## CSS相关

### 3. css3中box-shadow存在的问题

1. css3中box-shadow的溢出问题 子容器加了box-shadow在父容器中可能存在溢出 可以通过给父元素添加overflow:hidden解决
2. box-shadow阴影被覆盖问题
   1. 使用相对定位解决position:relative和z-index提高层数可以把阴影显示出来
   2. 开启gpu加速, 设置3d效果 { transform: translate3d(0, 0, 0); }

### 4. 完成左中右布局 左边固定200px 中间自适应右边固定100px 中间部分优先渲染 并解释原理

.box{  
 padding-left: 200px;  
 padding-right: 100px;  
 height: 100px;  
 position: relative;  
 box-sizing:border-box;  
}  
.middle{  
 height: 100px;  
}  
.left{  
 width: 200px;  
 height: 100px;  
 position: absolute;  
 left: 0;  
 top: 0;  
}  
.right{  
 width: 100px;  
 height: 100px;  
 position: absolute;  
 right: 0;  
 top: 0;  
}

<div class="box">  
 <div class="middle"></div>   
 <div class="left"></div>  
 <div class="right"></div>  
</div>

1. 中间部分优先渲染其实就是指把中间的盒子放到最前面写 html的元素是从上往下执行 所以写在越前面越优先渲染
2. 布局 思路就是把左右2遍固定 宽高 分别定位到左右2边 然后父元素相对定位 并且给父元素设置左右的padding 把中间的content内容区域往里面挤 如果父元素的宽度要固定 最好加上一个box-sizing:border-box;防止padding 撑大盒子范围

### 5. 使用CSS画一个扇形 并说出实现原理

.sector1 {  
 width: 100px;  
 height: 100px;  
 border-radius: 100px 0 0;  
 background: #ff0000;   
 /\* 原理是利用 边框圆角的属性设置盒子的某一个角为圆角 然后其他角设置为0 然后给整个盒子加背景色就只显示出一个扇形角 \*/  
}  
  
.sector2 {  
 width: 0;  
 height: 0;  
 border-radius: 100px;  
 border-width: 100px;  
 border-style: solid;  
 border-color: red transparent transparent transparent;  
 /\* 原理是利用边框颜色可以分别设置的属性 设置其中一条边框有颜色其他边框没有颜色 在把整个盒子变为圆角盒子没有宽高只显示出边框部分 突出扇形 \*/  
}

### 6. CSS选择器优先级

!important，加在样式属性值后，权重值为 10000  
  
  
内联样式，如：style=””，权重值为1000  
  
  
ID选择器，如：#content，权重值为100  
  
  
类，伪类和属性选择器，如： class="content"、:hover 权重值为10  
  
  
标签选择器和伪元素选择器，如：div、p、:before 权重值为1  
  
  
通用选择器（\*）、子选择器（>）、相邻选择器（+）、同胞选择器（~）、权重值为0  
  
继承（Inherited） 权重值最低级 只要有人任意其他的选择器都会覆盖(细节)

### 7. flex布局的常用api

采用 Flex 布局的元素，称为 Flex容器（flex container），简称"容器"。它的所有子元素自动成为容器成员，称为 Flex项目（flex item），简称“项目”。 1.父元素属性 | **属性名** | **属性值** | **备注** | display | flex | 定义了一个flex容器，它的直接子元素会接受这个flex环境 | | flex-direction | row, row-reverse, column, column-reverse | 决定主轴的方向  
| flex-wrap | nowrap, wrap, wrap-reverse | 如果一条轴线排不下，如何换行  
| flex-flow | [flex-direction] , [flex-wrap] | 是 flex-direction属性和 flex-wrap属性的简写形式， 默认值为 row nowrap | | justify-content | flex-start, flex-end, center, space-between, space-around | 设置或检索弹性盒子元素在主轴（横轴）方向上的对齐方式 | | align-items | flex-start, flex-end, center, baseline, stretch | 设置或检索弹性盒子元素在侧轴（纵轴）方向上的对齐方式 | | 2.子元素属性 | |  
| 属性名 | 属性值 | 备注  
| order | [int] | 默认情况下flex order会按照书写顺训呈现， 可以通过order属性改变， 数值小的在前面，还可以是负数。 | | flex-grow | [number] | 设置或检索弹性盒的扩展比率, 根据弹性盒子元素所设置的扩展因子作为比率来分配剩余空间 | | flex-shrink | [number] | 设置或检索弹性盒的收缩比率, 根据弹性盒子元素所设置的收缩因子作为比率来收缩空间 | | flex-basis | [length], auto | 设置或检索弹性盒伸缩基准值  
| align-self | auto,flex-start,flex-end,center,baseline,stretch | 设置或检索弹性盒子元素在侧轴（纵轴）方向上的对齐方式， 可以覆盖父容器align-items的设置 |

### 8. CSS中实现0.5px边框 解决安卓兼容性问题

/\*标准1px边框\*/  
 .b1{  
 height: 40px;  
 border: 1px solid #ff0000;  
 }  
 /\*1.可以利用利用渐变样式=>实现.5px\*/  
 .a1{  
 height: 1px;  
 margin-top: 20px;  
 background-image: linear-gradient(0deg, #f00 50%, transparent 50%);  
 }  
 /\*2.可以利用缩放-发虚=>实现.5px\*/  
 .a2{  
 height: 1px;  
 margin-top: 20px;  
 background-color: #f00;  
 -webkit-transform: scaleY(.5);  
 transform:scaleY(.5);  
 }  
 /\*3.四条边框都需要的样式\*/  
 .scale-half {  
 margin-top: 20px;  
 height: 100px;  
 border:1px solid #f00;  
 -webkit-transform-origin: 0 0;  
 transform-origin: 0 0;  
 -webkit-transform: scale(.5, .5);  
 transform: scale(.5, .5);  
 }  
 /\*4.给伪元素添加设置边框\*/  
 .border3{  
 position: relative;  
 }  
 .border3:before{  
 content: '';  
 position: absolute;  
 width: 200%;  
 height: 200%;  
 border: 1px solid blue;  
 -webkit-transform-origin: 0 0;  
 -moz-transform-origin: 0 0;  
 -ms-transform-origin: 0 0;  
 -o-transform-origin: 0 0;  
 transform-origin: 0 0;  
 -webkit-transform: scale(.5, .5);  
 -ms-transform: scale(.5, .5);  
 -o-transform: scale(.5, .5);  
 transform: scale(.5, .5);  
 -webkit-box-sizing: border-box;  
 -moz-box-sizing: border-box;  
 box-sizing: border-box;  
 }