圣杯布局、双飞翼布局、Flex布局

优劣比较

—— 完成三栏布局

## 实现过程

三栏布局，顾名思义为两边部分宽度固定，中间部分宽度随浏览器宽度自适应。最终完成效果：



实现中，分别采用了三种方法：圣杯布局、双飞翼布局、flex布局。现比较其优劣如下：

圣杯布局和双飞翼布局很像，但圣杯布局的HTML结构相对简单，样式定义较为复杂；双飞翼布局HTML布局稍显复杂，样式简单。相对而言，Flex布局在了解其原理和相关属性之后，较前两者更加简单实用，但不能兼容IE8及以下的浏览器。在实现三栏布局时，它们的共同点主要是中间栏都要在放在文档流前面以优先渲染。

圣杯布局和双飞翼布局解决问题的方案在前一半是相同的，也就是三栏全部float浮动，但左右两栏加上负margin让其跟中间栏div并排，以形成三栏布局。不同在于解决"中间栏div内容不被遮挡"问题的思路不一样：

1. 圣杯布局，为了中间div内容不被遮挡，将三栏的父级div设置了左右padding-left和padding-right后，将左右两个div用相对定位position:relative并分别配合right和left属性，以便左右两栏div移动后不遮挡中间div。
2. 双飞翼布局，为了中间div内容不被遮挡，直接在中间div内部创建子div用于放置内容，在该子div里用margin-left和margin-right为左右两栏div留出位置。多了1个div，少用大致4个css属性（圣杯布局中间div的padding-left和padding-right这2个属性，加上左右两个div用相对布局position:relative及对应的right和left共4个属性，一共6个：而双飞翼布局子div里用margin-left和margin-right共2个属性，6-2=4）。

