一阶段面试题集锦

1、 rem em vw vw 百分比区别

- 相同点：
  - rem，em，vw，vh，vw属于前端开发除了px单位之外的另外几种单位取值；但是具体含义存在区别
- 不同点：
  - px：是像素单位，属于绝对单位，是一个具体的大小
  - rem：相对于根元素进行设置
  - em：如果自身有字体大小的设置，那么就相对于自身字体大小设置，如果自身没有字体大小设置，那么就相当于父元素进行设置。
  - vw：1vw相当于浏览器窗口宽度的百分之一
  - vh：1vh相当于浏览器窗口高度的百分之一
  - 百分比：相对于父元素宽度或者高度的百分之几
  - 注意：vw和vh是视口可以观看的区域的大小；如果没有滚动条的话，则宽度设置成100vw和100%的时候实现的效果一直；如果有滚动的话，则100vw中不是包括滚动条的，100%是包括滚动条的距离的
  - ```
    <!--注意里面的代码解释说明-->
    <style>
         *{margin:0;padding:0}
         html,body{
             height: 100%;
         }
         .box1{
             width: 100%;
             height:500px;
             background-color: red;
         }
         .box2{
             width: 100vw;
             height:500px;
             background-color: green;
         }
    </style>

    750的设计图，20px为多少rem
    ```

2、750的设计图，20px为多少rem

首先设计图是UI设计提供给前端开发工程师的设计稿，750的设计图出自于手机屏幕为375px*667px的iphone678手机屏幕的设计稿，这里也可以称之为2倍图。20px的字体大小涉及到了单位的转换，页面中默认的字体大小为16px，那么此时的1rem = 16px;
那么20px通过计算得出公式 1rem = 16px; ?rem = 20px; 得出结果为:1.25rem
    ```

扩展1：
因为他的设备像素比（dpr=物理像素/CSS像素）是2；dpr是一个固定的比值；不同的手机型号比值不应：
物理像素：可以理解成你ps里面测量的距离的大小
CSS像素：你编写开发代码的时候CSS里面给定的像素值；
dpr固定值如下如：

!(pic1.png)

3、app怎么做适配的

- 基于手机系统开发的app(原生Android/IOS)
- 另外一种是webapp
  - 设置以下标签：

    具体的含义为：app完成的页面的宽度等于设备的宽度，页面的缩放比例为1.0，不允许最大缩放；
- 使用媒体查询和响应式做适配
  - 使用媒体查询检测设备屏幕的大小改变布局样式，但是成本耗费比较大，不易操作
- 使用单位自己单位中经常使用的封装好的flexble.js文件做适配
  - 封装好的flexble.js文件文件可以做到适配，并且原理是已经封装好的视口和设备像素比基于webapp开发

4、bfc是什么，清楚浮动的原理

- BFC含义：
  - 块格式化上下文（Block Formatting Context，BFC） 是Web页面的可视CSS渲染的一部分，是块盒子的布局过程发生的区域，也是浮动元素与其他元素交互的区域
- BFC触发条件：
  - 根元素
  - float属性不为none（脱离文档流）
  - position为absolute或fixed
  - display为inline-block,table-cell,table-caption,flex,inine-flex
  - overflow不为visible
- BFC布局规则：
  - 内部的Box会在垂直方向，一个接一个地放置。
  - Box垂直方向的距离由margin决定。属于同一个BFC的两个相邻Box的margin会发生重叠（按照最大margin值设置）
  - 每个元素的margin box的左边， 与包含块border box的左边相接触
  - BFC的区域不会与float box重叠。
  - BFC就是页面上的一个隔离的独立容器，容器里面的子元素不会影响到外面的元素。
  - 计算BFC的高度时，浮动元素也参与计算

5、简单的一个盒子移动到另一个盒子，你用什么方式实现动画效果

- 方法一：使用HTML+CSS里面的transition过渡动画结合2d的位移translate设置

```
<div class="box1"></div>
<div class="box2"></div>
```

```
<style>
    *{margin:0;padding:0}
    div{
        float: left;
    }
    .box1{
        width: 200px;
        height:200px;
        background-color: red;
    }
    .box2{
        width: 100px;
        height:100px;
        background-color: green;
        transition: all linear 1s;
    }
    .box1:hover+.box2{
        transform: translateX(-100px);
    }
</style>
```

- 方法二：使用HTML5+CSS3中的animation动画属性结合2d里面的位移translate进行实现

```
<div class="box1"></div>
<div class="box2"></div>
<style>
     *{margin:0;padding:0}
     div{
         float: left;
     }
     .box1{
         width: 200px;
         height:200px;
         background-color: red;
     }
     .box2{
         width: 100px;
         height:100px;
         background-color: green;
         animation: mover linear 1s;
     }
     @keyframes mover{
         0%{
             transform: translateX(0px);
         }
         100%{
             transform: translateX(-100px);
         }
     }
</style>
```

- 方法三：复杂方法，可以使用js封装一个动画函数，直接使用鼠标移动移入事件或者点击事件触发移动

```
<script>
    //获取元素
    //设置x和y的值
    //绑定鼠标移入事件==缓慢移动   x的位置进行移动   自减
```

```
    //绑定鼠标移出事件==缓慢移动   x的位置进行移动   自增
</script>
<style>
    *{margin:0;padding:0}
    div{float:left}
    .box1{width:300px;height:300px;background-color:red}
    .box2{width:100px;height:100px;background-color:green}
</style>
<div class="box1"></div>
<div class="box2"></div>
```

6、boder实现0.5像素

实现方法：CSS3有缩放的属性，我们可以利用这个属性，缩小50%的1px的边框，来实现这个功能代码如下：

