

43、div水平居中方式：

1、margin: 0 auto 不能用于浮动元素

a、给予元素设置 margin: 0 auto 注意：居中块级元素首先要设置宽度，内联元素无法用这种方式居中。

```
<div style="width: 400px;background-color: red;height: 100px;">
  <div style="margin: 0 auto;background-color: green;width:
100px;">sss</div>
  <!-- <span style="background: green">sdfs</span> -->
</div>
```

b、table+margin: 0 auto 不能用于浮动元素

```
<div>
  <div style="display: table;margin: 0 auto">table+margin:0
auto</div>
</div>
兼容IE8以上
```

c、display:flex+margin: 0 auto: 浮动

```
<div style="display: flex;">
  <div style="margin: 0 auto">display:flex+margin: 0 auto</div>
</div>
```

2、div中的align:center 不能用于浮动元素

例如：<div align="center"></div>，div中的子元素居中

```
<div align="center" style="width: 400px;background-color: red;height:
100px;">
  <div style="background-color: green;width: 100px">sdfs</div>
</div>
```

3、margin+positon 浮动

```
<div style="width: 500px;height: 300px;margin: -150px 0 0
-250px;background-
color: pink;position: relative;top: 50%;left: 50%">
  hello
</div>
```

需要设置子元素宽度，百分比是相对于父元素

4、inline-block+text-align: 不能用于浮动元素

```
<div style="text-align:center;">
```

```
<div style="text-align: left;display: inline-block;">inline-block+text-align</div>
```

```
</div>
```

兼容性较好，在IE6、7中，通过display:inline;zoom:1来代替inline-block。

text-align会继承，所以需要在子元素中加上text-align:left防止字体或者里面的内容居中。

5、absolute + transform: 浮动

```
<div style="position: relative;">
```

```
  <div style="position: absolute;left: 50%;transform:
  translateX(-50%);">absolute +
  transform</div>
```

```
</div>
```

CSS3新增的，对IE678无法实现

6、flex + justify-content 浮动

```
<div style="display: flex;justify-content: center;">
```

```
  <div>flex + justify-content</div>
```

```
</div>
```

父元素设置了flex，里面的内容就为flex item，宽度为auto
对子元素设置flex:x(0) x(1) x，可以分配grow/shrink/basis
只需设置parent。

浮动元素居中：

1、margin+position

```
<div>
```

```
  <div style="float: left;position: relative;width: 300px;left:
  50%;margin-left:
  -150px;">居中浮动元素</div>
```

```
</div>
```

relative和absolute都可以，但这里使用relative更合适，以免对后续元素有影响

43、div垂直居中方式

1、table-cell + vertical-align:

```
<div style="height: 200px;background-color: red;display: table-  
cell;vertical-align: middle;">  
    <div style="height: 100px;background-color: green"> table-cell +  
vertical-align</div>  
</div>
```

利用了：在表单元格中，这个属性会设置单元格框中的单元格内容的对齐方式。

2、absolute + transform:

```
<div style="height: 200px;background-color: red;position: relative;">  
    <div style="height: 100px;background-color: green;position:  
absolute;top: 50%;transform: translateY(-50%);">absolute + transform:  
</div>  
</div>
```

绝对居中，子元素不会干扰其他元素。

缺点：兼容性不好

3、flex + align-items:

```
<div style="height: 200px;background-color: red;display: flex;align-  
items: center;">  
    <div style="height: 100px;background-color: green">flex + align-  
items</div>  
</div>
```

只需要设置parent。

兼容性。

4、vertical-align:middle:

```
<div style="height: 200px;background-color: red">  
    <span style="width: 80px;display: inline-block;vertical-align:  
middle;">  
        Hello Hello Hello Hello Hello Hello  
    </span>  
    <span style="height: 100%;display: inline-block;width: 0px;vertical-  
align: middle;background-color: green"></span><!--这是辅助元素-->  
</div>
```

