

9、CSS3 新增的 transform 变形功能

利用 transform 功能能够实现文字或图像的**旋转、缩放、倾斜、移动**的变形效果，并且这四种变形效果如果使用的先后顺序不同，最终页面呈现的效果也会不同。要实现变形功能必须借助 **transform 属性**，语法是 transform:变形效果(参数)，考虑到兼容性问题，最好在 transform 前面加上前缀，比如 -webkit-（兼容 Chrome 和 Safari 的引擎）、-moz-（兼容 Firefox 的引擎）、-o-（兼容 Opera 的引擎）、-ms-（兼容 IE9+），下面是 transform 属性的四个可选属性值：

（1）旋转（rotate）

语法是：-浏览器前缀-transform:rotate(旋转角度 deg)，默认顺时针旋转，deg 是 CSS3 专有的角度单位，例如

```
<style type="text/css">
```

```
div{
    width: 100px;
    height: 100px;
    margin-left: 20px;
    background-color: #f60;
}
```



```
/*利用 UI 元素状态伪类选择器实现动态效果*/
```

```
div:hover{
```

```
    /*考虑到兼容性问题,要把所有浏览器的前缀都加上,当鼠标移动到 div 元素上
    时，div 元素就顺时针旋转 45 度*/
```

```
    -ms-transform:rotate(45deg);
```

```
    -webkit-transform:rotate(45deg);
```

```
    -moz-transform:rotate(45deg);
```

```
    -o-transform:rotate(45deg);
```

```
}
```

旋转（rotate）



```
</style>
```

（2）缩放（scale）

语法是：-浏览器前缀-transform:scale(缩放倍率)，例如 0.5 代表缩放到原来的 50%，有三种情况：scale(x,y)代表元素在 X 轴和 Y 轴同时缩放、scaleX(x)代表仅在 X 轴缩放、scaleY(y)代表仅在 Y 轴缩放，例如：

```
<style type="text/css">
```

```
div{
    width: 100px;
    height: 100px;
    margin-left: 20px;
    background-color: #f60;
}
```

```
/*利用 UI 元素状态伪类选择器实现动态效果*/
```

```
div:hover{
```

```
    /*考虑到兼容性问题,要把所有浏览器的前缀都加上,当鼠标移动到 div 元素上
    时，div 元素就缩放指定的倍数*/
```

```
    -ms-transform:scale(2,2);/*同时放大*/
```

```
    -webkit-transform:scale(0.1,0.1);/*同时缩小*/
```

```

        -moz-transform:scaleX(2);/* 只在 X 轴上放大*/
        -o-transform:scaleY(2);/* 只在 Y 轴上放大*/
    }
</style>

```

（3）倾斜（skew）

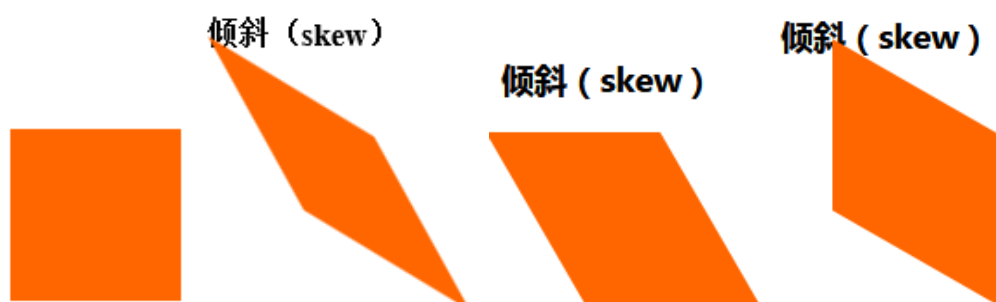
语法是：-浏览器前缀-transform:skew(倾斜角度 deg)，deg 是 CSS3 专有的角度单位，有三种情况：skew(x,y)代表元素在水平方向和垂直方向同时扭曲变形，即 X 轴和 Y 轴同时按照指定角度倾斜，只有一个参数时代表在水平方向上倾斜；skewX(x)代表元素仅在水平方向变形，即 Y 轴倾斜；skewY(y)代表元素仅在垂直方向变形，即 X 轴倾斜，例如：

```

<style type="text/css">
    div{
        width: 100px;
        height: 100px;
        margin-left: 20px;
        background-color: #f60;
    }
    /*利用 UI 元素状态伪类选择器实现动态效果*/
    div:hover{
        /*考虑到兼容性问题,要把所有浏览器的前缀都加上,当鼠标移动到 div 元素上
        时，div 元素就按照一定的角度扭曲变形*/
        -ms-transform:skew(30deg,30deg); /*同时扭曲*/
        -webkit-transform:skew(30deg,30deg);
        -moz-transform:skewX(30deg);/* 只在水平方向上扭曲*/
        -o-transform:skewY(30deg);/* 只在垂直方向上扭曲*/
    }
</style>

