



# CSS3 Flexible Box Layout



jCodeLife

0.224

2020.08.27 17:31:40 字数 1,378 阅读 17

[编辑文章](#)

在传统的布局中，我们常常使用定位`position`、`margin`、浮动`float`等来实现页面的布局，虽然功能可以实现没有问题，但是它有缺点不够灵活，用起来比较费劲。比如我要实现多个同级元素在父级容器中水平垂直居中，`float`做不到，可以通过`position`定位来实现，但那当设置`position: absolute`时，所被设置的元素都会因为脱离文档流而重叠在一起，每个元素的有确切的位置，那就得计算每个元素应该在的位置，这样不太好吧，一来，计算太麻烦，二来后面如果父级再增加或者删除一些子元素那又得重新计算，那想想就头大，而且这样的情况还不少。

```
1 <head>
2   <style type="text/css">
3     .wrapper{
4       width: 400px;
5       height: 250px;
6       background-color: pink;
7       margin: 150px auto;
8       overflow: hidden;
9       position: relative;
10    }
11    .wrapper *{
12      box-sizing: border-box;
13      background-color: deeppink;
14      border: 1px solid #0e0;
15      width: 80px;
16      height: 80px;
17      text-align: center;
18      line-height: 80px;
19      float: left;
20      position: absolute;
21    }
22    /*设置位置*/
23    .wrapper div:nth-of-type(1){
24      left: 80px;
25      top: 85px;
26    }
27    .wrapper div:nth-of-type(2){
28      left: 160px;
29      top: 85px;
30    }
31    .wrapper div:nth-of-type(3){
32      left: 240px;
33      top: 85px;
34    }
35  </style>
36 </head>
37 <body>
38   <div class="wrapper">
39     <div>1</div>
40     <div>2</div>
41     <div>3</div>
42   </div>
43 </body>
```

## 推荐阅读

[为什么总有人觉得前端很简单？](#)

阅读 175

[面试了几个前端，给爷整哭了！](#)

阅读 56,709

[VUE响应式原理](#)

阅读 147

[Vue干货小技巧——学会再也不做加班狗](#)

阅读 2,234

[vue优化页面](#)

阅读 1,143



当然也可以给要居中的元素添加一个额外的父级，通过margin来实现

```
1 <head>
2   <style type="text/css">
3     .wrapper {
4       width: 400px;
5       height: 250px;
6       background-color: pink;
7       margin: 150px auto;
8       overflow: hidden;
9     }
10
11     .wrapper section * {
12       box-sizing: border-box;
13       background-color: deeppink;
14       border: 1px solid #0e0;
15       width: 80px;
16       height: 80px;
17       text-align: center;
18       line-height: 80px;
19       float: left;
20     }
21     .wrapper section{
22       margin:0 auto;
23       width: 240px;
24       height: 80px;
25       margin-top:75px;
26     }
27     .clearFloat::after{
28       clear: both;
29       content: "";
30       display: block;
31     }
32   </style>
33 </head>
34 <body>
35   <div class="wrapper">
36     <section class="clearFloat">
37       <div>1</div>
38       <div>2</div>
39       <div>3</div>
40     </section>
41   </div>
42 </body>
```

效果跟上面一模一样，这里是通过给父级设置margin:0 auto设置水平居中，通过调整margin-top让其水平居中。但是这样也麻烦，一 得清除浮动带来的影响，二 添加的父级得计算一下宽高，三 也是最不好的 为了样式修改了HTML结构。

