# 做笔记

# **一、HTML+CSS**

## 1．Doctype作用？标准模式与兼容模式各有什么区别?

         （1）<!DOCTYPE>声明位于位于HTML文档中的第一行，处于 <html> 标签之前。告知浏览器的解析器用什么文档标准解析这个文档。

                   DOCTYPE不存在或格式不正确会导致文档以兼容模式呈现。

         （2）标准模式的排版 和JS运作模式都是以该浏览器支持的最高标准运行。在兼容模式中，页面以宽松的向后兼容的方式显示,模拟老式浏览器的行为以防止站点无法工作。

## 2.行内元素有哪些？块级元素有哪些？ 空(void)元素有那些？

首先：CSS规范规定，每个元素都有display属性，确定该元素的类型，每个元素都有默认的display值，如div的display默认值为“block”，则为“块级”元素；

span默认display属性值为“inline”，是“行内”元素。

（1）行内元素有：a b span img input select strong（强调的语气）

（2）块级元素有：div ul ol li dl dt dd h1 h2 h3 h4…p

（3）常见的空元素：

    <br> <hr> <img> <input> <link> <meta>

    鲜为人知的是：

<area> <base> <col> <command> <embed> <keygen> <param> <source> <track> <wbr>

## 3.页面导入样式时，使用link和@import有什么区别？

（1）link属于XHTML标签，除了加载CSS外，还能用于定义RSS, 定义rel连接属性等作用；而@import是CSS提供的，只能用于加载CSS;

（2）页面被加载的时，link会同时被加载，而@import引用的CSS会等到页面被加载完再加载;

（3）import是CSS2.1 提出的，只在IE5以上才能被识别，而link是XHTML标签，无兼容问题;

## 4.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。

## 5.简述一下你对HTML语义化的理解？

  用正确的标签做正确的事情。

  html语义化让页面的内容结构化，结构更清晰，便于对浏览器、搜索引擎解析;

  即使在没有样式CSS情况下也以一种文档格式显示，并且是容易阅读的;

  搜索引擎的爬虫也依赖于HTML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。

## 6.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>

## 7.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>

## 8.网页验证码是干嘛的，是为了解决什么安全问题。

         区分用户是计算机还是人的公共全自动程序。可以防止恶意破解密码、刷票、论坛灌水；

         有效防止黑客对某一个特定注册用户用特定程序暴力破解方式进行不断的登陆尝试。

## 9.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。

## 10.介绍一下标准的CSS的盒子模型？低版本IE的盒子模型有什么不同的？

         （1）有两种， IE 盒子模型、W3C 盒子模型；

         （2）盒模型： 内容(content)、填充(padding)、边界(margin)、 边框(border)；

         （3）区  别： IE的content部分把 border 和 padding计算了进去;

## 11.CSS选择符有哪些？哪些属性可以继承？

         1.id选择器（ # myid）

    2.类选择器（.myclassname）

    3.标签选择器（div, h1, p）

    4.相邻选择器（h1 + p）

    5.子选择器（ul > li）

    6.后代选择器（li a）

    7.通配符选择器（ \* ）

    8.属性选择器（a[rel = "external"]）

    9.伪类选择器（a:hover, li:nth-child）

 可继承的样式： font-size font-family color, UL LI DL DD DT;

 不可继承的样式：border  padding  margin  width  height ;

## 12.CSS优先级算法如何计算？

        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.如何居中div？

         水平居中：给div设置一个宽度，然后添加margin:0 auto属性

         div{width:200px;margin:0 auto;}

         让绝对定位的div居中

         div {position: absolute;width: 300px;height: 300px;margin: auto;top: 0;

             left: 0;bottom: 0;right: 0;background-color: pink; /\* 方便看效果 \*/}

         水平垂直居中一

         确定容器的宽高 宽500 高 300 的层,设置层的外边距

         div {position: relative;     /\* 相对定位或绝对定位均可 \*/

             width:500px;height:300px;top: 50%;left: 50%;margin: -150px 0 0 -250px;

/\* 外边距为自身宽高的一半 \*/

             background-color: pink;     /\* 方便看效果 \*/}

         水平垂直居中二

         未知容器的宽高，利用 `transform` 属性

         div {position: absolute;     /\* 相对定位或绝对定位均可 \*/

    width:500px;height:300px;top: 50%;left: 50%;transform: translate(-50%, -50%);

    background-color: pink;     /\* 方便看效果 \*/}

         水平垂直居中三

                   利用 flex 布局,实际使用时应考虑兼容性

         .container {display: flex; align-items: center;/\* 垂直居中 \*/justify-content: center;/\* 水平居中 \*/}