```

div 元素的初始状态、水平方向和垂直方向同时扭曲变形、仅在水平方向变形（仅 Y 轴倾斜）、仅在垂直方向变形（仅 X 轴倾斜）的效果图如下：



（4）移动（translate）

语法是：-浏览器前缀-transform:translate(移动距离 px)，有三种情况：translate(x,y)代表元素在水平方向和垂直方向同时移动，只有一个参数时代表在水平方向上移动；translateX(x)代表元素仅在水平方向移动，translateY(y)代表元素仅在垂直方向移动，例如：

```

<style type="text/css">
    div{
        width: 100px;
        height: 100px;
    }

```

```

        margin-left: 30px;
        background-color: #f60;
    }
    /*利用 UI 元素状态伪类选择器实现动态效果*/
    div:hover{
        /*考虑到兼容性问题,要把所有浏览器的前缀都加上,当鼠标移动到 div 元素上
        时, div 元素就按照指定的距离移动*/
        -ms-transform:translate(50px,50px);
        -webkit-transform:translate(50px,50px);
        -moz-transform:translateX(50px);/*只在水平方向上移动*/
        -o-transform:translateY(50px);/*只在垂直方向上移动*/
    }
</style>

```

（5）对一个元素同时使用多种变形方法（很重要）

语法是：-浏览器前缀-transform:方法 1 方法 2 方法 3 方法 4，具体 demo 如下：

```

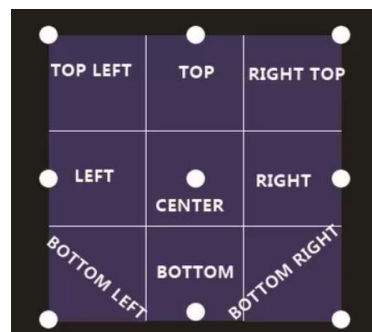
<style type="text/css">
    div{
        width: 100px;
        height: 100px;
        margin-left: 30px;
        background-color: #f60;
    }
    /*利用 UI 元素状态伪类选择器实现动态效果*/
    div:hover{
        /*考虑到兼容性问题,要把所有浏览器的前缀都加上,当鼠标移动到 div 元素上
        时, div 元素先平移，再旋转，最后缩放*/
        -ms-transform:translate(50px,50px) rotate(40deg) scale(2,2);
        -webkit-transform:translate(50px,50px) rotate(40deg) scale(2,2);
        -moz-transform:translateX(50px) rotate(40deg) scale(2,2);
        -o-transform:translateY(50px) rotate(40deg) scale(2,2);
    }
</style>

```

（6）改变元素变形的基点：利用 transform-origin 属性

可选属性值及所有基点如下：

top left	top	top right
left	center	right
bottom left	bottom	bottom right



例如旋转的默认基点为 center，可以改变旋转基点为 bottom right，利用 transform-origin: bottom right;

10、CSS3 新增的动画功能

在 CSS3 中使用动画功能可以使网页上的文字或图像具有动画效果，比如使背景颜色从一种颜色平滑过渡到另外一种颜色，主要的动画功能包括两个：

（1）Transition 功能

支持在一定的时间区间内从一个属性值平滑过渡到另外一个属性值，这种效果可以在鼠标单击、获得焦点等任何事件中触发，基本语法是 **transition:property duration timing-function delay**，考虑到兼容性问题，最好在 transition 前面加上前缀，比如-webkit-（兼容 Chrome 和 Safari 的引擎）、-moz-（兼容 Firefox 的引擎）、-o-（兼容 Opera 的引擎）、-ms-（兼容 IE9+）。transition 必需的四个属性值为：

