1.弹性布局

flex 是 Flexible Box 的缩写，意为"弹性布局"。指定容器display: flex即可。

容器有以下属性：flex-direction，flex-wrap，flex-flow，justify-content，align-items，align-content。

flex-direction属性决定主轴的方向；

flex-wrap属性定义，如果一条轴线排不下，如何换行；

flex-flow属性是flex-direction属性和flex-wrap属性的简写形式，默认值为row nowrap；

justify-content属性定义了项目在主轴上的对齐方式。

align-items属性定义项目在交叉轴上如何对齐。

align-content属性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。

项目（子元素）也有一些属性：order，flex-grow，flex-shrink，flex-basis，flex，align-self。

order属性定义项目的排列顺序。数值越小，排列越靠前，默认为0。

flex-grow属性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。

flex-shrink属性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。

flex-basis属性定义了在分配多余空间之前，项目占据的主轴空间（main size）。

flex属性是flex-grow, flex-shrink 和 flex-basis的简写，默认值为0 1 auto。后两个属性可选。

align-self属性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。

2.垂直居中

父元素固定宽高，利用定位及设置子元素margin值为自身的一半。

父元素固定宽高，子元素设置**position: absolute，margin：auto**平均分配margin

css3属性transform。子元素设置**position: absolute; left: 50%; top: 50%; transform: translate(-50%,-50%);即可。**

将父元素设置成**display: table**, 子元素设置为单元格 **display: table-cell**。

弹性布局**display: flex**。设置**align-items: center; justify-content: center;**

3.如何实现元素的垂直居中

参考回答：

法一：父元素display:flex,align-items:center;

法二：元素绝对定位，top:50%，margin-top：-（高度/2）

法三：高度不确定用transform：translateY（-50%）

法四：父元素table布局，子元素设置vertical-align:center;

4.说一下块元素和行元素

块元素：独占一行，并且有自动填满父元素，可以设置margin和pading以及高度和宽度

行元素：不会独占一行，width和height会失效，并且在垂直方向的padding和margin会失效。

5.  visibility=hidden, opacity=0，display:none

参考回答：

opacity=0，该元素隐藏起来了，但不会改变页面布局，并且，如果该元素已经绑定一些事件，如click事件，那么点击该区域，也能触发点击事件的visibility=hidden，该元素隐藏起来了，但不会改变页面布局，但是不会触发该元素已经绑定的事件display=none，把元素隐藏起来，并且会改变页面布局，可以理解成在页面中把该元素删除掉一样。

5.  position属性 比较

参考回答：

1. .固定定位fixed：

元素的位置相**对于浏览器窗口是固定位置**，即使窗口是滚动的它也不会移动。Fixed定位使元素的位置与文档流无关，因此不占据空间。 Fixed定位的元素和其他元素重叠。

②.相对定位relative：

如果对一个**元素进行相对定位，它将出现在它所在的位置上。然后，可以通过**设置垂直或水平位置，让这个元素“相对于”它的起点进行移动。 在使用相对定位时，无论是否进行移动，元素仍然占据原来的空间。因此，移动元素会导致它覆盖其它框。

1. .绝对定位absolute：

绝对定位的元素**的位置相对于最近的已定位父元素，如**果元素没有已定位的父元素，那么它的位置相对于<html>。 absolute 定位使元素的位置与文档流无关，因此不占据空间。 absolute 定位的元素和其他元素重叠。

1. .粘性定位sticky：

元素先按照普通文档流定位，然后相对于该元素在流中的flow root（BFC）和 containing block（最近的块级祖先元素）定位。而后，元素定位表现为在跨越特定阈值前为相对定位，之后为固定定位。

1. .默认定位Static：

默认值。没有定位，元素出现在正常的流中（忽略top, bottom, left, right 或者 z-index 声明）。

1. . inherit:

规定应该从父元素继承position 属性的值。

浮动清除

参考回答：

1. 方法一：使用带clear属性的空元素

在浮动元素后使用一个空元素如<div class="clear"></div>，并在CSS中赋予.clear{clear:both;}属性即可清理浮动。亦可使用<br class="clear" />或<hr class="clear" />来进行清理。

1. 方法二：使用CSS的overflow属性

给浮动元素的容器添加overflow:hidden;或overflow:auto;可以清除浮动，另外在 IE6 中还需要触发 hasLayout ，例如为父元素设置容器宽高或设置 zoom:1。

在添加overflow属性后，浮动元素又回到了容器层，把容器高度撑起，达到了清理浮动的效果。

1. 方法三：给浮动的元素的容器添加浮动
2. 给浮动元素的容器也添加上浮动属性即可清除内部浮动，但是这样会使其整体浮动，影响布局，不推荐使用。
3. 方法四：使用邻接元素处理
4. 什么都不做，给浮动元素后面的元素添加clear属性。
5. 方法五：使用CSS的:after伪元素

结合:after 伪元素（注意这不是伪类，而是伪元素，代表一个元素之后最近的元素）和 IEhack ，可以完美兼容当前主流的各大浏览器，这里的 IEhack 指的是触发 hasLayout。

1. 给浮动元素的容器添加一个clearfix的class，然后给这个class添加一个:after伪元素实现元素末尾添加一个看不见的块元素（Block element）清理浮动。

对CSS的新属性有了解过的吗？

参考回答：

CSS3的新特性中，在布局方面新增了flex布局，在选择器方面新增了例如first-of-type,nth-child等选择器，在盒模型方面添加了box-sizing来改变盒模型，在动画方面增加了animation，2d变换，3d变换等，在颜色方面添加透明，rbga等，在字体方面允许嵌入字体和设置字体阴影，最后还有媒体查讯等

用的最多的css属性是啥？

参考回答：

用的目前来说最多的是flex属性，灵活但是兼容性方面不强