```
<div class="border3">
    <div class="content">伪类设置的边框</div>
</div>
<style>
*{
     margin:0;padding:0
}
.border3{
     position: relative;
}
.border3:before{
    content: '';
    position: absolute;
    width: 200%;
    height: 200%;
    border: 1px solid red;
    transform-origin: 0 0;
    transform: scale(0.5, 0.5);
    box-sizing: border-box;
 }
</style>
```

7、场景题（两个盒子，左边固定宽，右边自适应，你能想到几种方法）

- 公共HTML代码部分

```
<div class="content">
    <div class="left"></div>
    <div class="right"></div>
</div>
```

- 方法一：float来和BFC实现

```
<style>
.content {
    border: 1px solid #000;
    height: 800px;
    padding: 20px;
```

```
}
.left {
    width: 200px;
    height: 100%;
    background: red;
    float: left;
}
.right {
    height: 100%;
    background: pink;
    overflow: hidden;
}
</style>
```

- 方法二：absolute定位和margin值实现

```
<style>
 .content {
     border: 1px solid #000;
     height: 800px;
     padding: 20px;
    }
    .left {
        width: 200px;
        height: 100%;
        background: red;
        position: absolute;
    }
    .right {
        height: 100%;
        background: pink;
        margin-left: 200px;
    }
</style>
```

- 方法三：calc(100% - 固定内容的宽度) 用calc函数动态计算数值

```
<style>
 .content {
     border: 1px solid #000;
     height: 800px;
     padding: 20px;
    }
    .left {
        width: 200px;
        height: 100%;
        background: red;
        float: left;
    }
    .right {
        height: 100%;
        background: pink;
        float: left;
        width: calc(100% - 200px);
    }
</style>
```

- 方法四：flex布局轻松搞定

```
<style>
    .content {
        border: 1px solid #000;
        height: 800px;
        padding: 20px;
        display: flex;
    }
    .left {
        width: 200px;
        height: 100%;
        background: red;
    }
    .right {
        height: 100%;
        background: pink;
        flex: 1;
    }
</style>
```

- 方法五：使用table和table-cell实现

```
 <style>
     .content {
         border: 1px solid #000;
         width: 100%;
         height: 800px;
         display: table;
     }
     .left {
         width: 200px;
         height: 100%;
         background: red;
         display: table-cell;
     }
     .right {
         height: 100%;
         background: pink;
         display: table-cell;
     }
</style>
```

- 方法六：使用inline-block携手calc函数设置宽度

```
<style>
    .content {
        border: 1px solid #000;
        width: 100%;
        height: 800px;
        font-size: 0;
    }
    .left {
        width: 200px;
        height: 100%;
        background: red;
        display: inline-block;
        vertical-align: top;
    }
    .right {
```

```
        height: 100%;
        background: pink;
        display: inline-block;
        vertical-align: top;
        width: calc(100% - 200px);
        font-size: 16px;
    }
</style>
```

8、 css 选择器有哪些，权重是什么样的

```
选择器含义：选择器是查找页面元素的一种方式方法，选择器的种类有很多种。
常用的选择器有以下内容：
1、ID    #id
2、class    .class
3、标签    p
4、通用    *
5、属性    type="text"
6、伪类    :link  :visited  :hover  :active
7、伪元素    ::first-line  :first-letter
8、子选择器  div>p
9、后代选择器  div p
10、相邻兄弟  div+p
11、通用兄弟  div~p
12、结构伪类  :nth-child  :first-child  :last-child

权重计算规则：
1、第一等：代表内联样式，如：style=""，权值为1000。
2、第二等：代表ID选择器，如：#content，权值为0100。
3、第三等：代表类，伪类和属性选择器，如.content，权值为0010。
4、第四等：代表类型选择器和伪元素选择器，如div p，权值为0001。
5、通配符、*权值为0000。
6、继承的样式没有权重值。

important > 内联 > ID > 类| 伪类 | 属性选择|伪对象 > 标签  >  继承 > 通配符
```

9、 css布局，左侧宽度最小150px，最大25 %，右侧自适应。怎么实现

- 方法一：1.float+BFC【BFC区域不会和设置浮动的区域重合】

```
<!--左侧设置最小宽度150px，最大宽度25%，并设置浮动；右侧不设置宽度，加overdlow：hidden触发BFC-->
<style>
    .left{
        min-width: 150px;
        max-width: 25%;
        height: 300px;
        background-color: red;
        float: left;
    }
    .right{
        height: 300px;
        background-color: yellow;
        overflow: hidden;
```

```
    }
</style>
<body>
    <div class="left">
        内容
    </div>
    <div class="right"></div>
</body>
```

- 方法二：弹性盒子

