

垂直居中总结

常用方法1：height与line-height

我是内部文字

适用于父元素高度一定

常用方法2：padding

我是内部文字
我是内部文字

适用于父元素高度不固定

常用方法3：margin

我是内部文字
我是内部文字

适用于父元素高度不固定

常用方法4：margin

我是内部文字
我是内部文字

父元素没有边框 垂直外边距合并问题

1: 子元素上移到父元素边框位置处；

2: 父元素与外面的包裹的div的margin-top由规定的0px, 变为50px;

常用方法5：padding

我是内部文字
我是内部文字

我是左侧文字 我是内部文字

多个元素高度不同、不固定，导致不能仅仅使用padding和margin处理

错误例子

错误1：vertical-align:middle不起作用

我是内部文字

父元素设置高度等于行高，子元素设置为vertical-align:middle;

错误2：大小固定元素的垂直居中布局



设置了display:table-cell的元素对宽度高度敏感，对margin值无反应,明明规定了box的高度，没有起作用

错误3：使用table-cell不起作用

内部文字1
内部文字2
内部文字3
内部文字3

右侧内容 >>

使用table-cell不起作用

◎ 香蕉 ◎ 苹果 ◎ 橘子

使用vertical-align,radio无论怎么设置总是不会与文字对齐

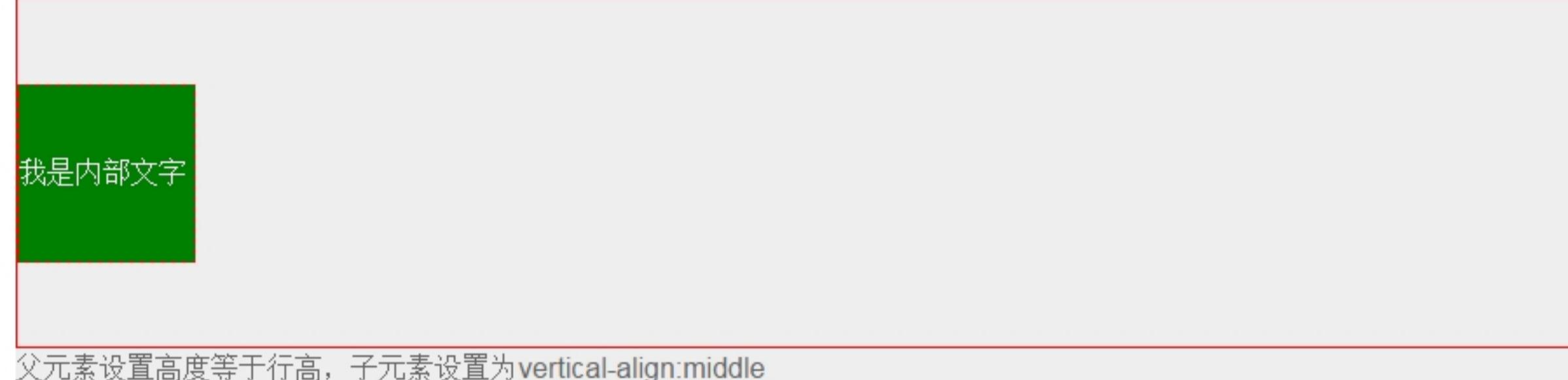
◎ 香蕉 ◎ 苹果 ◎ 橘子

vertical-align垂直居中总结

四种情况组合

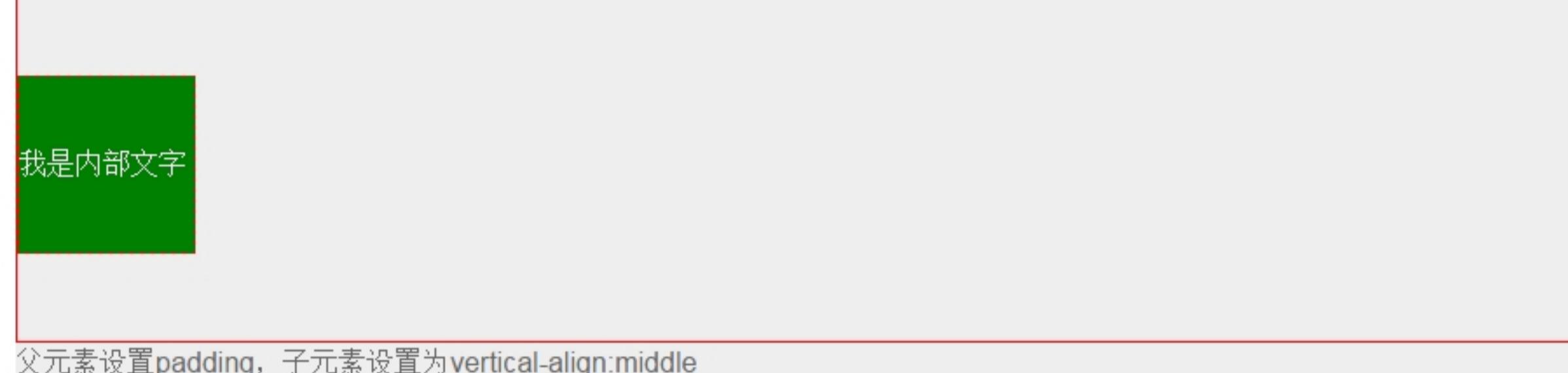


例子1:父元素有height, 没有设置padding (组合: 1-2)



父元素设置高度等于行高, 子元素设置为vertical-align:middle

例子2:父元素没有固定height, 设置padding (组合: 3-2)



父元素设置padding, 子元素设置为vertical-align:middle

例子3:多个固定高度子元素垂直居中 (组合: 1-2、3-2)



黄色区域高度固定,父元素高度(line-height)固定或者自适应(padding注释)

例子4:多个自适应高度子元素垂直居中组合: 1-4、3-4)



1.红框父元素高度不固定、内部元素高度不固定

2.父元素高度固定有行高时, 内部元素也要设置行高

例子5:vertical-align支持在有浮动的情况下垂直居中



float不会影响vertical-align (这种情况下一般不支持左右两列高度联动, 因此适应于高度固定情况)

例子6

○ 香蕉 ○ 苹果 ○ 橘子