正是有这样那样的问题，在CSS2升级到CSS3时新增了Flex布局方式，这种方式也叫弹性布局、伸缩布局。一起来看看

由于属性比较多我将其分父级容器和子级项两类记忆，整理如下图：

## 推荐阅读

为什么总有人觉得前端很简单？

阅读 175

面试了几个前端，给爷整哭了！

阅读 56,709

VUE响应式原理

阅读 147

Vue干货小技巧——学会再也不做加班狗

阅读 2,234

vue优化页面

阅读 1,143



在讲之前我想先铺垫一个知识点，那就是主轴和侧轴

CSS3中引入了轴向的概念，包括主轴和侧轴。这跟我们平时说的水平X轴和垂直Y轴有点区别，在这轴向是可以改变的，主轴和侧轴不一定对应就是水平或者垂直方向。

在flex布局中，容器的轴向是主还是侧是由flex容器中的属性设定，轴的两端起点和终点分别对应flex-start和flex-end，子级项目对齐和排列方式均以设定的轴向为基准。在默认情况下主轴就是水平方向，起点在左，终点在右；侧轴就是垂直方向，起点在上，终点在下  
通过 `display: flex` 即可激活元素的Flex特性，注意仅对容器内的第一层元素起作用，在容器外或者容器内的孙子辈的不起作用。

下面开始讲解Flex布局中用到的属性及属性值

首先是父级容器上的属性

### 1. flex-direction

flex-direction 用于设置主轴方向，对应的值有row、row-reverse、column、column-reverse，分别表示水平、水平翻转、垂直、垂直翻转

```
1 <head>
2   <style type="text/css">
3     body>div{
4       float: left;
5     }
6     .wrapper {
7       width: 280px;
8       height: 280px;
9       background-color: rgb(247, 176, 188);
10      margin: 10px;
11      overflow: hidden;
12      display: flex;
13    }
14    .wrapper div{
15      box-sizing: border-box;
16      background-color: rgb(255, 40, 154);
17      width: 60px;
18      height: 60px;
19      text-align: center;
20      line-height: 60px;
21    }
22    .wrapper div:nth-of-type(2n){
23      background-color: rgb(167, 82, 127);
24    }
25    .wrapper.one{
26      flex-direction: row;
27    }
28    .wrapper.two{
29      flex-direction: row-reverse;
30    }
31    .wrapper.three{
32      flex-direction: column;
33    }
34    .wrapper.four{
35      flex-direction: column-reverse;
36    }
37  </style>
38 </head>
39
40 <body>
41   <div class="wrapper one">
42     <div>1</div>
43     <div>2</div>
44     <div>3</div>
45     <div>4</div>
46   </div>
47   <div class="wrapper two">
48     <div>1</div>
49     <div>2</div>
50     <div>3</div>
51     <div>4</div>
52   </div>
53   <div class="wrapper three">
```

### 推荐阅读

为什么总有人觉得前端很简单？

阅读 175

面试了几个前端，给爷整哭了！

阅读 56,709

VUE响应式原理

阅读 147

Vue干货小技巧——学会再也不做加班狗

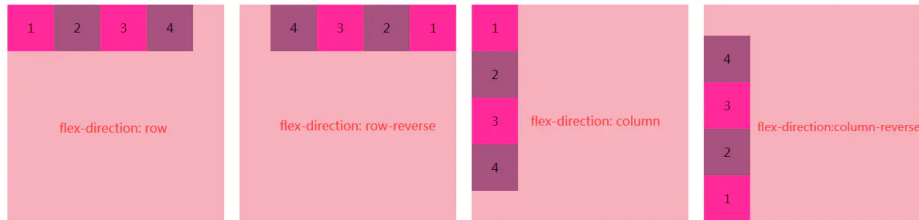
阅读 2,234

vue优化页面

阅读 1,143



```
59 <div class="wrapper four">
60   <div>1</div>
61   <div>2</div>
62   <div>3</div>
63   <div>4</div>
64 </div>
65 </body>
```



