**HTML5 + CSS3面试题集锦**

# HTML5面试题

## 1.1 H5的新特性有哪些

1. 画布(Canvas) API
2. 地理(Geolocation) API
3. 音频、视频API(audio,video)
4. localStorage和sessionStorage
5. webworker和 websocket
6. header,nav,footer,aside,article,section
7. web worker是运行在浏览器后台的js程序，他不影响主程序的运行，是另开的一个js线程，可以用这个线程执行复杂的数据操作，然后把操作结果通过postMessage传递给主线程，这样在进行复杂且耗时的操作时就不会阻塞主线程了

## 1.2 Label的作用是什么？是怎么用的？

label标签来定义表单控制间的关系,当用户选择该标签时，浏览器会自动将焦点转到和标签相关的表单控件上。

<label for="Name">Number:</label>

<input type=“text“name="Name" id="Name"/>

<label>Date:<input type="text" name="B"/></label>

## 1.3 HTML5的form如何关闭自动完成功能

给不想要提示的 form 或某个 input 设置为 autocomplete=off

## 1.4 dom如何实现浏览器内多个标签页之间的通信? (阿里)

1. WebSocket、SharedWorker；
2. 也可以调用localstorge、cookies等本地存储方式；localstorge另一 个浏览上下文里被添加、修改或删除时，它都会触发一个事件，我们通 过监听事件，控制它的值来进行页面信息通信；
3. 注意quirks：Safari 在无痕模式下设置localstorge值时会抛出 QuotaExceededError 的异常；

## 1.5 实现不使用 border 画出1px高的线，在不同浏览器的标准模式与怪异模式下都能保持一致的效果

<div style="height:1px;overflow:hidden;background:red"></div>

## 1.6 title与h1的区别、b与strong的区别、i与em的区别？

title属性没有明确意义只表示是个标题，H1则表示层次明确的标题，对页面信息的抓取也有很大的影响；

strong是标明重点内容，有语气加强的含义，使用阅读设备阅读网络时： <strong>会重读，而<B>是展示强调内容。

i内容展示为斜体，em表示强调的文本；

Physical Style Elements -- 自然样式标签

b, i, u, s, pre

Semantic Style Elements -- 语义样式标签

strong, em, ins, del, code

应该准确使用语义样式标签, 但不能滥用, 如果不能确定时首选使用自然样式标签。

## 1.7 你做的页面在哪些流览器测试过？这些浏览器的内核分别是什么?

Trident内核:IE系列

Gecko内核:Firefox

Webkit内核:Safari

Blink内核：是基于Webkit内核的子项目,使用的浏览器有：

Chrome/opera等除IE、Firefox、Safari之外的几乎所有浏览器

几乎所有国产双内核浏览器（Trident/Blink）如360、猎豹、qq、百度等

## 1.8 每个HTML文件里开头都有个很重要的东西，Doctype，知道这是干什么的吗？

文档声明。

<!DOCTYPE> 声明位于文档中的最前面的位置，处于 <html> 标签之前。此标签可告知浏览器文档使用哪种 HTML 或 XHTML 规范。（重点：告诉浏览器按照何种规范解析页面）

IE下如不书写文档声明会使用怪异模式解析网页导致一系列CSS兼容性问题

## 1.9 div+css的布局较table布局有什么优点

正常场景一般都适用div+CSS布局，

优点：

1. 结构与样式分离

2. 代码语义性好

3. 更符合HTML标准规范

4. SEO友好

Table布局的适用场景：

某种原因不方便加载外部CSS的场景，例如邮件正文，此时用table布局可以在无css情况下保持页面布局正常

## 1.10 img的alt与title有何异同？ strong与em的异同

alt(alt text):为不能显示图像、窗体或applets的用户代理（UA），alt属性用来指定替换文字。替换文字的语言由lang属性指定。(在IE浏览器下会在没有title时把alt当成 tool tip显示)

title(tool tip):该属性为设置该属性的元素提供建议性的信息。

em:表现为斜体，语义表示强调

strong:表现为粗体，语义为强烈语气，强调程度超过em

## 1.11 简述一下src与href的区别

src用于替换当前元素，href用于在当前文档和引用资源之间确立联系。

src是source的缩写，指向外部资源的位置，指向的内容将会嵌入到文档中当前标签所在位置；在请求src资源时会将其指向的资源下载并应用到文档内，例如js脚本，img图片和frame等元素。