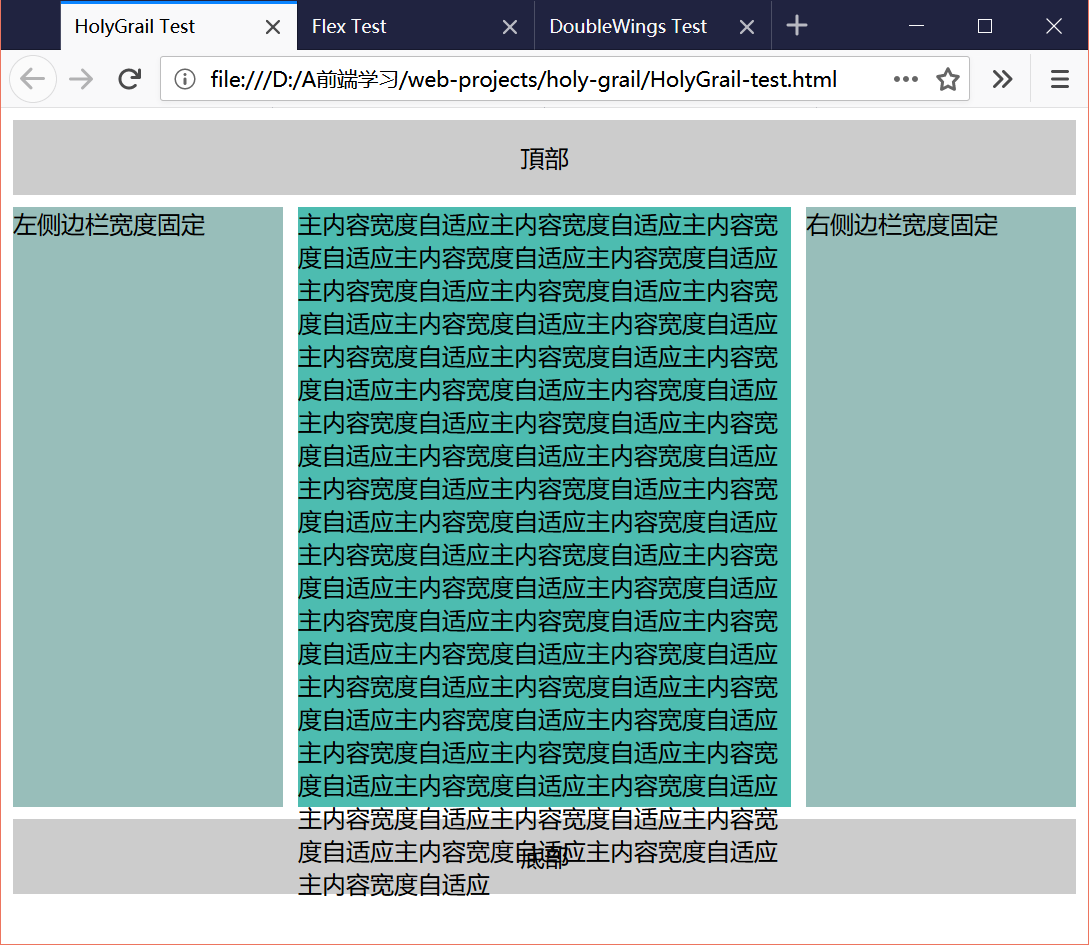
简单而言，就是双飞翼布局比圣杯布局多创建了一个div，但不采用相对位置了。

Flex布局将整个body部分设置为flex布局，且为纵向分布（顶部、中间三栏、底部）。然后令中部三栏区域自由伸缩并且justify-content设置为space-between令项目间隔相等（因未给main设置高度,main高度由子元素最高者决定）。最后横向区域中，设置flex的三个值使中间部分宽度自适应、左右两边固定宽度。

