Flex布局主要思想是让容器有能力让其子元素能够改变其宽度、高度甚至顺序，以最佳方式利用可用空间。也就是，当容器大小发生改变时，子元素能够自适应地改变其自身的大小，充分利用可用空间。

优点：

1. 在屏幕和浏览器的窗口大小发生改变时也可以灵活调整布局。
2. 可以很方便的实现垂直方面的布局，就是改变主轴的方向。
3. 可以不同DOM所指定的排序方式，可以重新对屏幕上的元素排序。

伸缩容器：

通过display：flex或者inline-flex的元素，就会成为一个伸缩容器。

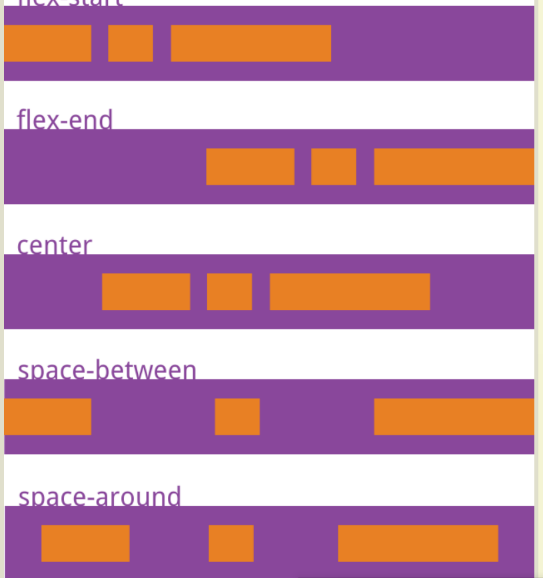
伸缩容器会创建一个新的伸缩上下文。

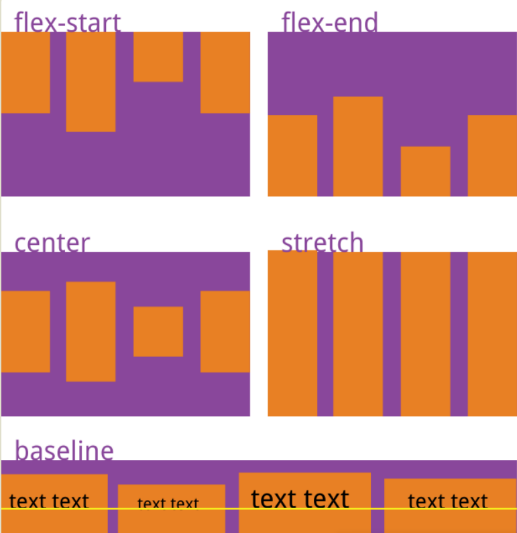
伸缩容器的每个子元素都会称为一个伸缩项目，且用户代理会将任何直接在伸缩容器里的连续文字块包裹起来成为匿名伸缩项目。

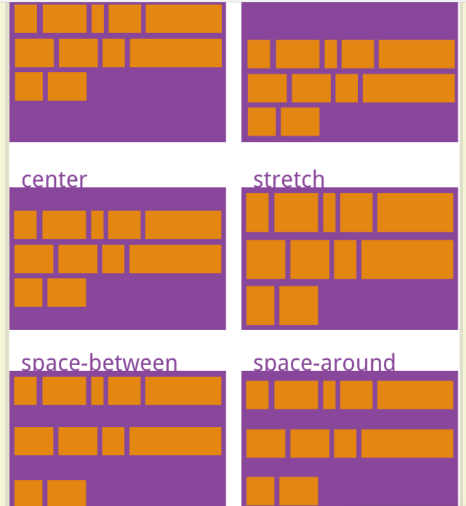
老版本声明的时候使用box/inline-box。

伸缩容器的属性？  
1.display：flex/inline-flex  
2.flex-direction：row/column/row-reverse/column-reverse



3.flex-wrap: nowrap/wrap/wrap-reverse  
  
4.flex-flow: <flex-direction>||<flex-wrap>  
5.justify-content: flex-start | flex-end | center | space-around | space-between  
这是当所有的伸缩项目都不可伸缩时或者可伸缩但已经到达最大长度时，如何对伸缩容器内的额外空间分配，本来这些额外空间是要分配到伸缩项目。  


6.aligin-items：flex-start | flex-end | center | baseline | stretch  
指定伸缩项目在侧轴的对齐方式。默认值是stretch，它会拉伸每个伸缩项目以至于填充整个容器，会使每个伸缩项目等高，自适应两栏布局时不要这样，要设置aligin-items：flex-start。  


7.align-content：flex-start | flex-end | center | space-around | space-between  
该属性定义了多根轴线对齐时的对齐方式，如果项目只有一根轴线，该属性不起作用。  
默认值是stretch。  


伸缩项目？  
1.flex-grow:<int>(默认值为0) ，定义扩展比例

2.flex-shrink:<int>(默认值是1)，定义收缩比例

如果都是1，那么每个伸缩项目都会设置为一个大小相等的额外空间。2的话，这个伸缩项目所占的额外空间是其他的2倍。

3.flex-basis：<length>||auto（默认值为auto）  
它指的是在分配多余的额外空间之前，项目占据的主轴空间。根据这个属性，浏览器会计算主轴是否有多余的空间。默认值是auto，即项目原来的大小。  
4.flex上述3个综合  
 flex:none;相当于flex:0 0 auto;  
5.align-self：flex-start || flex-end || center || stretch || baseline  
 匿名伸缩项目的align-self永远和align-items相同。  
6.order：<int>  
 越小越靠前。