- 1) **property**: 指定应用过渡效果的 CSS 属性名称，当指定的 CSS 属性值改变时，过渡效果将开始，可选值有：none（没有属性会获得过渡效果）、all（所有属性都会获得过渡效果）、properties（应用过渡效果的多个属性设置，每四个参数列表中间以逗号分隔）
- 2) **duration**: 指定完成过渡效果需要花费的时间，单位可以是秒或毫秒，默认值为 0
- 3) **timing-function**: 指定过渡变换的速率变化函数，可选值有 ease（默认值，开始时很快然后逐渐变慢）、linear（匀速）、ease-in（中间突然加速）、ease-out（中间突然减速）、ease-in-out（突然加速再突然减速）、cubic-bezier(n,n,n,n)（利用贝塞尔曲线函数设置自定义速率，其中 n 在 0-1 之间取值）
- 4) **delay**: 过渡变换开始前的延迟时间

具体 demo 如下：

```
<style type="text/css">
```

```
    div{
        width:200px;
        height: 200px;
        background-color: green;
        /*过渡效果应用的是 div 元素的所有属性名，过渡持续时间为 1 秒，速率变化
        为匀速，不延迟，即一旦属性值发生变化，立马开始过渡*/
        -moz-transition:all 1s linear 0;
        /*等价语法是-moz-transition:background-color 1s linear 0,width 1s linear 0,height 1s linear 0;*/
        -webkit-transition:all 1s ease 0;
        -o-transition:all 1s ease-in 0;
        /*过渡效果应用的是 div 元素的所有属性名，过渡持续时间为 1 秒，速率变化
        为突然减速，属性值发生变化 2 秒后才开始过渡*/
        -ms-transition:all 1s ease-out 2s;
    }
    /*当鼠标移动到 div 元素上时，背景颜色发生变化，这时过渡效果就开始了*/
    div:hover{
        background-color: #f60;
        width:600px;
        height: 600px
    }
}
```

```
</style>
```

演示效果见本地文件 C:\Users\Administrator\Desktop\css3 新增标签\动画功能\index1.html

注意：transition 只能通过指定属性的开始值与结束值来实现平滑过渡的简单动画效果，要实现更复杂的动画效果必须使用 Animations 功能

（2）Animations 功能：通过创建关键帧以及设置每个关键帧的样式在页面上产生更复杂的动画效果，基本语法是@-浏览器前缀-keyframes 关键帧合集名称{关键帧创建的代码}

其中关键帧创建的语法是：0%~100%{对应关键帧中的样式}

注意：关键帧的创建都是以百分比开头的，如果将一个动画划分成 100 帧，则 10%代表动画执行到第 10 帧的效果

关键帧创建好以后，还要在元素的样式中使用该关键帧，方法如下：

```
元素{
-webkit-animation-name:关键帧合集名称;
-webkit-animation-duration:整个动画持续时间;
-webkit-animation-timing-function:动画速率变化函数;
-webkit-animation-iteration-count:动画播放次数;（infinite 表示无限循环播放）
}
```

具体 demo 如下：

```
<!DOCTYPE html>
<html>
<head>
  <meta charset='utf-8'>
  <title>Animations 功能</title>
  <style type="text/css">
    div{
      width:20px;
      height: 200px;
      background-color: #f60;
    }
    /*创建关键帧合集并设置每个关键帧的样式，可以设置背景相关样式、各种
    transform 变换功能*/
    @-webkit-keyframes mycolor{
      0%{
        background-color: #f60;
        -webkit-transform:rotate(0deg);
      }
      10%{
        background-color: green;
        width:100px;
        -webkit-transform:rotate(10deg);
      }
      20%{
        background-color: red;
        width:200px;
        -webkit-transform:rotate(20deg);
      }
    }
```

```

        80%{
            background-color: blue;
            width:800px;
            -webkit-transform:rotate(80deg);
        }
        100%{
            background-color: #f60;
            width:1000px;
            -webkit-transform:rotate(100deg);
        }
    }
    /*当鼠标移动到 div 元素上时，背景颜色不断变化,同时宽度也在不断变大，旋转角
    度也在不断变大*/
    div:hover{
        -webkit-animation-name:mycolor;
        -webkit-animation-duration:5s;
        -webkit-animation-timing-function:linear;
        -webkit-animation-iteration-count:infinite;
    }
</style>
</head>
<body>
<h3>Animations 功能</h3>
<div></div>
</body >
</html>

