**一、圣杯布局**

1.圣杯布局的DOM结构如下：

圣杯特点简单记为：有头、有尾、包三列，圣杯布局中间有container大容器包裹着左、中、右三列区域。

*<!--特点：有头、有尾、包三列(有个container容器包裹：左、中、右三列)-->*

<header>我是圣杯布局头部</header>

*<!--包裹middle、left、right的container容器-->*

<div class="container clearFloat">

<div class="middle">

我是中间区域<br>

</div>

<div class="left">我是左侧</div>

<div class="right">我是右侧</div>

</div>

<footer>我是圣杯布底部部</footer>

**2.圣杯布局思路：**

1. 被container包裹的：header、footer加：width:100%，height:30px（高度随意）见下图：

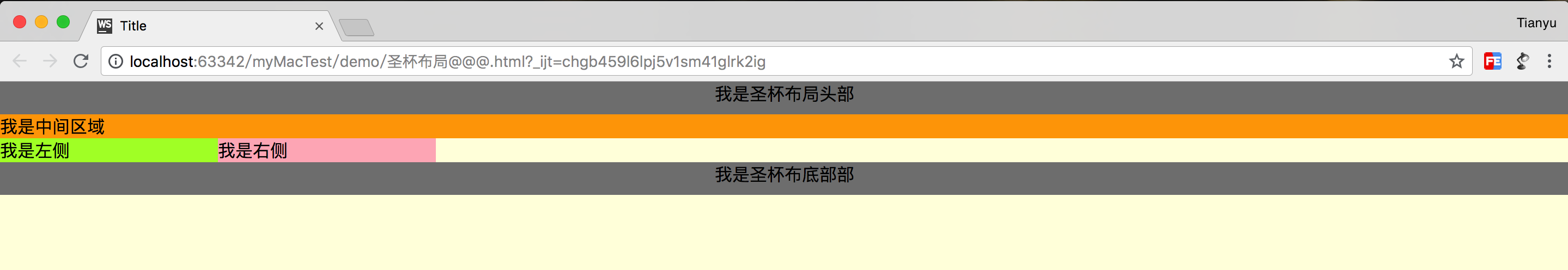


2. container包裹的：中间区域（middle）宽度设置100%；

然后middle，left均给200px；

middle，left，right均左浮动；

此时middle占了100%，宽度不够left，right被挤到下一行，见下图：



3. left设置margin-left:-100%，right设置margin-left:-200px；

此时left区域上去了，靠左，right上去了，靠右；

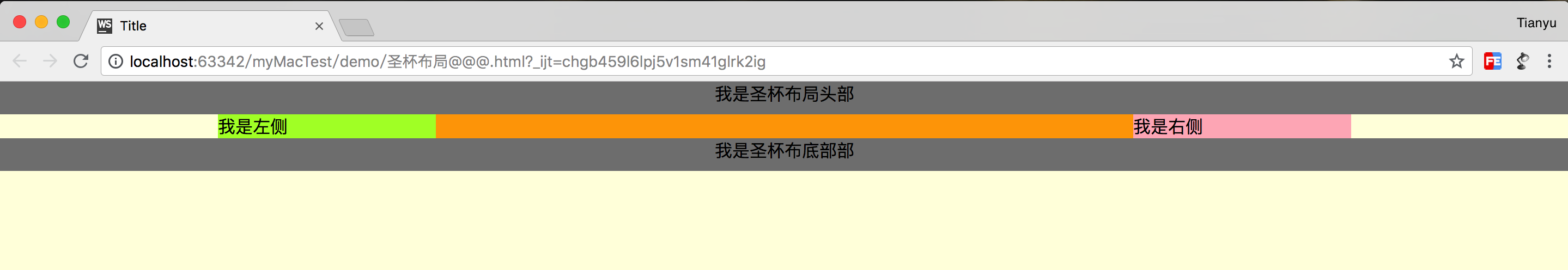
但是left遮挡住了middle的左侧部分，见下图。



4. 设置container的padding:200px；

让左右让出200px区域给left和right；

但此时left、right也跟着过去了。



5. 给left和right加relative相对定位；

然后left区域：left:-200px；然后right区域：right:200px；

为了美观可以给middle、left、right都加个高度；

中间正常显示，圣杯布局完成。



备注：

本文为了观看清晰，body加了一个浅黄色背景，header和footer加了文字居中等样式。

等高问题：

当左侧内容很多时，header、right区域会不与left等高，如下图：

解决办法：

利用magrin为负处理这个问题，给left、right、middle都加上padding-bottom:10000px，同时设置margin-bottom:10000px，将父容器container边缘推上去，同时父容器container设置overflow:hidden，达到如下效果：



