

编码三

利用浮动创建一个首页布局，css要求如下

- 示例图分为头部、底部、左侧，主体内容，右侧
- 头部和底部各自占领屏幕所有宽度。
- 上下部之间的部分是一个三栏布局,左右两栏宽度固定，中间一栏根据父元素宽度填满，最外面的框应理解为浏览器
- 中间三列区域的高度取决于三个子元素中最最高的高度，三列等高。
- 调节浏览器宽度，固定宽度和自适应宽度的效果始终符合预期
- 清除浮动尝试用多种方法，体会它们之间的区别



html如下:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Task3</title>
6     <link rel="stylesheet" href="task3.css">
7 </head>
8 <body>
9 <div class="header">
10 </div>
11 <div class="medium">
12     <div class=" inner m1"></div>
13     <div class=" inner m2"></div>
14     <div class=" inner m3"></div>
15 </div>
16 <div class="footer">
17 </div>
18 </body>
19 </html>
```

css如下

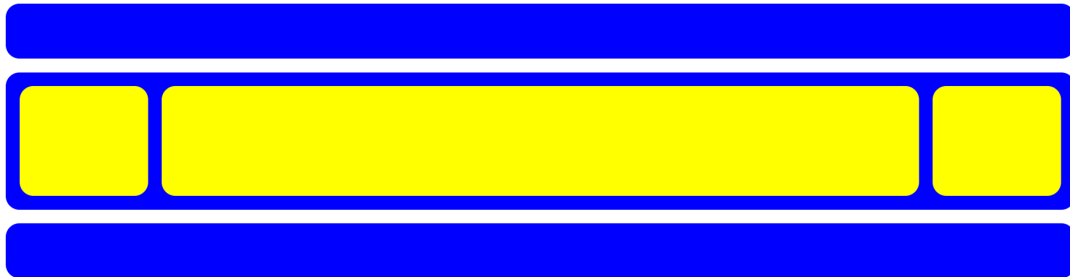
```
1  :root {
2      box-sizing: border-box;
3  }
4  *,
5  ::before,
6  ::after {
7      box-sizing: inherit;
8  }
9  .header {
10     width: 100%;
11     padding: 2em 0;
12     background-color: blue;
13     margin: 1em auto;
14     border-radius: 1em;
15 }
16 .medium {
17     width: 100%;
18     padding: 0;
19     background-color: blue;
20     margin: 0 auto;
21     border-radius: 1em;
22     /*新建BFC以清除浮动*/
23     overflow: hidden;
24 }
25 .footer {
26     width: 100%;
27     padding: 2em 0;
28     background-color: blue;
29     margin: 1em auto;
30     border-radius: 1em;
31 }
32 .inner {
33     background-color: yellow;
34     /*撑开容器*/
35     padding: 4em 0;
36     margin: 1em;
37     float: left;
38     border-radius: 1em;
39 }
40 .m1 {
41     width: 150px;
42     /*定宽*/
43 }
44 .m2 {
45     /*计算自适应容器宽度*/
46     width: calc(100% - 364px);
47     margin: 1em 0; /*取消外边距折叠*/
```

```

48 }
49 .m3 {
50     width:150px;
51     /*定宽*/
52 }

```

效果图如下:



另一种利用浮动布局的方法:

通过将左容器左浮动, 右容器右浮动, 左右容器不占据正常文档流位置。所以中间容器设定合适的外边距, 否则会重叠。该布局法的好处是受外界影响小, 但是不足是三个元素的顺序, center一定要放在最后, 这是和绝对定位不一样的地方, center占据文档流位置, 所以一定要放在最后, 左右两个元素位置没有关系。

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Task3</title>
6      <link rel="stylesheet" href="task3.css">
7  </head>
8  <body>
9  <div class="header">
10 </div>
11 <div class="medium">
12     <div class=" m1"></div>
13     <div class=" m3"></div>
14     <!--中间容器一定要在左右容器后定义, 否则会换行出错-->
15     <div class=" m2"></div>
16 </div>
17 <div class="footer">
18 </div>
19 </body>
20 </html>

