

Flex布局（Flexible Box）是CSS3中一种强大的布局模型，特别适合构建一维布局（行或列）。它通过简单的属性设置，能高效控制子元素的排列、对齐和空间分配。

练习1：启动flex容器布局

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    *{
      margin: 0;
      padding: 0;
    }

    .wrap{
      display: flex;
      /* 使用了display: flex 属性后，父元素不设置宽度，宽度就是100%；不会被子元素宽度撑开； */
      flex-wrap: wrap;
      /* 允许盒子换行 */
      background-color: #ff0000;
    }

    .wrap1{
      display: inline-flex;
      /* 使用了display: inline-flex 属性后，父元素不设置宽度，宽度就是所有的子元素宽度之和，会被子元素宽度撑开，实现宽度自适应； */
      background-color: #0000ff;
      margin-top: 20px;
    }

    .item{
      width: 100px;
      height: 100px;
      background: #000000;
    }
  </style>
</head>
<body>
  <div class="wrap">
    <div class="item"></div>
    <div class="item"></div>
    <div class="item"></div>
  </div>

  <div class="wrap1">
    <div class="item"></div>
    <div class="item"></div>
    <div class="item"></div>
  </div>
</body>
</html>
```

好的，接下来我们来学习flex的属性（控制x轴/y轴）

练习2：主轴方向flex-direction

- flex-direction: row; /* 默认：水平从左到右 */
- flex-direction: row-reverse; /* 水平从右到左 */
- flex-direction: column; /* 垂直从上到下 */
- flex-direction: column-reverse; /* 垂直从下到上 */

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15      /* 默认为row，从左到右 */
16    }
17
18    .item{
19      width: 100px;
20      height: 100px;
21      background: #000000;
22      margin-left: 20px;
23    }
24  </style>
25 </head>
26 <body>
27   <div class="wrap">
28     <div class="item"></div>
29     <div class="item"></div>
30     <div class="item"></div>
31   </div>
32 </body>
33 </html>
```

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15      flex-direction: row-reverse;
16      /* 从右到左排列 */
17    }
18
19    .item{
20      width: 100px;
21      height: 100px;
22      background: #black;
23      margin-left: 20px;
24    }
25  </style>
26 </head>
27 <body>
28   <div class="wrap">
29     <div class="item"></div>
30     <div class="item"></div>
31     <div class="item"></div>
32   </div>
33 </body>
34 </html>
```



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15      flex-direction: column;
16      /* 从下到上排列 */
17    }
18
19    .item{
20      width: 100px;
21      height: 100px;
22      background: #rgb(6, 108, 243);
23      margin: 20px;
24    }
25  </style>
26 </head>
27 <body>
28   <div class="wrap">
29     <div class="item">盒子1</div>
30     <div class="item">盒子2</div>
31     <div class="item">盒子3</div>
32   </div>
33 </body>
34 </html>
```

盒子1

盒子2

盒子3

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15      flex-direction: column-reverse;
16      /* 从下到上排列 */
17    }
18
19    .item{
20      width: 100px;
21      height: 100px;
22      background: #rgb(6, 108, 243);
23      margin: 20px;
24    }
25  </style>
26 </head>
27 <body>
28   <div class="wrap">
29     <div class="item">盒子1</div>
30     <div class="item">盒子2</div>
31     <div class="item">盒子3</div>
32   </div>
33 </body>
34 </html>
```

盒子3

盒子2

盒子1

练习3: 换行方式 flex-wrap

- flex-wrap: nowrap; /* 默认: 不换行 (可能压缩项目) */
- flex-wrap: wrap; /* 换行 (第一行在上方) 当子元素的总宽度超过容器的宽度时, 子元素会自动换行 */
- flex-wrap: wrap-reverse; /* 换行 (第一行在下方) */

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15      flex-wrap: wrap;
16      /* 换行：不换行（默认） */
17    }
18
19    .item{
20      margin: 20px;
21      background-color: #2600ff;
22    }
23  </style>
24 </head>
25 <body>
26   <div class="wrap">
27     <div class="item">盒子1</div>
28     <div class="item">盒子2</div>
29     <div class="item">盒子3</div>
30   </div>
31 </body>
32 </html>
```