**3.圣杯布局最终CSS样式：**

<style>

\*{

margin: 0;

padding: 0;

}

*/\*为便于观察body加了淡黄色背景\*/*

body{

background-color: #FFFFE0;

}

*/\*清除浮动\*/*

.clearFloat:after{

content: "";

display: block;

clear: both;

visibility: hidden;

height: 0;

}

.clearFloat{

zoom: 1;

}

header{

width: 100%;

height: 30px;

background-color: gray;

text-align: center;

}

footer{

width: 100%;

height: 30px;

background-color: gray;

text-align: center;

}

.container{

padding: 0 200px;*/\*给左右让空间\*/*

overflow: hidden;*/\*配合超大padding和超大负margin实现等高布局\*/*

}

.middle{

float: left;

width: 100%;

background-color: orange;

text-align: center;

padding-bottom: 10000px;*/\*为了实现等高\*/*

margin-bottom: -10000px;*/\*为了实现等高\*/*

}

.left{

float: left;

width: 200px;

background-color: greenyellow;

text-align: center;

margin-left: -100%;*/\*让left上去，靠左\*/*

position: relative;*/\*给left推到左边\*/*

left: -200px;*/\*给left推到左边\*/*

padding-bottom: 10000px;*/\*为了实现等高\*/*

margin-bottom: -10000px;*/\*为了实现等高\*/*

}

.right{

float: left;

width: 200px;

background-color: lightpink;

text-align: center;

margin-left: -200px;