```
<!--给父元素设置弹性盒子，然后给左侧设置最小宽度150px，最大宽度25%;右侧不设置宽度，加属性flex:1,让右侧自适应剩下的宽度-->
<style>
    *{
        padding: 0;
        margin: 0;
    }
    body{
        display: flex;
    }
    .left{
        min-width: 150px;
        max-width: 25%;
        height: 300px;
        background: green;
    }
    .right{
        flex: 1;
        height: 300px;
        background: red;
    }
</style>
<body>
    <div class="left"></div>
    <div class="right"></div>
</body>
```

10、CSS的基本语句构成是?

- CSS被称之为：层叠样式表（Cascading Style Sheets）是对页面结构的一种修饰；
- CSS的基本语法是：选择器{属性:属性值;属性:属性值;属性:属性值}
- CSS基本语法构成两个部分组成：选择器和{}样式规则（样式声明）组成；样式规则（样式声明）由两个部分组成的分别是属性和属性值；
- 使用CSS语法注意事项是
    - 属性和属性值使用（：）链接
    - 属性和属性值结束之后需要使用（；）结束
    - 如果属性和属性值是最后一组的话可以不用分号结束；但是建议添加上为了防止后面继续添加属性

11、css复用

- CSS复用代表的是CSS的重复使用，主要是为了做到网站开发的优化，可以简化多重CSS

- CSS复习：目的是为了创建一套可以不依赖内容的可重复使用的类名及公共的样式,沿着复用这条思路走下去,久而久之基本可以构建一套全新的 UI 组件库而无需编写过多新的 CSS。
- 例如：我们在单位中实际开发的时候经常使用的公共样式表，重置样式表是一样的道理

12、CSS选择符有哪些？哪些属性可以继承？优先级算法如何计算?

- CSS选择符（就是选择器）
  - 基础选择器

```
通配符选择器      *{}
标签选择器标签名  p{}
class类选择器     .class属性值{}
多类名选择器      .类名n{}  标签中class属性为  class="类名1 类名2 ... 类名n"
id选择器          #id属性值{}
群组选择器        选择器1,选择器2,...{}
```

  - 结构选择器

```
子元素选择器E>F{}  F必须是E的子元素
后代选择器E F{}    F必须是E的后代
相邻兄弟选择器E+F{}  F是紧挨这E后面的兄弟元素
通用选择器E~F{}  F是E后面所有的兄弟元素
```

  - 属性选择器

```
[Eattr]           元素E中存在attr属性
[Eattr="value"]   元素E中存在attr属性,并且attr的属性值为value
[Eattr~="value"] 元素E中存在attr属性,并且attr的属性值为value或者"value value1 ..."的形式
[Eattr^="value"] 元素E中存在attr属性,并且attr的属性值以value开始
[Eattr$="value"] 元素E中存在attr属性,并且attr的属性值以value结尾
[Eattr*="value"] 元素E中存在attr属性,并且attr的属性值存在value
[Eattr|="value"] 元素E中存在attr属性,并且attr的属性值为value或者value-的形式
```

  - 结构伪类选择器

```
X:first-child 匹配子集的第一个元素
X:last-child匹配父元素中最后一个X元素
X:nth-child(n)用于匹配索引值为n的子元素。索引值从1开始
X:only-child这个伪类一般用的比较少，比如上述代码匹配的是div下的有且仅有一个的p，也就是说，如果div内有多个p，将不匹配。
X:root匹配文档的根元素。在HTML（标准通用标记语言下的一个应用）中，根元素永远是HTML
X:empty匹配没有任何子元素（包括包含文本）的元素
```

  - 目标伪类

```
E:target 选择匹配E的所有元素，且匹配元素被相关URL指向
```

  - UI状态伪类

```
E:enabled 匹配所有用户界面（form表单）中处于可用状态的E元素
E:disabled 匹配所有用户界面（form表单）中处于不可用状态的E元素
E:checked 匹配所有用户界面（form表单）中处于选中状态的元素E
E:selection 匹配E元素中被用户选中或处于高亮状态的部分
```

  - 否定伪类

```
E:not(s) （IE6-8浏览器不支持:not()选择器。）匹配所有不匹配简单选择符s的元素E
```

- 动态伪类

```
E:link      链接伪类选择器 选择匹配的E元素，而且匹配元素被定义了超链接并未被访问过。常用于链接描点上
E:visited   链接伪类选择器 选择匹配的E元素，而且匹配元素被定义了超链接并已被访问过。常用于链接描点上
E:active    用户行为选择器 选择匹配的E元素，且匹配元素被激活。常用于链接描点和按钮上
E:hover     用户行为选择器  选择匹配的E元素，且用户鼠标停留在元素E上。IE6及以下浏览器仅支持a:hover
E:focus     用户行为选择器  选择匹配的E元素，而且匹配元素获取焦点
```

- 属性继承
  - 继承：html元素可以从父元素那里继承一部分css属性，即使当前元素没有定义该属性。

```
1.字体系列属性 font，font-family，font-weight，font-size，font-style
2.文本系列属性 text-indent，text-align，line-height，word-spacing，letter-spacing，text-transform，color
3.元素可见性 visibility
4.表格布局属性 caption-side，border-collapse，border-spacing，empty-cells，table-layout
5.列表布局属性 list-style-type，list-style-image，list-style-position，list-style
```

- 选择器优先级的算法
  - 优先级就近原则，同权重情况下样式定义最近者为准;
  - 载入样式以最后载入的定位为准;
  - 优先级为：同权重: 内联样式表（标签内部）> 嵌入样式表（当前文件中）> 外部样式表（外部文件中）。
  - !important > id > class > tag　　important 比 内联优先级高

13、css引入的方式有哪些，link和@import的区别是什么

- 区别1：link属于XHTML标签，而@import完全是CSS提供的一种方式。
- 区别2：当一个页面被加载的时候（就是被浏览者浏览的时候），link引用的CSS会同时被加载，而@import引用的CSS会等到页面全部被下载完再被加载。所以有时候浏览@import加载CSS的页面时开始会没有样式（就是闪烁），网速慢的时候还挺明显。
- 区别3：@import是CSS2.1提出的，所以老的浏览器不支持，@import只有在IE5以上的才能识别，而link标签无此问题。
- 区别4：使用dom(document object model文档对象模型)控制样式时的差别：当使用javascript控制dom去改变样式的时候，只能使用link标签，因为@import不是dom可以控制的.

14、css：一个容器（页面），里面有两个div左右摆放并且高度和容器高度一致，左div不会随着页面左右伸缩而变化，右div随页面左右伸缩宽度自适应（手写）

