

第八章 弹性盒子

1. 概念

弹性布局（Flex布局）是一种现代的CSS布局方式，通过使用display: flex属性来创建一个弹性容器，并在其中使用灵活的盒子模型来进行元素的排列和定位。

2. 优点

弹性盒子最重要的知识点是突出弹性两个字，所谓弹性就是赋予盒子最大的灵活性，不用去考虑各个浏览器的兼容性，因为基本市场上所见的浏览器都能够兼容弹性盒子布局，给开发移动端带来了极大的便利。

3. 特点

1. 主轴与交叉轴：弹性容器具有主轴（main axis）和交叉轴（cross axis）。默认情况下，主轴是水平方向，交叉轴是垂直方向。

2. 弹性容器：通过将父元素的display属性设置为flex或inline-flex来创建弹性容器。

3. 子元素的弹性项目：弹性容器中的每个子元素都成为弹性项目。子元素可以指定各自在主轴和交叉轴上的大小、顺序以及对齐方式等。4. 主轴对齐：弹性项目可以在主轴上按照一定比例分配空间，也可以使用justify-content属性定义主轴的对齐方式。

5. 交叉轴对齐：弹性项目可以在交叉轴上进行对齐，包括顶部对齐、底部对齐、居中对齐等，使用align-items属性定义交叉轴对齐方式。

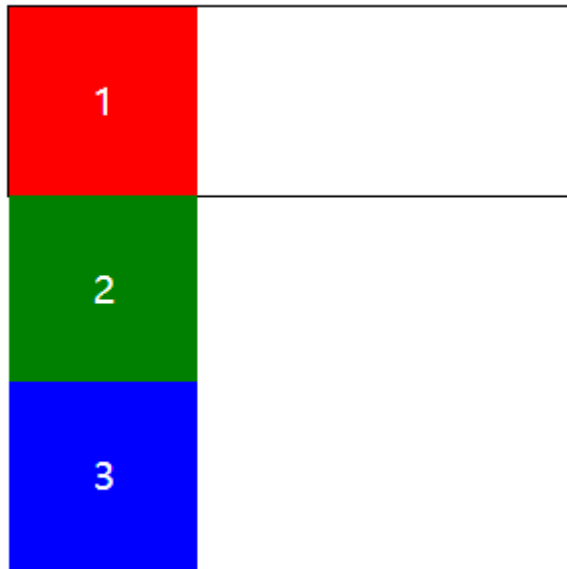
6. 换行与自动调整：可控制弹性项目是否换行，并且具备自动调整元素大小的能力。

4. 使用

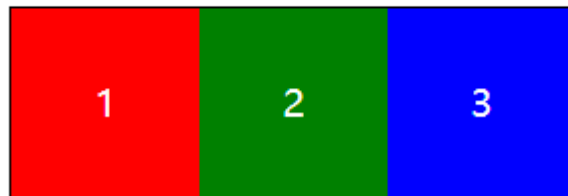
4.1 display: flex 属性

display: flex 定义标签弹性盒子布局，控制盒子的位置和排列方式。div标签下的块级元素在一行进行显示

不使用弹性盒子时



使用弹性盒子时



代码:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>弹性盒子</title>
    <style>
      *{
        padding: 0;
        margin: 0;
      }
      #box{
        width: 300px;
        height: 100px;
        margin: 300px auto;
        border: 1px solid black;
        /*设置弹性布局 */
      }
    </style>
  </head>
  <body>
    <div id="box">
      <div>1</div>
      <div>2</div>
      <div>3</div>
    </div>
  </body>
</html>
```

```
        display: flex;
    }
    #box div{
        height: 100px;
        width: 100px;
        text-align: center;
        color: white;
        font-size: 20px;
        line-height: 100px;
    }
    #box div:nth-child(1){
        background-color: red;
    }
    #box div:nth-child(2){
        background-color: green;
    }
    #box div:nth-child(3){
        background-color: blue;
    }
</style>
</head>
<body>
    <div id="box">
        <div>1</div>
        <div>2</div>
        <div>3</div>
    </div>
</body>
</html>
```

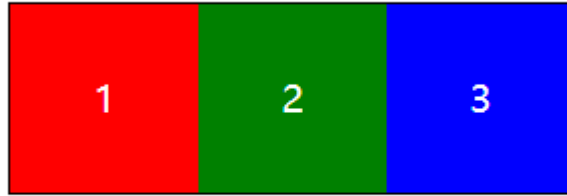
4.2 flex-direction:主轴方向

通过 display-flex 属性设置 标签主轴方向，子标签沿着主轴方向进行排列

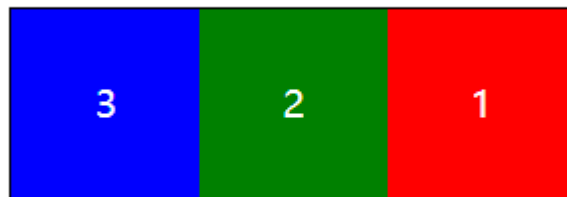
弹性盒子的主轴方向有四个，分别是

属性值	方向
row	默认值，主轴沿水平方向从左到右
row-reverse	主轴沿水平方向从右到左
column	主轴沿垂直方向从上到下
column-reverse	主轴沿垂直方向从下到上

row 从左至右：



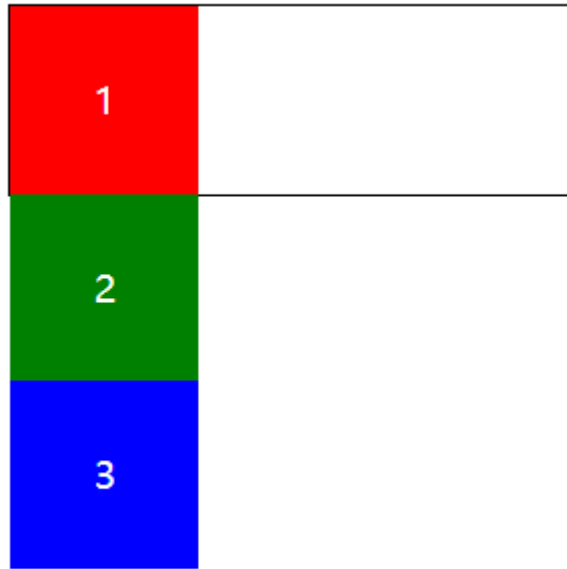
row-reverse 从右至左:



代码:

```
#box{  
    width: 300px;  
    height: 100px;  
    margin: 300px auto;  
    border: 1px solid black;  
    /*设置弹性布局 */  
    display: flex;  
    /*设置 主轴方向 */  
    flex-direction: row-reverse;  
}
```

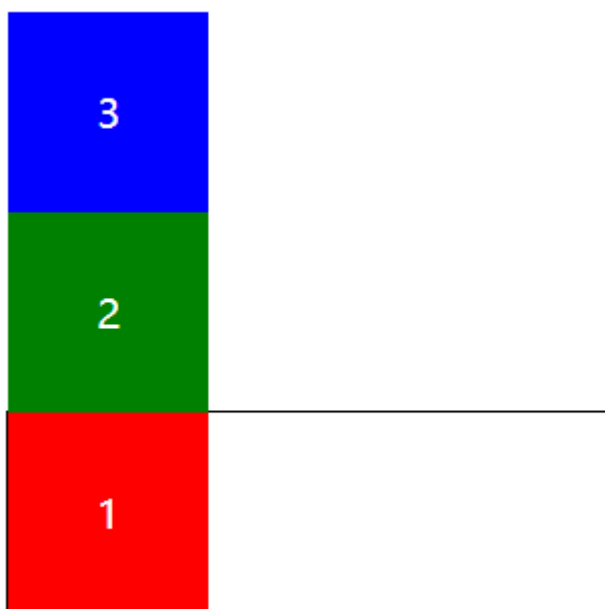
column 垂直方向: 从上到下



代码:

```
#box{  
    width: 300px;  
    height: 100px;  
    margin: 300px auto;  
    border: 1px solid black;  
    /*设置弹性布局 */  
    display: flex;  
    /*设置 主轴方向 */  
    flex-direction: column;  
}
```

column-reverse 垂直方向: 从下到上



代码:

