

Flex Layout

flex 是 Flexible Box (弹性盒子) 的缩写，意为"弹性布局"，可以简便、完整、响应式地实现各种页面布局, 也是HarmonyJS UI 容器默认的布局模式
当前版本Harmony上flex未对齐W3C发布的flex, 存在某些属性缺失和不一致, 本文是基于 v2.1.1.21 功能讲解

1. 基础概念

采用 Flex 布局的元素，称为 Flex 容器（flex container），简称"容器"。它的所有子元素自动成为容器成员，称为 Flex 元素（flex item），简称"元素"。

1.1 flex容器属性

- flex-direction**
flex-direction 属性决定主轴的方向（即元素的排列方向）

属性值	描述	示例
row	(默认值) 主轴为水平方向	<div>flex-direction: row</div> <div>A B C D</div>
column	主轴为垂直方向	<div>flex-direction: column</div> <div>A</div> <div>B</div> <div>C</div> <div>D</div>

- flex-wrap**
flex-wrap 属性决定当子元素铺满一条轴时是否应该换行

属性值	描述	示例
nowrap	(默认值) 不换行, 缩小子元素	<div>flex-wrap: nowrap</div> <div>A B C D E</div>
wrap	换行, 类似Android中的流布局	<div>flex-wrap: wrap</div> <div>A B C</div> <div>D E</div>

- justify-content**
justify-content 属性定义了元素在主轴上的对齐方式

属性值	描述	示例 (flex-direction: column)
-----	----	-----------------------------

属性值	描述	示例 (flex-direction: column)
flex-start	(默认值) 对齐主轴起点	<div>justify-content: flex-start</div> <div><div>A</div><div>B</div><div>C</div><div>D</div></div>
flex-end	对齐主轴终点	<div>justify-content: flex-end</div> <div><div>A</div><div>B</div><div>C</div><div>D</div></div>
center	主轴居中	<div>justify-content: center</div> <div><div>A</div><div>B</div><div>C</div><div>D</div></div>
space-between	两端对齐，元素之间的间隔都相等	<div>justify-content: space-between</div> <div><div>A</div><div>B</div><div>C</div><div>D</div></div>
space-around	每个元素两侧的间隔相等。所以，元素之间的间隔比元素与边框的间隔大一倍	<div>justify-content: space-around</div> <div><div>A</div><div>B</div><div>C</div><div>D</div></div>

• align-items

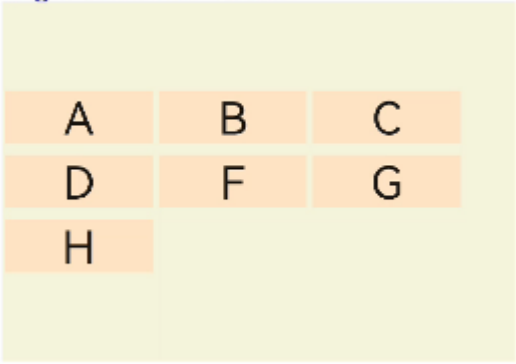
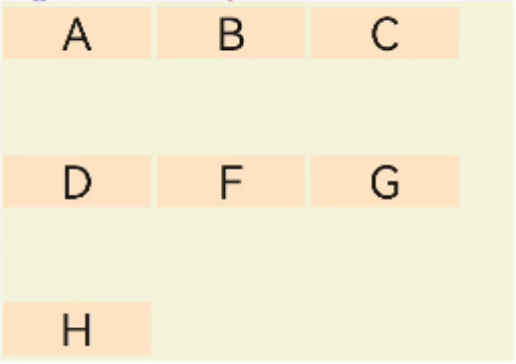
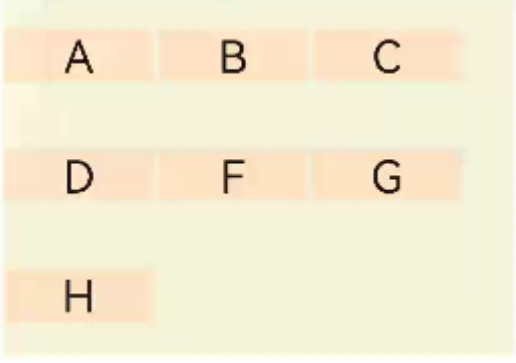
align-items 属性定义元素在交叉轴上如何对齐。

属性值	描述	示例(flex-direction: column)
flex-start	(默认值)交叉轴的起点对齐	<div>align-items: flex-start</div> <div><div>A</div><div>B</div><div>C</div><div>D</div></div>