         .container div {width: 100px;height: 100px;background-color: pink;/\* 方便看效果 \*/}

## 14.display有哪些值？说明他们的作用。

         block         块类型。默认宽度为父元素宽度，可设置宽高，换行显示。

    none          缺省值。象行内元素类型一样显示。

    inline        行内元素类型。默认宽度为内容宽度，不可设置宽高，同行显示。

    inline-block  默认宽度为内容宽度，可以设置宽高，同行显示。

    list-item     象块类型元素一样显示，并添加样式列表标记。

    table         此元素会作为块级表格来显示。

inherit       规定应该从父元素继承 display 属性的值。

## 15.position的值relative和absolute定位原点是？

         absolute

             生成绝对定位的元素，相对于值不为 static的第一个父元素进行定位。

    fixed （老IE不支持）

             生成绝对定位的元素，相对于浏览器窗口进行定位。

    relative

             生成相对定位的元素，相对于其正常位置进行定位。

    static

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

    inherit

             规定从父元素继承 position 属性的值。

## 16.一个满屏 ‘品’字布局 如何设计?

         简单的方式：

             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

## 17.css多列等高如何实现？

         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。

## 18.经常遇到的浏览器的兼容性有哪些？原因，解决方法是什么，常用hack的技巧 ？

         png24位的图片在iE6浏览器上出现背景，解决方案是做成PNG8.

        浏览器默认的margin和padding不同。解决方案是加一个全局的\*{margin:0;padding:0;}来统一。

        IE6双边距bug:块属性标签float后，又有横行的margin情况下，在ie6显示margin比设置的大。

     浮动ie产生的双倍距离 #box{ float:left; width:10px; margin:0 0 0 100px;}

        这种情况之下IE会产生20px的距离，解决方案是在float的标签样式控制中加入 ——\_display:inline;将其转化为行内属性。(\_这个符号只有ie6会识别)

       渐进识别的方式，从总体中逐渐排除局部。

       首先，巧妙的使用“\9”这一标记，将IE游览器从所有情况中分离出来。

         接着，再次使用“+”将IE8和IE7、IE6分离开来，这样IE8已经独立识别。

           css

               .bb{

                   background-color:red;/\*所有识别\*/

                   background-color:#00deff\9; /\*IE6、7、8识别\*/

                   +background-color:#a200ff;/\*IE6、7识别\*/

                   \_background-color:#1e0bd1;/\*IE6识别\*/

               }

    IE下,可以使用获取常规属性的方法来获取自定义属性,

            也可以使用getAttribute()获取自定义属性;

            Firefox下,只能使用getAttribute()获取自定义属性。

            解决方法:统一通过getAttribute()获取自定义属性。

          IE下,even对象有x,y属性,但是没有pageX,pageY属性;

     Firefox下,event对象有pageX,pageY属性,但是没有x,y属性。

           解决方法：（条件注释）缺点是在IE浏览器下可能会增加额外的HTTP请求数。

     Chrome 中文界面下默认会将小于 12px 的文本强制按照 12px 显示,

            可通过加入 CSS 属性 -webkit-text-size-adjust: none; 解决。

         超链接访问过后hover样式就不出现了 被点击访问过的超链接样式不在具有hover和active了解决方法是改变CSS属性的排列顺序:

         L-V-H-A :  a:link {} a:visited {} a:hover {} a:active {}

## 19.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。

## 20.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;} （强烈不建议）

         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, \5b8b\4f53; }

         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; }

## 21.absolute的containing block(容器块)计算方式跟正常流有什么不同？

         无论属于哪种，都要先找到其祖先元素中最近的 position 值不为 static 的元素，然后再判断：

         1、若此元素为 inline 元素，则 containing block 为能够包含这个元素生成的第一个和最后一个 inline box 的 padding box (除 margin, border 外的区域) 的最小矩形；

