1.如何理解标签语义化？

(1)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;

(2)即使在没有样式 CSS 的情况下也能以一种文档格式显示，并且是容易阅读的;

(3)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，有利于 SEO;

(4)使阅读源代码的人更容易将网站分块，便于阅读、维护和理解。

2.HTML5新增标签及使用？

1、结构标签

（1）section：独立内容区块，可以用h1~h6组成大纲，表示文档结构，也可以有章节、页眉、页脚或页眉的其他部分；

（2）article：特殊独立区块，表示这篇页眉中的核心内容；

（3）aside：标签内容之外与标签内容相关的辅助信息；

（4）header：某个区块的头部信息/标题；

（5）hgroup：头部信息/标题的补充内容；

（6）footer：底部信息；

（7）nav：导航条部分信息

（8）figure：独立的单元，例如某个有图片与内容的新闻块。

2、表单标签

（1）email：必须输入邮件；

（2）url：必须输入url地址；

（3）number：必须输入数值；

（4）range：必须输入一定范围内的数值；

（5）Date Pickers：日期选择器；

a.date：选取日、月、年

b.month：选取月、年

c.week：选取周和年

d.time：选取时间（小时和分钟）

e.datetime：选取时间、日、月、年（UTC时间）

f.datetime-local：选取时间、日、月、年（本地时间）

（6）search：搜索常规的文本域；

（7）color：颜色

3、媒体标签

（1）video：视频

（2）audio：音频

（3）embed：嵌入内容（包括各种媒体），Midi、Wav、AU、MP3、Flash、AIFF等。

4、其他功能标签

（1）mark：标注（像荧光笔做笔记）

（2）progress：进度条；<progress max="最大进度条的值" value="当前进度条的值

（3）time：数据标签，给搜索引擎使用；发布日期<time datetime="2014-12-25T09:00">9：00</time>更新日期<time datetime="2015- 01-23T04:00" pubdate>4:00</time>

（4）ruby和rt：对某一个字进行注释；<ruby><rt>注释内容</rt><rp>浏览器不支持时如何显示</rp></ruby>

（5）wbr：软换行，页面宽度到需要换行时换行；

（6）canvas：使用JS代码做内容进行图像绘制；

（7）command：按钮；

（8）deteils ：展开菜单；

（9）dateilst：文本域下拉提示；

（10）keygen:加密；

3.如何创建锚点？

锚点是文档中某行的一个记号，类似于书签，用于链接到文档中的某个位置。

锚点标签：带交互的标签，可以实现跳转

target属性：用来跳转后页面的打开方式

\_self:在当前窗口打开

\_black:在新窗口打开-->

<!-- 跳转到万维网 -->

<!-- 浏览器的本质 文档的查看器 -->

<a href="http://www.baidu.com">百度一下，你就知道。</a>

<!-- 跳转到其他文档 -->

<a href="1.周报-.html">1.周报</a>

<a href="IMG\_20190723\_183912.jpg">图片</a>

<!-- 跳转到当前文档的指定位置 -->

1. .通过在文档的特定位置标签上设置id属性，在a标签的href属性中填入该id属性值，在点击时， 整个文档就会发生移动，将该行显示在浏览器可视区域中。

<a href="#top">回到顶部</a>

<!-- <hi id="top" --></hi> -->

应用场景：回到顶部，目录列表。

4.标签分类，以及各自特点？

块级标签的特点: block

(1).标签独占一行,可以让超文本从上到下排列.

(2).支持所有的css样式设置

(3).垂直方向上的margin会重叠

(4).在不设置宽度的时候,width大小为自动的情况下,块级标签盒子在界面上

默认占满父级内容宽度.此时设置 水平方向的margin border padding时,都会向内压缩内容空间(width空间),而不会向外扩.

(5).如果设置了盒子宽度,盒子移动,符合margin padding基本移动规律

(6).外边距坍塌问题,只出现在父级和子级都是块级标签的情况下

(7).块级标签的居中问题,块级标签要在其父级的水平方向居中,是通过给标签自身设置margin 上下 auto来完成.

(8).最常见的块级标签:div

内块级特点: Inline-block

(1).可以让盒子同行排列

(2).支持所有的CSS样式设置

(3).间隙问题:

在浏览器上同行排列的盒子,会将文档中对应标签之间的空白符

解析成一个间隙,并且该间隙大小不固定(不同浏览器,同一浏览器不同版本,相同浏览器在不同系统平台上,该间隙大小不一致,具有危害性,需要消除.

消除方法:为同行排列盒子的父级设置font-size:0; 同时由于font-size:具有继承性,因此需要给予子级恢复字体大小.)