```
<div class="container">
```

```
    <div class="left"></div>
    <div class="right">北京千锋互联科技有限公司（下面简称“千锋教育”），成立于2011年1月，立足于职业教育培训领域，公司现有教育培训、高校服务、企业服务三大业务板块。教育培训业务分为大学生技能培训和职后技能培训；高校服务业务主要提供校企合作全解决方案与定制服务；企...</div>
</div>
<style>
    *{
        margin: 0;
        padding: 0;
    }
    .container{
        max-width: 1000px;
        height: 500px;
        background-color: yellow;
        margin: 20px auto;
        display: flex;
    }
    .container .left{
        width: 300px;
        background-color: orange;
    }
    .container .right{
        flex: 1;
        background-color: red;
    }
</style>
```

15、CSS3动画知道吗，怎么实现的

- 知道的，CSS3动画主要是通过animation这个属性来进行实现，使用动画的时候需要先进行声明动画然后再去调用，哪里需要产生动画效果哪里就使用animation进行调用
- 基本语法

```
<style>
    *{margin:0;padding:0}
    div{
        width:200px;
        height:200px;
        background-color:red;
        /*调用动画*/
        animation:movers 20s linear infinite alternate
    }

    /*声明动画*/
    @keyframes movers{
        form{
            width:200px;
            height:200px;
            background-color:red;
        }
        to{
            width:400px;
            height:400px;
            background-color:green;
            border-radius:50%;
        }
```

```
    }
</style>
<div></div>
```

- 语法注意事项

```
animation:复合属性
animation-name  规定需要绑定到选择器的 keyframe 名称。。
animation-duration  规定完成动画所花费的时间，以秒或毫秒计。
animation-timing-function   规定动画的速度曲线。
animation-delay 规定在动画开始之前的延迟。
animation-iteration-count   规定动画应该播放的次数。
animation-direction 规定是否应该轮流反向播放动画。

form  初始状态
to    结束装填
可以替换成
0%    初始状态
100%  结束状态
后面的百分比可以称之为关键帧动画
```

16、CSS3新增了哪些东西

- CSS3里面的新增主要有：CSS选择器，盒子中的修饰，背景，文本效果，字体，2D/3D，过渡动画，多列布局等等
- CSS选择器

```
1、p:first-of-type   选择属于其父元素的首个 <p> 元素的每个 <p> 元素。
2、p:last-of-type   选择属于其父元素的最后 <p> 元素的每个 <p> 元素。
3、p:only-of-type   选择属于其父元素唯一的 <p> 元素的每个 <p> 元素。
4、p:only-child   选择属于其父元素的唯一子元素的每个 <p> 元素。
5、p:nth-child(2)   选择属于其父元素的第二个子元素的每个 <p> 元素。
```

- 盒子修饰

```
新增了边框属性：
1、border-radius
    支持浏览器：IE9+、Firefox、Chrome、Safari 、Opera
2、box-shadow  向方框添加一个或多个阴影
    支持浏览器：IE9+、Firefox、Chrome、Safari 、Opera
3、border-image
    支持浏览器：Firefox（旧版本需要前缀-moz-）、Chrome（旧版本需要前缀-webkit-）、
Safari（Safari 5 以及更老的版本需要前缀 -webkit-） 、Opera（ 需要前缀 -o-）
```

- CSS新增背景模块

```
1、background-size   规定背景图片的尺寸
    支持浏览器：IE9+、Firefox（旧版本需要前缀-moz-）、Chrome、Safari、Opera
2、background-origin    规定背景图片的定位区域，背景图片可以放置于 content-box、padding-
box 或 border-box 区域。
    支持浏览器：IE9+、Firefox、Chrome、Safari、Opera
3、background-clip   规定背景的绘制区域
    支持浏览器：IE9+、Firefox、Chrome、Safari、Opera
```

- CSS文本效果模块

```
1、text-shadow    可向文本应用阴影
    支持浏览器：IE10、Firefox、Chrome、Safari、Opera
2、word-wrap    允许文本强制文本进行换行 - 即使这意味着会对单词进行拆分
    支持浏览器：所有主流浏览器
```

- CSS引入字体模块

```
@font-face
```

- 2D/3D

```
1、transform    向元素应用 2D 或 3D 转换
    支持浏览器：
        IE10：2D、3D都支持（2D IE9 需要前缀 -ms-）；
        Firefox：2D、3D都支持；
        Chrome：2D、3D都支持（2D、3D需要前缀 -webkit-）；
        Safari：2D、3D都支持（2D、3D需要前缀 -webkit-）；
        Opera：只支持2D
```

- 过渡动画

