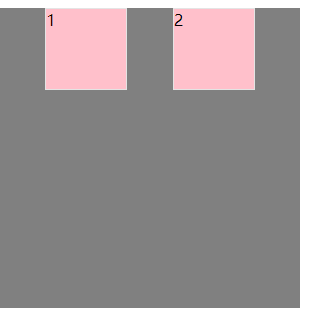
# 主轴排列方式(justify-content)

## 1.内部view平均分

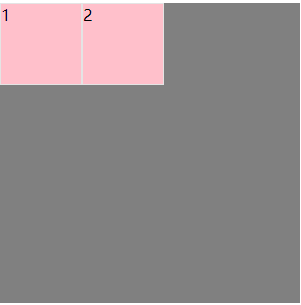
* 1. .outer{
  2. width: 300px;
  3. height: 300px;
  4. background-color: gray;
  5. display: flex;
  6. justify-content: space-evenly;
  7. }
  8. .inner{
  9. width: 80px;
  10. height: 80px;
  11. background-color:pink;
  12. border: 1px solid #e4e4e4;
  13. }



## flex-start

display: flex;

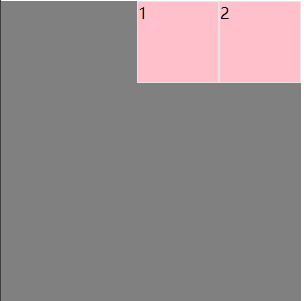
justify-content: flex-start;



## flex-end

display: flex;

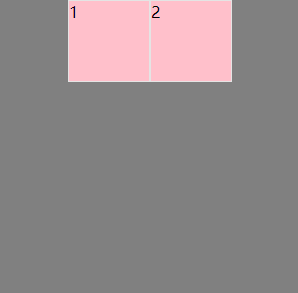
justify-content: flex-end;



## 4.center

display: flex;

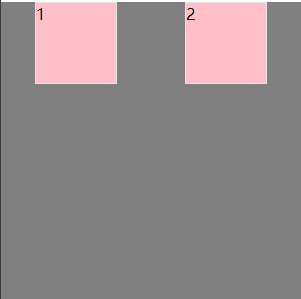
  justify-content: center;



## space-around(1,2到父容器的距离使1，2直接距离的一半)

display: flex;

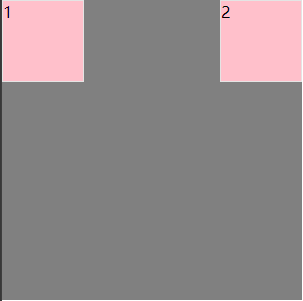
justify-content: space-around;



## space-between

display: flex;

justify-content: space-between;

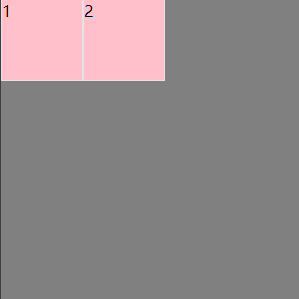


# 侧轴方向排列方式

## flex-start

display: flex;

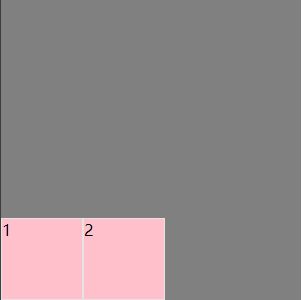
align-items: flex-start;



## flex-end

display: flex;

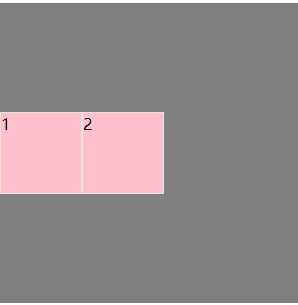
align-items: flex-end;



## center

display: flex;

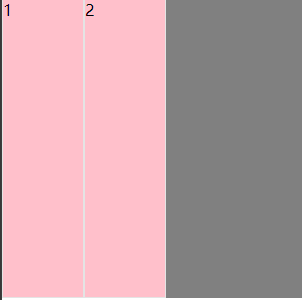
align-items: center;



## stretch

* 1. .outer{
  2. width: 300px;
  3. height: 300px;
  4. background-color: gray;
  5. display: flex;
  6. align-items: stretch;
  7. }
  8. .inner{
  9. width: 80px;
  10. background-color:pink;
  11. border: 1px solid #e4e4e4;
  12. }

