## 前端面试题（2）

### 1、link 标签和import 标签的区别

link 属于html 标签，而@import 是css 提供的  
页面被加载时，link 会同时被加载，而@import 引用的css 会等到页面加载结束后加载。  
link 是html 标签，因此没有兼容性，而@import 只有IE5 以上才能识别。  
link 方式样式的权重高于@import 的

### 2、transition 和animation 的区别

Animation 和transition 大部分属性是相同的，他们都是随时间改变元素的属性值，他们  
的主要区别是transition 需要触发一个事件才能改变属性，而animation 不需要触发任何  
事件的情况下才会随时间改变属性值，并且transition 为2 帧，从from .... to，而animation  
可以一帧一帧的。

### 3、Flex 布局

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。  
布局的传统解决方案，基于盒状模型，依赖display 属性+ position 属性+ float 属性。它  
对于那些特殊布局非常不方便，比如，垂直居中就不容易实现。  
简单的分为容器属性和元素属性  
容器的属性：  
flex-direction：决定主轴的方向（即子item 的排列方法）  
.box {  
flex-direction: row | row-reverse | column | column-reverse;  
}  
flex-wrap：决定换行规则  
.box{  
flex-wrap: nowrap | wrap | wrap-reverse;  
}  
flex-flow：  
.box {  
flex-flow: || ;  
}  
justify-content：对其方式，水平主轴对齐方式  
align-items：对齐方式，竖直轴线方向  
项目的属性（元素的属性）：  
order 属性：定义项目的排列顺序，顺序越小，排列越靠前，默认为0  
flex-grow 属性：定义项目的放大比例，即使存在空间，也不会放大flex-shrink 属性：定义了项目的缩小比例，当空间不足的情况下会等比例的缩小，如果  
定义个item 的flow-shrink 为0，则为不缩小  
flex-basis 属性：定义了在分配多余的空间，项目占据的空间。  
flex：是flex-grow 和flex-shrink、flex-basis 的简写，默认值为0 1 auto。  
align-self：允许单个项目与其他项目不一样的对齐方式，可以覆盖align-items，默认属  
性为auto，表示继承父元素的align-items  
比如说，用flex 实现圣杯布局

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。

### 4、BFC（块级格式化上下文，用于清楚浮动，防止margin 重叠等）

直译成：块级格式化上下文，是一个独立的渲染区域，并且有一定的布局规则。  
BFC 区域不会与float box 重叠  
BFC 是页面上的一个独立容器，子元素不会影响到外面  
计算BFC 的高度时，浮动元素也会参与计算  
那些元素会生成BFC：  
根元素  
float 不为none 的元素  
position 为fixed 和absolute 的元素  
display 为inline-block、table-cell、table-caption，flex，inline-flex 的元素  
overflow 不为visible 的元素

### 5、垂直居中的方法

(1)margin:auto 法  
css:

div{  
width: 400px;  
height: 400px;  
position: relative;  
border: 1px solid #465468;  
}  
img{  
position: absolute;  
margin: auto;  
top: 0;  
left: 0;  
right: 0;  
bottom: 0;  
}  
html:<div><img src="mm.jpg"></div>  
//定位为上下左右为0，margin：0 可以实现脱离文档流的居中.

(2)margin 负值法

.container{  
width: 500px;  
height: 400px;  
border: 2px solid #379;  
position: relative;  
}  
.inner{  
width: 480px;  
height: 380px;  
background-color: #746;  
position: absolute;  
top: 50%;  
left: 50%;  
margin-top: -190px; /\*height 的一半\*/  
margin-left: -240px; /\*width 的一半\*/  
}  
//补充：其实这里也可以将marin-top 和margin-left 负值替换成，  
transform：translateX(-50%)和transform：translateY(-50%)

(3)table-cell（未脱离文档流的）  
设置父元素的display:table-cell,并且vertical-align:middle，这样子元素可以实现垂直居中。  
css:

(4)利用flex  
将父元素设置为display:flex，并且设置align-items:center;justify-content:center;

div{  
width: 300px;  
height: 300px;  
border: 3px solid #555;  
display: table-cell;  
vertical-align: middle;  
text-align: center;  
}  
img{  
vertical-align: middle;  
}

(4)利用flex  
将父元素设置为display:flex，并且设置align-items:center;justify-content:center;

css:
  
.container{
  
width: 300px;
  
height: 200px;
  
border: 3px solid #546461;
  
display: -webkit-flex;
  
display: flex;
  
-webkit-align-items: center;
  
align-items: center;
  
-webkit-justify-content: center;
  
justify-content: center;
  
}
  
.inner{
  
border: 3px solid #458761;
  
padding: 20px;
  
}

### 6、关于JS 动画和css3 动画的差异性

渲染线程分为main thread 和compositor thread，如果css 动画只改变transform 和opacity，  
这时整个CSS 动画得以在compositor trhead 完成（而JS 动画则会在main thread 执行，然  
后出发compositor thread 进行下一步操作），特别注意的是如果改变transform 和opacity  
是不会layout 或者paint 的。  
区别：  
功能涵盖面，JS 比CSS 大  
实现/重构难度不一，CSS3 比JS 更加简单，性能跳优方向固定  
对帧速表现不好的低版本浏览器，css3 可以做到自然降级  
css 动画有天然事件支持  
css3 有兼容性问题