```
#box{  
    width: 300px;  
    height: 100px;  
    margin: 300px auto;  
    border: 1px solid black;  
    /*设置弹性布局 */  
    display: flex;  
    /*设置 主轴方向 */  
    flex-direction:column-reverse;  
}
```

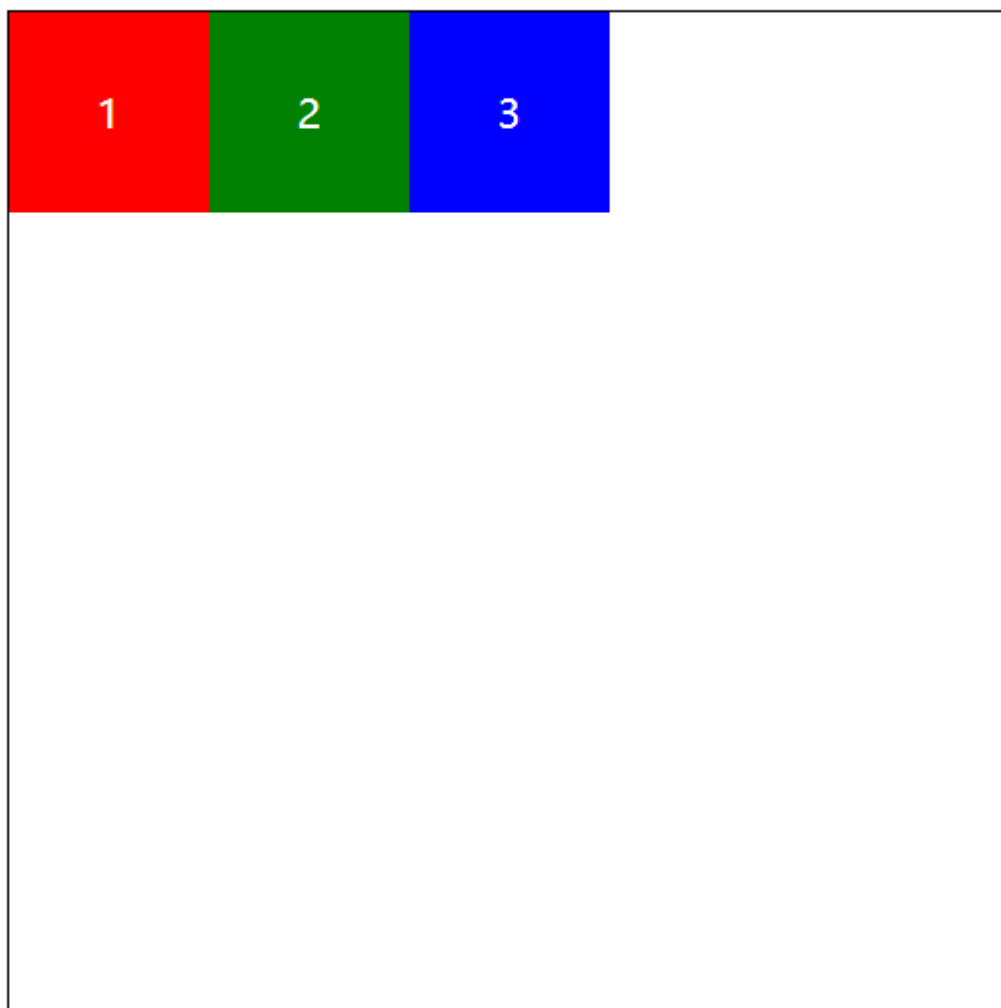
4.3 justify-content:主轴对齐方向

justify-content: 设置弹性盒子中元素在主轴（横轴）方向上的对齐方式

对齐方式有6个，分别是：

属性值	方向
flex-start	默认值, 起点开始依次排列（左对齐）
flex-end	终点开始依次排列（右对齐）
space-around	先居中对齐，然后等间距的分布在容器中
space-between	两端对齐，项目之间的间隔是相等的
space-evenly	注重间距，先将所有间距等分，再排列子元素
center	居中

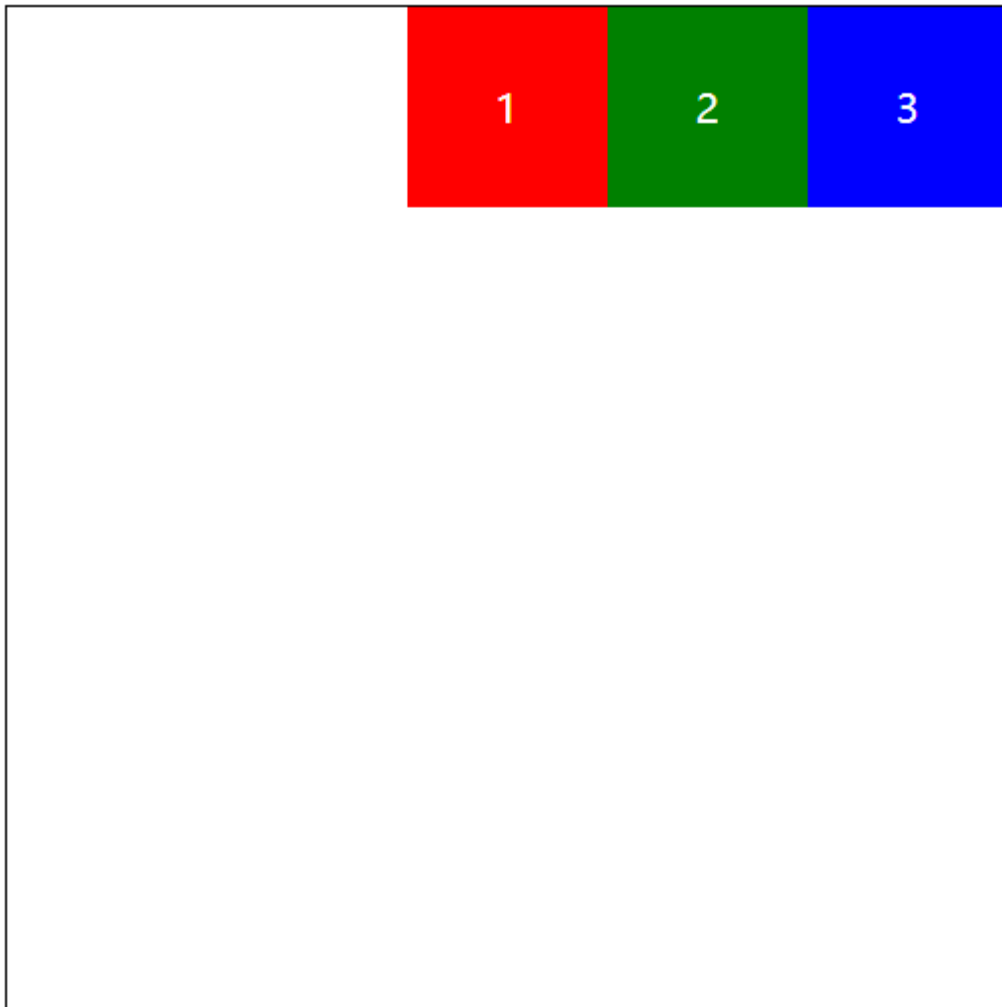
flex-start 默认值, 起点开始依次排列 (左对齐)



代码:

```
#box{  
    width: 500px;  
    height: 500px;  
    margin: 300px auto;  
    border: 1px solid black;  
    /*设置弹性布局 */  
    display: flex;  
    /*主轴方向 */  
    flex-direction: row;  
    /*设置 主轴对齐方向 */  
    justify-content: flex-start;  
}
```

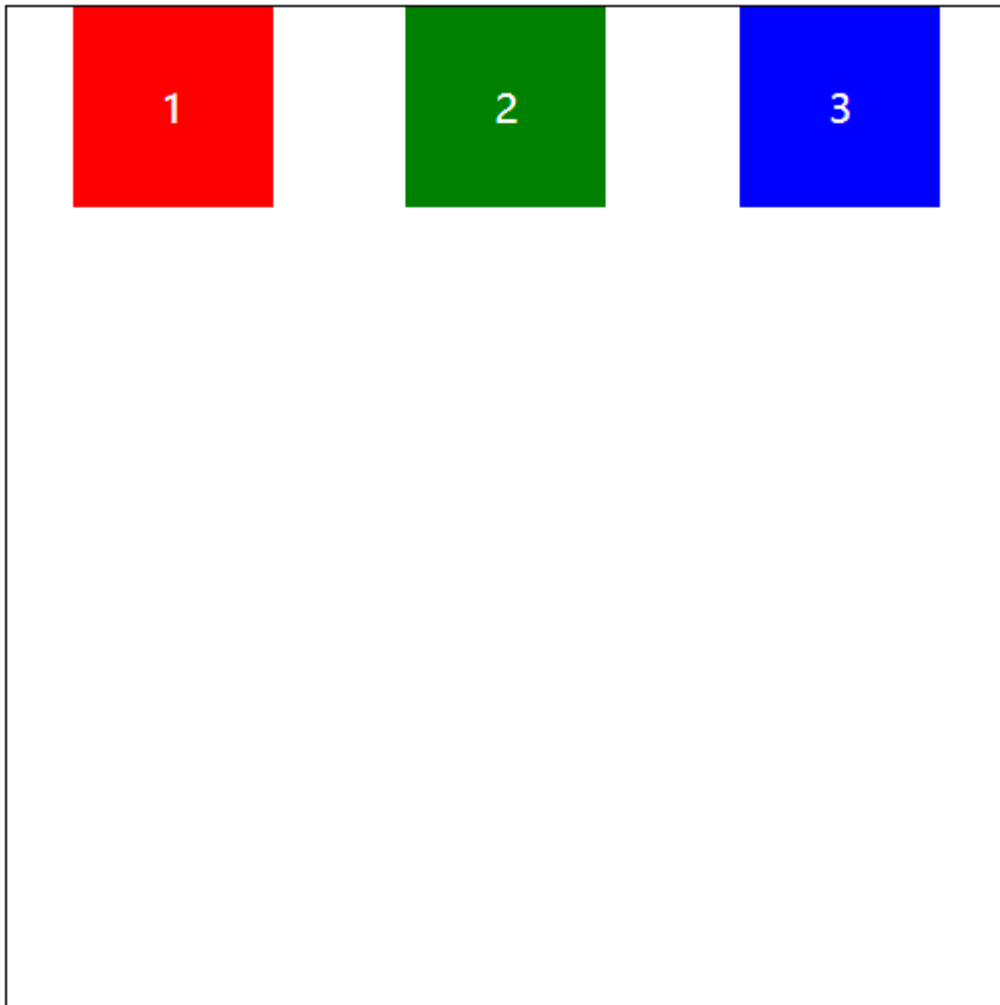
flex-end 终点开始依次排列 (右对齐)



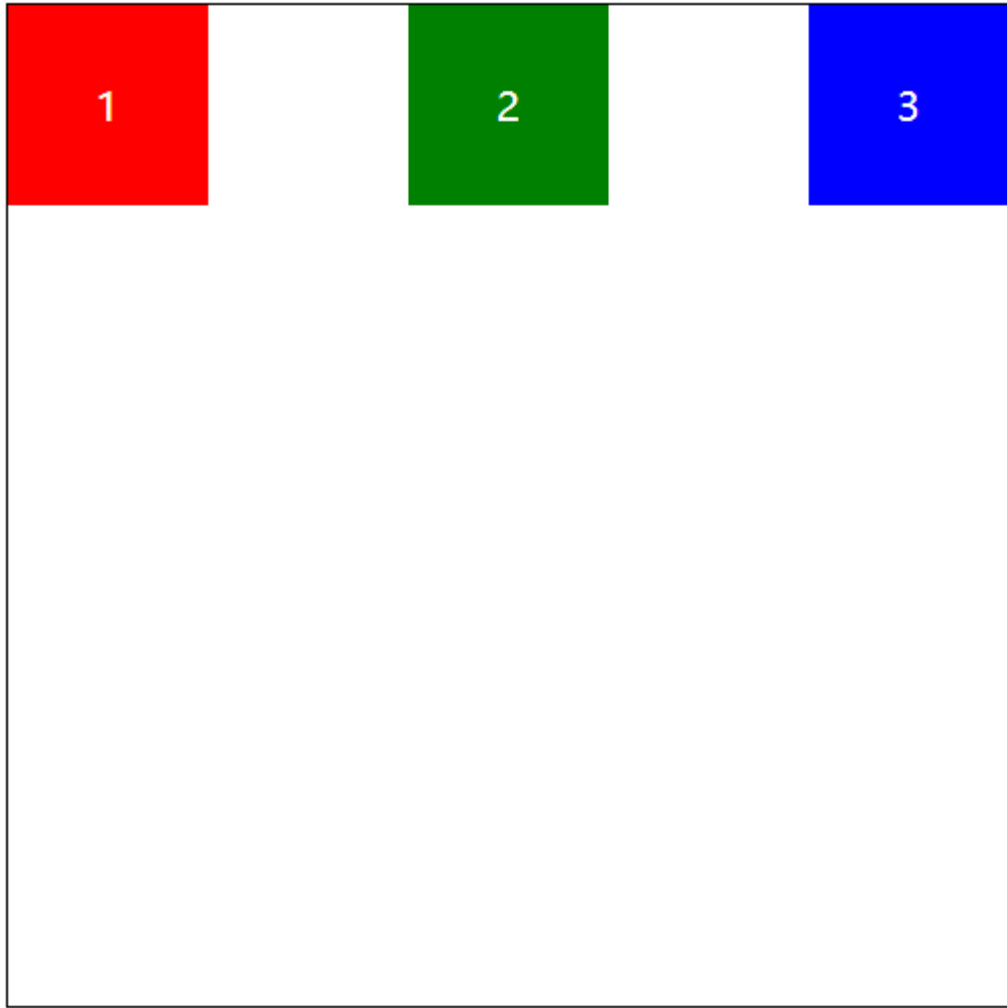
代码:

```
#box{  
    width: 500px;  
    height: 500px;  
    margin: 300px auto;  
    border: 1px solid black;  
    /*设置弹性布局 */  
    display: flex;  
    /*主轴方向 */  
    flex-direction: row;  
    /*设置 主轴对齐方向 */  
    justify-content: flex-end;  
}
```

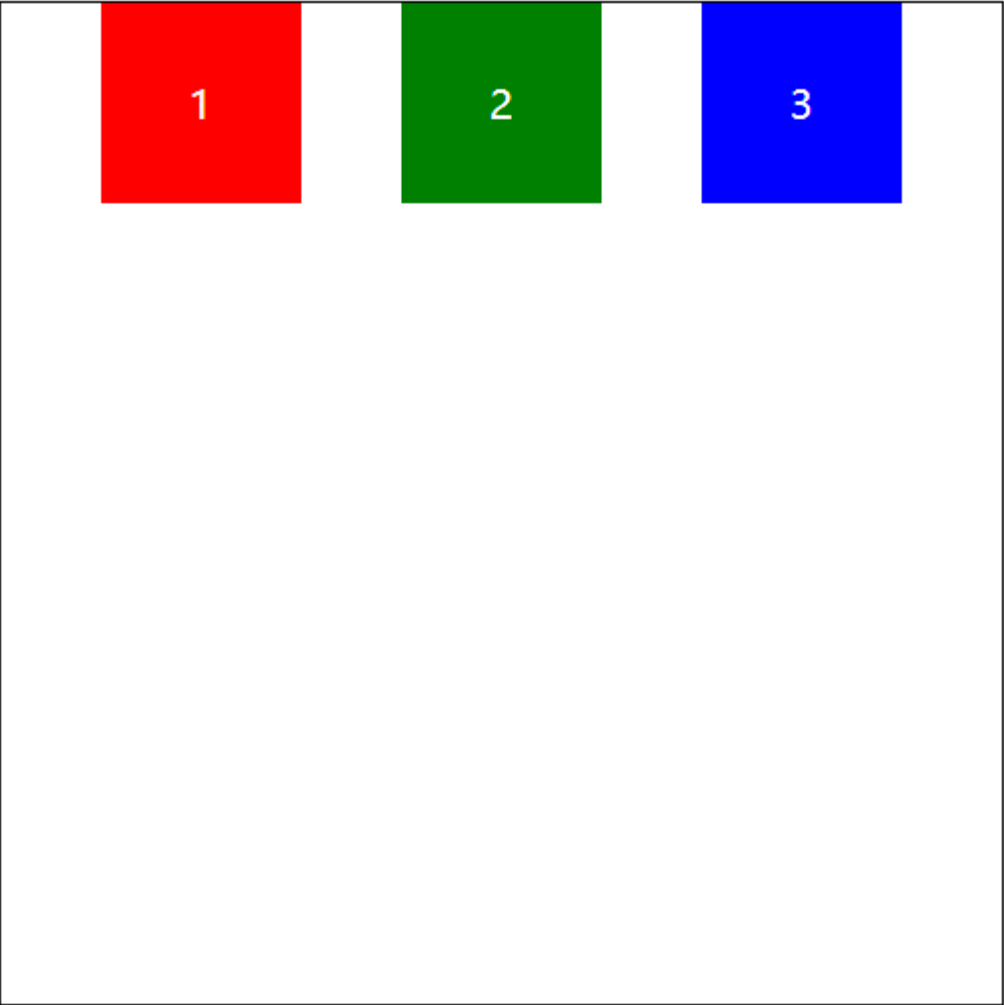
space-around 先居中对齐，然后等间距的分布在容器中



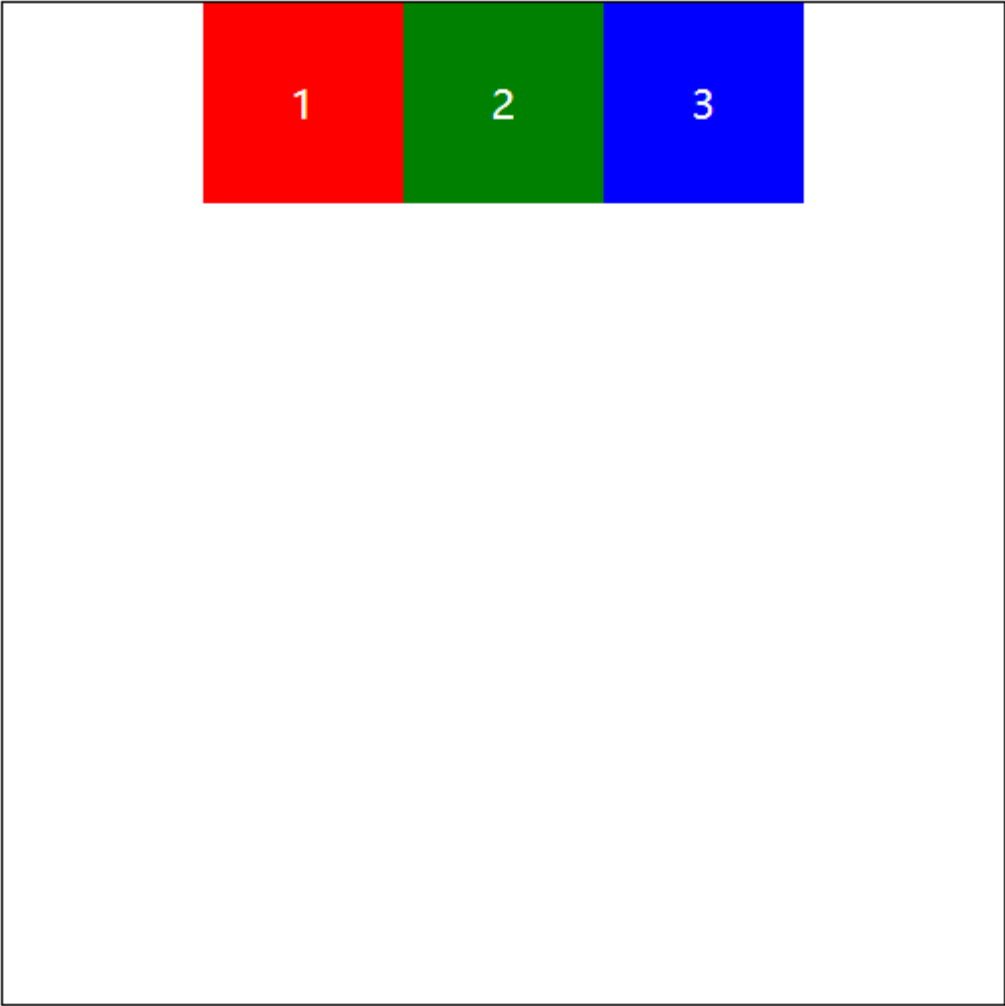
space-between 两端对齐，项目之间的间隔是相等的



space-between 注重间距，先将所有间距等分，再排列子元素



