

Flex 布局示例

感谢阮一峰老师的教程<http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html>

本示例将教程上所有的布局都简单的实现了一遍，并保存GitHub上面

<https://github.com/JailBreakC/flex-box-demo>

容器的属性

1、flex-direction属性

flex-direction属性决定主轴的方向（即项目的排列方向）。

```
.box { flex-direction: row | row-reverse | column | column-reverse; }
```

1 2 3 4

2、flex-wrap属性

默认情况下，项目都排在一条线（又称“轴线”）上。flex-wrap属性定义，如果一条轴线排不下，如何换行。

```
.box{ flex-wrap: nowrap | wrap | wrap-reverse; }
```

nowrap（默认）：不换行。

1 2 3 4 5 6 7

wrap : 换行 , 第一行在上方。

1

2

3

4

5

6

7

wrap-reverse：换行，第一行在下方。

7

4

5

6

1

2

3

3、flex-flow

flex-flow属性是flex-direction属性和flex-wrap属性的简写形式，默认值为row nowrap。

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

1 2 3 4 5 6 7

4、justify-content属性

justify-content属性定义了项目在主轴上的对齐方式。

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

flex-start (默认值) : 左对齐

1 2 3 4

flex-end : 右对齐

1 2 3 4

center : 居中



space-between : 两端对齐 , 项目之间的间隔都相等。



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。



5、align-items属性

align-items属性定义项目在交叉轴上如何对齐。

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

flex-start : 交叉轴的起点对齐。

1

3

2

4

flex-end : 交叉轴的终点对齐。

center : 交叉轴的中点对齐。

1

3

1

2

3

4

baseline: 项目的第一行文字的基线对齐。

1

2

3

4

stretch (默认值) : 如果项目未设置高度或设为auto , 将占满整个容器的高度。

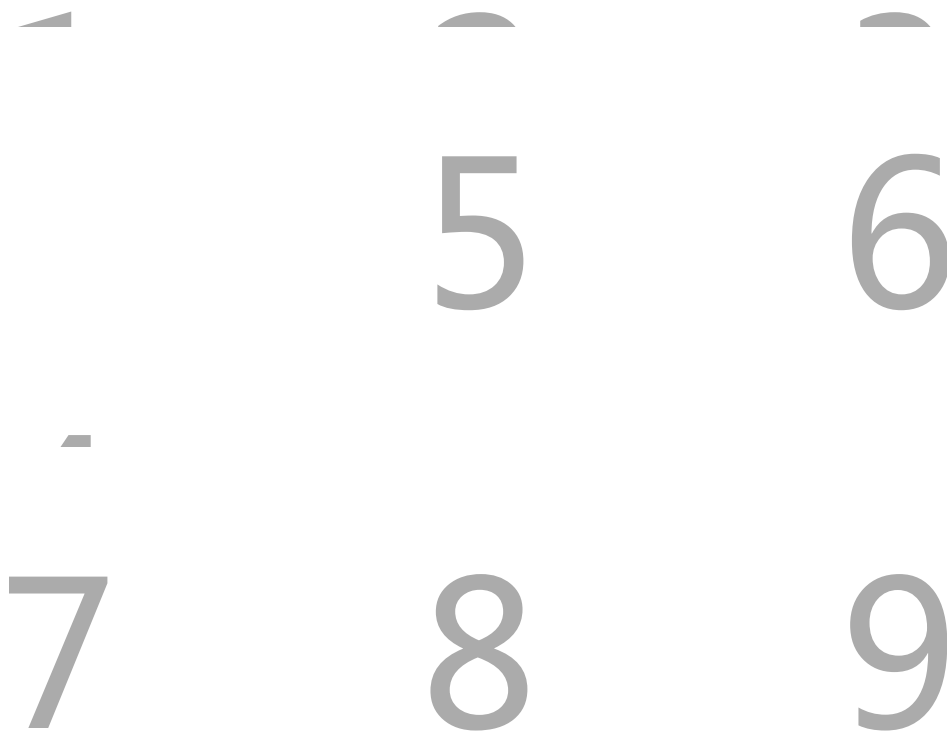


6、align-content属性

align-content属性定义了多根轴线 (多行) 的对齐方式。如果项目只有一根轴线, 该属性不起作用。

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

flex-start : 交叉轴的起点对齐。



flex-end : 与交叉轴的终点对齐。

1	2	3
	5	6
4		
7	8	9

center : 与交叉轴的中点对齐。

1	2	3
---	---	---



4

5

6

9

7

8

space-between：与交叉轴两端对齐，轴线之间的间隔平均分布。

1

2

3

4

5

6

7 8 9

space-around : 每根轴线两侧的间隔都相等。所以，轴线之间的间隔比轴线与边框的间隔大一倍。

1 2 3

4 5 6



/

8

9

stretch (默认值) : 轴线占满整个交叉轴。

1

2

3

4

5

6

7

8

9

项目的属性

1、order属性

order属性定义项目的排列顺序。数值越小，排列越靠前，默认为0。

```
.item { order: <integer>; }
```



2、flex-grow属性

flex-grow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。

如果所有项目的flex-grow属性都为1，则它们将等分剩余空间（如果有的话）。如果一个项目的flex-grow属性为2，其他项目都为1，则前者占据的剩余空间将比其他项多一倍。

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

3、flex-shrink

1

flex-grow: 1

2

flex-grow: 2

3

flex-grow: 1

flex-shrink属性定义了项目的缩小比例，默认为1，即如果空间不足，该项目将缩小。

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。

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

1

flex-shrink: 0

2

3

4、flex-basis属性

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。

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

5、flex属性

1

2

3

flex属性是flex-grow, flex-shrink 和 flex-basis的简写，默认值为0 1 auto。后两个属性可选。

该属性有两个快捷值：auto (1 1 auto) 和 none (0 0 auto)。

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

6、align-self属性

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。

```
.item { align-self: auto | flex-start | flex-end | center | baseline | stretch; }
```

1

2

4

3

flex-end

