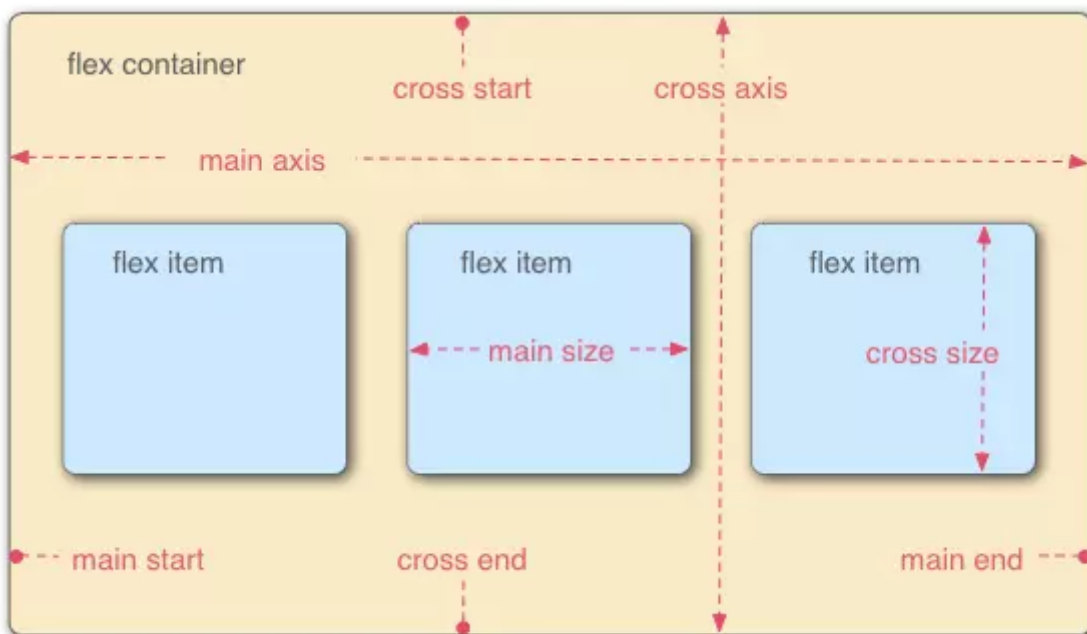


<https://juejin.im/post/599970f4518825243a78b9d5>

方法：display : flex

效果：子元素的float , clear和vertical-align失效



在使用 flex 的元素中，默认存在两根轴：

水平的主轴（main axis）和垂直的交叉轴（cross axis）

### 搞懂主轴的概念很重要

主轴就是确定这个框水平方向的轴

交叉轴就是主轴的90°垂直方向

主轴开始的位置称为 main start

主轴结束的位置称为 main end

交叉轴开始的位置称为 cross start

交叉轴结束的位置称为 cross end

在使用 flex 的子元素中

占据的主轴空间叫做 main size

占据的交叉轴空间叫做 cross size

## 父容器属性

# FLEX PROPERTIES

display: flex   inline-flex;	flex-grow: <number>;
flex-direction: row   column	flex-shrink: <number>;
flex-wrap: wrap   nowrap   wrapreverse	flex: <integer>;
flex-basis: <length>	order:<integer>;
justify-content: flex-start   flex-end   center	
align-self: flex-start   flex-end   center	
align-items: flex-start   flex-end   center	
align-content: flex-start   flex-end   center	

父容器上有六个属性 [测试](#)

1, 主轴方向：

```
1 flex-direction: row | row-reverse | column | column-reverse
2 // 确定主轴的方向，是行（水平），还是列（垂直），加上反向
```

2, 超出样式：

```
1 flex-wrap: nowrap | wrap | wrap-reverse
2 // 当排列的子项目超出父元素时的样式
```

nowrap: 不会超出，最大不超过父元素，相当于max-width = 100%

wrap: 超出了，另起一行，顺序排列，相当于float:left的效果

wrap-reverse: wrap倒过来

3, flex-flow

```
1 flex-flow: flex-direction || flex-wrap
2 // 1+2
```

#### 4, 主轴对齐方式（水平对齐）

```
1 justify-content: flex-start | flex-end | center | space-between | space-around;
```

左对齐：flex-start

右对齐：flex-end

居中对齐：center

间距相等（每个项目左右的空白距离一样）：space-between

外间距数值相等（所有项目左右margin相同，左右累加）：space-around

#### 5，交叉轴对齐方式（垂直对齐, 主要是单轴线）

```
1 align-items: flex-start | flex-end | center | baseline | stretch;
```

备注: 垂直对齐需要指定行高和项目高度，否则会默认auto，填满100%高度

顶端对齐：flex-start

尾部对齐：flex-end

居中对齐：center

拉伸（100%高度）：stretch

底线对齐（第一行文字底部对齐）：baseline

#### 6，多轴线对齐方式（针对多轴线的垂直对齐）

```
1 align-content: flex-start | flex-end | center | baseline | stretch;
```

- flex-direction：主轴的方向。
- flex-wrap：超出父容器子容器的排列样式。
- flex-flow：flex-direction 属性和 flex-wrap 属性的简写形式。
- justify-content：子容器在主轴的排列方向。
- align-items：子容器在交叉轴的排列方向。
- align-content：多根轴线的对齐方式。