1.一列布局：底部是定宽。

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>布局</title>
    <style type="text/css">
    body{
        margin: 0;
        padding: 0;
    }
    .top{
        height: 100px;
        background-color: blue;
    }
    .main{
        width: 800px;
        height: 300px;
        background-color: #ccc;
        margin: 0 auto;
    }
    .foot{
        width: 800px;
        height: 100px;
        background-color: #900;
        margin: 0 auto;
    }
    </style>
</head>
<body>
    <div class="top"></div>
    <div class="main"></div>
    <div class="foot"></div>
</body>
</html>
```

Div{width:800px; height:500px; margin:0 auto},上面这段样式,可以让 div 在页面的：居中布局。

2.两列布局

要设置浮动。

如果要为自适应宽度的话，就要将宽度改为百分比。

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>布局</title>
    <style type="text/css">
        body{
            margin: 0;padding: 0;
        }
        .left{
            width: 20%;height: 500px;
            float: left;
            background-color: #ccc;
        }
        .right{
            width: 80%;height: 500px;
            float: right;
            background-color: #ddd;
        }
    </style>
</head>
<body>
    <div class="left"></div>
    <div class="right"></div>
</body>
</html>
```

固定宽度的两列布局：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>布局</title>
    <style type="text/css">
        body{
            margin: 0;padding: 0;
        }
        .main{
            width: 800px;margin: 0 auto;
```

```
        }
        .left{
            width: 220px;height: 500px;
            float: left;
            background-color: #ccc;
        }
        .right{
            width: 580px;height: 500px;
            float: right;
            background-color: #ddd;
        }
    </style>
</head>
<body>
    <div class="main">
        <div class="left"></div>
        <div class="right"></div>
    </div>
</body>
</html>
```

三列布局：

自适应的三列布局：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>布局</title>
    <style type="text/css">
        body{
            margin: 0;padding: 0;
        }

        .left{
            width: 33.33%;height: 500px;
            float: left;
            background-color: #ccc;
        }
        .middle{
            width: 33.33%;height: 500px;
            float: left;
            background-color: #999;
        }
        .right{
            width: 33.33%;height: 500px;
            float: right;
            background-color: #ddd;
        }
    </style>
</head>
<body>
        <div class="left"></div>
        <div class="middle"></div>
        <div class="right"></div>
</body>
</html>
```

特殊的三列布局:左右边为固定宽度，中间为自适应宽度。

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>布局</title>
```

```
    <style type="text/css">
        body{
            margin: 0;padding: 0;
        }

        .left{
            width: 200px;height: 500px;
            position: absolute; left: 0;top: 0; /*绝对定位，靠左边*/
            background-color: #ccc;
        }
        .middle{
            height: 500px;
            background-color: #999;
            margin: 0 300px 0 200px;/*左边 200px 就是留给左边的 div，右边 300px 就是留给右边的列*/
        }
        .right{
            width: 300px;height: 500px;
            background-color: #ddd;
            position: absolute; right: 0;top: 0; /*绝对定位，靠右边*/

        }
    </style>
</head>
<body>
        <div class="left">200px</div>
        <div class="middle">通用选择器是功能最强大的选择器，它使用一个（*）号指定，它的作用是匹配 html 中所有标签元素，如下使用下面代码使用 html 中任意标签元素字体颜色全部设置为红色：</div>
        <div class="right">300px</div>
</body>
</html>
```

若是左边的为自适应、中间和右边的为固定宽度。则为：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>布局</title>
    <style type="text/css">
        body{
            margin: 0;padding: 0;
        }
```

```
        .left{
            height: 500px;
            margin: 0 500px 0 0;/*左边 200px 就是留给左边的 div，右边 300px 就是留给右边的列*/
            background-color: #ccc;
        }
        .middle{
            width: 200px;
            height: 500px;
            background-color: #999;
            position: absolute; right: 300px;top: 0;
        }
        .right{
            width: 300px;height: 500px;
            background-color: #ddd;
            position: absolute; right: 0;top: 0;
        }
    </style>
</head>
<body>
        <div class="left">200px</div>
        <div class="middle">通用选择器是功能最强大的选择器，它使用一个（*）号指定，它的作用是匹配 html 中所有标签元素，如下使用下面代码使用 html 中任意标签元素字体颜色全部设置为红色：</div>
        <div class="right">300px</div>
</body>
</html>
```