         2、否则,则由这个祖先元素的 padding box 构成。

         如果都找不到，则为 initial containing block。

         补充：

         1. static(默认的)/relative：简单说就是它的父元素的内容框（即去掉padding的部分）

         2. absolute: 向上找最近的定位为absolute/relative的元素

         3. fixed: 它的containing block一律为根元素(html/body)，根元素也是initial containing block

25.CSS里的visibility属性有个collapse属性值是干嘛用的？在不同浏览器下以后什么区别？

         其实visibility可以有第三种值，就是collapse。当一个元素的visibility属性被设置成collapse值后，

         对于一般的元素，它的表现跟hidden是一样的。但例外的是，如果这个元素是table相关的元素，例如table行，

         table group，table列，table column group，它的表现却跟display: none一样，也就是说，它们占用的空间也会释放。

         在谷歌浏览器里，使用collapse值和使用hidden值没有什么区别

         在火狐浏览器、Opera和IE11里，使用collapse值的效果就如它的字面意思：table的行会消失，它的下面一行会补充它的位置。

## 22.对BFC规范(块级格式化上下文：block formatting context)的理解？

         （W3C CSS 2.1 规范中的一个概念,它是一个独立容器，决定了元素如何对其内容进行定位,以及与其他元素的关系和相互作用。）

         一个页面是由很多个 Box 组成的,元素的类型和 display 属性,决定了这个 Box 的类型。

        不同类型的 Box,会参与不同的 Formatting Context（决定如何渲染文档的容器）,

        因此Box内的元素会以不同的方式渲染,也就是说BFC内部的元素和外部的元素不会互相影响。

## 23.css定义的权重

         以下是权重的规则：标签的权重为1，class的权重为10，id的权重为100，以下例子是演示各种定义的权重值：

         权重为1   div{}

         权重为10  .class1{}

         权重为100   #id1{}

         权重为100+1=101   #id1 div{}

         权重为10+1=11  .class1 div{}

         权重为10+10+1=21  .class1 .class2 div{}

         如果权重相同，则最后定义的样式会起作用，但是应该避免这种情况出现

## 24.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;

             }

         4、SASS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。

## 25.什么是外边距合并？

         外边距合并指的是，当两个垂直外边距相遇时，它们将形成一个外边距。

         合并后的外边距的高度等于两个发生合并的外边距的高度中的较大者。

## 26.zoom:1的清除浮动原理?

         清除浮动，触发hasLayout；

         Zoom属性是IE浏览器的专有属性，它可以设置或检索对象的缩放比例。解决ie下比较奇葩的bug。

         譬如外边距（margin）的重叠，浮动清除，触发ie的haslayout属性等。

         来龙去脉大概如下：

         当设置了zoom的值之后，所设置的元素就会就会扩大或者缩小，高度宽度就会重新计算了，这里一旦改变zoom值时其实也会发生重新渲染，运用这个原理，也就解决了ie下子元素浮动时候父元素不随着自动扩大的问题。

         Zoom属是IE浏览器的专有属性，火狐和老版本的webkit核心的浏览器都不支持这个属性。然而，zoom现在已经被逐步标准化，出现在 CSS 3.0 规范草案中。

         目前非ie由于不支持这个属性，它们又是通过什么属性来实现元素的缩放呢？

         可以通过css3里面的动画属性scale进行缩放。

## 27.使用 CSS 预处理器吗？喜欢那个？

         SASS (SASS、LESS没有本质区别，只因为团队前端都是用的SASS)

## 28.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）；

## 29.浏览器是怎样解析CSS选择器的？

         样式系统从关键选择器开始匹配，然后左移查找规则选择器的祖先元素。

         只要选择器的子树一直在工作，样式系统就会持续左移，直到和规则匹配，或者是因为不匹配而放弃该规则。

## 30.margin和padding分别适合什么场景使用？

         margin是用来隔开元素与元素的间距；padding是用来隔开元素与内容的间隔。

         margin用于布局分开元素使元素与元素互不相干；

         padding用于元素与内容之间的间隔，让内容（文字）与（包裹）元素之间有一段

## 31.::before 和 :after中双冒号和单冒号 有什么区别？解释一下这2个伪元素的作用。

         单冒号(:)用于CSS3伪类，双冒号(::)用于CSS3伪元素。（伪元素由双冒号和伪元素名称组成）

         双冒号是在当前规范中引入的，用于区分伪类和伪元素。不过浏览器需要同时支持旧的已经存在的伪元素写法，

         比如:first-line、:first-letter、:before、:after等，

         而新的在CSS3中引入的伪元素则不允许再支持旧的单冒号的写法。

         想让插入的内容出现在其它内容前，使用::before，否者，使用::after；

         在代码顺序上，::after生成的内容也比::before生成的内容靠后。

         如果按堆栈视角，::after生成的内容会在::before生成的内容之上

## 32.如何修改chrome记住密码后自动填充表单的黄色背景 ？

         input:-webkit-autofill, textarea:-webkit-autofill, select:-webkit-autofill {

           background-color: rgb(250, 255, 189); /\* #FAFFBD; \*/

           background-image: none;

           color: rgb(0, 0, 0);