盒子1 盒子2 盒子3

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15      flex-wrap: wrap;
16      /* 换行（第一行在上）：当子元素的总宽度超过容器的宽度时，子元素会自动换行 */
17      background-color: #000000;
18      width: 1000px;
19    }
20
21    .item{
22      margin: 20px;
23      background-color: #2600ff;
24      width: 900px;
25    }
26  </style>
27 </head>
28 <body>
29   <div class="wrap">
30     <div class="item">盒子1</div>
31     <div class="item">盒子2</div>
32     <div class="item">盒子3</div>
33   </div>
34 </body>
35 </html>
```

盒子1
盒子2
盒子3

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15      flex-wrap: wrap-reverse;
16      /* 换行（第一行在下）：当子元素的总宽度超过容器的宽度时，子元素会自动换行 */
17      background-color: #000000;
18      width: 1000px;
19    }
20
21    .item{
22      margin: 20px;
23      background-color: #2600ff;
24      width: 900px;
25    }
26  </style>
27 </head>
28 <body>
29   <div class="wrap">
30     <div class="item">盒子1</div>
31     <div class="item">盒子2</div>
32     <div class="item">盒子3</div>
33   </div>
34 </body>
35 </html>
```

盒子3
盒子2
盒子1

练习3: 主轴对齐 justify-content (用x轴演示)

- justify-content: flex-start; /* 默认: 左对齐 */
- justify-content: flex-end; /* 右对齐 */
- justify-content: center; /* 居中 */
- justify-content: space-between; /* 两端对齐 (项目间等距) */
- justify-content: space-around; /* 项目两侧等距 */
- justify-content: space-evenly; /* 完全等距 (包括两端) */

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15      justify-content: flex-start;
16      /* 左对齐 */
17      background-color: #000000;
18    }
19
20    .item{
21      background-color: #2600ff;
22    }
23  </style>
24 </head>
25 <body>
26   <div class="wrap">
27     <div class="item">盒子1</div>
28     <div class="item">盒子2</div>
29     <div class="item">盒子3</div>
30   </div>
31 </body>
32 </html>
```

盒子1盒子2盒子3

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15      justify-content: flex-end;
16      /* 右对齐 */
17      background-color: #000000;
18    }
19
20    .item{
21      background-color: #2600ff;
22    }
23  </style>
24 </head>
25 <body>
26   <div class="wrap">
27     <div class="item">盒子1</div>
28     <div class="item">盒子2</div>
29     <div class="item">盒子3</div>
30   </div>
31 </body>
32 </html>
```

盒子1盒子2盒子3

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    *{
      margin: 0;
      padding: 0;
    }

    .wrap{
      display: flex;
      justify-content: center;
      /* 居中 */
      background-color: #000000;
    }

    .item{
      background-color: #2600ff;
    }
  </style>
</head>
<body>
  <div class="wrap">
    <div class="item">盒子1</div>
    <div class="item">盒子2</div>
    <div class="item">盒子3</div>
  </div>
</body>
</html>
```

盒子1盒子2盒子3

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15      justify-content: space-between;
16      /*
17       两端对齐（项目间等距）
18       表示在主轴上将子元素均匀分布，使得第一个子元素位于主轴的起点，
19       最后一个子元素位于主轴的终点，而中间子元素则分布在它们之间，间距相等
20       */
21      background-color: #e0e0e0;
22    }
23
24    .item{
25      background-color: #2600ff;
26    }
27  </style>
28 </head>
29 <body>
30   <div class="wrap">
31     <div class="item">盒子1</div>
32     <div class="item">盒子2</div>
33     <div class="item">盒子3</div>
34   </div>
35 </body>
36 </html>