```

演示效果见本地文件 C:\Users\Administrator\Desktop\css3 新增标签\动画功能\index2.html

11、布局相关样式

网页布局是指在页面中如何对标题、导航栏、主要内容、页脚、表单等各种构成要素进行一个合理的编排，CSS3 之前主要使用 `float` 属性或 `position` 属性进行页面的简单布局，但是它们存在一些缺点，比如两栏或者多栏布局的时候，可能出现元素内容高度不一致导致底部很难对齐，CSS3 中追加的新的布局方式可以解决这些问题

（1）多栏布局（每个栏目的宽度都是一样的）

方法一：利用 `column-count` 属性将一个元素中的内容分成多栏显示，兼容性语法是-浏览器前缀-`column-count:栏目数`，使用该属性时要将**元素宽度**设置成多个栏目的**总宽度**，例如：

```

<style type="text/css">
    div{
        width:600px;/*元素的总宽度*/
        background-color: #f60;
        -webkit-column-count:3;/*分成三栏，每一栏宽度都是 200px */
        -moz-column-count:3;
    }
</style>

```

column-count属性

网页布局是指在页面中如何对标题、导航栏、主要内容、页脚、表单等各种构成要素进行一个合理的编排,CSS3之前主要使用float属性或position属性进行页面的简单布局,但是它们存在一些缺点,比如两栏或者多栏布局的时候,可能出现元素的内容高度不一致导致底部很难对齐,CSS3中追加的新的布局方式可以解决这些问题.利用column-count属性将一个元素中的内容分成多栏显示,兼容性语法是-浏览器前缀-column-count:栏目数,在使用多栏布局时,要将元素的宽度设置成多个栏目的总宽度

效果图为:

方法二: 利用 **column-width** 属性指定每个栏目的同一宽度来生成分栏, 兼容性语法是-浏览器前缀-column-width:栏目宽度, 例如:

```
<style type="text/css">
    div{
        width:600px;
        -webkit-column-width:200px; /*每一栏宽度都是 200px */
        -moz-column-width:200px;
    }
</style>
```

方法三: 利用 **column-gap** 属性指定栏目与栏目之间的距离, 兼容性语法是-浏览器前缀-column-gap:栏目与栏目之间的距离, 例如:

```
<style type="text/css">
    div{
        width:100%;
        background-color: #f60;
        -webkit-column-gap:10px;
        -moz-column-gap:10px;
    }
</style>
```

方法四: 利用 **column-rule** 属性在栏目与栏目之间增加一条分割线, 兼容性语法是-浏览器前缀-column-rule:分割线的样式 宽度 颜色, 例如:

```
<style type="text/css">
    div{
        width:100%;
        background-color: #f60;
        -webkit-column-rule:solid 10px #000;
        -moz-column-rule:solid 10px #000;
    }
</style>
```

(2) 盒布局 (可以给每个栏目设置不同宽度)

CSS3 之前使用 **float** 属性和 **position** 属性时存在多栏底部不能对齐的缺点, 例如下面的 demo:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8'>
    <title>盒布局</title>
```

```

<style type="text/css">
    #box{
        width:1000px;
        margin:0 auto;/*让整个 div 元素居中显示*/

