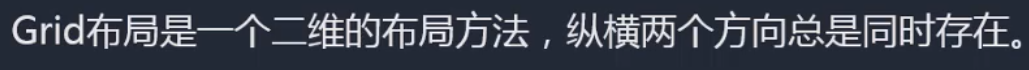
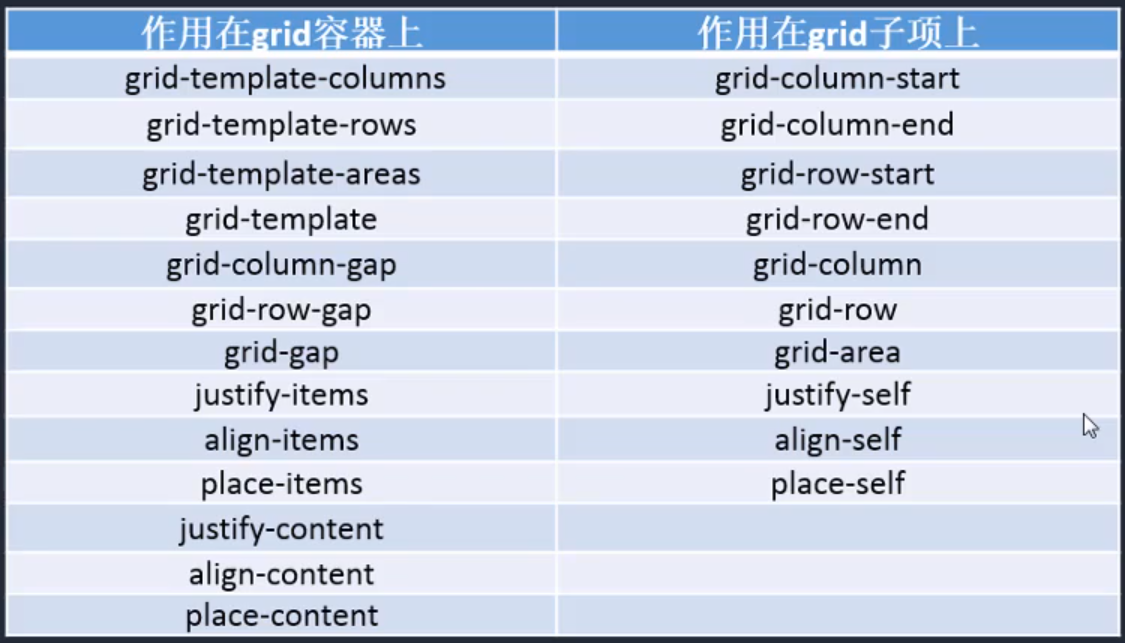
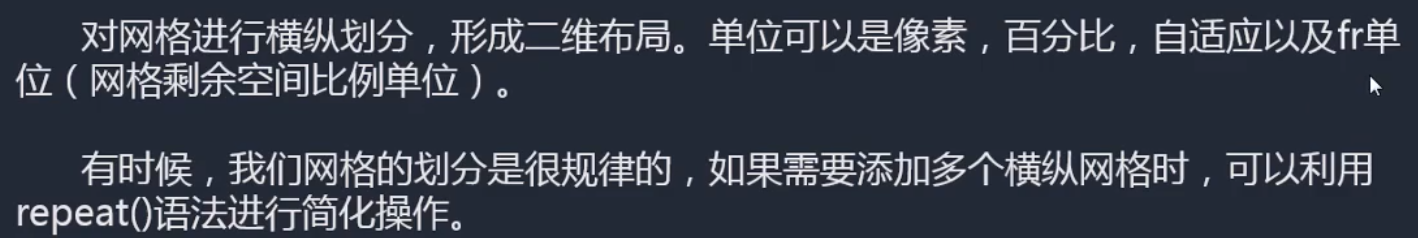
1. 概念：