父元素无固定高度时, 父元素设置padding(或不设置), span和input设置vertical-align和inline-block

总结: vertical-align

- 1.父元素有固定高度时, 父元素设置height、line-height, 同时子元素设置vertical-align和inline-block;
- 2.父元素无固定高度时, 父元素设置padding(或不设置), 同时子元素设置vertical-align和inline-block;
- 3.左右浮动布局, 不影响vertical-align布局;
- 4.自适应一般要加上padding属性扩展高度;
- 5.vertical-align: 支持IE8以上

上一页

下一页

tableCell垂直居中总结

四种情况组合

1、父元素固定高度	2、子元素固定高度
3、父元素自适应高度	4、子元素自适应高度

table-cell特点：

- 1:同行等高。
- 2:宽度自动调节。
- 3:table-cell，支持IE8以上

table-cell使用方法：

- 1:父元素设置display: table-cell;vertical-align: middle;
- 2:子元素设置display: inline-block;vertical-align: middle;。

方法1：大小不固定/固定单个子元素（父元素高度固定）的垂直居中布局（组合：1-2、1-4）



inner元素可以设置高度或者padding自适应高度

方法2：大小固定元素的垂直居中布局（父元素高度自适应）（组合：3-2）



table-cell：IE6、7不支持

方法3：大小不固定元素的垂直居中布局（父元素高度自适应）（组合：3-4）



table-cell：IE6、7不支持

方法4：使用table-cell两列自适应布局

壬戌之秋，七月既望，苏子与客泛舟游于赤壁之下。清风徐来，水波不兴。举酒属客，诵明月之诗，歌窈窕之章。少焉，月出于东山之上而和之。

我和左边等高

display:table-cell 元素生成的匿名table默认table-layout auto。宽度将基于单元格内容自动调整。所以设置width:1000px的用途是尽可能的宽的意思。这样就可以达到自适应的效果。

table-cell：IE6、7不支持

【手机】自用手机，交换100积分，或者交换同等价值的商品

简单说明一下，东西是我今天刚买的，因为不是太喜欢，现在进行转让，100积分，台通的米耐林27层找我拿货100积分进行转让。简单说明一下，东西是我今天刚买的，因为不是太喜欢，现在进行转让，100积分，台通的米耐林27层找我拿货100积分进行转让。简单说明一下，东西是我今天刚买的，因为不是太喜欢，现在进行转让，100积分，台通的米耐林27层找我拿货100积分进行转让。