    }
    div#left{
        float: left;
        width: 300px;
        padding: 10px;
        background-color: green;
    }
    div#center{
        float: left;
        width: 400px;
        padding: 10px;
        background-color: red;
    }
    div#right{
        float: right;
        width: 300px;
        padding: 10px;
        background-color: #f60;
    }
    div#left,div#center,div#right{
        -webkit-box-sizing:border-box;/*指定盒模型为 border-box*/
        -moz-box-sizing:border-box;
    }
</style>
</head>
<body>
<div id='box'>
    <div id='left'>
        <h2>栏目列表</h2>
        <ul>
            <li><a href="index.html">栏目名称</a></li>
            <li><a href="index.html">栏目名称</a></li>
            <li><a href="index.html">栏目名称</a></li>
        </ul>
    </div>
    <div id='center'>
        <p>CSS3 新增的背景样式、边框样式、变换功能、动画功能相关的语法规则在不同
        浏览器中可能是不一样的，页面的显示效果可能也是不一样的</p>
    </div>

```

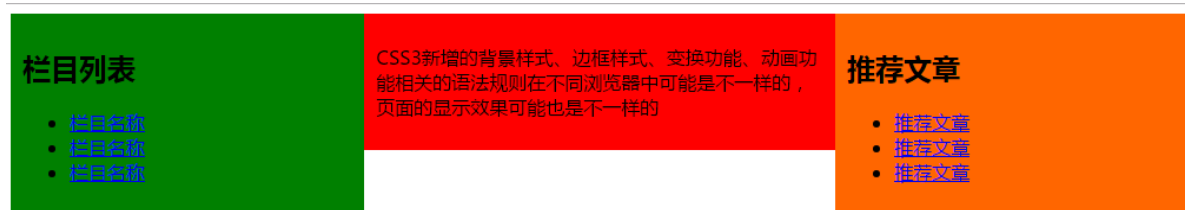


```

<div id='right'>
  <h2>推荐文章</h2>
  <ul>
    <li><a href="index.html">推荐文章</a></li>
    <li><a href="index.html">推荐文章</a></li>
    <li><a href="index.html">推荐文章</a></li>
  </ul>
</div>
</div>
</body >
</html>

```

效果图为：



但 CSS3 中使用**盒布局可以使多栏显示时底部对齐**，只要在最外层的盒子中设置 display 样式为 box，即 **display:-webkit-box** 即可，具体 demo 如下；

```

<style type="text/css">
  #box{
    width:1000px;
    margin:0 auto;/*让整个 div 元素居中显示*/
    display:-webkit-box;/*显示方式为盒布局，分栏显示时底部对齐*/
    display:-moz-box;
  }
  div#left{
    width: 300px;
    padding: 10px;
    background-color: green;
  }
  div#center{
    width: 400px;
    padding: 10px;
    background-color: red;
  }
  div#right{
    width: 300px;
    padding: 10px;
    background-color: #f60;
  }
  div#left,div#center,div#right{

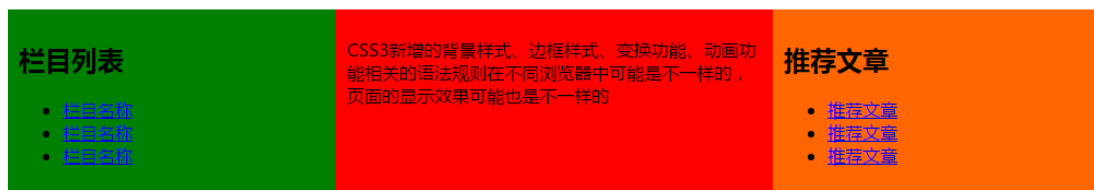
```

```

        -webkit-box-sizing: border-box; /* 指定盒模型宽高计算方式为 border-box */
        -moz-box-sizing: border-box;
    }
</style>
<body>
<div id='box'>
    <div id='left'>
        <h2>栏目列表</h2>
        <ul>
            <li><a href="index.html">栏目名称</a></li>
            <li><a href="index.html">栏目名称</a></li>
            <li><a href="index.html">栏目名称</a></li>
        </ul>
    </div>
    <div id='center'>
        <p>CSS3 新增的背景样式、边框样式、变换功能、动画功能相关的语法规则在不同的浏览器中可能是不一样的，页面的显示效果可能也是不一样的</p>
    </div>
    <div id='right'>
        <h2>推荐文章</h2>
        <ul>
            <li><a href="index.html">推荐文章</a></li>
            <li><a href="index.html">推荐文章</a></li>
            <li><a href="index.html">推荐文章</a></li>
        </ul>
    </div>
</div>
</body>

```

改进后的效果图为：



注意：多栏布局与盒布局的适用场景不同，多栏布局的每个栏目宽度都是一样的，比较适合在文字内容页面使用，比如一篇文章进行分栏显示；而整个网页编排时每个栏目的宽度不可能全都一样，所以只能使用盒布局

知识点扩展

1) 使用盒布局时，可以通过在每个元素中加入 **box-ordinal-group** 属性来改变各个元素的显示顺序，属性值是一个表示序号的正数，浏览器会根据序号从小到大来显示这些元素，兼容性语法为：-浏览器前缀-box-ordinal-group:一个正数，例如：