混合布局：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>布局</title>
    <style type="text/css">
    body{
        margin: 0;
        padding: 0;
    }
    .top{
        height: 100px;
        background-color: blue;
    }

    .head{
        width: 800px;
        height: 100px;
        background-color: #f60;
        margin: 0 auto;
    }
    .main{
        width: 800px;
        height: 300px;
        background-color: #ccc;
        margin: 0 auto;
    }
    .left{
        width: 200px;
        height: 300px;
        background-color: yellow;
        float: left;
    }
    .right{
        width: 600px;
        height: 300px;
        background-color: pink;
        float: right;
    }

    .sub_l{
        width: 400px;
        height: 300px;
        background-color: green;
```

```
            float: left;
        }

        .sub_r{
            width: 200px;
            height: 300px;
            background-color: #bbb;
            float: right;
        }
        .foot{
            width: 800px;
            height: 100px;
            background-color: #900;
            margin: 0 auto;
        }
        </style>
    </head>
    <body>
        <div class="top">
            <div class="head">
            </div>
        </div>
        <div class="main">
            <div class="left">
            </div>
            <div class="right">
                <div class="sub_l">
                </div>
                <div class="sub_r">
                </div>
            </div>
        </div>
        <div class="foot"></div>
    </body>
    </html>
```

网页布局的三种状态：块挨着块、块嵌套着块、块叠压着块。