结果如图

## 2. flex-wrap

**flex-wrap**用于控制子项排列超出边界时是否换行展示；取值可以是**nowrap**、**wrap**、**wrap-reverse**，分别表示不换行、换行、换行并且反转行

```
1 <head>
2   <style type="text/css">
3     body>div{
4       float: left;
5     }
6     .wrapper {
7       width: 280px;
8       height: 180px;
9       background-color: rgb(247, 176, 188);
10      margin: 10px;
11      overflow: hidden;
12      display: flex;
13    }
14    .wrapper div{
15      box-sizing: border-box;
16      background-color: rgb(255, 40, 154);
17      width: 60px;
18      height: 60px;
19      text-align: center;
20      line-height: 60px;
21    }
22    .wrapper div:nth-of-type(2n){
23      background-color: rgb(167, 82, 127);
24    }
25    .wrapper.one{
26      flex-wrap: nowrap;
27    }
28    .wrapper.two{
29      flex-wrap: wrap;
30    }
31    .wrapper.three{
32      flex-wrap: wrap-reverse;
33    }
34  </style>
35 </head>
36
37 <body>
38   <div class="wrapper one">
39     <div>1</div>
40     <div>2</div>
41     <div>3</div>
42     <div>4</div>
43     <div>5</div>
44     <div>6</div>
45     <div>7</div>
46   </div>
47   <div class="wrapper two">
48     <div>1</div>
```

## 推荐阅读

为什么总有人觉得前端很简单？

阅读 175

面试了几个前端，给爷整哭了！

阅读 56,709

VUE响应式原理

阅读 147

Vue干货小技巧——学会再也不做加班狗

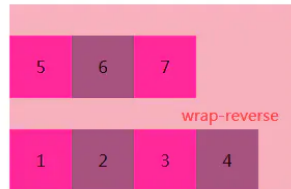
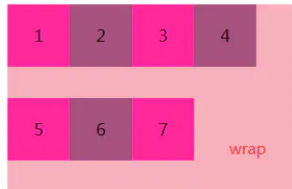
阅读 2,234

vue优化页面

阅读 1,143



```
55     <div>7</div>
56   </div>
57   <div class="wrapper three">
58     <div>1</div>
59     <div>2</div>
60     <div>3</div>
61     <div>4</div>
62     <div>5</div>
63     <div>6</div>
64     <div>7</div>
65   </div>
66 </body>
```



结果展示

### 3. flex-flow

**flex-flow**表示流动，即文档流，其实就是**flex-direction**和**flex-wrap**的复合写法

```
1  <head>
2    <style type="text/css">
3      body>div{
4        float: left;
5      }
6      .wrapper {
7        width: 280px;
8        height: 180px;
9        background-color: rgb(247, 176, 188);
10       margin: 10px;
11       overflow: hidden;
12       display: flex;
13     }
14     .wrapper div{
15       box-sizing: border-box;
16       background-color: rgb(255, 40, 154);
17       width: 60px;
18       height: 60px;
19       text-align: center;
20       line-height: 60px;
21     }
22     .wrapper div:nth-of-type(2n){
23       background-color: rgb(167, 82, 127);
24     }
25     .wrapper.one{
26       flex-flow: nowrap row;
27     }
28   </style>
29 </head>
30
31 <body>
32   <div class="wrapper one">
33     <div>1</div>
34     <div>2</div>
35     <div>3</div>
36     <div>4</div>
37     <div>5</div>
38     <div>6</div>
39     <div>7</div>
40   </div>
41 </body>
```