<script src =”js.js”></script>

当浏览器解析到该元素时，会暂停其他资源的下载和处理，直到将该资源加载、编译、执行完毕，图片和框架等元素也如此，类似于将所指向资源嵌入当前标签内。这也是为什么将js脚本放在底部而不是头部。

href是Hypertext Reference的缩写，指向网络资源所在位置，建立和当前元素（锚点）或当前文档（链接）之间的链接，如果我们在文档中添加

<link href=”common.css” rel=”stylesheet”/>

那么浏览器会识别该文档为css文件，就会并行下载资源并且不会停止对当前文档的处理。这也是为什么建议使用link方式来加载css，而不是使用@import方式。

## 1.12 知道的网页制作会用到的图片格式有哪些

png-8，png-24，jpeg，gif，svg。

但是上面的那些都不是面试官想要的最后答案。面试官希望听到是Webp。（是否有关注新技术，新鲜事物）

科普一下Webp：WebP格式，谷歌（google）开发的一种旨在加快图片加载速度的图片格式。图片压缩体积大约只有JPEG的2/3，并能节省大量的服务器带宽资源和数据空间。Facebook Ebay等知名网站已经开始测试并使用WebP格式。

在质量相同的情况下，WebP格式图像的体积要比JPEG格式图像小40%

## 1.13 在css/js代码上线之后开发人员经常会优化性能，从用户刷新网页开始，一次js请求一般情况下有哪些地方会有缓存处理

dns缓存，cdn缓存，浏览器缓存，服务器缓存

**1.14 一个页面上有大量的图片（大型电商网站），加载很慢，你有哪些方法优化这些图片的加载，给用户更好的体验**

* 图片懒加载，在页面上的未可视区域可以添加一个滚动条事件，判断图片位置与浏览器顶端的距离与页面的距离，如果前者小于后者，优先加载。
* 如果为幻灯片、相册等，可以使用图片预加载技术，将当前展示图片的前一张和后一张优先下载。
* 如果图片为css图片，可以使用CSSsprite，SVGsprite，Iconfont、Base64等技术。
* 如果图片过大，可以使用特殊编码的图片，加载时会先加载一张压缩的特别厉害的缩略图，以提高用户体验。
* 如果图片展示区域小于图片的真实大小，则因在服务器端根据业务需要先行进行图片压缩，图片压缩后大小与展示一致

**1.15你如何理解HTML结构的语义化**

1. 更符合W3C统一的规范标准，是技术趋势。
2. 没有样式时浏览器的默认样式也能让页面结构很清晰。
3. 对功能障碍用户友好。屏幕阅读器（如果访客有视障）会完全根据你的标记来“读”你的网页。
4. 对其他非主流终端设备友好。例如机顶盒、PDA、各种移动终端。
5. 对SEO友好。

**1.16谈谈以前端角度出发做好SEO需要考虑什么**

搜索引擎主要以:

外链数量和质量,网页内容和结构等来决定某关键字下的网页搜索排名。

前端应该注意网页结构和内容方面的情况：

* Meta标签优化：主要包括主题（Title)，网站描述(Description)。还有一些其它的隐藏文字比如Author（作者），Category（目录），Language（编码语种）等，符合W3C规范的语义性标签的使用
* 如何选取关键词并在网页中放置关键词：搜索就得用关键词。关键词分析和选择是SEO最重要的工作之一。首先要给网站确定主关键词（一般在5个上下），然后针对这些关键词进行优化，包括关键词密度（Density），相关度（Relavancy），突出性（Prominency）等等。

**1.17html5有哪些新特性、移除了那些元素**

新特性：

1. 拖拽释放(Drag and drop) API

2. 语义化更好的内容标签（header,nav,footer,aside,article,section）

3. 音频、视频API(audio,video)

4. 画布(Canvas) API

5. 地理(Geolocation) API

6. 本地离线存储 localStorage 长期存储数据，浏览器关闭后数据不丢失；

7. sessionStorage 的数据在浏览器关闭后自动删除

8. 表单控件，calendar、date、time、email、url、search

9. 新的技术webworker, websocket, Geolocation

移除的元素：

1. 纯表现的元素：basefont，big，center，font, s，strike，tt，u；

2. 对可用性产生负面影响的元素：frame，frameset，noframes；

**1.18如何处理HTML5新标签的浏览器兼容问题**

1. IE8/IE7/IE6支持通过 document.createElement 方法产生的标签，可以利用这一特性让这些浏览器支持 HTML5 新标签，浏览器支持新标签后，还需要添加标签默认的样式（当然最好的方式是直接使用成熟的框架、使用最多的是html5shim框架）：

<!--[if lt IE 9]>

<script> src="http://html5shim.googlecode.com/svn/trunk/html5.js"</script>

<![endif]-->

**1.19如何区分 HTML 和 HTML5？**

DOCTYPE声明新增的结构元素、功能元素

**1.20 HTML5 Canvas 元素有什么用**

Canvas 元素用于在网页上绘制图形，该元素标签强大之处在于可以直接在 HTML 上进行图形操作

**1.21 如何在 HTML5 页面中嵌入音频**

HTML 5 包含嵌入音频文件的标准方式，支持的格式包括 MP3、Wav 和 Ogg：

<audio controls>

<source src="jamshed.mp3" type="audio/mpeg">

Your browser does'nt support audio embedding feature.