         }

## 33.设置元素浮动后，该元素的display值是多少？

         自动变成了 display:block

## 34.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。

## 35.让页面里的字体变清晰，变细用CSS怎么做？

         -webkit-font-smoothing: antialiased;

## 36.font-style属性可以让它赋值为“oblique” oblique是什么意思？

         倾斜的字体样式

## 37.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

## 38.设置一个已知ID的div的html内容为xxxx,字体颜色设置为黑色

         方法一：

                   window.onload = function(){

                            var box = document.getElementById('box')

                            box.innerHTML = 'xxxx'

                            var styleNode = document.createElement('style')

                            styleNode.innerHTML += '#box{color:block;}'

                            document.head.appendChild(styleNode)

                   }

         方法二：

                   var dom = document.getElementById(“ID”);

                   dom.innerHTML = “xxxx”

                   dom.style.color = “#000”

## 39.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

## 40.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。

## 41.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：

结构与样式分离

代码语义性好

更符合HTML标准规范

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。

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

a: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

## 43.你能描述一下渐进增强和优雅降级之间的不同吗?

渐进增强 progressive enhancement：针对低版本浏览器进行构建页面，保证最基本的功能，然后再针对高级浏览器进行效果、交互等改进和追加功能达到更好的用户体验。

优雅降级 graceful degradation：一开始就构建完整的功能，然后再针对低版本浏览器进行兼容。

区别：优雅降级是从复杂的现状开始，并试图减少用户体验的供给，而渐进增强则是从一个非常基础的，能够起作用的版本开始，并不断扩充，以适应未来环境的需要。降级（功能衰减）意味着往回看；而渐进增强则意味着朝前看，同时保证其根基处于安全地带。

“优雅降级”观点：

“优雅降级”观点认为应该针对那些最高级、最完善的浏览器来设计网站。而将那些被认为“过时”或有功能缺失的浏览器下的测试工作安排在开发周期的最后阶段，并把测试对象限定为主流浏览器（如 IE、Mozilla 等）的前一个版本。

在这种设计范例下，旧版的浏览器被认为仅能提供“简陋却无妨 (poor, but passable)” 的浏览体验。你可以做一些小的调整来适应某个特定的浏览器。但由于它们并非我们所关注的焦点，因此除了修复较大的错误之外，其它的差异将被直接忽略。

“渐进增强”观点

“渐进增强”观点则认为应关注于内容本身。

内容是我们建立网站的诱因。有的网站展示它，有的则收集它，有的寻求，有的操作，还有的网站甚至会包含以上的种种，但相同点是它们全都涉及到内容。这使得“渐进增强”成为一种更为合理的设计范例。这也是它立即被 Yahoo! 所采纳并用以构建其“分级式浏览器支持 (Graded Browser Support)”策略的原因所在。

## 44.为什么利用多个域名来存储网站资源会更有效？

CDN缓存更方便

突破浏览器并发限制

节约cookie带宽

节约主域名的连接数，优化页面响应速度

防止不必要的安全问题

## 45.请谈一下你对网页标准和标准制定机构重要性的理解。

网页标准和标准制定机构都是为了能让web发展的更‘健康’，开发者遵循统一的标准，降低开发难度，开发成本，SEO也会更好做，也不会因为滥用代码导致各种BUG、安全问题，最终提高网站易用性。​​​​

## 46.请描述一下cookies，sessionStorage和localStorage的区别？

sessionStorage用于本地存储一个会话（session）中的数据，这些数据只有在同一个会话中的页面才能访问并且当会话结束后数据也随之销毁。因此sessionStorage不是一种持久化的本地存储，仅仅是会话级别的存储。而localStorage用于持久化的本地存储，除非主动删除数据，否则数据是永远不会过期的。

web storage和cookie的区别：

Web Storage的概念和cookie相似，区别是它是为了更大容量存储设计的。Cookie的大小是受限的，并且每次你请求一个新的页面的时候Cookie都会被发送过去，这样无形中浪费了带宽，另外cookie还需要指定作用域，不可以跨域调用。