属性值	描述	示例(flex-direction: column)
flex-end	交叉轴的终点对齐	<div><div>align-items: flex-end</div><div><div>A</div><div>B</div><div>C</div><div>D</div></div></div>
center	交叉轴居中	<div><div>align-items: center</div><div><div>A</div><div>B</div><div>C</div><div>D</div></div></div>
stretch	如果元素未设置高度或设为auto，将占满整个容器的高度	<div><div>align-items: stretch</div><div><div>A</div><div>B</div><div>C</div><div>D</div></div></div>

- align-content
 - align-content 属性定义了多根轴线的对齐方式。只在两种情况下有效果：
 - 子项多行且 flex 容器高度固定
 - 子项单行，flex 容器高度固定且设置了 flex-wrap:wrap；

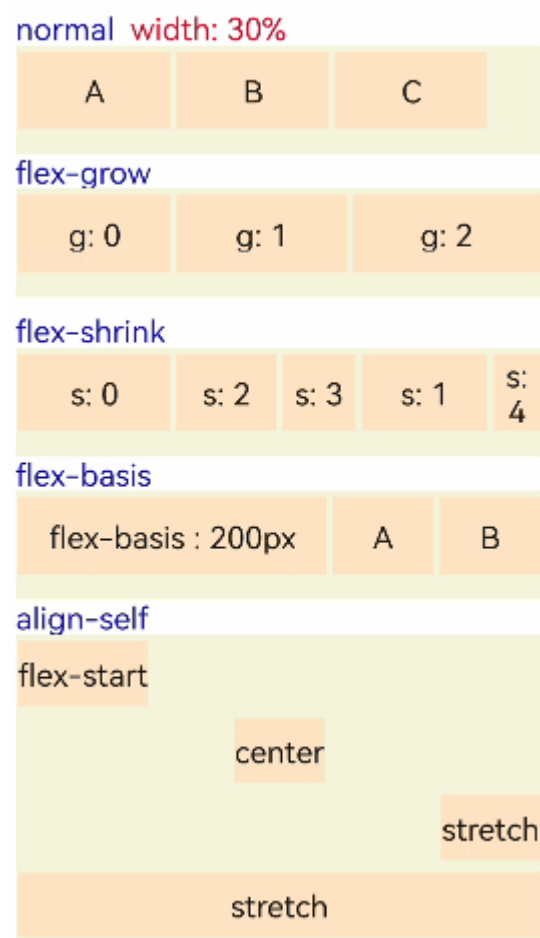
属性值	描述	示例(flex-direction: row)
flex-start	与交叉轴的起点对齐	<div><div>align-content: flex-start</div><div><div><div>A</div><div>B</div><div>C</div></div><div><div>D</div><div>F</div><div>G</div></div><div><div>H</div></div></div></div>
flex-end	与交叉轴的终点对齐	<div><div>align-content: flex-end</div><div><div><div>A</div><div>B</div><div>C</div></div><div><div>D</div><div>F</div><div>G</div></div><div><div>H</div></div></div></div>

属性值	描述	示例(flex-direction: row)
center	与交叉轴的中点对齐	<div>align-content: center</div> 
space-between	与交叉轴两端对齐，轴线之间的间隔平均分布	<div>align-content: space-between</div> 
space-around	每根轴线两侧的间隔都相等。所以，轴线之间的间隔比轴线与边框的间隔大一倍	<div>align-content: space-around</div> 

- align-items 和 align-content 的区别 + 1. align-items 的作用对象是**每一个item**, align-content 的作用对象是**item构成的行**

1.2 flex元素属性

- flex-grow**
 - flex-grow 属性定义元素的放大比例，默认为0，即如果存在**剩余空间**，也不放大。
 - 如果所有元素的 flex-grow 属性都为1，则它们将等分剩余空间（如果有的话）。如果一个元素的 flex-grow 属性为2，其他元素都为1，则前者占据的剩余空间将比其他项多一倍。
- flex-shrink**
 - flex-shrink 属性定义了元素的缩小比例，默认为1，即如果**空间不足**，该元素将缩小。
 - 如果所有元素的 flex-shrink 属性都为1，当空间不足时，都将等比例缩小。如果一个元素的 flex-shrink 属性为0，其他元素都为1，则空间不足时，前者不缩小。
 - 负值对该属性无效。
- flex-basis
 - flex-basis 属性定义了分配多余空间之前，元素占据的主轴空间（main size）。浏览器根据这个属性，计算主轴是否有多余空间。它的默认值为 auto，即元素的本来大小。
 - 它可以设为跟width或height属性一样的值（比如350px），则元素将占据固定空间。
- align-self**
 - align-self 属性允许单个元素有与其他元素不一样的对齐方式，可覆盖 align-items 属性。默认值为 auto，表示继承父元素的 align-items 属性，如果没有父元素，则等同于 stretch。



2. 实战练习
