



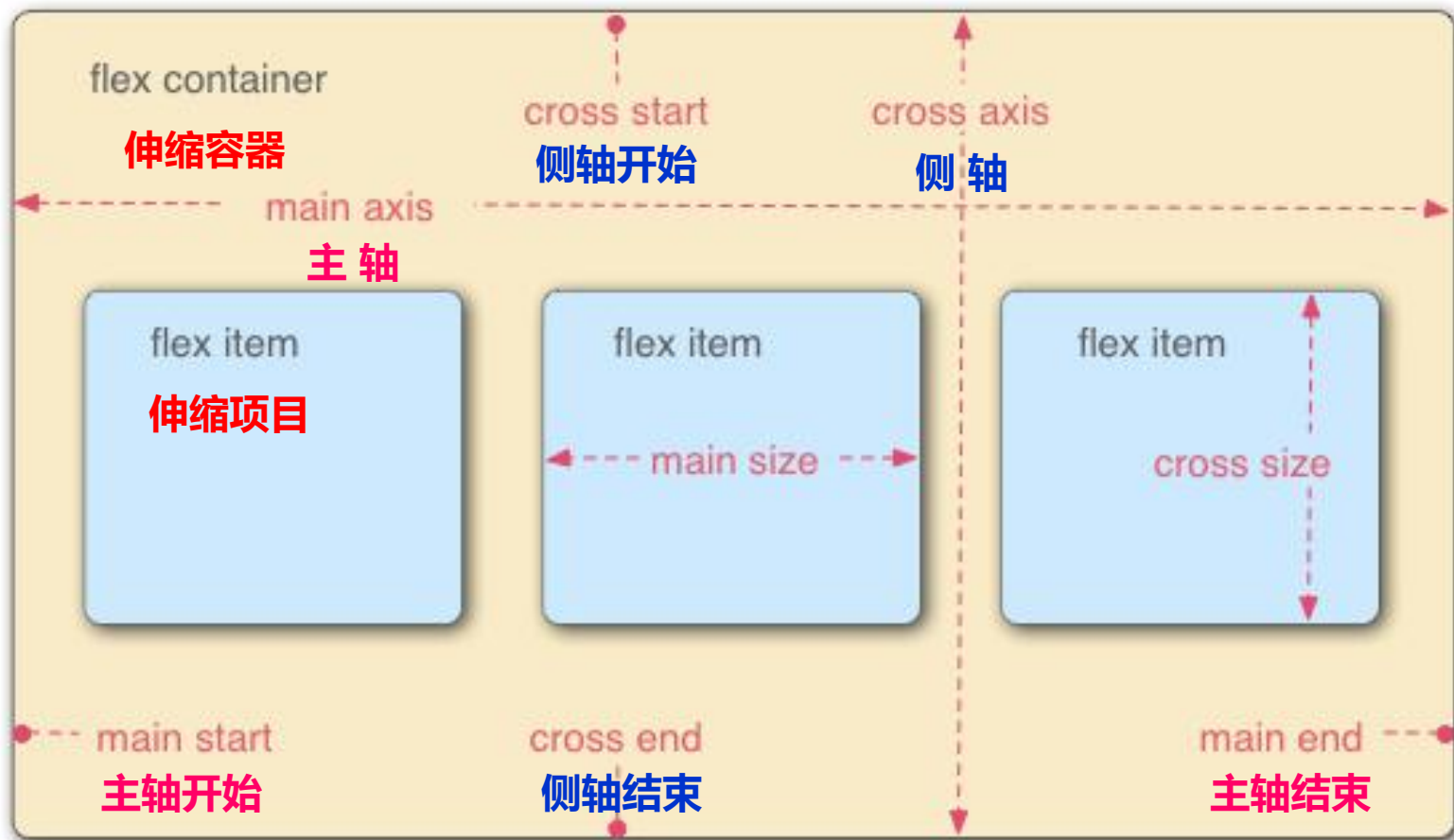
# CSS3 flex

讲师：许井龙

微信：ngsteel

# 伸缩盒模型 (flex) 简介

- 伸缩盒模型 flex 是CSS3中快速布局的利器。



- 为元素设置
  - display: flex
  - 或者 display: inline-flex
- 该元素即成为**伸缩容器 (flex container)** ,
  - 设置为 flex , 该元素会独占一行
  - 设置为 inline-flex, 可以与其他 inline 元素在同一行。
- 此时伸缩容器的子元素自动升级为**伸缩项目 (flex item)** , 伸缩项目的特点如下,
  1. **伸缩项目默认在一行排列。**
  2. **自动升级为块元素。**
  3. **所有伸缩项目默认在主轴的 start 处排列。**
  4. **伸缩项目也可以再次设置为flex, 即flex可以互相嵌套。**

- inline-flex 行内对齐特点
  - ❑ 伸缩容器中有文本内容，基于第一个文本的基线对齐。
  - ❑ 伸缩容器中没有有文本内容，也没有子元素，伸缩容器底边位于一行的基线处。
  - ❑ 伸缩容器中没有有文本内容，第一个子元素没有，第一个元素底边位于一行的基线处。



# 伸缩容器有足够空间 伸缩项目排列

# 设置伸缩项目在主轴的排列

- 当伸缩容器有“足够空间”时，所有伸缩项目在主轴start处排列，如同设置了
  - justify-content: flex-start (默认值)