布局的自适应可以为：不设置宽度（在只有一列时），设置的宽度为百分比（为多列时，但是这是一种比较不严格的方法），还有一种在多列时，其中一列就是不设置宽度，设置 Margin 值，然后其他固定宽度的列设置为绝对定位。在多列时，当有一列自适应，其他列固定宽度的话，我们可以设置固定宽度的列左右浮动，然后自适应的列设置为绝对定位，然后对其 margin 值进行改正。

1.使用 margin:0 auto；使得 div 居中；
2.使用 float 为 left 和 right 使得两个 div 处于同一行；
3.使用百分比宽度实现宽度自适应，使用绝对值使得宽度固定；
4.使用 position:absolute 实现 div 绝对定位，在三列布局中使得最左和最右绝对定义，中间可自适应；
5.混合布局可使用嵌套的方式，在横向 header、main 和 footer，然后在 main 中使用两列或三列布局。
6.宽度自适应，左侧固定宽度右侧自适应，父盒子

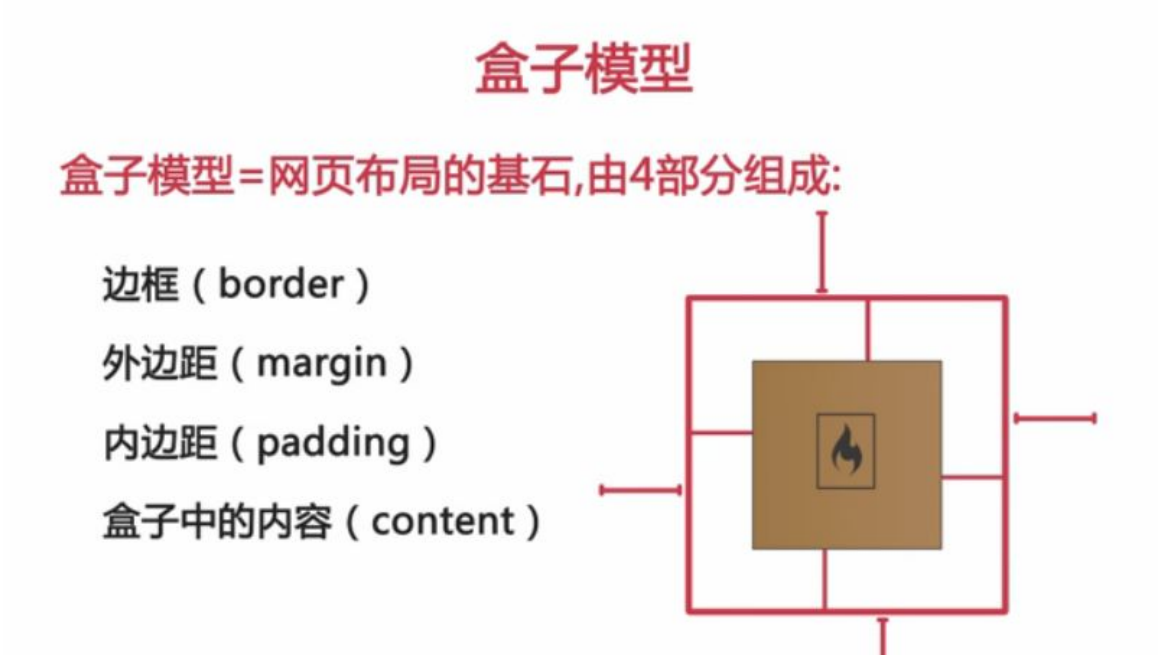
position:relative;左侧子盒子 position:absolute;右侧 margin-left。

网页布局基础：div+css

布局分为：流式布局、浮动布局、绝对定位布局
CSS 中网页布局的分类：
自动居中--列布局-------盒子模型使用方法
浮动布局-------float 属性解决浮动影响
绝对定位布局-------绝对定位实现横向两列或多列布局

css 3 种定位机制：
1.标准文档流：从上到下，从左到右、输出文档内容。由块级元素(从左到右独占一行、自动换行：div、ul、li、p)和行级元素（能在同一行内显示、不会改变 html 的文件结构:span、em、strong、input、img）组成。它们都是盒子模型。
2.浮动
3.绝对定位。

盒子模型：



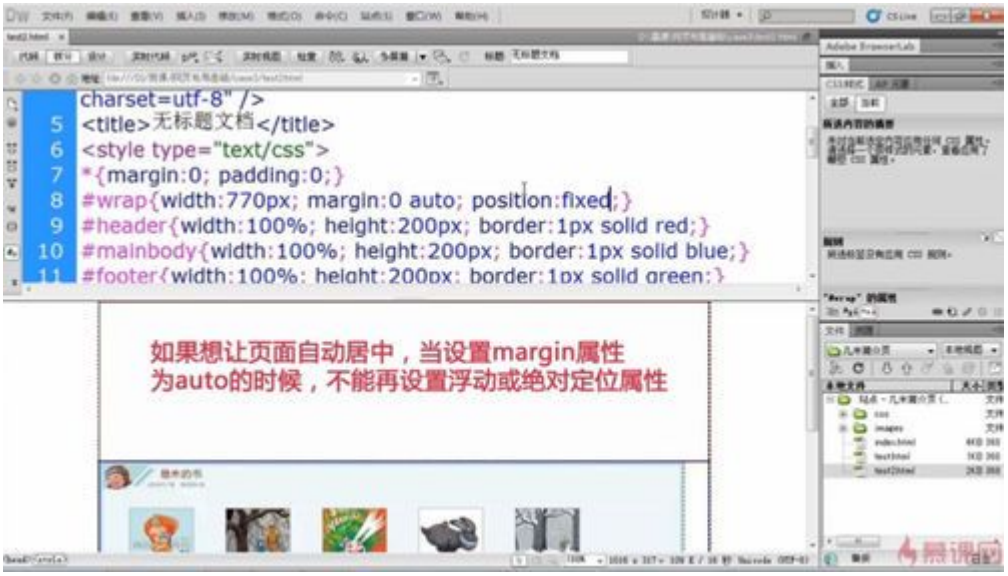

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>盒子模型</title>
    <style type="text/css">
    *{
        margin: 0;
        padding: 0;
    } /*清除浏览器自动的添加样式*/
    .content{
        border:4px solid #badbdb;
        padding: 50px 15px 15px;

    }
    .book{
        background-image:url(4.jpg);
        background-repeat:no-repeat;
        /*这两句等价于 background:url(4.jpg) no-repeat;*/
    }
    img{
        margin: 10px 18px;
        border: 1px solid #b1adad;/*三个属性值不能缺少任何一个*/
    }
    </style>
</head>
```

```
<body>
    <div class="content book">
        <img src="1.jpeg" width="400px" height="400px" />
        <img src="2.jpeg" width="400px" height="400px" />
        <img src="3.jpg" width="400px" height="400px" />
    </div>
</body>
</html>
```