```

CSS:

```

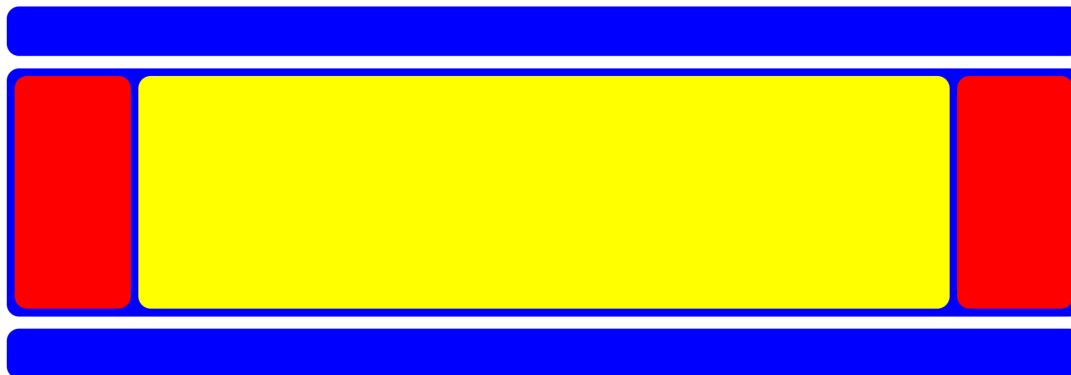
1  :root {
2      box-sizing: border-box;
3  }
4  *,
5  ::before,
6  ::after {
7      box-sizing: inherit;

```

```
8 }
9 .header {
10     width: 100%;
11     padding: 2em 0;
12     background-color: blue;
13     margin: 1em auto;
14     border-radius: 1em;
15 }
16 .medium {
17     width: 100%;
18     padding: 0;
19     background-color: blue;
20     margin: 0 auto;
21     border-radius: 1em;
22     /*新建BFC以清除浮动*/
23     overflow: hidden;
24 }
25 .footer {
26     width: 100%;
27     padding: 2em 0;
28     background-color: blue;
29     margin: 1em auto;
30     border-radius: 1em;
31 }
32
33 .m1 {
34     float: left;
35     margin: 10px;
36     width: 150px;
37     height: 300px;
38     /*定宽*/
39     background-color: red;
40     border-radius: 1em;
41 }
42 .m2 {
43     background-color: yellow;
44     height: 300px;
45     margin: 10px 170px;
46     border-radius: 1em;
47
48 }
49 .m3 {
50     float: right;
51     margin: 10px;
52     width: 150px;
53     /*定宽*/
54     background-color: red;
55     height: 300px;
56     border-radius: 1em;
```

```
57 }
58
```

效果：



- 清除浮动的方法：

1. 为达到清除内部浮动，我们可以触发浮动容器的父元素生成 BFC，那么父容器在计算高度时，父容器内部的浮动元素child也会参与计算。
2. 通过伪元素清除

```
1  :root {
2      box-sizing: border-box;
3  }
4
5  *,
6  ::before,
7  ::after {
8      box-sizing: inherit;
9  }
10 .header {
11     width: 100%;
12     padding: 2em 0;
13     background-color: blue;
14     margin: 1em auto;
15     border-radius: 1em;
16 }
17
18 .medium {
19     width: 100%;
20     padding: 0;
21     background-color: blue;
22     margin: 0 auto;
23     border-radius: 1em;
24     /*!*新建BFC以清除浮动*!*/
25     /*overflow: hidden;*/
26
27 }
28
29 .footer {
30     width: 100%;
```

```

31     padding: 2em 0;
32     background-color: blue;
33     margin: 1em auto;
34     border-radius: 1em;
35 }
36
37 .inner {
38     background-color: yellow;
39     /*撑开容器*/
40     padding: 4em 0;
41     margin: 1em;
42     float: left;
43     border-radius: 1em;
44 }
45
46 .m1 {
47     width: 150px;
48     /*定宽*/
49 }
50
51 .m2 {
52     /*计算自适应容器宽度*/
53     width: calc(100% - 364px);
54     margin: 1em 0; /*取消外边距折叠*/
55 }
56
57 .m3 {
58     width: 150px;
59     /*定宽*/
60 }
61
62 /*选中容器末尾的伪元素*/
63 .clearfix::after{
64     /*将伪元素的display设置为非inline，并给定一个content值，以便让伪元素出现在文档中*/
65     display: block;
66     content: " ";
67     clear: both;
68 }

```

3. 将一个带有clear属性的空div放置主容器的末尾（不建议使用）

拓展：

(29条消息) 前端经典面试题之CSS实现三栏布局，左右宽度固定，中间宽度自适应_ZXH2017的博客-CSDN博客_css三栏布局,左右固定宽度,中间自适应