**一、flex布局是什么？**

1、Flex 是 Flexible Box 的缩写，意为"弹性布局"，用来为盒状模型提供最大的灵活性。

任何一个容器都可以指定为 Flex 布局。

2、注意，设为 Flex 布局以后，子元素的float、clear和vertical-align属性将失效。

3、浏览器对最新flexbox规范的支持情况：

Chrome 29+，Firefox 28+，Internet Explorer 11+，Opera 17+，Safari 6.1+ (prefixed with -webkit-)，Android 4.4+，iOS 7.1+ (prefixed with -webkit-)

4、使用Flexbox布局只要在父容器元素上设置display属性；

**二、基本概念**

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 项目（flex item），简称"项目"。



容器默认存在两根轴：水平的主轴（main axis）和垂直的交叉轴（cross axis）。主轴的开始位置（与边框的交叉点）叫做main start，结束位置叫做main end；交叉轴的开始位置叫做cross start，结束位置叫做cross end。

项目默认沿主轴排列。单个项目占据的主轴空间叫做main size，占据的交叉轴空间叫做cross size。

**三、容器的属性**

1、flex-direction：属性决定主轴的方向（即项目的排列方向）

（1）row（默认值）：主轴为水平方向，起点在左端。

（2）row-reverse：主轴为水平方向，起点在右端。

（3）column：主轴为垂直方向，起点在上沿。

（4）column-reverse：主轴为垂直方向，起点在下沿。

2、flex-wrap：默认情况下，项目都排在一条线（又称"轴线"）上。flex-wrap属性定义，如果一条轴线排不下，如何换行。

（1）nowrap（默认）：不换行。

（2）wrap：换行，第一行在上方。

（3）wrap-reverse：换行，第一行在下方。

3、flex-flow：是flex-direction属性和flex-wrap属性的简写形式，默认值为row nowrap。

4、justify-content：定义了项目在主轴上的对齐方式。

具体对齐方式与轴的方向有关。下面假设主轴为从左到右。

（1）flex-start（默认值）：左对齐

（2）flex-end：右对齐

（3）center： 居中

（4）space-between：两端对齐，项目之间的间隔都相等。

（5）space-around：每个项目两侧的间隔相等。所以，项目之间的间隔比项目与边框的间隔大一倍。

5、align-items: 定义项目在交叉轴上如何对齐。

具体的对齐方式与交叉轴的方向有关，下面假设交叉轴从上到下。

（1）flex-start：交叉轴的起点对齐。

（2）flex-end：交叉轴的终点对齐。

（3）center：交叉轴的中点对齐。

（4）baseline: 项目的第一行文字的基线对齐。

（5）stretch（默认值）：如果项目未设置高度或设为auto，将占满整个容器的高度。

6、align-content：定义了多根轴线的对齐方式。如果项目只有一根轴线，该属性不起作用。

flex-start：与交叉轴的起点对齐。

（1）flex-end：与交叉轴的终点对齐。

（2）center：与交叉轴的中点对齐。

（3）space-between：与交叉轴两端对齐，轴线之间的间隔平均分布。

（4）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。

（5）stretch（默认值）：轴线占满整个交叉轴。

**四、项目属性**

1、order：定义项目的排列顺序。数值越小，排列越靠前，默认为0

2、flex-grow：定义项目的放大比例，默认为0，即如果存在剩余空间，也不放大。如果所有项目的flex-grow属性都为1，则它们将等分剩余空间（如果有的话）。如果一个项目的flex-grow属性为2，其他项目都为1，则前者占据的剩余空间将比其他项多一倍。

3、flex-shrink：定义了项目的缩小比例，默认为1，即如果空间不足，该项目将缩小。

如果所有项目的flex-shrink属性都为1，当空间不足时，都将等比例缩小。如果一个项目的flex-shrink属性为0，其他项目都为1，则空间不足时，前者不缩小。

4、flex-basis：定义了在分配多余空间之前，项目占据的主轴空间（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，即项目的本来大小。它可以设为跟width或height属性一样的值（比如350px），则项目将占据固定空间。

5、flex：flex-grow, flex-shrink 和 flex-basis的简写，默认值为0 1 auto。后两个属性可选。该属性有两个快捷值：auto (1 1 auto) 和 none (0 0 auto)。

6、align-self：允许单个项目有与其他项目不一样的对齐方式，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。该属性可能取6个值，除了auto，其他都与align-items属性完全一致。

**五、box属性**

必须给父容器wrap定义css属性display:box其子容器才可以进行划分(如果定了display:box则该容器则定义为了内联元素，使用margin:0px auto让其居中是无效的，要想使其居中只能通过它的父容器的text-align:center);

css3弹性盒子模型之box-flex的box属性包括：box-orient | box-direction | box-align | box-pack | box-lines

1、box-orient

box-orient(orient译为排列更准确)用来确定父容器里子容器的排列方式，是水平还是垂直。可选属性如下所示：

horizontal | vertical | inline-axis | block-axis | inherit

①horizontal、inline-axis：将子容器水平排列，

如果父容器定义了高度值，其子容器的高度值设置则无效状态，所有子容器的高度等于父容器的高度值；如果父容器不设置高度值，其子容器的高度值才有效并且取最大高度值的子容器的高度。

②vertical、block-axis：将子容器垂直排列

如果父容器定义了宽度值，其子容器的宽度值设置则无效状态；如果父容器不设置宽度值，其子容器的宽度值才有效并且取最大宽度值的子容器的宽度。

inherit属性则是让子容器继承父容器的相关属性。

2、box-direction

box-direction用来确定父容器里的子容器排列顺序3、box-align

3、box-align表示父容器里面子容器的垂直对齐方式，可选参数如下所示：

start | end | center | baseline | stretch，参考align-items

4、box-pack

box-pack表示父容器里面子容器的水平对齐方式，可选参数如下所示：

start | end | center | justify，参考justify-content