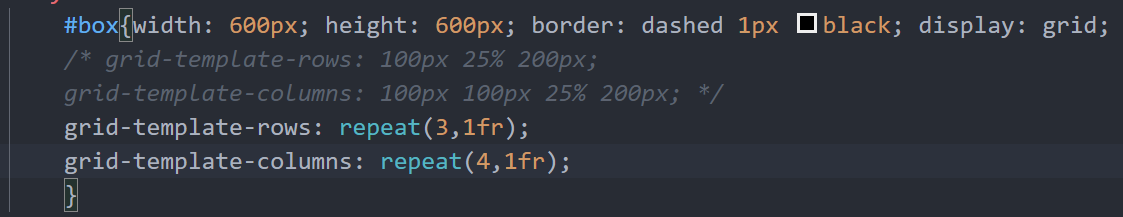


1. 添加在父容器上的属性：
   1. grid-template-columns和grid-template-rows：



[注]fr是比例单位，表示该格占总容器的比例是多少，但行的fr或者列的fr累加和大于等于1时，会将行或者列占满，小于1时则无法占满。

写法：

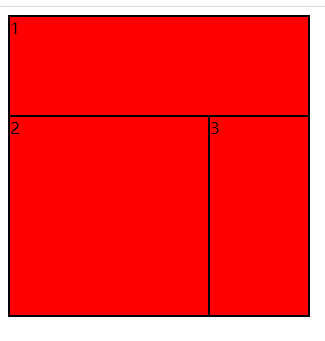


* 1. grid-template-ares和grid-template：

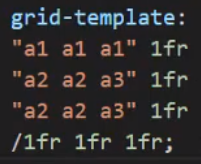


* + 1. grid-template-ares:可以用少量的子容器形成一定的矩形区域：

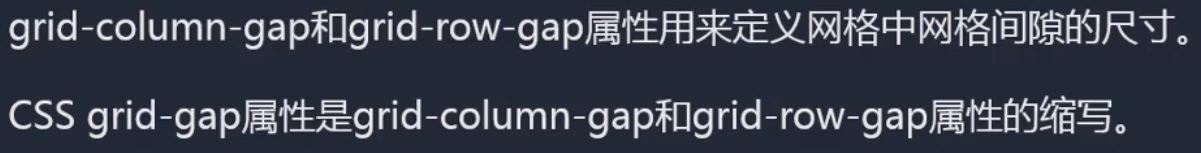




* 1. grid-template的复合写法：



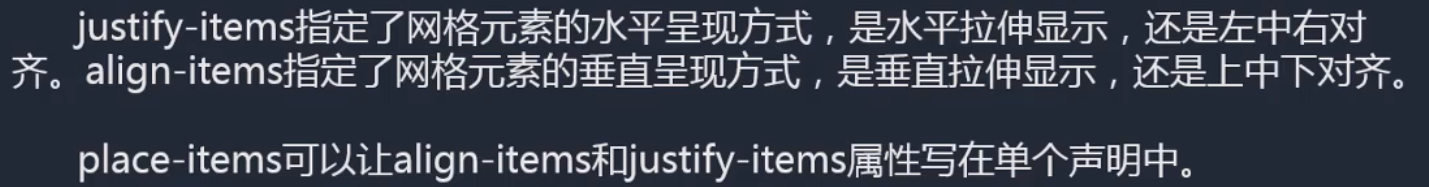
* 1. grid-column-gap和grid-row-gap：

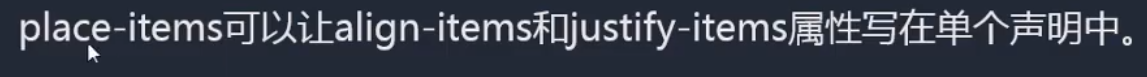


* + 1. grid-column-gap定义纵向间距，grid-row-gap定义横向间距
    2. grid-gap：复合写法，先写横向间距，再写纵向间距：



* 1. justify-items和align-items：items针对每一个网格内的元素的排列：







* 1. justify-content和align-content：规定整个网格的排列方式，(在父容器有空余空间时才能看出效果)：

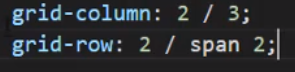




1. 添加在子项上的属性：



* 1. grid-column:表示网格左右两条线(纵向)的起始和结束位置
  2. grid-row：表示网格上下两条线(横向)的起始和结束位置；
     1. span：若设置-end: span 2，则表示该网格在横向或者纵向占用的个数；
  3. grid-column和grid-row的复合写法：



* 1. grid-area:表示当前网格所占用的区域，可以用名字表示法，也可以用位置表示法(和父容器中grid-template-areas配合使用)：
     1. 位置表示法：row-start / column-start / row-end / column-end：



* 1. justify-self,align-self,place-self：和父容器中place-item一样，是控制单个网格内元素排列方式：