</audio>

**1.22 如何在HTML5页面中嵌入视频**

和音频一样，HTML5 定义了嵌入视频的标准方法，支持的格式包括：MP4、WebM 和 Ogg：

<video width="450" height="340" controls>

<source src="jamshed.mp4" type="video/mp4">

Your browser does'nt support video embedding feature.

</video>

**1.23 HTML5 引入什么新的表单属性**

datalist datetime output keygen date month week time number range emailurl

**1.24语义化的理解**

用正确的标签做正确的事情！

html语义化就是让页面的内容结构化，便于对浏览器、搜索引擎解析；

在没有样式CSS情况下也以一种文档格式显示，并且是容易阅读的。

搜索引擎的爬虫依赖于标记来确定上下文和各个关键字的权重，利于 SEO。

使阅读源代码的人对网站更容易将网站分块，便于阅读维护理解

**1.25介绍一下你对浏览器内核的理解**

主要分成两部分：渲染引擎(layout engineer或Rendering Engine)和JS引擎

渲染引擎：负责取得网页的内容（HTML、XML、图像等等）、整理讯息（例如加入CSS等），以及计算网页的显示方式，然后会输出至显示器或打印机。浏览器的内核的不同对于网页的语法解释会有不同，所以渲染的效果也不相同。所有网页浏览器、电子邮件客户端以及其它需要编辑、显示网络内容的应用程序都需要内核

JS引擎则：解析和执行javascript来实现网页的动态效果

最开始渲染引擎和JS引擎并没有区分的很明确，后来JS引擎越来越独立，内核就倾向于只指渲染引擎

**1.26浏览器是怎么对HTML5的离线储存资源进行管理和加载的呢**

在线的情况下，浏览器发现html头部有manifest属性，它会请求manifest文件，如果是第一次访问app，那么浏览器就会根据manifest文件的内容下载相应的资源并且进行离线存储。如果已经访问过app并且资源已经离线存储了，那么浏览器就会使用离线的资源加载页面，然后浏览器会对比新的manifest文件与旧的manifest文件，如果文件没有发生改变，就不做任何操作，如果文件改变了，那么就会重新下载文件中的资源并进行离线存储。

离线的情况下，浏览器就直接使用离线存储的资源

**1.27请描述一下 cookies，sessionStorage 和 localStorage 的区别**

cookie是网站为了标示用户身份而储存在用户本地终端（Client Side）上的数据（通常经过加密）

cookie数据始终在同源的http请求中携带（即使不需要），记会在浏览器和服务器间来回传递

sessionStorage和localStorage不会自动把数据发给服务器，仅在本地保存

存储大小：

cookie数据大小不能超过4k

sessionStorage和localStorage虽然也有存储大小的限制，但比cookie大得多，可以达到5M或更大

有期时间：

localStorage 存储持久数据，浏览器关闭后数据不丢失除非主动删除数据

sessionStorage 数据在当前浏览器窗口关闭后自动删除

cookie 设置的cookie过期时间之前一直有效，即使窗口或浏览器关闭

**1.28 css sprite是什么,有什么优缺点**

概念：将多个小图片拼接到一个图片中。通过background-position和元素尺寸调节需要显示的背景图案。

优点：

减少HTTP请求数，极大地提高页面加载速度

增加图片信息重复度，提高压缩比，减少图片大小

更换风格方便，只需在一张或几张图片上修改颜色或样式即可实现

缺点：

图片合并麻烦

维护麻烦，修改一个图片可能需要从新布局整个图片，样式

## 1.29 canvas 如何绘制一个三角形|正方形

moveto 是移动到某个坐标， lineto 是从当前坐标连线到某个坐标。  
这两个函数加起来就是画一条直线。 画线要用“笔”，那么MoveTo()把笔要画的起始位置固定了（x,y）然后要固定终止位置要用到LineTo函数确定终止位置（xend,yend）,这样一条线就画出来了。 每次与前面一个坐标相连 。   
stroke() 方法会实际地绘制出通过 moveTo() 和 lineTo() 方法定义的路径。默认颜色是黑色。

|  |
| --- |
| <!DOCTYPE HTML5><html><head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  <title>画布</title>  </head> <body>  <canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">  Your browser does not support the canvas element.  </canvas>  <script type="text/javascript">  var c=document.getElementById("myCanvas");//三角形  var cxt=c.getContext("2d");  cxt.moveTo(10,10);  cxt.lineTo(50,50);  cxt.lineTo(10,50);  cxt.lineTo(10,10);  cxt.stroke();//正方形  var cxt2=c.getContext("2d");  cxt2.moveTo(60,10);  cxt2.lineTo(100,10);  cxt2.lineTo(100,50);  cxt2.lineTo(60,50);  cxt2.lineTo(60,10);  cxt2.stroke();  </script></body></html> |