```



```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15      justify-content: space-around;
16      /*
17       项目内侧等距排列
18       在每个子元素（即每个盒子）周围分配相等的空间，
19       这意味着第一个子元素“前面”、最后一个子元素“后面”以及各个子元素“之间”的空间会是相同的
20       */
21      background-color: #e0e0e0;
22    }
23
24    .item{
25      background-color: #2600ff;
26    }
27  </style>
28 </head>
29 <body>
30   <div class="wrap">
31     <div class="item">盒子1</div>
32     <div class="item">盒子2</div>
33     <div class="item">盒子3</div>
34     <div class="item">盒子4</div>
35   </div>
36 </body>
37 </html>

```



```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15      justify-content: space-evenly;
16      /*
17       完全等距（包括两端）排列
18       这意味着，不仅子元素之间会有相同的间距，
19       而且子元素与父容器的左侧和右侧边缘也会有同样的间距。
20       从而在整体上实现更均匀的分布
21       */
22      background-color: #e0e0e0;
23    }
24
25    .item{
26      background-color: #2600ff;
27    }
28  </style>
29 </head>
30 <body>
31   <div class="wrap">
32     <div class="item">盒子1</div>
33     <div class="item">盒子2</div>
34     <div class="item">盒子3</div>
35     <div class="item">盒子4</div>
36   </div>
37 </body>
38 </html>

```



练习4: 交叉轴对齐 align-items

align-items: stretch; /* 默认: 拉伸填满容器高度 */

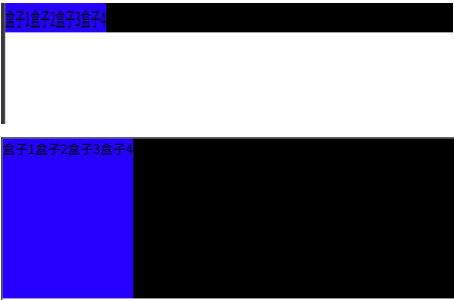
align-items: flex-start; /* 顶部对齐 */

align-items: flex-end; /* 底部对齐 */

align-items: center; /* 垂直居中 */

align-items: baseline; /* 基线对齐 */

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15      align-items: stretch;
16      /*
17       *flex项目会沿着y轴拉伸以填满容器
18       *如果flex项目没有设置属性（在flex容器是横方向时），它们的表现会由父容器flex容器的属性
19       *确定，如果flex容器是y轴方向时，则flex项目没有设置属性，它们的表现会由父容器flex容器的属性
20       */
21      background-color: #000000;
22      height: 200px;
23    }
24
25    .item{
26      background-color: #2000ff;
27    }
28  </style>
29 </head>
30 <body>
31   <div class="wrap">
32     <div class="item">盒子1</div>
33     <div class="item">盒子2</div>
34     <div class="item">盒子3</div>
35     <div class="item">盒子4</div>
36   </div>
37 </body>
38 </html>
```



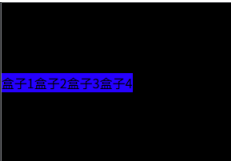
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15      align-items: flex-start;
16      /*
17       *y轴顶部对齐
18       */
19      background-color: #000000;
20      height: 200px;
21    }
22
23    .item{
24      background-color: #2000ff;
25    }
26  </style>
27 </head>
28 <body>
29   <div class="wrap">
30     <div class="item">盒子1</div>
31     <div class="item">盒子2</div>
32     <div class="item">盒子3</div>
33     <div class="item">盒子4</div>
34   </div>
35 </body>
36 </html>
```



