: 10px 10px 10px 0px;

border-radius:7px;

}

.receiver div img,

.sender div img{

width:50px;

height: 50px;

}

.sender1{

clear:both;

}

.sender1 div:nth-of-type(1){

float: left;

}

.sender1 div:nth-of-type(2){

background-color: aquamarine;

float: left;

margin: 0 20px 10px 15px;

padding: 10px 10px 10px 0px;

border-radius:7px;

}

.sender1 div img{

width:50px;

height: 50px;

}

.receiver{

clear:both;

}

.receiver div:nth-child(1){

float: right;

}

.receiver div:nth-of-type(2){

float:right;

background-color: gold;

margin: 0 10px 10px 20px;

padding: 10px 0px 10px 10px;

border-radius:7px;

}

.left\_triangle{

height:0px;

width:0px;

border-width:8px;

border-style:solid;

border-color:transparent aquamarine transparent transparent;

position: relative;

left:-16px;

top:3px;

}

.right\_triangle{

height:0px;

width:0px;

border-width:8px;

border-style:solid;

border-color:transparent transparent transparent gold;

position: relative;

right:-16px;

top:3px;

}

</style>

</head>

<body>

<!-- Left -->

<div class="sender">

<div>

<img src="images/3.png">

</div>

<div>

<div class="left\_triangle"></div>

<span> 你觉得你现在的工作怎么样，还顺利吗？ </span>

</div>

</div>

<!-- Right -->

<div class="receiver">

<div>

<img src="images/1.png">

</div>

<div>

<div class="right\_triangle"></div>

<span> 我还是比较喜欢专职做开发！ </span>

</div>

<br/>

<br/>

<br/>

<div class="receiver">

<div>

<img src="images/1.png">

</div>

<div>

<div class="right\_triangle"></div>

<span> 我要不要做做业务啊！我只想做开发来着 </span>

</div>

</div>

</body>

</html>

## 清除浮动的方式

清除浮动的意义在于

1、通过添加额外的元素清除

<div class="logo clearfix"><!-- 左浮动 -->

<div>fesfe</div>

<div>fesfe</div>

<div class=”clear”>fesfe</div>

</div>

Css:

.clear {

Clear:both;

}

2、给父节点清除浮动

.logo {

Overflow:hidden;

}

3、使用伪元素清除浮动(可作为初始化定义使用)

.clearfix:after{

content: "";

display: block;

height： 0;

clear: both;

visibility: none;

}

IE7以下则要使用

.clearfix {

\*zoom:1;//清除浮动的方式，IE7、IE6才会执行

}

4、双伪元素清除浮动

.clearfix:before,.clearfix:after {

content: "";

display: table;

}

.clearfix:after {

clear: both;

}

.clearfix {

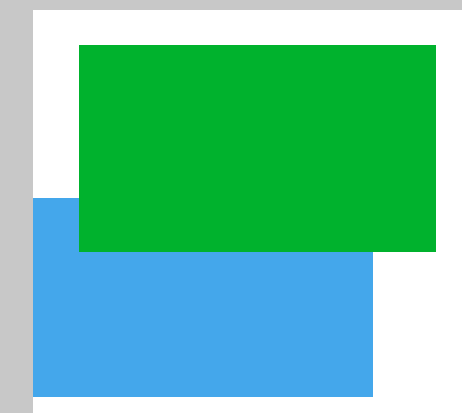
\*zoom: 1;