center 居中



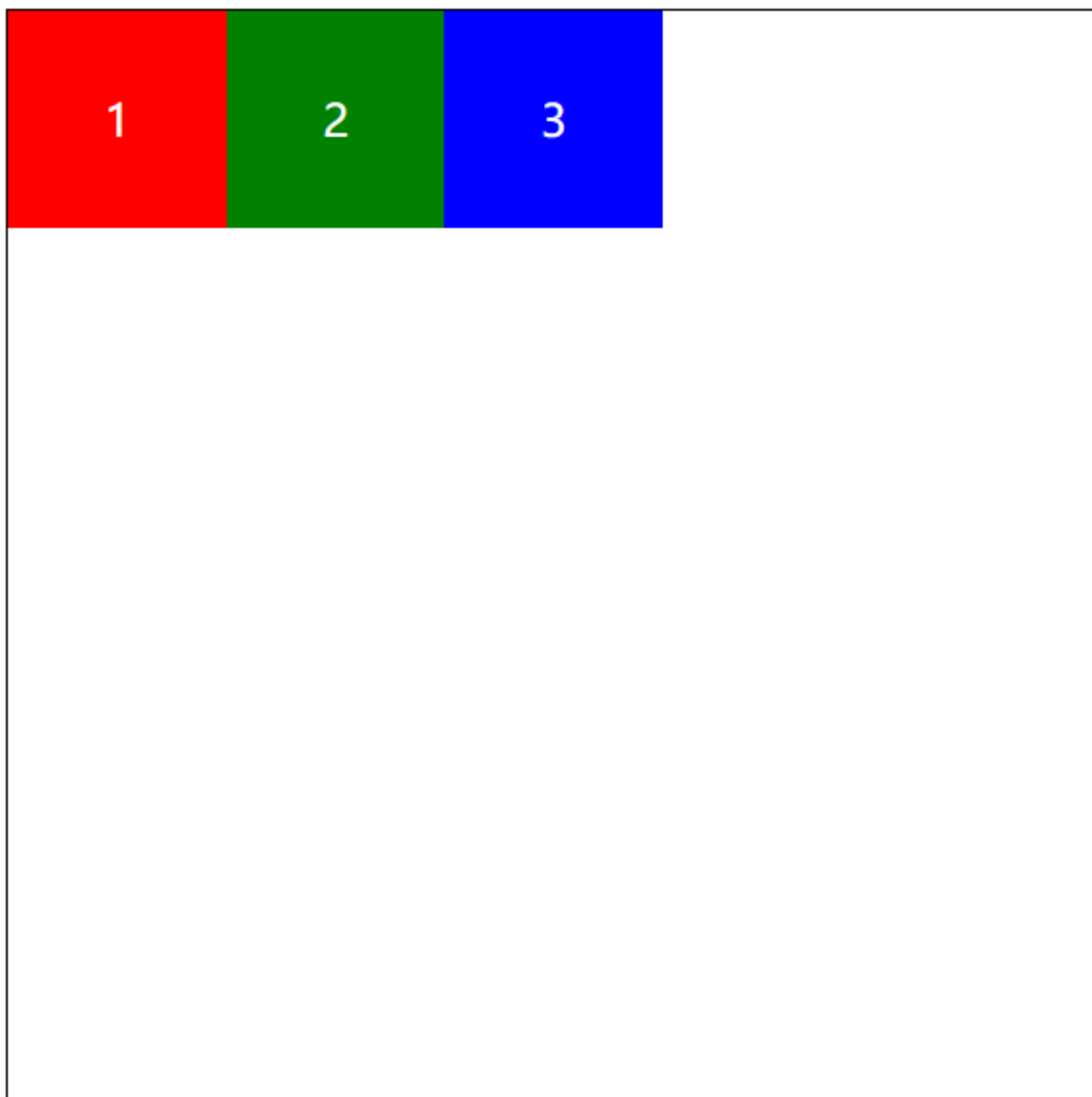
当 主轴方向为 垂直方向 时，对齐效果应该怎样呢？

4.4 align-items:侧轴对齐方式

align-items 属性是属于交叉轴（纵轴）排列方式，它定义flex子项在flex容器的当前行的侧轴（纵轴）方向上的对齐方式。

属性值	方向
flex-start	起点开始依次排列
flex-end	终点开始依次排列
center	沿侧轴居中排列，垂直居中
stretch	默认效果, 弹性盒子沿着侧轴线被拉伸至铺满容器
baseline	行内轴与侧轴为同一条时，值与flex-start效果一样

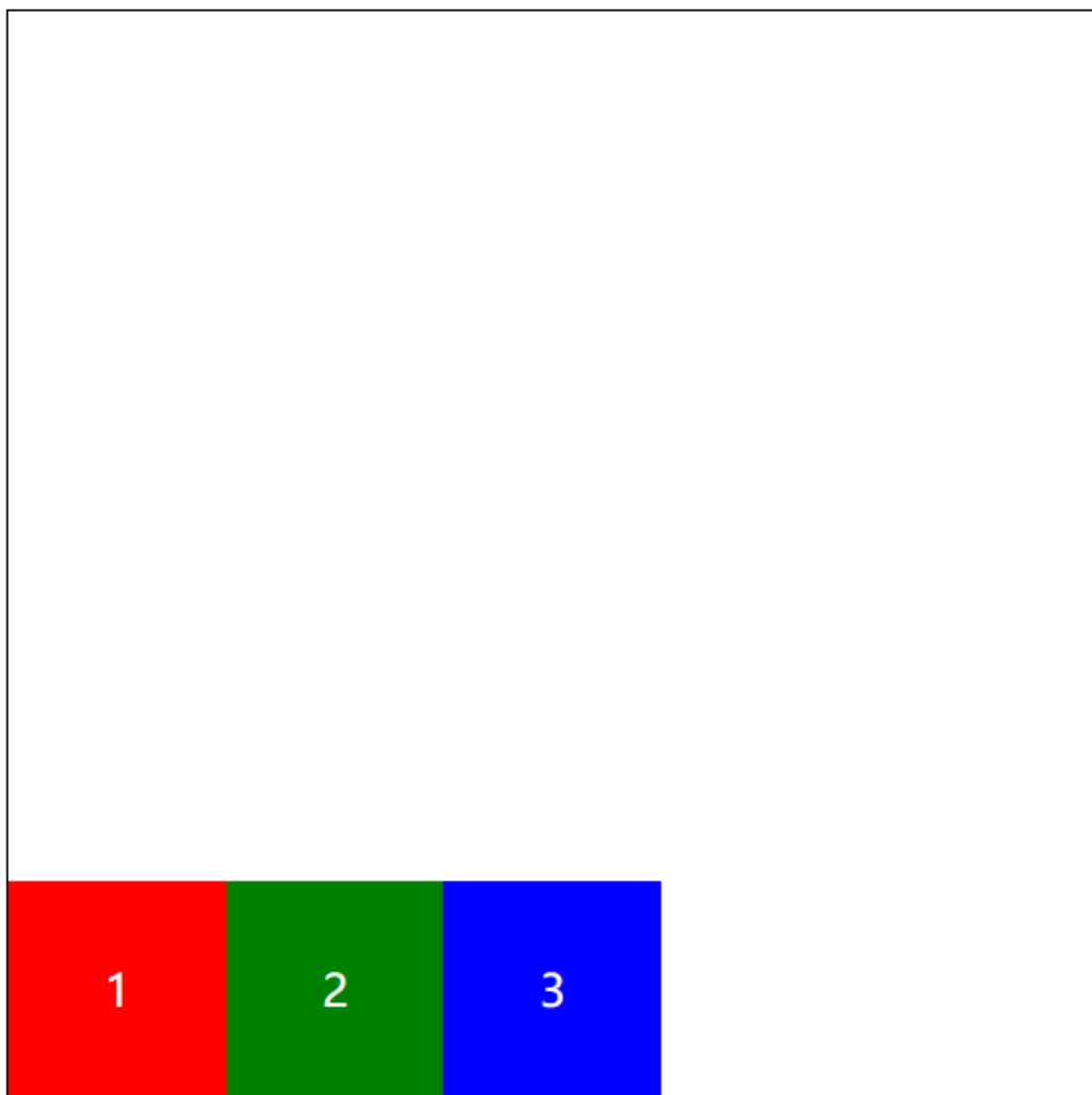
flex-start 起点开始依次排列



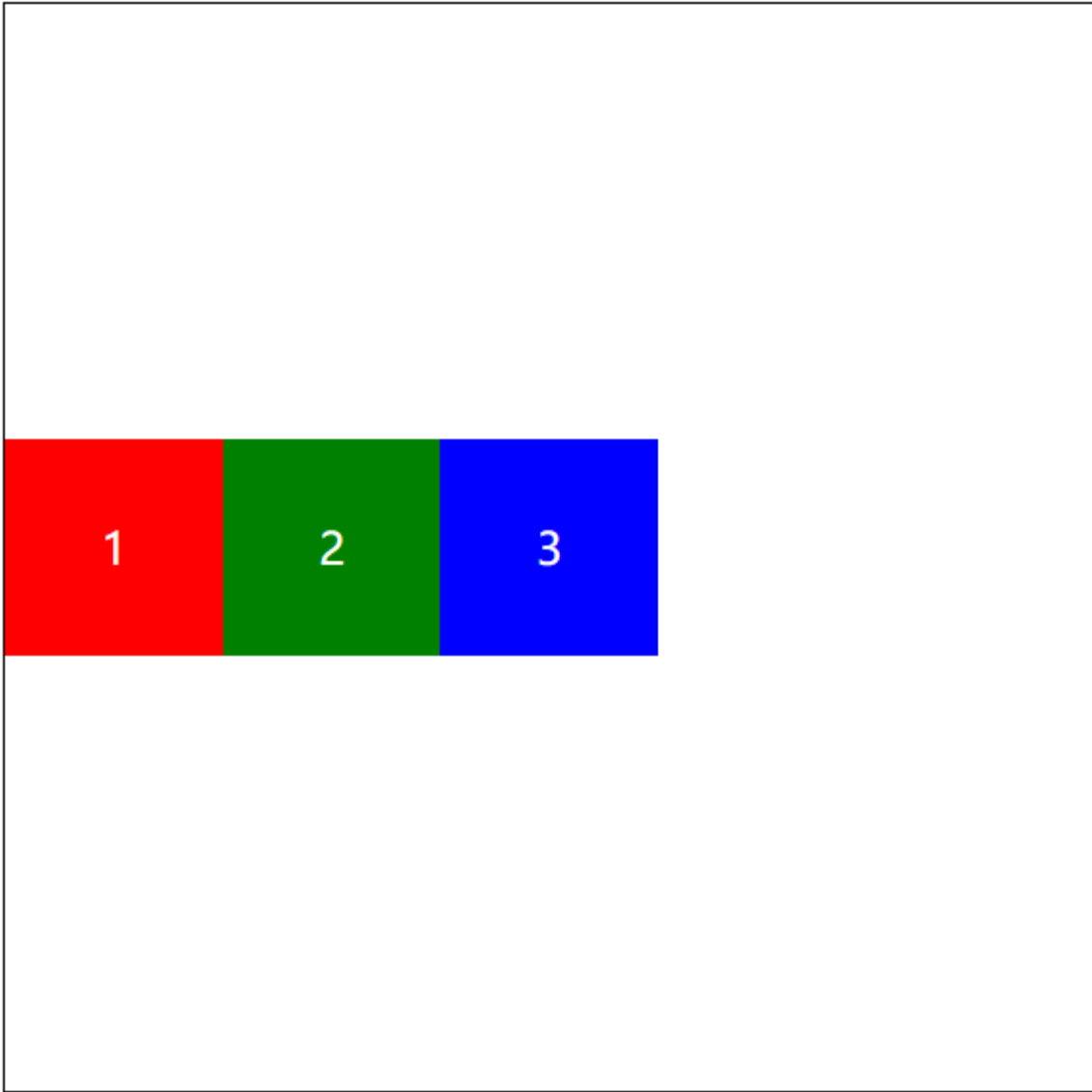
代码:

```
#box{  
    width: 500px;  
    height: 500px;  
    margin: 300px auto;  
    border: 1px solid black;  
    /*设置弹性布局 */  
    display: flex;  
    /*主轴方向 */  
    flex-direction: row;  
    /*设置 侧轴对齐方向 */  
    align-items: stretch;  
}
```

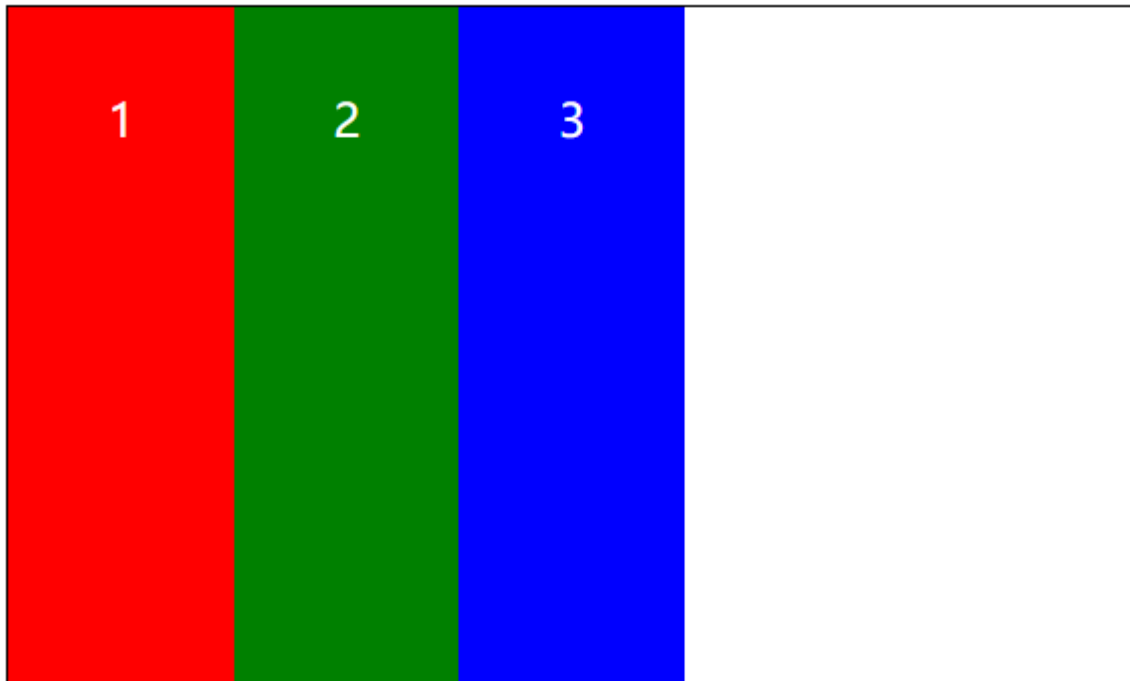
flex-end 终点开始依次排列



center 沿侧轴居中排列，垂直居中



stretch 默认效果,如果指定侧轴大小的属性值为'auto', 弹性盒子沿着侧轴线被拉伸至铺满容器



代码:

```
#box{
    width: 500px;
    height: 300px;
    margin: 300px auto;
    border: 1px solid black;
    /*设置弹性布局 */
    display: flex;
    /*主轴方向 */
    flex-direction: row;
    /*设置 侧轴对齐方向 */
    align-items: stretch;
}

#box div{
    /*子元素侧轴为auto*/
    height: auto;
    width: 100px;
    text-align: center;
    color: white;
    font-size: 20px;
    line-height: 100px;
}
```

baseline 行内轴与侧轴为同一条时, 值与flex-start效果一样



疑问：当主轴为 垂直方向时，效果又会怎样呢？

4.5 flex-wrap属性

flex-wrap 属性用于弹性盒子内的元素溢出时换行排列。

属性值	特点
nowrap	默认， 弹性容器为单行
wrap	元素溢出时换行， 从盒子顶部开始排列， 第一 行在上方
wrap-reverse	元素溢出时换行， 从盒子低部开始排列， 第一行在下发

nowrap 默认， 弹性容器为单行



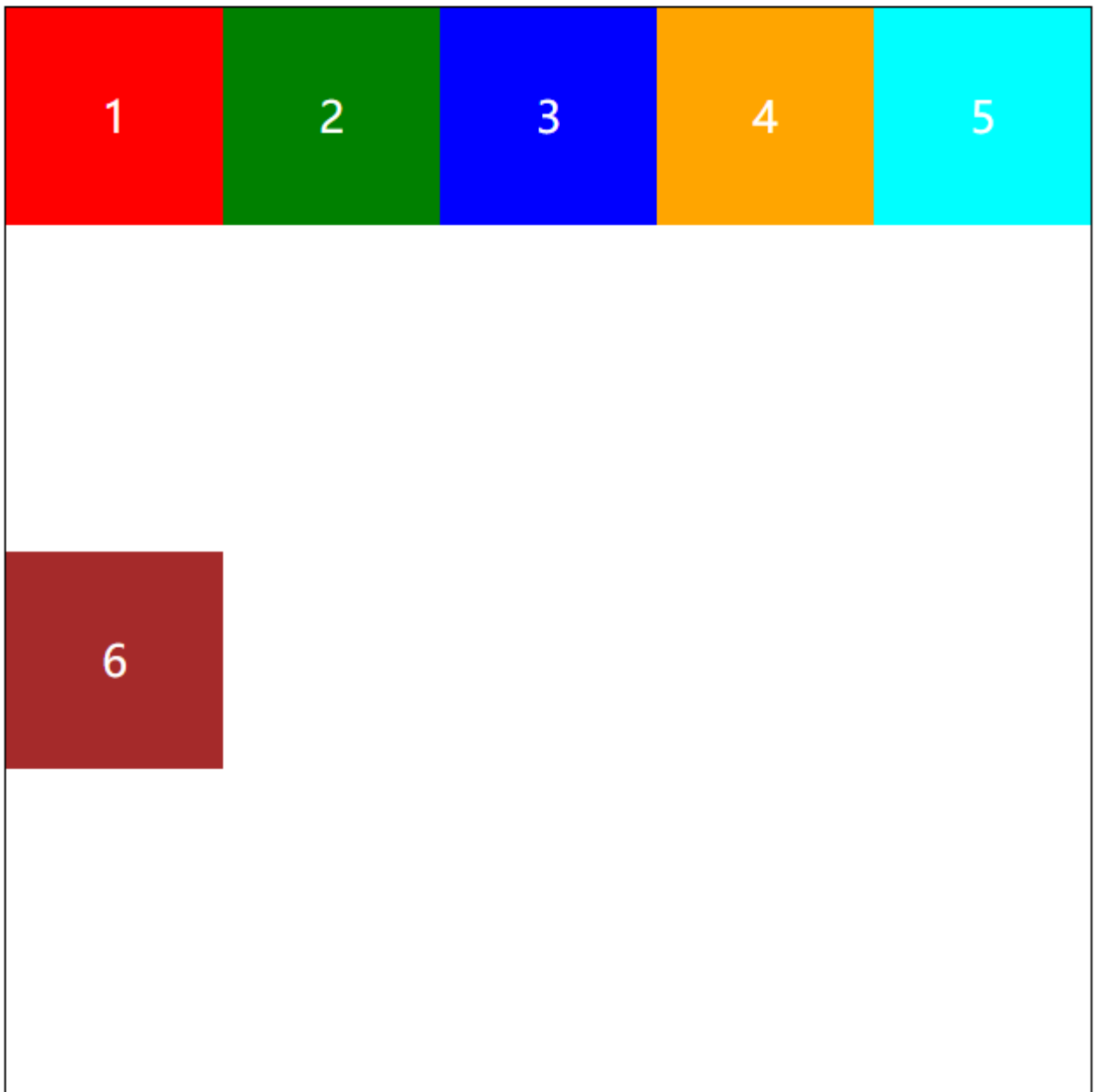
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>弹性盒子</title>
    <style>
      *{
        padding: 0;
        margin: 0;
      }
      #box{
        width: 500px;
        height: 500px;
        margin: 300px auto;
        border: 1px solid black;
        /*设置弹性布局 */
      }
    </style>
  </head>
  <body>
    <div id="box">
      <div>1</div>
      <div>2</div>
      <div>3</div>
      <div>4</div>
      <div>5</div>
      <div>6</div>
    </div>
  </body>
</html>
```

```

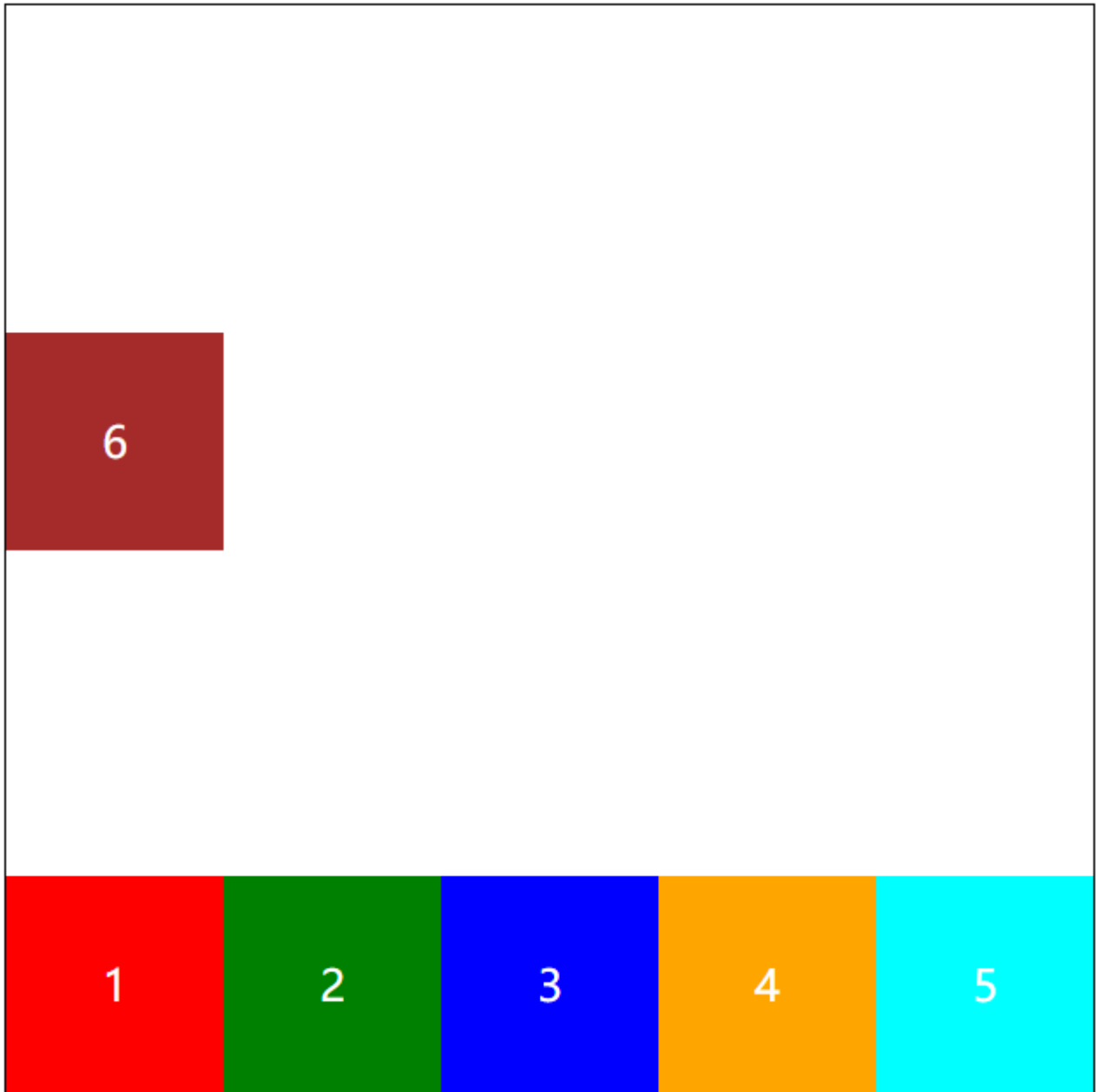
        display: flex;
        /*主轴方向 */
        flex-direction: row;
        /*设置 换行 */
        flex-wrap: nowrap;
    }
    #box div{
        height: 100px;
        width: 100px;
        text-align: center;
        color: white;
        font-size: 20px;
        line-height: 100px;
    }
    #box div:nth-child(1){
        background-color: red;
    }
    #box div:nth-child(2){
        background-color: green;
    }
    #box div:nth-child(3){
        background-color: blue;
    }
    #box div:nth-child(4){
        background-color: orange;
    }
    #box div:nth-child(5){
        background-color: cyan;
    }
    #box div:nth-child(6){
        background-color: brown;
    }
</style>
</head>
<body>
    <div id="box">
        <div>1</div>
        <div>2</div>
        <div>3</div>
        <div>4</div>
        <div>5</div>
        <div>6</div>
    </div>
</body>
</html>

```

wrap 元素溢出时换行，从盒子顶部开始排列，第一行在上方



wrap-reverse 元素溢出时换行，从盒子低部开始排列，第一行在下发



4.6 flex-flow属性

flex-flow属性是 flex-direction 和 flex-wrap 属性的复合属性，其写法可以为{flex-flow:row-reverse wrap;}

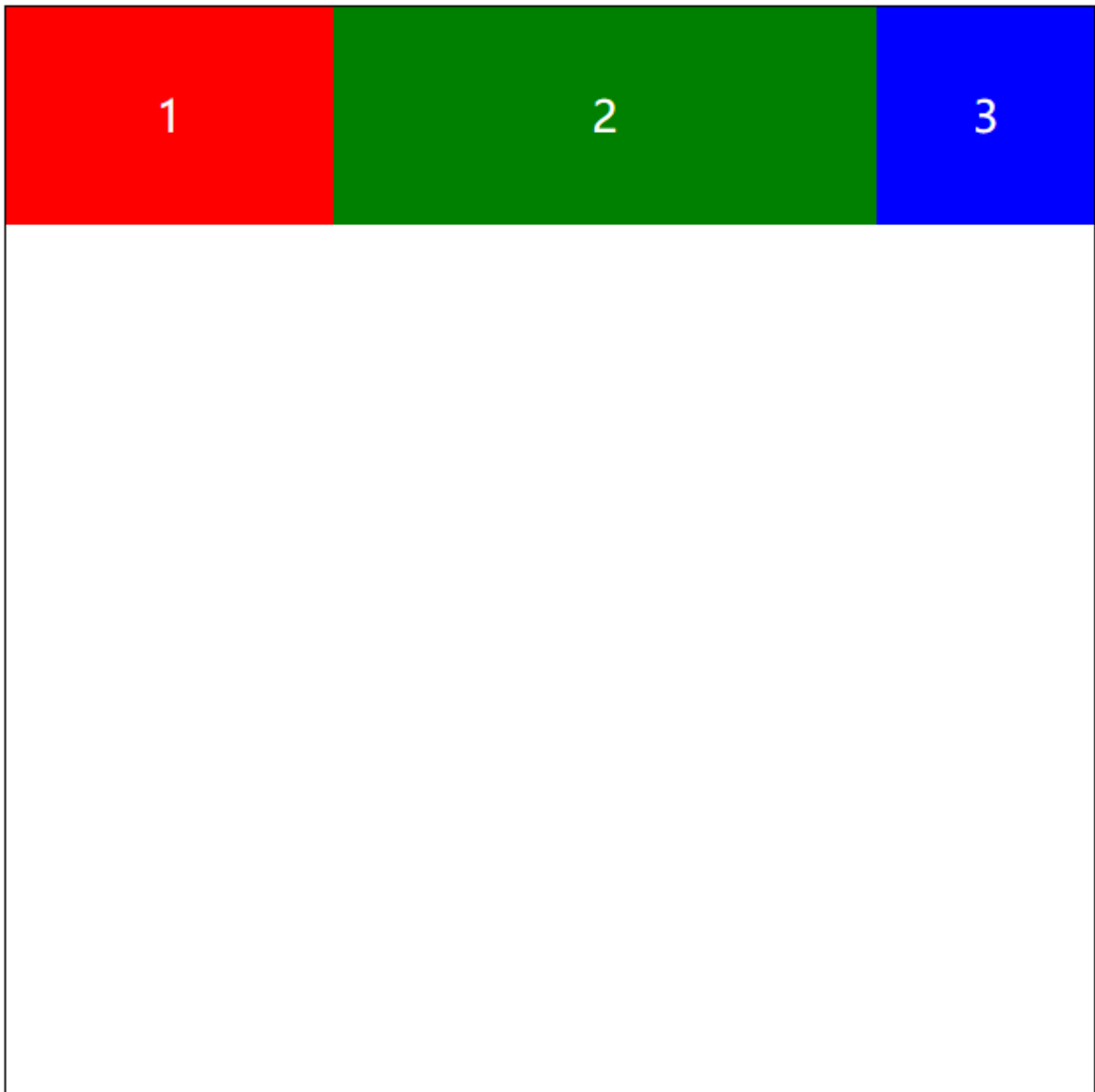


```
#box{  
    width: 500px;  
    height: 500px;  
    margin: 300px auto;  
    border: 1px solid black;  
    /*设置弹性布局 */  
    display: flex;  
    /*设置 flex-flow */  
    flex-flow:column wrap;  
}
```