## 1.30弹性盒子模型? flex|box区别?

1. 引入弹性盒布局模型的目的是提供一种更加有效的方式来对一个容器中的条目进行排列、对齐和分配空白空间。  
     
   即便容器中条目的尺寸未知或是动态变化的，弹性盒布局模型也能正常的工作。在该布局模型中，容器会根据布局的需要，调整其中包含的条目的尺寸和顺序来最好地填充所有可用的空间。  
     
   当容器的尺寸由于屏幕大小或窗口尺寸发生变化时，其中包含的条目也会被动态地调整。比如当容器尺寸变大时，其中包含的条目会被拉伸以占满多余的空白空间；当容器尺寸变小时，条目会被缩小以防止超出容器的范围。弹性盒布局是与方向无关的。  
     
   在传统的布局方式中，block 布局是把块在垂直方向从上到下依次排列的；  
   而 inline 布局则是在水平方向来排列。弹性盒布局并没有这样内在的方向限制，可以由开发人员自由操作。   
     
   (2) flex和box的区别: display：box 是老规范，要兼顾古董机子就加上它； 父级元素有display:box;属性之后。他的子元素里面加上box-flex属性。可以让子元素按照父元素的宽度进行一定比例的分占空间。 flex是最新的，董机老机子不支持的；  
     
   父元素设置display:flex后，子元素宽度会随父元素宽度的改变而改变，而display:box不会。 Android UC浏览器只支持display: box语法；而iOS UC浏览器则支持两种方式。

## 1.31 解释在ie低版本下的怪异盒模型和c3的怪异盒模型 和 弹性盒模型?

IE当padding+border的值小于width或者height:   
盒模型的宽度=margin(左右)+width（width已经包含了padding和border的值）   
盒模型的高度=margin(上下)+height（height已经包含了padding和border的值）   
当padding+border的值大于width或者height:   
盒模型的宽度=margin(左右)+padding(左右)+border(左右)   
盒模型的高度=margin(上下)+padding(上下)+border(上下)+19px （加一个默认行高 19px） 所以相当于是padding+border和width或者height比大小，谁大取谁。   
以上几种DOCTYPE都是标准的文档类型，无论使用哪种模式完整定义DOCTYPE，都会触发标准模式，而如果DOCTYPE缺失则在ie6，ie7，ie8下将会触发怪异模式（quirks 模式） CSS3box-sizing有两个值一个是content-box，另一个是border-box。   
当设置为box-sizing:content-box时，将采用标准模式解析计算，也是默认模式；   
当设置为box-sizing:border-box时，将采用怪异模式解析计算；   
Css3弹性盒模型引入了新的盒子模型—弹性盒模型，该模型决定一个盒子在其他盒子中的分布方式以及如何处理可用的空间。使用该模型，可以很轻松的创建自适应浏览器窗口的流动布局或自适应字体大小的弹性布局。

# CSS3面试题

## 2.1 解释box-sizing

box-sizing属性主要用来控制元素的盒模型的解析模式。默认值是content-box。

content-box：让元素维持W3C的标准盒模型。元素的宽度/高度由border + padding + content的宽度/高度决定，设置width/height属性指的是content部分的宽/高

border-box：让元素维持IE传统盒模型（IE6以下版本和IE6~7的怪异模式）。设置width/height属性指的是border + padding + content

应用场景：统一风格的表单元素 表单中有一些input元素其实还是展现的是传统IE盒模型，带有一些默认的样式，而且在不同平台或者浏览器下的表现不一，造成了表单展现的差异。此时我们可以通过box-sizing属性来构建一个风格统一的表单元素

**2.2 水平垂直居中的方法**

行内元素布局

line-height + text-align

行内块元素布局

vertical-align + line-height + text-align

块布局

position absolute + margin auto +top,bottom,left,right=0

position absolute + top:50%,left50% + translate(-50%, -50%)

position absolute + top:50%,left50% +

margin:-元素宽度一半 0 0 -元素高度一半

父容器子容器不确定宽高的块级元素，做上下居中

1.flex

#wrap{

display:flex;

justify-content:center;

align-items:center;

}

2.table

.parent {

text-align: center;//水平居中

display: table-cell;

vertical-align: middle;//垂直居中

}

.child {

display: inline-block;//防止块级元素宽度独占一行，内联元素可不设置

}

3.webkit-box

//对父级元素设置

position: relative;

display: -webkit-box;

-webkit-box-align: center;

-webkit-box-pack: center;