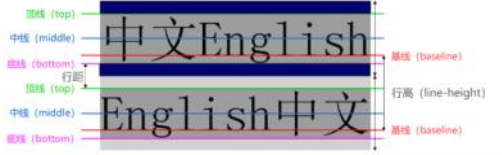
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15      align-items: flex-end;
16      /*
17       *y轴底部对齐
18       */
19      background-color: #000000;
20      height: 200px;
21    }
22
23    .item{
24      background-color: #2000ff;
25    }
26  </style>
27 </head>
28 <body>
29   <div class="wrap">
30     <div class="item">盒子1</div>
31     <div class="item">盒子2</div>
32     <div class="item">盒子3</div>
33     <div class="item">盒子4</div>
34   </div>
35 </body>
36 </html>
```



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15      align-items: center;
16    }
17    /*
18     * 垂直居中
19     */
20    background-color: #000000;
21    height: 200px;
22
23    .item{
24      background-color: #2600ff;
25    }
26  </style>
27 </head>
28 <body>
29   <div class="wrap">
30     <div class="item">盒子1</div>
31     <div class="item">盒子2</div>
32     <div class="item">盒子3</div>
33     <div class="item">盒子4</div>
34   </div>
35 </body>
36 </html>
```



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15      align-items: baseline;
16    }
17    /*
18     * 沿基线对齐
19     * 表示子元素的基线将与 flex 容器的基线对齐。
20     * 这意味着如果子元素的行高或字体大小不同，它们的文本也会在视觉上对齐
21     * 基线通常用于文本对齐，使得文本行的基线在视觉上对齐
22     */
23    background-color: #000000;
24    height: 200px;
25
26    .item{
27      background-color: #2600ff;
28    }
29  </style>
30 </head>
31 <body>
32   <div class="wrap">
33     <div class="item">盒子1</div>
34     <div class="item">盒子2</div>
35     <div class="item">盒子3</div>
36     <div class="item">盒子4</div>
37   </div>
38 </body>
39 </html>
```



练习5: 多行对齐 align-content (多行时才生效)

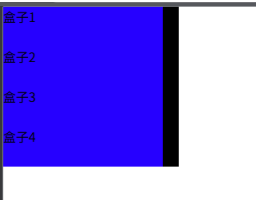
align-content: stretch; /* 拉伸行填满容器 */

align-content: flex-start; /* 多行顶部对齐 */

align-content: center; /* 多行垂直居中 */

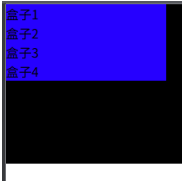
/* 其他值同 justify-content */

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15      flex-wrap: wrap;
16      align-content: stretch;
17      /*
18       *拉伸填满容器
19       * Flex 容器有多行时, align-content: stretch;
20       * 会使每一行的 Flex 子元素在交叉轴方向上拉伸, 以填满整个交叉轴的空间。
21       * 如果没有设置高度, 子元素的高度会自动拉伸以填充 Flex 容器的高度。
22       */
23      background-color: #000000;
24      /* height: 200px; */
25      width: 220px;
26    }
27
28    .item{
29      background-color: #2600ff;
30    }
31  </style>
32 </head>
33 <body>
34   <div class="wrap">
35     <div class="item">盒子1</div>
36     <div class="item">盒子2</div>
37     <div class="item">盒子3</div>
38     <div class="item">盒子4</div>
39   </div>
40 </body>
41 </html>
```

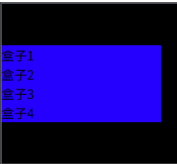


3. 两者的区别			
属性	作用范围	生效条件	具体效果
align-items: stretch;	每个 Flex 子元素	无论容器是否有多行	子元素沿着交叉轴拉伸, 填满容器的交叉轴空间。
align-content: stretch;	Flex 容器内部的多行内容	容器有多行 (flex-wrap: wrap);	所有行的 Flex 子元素沿着交叉轴拉伸, 填满容器的交叉轴空间。