```

<style type="text/css">
    body{

```

```

        margin:0;
        padding: 0;
    }
    #box{
        margin:0 auto;/*让整个 div 元素居中显示*/
        display:-webkit-box;/*让盒模型分栏显示时底部对齐*/
        display: -moz-box;
    }
    /*三个 div 元素的显示顺序从左、中、右变成中、右、左*/
    div#left{
        width: 300px;
        padding: 10px;
        background-color: green;
        -webkit-box-ordinal-group:3;/*指定该盒模型第三个显示*/
        -moz-box-ordinal-group:3;
    }
    div#center{
        padding: 10px;
        background-color: red;
        -webkit-box-ordinal-group:1;/*指定该盒模型第一个显示*/
        -moz-box-ordinal-group:1;
    }
    div#right{
        width: 300px;
        padding: 10px;
        background-color: #f60;
        -webkit-box-ordinal-group:2;/*指定该盒模型第二个显示*/
        -moz-box-ordinal-group:2;
    }
    div#left,div#center,div#right{
        -webkit-box-sizing:border-box;/*指定盒模型宽高计算方式为 border-box*/
        -moz-box-sizing:border-box;
    }
}
</style>

```

2) 在使用盒布局时, 可以通过 **box-orient** 属性来指定多个元素的排列方向, 属性值有: **horizontal**(从左往右水平排列子元素)、**vertical**(从上往下垂直排列子元素), 兼容性语法为: **-浏览器前缀-box-orient:属性值**, 例如:

```

#box{
    margin:0 auto;/*让整个 div 元素居中显示*/
    display:-webkit-box;/*让盒模型分栏显示时底部对齐*/
    display: -moz-box;
    -webkit-box-orient:vertical;/* 让三个元素按照从上往下垂直排列*/
    -moz-box-orient:vertical;
}

```

```
div#left{
    width: 300px;
    padding: 10px;
    background-color: green;
}
div#center{
    padding: 10px;
    background-color: red;
}
div#right{
    width: 300px;
    padding: 10px;
    background-color: #f60;
}
```

3) 使用盒布局时, 可以使用 **box-align** 属性及 **box-pack** 属性指定元素中的文字、图像的水平对齐方式和垂直对齐方式, 兼容性语法为: -浏览器前缀-box-align:属性值 或 -浏览器前缀-box-pack:属性值, 属性值有: start (左对齐、顶部对齐)、center (中部对齐)、end (右对齐、底部对齐), 例如:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8'>
    <title>盒布局</title>
    <style type="text/css">
        div{
            width:300px;
            height:200px;
            background-color: green;
            border:solid 5px #000;
            display:-webkit-box;/*显示为盒布局*/
            display: -moz-box;
            -webkit-box-align:center;/*水平对齐方式*/
            -moz-box-align:center;
            -webkit-box-pack:center;/*垂直对齐方式*/
            -moz-box-pack:center;
        }
    </style>
</head>
<body>
<div>指定水平和垂直对齐方式</div>
</body >
</html>
```



(3) 弹性盒布局 (盒模型能够自适应窗口)

使用 **box-flex** 属性可以使盒布局变成弹性盒布局, 也就是盒模型大小可以随浏览器窗口变化

而变化，只要在需要弹性变化的盒模型中加入 **box-flex** 属性即可，兼容性语法为：**-浏览器前缀-box-flex:伸缩比例**，注意若每个子元素的伸缩比例都相同，则最终呈现效果相当于均分父元素，具体 demo 如下：

```
<style type="text/css">
    body{
        margin:0;
        padding:0;
    }
    #box{
        margin:0 auto;/*让整个 div 元素居中显示*/
        display:-webkit-box;/*让盒模型分栏显示时底部对齐*/
        display: -moz-box;
    }
    div#left{
        width: 300px;
        padding: 10px;
        background-color: green;
    }
    div#center{
        padding: 10px;
        background-color: red;
        -webkit-box-flex:1;/*让中间元素变成弹性盒布局，即可伸缩，则最外层整个 div
        元素会随着浏览器窗口大小的改变而变化*/
        -moz-box-flex:1;
    }
    div#right{
        width: 300px;
        padding: 10px;
        background-color: #f60;
    }
    div#left,div#center,div#right{
        -webkit-box-sizing:border-box;/*指定盒模型宽高计算方式为 border-box*/
        -moz-box-sizing:border-box;
    }
</style>
```