除此之外，Web Storage拥有setItem,getItem,removeItem,clear等方法，不像cookie需要前端开发者自己封装setCookie，getCookie。但是Cookie也是不可以或缺的：Cookie的作用是与服务器进行交互，作为HTTP规范的一部分而存在 ，而Web Storage仅仅是为了在本地“存储”数据而生。

## 47.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。

## 48.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%

## 49.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？

答案：dns缓存，cdn缓存，浏览器缓存，服务器缓存。

## 50.一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。

## 51.你如何理解HTML结构的语义化？

HTML结构语义化：

更符合W3C统一的规范标准，是技术趋势。

没有样式时浏览器的默认样式也能让页面结构很清晰。

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。

对其他非主流终端设备友好。例如机顶盒、PDA、各种移动终端。

对SEO友好。

## 52.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

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。

## 53.有哪项方式可以对一个DOM设置它的CSS样式？

外部样式表，引入一个外部css文件

内部样式表，将css代码放在 <head> 标签内部

内联样式，将css样式直接定义在 HTML 元素内部

## 54.CSS都有哪些选择器？

一、基本选择器

1.      \*       通用元素选择器，匹配任何元素

2.      E       标签选择器，匹配所有使用E标签的元素

3.      .info         class选择器，匹配所有class属性中包含info的元素

4.      #footer   id选择器，匹配所有id属性等于footer的元素

二、多元素的组合选择器

5.      E,F   多元素选择器，同时匹配所有E元素或F元素，E和F之间用逗号分隔

6.      E F    后代元素选择器，匹配所有属于E元素后代的F元素，E和F之间用空格分隔

7.      E > F         子元素选择器，匹配所有E元素的子元素F

8.      E + F         毗邻元素选择器，匹配所有紧随E元素之后的同级元素F

三、CSS 2.1 属性选择器

9.      E[att]       匹配所有具有att属性的E元素，不考虑它的值。（注意：E在此处可以省略，比如"[cheacked]"。以下同。）

10.   E[att=val]        匹配所有att属性等于"val"的E元素

11.   E[att~=val]      匹配所有att属性具有多个空格分隔的值、其中一个值等于"val"的E元素

12.   E[att|=val]      匹配所有att属性具有多个连字号分隔（hyphen-separated）的值、其中一个值以"val"开头的E元素，主要用于lang属性，比如"en"、"en-us"、"en-gb"等等

四、CSS 2.1中的伪类

13.   E:first-child     匹配父元素的第一个子元素

14.   E:link        匹配所有未被点击的链接

15.   E:visited  匹配所有已被点击的链接

16.   E:active   匹配鼠标已经其上按下、还没有释放的E元素

17.   E:hover   匹配鼠标悬停其上的E元素

18.   E:focus    匹配获得当前焦点的E元素

19.   E:lang(c) 匹配lang属性等于c的E元素

五、 CSS 2.1中的伪元素

20.   E:first-line       匹配E元素的第一行

21.   E:first-letter   匹配E元素的第一个字母

22.   E:before 在E元素之前插入生成的内容

23.   E:after     在E元素之后插入生成的内容

六、CSS 3的同级元素通用选择器

24.   E ~ F         匹配任何在E元素之后的同级F元素

七、CSS 3 属性选择器

25.   E[att^="val"]  属性att的值以"val"开头的元素

26.   E[att$="val"]  属性att的值以"val"结尾的元素

27.   E[att\*="val"]  属性att的值包含"val"字符串的元素

八、CSS 3中与用户界面有关的伪类

28.   E:enabled        匹配表单中激活的元素

29.   E:disabled       匹配表单中禁用的元素

30.   E:checked       匹配表单中被选中的radio（单选框）或checkbox（复选框）元素

31.   E::selection     匹配用户当前选中的元素

九、CSS 3中的结构性伪类

32.   E:root      匹配文档的根元素，对于HTML文档，就是HTML元素

33.   E:nth-child(n)          匹配其父元素的第n个子元素，第一个编号为1

34.   E:nth-last-child(n) 匹配其父元素的倒数第n个子元素，第一个编号为1

35.   E:nth-of-type(n)    与:nth-child()作用类似，但是仅匹配使用同种标签的元素

36.   E:nth-last-of-type(n)    与:nth-last-child() 作用类似，但是仅匹配使用同种标签的元素

37.   E:last-child      匹配父元素的最后一个子元素，等同于:nth-last-child(1)

38.   E:first-of-type         匹配父元素下使用同种标签的第一个子元素，等同于:nth-of-type(1)

39.   E:last-of-type          匹配父元素下使用同种标签的最后一个子元素，等同于:nth-last-of-type(1)