left: 200px;*/\*给right推到右边\*/*

position: relative;*/\*给right推到右边\*/*

padding-bottom: 10000px;*/\*实现等高\*/*

margin-bottom: -10000px;*/\*实现等高\*/*

}

</style>

**二、双飞翼布局**

1.双飞翼布局DOM结构

<header>我是双飞翼头部区域</header>

<div id="middle">

<div id="inside">我是中间区域</div>

</div>

<div id="left">我是左侧</div>

<div id="right">我是右侧</div>

<footer>我是双飞翼底部区域</footer>

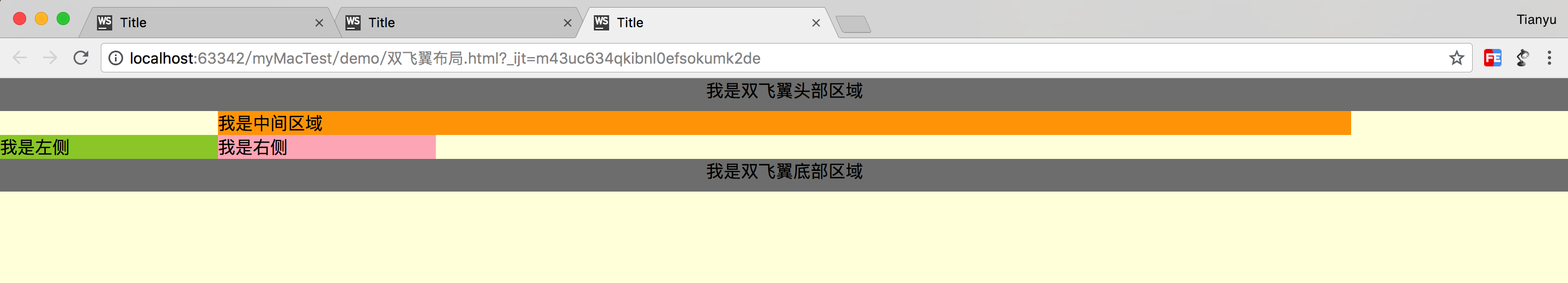
2.双飞翼布局思路（前三步与圣杯相同）

1. header、footer宽度100%，同圣杯布局第一步；

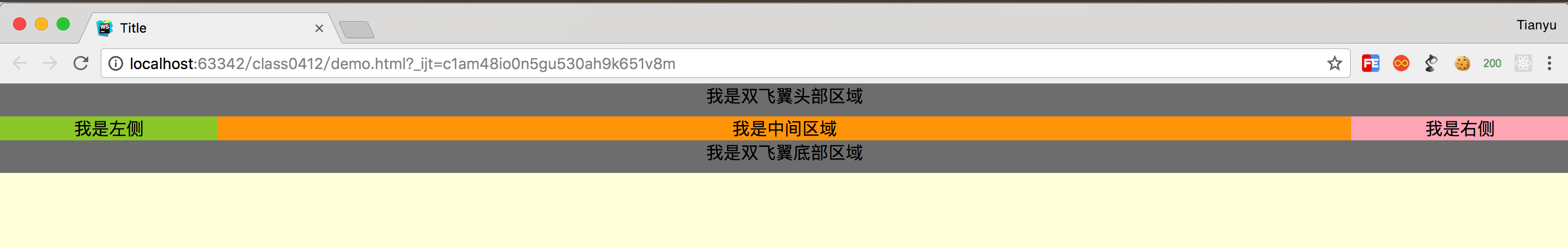
2.left、right、middle三个区域全左浮动，同圣杯布局第二步；

3.left、right设置margin-left，让左靠左，右靠右，此时left、right遮挡了middle区域同圣杯第三步；

4.将被middle包裹的inside设置margin:0 200px; 给左右留出空间；



5.给左侧设置：margin-left: -100%; 给右侧设置：margin-left: -200px; 让左靠左，右靠右，完成双飞翼布局。



**3.双飞翼布局最终CSS样式：**

<style>

\*{

margin: 0;

padding: 0;

}

body{

background-color: #FFFFE0;*/\*便于观看加的样式\*/*

}

header{

height:30px;

background: gray;

text-align: center;

}

footer{

clear: both;

height:30px;

background: gray;

text-align: center;

}

#middle{

float:left;

width:100%;

text-align: center;

}

#left{

float:left;

width:200px;

text-align: center;

margin-left: -100%;

background:yellowgreen;

}

#right{

float:left;

width:200px;

text-align: center;

margin-left: -200px;

background:lightpink;

}

*/\*给内部div添加margin，把内容放到中间栏，其实整个背景还是100%\*/*

#inside{

background-color: orange;

text-align: center;

margin:0 200px;

}

</style>

**三、圣杯和双飞翼布局的区别**

二者的区别就在：解决中间区域被遮挡问题的方式上。

**1.圣杯解决被遮盖的方式：**

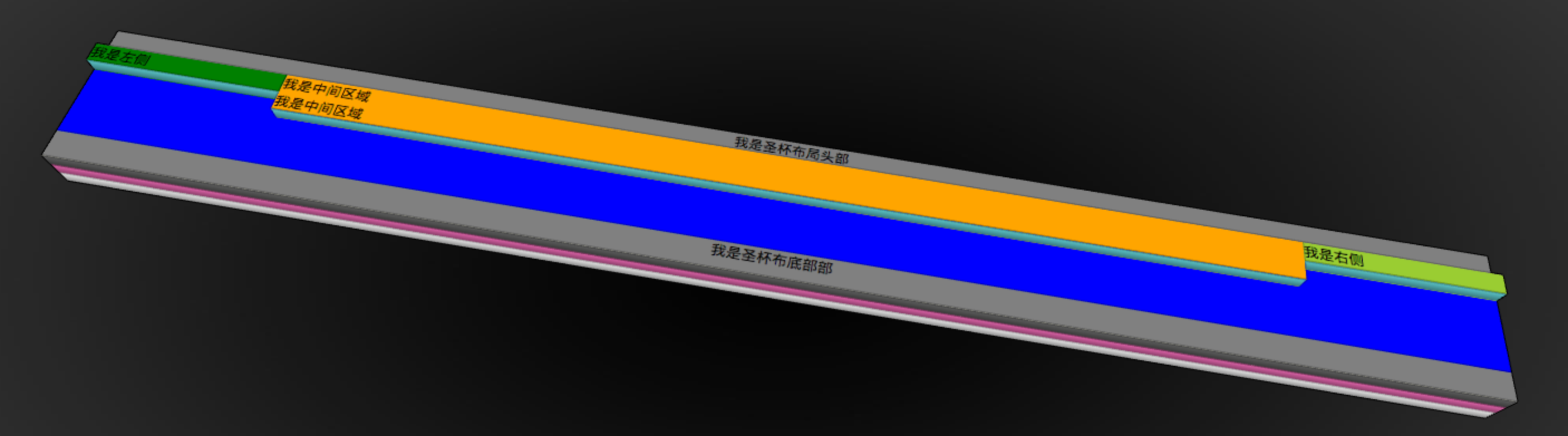
在最开始用容器包裹了：左、中、右，随后通过设置包裹容器container的padding，让中间区域左右留白，也就是说圣杯布局的左、中、右是完全独立的，他们之间是有缝隙的（如果缝隙可见的话），例如我们接下来：

