**HTML**

**1. Doctype作用，HTML5 为什么只需要写**

doctype是一种标准通用标记语言的文档类型声明，目的是告诉标准通用标记语言解析器要使用什么样的文档类型定义（DTD）来解析文档.<!DOCTYPE>声明必须是HTML文档的第一行，位于html标签之前

HTML5不基于SGML，所以不需要引用DTD。在HTML5中<!DOCTYPE>只有一种

SGML: 标准通用标记语言,是现时常用的超文本格式的最高层次标准

**2. 行内元素有哪些，块级元素有哪些，空(void)元素有那些**

行内元素：a span i img input select b 等

块级元素：div ul ol li h1~h6 p table 等

空元素：br hr link 等

**3. 简述一下你对HTML语义化的理解**

简单来说，就是合适的标签做合适的事情，这样具有以下好处：

* 有助于构架良好的HTML结构，有利于搜索引擎的建立索引、抓取，利于SEO
* 有利于不同设备的解析
* 有利于构建清晰的机构，有利于团队的开发、维护

**4. 常见的浏览器内核有哪些，介绍一下你对浏览器内核的理解**

Trident内核：IE

Gecko内核：NETSCAPE6及以上版本，火狐

Presto内核：Opera7及以上。[Opera内核原为：Presto，现为：Blink;]

Webkit内核：Safari，Chrome等。[Chrome的：Blink（WebKit的分支）]

浏览器内核又可以分成两部分：渲染引擎和JS引擎。 渲染引擎主要负责取得网页的内容、整理讯息、计算网页的显示方式等，JS引擎则是解析Javascript语言，执行javascript语言来实现网页的动态效果。

**5. html5有哪些新特性**

* 语义化标签: header footer nav section article aside 等
* 增强型表单：date(从一个日期选择器选择一个日期) email(包含 e-mail 地址的输入域) number(数值的输入域) range(一定范围内数字值的输入域) search(用于搜索域) tel(定义输入电话号码字段) 等
* 视频和音频：audio video
* Canvas绘图 SVG绘图
* 地理定位：Geolocation
* 拖放API：drag
* web worker：是运行在后台的 JavaScript，独立于其他脚本，不会影响页面的性能
* web storage: localStorage sessionStorage
* WebSocket: HTML5开始提供的一种在单个 TCP 连接上进行全双工通讯的协议

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 特性 | | Cookie | localStorage | | sessionStorage | |
| 生命周期 | | 可设置失效时间，没有设置的话，默认是关闭浏览器后失效 | 除非被手动清除，否则将会永久保存 | | 仅在当前网页会话下有效，关闭页面或浏览器后就会被清除 | |
| 存放数据大小 | | 4KB左右 | 可以保存5MB的信息 | | | |
| http请求 | | 每次都会携带在HTTP头中，如果使用cookie保存过多数据会带来性能问题 | 仅在客户端（即浏览器）中保存，不参与和服务器的通信 | | | |
|  |  | | |  | |  | |

**6. 描述一下 cookie，sessionStorage 和 localStorage 的区别**

**7. 如何实现浏览器内多个标签页之间的通信**[多个标签页之间的通信](https://juejin.im/post/5acdba01f265da23826e5633)

* 使用localStorage: localStorage.setItem(key,value)、localStorage.getItem(key)
* websocket协议
* webworker

**8. HTML5的离线存储怎么使用，解释一下工作原理**[HTML5的离线存储](https://segmentfault.com/a/1190000006984353)

HTML5的离线存储是基于一个新建的.appcache文件的，通过这个文件上的解析清单离线存储资源，这些资源就会像cookie一样被存储了下来。之后当网络在处于离线状态下时，浏览器会通过被离线存储的数据进行页面展示。

就像cookie一样，html5的离线存储也需要服务器环境。

**解析清单**

在开始之前要先了解下manifest（即.appcache文件），上面的解析清单要怎么写。manifest 文件是简单的文本文件，它告知浏览器被缓存的内容（以及不缓存的内容）。manifest 文件可分为三个部分：CACHE MANIFEST - 在此标题下列出的文件将在首次下载后进行缓存NETWORK - 在此标题下列出的文件需要与服务器的连接，且不会被缓存FALLBACK - 在此标题下列出的文件规定当页面无法访问时的回退页面（比如 404 页面）在线的情况下,用户代理每次访问页面，都会去读一次manifest.如果发现其改变, 则重新加载全部清单中的资源。