### 推荐阅读

为什么总有人觉得前端很简单？

阅读 175

面试了几个前端，给爷整哭了！

阅读 56,709

VUE响应式原理

阅读 147

Vue干货小技巧——学会再也不做加班狗

阅读 2,234

vue优化页面

阅读 1,143



默认就是不换行 主轴水平从左往右

#### 4. justify-content

justify-content用于设置主轴方向的对齐方式，取值有5个，分别是：flex-start、flex-end、center、space-

between、space-around，表示主轴起点、主轴终点、居中、空白在元素与元素之间、空白环绕着元素

```
1 <head>
2   <style type="text/css">
3     body>div{
4       float: left;
5     }
6     .wrapper {
7       width: 280px;
8       height: 150px;
9       background-color: rgb(247, 176, 188);
10      margin: 10px;
11      overflow: hidden;
12      display: flex;
13    }
14    .wrapper div{
15      box-sizing: border-box;
16      background-color: rgb(255, 40, 154);
17      width: 60px;
18      height: 60px;
19      text-align: center;
20      line-height: 60px;
21    }
22    .wrapper div:nth-of-type(2n){
23      background-color: rgb(167, 82, 127);
24    }
25    .wrapper.one{
26      justify-content: flex-start;
27    }
28    .wrapper.two{
29      justify-content: flex-end;
30    }
31    .wrapper.three{
32      justify-content: center;
33    }
34    .wrapper.four{
35      justify-content: space-between;
36    }
37    .wrapper.five{
38      justify-content: space-around;
39    }
40  </style>
41 </head>
42
43 <body>
44   <div class="wrapper one">
45     <div>1</div>
46     <div>2</div>
47     <div>3</div>
48   </div>
49   <div class="wrapper two">
50     <div>1</div>
51     <div>2</div>
52     <div>3</div>
53   </div>
54   <div class="wrapper three">
55     <div>1</div>
56     <div>2</div>
57     <div>3</div>
58   </div>
59   <div class="wrapper four">
60     <div>1</div>
61     <div>2</div>
62     <div>3</div>
63   </div>
64   <div class="wrapper five">
```

#### 推荐阅读

为什么总有人觉得前端很简单？

阅读 175

面试了几个前端，给爷整哭了！

阅读 56,709

VUE响应式原理

阅读 147

Vue干货小技巧——学会再也不做加班狗

阅读 2,234

vue优化页面

阅读 1,143

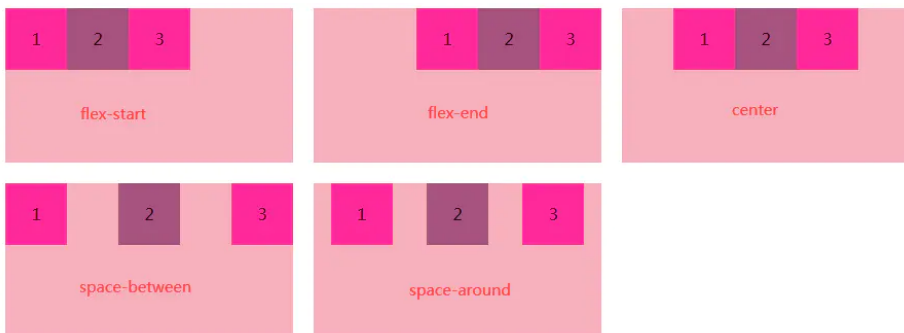


写下你的评论...

评论 0

赞 2





对应结果

## 5. align-items

`align-items`与`justify-content`对应，`justify-content`表示主轴方向的对齐方式，而`align-items`表示侧轴上的对齐方式。但是取值有点不同，`align-items`的取值有：`flex-start`、`flex-end`、`center`、`baseline`、`stretch`(默认值)，分别表示起点对齐、终点对齐、居中对齐、基线对齐、拉伸对齐

```
1 <head>
2   <style type="text/css">
3     body>div{
4       float: left;
5     }
6     .wrapper {
7       width: 280px;
8       height: 200px;
9       background-color: rgb(247, 176, 188);
10      margin: 10px;
11      overflow: hidden;
12      display: flex;
13      flex-flow: wrap;
14    }
15    .wrapper div{
16      box-sizing: border-box;
17      background-color: rgb(255, 40, 154);
18      text-align: center;
19      line-height: 60px;
20    }
21    .wrapper div:nth-of-type(2n){
22      background-color: rgb(167, 82, 127);
23    }
24    .wrapper.one{
25      align-items: flex-start;
26    }
27    .wrapper.two{
28      align-items: flex-end;
29    }
30    .wrapper.three{
31      align-items: center;
32    }
33    .wrapper.four{
34      align-items: baseline;
35    }
36    .wrapper.five{
37      align-items: stretch;
38    }
39  </style>
40 </head>
41
42 <body>
43   <div class="wrapper one">
44     <div>1</div>
45     <div>2</div>
46     <div>3</div>
47   </div>
48   <div class="wrapper two">
49     <div>1</div>
```