}

## 定位（定位类型和边偏移）

Static：可以使用它来取消定位，变为标准流定位，position：static

Relative：



和浮动有很大的差别，原来的位置还是保留的。

Absolute不占位置和浮动一样，基准点是当前的屏幕。浏览器为准。

1、不存在父元素的

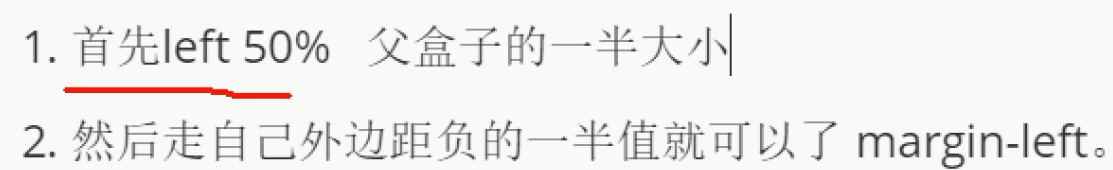


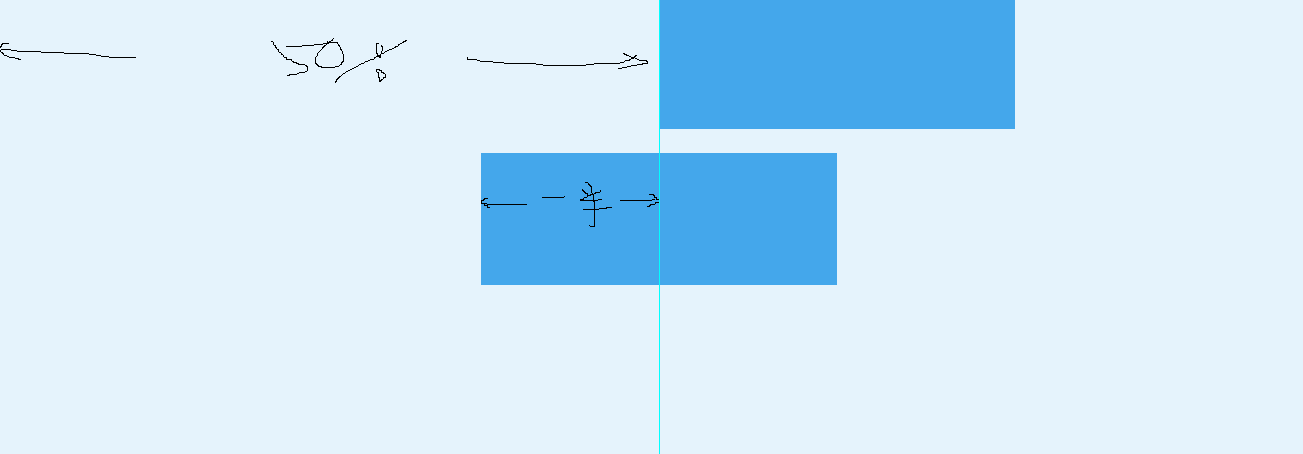
整个网页页面

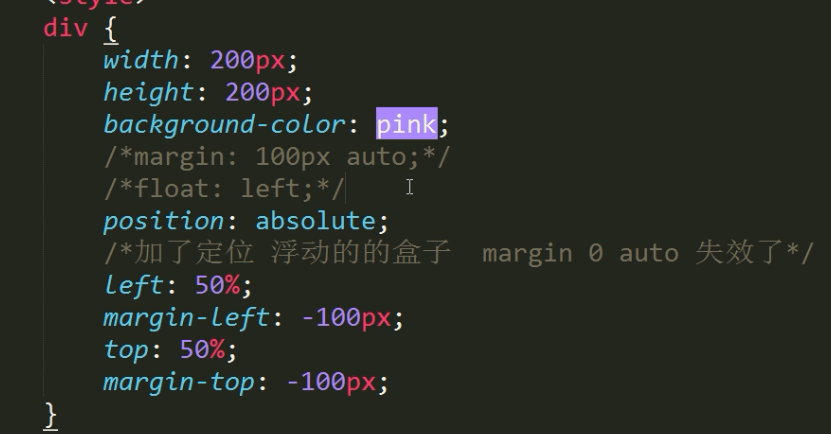
2、存在父元素但是父元素没有定位的（同1）

3、存在父元素且父元素有定位的，以最近的父元素定位自己

## 如何让绝对定位的盒子居中显示



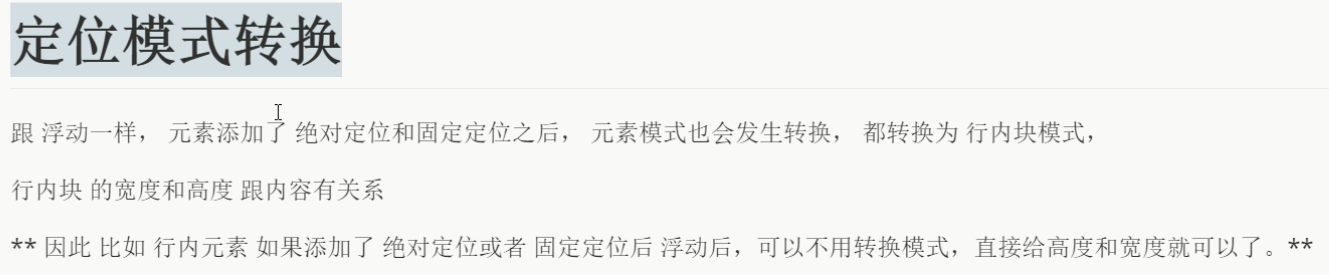




Fixed

只根据body定位，不管自己是否有父元素，父元素是否已经定位。

## 定位模式转换





写了FIXED就一定要给一个宽度，或者本身元素就具有一定的宽度。