(4).对齐问题:默认情况下,同行排列的盒子,是按照盒子内部最后一行文字的底部对齐,

当文字行数不一致时,会导致不对齐.通过vertical-align属性可以设置,盒子的

对齐基线.一般情况常用三个值 top顶对齐,middle居中对齐 bottom底对齐.

在使用时,一般情况下要给同行的盒子同时设置相同的对齐方式.

(5).居中问题:通过给父级设置text-align:center来实现

(6).在不设置宽高大小的时候,由内容撑开宽度高度.

inline型标签,只用来放文字

(1).不支持宽高设置

(2).不支持上下margin设置,因此inline型标签需要从上到下移动,只能通过其他盒子推着它走,或者父级盒子带着他走.

(3).水平方向上的margin有效,并且相互叠加

(4).其盒子大小只能由内部的文字大小撑开

(5).最常用的inline型标签 span

(6).同样具有间隙问题,解决方法与inline-block一致

(7).水平居中:给父级设置text-aline:center

1. 三种定位的使用？

相对定位：positive：relative

（1）盒子特性与默认文本流特性保持一致

（2）改变盒子位置，可以通过

top：盒子上边到默认文本流盒子位置上边之间的距离。

right：盒子右边到默认文本流盒子位置右边之间的距离。

bottom：盒子下边到默认文本流盒子位置下边之间的距离。

left：盒子左边到默认文本流盒子位置左边之间的距离。

（3）盒子在位置改变过程当中，对默认文本流的布局之间没有影响。在默认文本流之上，比默认文本流高半层。

（4）应用场景：

a:在默认文本流布局以后，需要对某些盒子进行微调整，并且这种微调不希望对其他默认文本流盒子产生影响的情况下，使用相对定位来完成。

b.作为绝对定位盒子的定位父级标志。

固定定位：position:fixed

（1）盒子特性：盒子位置属性设置成fixed,脱离默认文本流，进入到新的一层，不在具备默认文本流下的盒子特性，而是呈现新的特点，支持所有css，不设置宽高时，靠内容撑开。盒子将固定在浏览器的指定位置，不随页面滚动。

（2）移动盒子，将通过top,left,bottom,right

top:指盒子的上边到浏览器的可视区域上边之间的距离。

right:指盒子的右边到浏览器的可视区域右边之间的距离。

bottom:指盒子的下边到浏览器的可视区域下边之间的距离。

left:指盒子的左边到浏览器的可视区域左边之间的距离。

（3）盒子移动过程中，互相没有任何影响，当盒子位置重叠时，可以通过z-index;来设置盒子显示层级的优先级。z-index的值是数字，没有单位，值越大对应的盒子越靠前。

（4）用在浏览器中，不随页面滚动的布局上，常见的案例：

固定的导航，固定的广告悬浮框，固定的客服窗口，登录居中弹窗，后台管理系统布局。

（5）定位的四件套：

position：属性

定位操作：top left

设置显示层级关系 z-index 缺一不可！

绝对定位：position: absolute

盒子设置绝度定位以后，脱离原来文本流，盖在其他默认文本流之上，盒子不再有默认文本流的盒子特性，而是具有新的特点。支持所有的css样式，在不设置大小时，靠内容撑开。

绝对定位，可以让盒子按照某一个指定的父级盒子边界进行位置改变通过：

top left right bottom四个值

top：盒子的上边界到定位父级上边界之间的距离。

right：盒子的右边界到定位父级右边界之间的距离。

bottom：盒子的下边界到定位父级下边界之间的距离。

left：盒子的左边界到定位父级左边界之间的距离。

绝对定位盒子之间进行位置改变，互相没有影响，当位置重叠时，可通过z-index 来设置该盒子的显示层级。 值越大，显示越靠前。

1. display和visibility属性的作用与区别？

**visibility和display两个属性都有隐藏元素的功能。**

将元素display属性设为 none，隐藏该元素内容，且不占用域的空间。

将元素visibility属性设为 hidden，隐藏该元素内容，但占用域的空间。

1. 简述盒模型？怪异盒模型？

盒模型基本结构;

内容区域: (content) width height：

所有写在标签中的内容,只出现在这个区域.