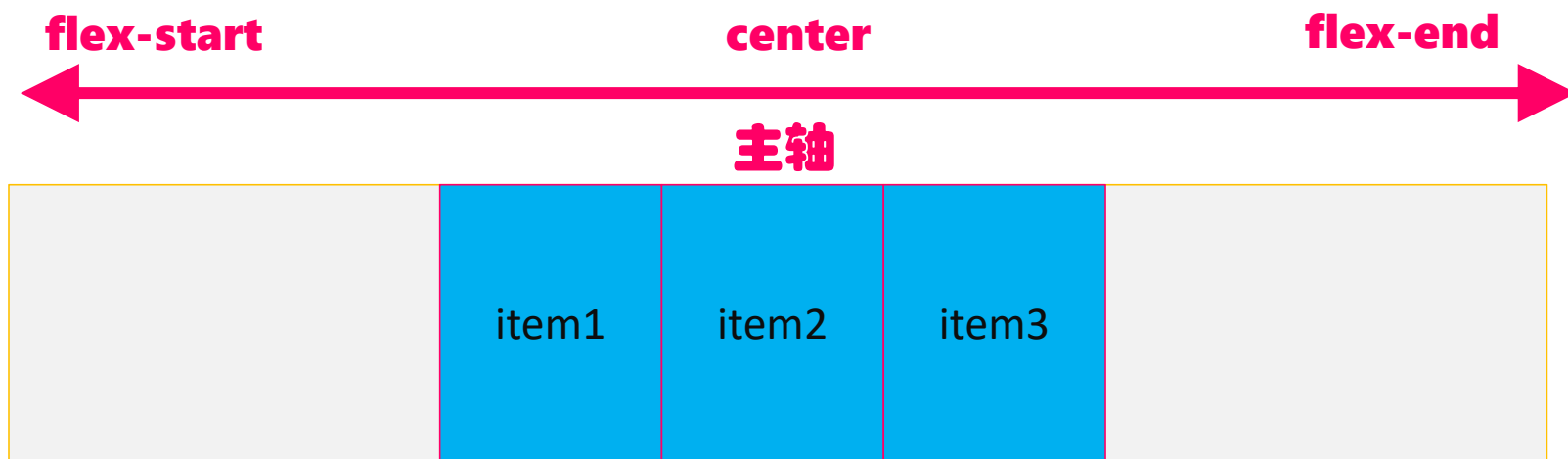
# 设置伸缩项目在主轴的排列

- 当伸缩容器有“足够空间”时，也可以设置，
  - justify-content: flex-end
- 此时所有的伸缩项目在伸缩容器主轴的end处排列



# 设置伸缩项目在主轴的排列

- 当伸缩容器有“足够空间”时，设置
  - justify-content: center
- 所有的伸缩项目位于伸缩容器主轴的中间处





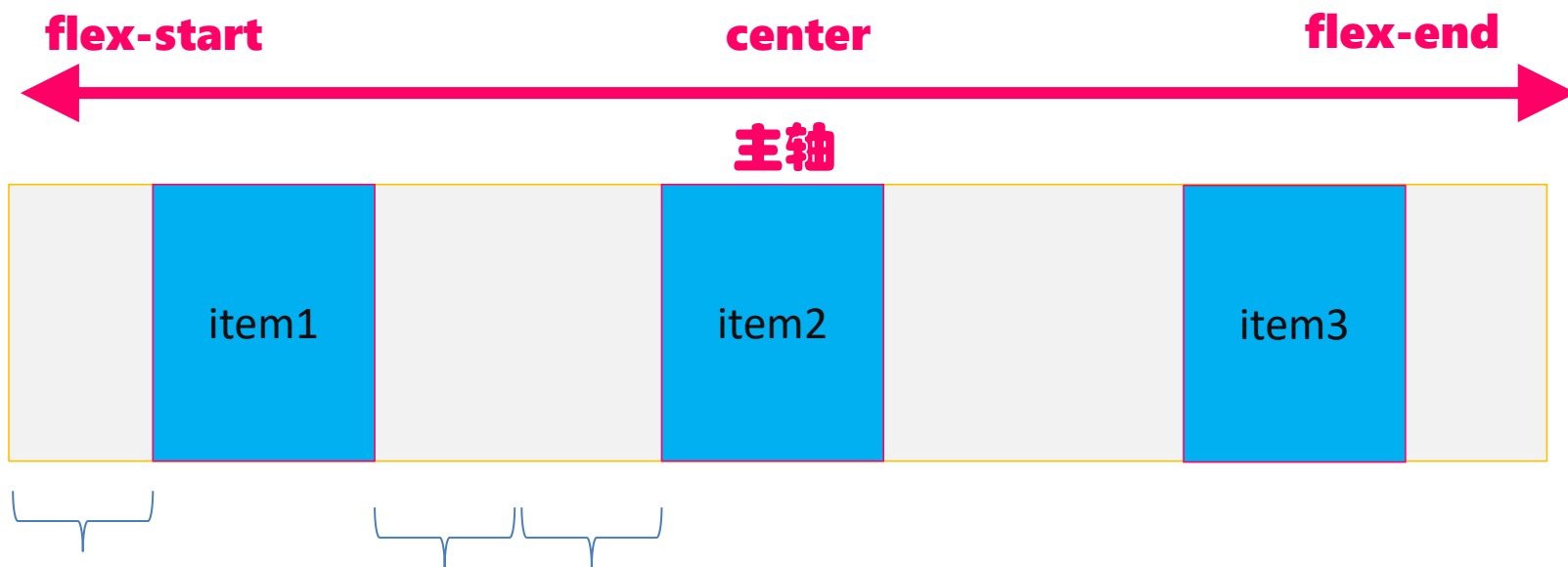
# 设置伸缩项目在主轴的排列

- 当伸缩容器有“足够空间”时，设置
  - justify-content: space-between
- 此时伸缩项目沿着伸缩容器的主轴均匀分布



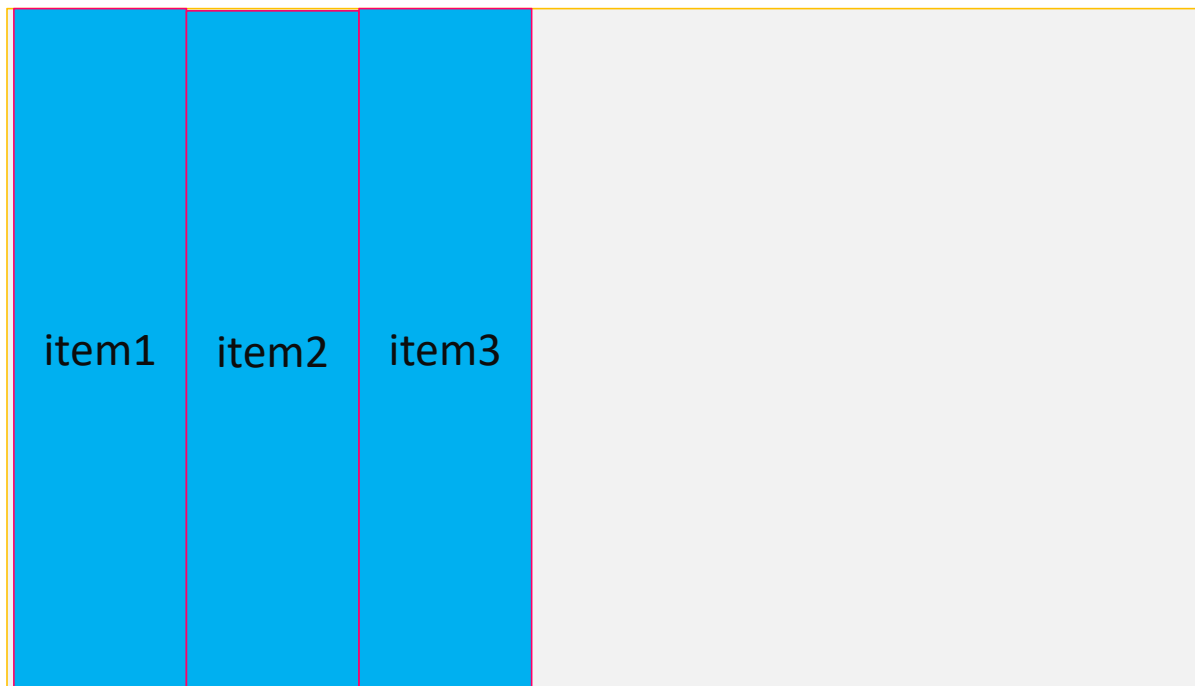
# 设置伸缩项目在主轴的排列

- 当伸缩容器有“足够空间”时，设置
  - justify-content: space-around
- 伸缩项目沿着伸缩容器的主轴均匀分布，但是剩余空间会包裹着每个伸缩项目