**2.3 实现左边定宽右边自适应效果**

1.table(父级元素)与tabel-cell（两个子集元素）

2.flex(父级元素)+flex :1（右边子元素）

3.左边定宽，并且左浮动；右边设置距离左边的宽度

4.左边定宽，左边设置position:absolute；右边设置距离左边的宽度

**2.4 三列布局（中间固定两边自适应宽度）**

1. 采用浮动布局（左边左浮动，右边右浮动，中间margin：0 宽度值）

2. 绝对定位方式（左右绝对定位，左边left0右边right0，中间上同）

3. 圣杯布局与双飞翼布局

**2.5 清除浮动元素的方法和各自的优缺点**

清除浮动，实际上是清除父元素的高度塌陷。因为子元素脱离了父元素的文档流，所以，父元素失去了高度，导致了塌陷。要解决这个问题，就是让父元素具有高度。

浮动元素的特性： 在正常布局中位于该浮动元素之下的内容，此时会围绕着浮动元素，填满其右侧的空间。浮动到右侧的元素，其他内容将从左侧环绕它（浮动元素影响的不仅是自己，它会影响周围的元素对其进行环绕。float仍会占据其位置，position:absolute不占用页面空间 会有重叠问题 ）

1.在浮动元素末尾添加空标签清除浮动 clear:both （缺点：增加无意义标签）

<div style="clear:both;"></div>

2.给父元素设置 overflow:auto属性

3.after伪元素

**2.6 css实现自适应正方形**

方案一：CSS3 vw 单位（相对值的单位）

方案二：设置垂直方向的padding撑开容器

方案三：利用伪元素的 margin(padding)-top 撑开容器

**2.7 介绍position的值**

1. absolute :生成绝对定位的元素， 相对于最近一级的 定位不是 static 的父元素来进行定位；如果没有定位父级，则参照初始包含块进行定位。

2. fixed （老IE不支持）生成绝对定位的元素，通常相对于浏览器窗口。

3. relative 生成相对定位的元素，相对于其在普通流中的位置进行定位。

4. static 默认值。没有定位，元素出现在正常的流中

**2.8 如何在页面上实现一个圆形的可点击区域**

1、字体图标

2、border-radius

3、纯js实现 需要求一个点在不在圆上简单算法、获取鼠标坐标等等

**2.9 CSS3新增伪类有那些**

p:first-of-type 选择属于其父元素的首个<p>元素的每个<p>元素

p:last-of-type 选择属于其父元素的最后<p> 元素的每个<p>元素

p:only-of-type 选择属于其父元素唯一的<p>元素的每个<p>元素

p:only-child 选择属于其父元素的唯一子元素的每个<p>元素

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

::after 在元素之前添加内容,也可以用来做清除浮动

::before 在元素之后添加内容

:disabled 控制表单控件的禁用状态

:checked 单选框或复选框被选中

**2.10 CSS3有哪些新特性**

新增各种CSS选择器 （: not(.input)：所有 class 不是“input”的节点）

圆角 （border-radius:8px）

多列布局 （column）

阴影和反射 （Shadow\Reflect）

文字特效 （text-shadow）

线性渐变 （gradient）

旋转，缩放,定位,倾斜 （transform）

动画 （Animation）

多背景，背景裁剪

**2.11 请解释一下CSS3的Flexbox（弹性盒布局模型）,以及适用场景**

一个用于页面布局的全新CSS3功能，Flexbox可以把列表放在同一个方向（从上到下排列，从左到右），并让列表能延伸到占用可用的空间。

较为复杂的布局还可以通过嵌套一个伸缩容器（flex container）来实现。

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"。

常规布局是基于块和内联流方向，而Flex布局是基于flex-flow流可以很方便的用来做局中，能对不同屏幕大小自适应。

在布局上有了比以前更加灵活的空间

**2.12 用纯CSS创建一个三角形的原理是什么**

把上、左、右三条边隐藏掉（颜色设为 transparent）

#demo {width: 0;height: 0;border-width: 20px;border-style: solid;

border-color: transparent transparent red transparent;}

**2.13 一个满屏 品 字布局 如何设计**

上面的div宽100%，

下面的两个div分别宽50%，

然后用float或者inline使其不换行即可

**2.14 css等高布局如何实现**

利用padding-bottom|margin-bottom正负值相抵；

设置父容器设置超出隐藏（overflow:hidden），这样子父容器的高度就还是它里面的列没有设定padding-bottom时的高度，

当它里面的任 一列高度增加了，则父容器的高度被撑到里面最高那列的高度，其他比这列矮的列会用它们的padding-bottom补偿这部分高度差

**2.15 li与li之间有看不见的空白间隔是什么原因引起的？有什么解决办法？**

行框的排列会受到中间空白（回车\空格）等的影响，因为空格也属于字符,这些空白也会被应用样式，占据空间，所以会有间隔，把字符大小设为0，就没有空格了

**2.16 为什么要初始化CSS样式**

因为浏览器的兼容问题，不同浏览器对有些标签的默认值是不同的，如果没对CSS初始化往往会出现浏览器之间的页面显示差异。

当然，初始化样式会对SEO有一定的影响，但鱼和熊掌不可兼得，但力求影响最小的情况下初始化。

最简单的初始化方法： \* {padding: 0; margin: 0;}

淘宝的样式初始化代码：

body, h1, h2, h3, h4, h5, h6, hr, p, blockquote, dl, dt, dd, ul, ol, li, pre, form, fieldset, legend, button, input, textarea, th, td {margin:0; padding:0; }

