

## 弹性盒模型

笔记本: 7.17弹性盒

创建时间: 2020/7/17 14:35

更新时间: 2020/7/20 13:55

作者: 805427154@qq.com

---

## 弹性盒模型

### 青蛙练习:

<http://blog.xiaoboswift.com/flexbox/>

#### 一、设置在容器上（父元素上）

1、display: flex;父元素设置成弹性盒。

说明：规定子项目在父元素主轴方向（默认水平）一行显示，不会换行（当子元素）。侧轴方向没有数值，（默认垂直）会默认拉伸成父元素的高度

说明：

- 主轴（默认X轴）和侧轴（默认Y轴）方向不是固定的，可以相互调换。
- 主轴方向和侧轴方向永远垂直
- 子元素在父元素里面默认是主轴方向摆放
- 子元素在没给高度情况下，侧轴方向默认拉伸成父元素高度。

注意：

当子元素承载完内容宽度还有剩余时，子元素宽度会被压缩后，并列排在父元素的主轴行中。而当子元素的宽度被挤压到最小值时（仅能承受内容），子元素的宽度不会再被压缩，从而导致溢出父元素的边框。（如下图）



```
<div id="box2">
  <div>1111111</div>
  <div>2111111</div>
  <div>3111111</div>
  <div>1111111</div>
  <div>2111111</div>
  <div>3111111</div>
  <div>1111111</div>
  <div>2111111</div>
</div>
```

2、设置主轴方向：flex-direction（写给父元素）

属性值：

- row（默认）水平向右
- column垂直向下
- row-reverse水平向左
- column-reverse垂直向上

3、flex-wrap：伸缩换行（写给父元素）

属性值：

- nowrap默认不换行；
- wrap宽度不足换行显示（向下）；
- wrap-reverse宽度不足换行显示（向上）

4、flex-flow: flex-direction/flex-wrap伸缩流主轴方向及换行

说明：其实就是flex-direction和flex-wrap的简写形式。

5、设置子项目在主轴方向的对齐方式：justify-content（写给父元素）

属性值：

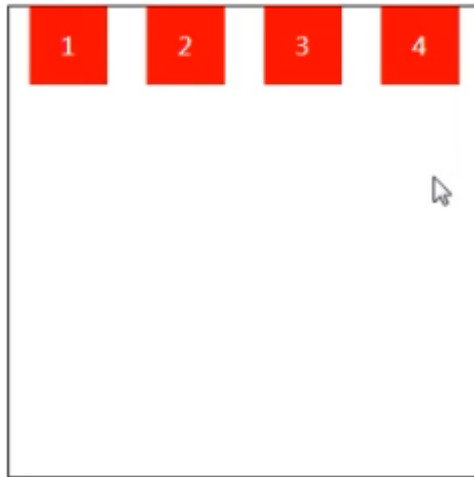
flex-start（默认值，在主轴方向起始位置对齐）

flex-end（在主轴方向结束位置对齐）

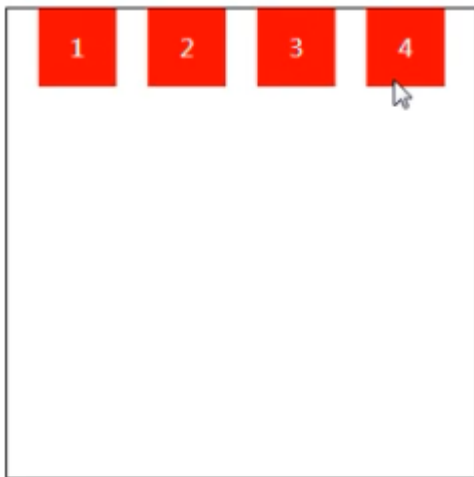
center（中间对齐）

space-between（表现为两端对齐。意思就是多余的空白间距只在元素中间区域分配）

space-around（意思是每个flex子项两侧都环绕着互不干扰的等宽空白间距，但是元素和边框两端的空隙会是元素之间空隙的50%）



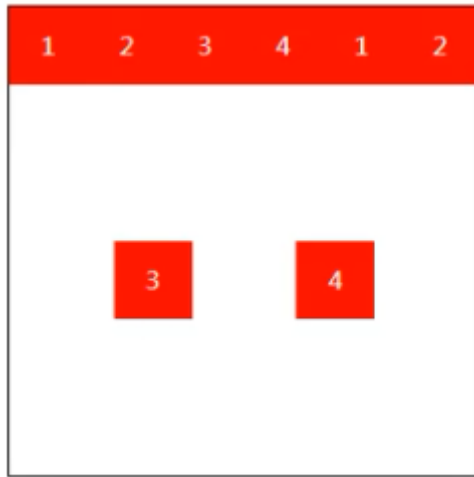
space-evenly（意思是每个flex子项两侧的空白间距完全相等。）



说明：

- 水平和垂直方向的对齐方式都可以设置，主要看主轴方向是X轴还是Y轴。

- space-around和space-evenly这两个也适合多行的情况，只是在元素太多的情况下，第一行会被占满间距而看不出来，第二行就可以看出来了。（如下图）