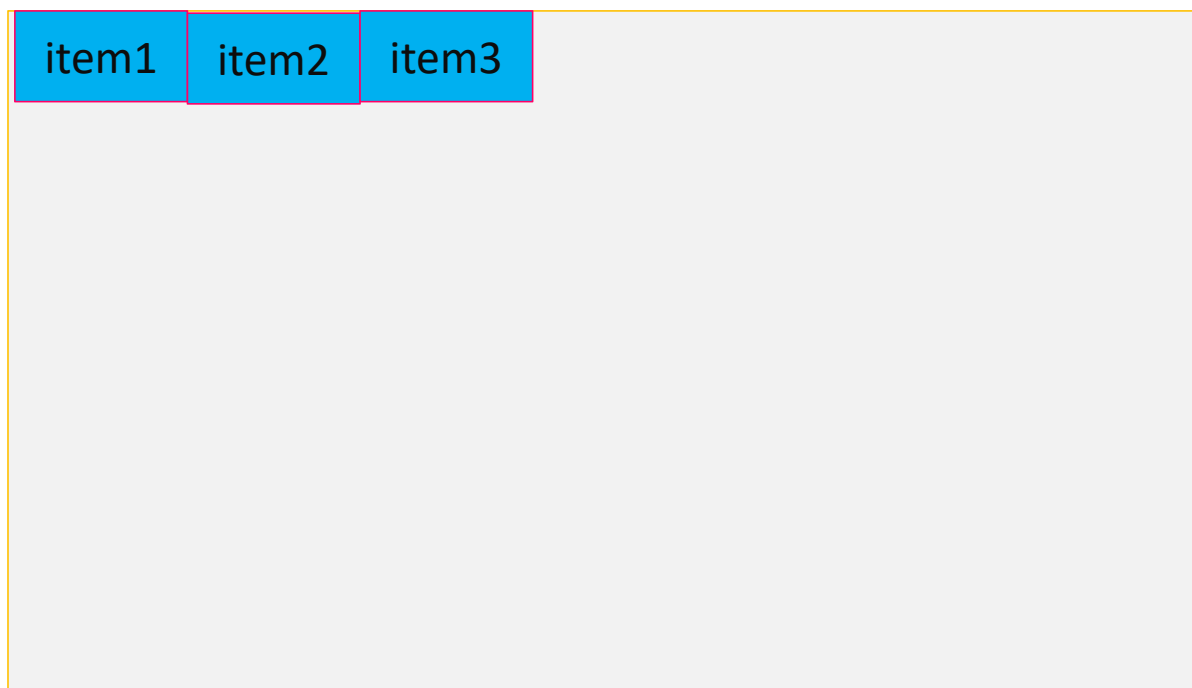
# 设置伸缩项目在侧轴排列

- 当伸缩容器有“足够空间”时，侧轴如同设置了
  - align-items: stretch
- 即每个伸缩项目会沿侧轴被拉伸。



# 设置伸缩项目在侧轴排列

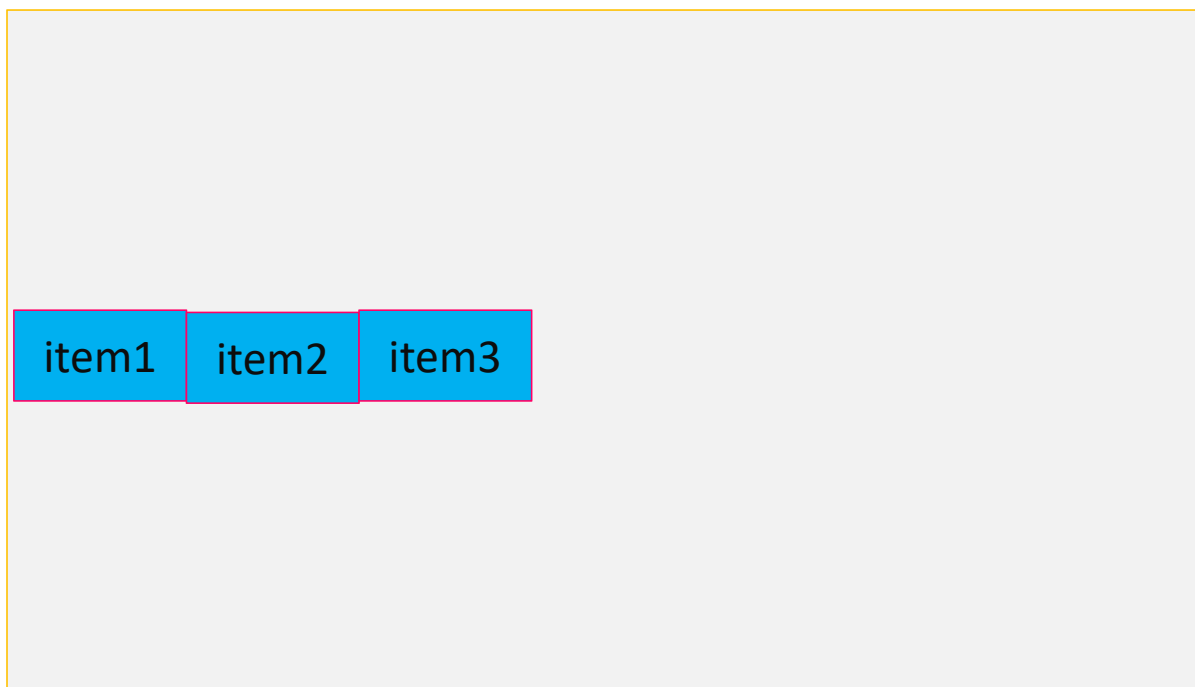
- 当伸缩容器有“足够空间”时，如果设置，
  - align-items: flex-start
- 此时所有伸缩项目位于伸缩容器侧轴start处





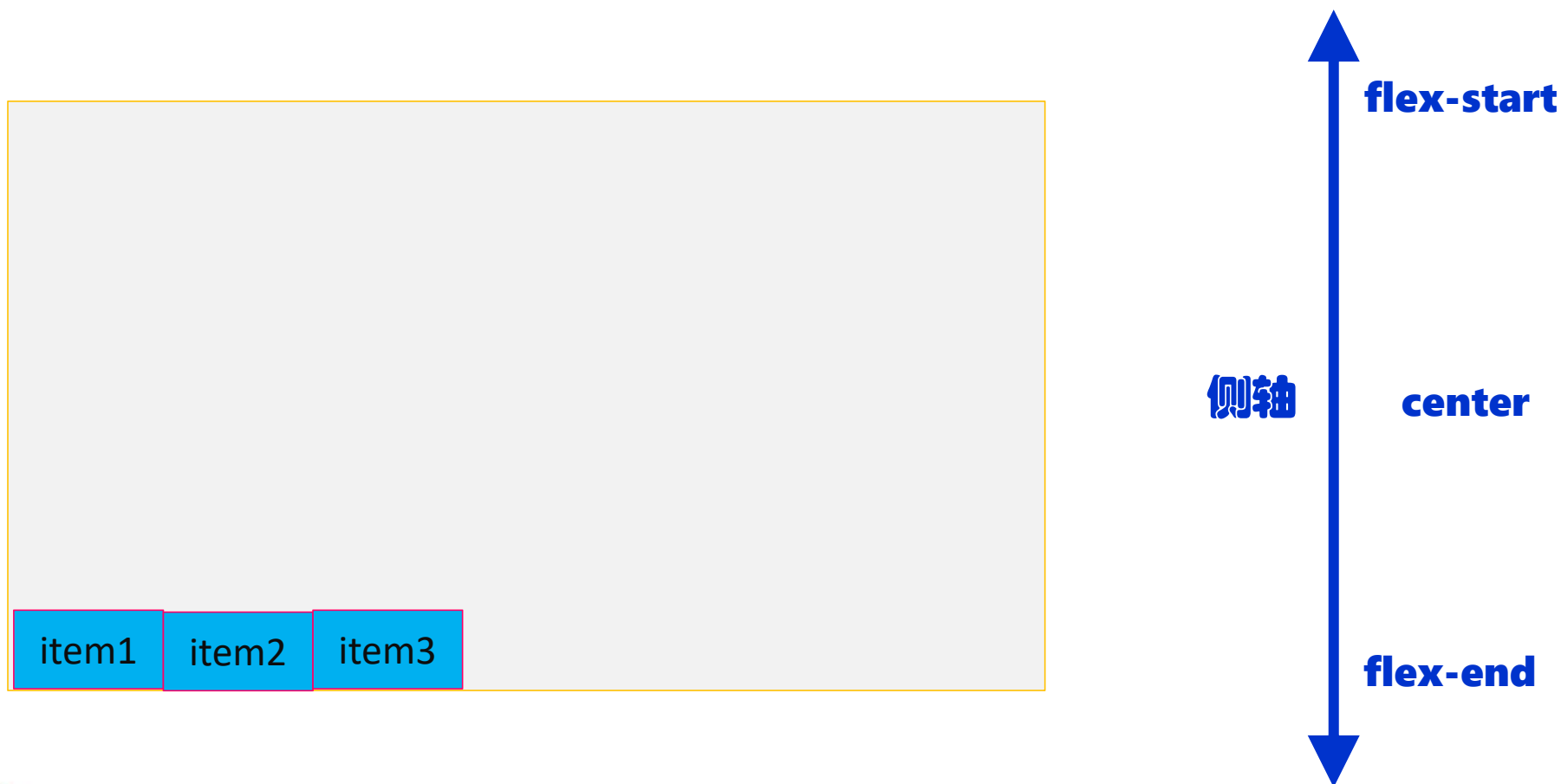
# 设置伸缩项目在侧轴排列

- 当伸缩容器有“足够空间”时，如果设置，
  - align-items: center
- 此时所有伸缩项目位于伸缩容器侧轴中部



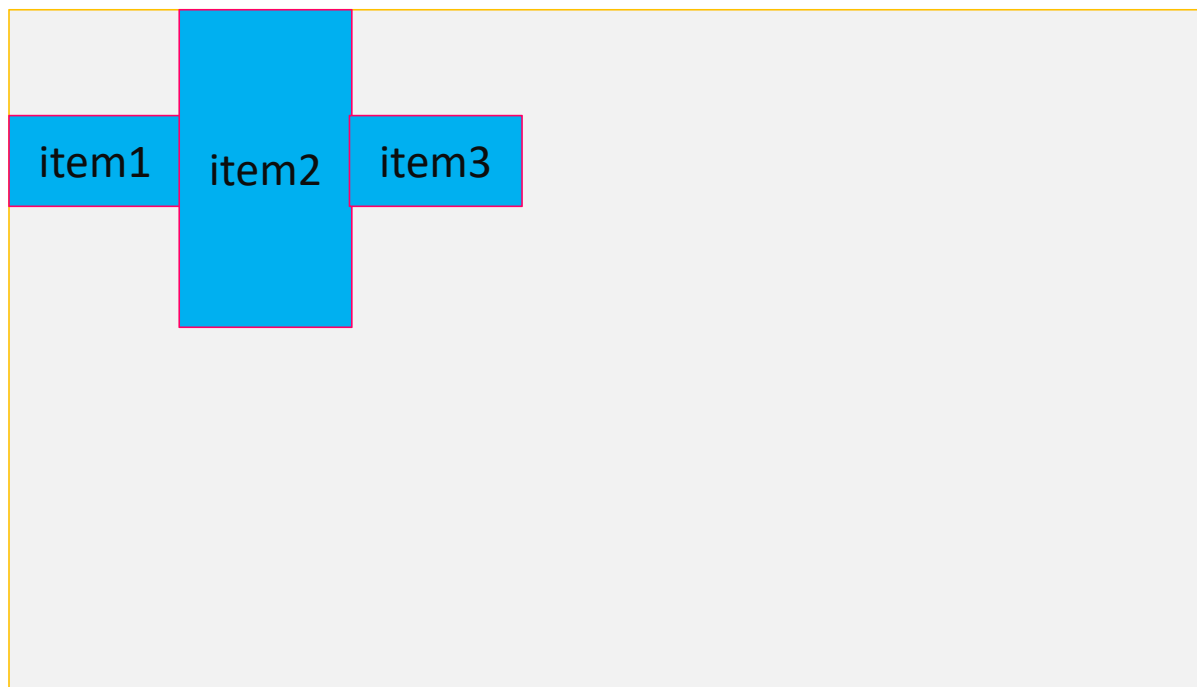
# 设置伸缩项目在侧轴排列

- 当伸缩容器有“足够空间”时，如果设置，
  - align-items: flex-end
- 此时所有伸缩项目位于伸缩容器侧轴end处



# 设置伸缩项目在侧轴排列

- 当伸缩容器有“足够空间”时，如果设置
  - align-items: baseline
- **主轴中伸缩项目基线最大的那个伸缩项目的基线作为所有伸缩项目的对齐基线**