## 提高显示层级（比较有趣）



目的：鼠标移动到块上的时候显示边框，但是边框的上下左右宽度为1px。

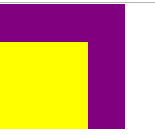
问题1、两个边框为1px的框重叠，重叠部分颜色会变深，所以达不到不改变边框。

解决办法：使用浮动并且left：-1px，覆盖掉1px边框。

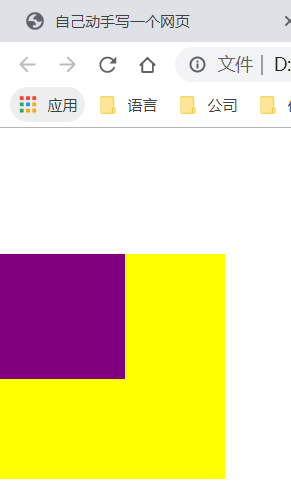
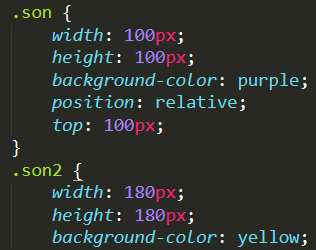
问题2、点击框过后中间的部分，没有显示右边框，因为被其右边的框给覆盖了，显示不出来

解决办法：使用相对定位，relative，因为定位会使得框的层级变高。

如：



这里的黄色在紫色之上。



紫色覆盖掉了黄色。

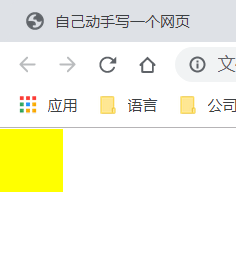
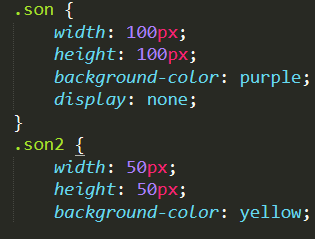
结论：定位会使得显示层级变高。

