

flex 布局

Flex : 是 Flexible Box 的缩写, 意为"弹性布局", 用来为盒状模型提供最大的灵活性。

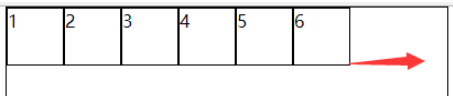
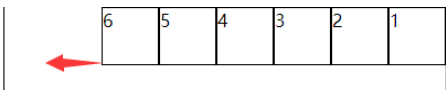
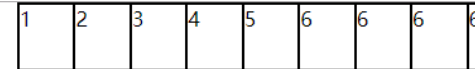
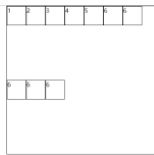
- 任何一个容器都可以指定为 Flex 布局。

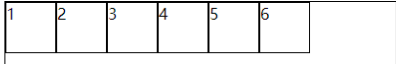
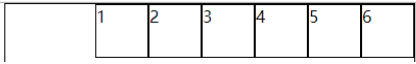
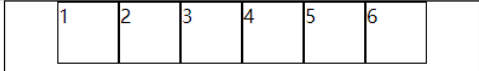
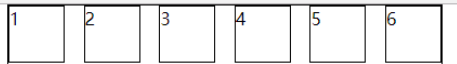
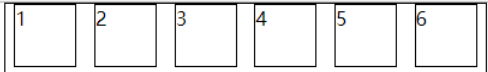
采用 Flex 布局的元素, 称为 **Flex 容器** (flex container), 简称"容器"。它的所有子元素自动成为容器成员, 称为 **Flex 项目** (flex item), 简称"项目"。就是一个大框里面有一个有很多小盒子加容器身上, 给容器加一个 display:

flex 像浮动又不是浮动

1. 有新旧版本之分
2. 容器布局方式
3. 项目的属性

新版本

| 属性 | 可取值 | 说明 | 运行图 |
|--|----------------|---------------------------------|---|
| flex-direction: 默认值: row 不可继承 | row | row (默认值): 主轴为水平方向, 起点在左端。 |  |
| | row-reverse | row-reverse: 主轴为水平方向, 起点在右端。 |  |
| | column | column: 主轴为垂直方向, 起点在上沿。 | 如箭头 |
| | column-reverse | column-reverse: 主轴为垂直方向, 起点在下沿。 | |
| flex-wrap: 默认值: nowrap 不可继承 | nowrap | 不换行, 有多少表格就按宽度的平均分 |  |
| | wrap | 自动换行, 两行的和空白的同时平分高的空间 |  |

| | | | |
|--|---------------------------|-------------------------------------|---|
| 旧版本没有这个属性，分两个容器写 | wrap-reverse | 整个里面的盒子倒过来 |  |
| flex-flow 是 flex-direction 和 flex-wrap 的合起来的值 | flex-flow:row wrap | | |
| justify-content: 默认中: flex-start 不可继承 | flex-start | 靠左居中 |  |
| | flex-end | 靠右居中 |  |
| | center | 居中 |  |
| | space-between | 两边靠边，中项目距离相同 |  |
| | space-around | 每个项目两侧的间隔相等。所以，项目之间的间隔比项目与边框的间隔大一倍。 |  |


| | | | |
|---|-------------------|--------------------------------|---|
| align-items 默认值: flex-start | flex-start | 整个项目盒子在正轴上 |  |
| | flex-end | 整个项目盒子在正轴的对面 |  |
| | center | 整个盒子在容器的正中间 |  |
| | stretch | 如果项目未设置高度或设为 auto, 将占满整个容器的高度。 |  |

| | | | |
|--|------------|-------------------------------|---|
| | baseline | 项目的第一行文字的基线对齐。，后面的文字也是按照这个来对齐 |  |
| align-content 当容器里面不止有一行的时候 就用这个属性， align-items 就失效 | flex-start | 整个盒子在贴着主线 |  |
| | flex-end | 整个盒子在主线的对面 |  |

| | | | |
|--|----------------------|-------------------------|---|
| | center | 整个盒子在容器中居中 |  |
| | stretch | 整个盒子如果没有设置高度就整个盒子拉得和高度高 |  |
| | space-between/around | 是竖着的对齐 | |

项目的属性：

| 属性 | 可取值 | 说明 | 图解 |
|------------------------------|---|---|------------------------|
| order: | 数字：第几位 | 属性定义项目的排列顺序。数值越小，排列越靠前，默认为 0 | |
| flex-grow: 拉伸因子 控制单个项目的宽度 | 0 每个项目都平均分 1 被选中的项目比其他放大 1 倍 2 被选中的项目比其他的放大 2 倍 | 属性定义项目的放大比例，默认为属性定义项目的放大比例，默认为 0，即如果存在剩余空间，也不放大。0，即如果存在剩余空间，也不放大。 | 这个数多少个 1 按照比例根据来算获得的宽度 |

| | | | |
|---------------------------------|--|--|---|
| <p>flex-shrink: 收缩因子</p> | <p>填数字</p> | <p>如果所有项目的 flex-shrink 属性都为 1，当空间不足时，都将等比例缩小。如果一个项目的 flex-shrink 属性为 0，其他项目都为 1，则空间不足时，前者不缩小。负值对该属性无效。</p> | <div data-bbox="1355 252 2033 399">  </div> <p>如果其他是 1，这个是 0，那么其他变小，为 0 的都不变</p> |
| <p>align-self</p> | <p>除了 auto，其他都与 align-items 属性完全一致。</p> | <p>align-self 属性允许单个项目有与其他项目不一样的对齐方式，可覆盖 align-items 属性。默认值为 auto，表示继承父元素的 align-items 属性，如果没有父元素，则等同于 stretch。</p> | <div data-bbox="1355 622 1758 821">  </div> |
| <p>flex-basis</p> | <p>flex-basis 属性定义了再分配多余空间之前，项目占据的主轴空间（main size）。浏览器根据这个属性，计算主轴是否有多余空间。它的默认值为 auto，即项目的本来大小</p> | | |

| | | | |
|------|--|---|--|
| flex | 该属性有两个快捷值： auto (1 1 auto) 和 none (0 0 auto)。 | flex 属性 flex-grow, flex-shrink 和 flex-basis 的简写，默认值为 0 1 auto。后两个属性可选。 不放大不缩小 | |
|------|--|---|--|