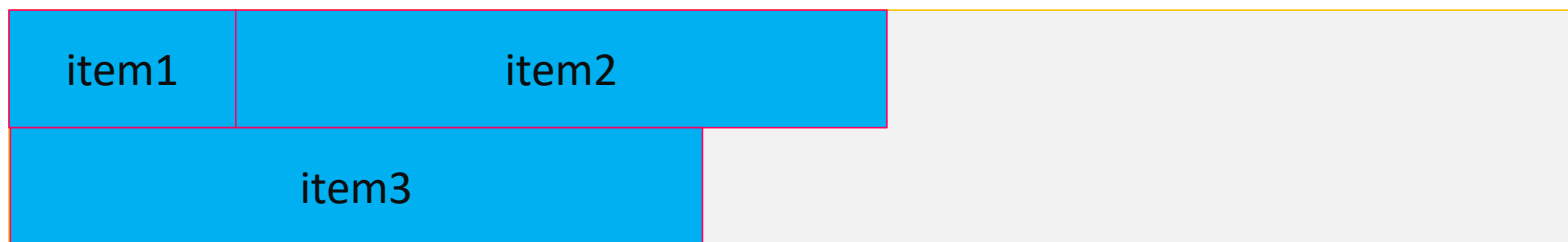
# 伸缩容器空间不足



- 
- flex-direction: row-reverse
- flex-wrap: nowrap
- item1 item2 item3

## 如何压缩，请参考伸缩项目压缩率计算规则

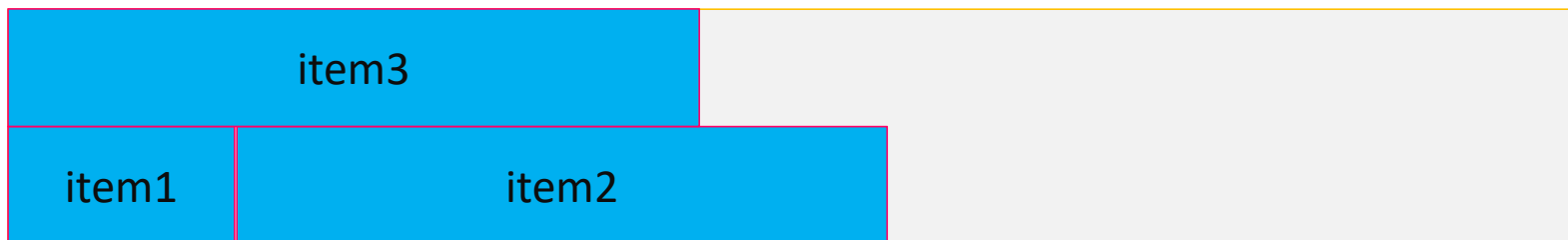
- 
- A horizontal blue double-headed arrow represents the flex axis (主轴). Above the arrow, the labels 'flex-start', 'center', and 'flex-end' are positioned from left to right. Below the arrow, the label 'flex-wrap: wrap' is centered.



- 当**伸缩项目**设置大小，且**伸缩容器**无法在主轴方向上容纳所有的伸缩项目时，在伸缩容器中设置，
  - flex-wrap: wrap-reverse
- 让**伸缩项目**自动换行，保持原来的大小。但此时伸缩项目在主轴**flex-start**和侧轴**flex-end**处开始排列。