发布时间：2016-07-07

方法5：使用table-cell多列自适应布局



中间这列高度自适应，其他两个随之联动。这个方法兼容性不错，但是table-cell：IE6、7不支持

北京职场交换

方法6：使用float浮动和table-cell布局



为了实现float下也能垂直居中，需要加一层标签，对比左侧，使用了table-cell的标签不能使用float属性

方法7：给label标签设置table-cell属性

- 香蕉
- 苹果
- 鸭梨

方法8：使用table-cell垂直居中,inner高度不固定

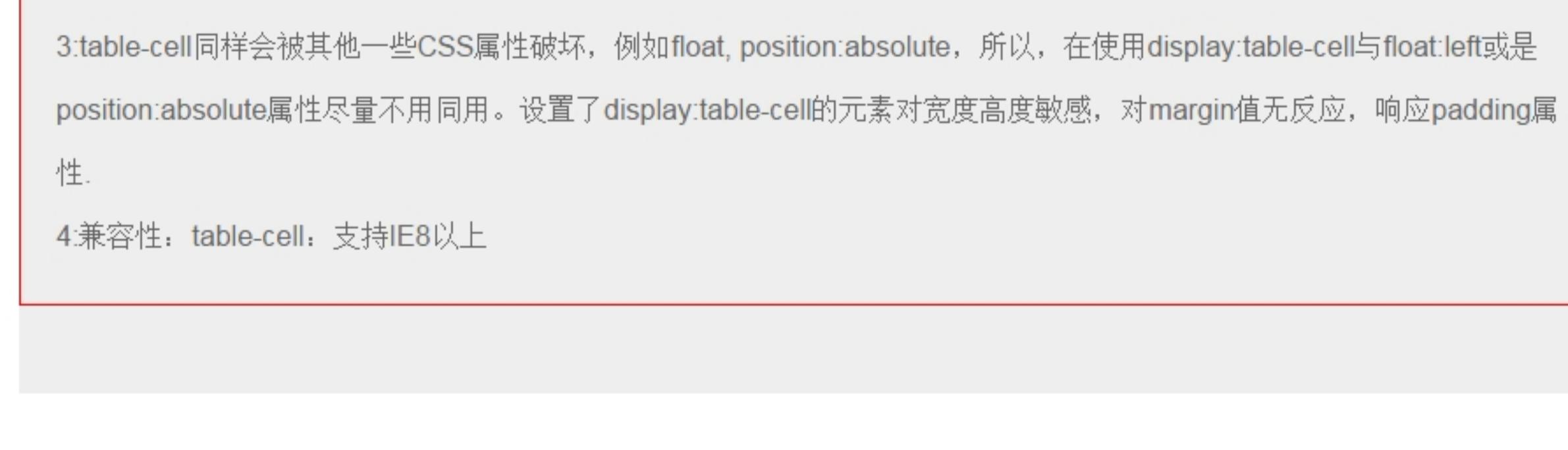


1.label标签包裹input和span标签，给其设置table-cell属性，内部元素居中对齐；

2.label标签的父元素高度不固定，设置padding撑开；

3.表单radiocheckbox都有默认的margin；

方法9：使用table-cell垂直居中,inner高度固定



外层高度固定，可以嵌套两层table-cell处理两层垂直居中

总结：vertical-align

1:父元素有固定高度/自适应时，子元素有固定高度/自适应；父元素设置display: table-cell;vertical-align: middle;，同时子元素设置vertical-align和inline-block；

2:可以根据同行等高、宽度自动调节来设置自适应多列布局

3:table-cell同样会被其他一些CSS属性破坏，例如float, position:absolute，所以，在使用display:table-cell与float:left或是position: absolute属性尽量不用同用。设置了display:table-cell的元素对宽度高度敏感，对margin值无反应，响应padding属性。