44、水平垂直居中：

1、inline-block + text-align（水平） + table-cell + vertical-align（垂直）

兼容性高

2、absolute + transform

3、flex + justify-content（水平） + align-items（垂直）

4、flex + margin: auto：没有flex只能水平居中

```
<div style="height: 200px;background-color: red;display: flex;">
  <div style="height: 100px;background-color: green;margin:
auto">margin+flex</div>
</div>
```

其他几种有关于margin的方法不行。

64、页面布局的方式

1、经典三列 / 两列布局（双飞翼布局）：中间宽度自适应，两边定宽

a、float + margin：（两列和三列都可以）

```
<div>
  <div style="background-color: red;width: 100px;float: left;">
    <p>left</p>
  </div>
  <div style="background-color: green;float: right;width: 100px">
    <p>right</p>
    <p>right</p>
  </div>
  <div style="background-color: yellow;margin: 0 120px;">
    <p>++right</p>
  </div>
</div>
```

优点：容易理解

缺点：兼容性。因为一个浮动一个不浮动IE6里面会产生3px的bug，文字会往右缩3px。清

除浮动会产生bug（在right的第一个p上清除浮动可以模拟）

b、float + margin +（fix）：改造成兼容性好且没有清除浮动问题。（两列）


```

<div>
  <div style="float: left;width: 100px;background-color: red;position:
relative;">
    <p>left</p>
  </div>
  <div style="float: right;width: 100%;margin-left: -100px">
    <div style="margin-left: 120px;background-color: yellow">
      <p>right</p>
      <p>right</p>
    </div>
  </div>
</div>

```

为什么要margin-left: -100px, 因为right也float以后, 就不会跟left重合了

float以后, 默认元素宽度为内容宽度, 而div, 默认为容器宽度且可以通过设置margin来改

变大小

c、**float + overflow**: (两列和三列都可以)

```

<div>
  <div>
    <p style="float: left;width: 100px;background-color:
red;margin-right: 20px;">left</p>
  </div>
  <div style="float: right;width: 100px;background-color:
green;margin-left: 20px">
    <p>middle</p>
  </div>
  <div style="background-color: yellow;overflow: hidden;">
    <p>right</p>
    <p>right</p>
  </div>
</div>

```

d、**table**: (两列和三列都可以)

```

<div style="display: table;table-layout: fixed;width: 100%">

```

```

    <div style="background-color: red;display: table-cell;width:
100px;padding-right: 20px;background-clip: content-box;">
        <p>left</p>
    </div>
    <div style="background-color: green;display: table-cell;">
        <p>middle</p>
    </div>
    <div style="background-color: yellow;display: table-cell;width:
100px;padding-left: 20px;background-clip: content-box;">
        <p>right</p>
        <p>right</p>
    </div>
</div>

```

e、**flex**: (两列和三列都可以)

```

<div style="display: flex;">
    <div style="background-color: red;width: 100px;margin-right:
20px">
        <p>left</p>
    </div>
    <div style="background-color: green;flex: 1">
        <p>middle</p>
    </div>
    <div style="background-color: yellow;width: 100px;margin-left:
20px">
        <p>right</p>
        <p>right</p>
    </div>
</div>

```

2、两列 / 三列布局 不定宽 + 自适应

a、**float + overflow** (两列和三列都可以)

```

<div>
    <div style="float: left;margin-right: 20px;background-color: red">
        <p>left</p>
    </div>

```

```
<div style="float: right;margin-left: 20px;background-color: green;">
```

```
<p>middle</p>
```

```
</div>
```

```
<div style="overflow: hidden;background-color: yellow">
```

```
<p>right</p>
```

```
<p>right</p>
```

```
</div>
```

```
</div>
```

b、table:

```
<div style="display: table;width: 100%">
```

```
<div style="display: table-cell;background-color: red;padding-right: 20px;width: 0.1%;background-clip: content-box;">
```

```
<p>left</p>
```

```
</div>
```

```
<div style="display: table-cell;background-color: green;">
```

```
<p>middle</p>
```

```
</div>
```

```
<div style="display: table-cell;background-color: yellow;padding-left: 20px;width: 0.1%;background-clip: content-box;">
```

```
<p>right</p>
```

```
<p>right</p>
```

```
</div>
```

```
</div>
```