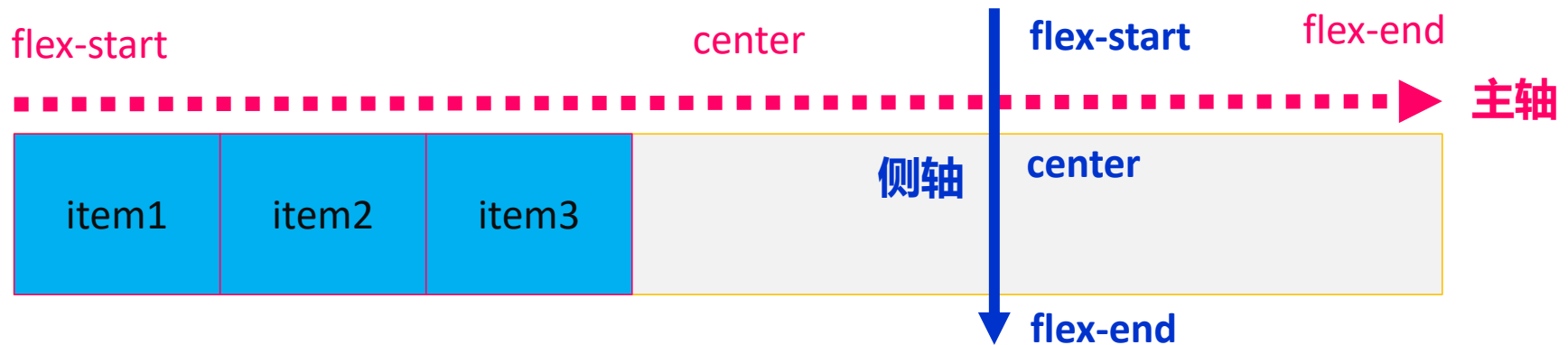
**flex-wrap: wrap-reverse;**



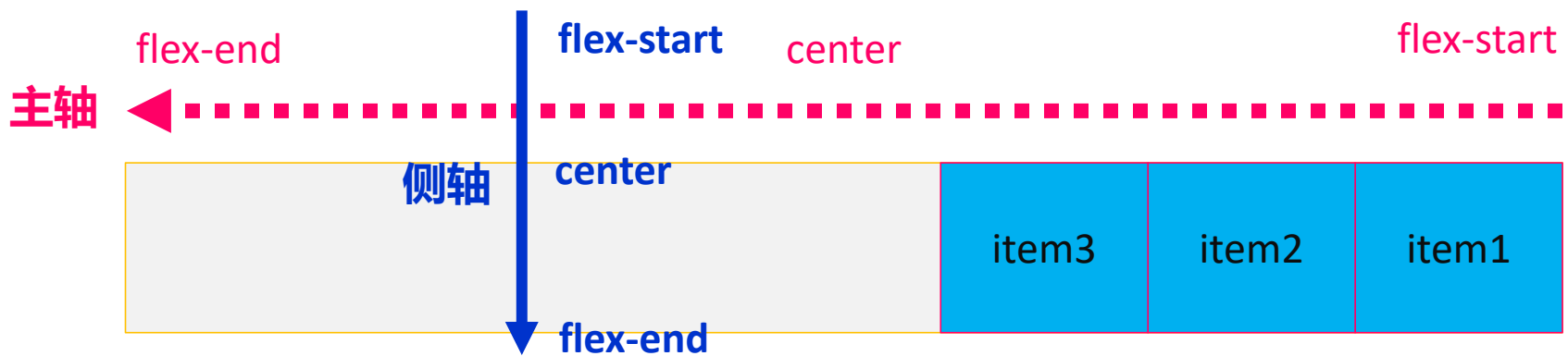
# 主轴及侧轴相关设置



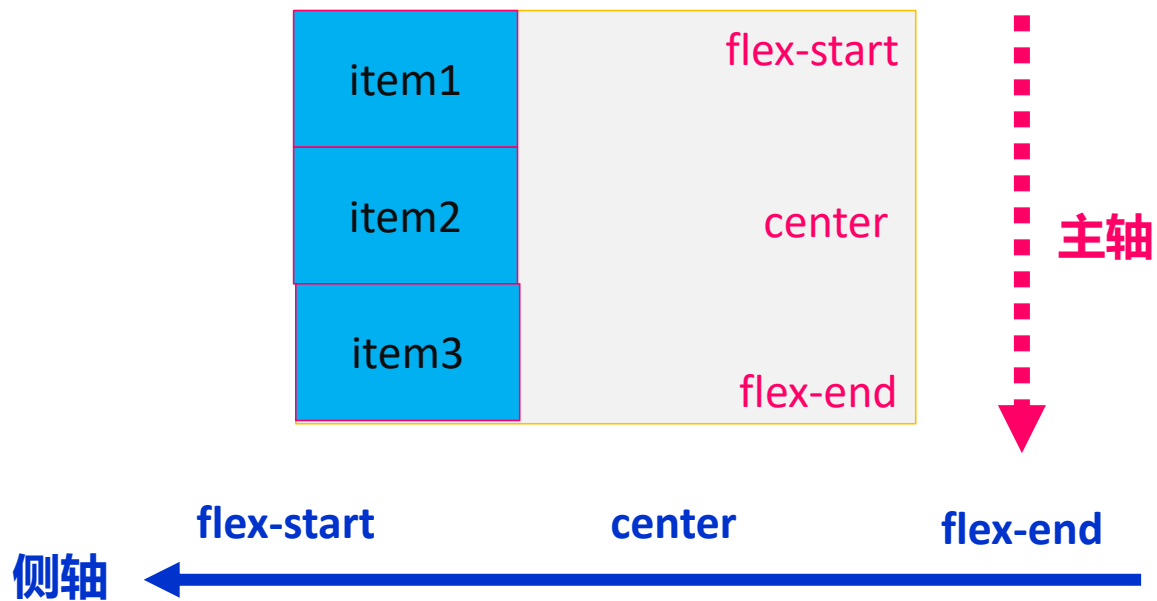
- 主轴默认方向,
  - flex-direction: row



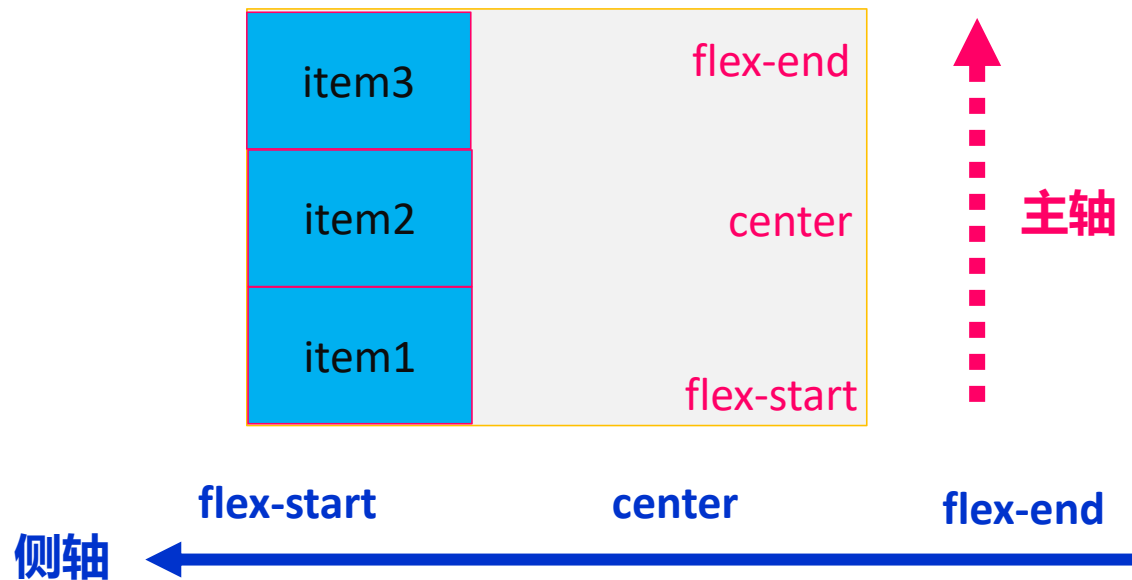
- 也可以设置
  - flex-direction: row-reverse



- 也可以通过设置
  - flex-direction: column



- 也可以设置
  - flex-direction: column -reverse

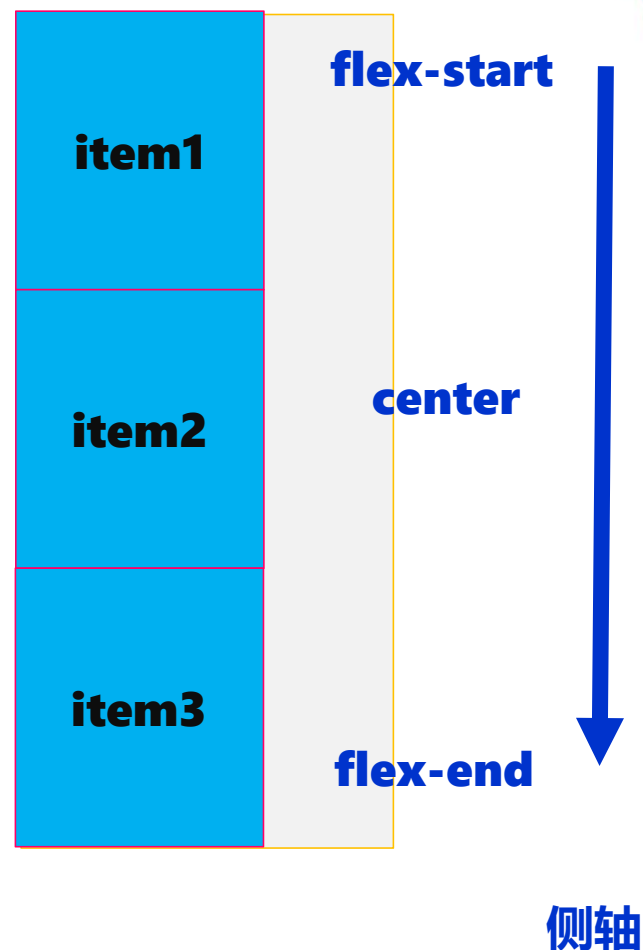




- 可以设置 flex-flow 来实现 flex-direction 和 flex-wrap 简写。
  - flex-flow : column wrap

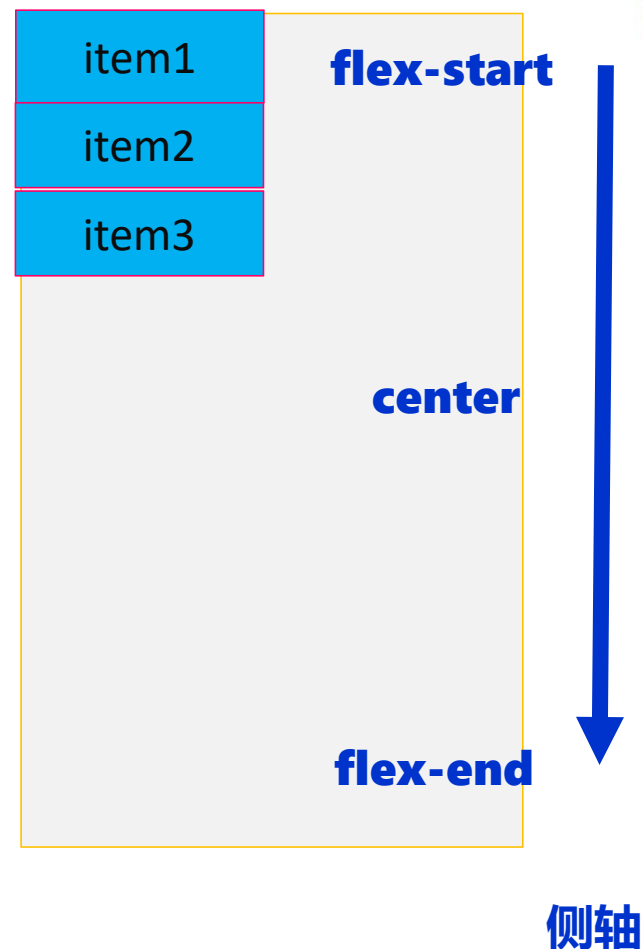
## 侧轴对齐 (伸缩项目换行)

- 当伸缩容器在**主轴上没有足够的空间**容纳所有的伸缩项目, 伸缩容器设置了flex-wrap: wrap, 并且伸缩项目没有设置大小, 完全由内容“撑起”, 如同在伸缩项目中设置了
  - align-content: stretch
- 所有伸缩项目默认均匀分配侧轴空间