4.7 flex项目属性

4.7.1 flex-grow属性

flex-grow属性值越大代表所占父元素位置内越大。

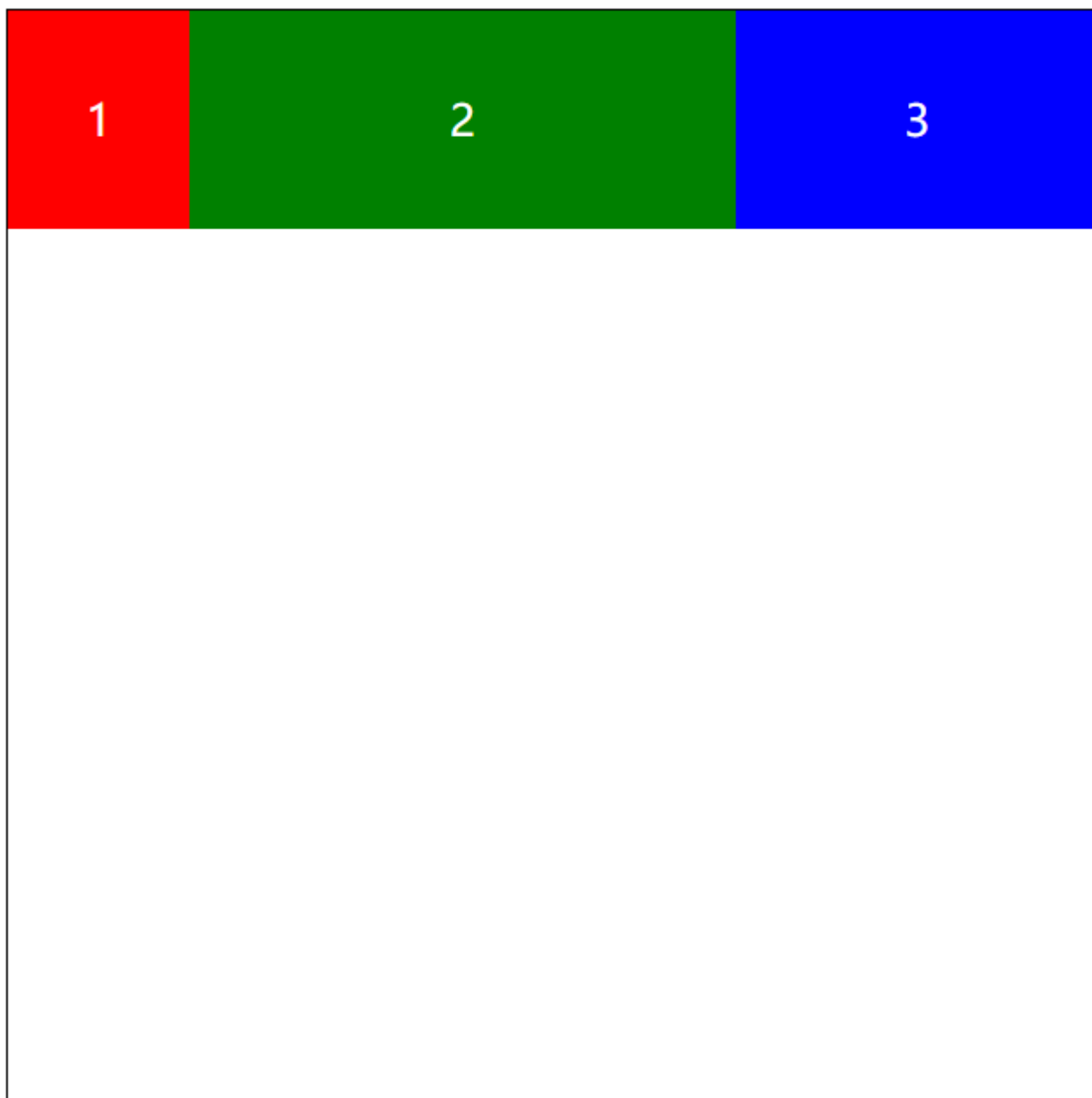


```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>弹性盒子</title>
    <style>
      *{
        padding: 0;
        margin: 0;
      }
      #box{
        width: 500px;
        height: 500px;
        margin: 300px auto;
        border: 1px solid black;
      }
    </style>
  </head>
  <body>
    <div id="box">
      <div>1</div>
      <div>2</div>
      <div>3</div>
    </div>
  </body>
</html>
```

```
        /*设置弹性布局 */
        display: flex;
        /*设置 flex-flow */
        flex-flow: row wrap;
    }
    #box div{
        height: 100px;
        width: 100px;
        text-align: center;
        color: white;
        font-size: 20px;
        line-height: 100px;
    }
    #box div:nth-child(1){
        background-color: red;
    }
    #box div:nth-child(2){
        background-color: green;
    }
    #box div:nth-child(3){
        background-color: blue;
    }
    .one{
        flex-grow: 1;
    }
    .two{
        flex-grow: 3;
    }
</style>
</head>
<body>
    <div id="box">
        <div class="one">1</div>
        <div class="two">2</div>
        <div>3</div>
    </div>
</body>
</html>
```

4.7.2 flex属性

flex 属性用于指定弹性子元素怎么分配空间，另外注意的一点是子元素的高是继承父元素的。



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>弹性盒子</title>
    <style>
      *{
        padding: 0;
        margin: 0;
      }
      #box{
        width: 500px;
        height: 500px;
        margin: 300px auto;
        border: 1px solid black;
        /*设置弹性布局 */
      }
    </style>
  </head>
  <body>
    <div id="box">
      <div>1</div>
      <div>2</div>
      <div>3</div>
    </div>
  </body>
</html>
```

```
        display: flex;
        /*设置 flex-flow */
        /*flex-flow: row wrap;*/
    }
    #box div{
        height: 100px;
        width: 100px;
        text-align: center;
        color: white;
        font-size: 20px;
        line-height: 100px;
    }
    #box div:nth-child(1){
        background-color: red;
    }
    #box div:nth-child(2){
        background-color: green;
    }
    #box div:nth-child(3){
        background-color: blue;
    }
    .one{
        flex:1;
    }
    .two{
        flex:3;
    }
    .three{
        flex:2;
    }
</style>
</head>
<body>
    <div id="box">
        <div class="one">1</div>
        <div class="two">2</div>
        <div class="three">3</div>
    </div>
</body>
</html>
```