1. 给container加一个蓝色背景色，并且设置高一些的高度。

2. 去左、中、右的等高布局，给middle再加一行文字。

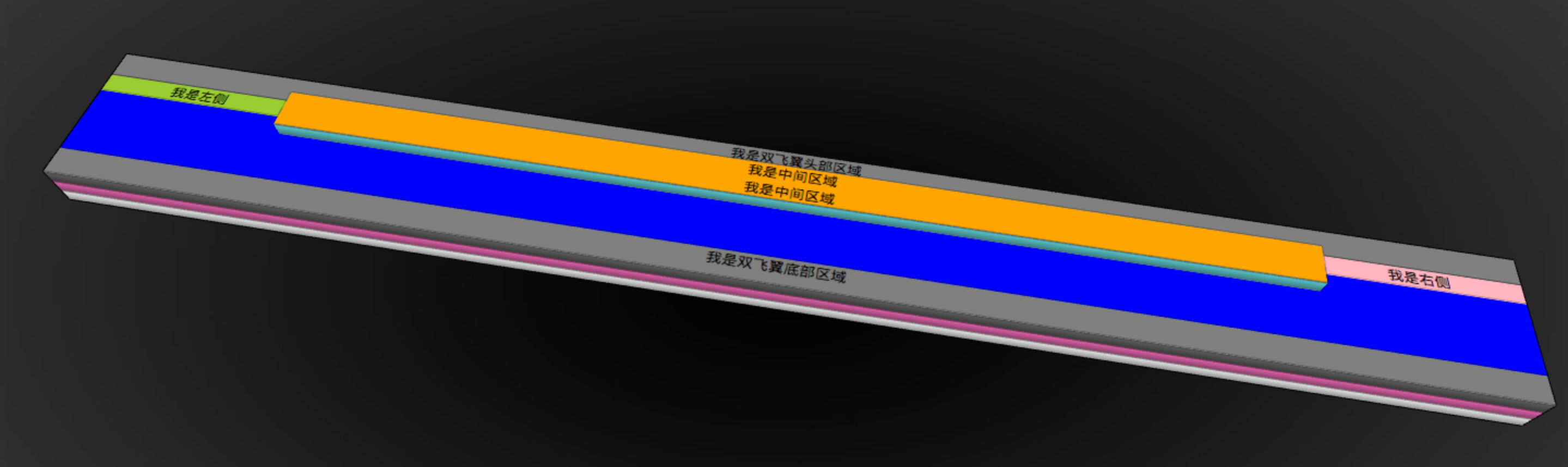
则从在空间立体视图上，左、中、右位置如下图：

可以看出，左、中、右是独立的三个区域，都处于一个层级，都由container这个容器（蓝色）承接，左右两侧是靠container的padding留出来的。



**2.双飞翼解决被遮盖的方式：**

双飞翼采用只处理中间解决遮盖问题，先给中间的包裹器middle加margin，让出左右空间，然后给left、right设置margin为负，把自己推上去，空间侧视图如下：



可以看出，左、中、右是独立的三个区域，中间区域属于最上面的层级（inner那个div），左右两侧和middle容器一个层级