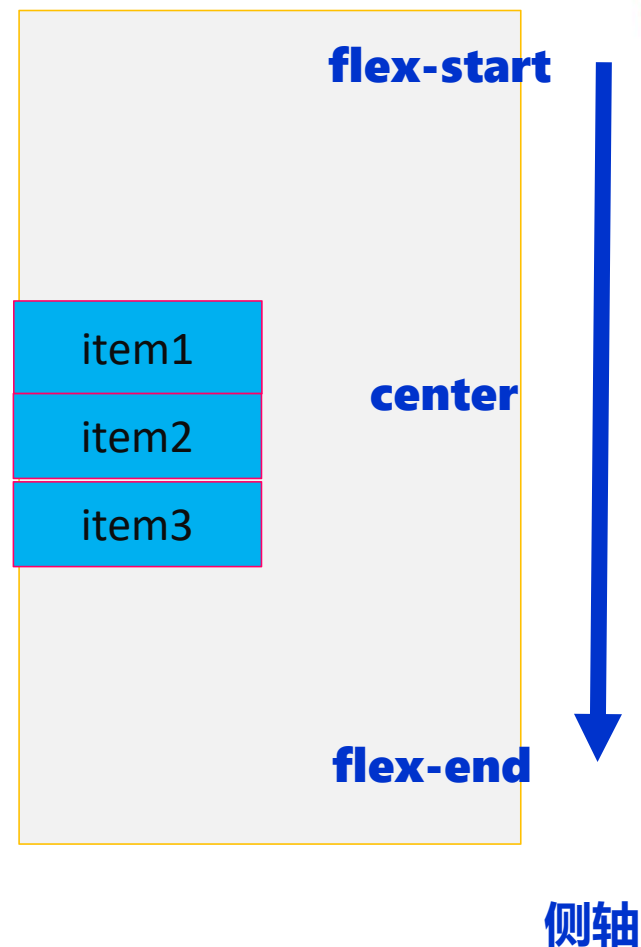
# 侧轴对齐 (伸缩项目换行)

- 当伸缩容器在**主轴上没有足够的空间**容纳所有的伸缩项目, 伸缩容器设置了flex-wrap: wrap, 并且伸缩项目没有设置大小, 完全由内容“撑起”, 如果设置
  - align-content: flex-start



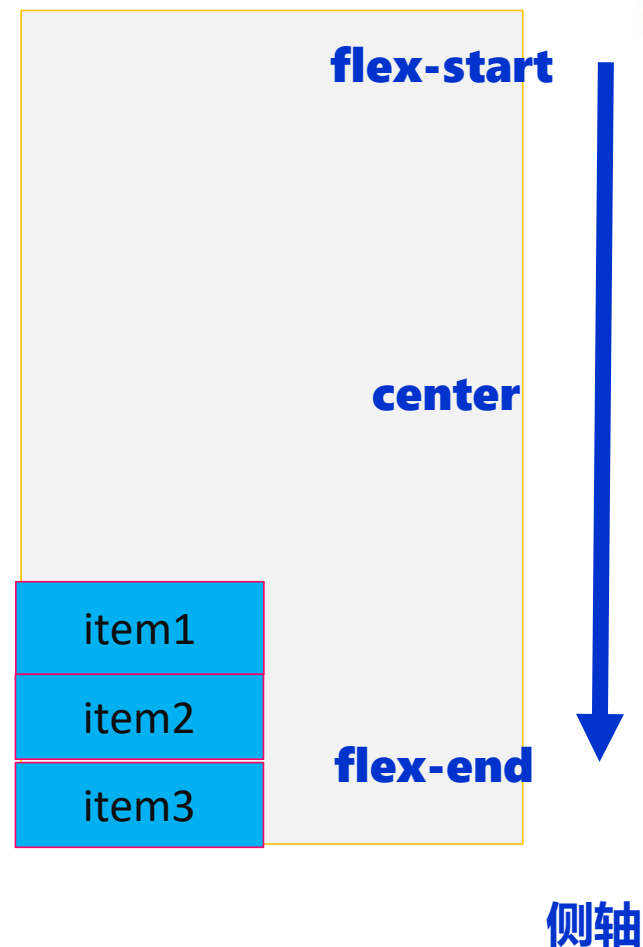
## 侧轴对齐 (伸缩项目换行)

- 当伸缩容器在**主轴上没有足够的空间**容纳所有的伸缩项目, 伸缩容器设置了`flex-wrap: wrap`, 并且伸缩项目没有设置大小, 完全由内容“撑起”, 如果设置
  - `align-content: center`



## 侧轴对齐 (伸缩项目换行)

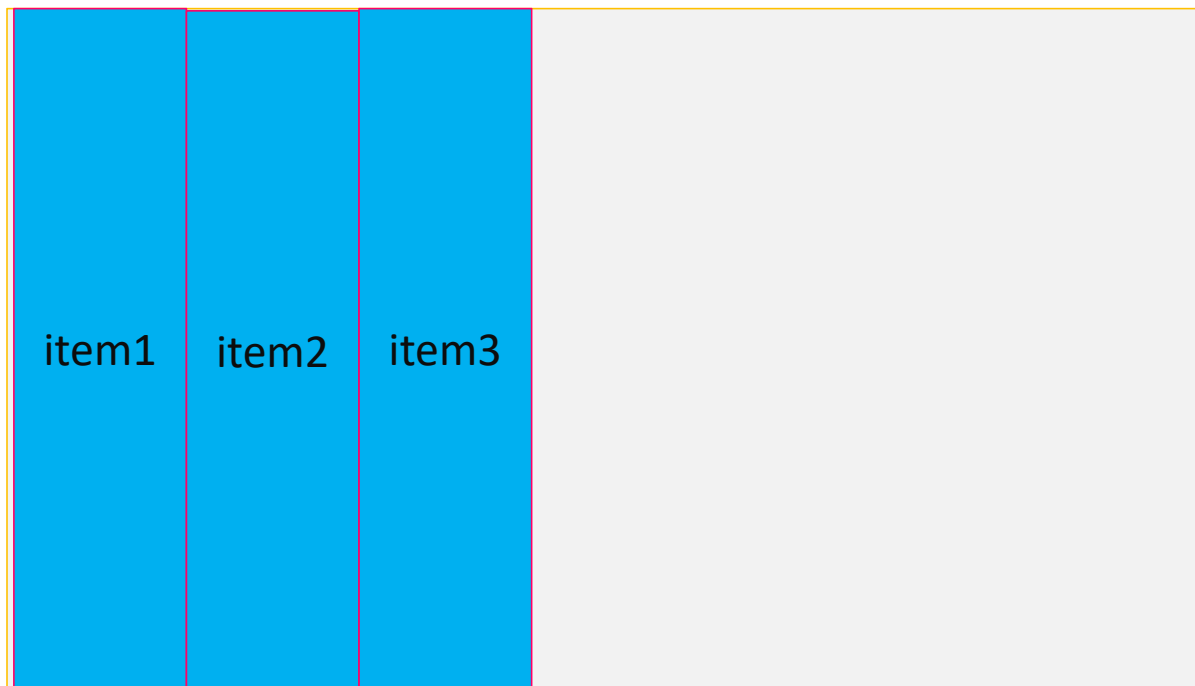
- 当伸缩容器在**主轴上没有足够的空间**容纳所有的伸缩项目, 伸缩容器设置了`flex-wrap: wrap`, 并且伸缩项目没有设置大小, 完全由内容“撑起”, 如果设置
  - `align-content: flex-end`



