

# 浅谈CSS3中display属性的Flex布局

最近在学习微信小程序，在设计首页布局的时候，新认识了一种布局方式display:flex

```
1 .container {  
2   display: flex;  
3   flex-direction: column;  
4   align-items: center;  
5   background-color: #b3d4db;  
6 }
```

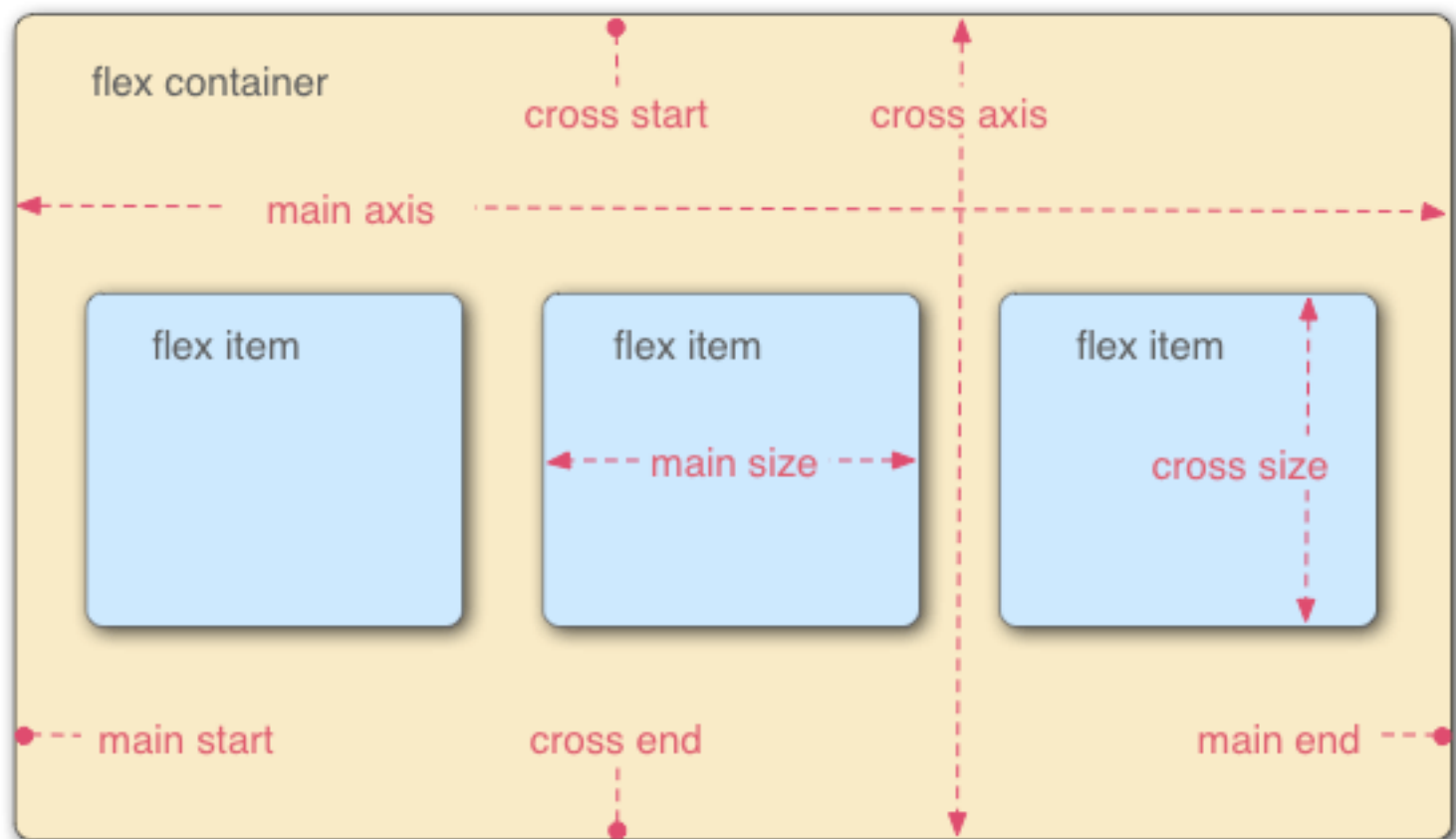
编译之后的效果很明显，界面的布局也很合理，看起来很清晰。那么究竟这个属性是干嘛用的呢？

**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。**设为Flex布局以后，子元素的float、clear和vertical-align属性将失效。

它即可以应用于容器中，也可以应用于行内元素。(以上说明结合微信开发者工具说明)2009年，W3C提出了一种新的方案----Flex布局，可以简便、完整、响应式地实现各种页面布局。目前，它已经得到了所有浏览器的支持，这意味着，现在就能很安全地使用这项功能。

## 基本概念

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"。容器默认存在两根轴：水平的主轴（main axis）和垂直的交叉轴（cross axis）。主轴的开始位置（与边框的交叉点）叫做main start，结束位置叫做main end；交叉轴的开始位置叫做cross start，结束位置叫做cross end。项目默认沿主轴排列。单个项目占据的主轴空间叫做main size，占据的交叉轴空间叫做cross size。



以下6个属性设置在容器上：

- - flex-direction      容器内项目的排列方向(默认横向排列)
  - flex-wrap      容器内项目换行方式
  - flex-flow      以上两个属性的简写方式
  - justify-content      项目在主轴上的对齐方式
  - align-items      项目在交叉轴上如何对齐
  - align-content      定义了多根轴线的对齐方式。如果项目只有一根轴线，该属性不起作用。