知识点扩展：

如果盒布局中有空白，可以使用**弹性盒布局**来消除空白，方法就是在其中一个子 **div** 元素或全部子 **div** 元素中加入 **box-flex** 属性，例如：

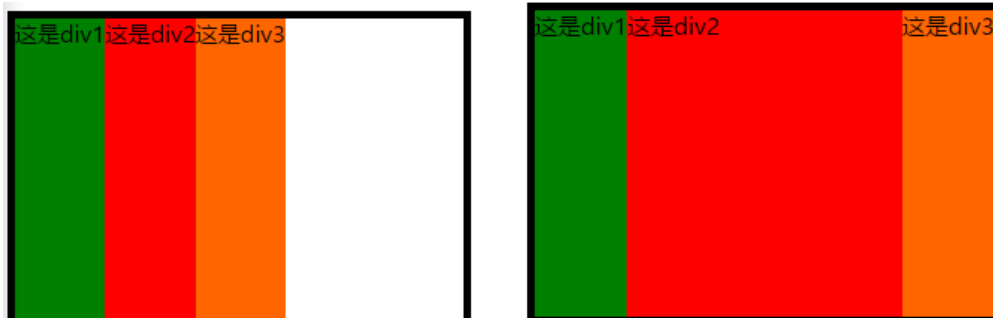
```
<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8'>
    <title>盒布局</title>
    <style type="text/css">
```

```

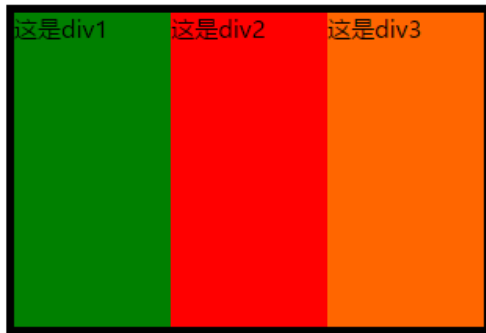
#box{
    width:300px;
    height:200px;
    border:solid 5px #000;
    display:-webkit-box;/*让盒模型分栏显示时底部对齐*/
    display: -moz-box;
    -webkit-box-orient:horizontal;/*让三个元素按照从左往右水平排列*/
    -moz-box-orient:horizontal;
}
div#div1{
    background-color: green;
}
div#div2{
    background-color: red;
    -webkit-box-flex:1;/*只让 div2 变成弹性盒布局来消除空白*/
    -moz-box-flex:1;
}
div#div3{
    background-color: #f60;
}
div#div1,div#div2,div#div3{
    -webkit-box-sizing:border-box;/*指定盒模型宽高计算方式为 border-box*/
    -moz-box-sizing:border-box;
    font-size: 15px;
}
</style>
</head>
<body>
<div id='box'>
    <div id='div1'>这是 div1</div>
    <div id='div2'>这是 div2</div>
    <div id='div3'>这是 div3</div>
</div>
</body >
</html>

```

未使用 box-flex 属性及使用 box-flex 属性之后的对比图如下：



注意：如果在每个子 `div` 元素中都加入同样的 `box-flex` 属性值，相当于三个 `div` 元素都是弹性可伸缩的，则消除空白后的效果如下图所示（相当于均分）：



12、兼容性问题

CSS3 新增的背景样式、边框样式、变换功能、动画功能相关的语法规则在不同浏览器中可能是不一样的，页面的显示效果可能也是不一样的，所以就需要考虑到兼容性问题，对于兼容性测试的工具具有：IETester 和 Multibrowser，用于查找各个浏览器中显示效果存在的缺陷

最后给大家推荐一个学习或者查找前端相关知识的网站：

www.w3school.com.cn