4:兼容性：table-cell，支持IE8以上

上一页

下一页

position垂直居中总结

四种情况组合

1、父元素固定高度	2、子元素固定高度
3、父元素自适应高度	4、子元素自适应高度

方法1：margin-top为自身的一半（组合：1-2）



设置要点：**margin-top:-xxpx**

1. 这个方法兼容性不错，但是有一个缺点：必须提前知道被居中块级元素的尺寸，否则无法准确实现垂直居中。

2. 子元素是绝对定位，导致脱离文档流，所以父元素不会随着子元素的高度变化而自适应高度。

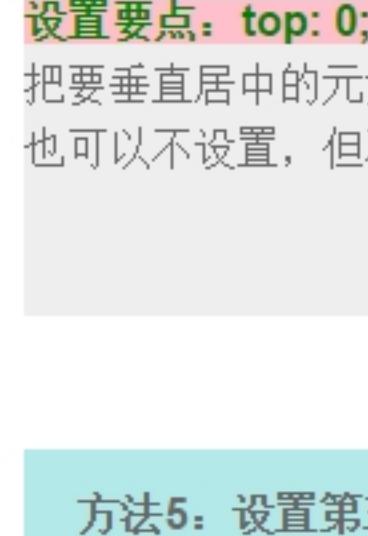
方法2：margin-top为百分比



设置要点：**margin-top:-xx%**

这种方式的原理实质上和上面相同。不同的是，margin的取值是百分比，这时这个值规定了该元素基于父元素尺寸的百分比。**思考：百分比如何算出来的？**

方法3：使用绝对定位和transform（组合：1-4）



设置要点：**transform: translate(0, -50%);**

这种方法有一个非常明显的好处就是不必提前知道被居中元素的尺寸了，因为transform中translate偏移的百分比就是相对于元素自身的尺寸而言的。

兼容IE8以上

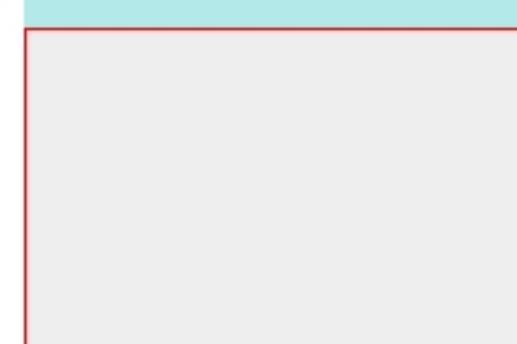
方法4：绝对定位结合margin: auto（组合：1-2）



设置要点：**top: 0; bottom: 0; left: 0; right: 0; margin: auto;**

把要垂直居中的元素相对于父元素绝对定位，top和bottom设为相等的值，再将要居中元素的margin设为auto，被居中元素的宽高也可以不设置，但不设置的话就必须是图片这种本身就包含尺寸的元素，否则无法实现。