body, button, input, select, textarea { font:12px/1.5tahoma, arial; }

h1, h2, h3, h4, h5, h6{ font-size:100%; }

address, cite, dfn, em, var { font-style:normal; }

code, kbd, pre, samp { font-family:couriernew, courier, monospace; }

small{ font-size:12px; }

ul, ol { list-style:none; }

a { text-decoration:none; }

a:hover { text-decoration:underline; }

sup { vertical-align:text-top; }

sub{ vertical-align:text-bottom; }

legend { color:#000; }

fieldset, img { border:0; }

button, input, select, textarea { font-size:100%; }

table { border-collapse:collapse; border-spacing:0; }

**2.17 请解释一下为什么需要清除浮动？清除浮动的方式**

清除浮动是为了清除使用浮动元素产生的影响。浮动的元素，高度会塌陷，而高度的塌陷使我们页面后面的布局不能正常显示。

1、父级div定义height；

2、父级div 也一起浮动；

3、常规的使用一个class；

.clearfix::before, .clearfix::after {

content: " ";

display: table;

}

.clearfix::after {

clear: both;

}

.clearfix {

\*zoom: 1;

}

**2.18 SASS/LESS编译的时候，浮动元素的父级div定义伪类:after**

&::after,&::before{

content: " ";

visibility: hidden;

display: block;

height: 0;

clear: both;

}

解析原理：

1) display:block 使生成的元素以块级元素显示,占满剩余空间;

2) height:0 避免生成内容破坏原有布局的高度。

3) visibility:hidden 使生成的内容不可见，并允许可能被生成内容盖住的内容可以进行点击和交互;

4）通过 content:"."生成内容作为最后一个元素，至于content里面是点还是其他都是可以的，例如oocss里面就有经典的 content:".",有些版本可能content 里面内容为空,一丝冰凉是不推荐这样做的,firefox直到7.0 content:”" 仍然会产生额外的空隙；

5）zoom：1 触发IE hasLayout。

通过分析发现，除了clear：both用来闭合浮动的，其他代码无非都是为了隐藏掉content生成的内容，这也就是其他版本的闭合浮动为什么会有font-size：0，line-height：0。

**2.19 CSS优化、提高性能的方法有哪些**

关键选择器（key selector）。选择器的最后面的部分为关键选择器（即用来匹配目标元素的部分）；

如果规则拥有 ID 选择器作为其关键选择器，则不要为规则增加标签。过滤掉无关的规则（这样样式系统就不会浪费时间去匹配它们了）；

提取项目的通用公有样式，增强可复用性，按模块编写组件；增强项目的协同开发性、可维护性和可扩展性;

使用预处理工具或构建工具（gulp对css进行语法检查、自动补前缀、打包压缩、自动优雅降级）；

**2.20 怎么让Chrome支持小于12px 的文字**

1、用图片：如果是内容固定不变情况下，使用将小于12px文字内容切出做图片，这样不影响兼容也不影响美观。

2、使用12px及12px以上字体大小：为了兼容各大主流浏览器，建议设计美工图时候设置大于或等于12px的字体大小，如果是接单的这个时候就需要给客户讲解小于12px浏览器不兼容等事宜。

3、继续使用小于12px字体大小样式设置：如果不考虑chrome可以不用考虑兼容，同时在设置小于12px对象设置-webkit-text-size-adjust:none，做到最大兼容考虑。

4、使用12px以上字体：为了兼容、为了代码更简单 从新考虑权重下兼容性。

**2.21 如果需要手动写动画，你认为最小时间间隔是多久，为什么？（阿里）**

多数显示器默认频率是60Hz，即1秒刷新60次，所以理论上最小间隔为1/60＊1000ms ＝ 16.7ms

**2.22 display:inline-block 什么时候会显示间隙？(携程)**

移除空格、使用margin负值、使用font-size:0、letter-spacing、word-spacing、浮动

**2.23 超链接访问过后hover样式就不出现的问题是什么？如何解决**

被点击访问过的超链接样式不再具有hover和active了,解决方法是改变CSS属性的排列顺序: L-V-H-A（link,visited,hover,active）

**2.24 rgba()和opacity的透明效果有什么不同**

rgba()和opacity都能实现透明效果，但最大的不同是opacity作用于元素，以及元素内的所有内容的透明度，

而rgba()只作用于元素的颜色或其背景色。（设置rgba透明的元素的子元素不会继承透明效果！）

**2.25 px和em的区别**

px和em都是长度单位，区别是：

px值固定，容易计算。

em值不固定，是相对单位，其相对应父级元素的字体大小会调整

## 2.26 描述一个”reset”的CSS文件并如何使用它。知道normalize.css吗？你了解他们的不同之处？

Reset样式的目的是清除浏览器某些默认样式，方便css书写，例如：

\*｛margin:0;padding:0;list-style:none;｝

Normalize的理念与reset不同，它并不清除浏览器默认样式，而是尽量将所有浏览器的默认样式统一

**2.27 Sass、LESS是什么？大家为什么要使用他们？**

他们是CSS预处理器。他是CSS上的一种抽象层。他们是一种特殊的语法/语言编译成CSS。

例如Less是一种动态样式语言. 将CSS赋予了动态语言的特性，如变量，继承，运算， 函数. LESS 既可以在客户端上运行 (支持IE 6+, Webkit, Firefox)，也可一在服务端运行 (借助 Node.js)。