```
1、transition    过渡属性
    支持浏览器：IE 10、Firefox、Chrome（Chrome 25 以及更早的版本，需要前缀 -webkit-）、
Opera、Safari（ 需要前缀 -webkit-）
2、@keyframes    用于创建动画。
    在 @keyframes 中规定某项 CSS 样式，就能创建由当前样式逐渐改为新样式的动画效果。
    支持浏览器：IE 10、Firefox、Chrome（Chrome 25 以及更早的版本，需要前缀 -webkit-）、
Opera、Safari（ 需要前缀 -webkit-）
3、animation    动画调用属性
    支持浏览器：IE 10、Firefox、Chrome（Chrome 25 以及更早的版本，需要前缀 -webkit-）、
Opera、Safari（ 需要前缀 -webkit-）
```

- 多列布局

```
1、column-count=======划分列数
2、column-gap=========属性规定列之间的间隔大小
3、column-rule========设置或检索对象的列与列之间的边框
4、column-fill========设置或检索对象所有列的高度是否统一
5、column-span========设置或检索对象元素是否横跨所有列。
6、column-width=======设置或检索对象每列的宽度
```

17、当margin-top、padding-top 的值是百分比时,分别是如何计算的?

- 可以对元素的margin设置百分数，百分数是相对于父元素的width计算，不管是margin-top/margin-bottom还是margin-left/margin-right。（padding同理）
- 如果没有为元素声明width，在这种情况下，元素框的总宽度包括外边距取决于父元素的width，这样可能得到“流式”页面，即元素的外边距会扩大或缩小以适应父元素的实际大小。如果对这个文档设置样式，使其元素使用百分数外边距，当用户修改浏览窗口的宽度时，外边距会随之扩大或缩小。
- 为什么margin-top/margin-bottom的百分数也是相对于width而不是height呢?

- 我们认为，正常流中的大多数元素都会足够高于包含其后代元素（包括外边距），如果一个元素的上下外边距是父元素的height的百分数，就可能导致一个无限循环，父元素的height会增加，以适应后代元素上下外边距的增加，而相应的，上下外边距因为父元素height的增加也会增加，如此循环。

```
<style>
    .fu {
        width: 400px;
        height: 300px;
        background: blue;
        overflow: hidden;
    }
    .zi {
        width: 20px;
        height: 20px;
        background: red;
        margin-top: 50%;
    }
</style>
<div class="fu">
    <div class="zi"></div>
</div>
```

18、定位知道吗，说下，几种不同的定位分别有什么特点，详细说明

- 定位的含义是将元素放在指定的位置上，在css中特指position属性，他一共有5种属性值。
- 分别是
    - static静态定位，是元素自带的默认的定位方式。
    - relative是相对定位，他是基于元素本身的位置进行定位的，不会脱离文档流。
    - fixed是固定定位是基于浏览器窗口进行定位的，会脱离文档流。
    - absolute是绝对定位，是基于最近的被设置了非静态定位的上级元素进行定位的，他会脱离文档流，常用的场景是子绝父相。
    - sticky是粘性定位，是css新增的属性值；可以说是相对定位relative和固定定位fixed的结合；它主要用在对scroll事件的监听上，简单说在滑动过程中，某个元素距离其父元素的距离达到 sticky 粘性定位 要求时；position:sticky 这时的效果就相对于 fixed 定位，固定到适当的位置。
- 使用定位的时候如果需要元素进行位置的调整需要配合偏移属性进行实现对应的效果。偏移属性有四个：top；right；bottom；left；定位的属性决定了定位的偏移参照物
- 参照物问题
    - 静态定位不会发生位置的调整所以不存在参照物的问题
    - 相对定位添加偏移属性后，相对于自己原来的位置进行位置调整
    - 绝对定位：如果父元素及外侧没有任何已经定位的元素，则参照浏览器屏幕左上角（body左上角）进行位置的调整；如果父元素或者是就近的父级元素有定位则相对于就进行元素的左上角进行位置的调整；绝对定位的参照物就是所谓的包含块的意思
    - 固定定位：参照物是浏览器可视窗口位置的左上角进行位置调整，不会受到滚动条的滚动而影响
    - 粘性定位：粘性定位参照物在实现固定吸顶效果的时候参照物与固定定位一样

19、display有哪几种属性值？分别代表什么？Img属于什么元素？

- **display属性的含义：display控制元素的显示类型；**
- **display属性的属性值可以分成两大类**
  - **常用的属性值**