**9. src与href的区别**

区别：src用于替代这个元素，而href用于建立这个标签与外部资源之间的关系

<link href="style.css" rel="stylesheet" />浏览器加载到这里的时候，html的渲染和解析不会暂停，css文件的加载是同时进行的

<script src="script.js"></script>当浏览器解析到这句代码时，页面的加载和解析都会暂停直到浏览器拿到并执行完这个js文件

**10. 表单提交中Get和Post方式的区别**

* Get一般用于从服务器上获取数据，Post向服务器传送数据
* Get传输的数据是拼接在Url之后的，对用户是可见的；Post的传输数据对用户是不可见的
* Get传送的数据量较小，不能大于2KB。Post传送的数据量较大，一般被默认为不受限制
* Get安全性非常低，Post安全性较高
* 在FORM提交的时候，如果不指定Method，则默认为Get请求

**CSS**

**1. css盒子模型，box-sizing属性的理解**

css的盒模型由content(内容)、padding(内边距)、border(边框)、margin(外边距)组成。但盒子的大小由content+padding+border这几部分决定

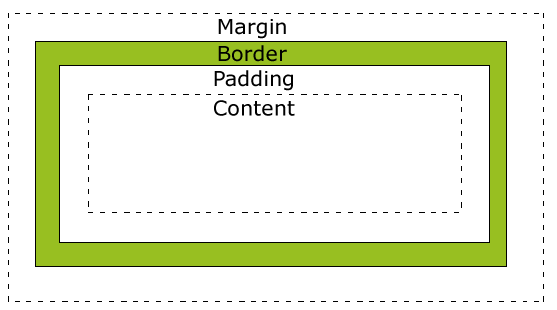
box-sizing是一个CSS3属性，与盒子模型有着密切联系。即决定元素的宽高如何计算，box-sizing有三个属性：

box-sizing: content-box|border-box|inherit:

* content-box 使得元素的宽高即为内容区的宽高(默认模式)
* border-box: 计算方式content + padding + border = 本身元素大小，即缩小了content大小
* inherit 指定box-sizing属性的值，应该从父元素继承

盒模型，以及这几种类型之间的区别；

W3C标准盒子模型和IE盒子模型，W3C盒子模型——属性高（height）和属性宽（width）这两个值不包含填充（padding）和边框（border）,而IE盒子模型中包含



一个盒子的 margin 为 20px，border 为 2px，padding 为 10px，content 的宽为 200px、高为 50px

ie盒子模型

盒子所占空间：width=20ｘ2+200=240 　　 height=20ｘ2+50=90

盒子实际大小：width=200 　　　　　　　 height=50

标准w3c盒子模型（盒子实际大小width+padding\*2+border\*2）

盒子所占空间：width=20ｘ2+2ｘ2+10ｘ2+200=264 　 height=20ｘ2+２ｘ2+10ｘ2 +50=114

盒子实际大小：width=200 +2ｘ2+10ｘ2 =224 　　　 height=50+2ｘ2+10ｘ2=74

CSS3中的box-sizing属性

1、content-box:使元素遵循标准 w3c 盒子模型（默认值）。

2、border-box:使元素遵循ie 盒子模型。

3、 inherit： 规定应从父元素继承 box-sizing 属性的值。

html的文件头部是不是有<!DOCTYPE html>，就说明遵循标准w3c

CSS3 规范中引入的新布局模型：弹性盒模型（flex box）

justify-content：center； 居中对齐

align-items: center； 垂直居中

align-content: center； 与交叉轴中点对齐

**2. 清除浮动，什么时候需要清除浮动，清除浮动都有哪些方法**

浮动的元素是脱离文档标准流的，如果我们不清楚浮动，那么就会造成父元素高度塌陷，影响页面布局。

清除浮动的方式：

* 为父元素设置高度
* 为父元素添加overflow:hidden
* 伪元素

.fix::after {

content:"";

display:block;

clear:both;

}

使用伪元素的好处：不增加冗余的DOM节点，符合语义化

overflow:hidden可以触发BFC机制。BFC：块级格式化上下文，创建了 BFC的元素就是一个独立的盒子，它规定了内部如何布局，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，计算BFC的高度时，浮动元素也参与计算

**3. 如何让一个不定宽高的盒子水平垂直居中**

定位的方式

.father {

position: relative;