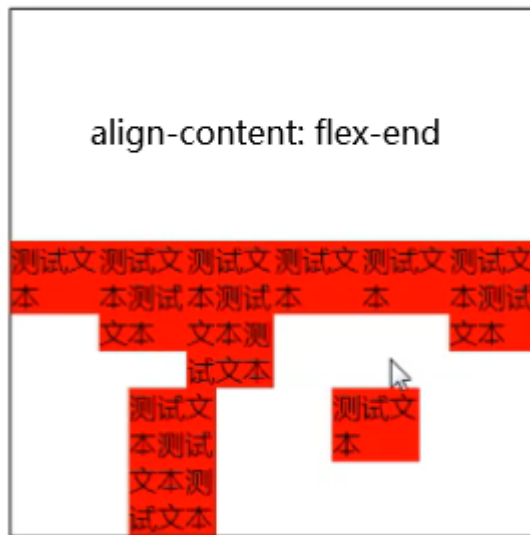
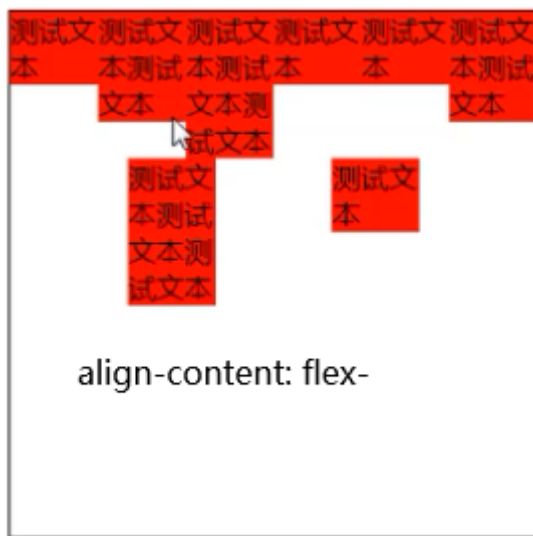


6、设置子项目的侧轴方向的对齐方式：align-content（**控制多行**）。

属性值参考justify-content。

说明：

如果所有的flex子项只有一行，align-content属性是没有任何效果的。



7、align-items指的就是flex子项们相对于flex容器在侧轴方向上的对齐方式（**写给父元素-单行**）

属性值：

stretch默认值，flex子项拉伸

flex-start表现为容器顶部对齐

flex-end表现为容器底部对齐

center垂直居中对齐

## 8、设置在子项目上

- order可以改变某一个flex子项的排序位置。所有flex子项的默认order属性值是0

1) 写了order的排在没有order属性的元素**后面**。

2) 写了order的，属性值越小的在越前面。

排列位置-按主轴方向：

order无 >> order (小) >> order (大)

- flex-grow: 扩展flex子项所占据的宽度，其实就是除去元素外的剩余空白间隙。默认值为0。



注意：0是最小值，设置属性值的时候，不能比0小。

### flex-grow和flex-shrink给的都是比例值

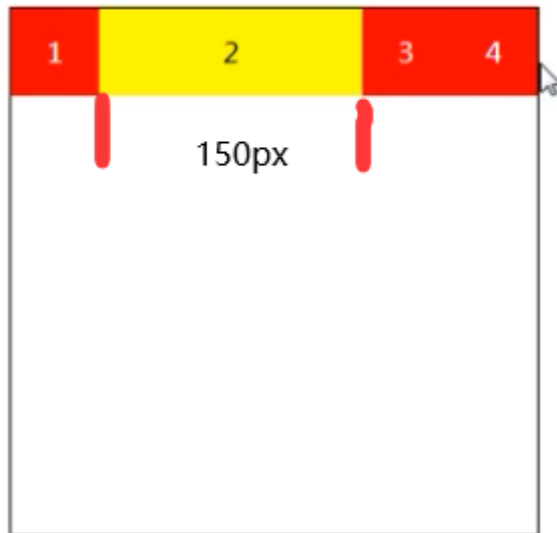
- flex-shrink: 0; 处理当flex容器空间不足的时候，单个元素的收缩比例，默认值是1.（让子元素在父元素中不被压缩）
  - 1) 默认值是1的时候，为正常收缩；值越大，收缩越厉害；
  - 2) 当shrink值为0时，子元素不被收缩；
- flex-basis: 定义在分配剩余空间之前元素的默认大小；（其实是相当于width的存在，给的是固定值，但是当flex-base和width一起写的时候，flex-basis的优先级是高于width的级别的，如图）

```

#box2 div:nth-child(2){ background: yellow; color: black; width: 100px; flex-basis: 150px; /*
flex-shrink: 2; */}
</style>
</head>
<body>
  <!-- <div id="box">
    <div>1</div>
    <div>2</div>
    <div>3</div>
    <div>4</div>
    <div>5</div>
  </div> -->

  <div id="box2">
    <div>1</div>
    <div>2</div>
    <div>3</div>
  </div>

```



- flex是flex-grow&flex-shrink&flex-basis的简写（复合写法）

语法: flex: flex-grow flex-shrink flex-basis

说明:

1) 在简写形式中, flex-shrink和flex-basis是可以省略的, **可以只写一个数值, 只写一个数值的情况下, 该数值代表flex-grow** (flex: number;其实就是侵占元素外的剩余空间来扩展自己的宽度; )

2) flex简写形式的优先级高于flex-grow, flex-grow小于width

flex (简写) >> width >> flex-grow

- **针对单个子项目**在其所在行的侧轴对齐方式align-self, 属性值参考align-items的属性值。（可以覆盖align-items）

**总结:**

**1、写给父元素的属性共有7种。display: flex、flex-direction、flex-wrap、(前两者简写)flex-flow、justify-content、align-items(单行)、align-content(多行);**

**2、写给子元素的属性共有6种。order、flex-grow、flex-shrink、flex-basis、(前三者按顺序的简写) flex、align-self**