让内容决定宽度，所以要去掉table-layout: fixed

left的宽度设为足够小，（不用1px是因为在IE8里面会有问题）

c、flex:

```
<div style="display: flex;">
```

```
<div style="background-color: red;margin-right: 20px;">
```

```
<p>left</p>
```

```
</div>
```

```
<div style="background-color: green;flex: 1">
```

```
<p>middle</p>
```

```
</div>
```

```
<div style="background-color: yellow;margin-left: 20px;">
  <p>right</p>
  <p>right</p>
</div>
```

```
</div>
```

3、多栏布局

a、float:

<div style="margin-left: -20px"> (margin -20, 这个div的宽度就大于屏幕宽度了)

```
<div style="float: left;width: 25%;padding-left: 20px;box-sizing:
border-box;background: red content-box"><p>column1</p></div>
```

```
<div style="float: left;width: 25%;padding-left: 20px;box-sizing:
border-box;background: red content-box"><p>column2</p></div>
```

```
<div style="float: left;width: 25%;padding-left: 20px;box-sizing:
border-box;background: red content-box"><p>column3</p></div>
```

```
<div style="float: left;width: 25%;padding-left: 20px;box-sizing:
border-box;background: red content-box"><p>column4</p></div>
```

```
</div>
```

列数发生变化的时候需要改css样式，结构和样式有耦合性。

b、table:

```
<div style="margin-left: -20px">
```

```
<div style="display: table;width: 100%;">
```

```
<div style="display: table-cell;background: red content-
box;padding-left: 20px"><p>column1</p></div>
```

```
<div style="display: table-cell;background: red content-
box;padding-left: 20px"><p>column2</p></div>
```

```
<div style="display: table-cell;background: red content-
box;padding-left: 20px"><p>column3</p></div>
```

```
<div style="display: table-cell;background: red content-
box;padding-left: 20px"><p>column4</p></div>
```

```
</div>
```

```
</div>
```

外面多一层。

c、flex:


```

<div style="display: flex;">
  <div class="column" style="background: red content-box;flex: 1">
<p>column1</p></div>
    <div class="column" style="background: red content-box;flex:
1;margin-left: 20px"><p>column2</p></div>
    <div class="column" style="background: red content-box;flex:
1;margin-left: 20px"><p>column3</p></div>
    <div class="column" style="background: red content-box;flex:
1;margin-left: 20px"><p>column4</p></div>
  </div>

```

使用类的时候就是 .column+column{margin-left:20px;}

4、左右等高

a、table:

```

<div style="display: table;width: 100%;table-layout: fixed;">
  <div style="display: table-cell;background: red content-box;">
    <p>left</p>
  </div>
  <div style="display: table-cell;background: red content-
box;padding-left: 20px">
    <p>right</p>
    <p>right</p>
  </div>
</div>

```

b、flex:

```

<div style="display: flex;">
  <div style="background: red content-box;margin-right: 20px">
    <p>left</p>
  </div>
  <div style="background: red content-box;flex: 1">
    <p>right</p>
    <p>right</p>
  </div>
</div>

```

c、float

```
<div style="overflow: hidden;">
  <div style="padding-bottom: 9999px;margin-bottom:
-9999px;float: left;margin-right: 20px;background-color: red">
    <p>left</p>
  </div>
  <div style="padding-bottom: 9999px;margin-bottom:
-9999px;overflow: hidden;background-color: red">
    <p>right</p>
    <p>right</p>
  </div>
</div>
```

伪等高

parent当中的overflow:hidden的作用——隐藏超出浏览器底下的内容，也就是9999px

- 5、弹性布局 (Flexbox)
- 6、瀑布流布局
- 7、流式布局 (Fluid)
- 8、响应式布局