为什么要使用它们？

结构清晰，便于扩展。

可以方便地屏蔽浏览器私有语法差异。这个不用多说，封装对浏览器语法差异的重复处理，减少无意义的机械劳动。

可以轻松实现多重继承。

完全兼容 CSS 代码，可以方便地应用到老项目中。LESS 只是在 CSS 语法上做了扩展，所以老的 CSS 代码也可以与 LESS 代码一同编译。

**2.28 为什么要初始化样式**

用于浏览器默认css样式的存在并且不同浏览器对相同HTML标签的默认样式不同，若不初始化会造成不同浏览器之间的显示差异

## 2.29 对WEB标准以及W3C的理解与认识

标签闭合、标签小写、不乱嵌套、提高搜索机器人搜索几率、使用外 链css和js脚本、结构行为表现的分离、文件下载与页面速度更快、内容能被更多的用户所访问、内容能被更广泛的设备所访问、更少的代码和组件，容易维 护、改版方便，不需要变动页面内容、提供打印版本而不需要复制内容、提高网站易用性

**2.30 使用css实现一个持续的动画效果**

animation:mymove 5s infinite;

@keyframes mymove {

from {top:0px;}

to {top:200px;}

}

**2.31 重排和重绘**

部分渲染树（或者整个渲染树）需要重新分析并且节点尺寸需要重新计算。这被称为重排。注意这里至少会有一次重排-初始化页面布局。

由于节点的几何属性发生改变或者由于样式发生改变，例如改变元素背景色时，屏幕上的部分内容需要更新。这样的更新被称为重绘。

**2.32 什么情况会触发重排和重绘？**

* 添加、删除、更新 DOM 节点
* 通过 display: none 隐藏一个 DOM 节点-触发重排和重绘
* 通过 visibility: hidden 隐藏一个 DOM 节点-只触发重绘，因为没有几何变化
* 移动或者给页面中的 DOM 节点添加动画
* 添加一个样式表，调整样式属性
* 用户行为，例如调整窗口大小，改变字号，或者滚动

## 2.33. animation对应的属性

写法：animation: name duration timing-function delay iteration-count direction;   
下面是对应的属性的介绍   
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。

## 2.34 transition?

css的transition允许css的属性值在一定的时间区间内平滑地过渡。这种效果可以在鼠标单击、获得焦点、被点击或对元素任何改变中触发，并圆滑地以动画效果改变CSS的属性值 注意区别transform，transform是进行2D转换的 如移动，比例化，反过来，旋转，和拉伸元素。

# 移动端面试题

**3.1 说说你知道的移动端web的兼容性bug**

1、一些情况下对非可点击元素如(label,span)监听click事件，ios下不会触发，css增加cursor:pointer就搞定了。

2.position 在Safari下的两个属性(top,left)需要都写，只写一个容易发生错乱

3.Input 的placeholder会出现文本位置偏上的情况

input 的placeholder会出现文本位置偏上的情况：PC端设置line-height等于height能够对齐，而移动端仍然是偏上，解决是设置line-height:normal

4.ios点击穿透问题

event.preventDefault + 重新给<a>元素绑定新的链接

**3.2 有哪些多屏适配方案**

1. media query + rem

2. flex

3. 弹性布局

**3.3 H5页面窗口自动调整到设备宽度，并禁止用户缩放页面**

**2018-03-28_165321**

## 3.4 忽略将页面中的数字识别为电话号码

<meta name="format-detection" content="telephone=no" />

**3.5 忽略Android平台中对邮箱地址的识别**

<meta name="format-detection" content="email=no" />

## 3.6 移动端click屏幕产生200-300 ms的延迟响应

移动设备上的web网页是有300ms延迟的，玩玩会造成按钮点击延迟甚至是点击失效。

以下是历史原因，来源一个公司内一个同事的分享：2007年苹果发布首款iphone上IOS系统搭载的safari为了将适用于PC端上大屏幕的网页能比较好的展示在手机端上，使用了双击缩放(double tap to zoom)的方案，比如你在手机上用浏览器打开一个PC上的网页，你可能在看到页面内容虽然可以撑满整个屏幕，但是字体、图片都很小看不清，此时可以快速双击屏幕上的某一部分，你就能看清该部分放大后的内容，再次双击后能回到原始状态。