.inner当中没有height元素, 所以会被拉升到父容器高度

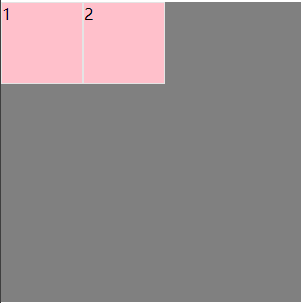


# 改变主轴和侧轴方向

## row

display: flex;

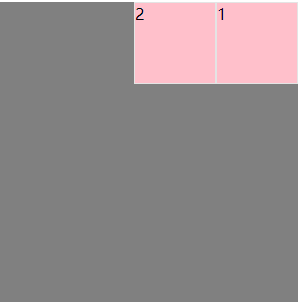
flex-direction: row;



## row-reverse

display: flex;

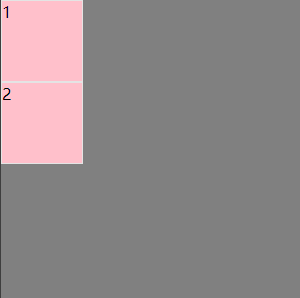
flex-direction: row-reverse;



## column

display: flex;

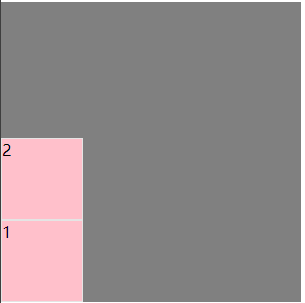
flex-direction: column;



## 4.column-reverse

display: flex;

flex-direction: column-reverse;

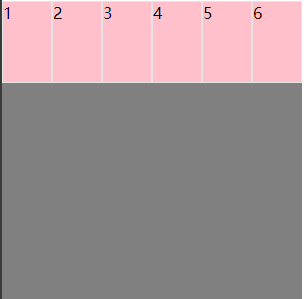


# 换行

## nowrap (默认)

display: flex;

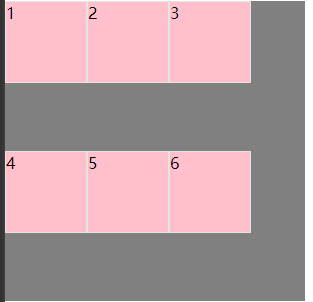
flex-wrap: nowrap;



## wrap

display: flex;

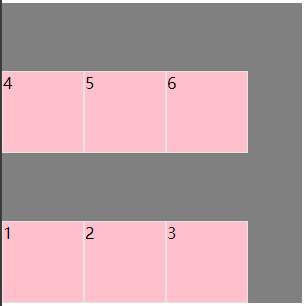
flex-wrap: wrap;



## wrap-reverse

display: flex;

flex-wrap: wrap-reverse;



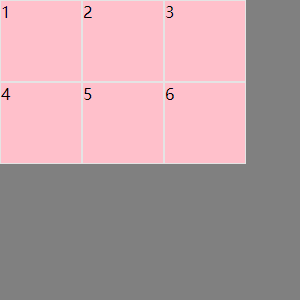
# 多行的排列方式(换行后紧紧挨着)

## flex-start

display: flex;

flex-wrap: wrap;

align-content: flex-start;

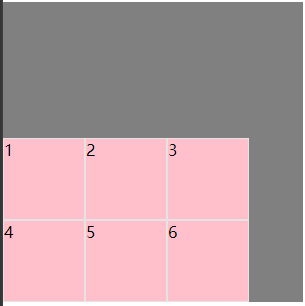


## flex-end

display: flex;

flex-wrap: wrap;

align-content: flex-end;

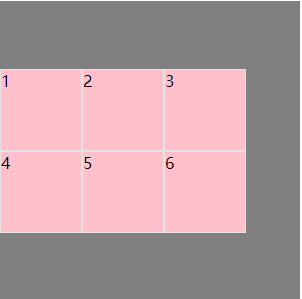


## center

display: flex;

flex-wrap: wrap;

align-content: center;