内边距区域:padding设置：

设置的时一个盒子的边框到内容区域之间的距离可以扩大盒子.边界区域:是由boder所设置的区域,边框.外边框区域:margin设置。

怪异盒模型：

box-sizing:border-box

盒模型变异为怪异盒模型，此时width，height

设置的不再是内容区域大小，而是盒子的尺寸。

盒子内容区域大小是通过计算来完成的，padding，border都会向内挤压。margin不受影响。

盒子内容区域宽度=width-左右border-左右padding

盒子内容区域高度=height-上下border-上下padding

应用场景：现阶段推荐仅仅在移动端使用。PC端IE9以下不支持。

1. 如何设置一个div水平居中？如何设置一个浮动元素水平居中？

水平居中一个div：

给div设置一个宽度，margin:0px auto。

position:absolute left:50%

弹性盒居中 justify-content:center

父元素设置 text-align:center子级居中

水平居中一个浮动元素：

#### 在浮动元素外面再嵌套一层div（套一个大盒子给它margin:0px auto。）

浮动元素水平居中-----宽度不固定的浮动元素

父元素和子元素同时左浮动，然后父元素相对左移动50%，再然后子元素相对右移动50%，或者子元素相对左移动-50%也就可以了。

9.简述精灵图（雪碧图、css sprites）？

通过将网页中固定不变的小图标，小图片整合到一张大图上，

将多次图片资源请求降低成一次图片资源请求，以达到页面加载速率

优化的目的。

在本地通过背景图定位来完后精灵图使用。

(1).设置盒子大小与要用的小图标保持一致

(2).设置背景图属性

(3).进行背景图定位，来让需要的小图片显示出来。

**缺点:**

1. .开发的时候比较麻烦，你要通过photoshop(PS)或其他工具测量计算每一个背景单元的精确位置，这是针线活，没什么难度，但是很繁琐。你需要考虑当前盒子会不会漏出其他的背景图，比较头疼的是，页面自适应时，背景图的位置就没那么容易掌控了。
2. .在维护的时候比较麻烦，因为他是多张图在一张图上，牵一发而动全身的感觉。轻易不要改变其他图片的位置，原位能放下就在原位改，放不下就在下面添加。

10.过渡与动画的使用和区别？

animation属性类似于transition，他们都是随着时间改变元素的属性值，

其主要区别在于：

transition需要触发一个事件才会随着时间改变其CSS属性；

animation在不需要触发任何事件的情况下，也可以显式的随时间变化来改变元素CSS属性，达到一种动画的效果

1）动画不需要事件触发，过渡需要。

2）过渡只有一组（两个：开始-结束） 关键帧，动画可以设置多个。

11.说说浮动？

浮动布局解决问题：

浮动解决的是布局中，盒子同行排列的问题，可以快速让所有盒子同行排列，并且顶对齐没有间隙。

浮动的原理：

盒子在当前行浮起来，进入到新的一层（浮动层） 按照设定的水平方向移动。直到遇到父级块的边缘，或者其他浮动块边缘停止，不能设置具体浮动多远，只能设定方向。

盒子浮动的先后顺序： 在文档中，先写的标签先浮动

浮动换行：

当一行没有足够的空间存放浮动块时，浮动块会在当前位置，垂直向下移动到能够存放的位置。

然后沿着设定的方向接着移动，直到遇到父级块的边缘，或者其他浮动块的边缘停止。

利用浮动换行可以简化布局的标签结构

浮动坍塌是指：

当父级不设置高度，靠内容撑开时，所有的子级都发生浮动，从而导致父级高度丢失的现象，称为浮动坍塌现象。

浮动坍塌会造成父级所在文本流的布局紊乱，因此需要将浮动坍塌问题消除。

而消除浮动坍塌问题的方法称为： 清浮动、

清浮动方式：

(1).给父级设置固定高，这种方式不灵活，主要用在布局完成后的微调场景，或者浏览器兼容配置场景。

(2).让父级也浮动起来，不主动设置，只做被动判断，看父级是否需要清浮动。

实际应用时，当需要给子级浮动时，看看父级是否已经浮动了，如果父级已经浮动，便不再清浮动。

(3).给父级设置 overflow：hidden；

(4).给父级设置 display：inline-block；

3/4是在父级区域形成一个BFC（块级格式化上下文）环境。这种环境可以保证子级浮动后，依然能够撑开父级高度。

应用方式：不主动设置，只做被动判断。

(5).利用clear：both 主动为父级清浮动

原理：在父级盒子内部最后一个子级位置上，添加一个block（div）标签，完成清浮动。

给该标签设置clear：lift属性的意思是：

在设置该属性的盒子3维空间上方，不能够设置了float：left；的盒子出现。由于浮动块不能自己向下移动，

只能够靠设置了clear属性的盒子，自己向下移动，躲开浮动块，移动过程中，将父级高度拉开，实现父级高度的恢复。

缺点：代码编写结构复杂，不容易维护。

(6).利用after伪元素清浮动功能上有。标签结构上没有伪元素，但高度坍塌的父级选择器：：after{

content：“”；

display:block;

clear:both;