## 推荐阅读

为什么总有人觉得前端很简单？

阅读 175

面试了几个前端，给爷整哭了！

阅读 56,709

VUE响应式原理

阅读 147

Vue干货小技巧——学会再也不做加班狗

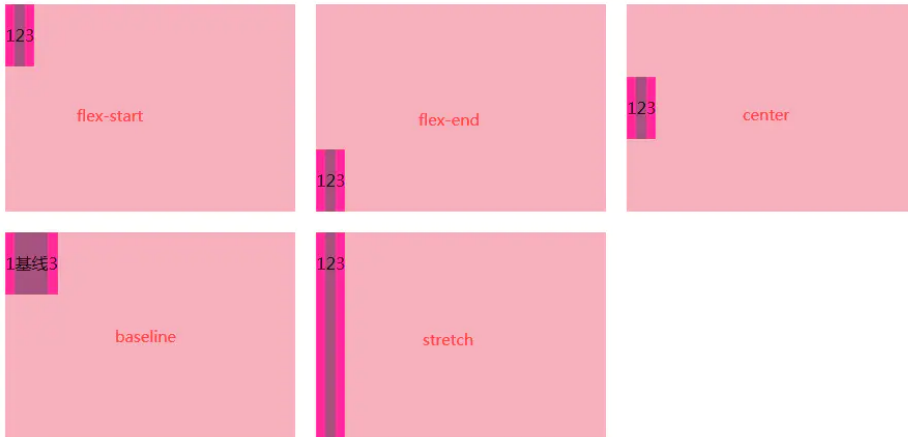
阅读 2,234

vue优化页面

阅读 1,143



```
56     <div>3</div>
57   </div>
58   <div class="wrapper four">
59     <div>1</div>
60     <div>基线</div>
61     <div>3</div>
62   </div>
63   <div class="wrapper five">
64     <div>1</div>
65     <div>2</div>
66     <div>3</div>
67   </div>
68 </body>
```



结果

## 6. align-content

**align-content** 子类多行排列时，子类在侧轴方向的对齐方式。注意如果是一行，该属性无效。即需要设置换行或者换行翻转才行，取值跟**justify-content**一致

```
1  <head>
2    <style type="text/css">
3      body>div{
4        float: left;
5      }
6      .wrapper {
7        width: 280px;
8        height: 200px;
9        background-color: rgb(247, 176, 188);
10       margin: 10px;
11       overflow: hidden;
12       display: flex;
13       flex-flow: wrap;
14     }
15     .wrapper div{
16       box-sizing: border-box;
17       background-color: rgb(255, 40, 154);
18       width: 80px;
19       text-align: center;
20       line-height: 60px;
21     }
22     .wrapper div:nth-of-type(2n){
23       background-color: rgb(167, 82, 127);
24     }
25     .wrapper.one{
26       align-content: flex-start;
27     }
28     .wrapper.two{
29       align-content: flex-end;
30     }
31     .wrapper.three{
```

## 推荐阅读

为什么总有人觉得前端很简单？

阅读 175

面试了几个前端，给爷整哭了！

阅读 56,709

VUE响应式原理

阅读 147

Vue干货小技巧——学会再也不做加班狗

阅读 2,234

vue优化页面

阅读 1,143





```
37     .wrapper.five{
38         align-content: space-around;
39     }
40     </style>
41 </head>
42
43 <body>
44     <div class="wrapper one">
45         <div>1</div>
46         <div>2</div>
47         <div>3</div>
48         <div>4</div>
49         <div>5</div>
50     </div>
51     <div class="wrapper two">
52         <div>1</div>
53         <div>2</div>
54         <div>3</div>
55         <div>4</div>
56         <div>5</div>
57     </div>
58     <div class="wrapper three">
59         <div>1</div>
60         <div>2</div>
61         <div>3</div>
62         <div>4</div>
63         <div>5</div>
64     </div>
65     <div class="wrapper four">
66         <div>1</div>
67         <div>2</div>
68         <div>3</div>
69         <div>4</div>
70         <div>5</div>
71     </div>
72     <div class="wrapper five">
73         <div>1</div>
74         <div>2</div>
75         <div>3</div>
76         <div>4</div>
77         <div>5</div>
78     </div>
79 </body>
```