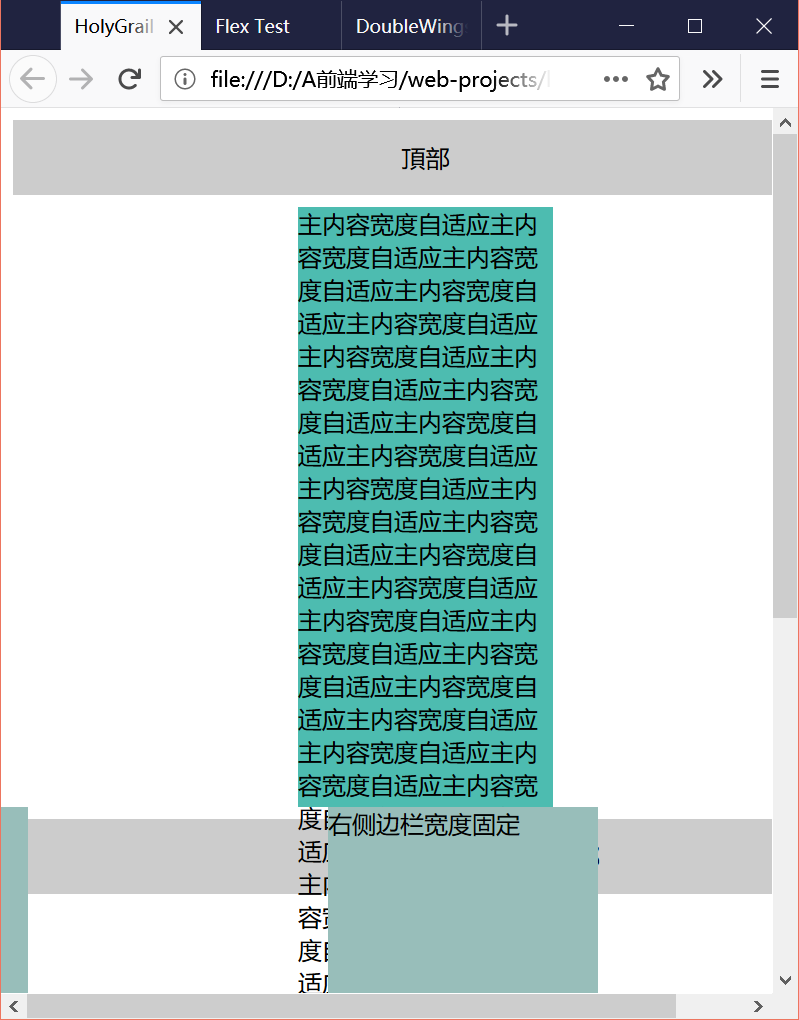
## 结果比较

圣杯布局：随着浏览器宽度减小，左右两栏与中间部分分离，最终显示十分混乱。



图一 圣杯布局1

图二 圣杯布局2



图三 圣杯布局3

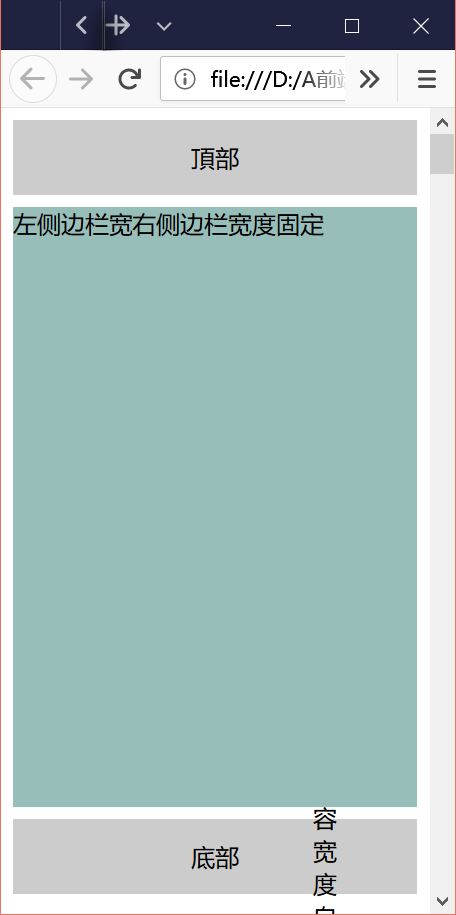
双飞翼布局：随着浏览器宽度减小，中间宽度逐渐减小至零，右侧栏可以覆盖左侧栏。



图四 双飞翼布局1



图五 双飞翼布局2

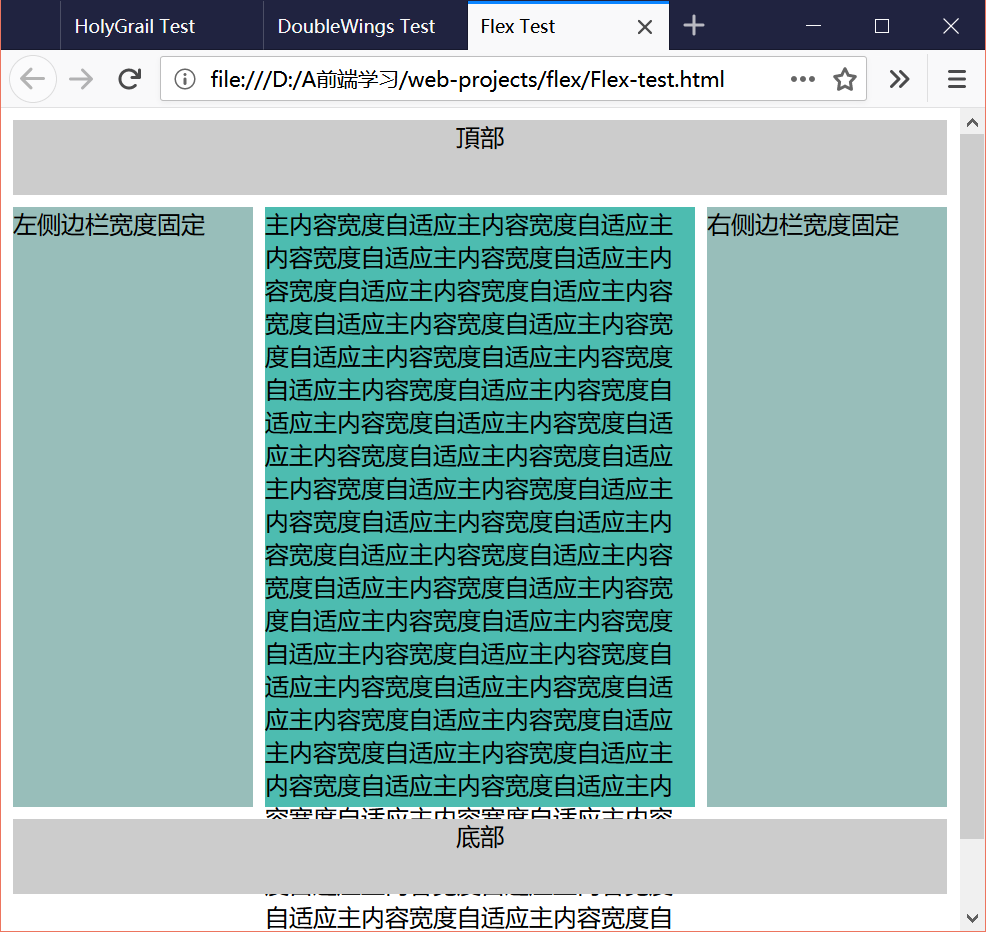


图六 双飞翼布局3

Flex布局：随着浏览器宽度减小，中间部分宽度也逐渐减小但最终仍然存在。



图七 Flex布局1



图八 Flex布局2



图九 Flex布局3