## flex-direction

**1 .box { 2 flex-direction: row | row-reverse | column | column-reverse; 3 }**

属性可选值的范围为row(默认)沿水平主轴由左向右排列、row-reverse沿水平主轴由右向左排列、column沿垂直主轴右上到下和column-reverse。

**flex-wrap 1 .box{ 2 flex-wrap: nowrap | wrap | wrap-reverse; 3 }**

属性可选值的范围为nowrap(默认)不换行、wrap换行（第一行在上方）和wrap-reverse（你懂的~）

## flex-flow

```
1 .box { 2 flex-flow: <flex-direction> || <flex-wrap>; 3 }
```

写法属性中，将上述两种方法的值用||连接即可

## justify-content

```
1 .box { 2 justify-content: flex-start | flex-end | center | space-between | space-around; 3 }
```

项目在主轴上的对齐方式(主轴究竟是哪个轴要看属性flex-direction的设置)

flex-start: 在主轴上由左或者上开始排列

flex-end: 在主轴上由右或者下开始排列

center: 在主轴上居中排列

space-between: 在主轴上左右两端或者上下两端开始排列

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

```
1 .box { 2 align-items: flex-start | flex-end | center | baseline | stretch; 3 }
```

这里面直接上图片说明的更清楚一些

flex-start



flex-end



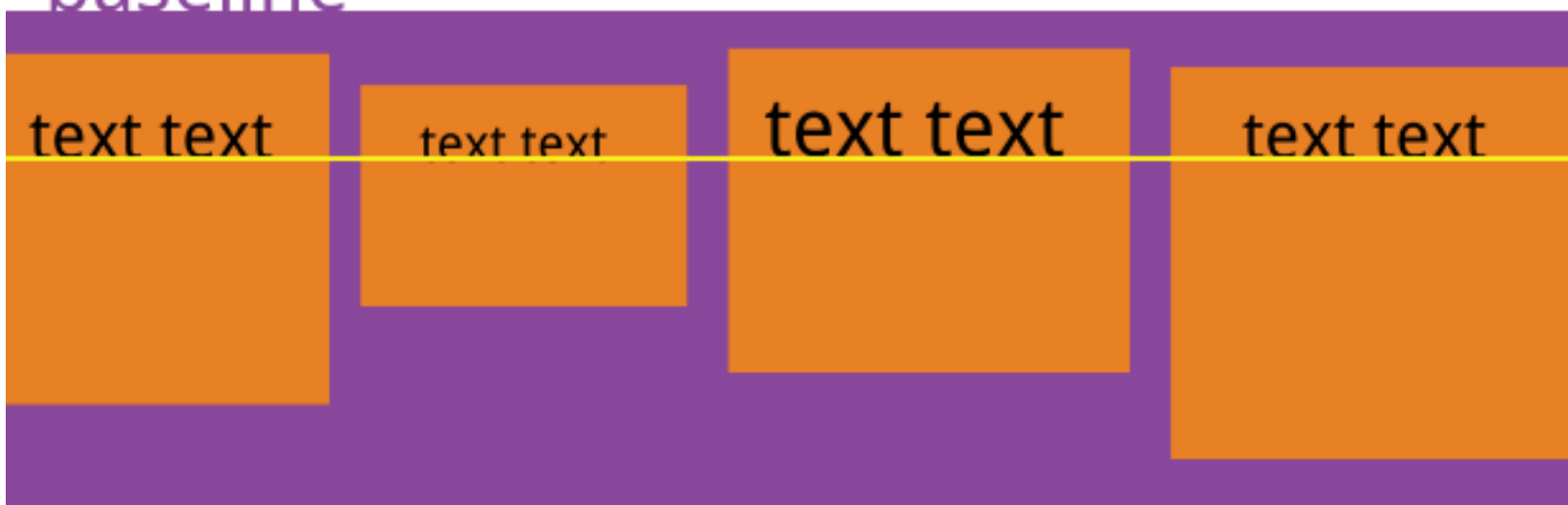
center



stretch



baseline



## align-content

1 .box { 2 align-content: flex-start | flex-end | center | space-between | space-around | stretch; 3 }

## flex-start



## flex-end



## center



## stretch



## space-between



## space-around



以上介绍完了容器中的属性，下面说一下容器中项目的属性：

- - `order` 项目的排列顺序。数值越小，排列越靠前，默认为0。
  - `flex-grow` 项目的放大比例，默认为0，即如果存在剩余空间，也不放大。

- `flex-shrink` 项目的缩小比例，默认为1，即如果空间不足，该项目将缩小。
- `flex-basis` 在分配多余空间之前，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。
- `flex` 是`flex-grow`，`flex-shrink` 和 `flex-basis`的简写，默认值为0 1 `auto`。后两个属性可选。
- `align-self` 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`。

## order

```
1 .item {
2     order: <integer>;
3 }
```

## flex-grow

```
1 .item {
2     flex-grow: <number>; /* default 0 */
3 }
```

## flex-shrink

```
1 .item {
2     flex-shrink: <number>; /* default 1 */
3 }
```

## flex-basis

```
1 .item {
2     flex-basis: <length> | auto; /* default auto */
3 }
```

## flex

```
1      .item {  
2          flex: none | [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'> ]  
3      }
```

## align-self

```
1      .item {  
2          align-self: auto | flex-start | flex-end | center | baseline | stre  
3      }
```

容器属性和项目属性是可以配合使用的，用法类似于CSS的行内式和嵌入式的道理一样。希望你可以在实际应用中熟练使用。

以上文章主要摘自[阮一峰](http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html?utm_source=tuicool)的《阮一峰的网络日志》，链接：  
[http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html?](http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html?utm_source=tuicool)  
[utm\\_source=tuicool](http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html?utm_source=tuicool)