### 7、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。

### 8、多行元素的文本省略号

display: -webkit-box
  
-webkit-box-orient:vertical
  
-webkit-line-clamp:3
  
overflow:hidden

### 9、visibility=hidden, opacity=0，display:none

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。

### 10、双边距重叠问题（外边距折叠）

多个相邻（兄弟或者父子关系）普通流的块元素垂直方向marigin 会重叠  
折叠的结果为：  
两个相邻的外边距都是正数时，折叠结果是它们两者之间较大的值。  
两个相邻的外边距都是负数时，折叠结果是两者绝对值的较大值。  
两个外边距一正一负时，折叠结果是两者的相加的和。

### 11、position 属性比较

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。  
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。  
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。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。  
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。  
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）。  
inherit:  
规定应该从父元素继承position 属性的值。

### 12、浮动清除

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，并在CSS 中赋  
予.clear{clear:both;}属性即可清理浮动。亦可使用或  
来进行清理。  
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。  
方法三：给浮动的元素的容器添加浮动  
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。  
方法四：使用邻接元素处理  
什么都不做，给浮动元素后面的元素添加clear 属性。  
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。  
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。

### 13、CSS 选择器有哪些，优先级呢

id 选择器，class 选择器，标签选择器，伪元素选择器，伪类选择器等  
同一元素引用了多个样式时，排在后面的样式属性的优先级高；  
样式选择器的类型不同时，优先级顺序为：id 选择器> class 选择器> 标签选择器；  
标签之间存在层级包含关系时，后代元素会继承祖先元素的样式。如果后代元素定义了  
与祖先元素相同的样式，则祖先元素的相同的样式属性会被覆盖。继承的样式的优先级  
比较低，至少比标签选择器的优先级低；  
带有!important 标记的样式属性的优先级最高；  
样式表的来源不同时，优先级顺序为：内联样式> 内部样式> 外部样式> 浏览器用户  
自定义样式> 浏览器默认样式

### 14、怎么样让一个元素消失，讲讲

display:none; visibility:hidden; opacity: 0; 等等

### 15、css 动画如何实现

创建动画序列，需要使用animation 属性或其子属性，该属性允许配置动画时间、时长  
以及其他动画细节，但该属性不能配置动画的实际表现，动画的实际表现是  
由@keyframes 规则实现，具体情况参见使用keyframes 定义动画序列小节部分。  
transition 也可实现动画。transition 强调过渡，是元素的一个或多个属性发生变化时产生  
的过渡效果，同一个元素通过两个不同的途径获取样式，而第二个途径当某种改变发生  
（例如hover）时才能获取样式，这样就会产生过渡动画。

### 16、如何实现图片在某个容器中居中的？

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

### 17、如何实现元素的垂直居中

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;

### 18、CSS3 中对溢出的处理

text-overflow 属性，值为clip 是修剪文本；ellipsis 为显示省略符号来表被修剪的文本；  
string 为使用给定的字符串来代表被修剪的文本。

### 19、三栏布局的实现方式，尽可能多写，浮动布局时，三个div 的生成顺序有没有影响

三列布局又分为两种，两列定宽一列自适应，以及两侧定宽中间自适应

两列定宽一列自适应：  
1、使用float+margin：  
给div 设置float：left，left 的div 添加属性margin-right：left 和center 的间隔px,right 的  
div 添加属性margin-left：left 和center 的宽度之和加上间隔  
2、使用float+overflow：  
给div 设置float：left，再给right 的div 设置overflow:hidden。这样子两个盒子浮动，另  
一个盒子触发bfc 达到自适应  
3、使用position：  
父级div 设置position：relative，三个子级div 设置position：absolute，这个要计算好盒  
子的宽度和间隔去设置位置，兼容性比较好，  
4、使用table 实现：  
父级div 设置display：table，设置border-spacing：10px//设置间距，取值随意,子级div  
设置display:table-cell，这种方法兼容性好，适用于高度宽度未知的情况，但是margin  
失效，设计间隔比较麻烦，  
5、flex 实现：  
parent 的div 设置display：flex；left 和center 的div 设置margin-right；然后right 的div  
设置flex：1；这样子right 自适应，但是flex 的兼容性不好  
6、grid 实现：  
parent 的div 设置display：grid，设置grid-template-columns 属性，固定第一列第二列宽  
度，第三列auto，  
对于两侧定宽中间自适应的布局，对于这种布局需要把center 放在前面，可以采用双飞  
翼布局：圣杯布局，来实现，也可以使用上述方法中的grid，table，flex，position 实现

### 20、z-index 的定位方法

z-index 属性设置元素的堆叠顺序，拥有更好堆叠顺序的元素会处于较低顺序元素之前，  
z-index 可以为负，且z-index 只能在定位元素上奏效，该属性设置一个定位元素沿z 轴的位置，如果为正数，离用户越近，为负数，离用户越远，它的属性值有auto，默认，堆叠顺序与父元素相等，number，inherit，从父元素继承z-index 属性的值

### 21、box-sizing 的语法和基本用处

box-sizing 规定两个并排的带边框的框，语法为box-sizing：content-box/border-box/inheritcontent-box：宽度和高度分别应用到元素的内容框，在宽度和高度之外绘制元素的内边距和边框  
border-box：为元素设定的宽度和高度决定了元素的边框盒，  
inherit：继承父元素的box-sizing