}

.son {

position: absolute;

top: 0;

right: 0;

bottom: 0;

left: 0;

margin: auto;

}

css3属性

.father {

position: relative;

}

.son {

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

}

flex布局

.father {

display: flex;

justify-content: center;

align-items: center;

}

**4. px和em和rem的区别**

px: 像素，相对长度单位。像素px是相对于显示器屏幕分辨率而言的

em的值并不是固定的，会继承父级元素的字体大小，代表倍数

rem的值并不是固定的，始终是基于根元素 <html> 的，也代表倍数

rpx小程序中参考

**5. position的值有哪些**

static： 默认值。没有定位，元素出现在正常的流中

relative（相对定位）：生成相对定位的元素,相对于其正常（原先本身）位置进行定位

absolute（绝对定位）：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（固定定位）：生成绝对定位的元素，相对于浏览器窗口进行定位

**6. display:none与visibility:hidden的区别**

| 区别 | display:none | visibility：hidden的 |
| --- | --- | --- |
| 是否占据空间 | 不占据任何空间，在文档渲染时，该元素如同不存在（但依然存在文档对象模型树中） | 该元素空间依旧存在 |
| 是否渲染 | 会触发reflow（回流），进行渲染 | 只会触发repaint（重绘），因为没有发现位置变化，不进行渲染 |
| 是否是继承属性 | 不是继承属性，元素及其子元素都会消失 | 是继承属性，若子元素使用了visibility:visible，则不继承，这个子孙元素又会显现出 |

**7. CSS中link 和@import的区别**

link属于XHTML标签，@import完全是CSS提供的一种方式,只能加载CSS

加载顺序的差别，当一个页面被加载的时候，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

兼容性的差别。由于@import是CSS2.1提出的所以老的浏览器不支持，而link标签无此问题

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

**8. 什么是响应式设计，响应式设计的基本原理是什么**

响应式网站设计是一个网站能够兼容多个终端，而不是为每一个终端做一个特定的版本。基本原理是通过媒体查询检测不同的设备屏幕尺寸做处理

**9. 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

初始化样式会对 SEO 有一定的影响

**10. CSS3有哪些新特性**

* 实现圆角border-radius，阴影box-shadow，边框图片border-image
* 对文字加特效text-shadow，强制文本换行word-wrap，线性渐变linear-gradient
* 实现旋转transform:rotate(90deg),缩放scale(0.85,0.90),translate(0px,-30px)定位,倾斜skew(-9deg,0deg);
* 增加了更多的CSS选择器、多背景、rgba()
* 唯一引入的伪元素是::selection；
* 实现媒体查询@media，多栏布局flex
* 过渡transition 动画animation

**11. ::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

**12. CSS优化、提高性能的方法有哪些**

* 移除空的css规则（Remove empty rules）
* 正确使用display的属性
* 不滥用浮动、web字体
* 不声明过多的font-size
* 不在选择符中使用ID标识符
* 遵守盒模型规则
* 尽量减少页面重排、重绘
* 抽象提取公共样式，减少代码量

**13. 重绘和回流**

[重绘和回流](https://juejin.im/post/5a9923e9518825558251c96a)：回流必将引起重绘，重绘不一定会引起回流

当Render Tree中部分或全部元素的尺寸、结构、或某些属性发生改变时，浏览器重新渲染部分或全部文档的过程称为回流。

会导致回流的操作：

* 页面首次渲染
* 浏览器窗口大小发生改变
* 元素尺寸或位置发生改变
* 元素内容变化（文字数量或图片大小等等）
* 元素字体大小变化
* 添加或者删除可见的DOM元素
* 激活CSS伪类（例如：:hover）
* 查询某些属性或调用某些方法

当页面中元素样式的改变并不影响它在文档流中的位置时（例如：color、background-color、visibility等），浏览器会将新样式赋予给元素并重新绘制它，这个过程称为重绘。

回流比重绘的代价要更高

如何避免：

避免使用table布局。

尽可能在DOM树的最末端改变class。

避免设置多层内联样式。

将动画效果应用到position属性为absolute或fixed的元素上。

避免使用CSS表达式（例如：calc()）。

**14. css预处理器**

提供了一种css的书写方式，常见的就是 [SAAS文档](http://sass.bootcss.com/docs/sass-reference/) 和 [LESS文档](https://less.bootcss.com/)