```
none代表为不显示：控制元素的隐藏
block代表显示为块级元素：还可以让元素控制元素显示
    块级元素的特点：默认站宽一整行，能设置宽度高度，纵向排列
    块级元素有:div,p,h1,h6,ol,ul,ol,li,dl,dt,dd,form,fieldset,legend,
    table,header,footer,section,main,nav,article,aside等等
inline代表行内元素：
    行内元素的特点：不能设置宽度高度，并且能横向显示
    行内元素有:a,b,strong,u,i,em,s,del,sup,sub,span,font,mark,var
inline-block代表行内块元素
    行内块元素的特点：能设置宽度高度并且横向显示
    行内块元素有:input,textarea,select,
flex代表的是触发弹性盒子
grid代表的是触发网格布局
```

  - **不常用的属性值**

list-item	此元素会作为列表显示。
run-in	此元素会根据上下文作为块级元素或内联元素显示。
compact	CSS 中有值 compact，不过由于缺乏广泛支持，已经从 CSS2.1 中删除。
marker	CSS 中有值 marker，不过由于缺乏广泛支持，已经从 CSS2.1 中删除。
table	此元素会作为块级表格来显示（类似 <table>），表格前后带有换行符。
inline-table	此元素会作为内联表格来显示（类似 <table>），表格前后没有换行符。
table-row-group	此元素会作为一个或多个行的分组来显示（类似 <tbody>）。
table-header-group	此元素会作为一个或多个行的分组来显示（类似 <thead>）。
table-footer-group	此元素会作为一个或多个行的分组来显示（类似 <tfoot>）。
table-row	此元素会作为一个表格行显示（类似 <tr>）。
table-column-group	此元素会作为一个或多个列的分组来显示（类似 <colgroup>）。
table-column	此元素会作为一个单元格列显示（类似 <col>）
table-cell	此元素会作为一个表格单元格显示（类似 <td> 和 <th>）
table-caption	此元素会作为一个表格标题显示（类似 <caption>）
inherit	规定应该从父元素继承 display 属性的值。

- **img属于什么元素**：浏览器中的computed计算属性中的图片的display的取值为inline，虽然为inline但是实际开发的时候图片是作为行内块元素进行使用的，因为遵循能设置宽度高度，并且还能横向显示；所以属于行内块元素，但是有些程序员也会把图片认为是行内元素也是可以，从另一个角度讲img也称作是置换元素；

20、display：inline-block后为什么有间距

- display:inline-block是让元素在一行显示，但是这些元素在html里面是上下行排列的，所以中间有换行符，于是并排显示就有了换行符带来的空隙。
- 解决这种问题的方式有：
  - 将html标签要display:inline-block 的元素写在一行。缺点：代码可读性差。

- 给父元素设置font-size:0,给子元素设置需要的font-size值。缺点:是子元素如果里面有文字，文字会消失不见，所以又要给子元素设置font-size,增加了代码量。
- 给元素设置float:left,缺点需要清除浮动。
- 设置子元素的margin-left为负值，但是元素之间的间隙大小是根据上下文的字体大小确定的，而每个浏览器的换行空隙大小不同，所以这个方法不通用。
- 设置父元素 display:table;word-spacing:-1em;目前这个方法可以完美解决，且兼容其他浏览器。

21、Doctype的作用？严格模式和混杂模式的区分，以及如何触发这2种模式？

- **Doctype的作用**
  - 1.<！DOCTYPE>声明叫做文档类型DTD，它的作用就是用来标识浏览器使用哪种文档类型，让浏览器知道以何种方式解析文档。
  - 2.必须位于HTML文档的第一行，处于标签之前，但是不属于HTML文档标签。
  - 3.声明文档的解析类型（document.compatMode），是为了避免浏览器的怪异模式。
- **严格模式和混杂模式的区别，以及如何触发两种模式**
  - **1.严格模式：**浏览器按照W3C的标准解析代码，又称为标准模式。
  - **2.混杂模式：**浏览器按照自己的方式来解析代码，以一种向后兼容的方式呈现。
  - **3.Doctype可声明的三种DTD类型：**严格版本，过渡版本，基于框架的HTML版本。
  - **4.区别：**浏览器使用严格模式和混杂模式，与文档中的DTD直接相关
    - （1）如果用文档中包含严格的DOCTYPE，则以严格模式呈现（严格DTD--严格模式）
    - （2）包含过渡DTD和URL的DOCTYPE，以严格模式呈现；包含过渡的DTD而没有URL，以混杂模式呈现；（过渡DTD+URL--严格，过渡DTD（无URL）-- 混杂模式））
    - （3）HTML5中没有DTD，没有严格和混杂模式的区分，HTML5中有相对宽松的语法，尽可能实现向后兼容
    - （4）DOCTYPE不存在或者格式不正确，以混杂模式呈现（DTD不存在或格式不正确--混杂模式）
  - **5.严格模式和混杂模式解析语句的不同点**
    - （1）可以设置行内元素的宽高，在严格模式下给内联元素设置宽高都不起作用，在混杂模式下生效
    - （2）可设置百分比高度在严格模式下，如果没有给父元素设置高度，而子元素的高度以百分比呈现，这时是不生效的
    - （3）盒模型的宽高包含padding和border在W3C的标准下，给一个元素设置宽高，则呈现的是内容的宽高。在IE5.5以下及其他浏览器的混杂模式下，盒子的宽度还包括padding和border。
    - （4）使用margin：0 auto在IE下会失效使用margin：0 auto在严格模式下会水平居中，而在混杂模式下会失效，但可以设置text-align：center来水平居中。
    - （5）混杂模式下的图片padding会失效，Table中的字体属性将无法继承父元素的设置，white-space：pre会失效。
  -
    - （1）如果用文档中包含严格的DOCTYPE，则以严格模式呈现（严格DTD--严格模式）
    - （2）包含过渡DTD和URL的DOCTYPE，以严格模式呈现；包含过渡的DTD而没有URL，以混杂模式呈现；（过渡DTD+URL--严格，过渡DTD（无URL）-- 混杂模式））
    - （3）HTML5中没有DTD，没有严格和混杂模式的区分，HTML5中有相对宽松的语法，尽可能实现向后兼容

        - （4）DOCTYPE不存在或者格式不正确，以混杂模式呈现（DTD不存在或格式不正确--混杂模式）

22、对WEB标准以及w3c的理解与认识?

- **web标准可以分为结构、表现和行为。**
    - 结构主要是有HTML标签组成。或许通俗点说，在页面body里面我们写入的标签都是为了页面的结构。
    - 表现即指css样式表，通过css可以是页面的结构标签更具美感。
    - 行为是指页面和用户具有一定的交互，同时页面结构或者表现发生变化，主要是有js组成。
- web标准一般是将该三部分独立分开，使其更具有模块化。但一般产生行为时，就会有结构或者表现的变化，也使这三者的界限并不那么清晰。W3C对web标准提出了规范化的要求，也就是在实际编程中的一些代码规范：**包含如下几点**
    - 1.对于结构要求：（标签规范可以提高搜索引擎对页面的抓取效率，对SEO很有帮助
        - 1）标签字母要小写
        - 2）标签要闭合
        - 3）标签不允许随意嵌套
    - 2.对于css和js来说
        - 1）尽量使用外链css样式表和js脚本。是结构、表现和行为分为三块，符合规范。同时提高页面渲染速度，提高用户的体验。
        - 2）样式尽量少用行间样式表，使结构与表现分离，标签的id和class等属性命名要做到见文知义，标签越少，加载越快，用户体验提高，代码维护简单，便于改版
        - 3）不需要变动页面内容，便可提供打印版本而不需要复制内容，提高网站易用性。

23、FireFox中标签的居中问题的解决办法