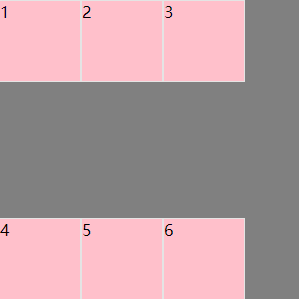


## space-between

display: flex;

flex-wrap: wrap;

align-content: space-between;

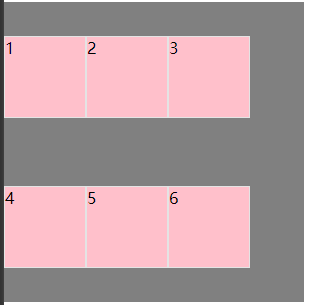


## space-around

display: flex;

flex-wrap: wrap;

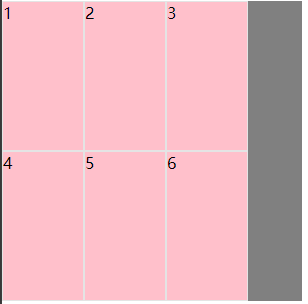
align-content: space-around;



## stretch

* 1. .outer{
  2. width: 300px;
  3. height: 300px;
  4. background-color: gray;
  5. display: flex;
  6. flex-wrap: wrap;
  7. align-content: stretch;
  8. }
  9. .inner{
  10. width: 80px;
  11. background-color:pink;
  12. border: 1px solid #e4e4e4;
  13. }

Inner当中取消height属性才能拉升

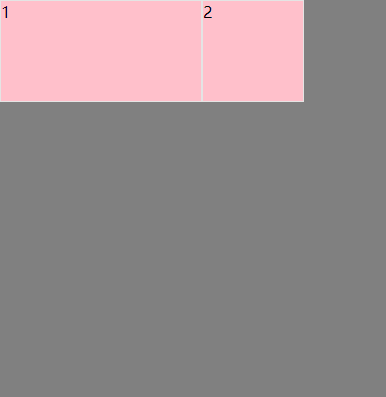


# 子元素的相关元素

## flex-basis

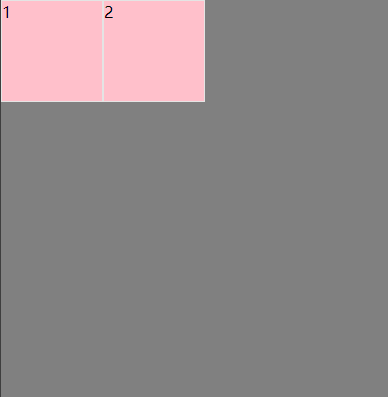
(1)设置为auto, inner1和inner2根据自己的宽度走

1. .outer .inner1{
2. width: 200px;
3. height: 100px;
4. background-color: pink;
5. border: 1px solid #e4e4e4;
6. }
7. .outer .inner2{
8. width: 100px;
9. height: 100px;
10. background-color: pink;
11. border: 1px solid #e4e4e4;
12. }
13. .outer .inner1{
14. flex-basis: auto;
15. }
16. .outer .inner2{
17. flex-basis: auto;
18. }



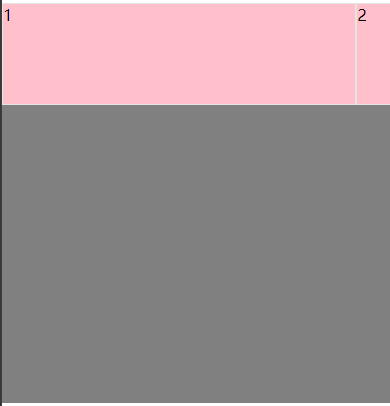
(2) 设置为固定值, 就按这个固定值大小进行显示

1. .outer .inner1{
2. flex-basis: 100px;
3. }
4. .outer .inner2{
5. flex-basis: 100px;
6. }



## 2. flex-grow (分比例划分)

1. .outer .inner1{
2. flex-grow: 10;
3. }
4. .outer .inner2{
5. flex-grow: 1;
6. }



## 3 flex-shrink

1. .inner1, .inner2, .inner3, .inner4, .inner5, .inner6{
2. width: 200px;
3. height: 100px;
4. background-color: pink;
5. border: 1px solid #e4e4e4;
6. }
7. .inner1{
8. flex-shrink: 0;
9. }

.inner1设置为0不允许缩小