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15      flex-wrap: wrap;
16      align-content: flex-start;
17      /*
18       *多行顶部对齐
19       */
20      background-color: #000000;
21      height: 200px;
22      width: 220px;
23    }
24
25    .item{
26      background-color: #2600ff;
27      width: 200px;
28    }
29  </style>
30 </head>
31 <body>
32   <div class="wrap">
33     <div class="item">盒子1</div>
34     <div class="item">盒子2</div>
35     <div class="item">盒子3</div>
36     <div class="item">盒子4</div>
37   </div>
38 </body>
39 </html>
```



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15      flex-wrap: wrap;
16      align-content: center;
17      /*
18       *多行居中
19       */
20      background-color: #000000;
21      height: 200px;
22      width: 220px;
23    }
24
25    .item{
26      background-color: #2600ff;
27      width: 200px;
28    }
29  </style>
30 </head>
31 <body>
32   <div class="wrap">
33     <div class="item">盒子1</div>
34     <div class="item">盒子2</div>
35     <div class="item">盒子3</div>
36     <div class="item">盒子4</div>
37   </div>
38 </body>
39 </html>
```



接下来学习在flex容器中对子元素进行操作

练习6: 调整顺序 order (控制单个项目)

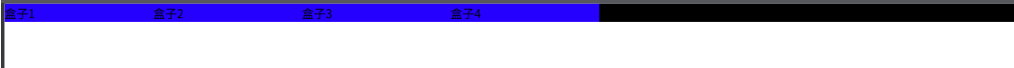
order: 2; /* 默认0, 数值越小越靠前 */

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15
16      background-color: #000000;
17    }
18
19    .item1{
20      background-color: #2600ff;
21      width: 200px;
22      order: 1
23    }
24
25    .item2{
26      background-color: #2600ff;
27      width: 200px;
28      order: 3
29    }
30
31    .item3{
32      background-color: #2600ff;
33      width: 200px;
34      order: 2
35    }
36
37    .item4{
38      background-color: #2600ff;
39      width: 200px;
40      order: 4
41    }
42  </style>
43 </head>
44 <body>
45   <div class="wrap">
46     <div class="item1">盒子1</div>
47     <div class="item2">盒子2</div>
48     <div class="item3">盒子3</div>
49     <div class="item4">盒子4</div>
50   </div>
51 </body>
52 </html>
```



练习7: 缩放比例 flex-grow

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     *{
9       margin: 0;
10      padding: 0;
11    }
12
13    .wrap{
14      display: flex;
15
16      background-color: #000000;
17    }
18
19    .item1{
20      background-color: #2600ff;
21      width: 200px;
22    }
23
24    .item2{
25      background-color: #2600ff;
26      width: 200px;
27    }
28
29    .item3{
30      background-color: #2600ff;
31      width: 200px;
32    }
33
34    .item4{
35      background-color: #2600ff;
36      width: 200px;
37      flex-grow: 1;
38      /* 默认0 (不拉伸)
39      按比例分配剩余空间 */
40    }
41  </style>
42 </head>
43 <body>
44   <div class="wrap">
45     <div class="item1">盒子1</div>
46     <div class="item2">盒子2</div>
47     <div class="item3">盒子3</div>
48     <div class="item4">盒子4</div>
49   </div>
50 </body>
51 </html>
```



练习8: 收缩比例 flex-shrink

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    {
      margin: 0;
      padding: 0;
    }

    .wrap{
      display: flex;
      background-color: #800000;
      width: 500px;
    }

    .item1{
      background-color: #8000ff;
      width: 200px;
    }

    .item2{
      background-color: #8000ff;
      width: 200px;
      flex: 2;
      flex-shrink: 1;
    }
    /*
    这是一个默认值，flex项目本身就有时
    表示这个flex项目在需要时可以缩小，它的最小值是1。
    这里需要注意的是，item2需要缩小到flex项目的高度的
    差不多两倍的长度，它实际不是（本来flex项目是500px，这个项目的值是100px），它按照比例缩小生成它到这个项目
    */
    .item3{
      background-color: #8000ff;
      width: 200px;
    }
    .item4{
      background-color: #8000ff;
      width: 200px;
    }
  }
</style>
</head>
<body>
  <div class="wrap">
    <div class="item">盒子1</div>
    <div class="item">盒子2</div>
    <div class="item">盒子3</div>
    <div class="item">盒子4</div>
  </div>
</body>
</html>
```