40.   E:only-child     匹配父元素下仅有的一个子元素，等同于:first-child:last-child或 :nth-child(1):nth-last-child(1)

41.   E:only-of-type        匹配父元素下使用同种标签的唯一一个子元素，等同于:first-of-type:last-of-type或 :nth-of-type(1):nth-last-of-type(1)

42.   E:empty  匹配一个不包含任何子元素的元素，注意，文本节点也被看作子元素

十、CSS 3的反选伪类

43.   E:not(s)   匹配不符合当前选择器的任何元素

十一、CSS 3中的 :target 伪类

44.   E:target   匹配文档中特定"id"点击后的效果

## 55.CSS中可以通过哪些属性定义，使得一个DOM元素不显示在浏览器可视范围内？

设置display属性为none，或者设置visibility属性为hidden

设置宽高为0，设置透明度为0，设置z-index位置在-1000em

设置text-indent:-9999px;

## 56.超链接访问过后hover样式就不出现的问题是什么？如何解决？

答案：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）

## 57.什么是Css Hack？ie6,7,8的hack分别是什么？

答案：针对不同的浏览器写不同的CSS code的过程，就是CSS hack。

示例如下：

   #test{

        background-color:yellow;    /\*ie8\*/

        +background-color:pink;        /\*ie7\*/

        \_background-color:orange;       /\*ie6\*/    }

更好的方式是使用IE条件判断语句：

<!–[if lte IE 6]>

内容

<![endif]–>

等

## 58.行内元素和块级元素的具体区别是什么？行内元素的padding和margin可设置吗？

块级元素(block)特性：

总是独占一行，表现为另起一行开始，而且其后的元素也必须另起一行显示;

宽度(width)、高度(height)、内边距(padding)、边框(border)和外边距(margin)都可控制;

内联元素(inline)特性：

和相邻的内联元素在同一行;

宽度(width)、高度(height)、内边距的top/bottom(padding-top/padding-bottom)和外边距的top/bottom(margin-top/margin-bottom)还有border top/bottom都不可改变（也就是padding和margin的left和right是可以设置的），就是里面文字或图片的大小。

浏览器还有默认的天生inline-block元素（拥有内在尺寸，可设置高宽，但不会自动换行）

答案：<input> 、<img> 、<button> 、<texterea> 、<label>。

## 59.什么是外边距重叠？重叠的结果是什么？

外边距重叠就是margin-collapse。

在CSS当中，相邻的两个盒子（可能是兄弟关系也可能是祖先关系）的外边距可以结合成一个单独的外边距。这种合并外边距的方式被称为折叠，并且因而所结合成的外边距称为折叠外边距。

折叠结果遵循下列计算规则：

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。

## 60.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！）

## 61.css中可以让文字在垂直和水平方向上重叠的两个属性是什么？

垂直方向：line-height

水平方向：letter-spacing

那么问题来了，关于letter-spacing的妙用知道有哪些么？

答案:可以用于消除inline-block元素间的换行符空格间隙问题。

## 62如何垂直居中一个浮动元素？

         // 方法一：已知元素的高宽

#div1{

    background-color:#6699FF;

    width:200px;

    height:200px;

    position: absolute;        /\*父元素需要相对定位\*/

    top: 50%;

    left: 50%;

    margin-top:-100px ;   /\*二分之一的height，width\*/

    margin-left: -100px;

    }

//方法二:未知元素的高宽

  #div1{

    width: 200px;

    height: 200px;

    background-color: #6699FF;

    margin:auto;

    position: absolute;        /\*父元素需要相对定位\*/

    left: 0;

    top: 0;

    right: 0;

    bottom: 0;

}

## 63.如何垂直居中一个<img>?

（用更简便的方法。）

         /\*<img>的容器设置如下\*/

#container {

                   display:table-cell;

                   text-align:center;

                   vertical-align:middle;}

## 64.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

## 65.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。

## 66.display:none与visibility:hidden的区别是什么？

display : 隐藏对应的元素但不挤占该元素原来的空间。

visibility: 隐藏对应的元素并且挤占该元素原来的空间。

即是，使用CSS display:none属性后，HTML元素（对象）的宽度、高度等各种属性值都将“丢失”;而使用visibility:hidden属性后，HTML元素（对象）仅仅是在视觉上看不见（完全透明），而它所占据的空间位置仍然存在。