自动居中一列布局

标准文档流，块级元素，margin 属性。

一般有三层，第一层是头部，第二层是主体部，第三层是尾部。



设置自动居中：margin{0 auto} auto 会根据浏览器的宽度自动设置两边的外边距，其原理为：（浏览器的宽度-外包含层宽度）/2=外边距。当要设置自动居中的层设置成浮动或者是绝对定位时，自动居中失效。
自动居中一列布局需要设置 margin 左右值设置为 auto，而且一定要设置 width 为一个定值。

浮动布局简介及 float 属性

float 属性：left right none。特点：元素会左移、右移直到触碰到容器位置。并且还处于标准文档流中。占据了标准文档流的空间，对文档流中的元素有影响
当元素没有设置宽度，设置了浮动特性时，元素的宽度会随着内容的变化而变化。
当元素设置浮动属性后，会对相邻的元素产生影响，相邻元素特指近邻后面的元素。

清除元素的常用方法

1.clear 属性——常用 clear:both(clear:left 或者 clear:right)
2.设置 width100%（或固定宽度）+overflow：hidden；
3.空标签
可以清除浮动，但是毫无意义。不建议使用

横向两列布局

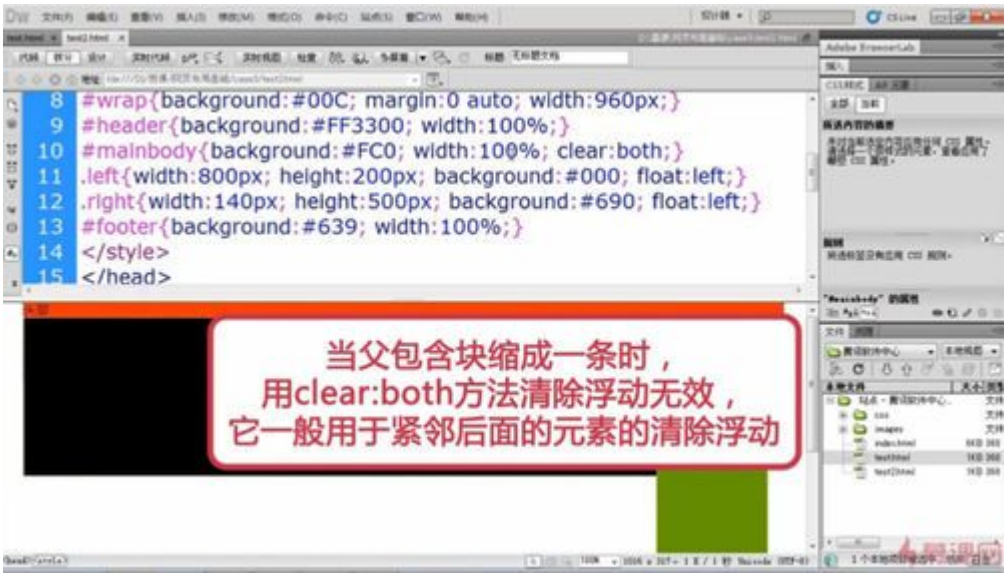
主要应用技能:float 属性—使纵向排列的块级元素，横向排列
margin 属性—设置两列之间的间距

1、DIV 尽量不设置高度，因为不设置高度，整个 DIV 的高度可以随内容增加或减少而增大或缩小， 如果是 DIV 父级，又设置了高度，如果想清除浮动，设置的高度会影响清除浮动;
2、当父包含块缩成一条时，用 clear：both 方法清除浮动无效，它一般

用于紧邻后面的元素的清除浮动,清除父级浮动

3、横向二列布局中，两个 DIV 横向平铺，DIV 之间的间隔，尽量用 float 来解决;