## Display和visibility

隐藏元素：display：none；

显示隐藏元素：display：block；

**注意：display：none；隐藏的元素，不保留位置；**

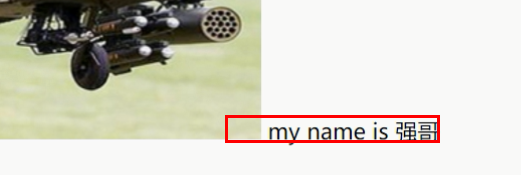


隐藏元素：visibility：hidden；

显示隐藏元素：visiblity：visible

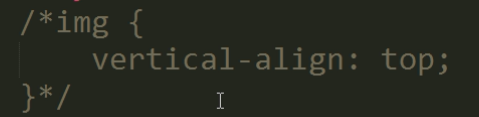
重点是：**visibility：hidden会保留位置。**

## 图片和文字是基线对齐的



所以会有一定的缝隙。

Vertical-align：middle；



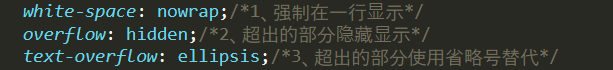
消除缝隙可以使用：

Vertical-align：bottom；

也可以使用：img{display:block;}转换为块元素消除，因为块元素占用一行，所以不会有缝隙。

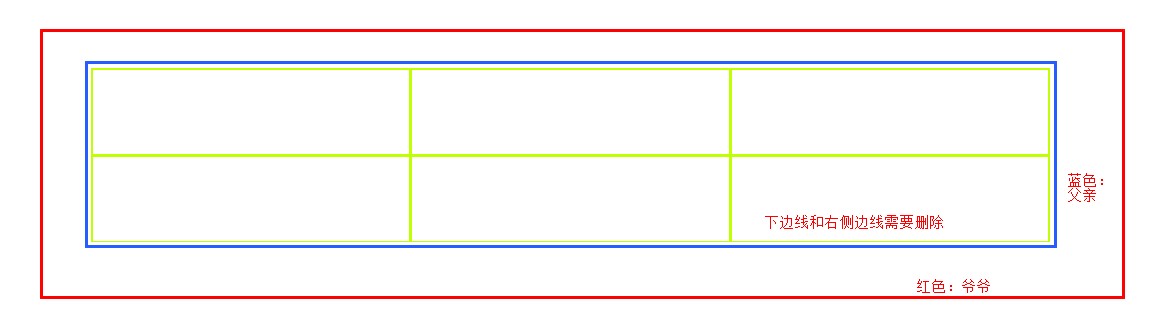
图片还有一种问题：会自带一个边框，所以一般在使用的时候都会加一个：**img{border：0；}**