## 67.简介盒子模型：

CSS的盒子模型有两种：IE盒子模型、标准的W3C盒子模型模型

盒模型：内容、内边距、外边距（一般不计入盒子实际宽度）、边框

## 68.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。

## 69.BFC是什么？

BFC就是“块级格式化上下文”的意思，创建了 BFC的元素就是一个独立的盒子，不过只有Block-level box可以参与创建BFC， 它规定了内部的Block-level Box如何布局，并且与这个独立盒子里的布局不受外部影响，当然它也不会影响到外面的元素。

BFC有一下特性：

内部的Box会在垂直方向，从顶部开始一个接一个地放置。

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会发生叠加

每个元素的margin box的左边， 与包含块border box的左边相接触(对于从左往右的格式化，否则相反)。即使存在浮动也是如此。

BFC的区域不会与float box叠加。

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，反之亦然。

计算BFC的高度时，浮动元素也参与计算。

如何触发BFC？

float 除了none以外的值

overflow 除了visible 以外的值（hidden，auto，scroll ）

display (table-cell，table-caption，inline-block, flex, inline-flex)

position值为（absolute，fixed）

fieldset元素

## 70.Doctype的作用？严格模式与混杂模式的区别？

<!DOCTYPE>,文档声明；用于告知浏览器该以何种模式来渲染文档

严格模式下：页面排版及JS解析是以该浏览器支持的最高标准来执行

混杂模式：不严格按照标准执行，主要用来兼容旧的浏览器，向后兼容

## 71.IE的双边距BUG：块级元素float后设置横向margin，ie6显示的margin比设置的较大。

解决：加入\_display：inline

## 72.HTML与XHTML——二者有什么区别？

1. 所有的标记都必须要有一个相应的结束标记

2. 所有标签的元素和属性的名字都必须使用小写

3. 所有的 XML 标记都必须合理嵌套

4. 所有的属性必须用引号 "" 括起来

5. 把所有 < 和 & 特殊符号用编码表示

6. 给所有属性赋一个值

7. 不要在注释内容中使用 "--"

8. 图片必须有说明文字

## 73.html常见兼容性问题？

1.双边距BUG float引起的  使用display

2.3像素问题 使用float引起的 使用dislpay:inline -3px

3.超链接hover 点击后失效  使用正确的书写顺序 link visited hover active

4.IE z-index问题 给父级添加position:relative

5.Png 透明 使用js代码 改

6.Min-height 最小高度 ！Important 解决’

7.select 在ie6下遮盖 使用iframe嵌套

8.为什么没有办法定义1px左右的宽度容器（IE6默认的行高造成的，使用over:hidden,zoom:0.08 line-height:1px）

9.IE5-8不支持opacity，解决办法：

.opacity {

    opacity: 0.4

    filter: alpha(opacity=60); /\* for IE5-7 \*/

    -ms-filter: "progid:DXImageTransform.Microsoft.Alpha(Opacity=60)"; /\* for IE 8\*/