兼容写法

首先是定义容器的 **display** 属性：

```
.box{
  display: -webkit-box; /* 老版本语法: Safari, iOS, Android browser, older WebKit browsers. */
  display: -moz-box; /* 老版本语法: Firefox (buggy) */
  display: -ms-flexbox; /* 混合版本语法: IE 10 */
  display: -webkit-flex; /* 新版本语法: Chrome 21+ */
  display: flex; /* 新版本语法: Opera 12.1, Firefox 22+ */
}
```

由于旧版语法并没有列入 W3C 标准，所以不用写 `display:box`。

这里还要注意的，如果子元素是行内元素，在很多情况下都要使用 `display:block` 或 `display:inline-block` 把行内子元素变成块元素（例如使用 `box-flex` 属性），这也是旧版语法和新版语法的区别之一。

子元素主轴对齐方式: **justify-content**


```
.box{
  -webkit-box-pack: center;
  -moz-justify-content: center;
  -webkit-justify-content: center;
  justify-content: center;
```

对应于老版本语法的: **box-pack**, 需要注意的是旧版语法有 4 个参数, 而新版语法有 5 个参数, 兼容写法新版语法的 **space-around** 是不可用的:

```
.box{
  box-pack: start | end | center | justify;
  /*主轴对齐: 左对齐(默认) | 右对齐 | 居中对齐 | 左右对齐*/

  justify-content: flex-start | flex-end | center | space-between | space-around;
  /*主轴对齐方式: 左对齐(默认) | 右对齐 | 居中对齐 | 两端对齐 | 平均分布*/
}
```

子元素交叉轴对齐方式: **align-items**

```
.box{
  -webkit-box-align: center;
  -moz-align-items: center;
  -webkit-align-items: center;
  align-items: center;
}
```

对应于老版本的: **box-align**, 这里只是参数写法不同, 但是功能没变:

```
.box{
  box-align: start | end | center | baseline | stretch;
  /*交叉轴对齐：顶部对齐（默认） | 底部对齐 | 居中对齐 | 文本基线对齐 | 上下对齐并铺满*/

  align-items: flex-start | flex-end | center | baseline | stretch;
  /*交叉轴对齐方式：顶部对齐（默认） | 底部对齐 | 居中对齐 | 上下对齐并铺满 | 文本基线对齐*/
}
```

子元素的显示方向: **flex-direction**(默认为 row)

兼容的写法:子元素的显示方向可通过 **box-direction + box-orient + flex-direction** 实现，下面请看实例：

左到右

```
.box{
  -webkit-box-direction: normal;
  -webkit-box-orient: horizontal;
  -moz-flex-direction: row;
  -webkit-flex-direction: row;
  flex-direction: row; }
```

右到左: (在老版本中的写法 **box-direction** 只是改变了子元素的排序，并没有改变对齐方式，需要新增一个 **box-pack** 来改变对齐方式。)

```
.box{
  -webkit-box-pack: end;
  -webkit-box-direction: reverse;
  -webkit-box-orient: horizontal;
  -moz-flex-direction: row-reverse;
```

```
    -webkit-flex-direction: row-reverse;
    flex-direction: row-reverse;
}
```

上到下

```
.box{
    -webkit-box-direction: normal;
    -webkit-box-orient: vertical;
    -moz-flex-direction: column;
    -webkit-flex-direction: column;
    flex-direction: column; }
```

下到上

```
.box{
    -webkit-box-pack: end;
    -webkit-box-direction: reverse;
    -webkit-box-orient: vertical;
    -moz-flex-direction: column-reverse;
    -webkit-flex-direction: column-reverse;
    flex-direction: column-reverse;
}
```

是否允许子元素伸缩

```
.item{
    -webkit-box-flex: 1.0;
    -moz-flex-grow: 1;
    -webkit-flex-grow: 1;
    flex-grow: 1;
```

```
}  
  
.item{  
    -webkit-box-flex: 1.0;  
    -moz-flex-shrink: 1;  
    -webkit-flex-shrink: 1;  
    flex-shrink: 1;  
}
```

在老版本的 box 语法中 **box-flex** 如果不是 0 就表示该子元素允许伸缩，而 **flex** 是分开的，上面 **flex-grow** 是允许放大（默认不允许），下面的 **flex-shrink** 是允许缩小（默认允许）。**box-flex** 默认值为 0；

```
.item{  
    box-flex: <value>;  
    /*伸缩： <一个浮点数，默认为 0.0，即表示不可伸缩，大于 0 的值可伸缩，柔性相对>*/  
  
    flex-grow: <number>; /* default 0 */  
    /*放大： 默认 0（即如果有剩余空间也不放大，值为 1 则放大，2 是 1 的双倍大小，以此类推）*/  
  
    flex-shrink: <number>; /* default 1 */  
    /*缩小： 默认 1（如果空间不足则会缩小，值为 0 不缩小）*/  
}
```

子元素的显示次序

```
.item{  
    -webkit-box-ordinal-group: 1;
```

```
-moz-order: 1;  
-webkit-order: 1;  
order: 1;
```

```
}
```