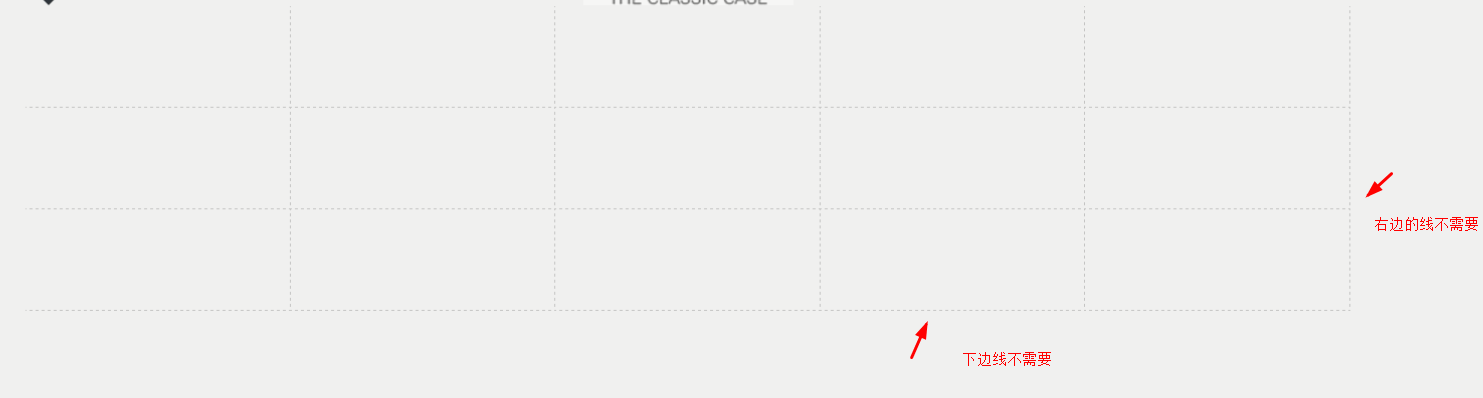
文字太长出现省略号的情况

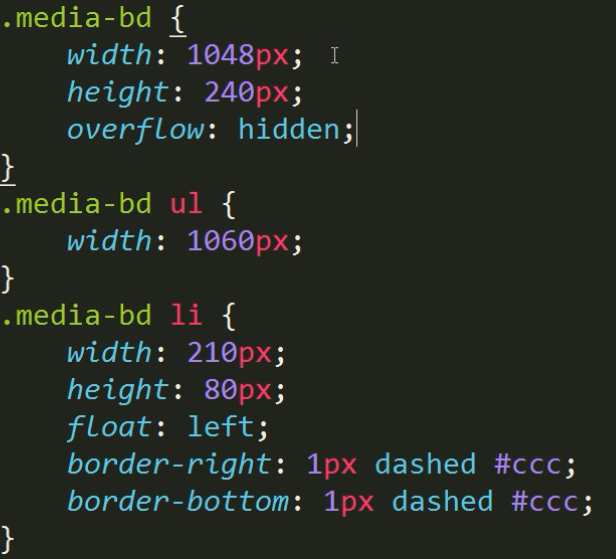


三行都要一起使用才行。

## 外边线和侧边线需要删除的







<ul>是父亲，父亲的大小可以装下li，但是爷爷media-bd不能装下li，

Hidden掉爷爷的外边部分，会使得li的下边框和右侧边框消失。

问题点：如果没有父亲的大小大于li的话，li会变有一行5个变成一行4个的，因为不够放，所以父亲很大，改用爷爷变小。

## 定义重置按钮

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

<form action="demo-form.php">

Email: <input type="text" name="email"><br>

Pin: <input type="text" name="pin" maxlength="4"><br>

<input type="reset" value="重置">

<input type="submit" value="提交">

</form>

<p>点击重置按钮重新设置表单。</p>

</body>

</html>

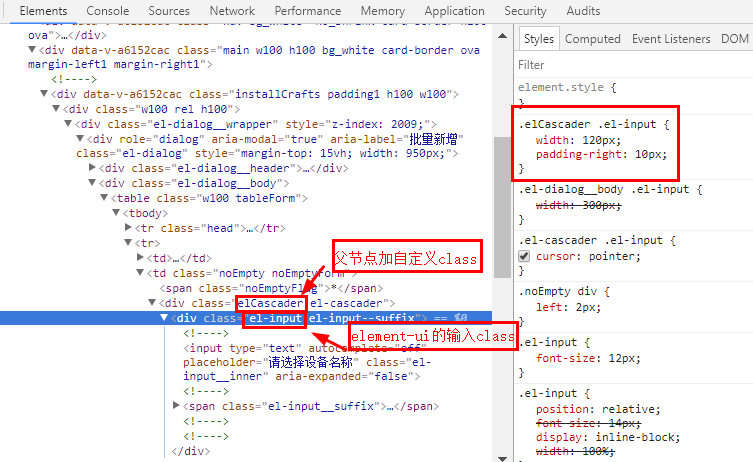


时间：20190708 <https://www.runoob.com/html/html-colors.html>

## 覆盖element-ui的本身样式的方法

1、设置给父节点设置class类名，避免覆盖全局的其他相同类型的组件

2、设置<style lang=”scss” >不要加scoped，加了scoped会被认为是私有的，覆盖不了全局设置



所有的style样式都可以使用这种方式覆盖。