```
<div id="a" style="width:200px;border:1px solid red;text-align:center;">
    <div id="b" style="background-color:blue;width:30px;margin:0 auto"> </div>
</div>
<!--以上面结构为例，在a中设置text-align属性为center之后，还需要设置b的横向margin为auto。设置添加b的CSS样式为：margin: 0 auto;-->
```

24、Flex的横向平铺填充

让父元素触发弹性和也就是display:flex;然后让所有的子元素均分使用flex:1;代码如下

```
<div style="width: 1000px;height: 500px;display: flex;">
    <span style="flex:1;">1</span>
    <span style="flex:1;">2</span>
    <span style="flex:1;">3</span>
    <span style="flex:1;">4</span>
    <span style="flex:1;">5</span>
</div>
```

25、flex用法场景

- 场景一：移动端和PC端布局中；能够高效解决间距调整的问题（移动端布局和PC端布局）

- 场景二：能快速高效实现元素水平垂直居中（点击删除的弹窗）
- 场景三：能够实现快速元素均分，避免了百分比设置的不确定性
- 场景四：快速实现多列布局，能高效实现瀑布流布局
- 场景五：后台管理系统的两栏和三栏布局

26、flex:0 1 100px什么意思？flex两个参数,三个参数什么意思？表示三个参数，flex-grow、flex-shrink、flex-basis分别是啥意思

- **flex属性是弹性布局添加给项目（子元素）的属性，**
  - **flex**属性是flex-grow（放大）、flex-shrink（缩小）、flex-basis（宽度）的简写属性。
  - **flex-grow**默认值为0，当值大于0时，当父元素有剩余空间时当前元素放大，父元素没有剩余空间时，该元素不放大。
  - **flex-shrnk**默认值为1，父元素有剩余空间时，该元素不缩小，父元素没有剩余空间时，该元素缩小。
  - **flex-basis**相当于width属性。
- **flex:0 1 100px；**表示父元素有剩余空间当前元素不放大，父元没有剩余空间当前元素会缩小。flex有很多中赋值方式，可以写一个值，两个值或者三个值。
  - **举例：**1.flex：1 解析为 flex：1 1 0%。2.flex：auto解析为 flex：1 1 auto。3.flex：none解析为flex：0 0auto。4.flex：解析为flex：0 0 auto。4.flex：0 auto解析为flex：0 1 auto。

27、父元素visibility：hidden；子元素设置visibility：visible，子元素显示吗

- 这时候子元素是显示的，原理如下：
- 当只是给父元素添加visibility：hidden，子元素会继承父元素的visibility：hidden的值，也会跟着父元素隐藏。
- 如果给父元素添加visibility：hidden；同时在给子元素添加visibility：visible，子元素是显示的。**原因：**添加给子元素visibility：visible是大于父元素添加的visibility：hidden的。所以优先执行子元素的visibility：visible属性。

28、知道渐变嘛，说下你的了解

- **CSS3 渐变**（gradients）可以让你在两个或多个指定的颜色之间显示平稳的过渡。
- 以前，你必须使用图像来实现这些效果。但，通过使用 CSS3 渐变（gradients），你可以减少下载的时间和宽带的使用。此外，渐变效果的元素在放大时看起来效果更好，**因为渐变（gradient）是由浏览器生成的。**
- CSS3 定义了两种类型的渐变（gradients）：使用都是background属性
  - **一、线性渐变**（Linear Gradients）- 向下/向上/向左/向右/对角方向
    - 基本语法:background-image: linear-gradient(direction, color-stop1, color-stop2, ...);
  - **二、径向渐变**（Radial Gradients）- 由它们的中心定义
    - 基本语法:background-image: radial-gradient(shape size at position, start-color, ..., last-color);
    - 为了创建一个径向渐变，你也必须至少定义两种颜色节点。
    - 颜色节点即你想要呈现平稳过渡的颜色。

        - 同时，你也可以指定渐变的中心、形状（圆形或椭圆形）、大小。默认情况下，渐变的中心是 center（表示在中心点），渐变的形状是 ellipse（表示椭圆形），渐变的大小是 farthest-corner（表示到最远的角落）
        - size的取值 closest-side，farthest-side，closest-corner，farthest-corner
    - **三、重复渐变**
        - repeating-linear-gradient() 函数用于重复线性渐变：