}

实际应用时会在样式初始化时定义一个清浮动选择器:

.clear::after{

content：“”；

display:block;

clear:both;

}

哪里需要清浮动，就给其添加 class clear

eg：<div class box1 clear>\*\*\* </div>

1/2/3/4:方法仅仅用于是否需要清浮动。

当给父级设置了固定高;设置了浮动;设置了display-inline block;overflow：hidden。中的任一一种时，其所有子级浮动，都不会造成父级高度坍塌，不需要清浮动。

1. 兼容css3新属性各个浏览器前缀？

前缀　　　　　　　　 浏览器

-webkit-　　　　　　chrome、safari

-moz-　　　　　　　 firefox

-ms-         IE

-o-           opera

13.margin和padding的使用？

margin用来移动盒子.

margin只能向右 向下移动盒子

嵌套情况下子集盒子的margin特点:

(1).移动方式与平级盒子一样.

(2).移动的起点是从父级内容区域开始的,与父级的padding,border没有关系.

(3).外边距坍塌问题

在垂直方向上,当父级没有明确的边界时

border-top

padding-top

border-bottom

padding-bottom

父级中的第一个子集的margin-top会传递给父级,导致父级跟子级一起向下移动.

父级中的最后一个子级的margin-bottom会传递给父级,导致无法撑开父级高度.

外边界坍塌问题出现在 Chrome Firefox Safari IE9以上

解决方式:给父级添加边界.

padding内边距是用来改变内容位置由于默认布局方向上是从上到下,从左到右的. 垂直方向上;

margin和padding都可以改变超文本位置

margin是改变盒子位置,让盒子带着超文本移动.

padding是通过改变盒子边界到盒子内容之间的距离,来推着超文本移动.

使用的时候根据效果图来判断使用哪个属性进行位置改变,当没有明确边界渲染时,margin padding都可以使用. 当明确边界渲染时,如果盒子之间有间隙,使用margin;如果盒子之间没有间隙,使用padding.

14.媒体查询？移动端viewport？

**媒体查询本质：**

就是监测当前文档运行在哪个媒介载体上，或者同一媒介载体的不同状态上，然后重新给文档中的标签设置样式。

引入媒体查询有三种样式：

(1).在样式表中引入媒体查询：写在所有样式之后

style

其他样式

@midia 条件{

样式列表

}

</style>

(2).在style标签中引入媒体查询：

<style midia="条件1">样式表1</style>

<style midia="条件2">样式表2</style>

根据不同的媒体查询结果，执行不同的内部样式表。

(3).在link标签中引入媒体查询：外部css

<link ref="stylesheet" href="样式1.css" media="条件1"> </link>

<link ref="stylesheet" href="样式2.css" media="条件2"> </link>

根据不同的媒体查询结果，引入不同的外部样式表.

媒体查询本质：

就是监测不同屏幕宽度，在这一宽度下，设置不同的盒子样式，盒子尺寸优先考虑流式布局。

比较常见的样式修改：

1.display 盒子的出现和消失

2.排列方式 (盒子宽度 调整)

3.字体大小调整（用rem）

媒体查询监测屏幕尺寸：

样式表内:

@media screen and (条件1) and （条件2）and （条件n）{

样式列表

}

min-width: 尺寸1：最小宽度，表示尺寸1<

max-width: 尺寸2: 最大宽度，表示尺寸2>

min-width: 表示的是浏览器的尺寸大小

min-device-width: 表示的是设备的尺寸大小

当文档中 <meta name="viewport" content="width=device-width, initial-scale=1.0">

min-width 和 min-device-width 是等效的

**viewport：**

(1).概念：是手机浏览器上虚拟的一层，网页在手机上打开时都是显示在这一层。

(2).作用：对手机屏幕进行分辨率设置。 仅仅限于浏览器打开时候。

与手机的物理分辨率，标称的分辨率无关。

(3). 默认情况下，viewpoint在不设置时，将大部分的手机，平板设置为980px。

(4).在HTML文档中可以通过<meta name="viewport" content="">来精准控制

viewpoint的显示特征。

(5). width=分辨率值 (viewpoint最重要的属性，用来配置分辨率默认980)

user-scalable=no,配置用户双指缩放权限（受限于浏览器的设置）

initial-scale=1.0,网页初始化缩放比例，默认1倍（不缩放）

minimum-scale=1.0,网页最小缩放比例，默认1倍（不缩放）

maximum-scale=1.0,网页最大缩放比例，默认1倍（不缩放）

<meta name="viewport" content="width=device-width, initial-scale=1.0">