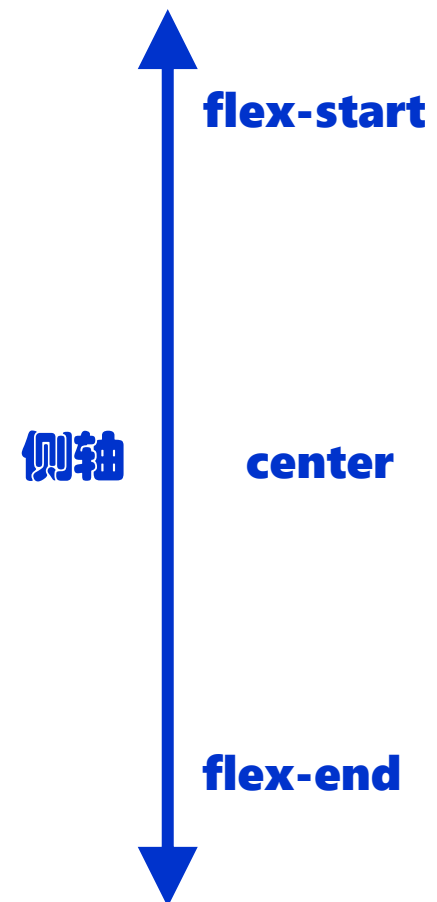
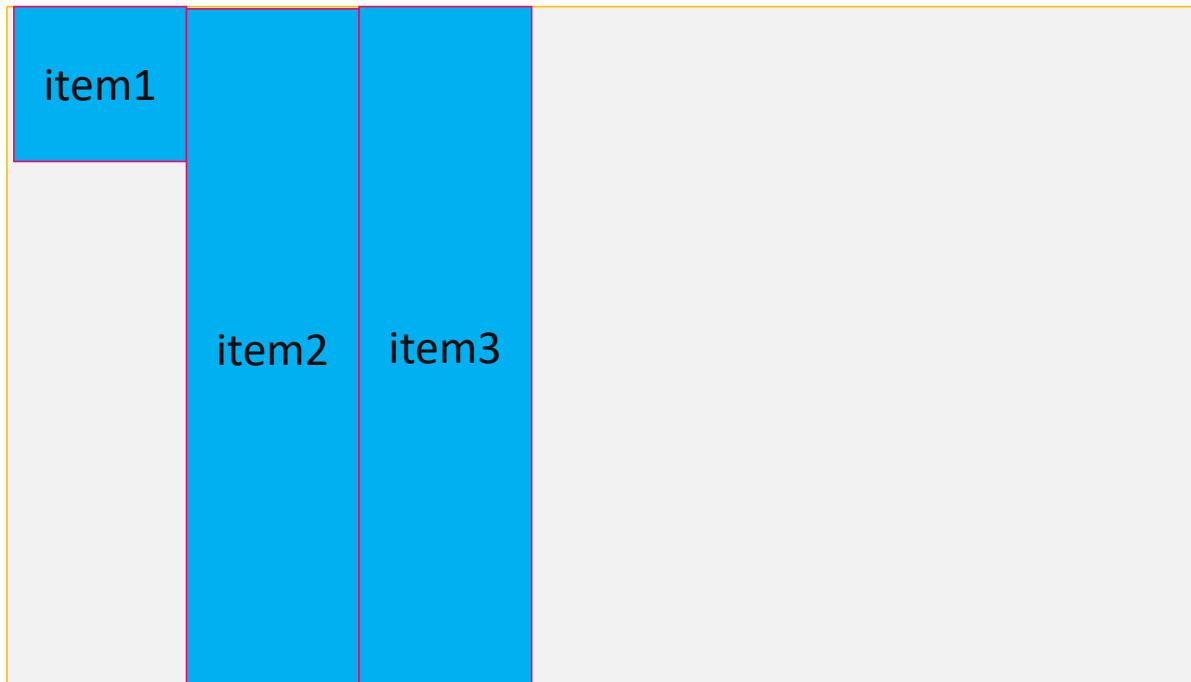


# 伸缩项目在侧轴排列

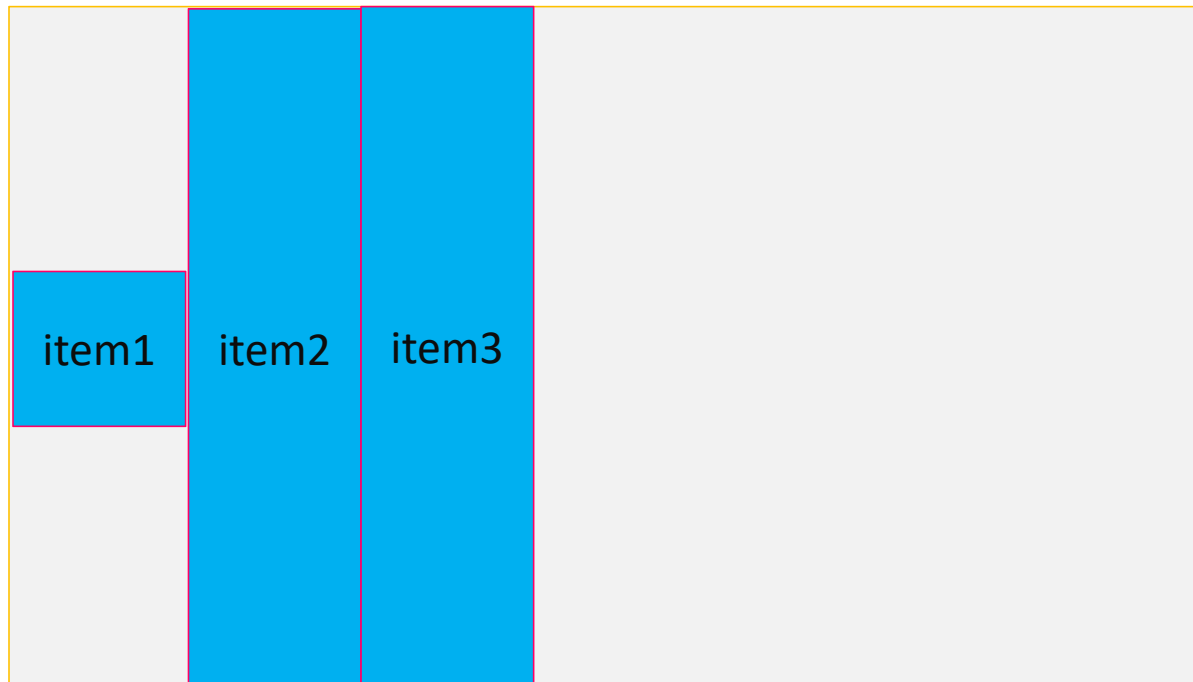
- 当伸缩容器足够大时，可以分别设置每个伸缩项目在伸缩容器侧轴的位置，例如默认值
  - align-self: stretch;



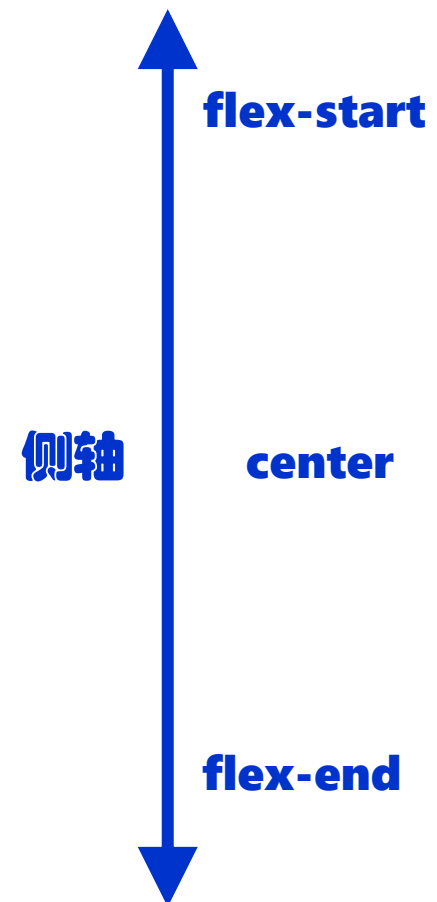
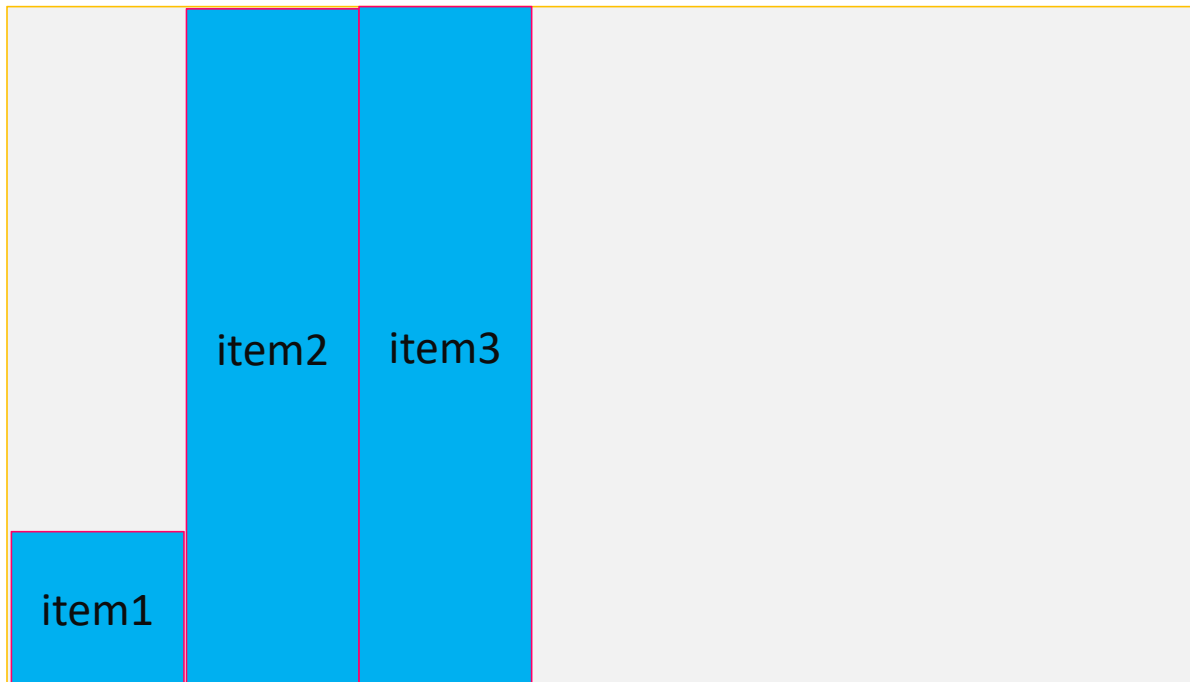
- 当伸缩容器足够大时，可以分别设置每个伸缩项目在伸缩容器侧轴的位置，例如为item1设置了
  - align-self: flex-start;