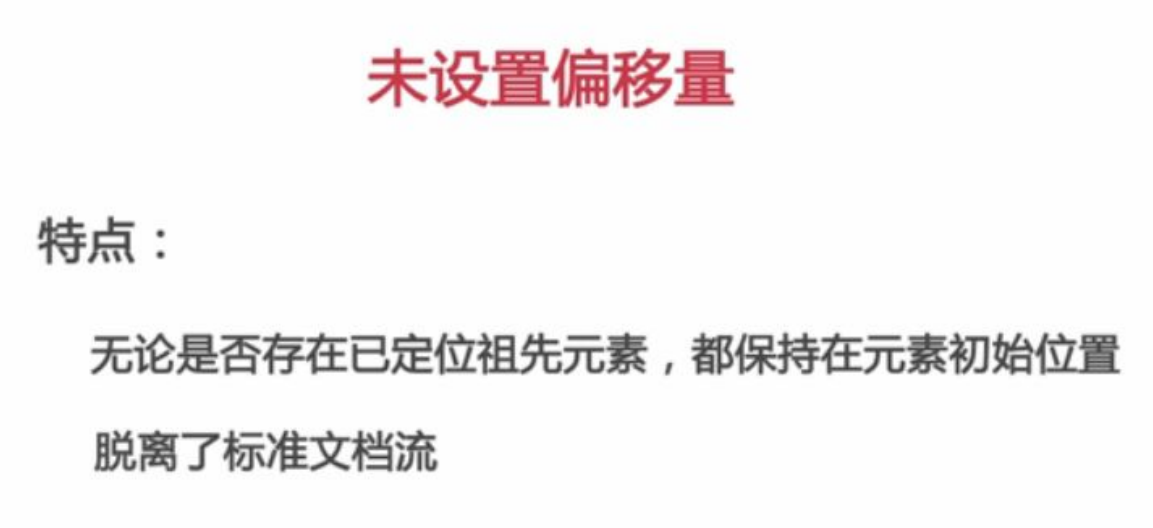
4、横向三列布局中，三个 DIV 横向平铺，中间的 DIV 对两边之间的间隔，需用 margin 来解决;



绝对定位布局

相对定位的特点：1.相对于自身原有位置进行偏移
2.仍处于标准文档流中
3.随机拥有偏移属性和 z-index 属性。

绝对定位：
绝对定位的特点是：1.建立了以包含块为基准的定位，
2.完全脱离了标准文档流
3.随机拥有偏移属性和 z-index 属性。



一个元素没有设置宽度但是设置了绝对定位，则其宽度会随着内容的改变而改变。

当一个绝对定位的元素需要有参照物的基准时,他的父级元素最好设置成相对定位，因为相对定位比较稳定。

设置偏移量

偏移参照基准:

无已定位祖先元素，以<html>为偏移参照基准

有已定位祖先元素，以距其最近的已定位祖先元素为偏移参照基准

横向两列布局

使用absolute实现横向两列布局
—常用于一列固定宽度，另一列宽度自适应的情况

主要应用技能:

relative—父元素相对定位

absolute—自适应宽度元素绝对定位

注意：固定宽度列的高度>自适应宽度的列

①display:inline:任何不是块级元素的可见元素都是内联元素。其表现的特性是“行布局”形式!(行布局:其表现形式始终以行进行显示)
②float:left:指定元素脱离普通的文档流而产生的特别的布局特性。并且 float 必需应用在块级元素之上，也就是说浮动并不应用于内联标签。或者可以说如果应用了 float 这个元素将被指定为块级元素。
那么两者的区别显而易见:display:inline 元素不能设置宽高，因为它属于行布局，其特性是在一行里进行布局，所以不能被设定宽高。
简单的说就是 float 可以设置宽高，然而 display:inline;虽然也会浮动，但是他不能设置宽高。

float:left;display:inline;可以解决双倍问题，可是 IE6 的双 bug 问题出现的前提什么，这种情况怎么会出现双倍间距问题呢?
前提是浮动方向跟外边距方向相反，所以才有使用负边距解决双倍边距问题的办法