双击缩放是指用手指在屏幕上快速点击两次，iOS 自带的 Safari 浏览器会将网页缩放至原始比例。

原因就出在浏览器需要如何判断快速点击上，当用户在屏幕上单击某一个元素时候，例如跳转链接<a href="#"></a>，此处浏览器会先捕获该次单击，但浏览器不能决定用户是单纯要点击链接还是要双击该部分区域进行缩放操作，所以，捕获第一次单击后，浏览器会先Hold一段时间t，如果在t时间区间里用户未进行下一次点击，则浏览器会做单击跳转链接的处理，如果t时间里用户进行了第二次单击操作，则浏览器会禁止跳转，转而进行对该部分区域页面的缩放操作。那么这个时间区间t有多少呢？在IOS safari下，大概为300毫秒。这就是延迟的由来。造成的后果用户纯粹单击页面，页面需要过一段时间才响应，给用户慢体验感觉，对于web开发者来说是，页面js捕获click事件的回调函数处理，需要300ms后才生效，也就间接导致影响其他业务逻辑的处理。

解决方案：

fastclick可以解决在手机上点击事件的300ms延迟

zepto的touch模块，tap事件也是为了解决在click的延迟问题

**3.7 触摸事件的响应顺序**

1. ontouchstart
2. ontouchmove
3. ontouchend
4. onclick

**3.8 什么是Retina 显示屏，带来了什么问题**

retina：一种具备超高像素密度的液晶屏，同样大小的屏幕上显示的像素点由1个变为多个，如在同样带下的屏幕上，苹果设备的retina显示屏中，像素点1个变为4个

在高清显示屏中的位图被放大，图片会变得模糊，因此移动端的视觉稿通常会设计为传统PC的2倍

那么，前端的应对方案是：



**3.9 ios系统中元素被触摸时产生的半透明灰色遮罩怎么去掉**

ios用户点击一个链接，会出现一个半透明灰色遮罩, 如果想要禁用，可设置-webkit-tap-highlight-color的alpha值为0，也就是属性值的最后一位设置为0就可以去除半透明灰色遮罩

a,button,input,textarea{-webkit-tap-highlight-color: rgba(0,0,0,0;)}

**3.10 webkit表单输入框placeholder的颜色值怎么改变？**

input::-webkit-input-placeholder{color:#AAAAAA;}

input:focus::-webkit-input-placeholder{color:#EEEEEE;}

**3.11 打电话发短信写邮件怎么实现**

****

**3.12 解释一下移动端的缩放**

放大

会使一个css像素的面积变大

一个css像素所包含的物理像素变多

视觉视口所包含的css像素的个数变少

缩小（默认情况下（没有meta标签）其实已经有了一个缩小操作了）

会使一个css像素的面积变小

一个css像素所包含的物理像素变少

视觉视口所包含的css像素的个数变多

## 3.13手机端上图片长时间点击会选中图片，如何处理?

onselect=function() {  
 return false  
}

## 3.14 video标签的几个个属性方法

1. src：视频的URL
2. poster：视频封面，没有播放时显示的图片
3. preload：预加载
4. autoplay：自动播放
5. loop：循环播放
6. controls：浏览器自带的控制条
7. width：视频宽度
8. height：视频高度

## 3.15 常见的视频编码格式有几种?视频格式有几种?

视频格式：MPEG-1、MPEG-2和MPEG4 、AVI 、RM、ASF和WMV格式   
视频编码格式：H.264、MPEG-4、MPEG-2、WMA-HD以及VC-1

## 3.16 canvas在标签上设置宽高 和在style中设置宽高有什么区别？

canvas标签的width和height是画布实际宽度和高度，绘制的图形都是在这个上面。而style的width和height是canvas在浏览器中被渲染的高度和宽度。如果canvas的width和height没指定或值不正确，就被设置成默认值 。

## 3.17 viewport 所有属性 ？

(1)width :设置layout viewport的宽度，为一个正整数，或字符串'device-width';  
(2)initial-scale:设置页面的初始缩放值，为一个数字，可以带小数。  
(3)minimum-scale:允许用户的最小缩放值，为一个数字，可以带小数。  
(4)maximum-scale:允许用户的最大缩放值，为一个数字，可以带小数。  
(5)height:设置layout viewport的高度，这个属性对我们并不重要，很少使用  
(6)user-scalable:是否允许用户进行缩放，值为‘no’或者‘yes’。  
安卓中还支持：target-densitydpi，表示目标设备的密度等级，作用是决定css中的1px 代表多少物理像素  
(7)target-densitydpi:值可以为一个数值或者 high-dpi 、 medium-dpi、 low-dpi、 device-dpi 这几个字符串中的一个

## 3.18 media属性？screen？All？max-width?min-width?

media 属性规定被链接文档将显示在什么设备上。media 属性用于为不同的媒介类型规定不同的样式。Screen计算机默认屏幕，all适用于所有设备，max-width超过最大宽度就不执行，min-width必须大于最小宽度才执行。