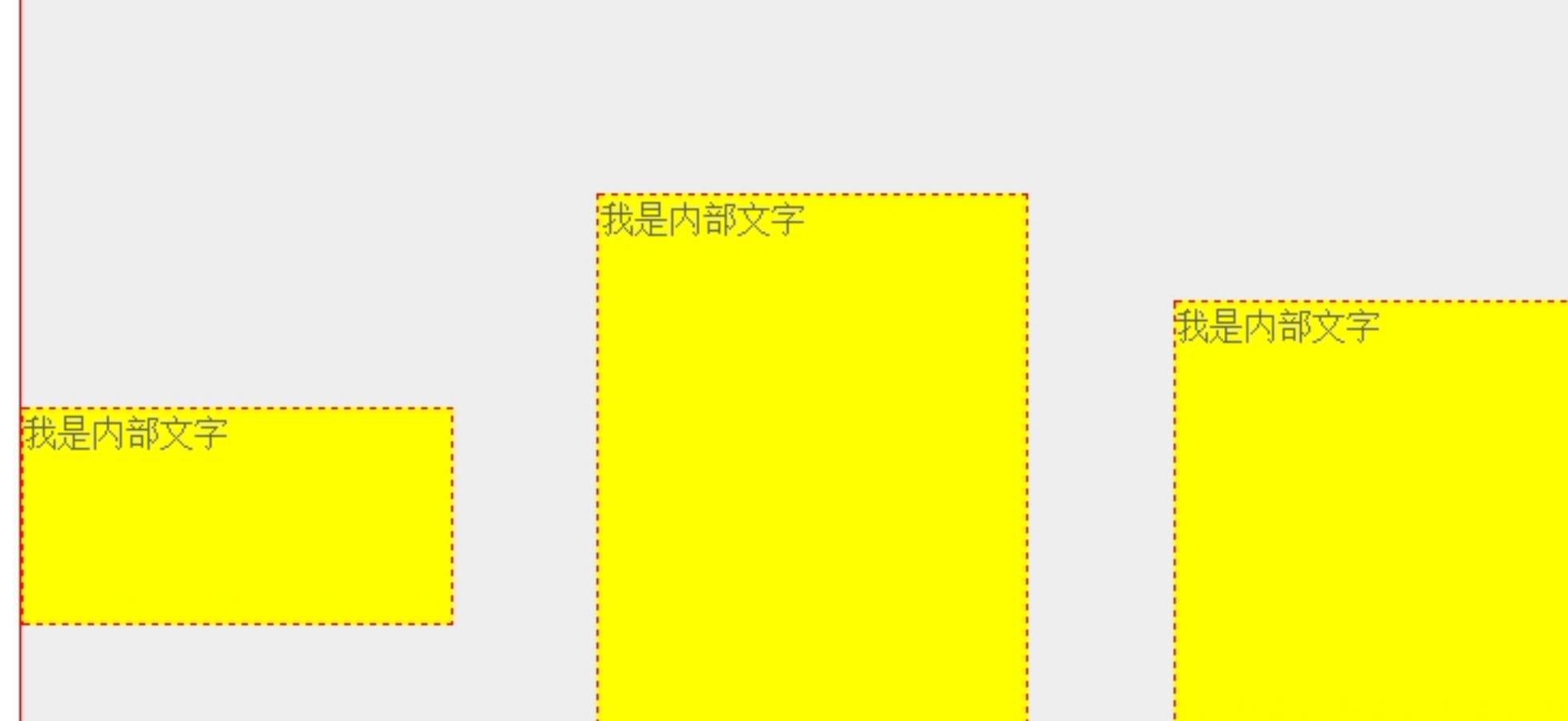
方法5：设置第三方基准（组合：1-2 1-4）



优点：不需要设置position首先设置一个高度等于父元素高度一半的第三方基准元素，那么此时该基准元素的底边线自然就是父元素纵向上的中分线，做完这些之后再给要垂直居中的元素设置一个margin-top，值的大小是它自身高度的一半取负，则实现垂直居中。