```
#grad {
  /* 标准的语法 */
  background-image: repeating-linear-gradient(red, yellow 10%, green
20%);
}
```

        - repeating-radial-gradient() 函数用于重复径向渐变：

```
#grad {
  background-image: repeating-radial-gradient(red, yellow 10%, green
15%);
}
```

29、一条0.5px的线，几种方法

- 方法一：采用meta viewport的方式

```
<meta name="viewport" content="width=device-width, initial-scale=0.5, minimum-
scale=0.5, maximum-scale=0.5"/>
<!--
这样子就能缩放到原来的0.5倍，如果是1px那么就会变成0.5px
要记得viewport只针对于移动端，只在移动端上才能看到效果
-->
```

- **方法二：采用transform: scale()的方式**
    - transform: scale(0.5,0.5);
- **方法三：直接利用边框**
    - border: 0.5px solid red;//ios8以上支持，以下显示为0
- **方法四：渐变模拟：设置 1px 通过 css 实现的背景图片，50%有颜色，50%透明。**
-

```
.border {
    background-image:linear-gradient(180deg, red, red 50%, transparent 50%),
linear-gradient(270deg, red, red 50%, transparent 50%), linear-gradient(0deg, red,
red 50%, transparent 50%), linear-gradient(90deg, red, red 50%, transparent 50%);
    background-size: 100% 1px, 1px 100%, 100% 1px, 1px 100%;
    background-repeat: no-repeat;
    background-position: top, right top, bottom, left top;
    padding: 10px;
}
/*
优点：兼容性较好，单边框、多边框可实现，大小、颜色可配置。
缺点：代码量多、无法实现圆角、同时占用了背景样式
*/
```

- **方法五：利用阴影**

```
-webkit-box-shadow: 0 1px 1px -1px rgba(0, 0, 0, 0.5);
/*
利用 css 对阴影处理的方式模拟。
优点：兼容性较好，单边框、多边框、圆角可实现，大小、颜色、可配置。
缺点：模拟效果强差人意，颜色不好配置。
*/
```

- **方法六：边框图片:**

```
border-image: url() 2 0 stretch; border-width: 0 0 1px;
/*缺点：图片边缘模糊，大小、颜色更改不灵活。*/
```

30、行内元素有那些。块级元素有那些。空元素有哪些

- 行内元素：不能实现宽度高度，横向显示
  - a,span,i,em,var,b,strong,sup,sub,s,del,label,font,mark
- 块级元素：能设置宽度高度，纵向显示，并且默认站宽一整行
  - h1~h6,p,div,ul,ol,li,dl,dt,dd,table,form,hr,fieldset,legend,marquee,iframe,header,nav,footer,section,main,figure,figcaption,hgroup,aside,article
- 行内块元素：能设置宽度高度，并且横向显示
  - input,img,select,textarea
- 空元素：就是单标签
  - input,img,br,meta,link,hr
- 置换元素：因为默认自带宽度告诉浏览器能进行根据类型路径加载对应的内容
  - input,img,select,textarea

31、Html和xhtml有什么区别?

- **含义不同：**HTML是一种基本的WEB网页设计语言，XHTML是一个基于XMl的置标语言
- **最主要的不同**
  - XHTML元素必须被正确地嵌套。
  - XHTML元素必须被关闭
  - XHTML标签名必须用小写字母
  - XHTMl文档必须拥有根元素在XHTML中，、、都是必需的标签。
  - XHTML必须设置标签的xmlns属性，且其值为“ http://www.w3.org/1999/xhtml ”。
  - XHTML任何属性值要么用单引号引起来，要么用双引号引起来。如class=page就是不合法的，而

    class='page'和class="page"均是合法的。所有属性必须有值。有些属性，比如 <input type="checkbox" checked /> 标签的checked属性，在HTML中可以使用简写形式，即 <input type="checkbox" checked /> 数据，然后在XHTML中，必须这样编写： <input type="checkbox" checked="checked" /> 数据

32、html页面渲染方式和流程

- **用户输入url地址，浏览器根据域名寻找IP地址**

- 浏览器向服务器发送http请求，如果服务器段返回以301之类的重定向，浏览器根据相应头中的location再次发送请求
- 服务器端接受请求，处理请求生成html代码，返回给浏览器，这时的html页面代码可能是经过压缩的
  浏览器接收服务器响应结果，如果有压缩则首先进行解压处理，紧接着就是页面解析渲染

- **解析渲染该过程主要分为以下步骤**：
  - 解析HTML----构建DOM树----DOM树与CSS样式进行附着构造呈现树------布局、绘制
- **详细过程如下**：
  - 1)用户输入网址（假设是个html页面，并且是第一次访问），浏览器向服务器发出请求，服务器返回html文件。
  - 2)浏览器开始载入html代码，发现标签内有一个标签引用外部CSS文件。
  - 3)浏览器又发出CSS文件的请求，服务器返回这个CSS文件。
  - 4)浏览器继续载入html中部分的代码，并且CSS文件已经拿到手了，可以开始渲染页面了。
  - 5)浏览器在代码中发现一个标签引用了一张图片，向服务器发出请求。此时浏览器不会等到图片下载完，而是继续渲染后面的代码。
  - 6)服务器返回图片文件，由于图片占用了一定面积，影响了后面段落的排布，因此浏览器需要回过头来重新渲染这部分代码。
  - 7)浏览器发现了一个包含一行Javascript代码的标签，赶快运行它。
  - 8)Javascript脚本执行了这条语句，它命令浏览器隐藏掉代码中的某个