结果分别对应，flex-start、flex-end、center、space-between、space-around

接着来看子级项目的相关属性，其主要功能是控制子级项的自适应伸缩及分配

#### 1. align-self

align-self用于设置子级项目自身在侧轴上的对齐方式，其实就是单独设置一些特殊子类的侧轴对齐方式，用于覆盖父级得 align-items。所以优先级高于align-items。取值也与align-items一致：flex-start、flex-end、center、baseline、stretch

#### 推荐阅读

为什么总有人觉得前端很简单？

阅读 175

面试了几个前端，给爷整哭了！

阅读 56,709

VUE响应式原理

阅读 147

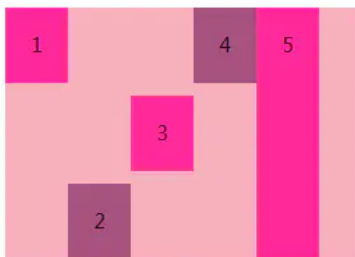
Vue干货小技巧——学会再也不做加班狗

阅读 2,234

vue优化页面

阅读 1,143

```
7      width: 280px;
8      height: 200px;
9      background-color: rgb(247, 176, 188);
10     margin: 10px;
11     overflow: hidden;
12     display: flex;
13     align-items: flex-start;
14   }
15   .wrapper div{
16     box-sizing: border-box;
17     background-color: rgb(255, 40, 154);
18     width: 50px;
19     text-align: center;
20     line-height: 60px;
21   }
22   .wrapper div:nth-of-type(2n){
23     background-color: rgb(167, 82, 127);
24   }
25   .wrapper.one div:nth-of-type(1){
26     align-self: flex-start;
27   }
28   .wrapper.one div:nth-of-type(2){
29     align-self: flex-end;
30   }
31   .wrapper.one div:nth-of-type(3){
32     align-self: center;
33   }
34   .wrapper.one div:nth-of-type(4){
35     align-self: baseline;
36   }
37   .wrapper.one div:nth-of-type(5){
38     align-self: stretch;
39   }
40 </style>
41 </head>
42 <body>
43   <div class="wrapper one">
44     <div>1</div>
45     <div>2</div>
46     <div>3</div>
47     <div>4</div>
48     <div>5</div>
49   </div>
50 </body>
```



结果对应flex-start、flex-end、center、baseline、stretch

## 2. order:n

用于调整特定子类中在所有子类中的排序，n是整数，数值越小，排列越靠前，默认为0，可以是负数

上例中将最后一个放到前面去

```
1 .wrapper.one div:nth-of-type(5){
2   align-self: stretch;
3   order:-1;
4 }
```

## 推荐阅读

为什么总有人觉得前端很简单？

阅读 175

面试了几个前端，给爷整哭了！

阅读 56,709

VUE响应式原理

阅读 147

Vue干货小技巧——学会再也不做加班狗

阅读 2,234

vue优化页面

阅读 1,143



结果

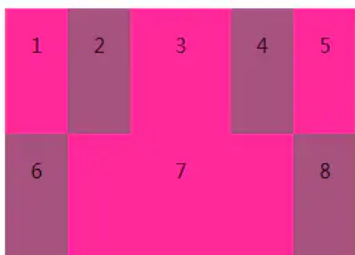
我们知道Flex的特性是自适应伸缩，即按比例分配容器的剩余空间，下面几个属性跟这个有关。