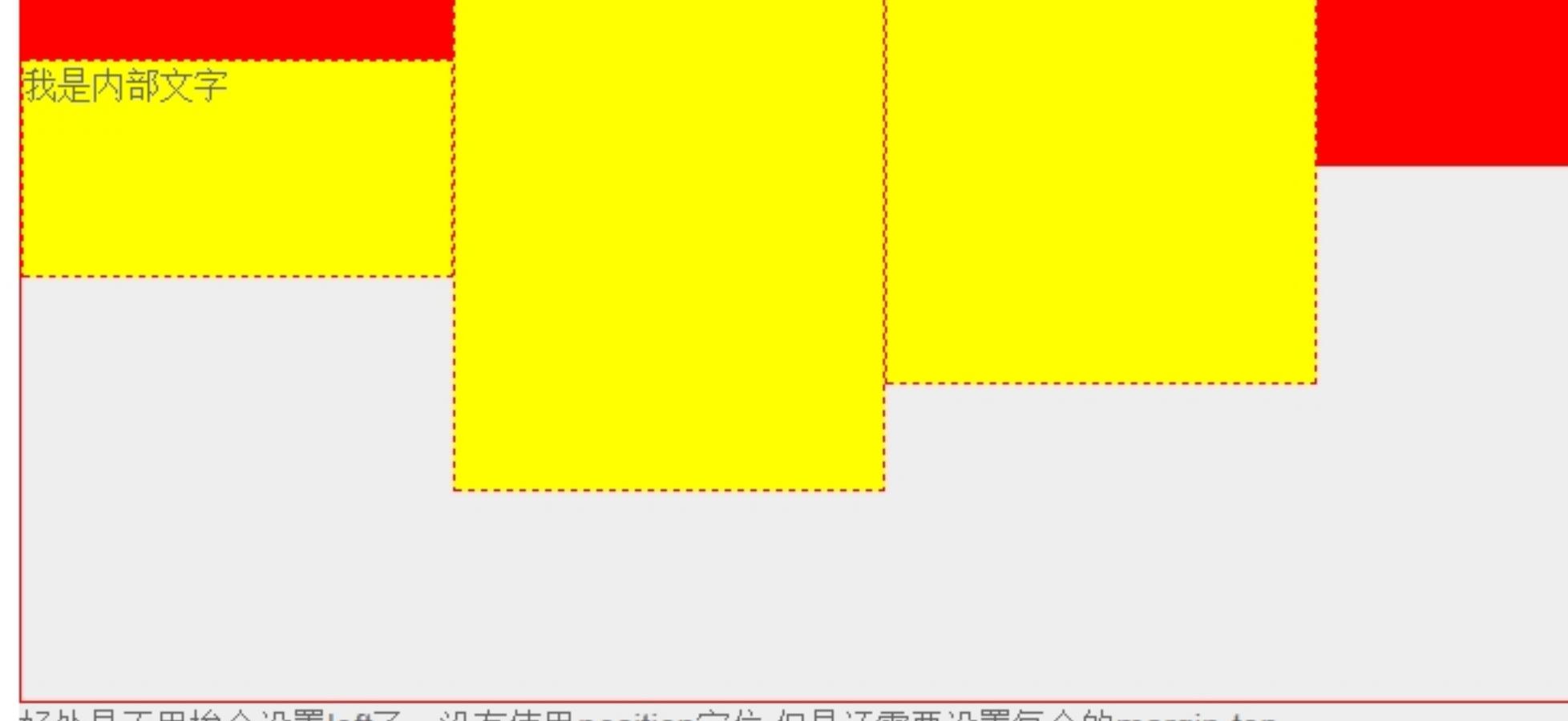
实战部分

使用方法1：实现多个不同高度的div垂直居中



使用方法1虽然可以实现，垂直居中，但是缺点是每个都要计算margin-top，不方便。

使用方法2：实现多个不同高度的div垂直居中



使用方法2明显比方法1简单，不用计算margin-top。

使用方法4：实现多个不同高度的div垂直居中



使用方法4和方法2区别不大，但是兼容性更好，支持IE8

使用transform: translate(0, -50%);



float浮动不影响translate

实战部分

如果使用position还需要定义每个left，不建议使用



此问题属于：自身高度不固定，但是要多个元素垂直居中，类似于使用多个div垂直居中，使用绝对定位过于复杂，需要给出每个label的left或者自身高度，因此不建议使用绝对定位来处理该情况。

总结：position

1. 由于元素position定位，所以会脱离文档流，因此父元素不能使用padding来自适应高度，父元素需有固定高度；

2. 一般相对定位/绝对定位，常用于父元素和子元素高度一定，且是单个元素垂直居中的情况；

3. float浮动不影响position

4. transform: translate兼容性不是很理想，IE8就不支持

5. 如果使用position还需要定义每个left，不建议使用

上一页

下一页

flex 垂直居中总结

四种情况组合

1、父元素固定高度	2、子元素固定高度
3、父元素自适应高度	4、子元素自适应高度

1. 使用flex布局，设置单个元素垂直居中（1234任意组合）

我是内部文字我要垂直居中

使用方法：父元素设置display:flex;（默认水平轴为主轴） align-items: center;(次轴方向居中) [flex布局链接](#)

2. 使用flex布局，设置多个元素垂直居中（1234任意组合）

我是内部文字我要垂直居中

3. 改变主轴方向

我是内部文字我要垂直居中

我是内部文字我要垂直居中

我是内部文字我要垂直居中

垂直方向为主轴，设置次轴也就是水平方向居中

4.flex设置两列高度固定，宽度自适应布局

左侧内容1 左侧内容1 左侧内容1	右侧内容>>
-------------------------	--------

5.flex设置宽度/高度...自适应布局

左侧内容1 左侧内容1 左侧内容1 左侧内容1	右侧内容>>
----------------------------------	--------

- 总结：

 - 1.父元素设置为display: flex;
 - 2.子元素设置flex-grow: 占空白位置的宽度比例
 - 3.父元素设置flex-direction为主轴方向
 - 4.父元素设置align-items次轴位置
 - 5.父元素设置justify-content主轴位置
 - 6.-webkit-box 是 -webkit-flex 的旧写法,正确的css覆盖顺序是
display: -webkit-box;
display: -webkit-flex;
display: -ms-flexbox;
display: flex;

1