}

10. IE6不支持PNG透明背景，解决办法: IE6下使用gif图片

## 74.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。

## 75.行内元素有哪些?块级元素有哪些?

答：块级元素：div p h1 h2 h3 h4 form ul

行内元素: a b br i span input select

## 76.前端页面有哪三层构成，分别是什么?作用是什么?

答：结构层 Html 表示层 CSS 行为层 js。

## 77.Doctype作用? 严格模式与混杂模式-如何触发这两种模式，区分它们有何意义?

（1）、<!DOCTYPE> 声明位于文档中的最前面，处于 <html> 标签之前。告知浏览器的解析器，用什么文档类型 规范来解析这个文档。

（2）、严格模式的排版和 JS 运作模式是,以该浏览器支持的最高标准运行。

（3）、在混杂模式中，页面以宽松的向后兼容的方式显示。模拟老式浏览器的行为以防止站点无法工作。

（4）、DOCTYPE不存在或格式不正确会导致文档以混杂模式呈现。

## 78.列出display的值，说明他们的作用。position的值， relative和absolute定位原点是？

  1. block 象块类型元素一样显示。

  none 缺省值。向行内元素类型一样显示。

  inline-block 象行内元素一样显示，但其内容象块类型元素一样显示。

  list-item 象块类型元素一样显示，并添加样式列表标记。

  2. position的值

  \*absolute

        生成绝对定位的元素，相对于 static 定位以外的第一个父元素进行定位。

  \*fixed （老IE不支持）

        生成绝对定位的元素，相对于浏览器窗口进行定位。

  \* relative

        生成相对定位的元素，相对于其正常位置进行定位。

  \* static  默认值。没有定位，元素出现在正常的流中

  \*（忽略 top, bottom, left, right z-index 声明）。

  \* inherit 规定从父元素继承 position 属性的值。

## 79.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.

## 80.css的基本语句构成是?

选择器{属性1:值1;属性2:值2;……}

## 81.CSS中可以通过哪些属性定义，使得一个DOM元素不显示在浏览器可视范围内？

最基本的：

设置display属性为none，或者设置visibility属性为hidden

技巧性：

设置宽高为0，设置透明度为0，设置z-index位置在-1000

## 82.行内元素和块级元素的具体区别是什么？行内元素的padding和margin可设置吗？

块级元素(block)特性：

总是独占一行，表现为另起一行开始，而且其后的元素也必须另起一行显示;

宽度(width)、高度(height)、内边距(padding)和外边距(margin)都可控制;

内联元素(inline)特性：

和相邻的内联元素在同一行;

宽度(width)、高度(height)、内边距的top/bottom(padding-top/padding-bottom)和外边距的top/bottom(margin-top/margin-bottom)都不可改变（也就是padding和margin的left和right是可以设置的），就是里面文字或图片的大小。

那么问题来了，浏览器还有默认的天生inline-block元素（拥有内在尺寸，可设置高宽，但不会自动换行），有哪些？

答案：<input> 、<img> 、<button> 、<textarea> 、<label>

## 83.什么是外边距重叠？重叠的结果是什么？

外边距重叠就是margin-collapse。

在CSS当中，相邻的两个盒子（可能是兄弟关系也可能是祖先关系）的外边距可以结合成一个单独的外边距。这种合并外边距的方式被称为折叠，并且因而所结合成的外边距称为折叠外边距。

折叠结果遵循下列计算规则：

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。

## 84.说display属性有哪些？可以做什么？

display:block行内元素转换为块级元素

display:inline块级元素转换为行内元素

display:inline-block转为内联元素

## 85.哪些css属性可以继承？

可继承： font-size  font-family  color

不可继承 ：border  padding  margin  width  height ;

## 86.css优先级算法如何计算？

!important >  id > class > 标签

!important 比 内联优先级高

\*优先级就近原则，样式定义最近者为准;

\*以最后载入的样式为准;

## 87.b标签和strong标签,i标签和em标签的区别？

后者有语义，前者则无。

## 88.有那些行内元素、有哪些块级元素、盒模型？

1.内联元素(inline element)

a – 锚点

abbr – 缩写

acronym – 首字

b – 粗体(不推荐)

big – 大字体

br – 换行

em – 强调

font – 字体设定(不推荐)

i – 斜体

img – 图片

input – 输入框

label – 表格标签

s – 中划线(不推荐)

select – 项目选择

small – 小字体文本

span – 常用内联容器，定义文本内区块

strike – 中划线

strong – 粗体强调

sub – 下标

sup – 上标

textarea – 多行文本输入框

tt – 电传文本

u – 下划线

var – 定义变量

2、块级元素

address – 地址

blockquote – 块引用

center – 举中对齐块

dir – 目录列表

div – 常用块级容易，也是css layout的主要标签

dl – 定义列表

fieldset – form控制组

form – 交互表单

h1 – 大标题

h2 – 副标题

h3 – 3级标题

h4 – 4级标题

h5 – 5级标题

h6 – 6级标题

hr – 水平分隔线

isindex – input prompt

menu – 菜单列表

noframes – frames可选内容，（对于不支持frame的浏览器显示此区块内容）

noscript – ）可选脚本内容（对于不支持script的浏览器显示此内容）

ol – 排序表单

p – 段落

pre – 格式化文本

table – 表格

ul – 非排序列表

3.CSS盒子模型包含四个部分组成：

内容、填充（padding）、边框（border）、外边界（margin）

## 89.有哪些选择符，优先级的计算公式是什么？行内样式和！important哪个优先级高？

#ID > .class > 标签选择符  !important优先级高

## 90.我想让行内元素跟上面的元素距离10px，加margin-top和padding-top可以吗？

margin-top,padding-top无效