### 3. flex-grow

flex-grow表示子项的扩展比例，默认0 不扩展。注意：行中要存在可分配的剩余空间，子项多行时，以行为单位进行扩展，

即我这行只能在我这行的剩余空间中按比例分

```
1 <head>
2   <style type="text/css">
3     body > div{
4       float: left;
5     }
6     .wrapper {
7       width: 280px;
8       height: 200px;
9       background-color: rgb(247, 176, 188);
10      margin: 10px;
11      overflow: hidden;
12      display: flex;
13      flex-flow: wrap;
14    }
15    .wrapper div{
16      box-sizing: border-box;
17      background-color: rgb(255, 40, 154);
18      width: 50px;
19      text-align: center;
20      line-height: 60px;
21    }
22    .wrapper div:nth-of-type(2n){
23      background-color: rgb(167, 82, 127);
24    }
25    .wrapper.one div:nth-of-type(3){
26      flex-grow: 1;
27    }
28    .wrapper.one div:nth-of-type(7){
29      flex-grow: 1;
30    }
31  </style>
32 </head>
33 <body>
34   <div class="wrapper one">
35     <div>1</div>
36     <div>2</div>
37     <div>3</div>
38     <div>4</div>
39     <div>5</div>
40     <div>6</div>
41     <div>7</div>
42     <div>8</div>
43   </div>
44 </body>
```



将每行的剩余空间作为1份分给最中间的元素

### 推荐阅读

为什么总有人觉得前端很简单？

阅读 175

面试了几个前端，给爷整哭了！

阅读 56,709

VUE响应式原理

阅读 147

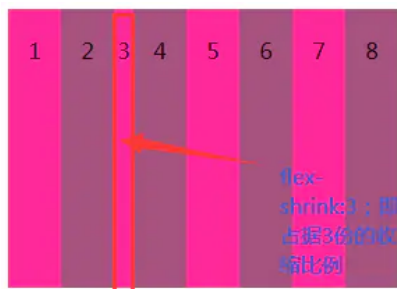
Vue干货小技巧——学会再也不做加班狗

阅读 2,234

vue优化页面

阅读 1,143

```
1 <head>
2   <style type="text/css">
3     body > div{
4       float: left;
5     }
6     .wrapper {
7       width: 280px;
8       height: 200px;
9       background-color: rgb(247, 176, 188);
10      margin: 10px;
11      overflow: hidden;
12      display: flex;
13    }
14    .wrapper div{
15      box-sizing: border-box;
16      background-color: rgb(255, 40, 154);
17      width: 50px;
18      text-align: center;
19      line-height: 60px;
20    }
21    .wrapper div:nth-of-type(2n){
22      background-color: rgb(167, 82, 127);
23    }
24    .wrapper one div:nth-of-type(3){
25      flex-shrink: 3;
26    }
27  </style>
28 </head>
29 <body>
30   <div class="wrapper one">
31     <div>1</div>
32     <div>2</div>
33     <div>3</div>
34     <div>4</div>
35     <div>5</div>
36     <div>6</div>
37     <div>7</div>
38     <div>8</div>
39   </div>
40 </body>
```



结果

## 5. flex-basis

定义子项在主轴上的基准尺寸（初始大小）

默认值：**auto**，即项目原定大小；如果设定值，则优先于项目原定大小

## 6 flex

**flex**是**flex-grow**、**flex-shrink** 和 **flex-basis**的简写。多个值以空格隔开，按既定顺序排列（**grow**、**shrink**、**basis**）

以上就是**Flex**布局的全部属性和用法！

## 推荐阅读

为什么总有人觉得前端很简单？

阅读 175

面试了几个前端，给爷整哭了！

阅读 56,709

VUE响应式原理

阅读 147

Vue干货小技巧——学会再也不做加班狗

阅读 2,234

vue优化页面

阅读 1,143



2人点赞 >



笔记 (2)



写下你的评论...

评论 0

赞 2

...



赞赏又付

还没有人赞赏，支持一下



jCodeLife



书山有路勤为径，学海无涯苦作舟

总资产18 (约1.57元) 共写了13.0W字 获得259个赞 共23个粉丝



写下你的评论...

全部评论 0

只看作者

关闭评论

按时间倒序

按时间正序

被以下专题收入，发现更多相似内容

投稿管理

+ 收入我的专题



前段开发那些事儿

## 推荐阅读

为什么总有人觉得前端很简单？

阅读 175

面试了几个前端，给爷整哭了！

阅读 56,709

VUE响应式原理

阅读 147

Vue干货小技巧——学会再也不做加班狗

阅读 2,234

vue优化页面

阅读 1,143



写下你的评论...

评论 0

赞 2