- 当伸缩容器足够大时，可以分别设置每个伸缩项目在伸缩容器侧轴的位置，例如为item1设置了
  - align-self: center;



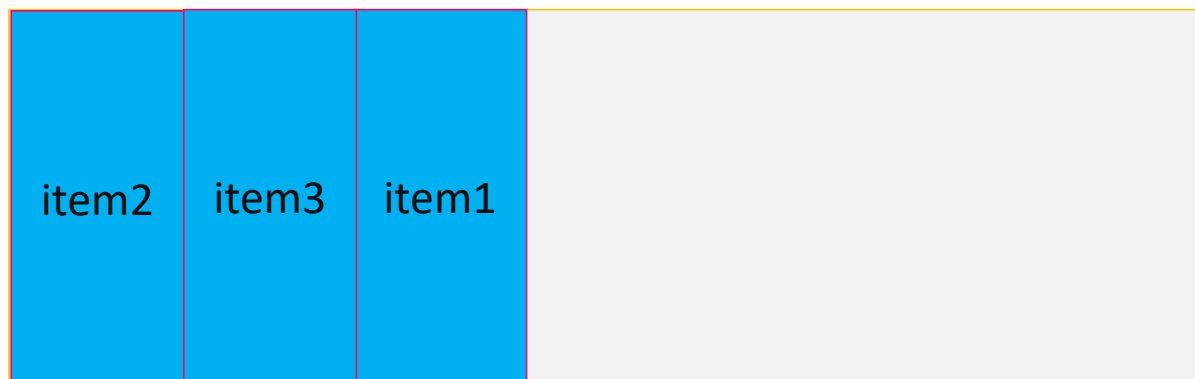
- 当伸缩容器足够大时，可以分别设置每个伸缩项目在伸缩容器侧轴的位置，例如为item1设置了
  - align-self: flex-end;



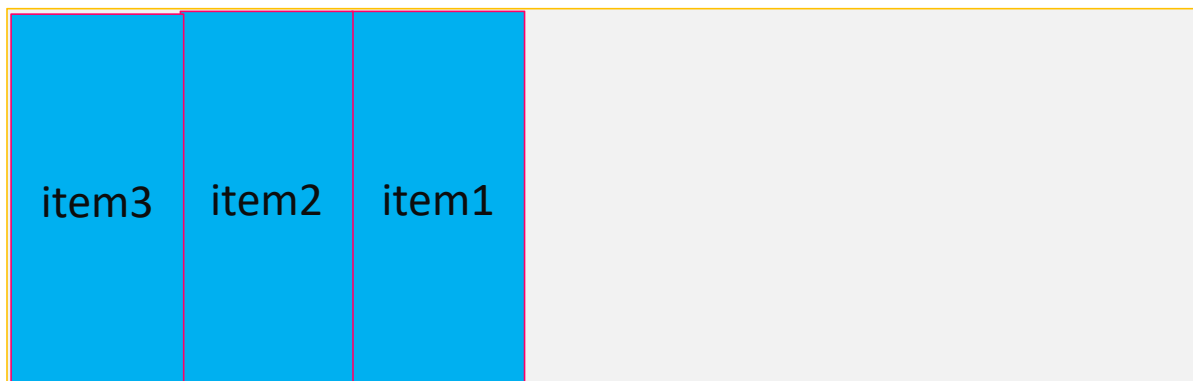


# 伸缩项目在主轴排列

- 当伸缩容器足够大时，可以分别设置每个伸缩项目在伸缩容器主轴的位置，例如为item1设置了
  - `order: 1;`



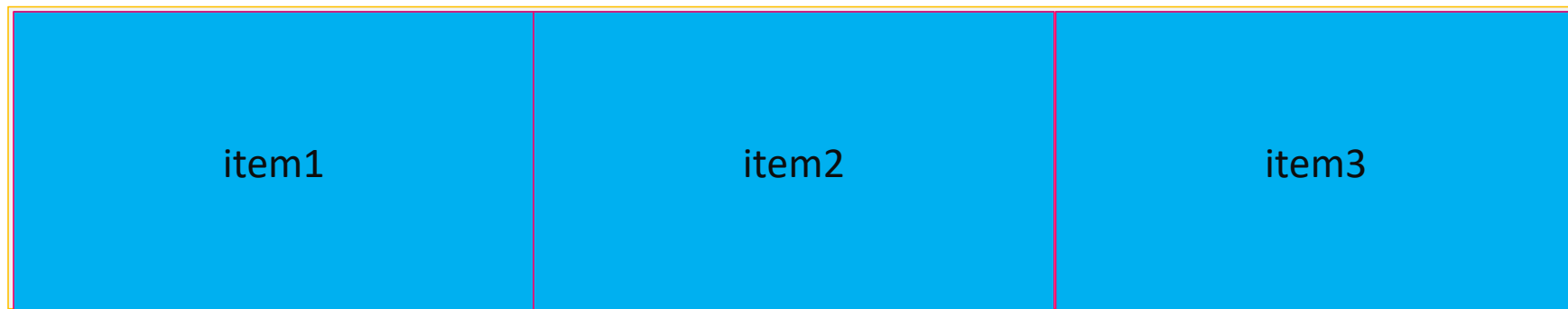
- 当伸缩容器足够大时，可以分别设置每个伸缩项目在伸缩容器主轴的位置，例如为item1，item2分别设置
  - order: 2;
  - order: 1;



# 伸缩项目 分配主轴剩余空间

- 当伸缩容器主轴空间足够大时，可以分别设置每个伸缩项目如何分配主轴空间。其属性为，
  - flex-grow

- 





伸缩容器  
**widht: 900px**

计算公式:  
$$900 / ( 1 + 1 + 1 ) = 300$$

伸缩项目1

flex-grow: 1

伸缩项目2

flex-grow: 1

伸缩项目3

flex-grow: 1

伸缩项目1  
实际宽度  
**300px**

伸缩项目2  
实际宽度  
**300px**

伸缩项目3  
实际宽度  
**300px**

伸缩容器  
**widht: 900px**

计算公式:  
 $900 / (2 + 1 + 1) = 225$

伸缩项目1

flex-grow: 2

伸缩项目2

flex-grow: 1

伸缩项目3

flex-grow: 1

伸缩项目1  
实际宽度  
**510px**

伸缩项目2  
实际宽度  
**225px**

伸缩项目3  
实际宽度  
**225px**

# 设置伸缩项目 初始宽度

- 有两种方式可以设置伸缩项目的初始宽度
  - ① width : length
  - ② flex-basis : length
- 使用flex-basis为元素指定宽度，效果与width相同
- 当元素设置了初始宽度，且伸缩容器主轴空间不足，伸缩项目会被压缩。

# 伸缩项目 如何被压缩



- 当设置**伸缩项目**大小，且**伸缩容器**无法在主轴方向上容纳所有的伸缩项目时，**每个伸缩项目都会被按比例被压缩**。默认如同设置了
  - flex-shrink: 1
- 如果不期望伸缩项目被压缩，可以设置
  - flex-shrink: 0
  - **以上设置会导致伸缩项目在主轴方向“溢出”**

伸缩容器  
width: 400px

伸缩项目1

flex-shrink : 1  
flex-basis: 100px

伸缩项目2

flex-shrink 2  
flex-basis: 200px

伸缩项目3

flex-shrink : 1  
flex-basis: 300px



一、需要压缩的宽度

$$100 + 200 + 300 - 400 = 200$$

二、“总压缩基数”

$$100 * 1 + 200 * 2 + 300 * 1 = 800$$

三、每个伸缩项目实际压缩率

$$100 * 1 / 800 = 0.125$$

$$200 * 2 / 800 = 0.5$$

$$300 * 1 / 800 = 0.375$$

四、每个伸缩项目实际需要压缩空间

$$200 * 0.125 = 25$$

$$200 * 0.5 = 100$$

$$200 * 0.375 = 75$$

五、每个伸缩项目实际的宽度

$$100 - 25 = 75$$

$$200 - 100 = 100$$

$$300 - 75 = 225$$

伸缩项目1  
实际宽度  
75px

伸缩项目2  
实际宽度  
100px

伸缩项目3  
实际宽度  
225px

