

HTML

1. Doctype作用？

DOCTYPE是html5标准网页声明，且必须声明在HTML文档的第一行。来告知浏览器的解析器用什么文档标准解析这个文档。

文档解析类型有：

- BackCompat：怪异模式，浏览器使用自己的怪异模式解析渲染页面。（如果没有声明DOCTYPE，默认就是这个模式）
- CSS1Compat：标准模式，浏览器使用W3C的标准解析渲染页面。

其中最大的不同就是：就是盒子模型的不同。

如何触发怪异模型：不写 DOCTYPE

```
// 告知文档使用 html5 标准来解析
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

2. 你是怎么理解HTML语义化

Step 1：先举例说明

语义化简单的说就使用正确的标签的标签做正确的事，比如标题可以使用h1~h6，导航使用nav, 块使用section等

Step 2：说说为什么需要使用语义化标签

- 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标记确定上下文和各个关键字的权重，利于SEO；
- 是其他的开发者

出彩地回答 - 举例法/阐述法

我认为 html 语义化主要指的是我们应该使用合适的标签来划分网页内容的结构。html 的本质作用其实就是定义网页文档的结构。一个语义化的文档，能够使页面的结果更加清晰，易于理解。这样不仅有利于开发者的维护和理解，同时也能够使机器对文档内容进行正确的解读。

比如我们常用的 b 标签和 strong 标签，它们在样式上都是文字的加粗，但是 strong 标签拥有强调的语义。对于一般显示来说，可能我们看上去没有差异，但是对于机器来说，就会有很大的不同。如果用户使用的是屏幕阅读器来访问网页的话，使用 strong 标签就会有明显的语调上的变化，而 b 标签则没有。

如果是搜索引擎的爬虫对我们的网页进行分析的话，那么它会依赖于 html 标签来确定上下文和各个关键字的权重，一个语义化的文档对爬虫来说是友好的。是有利于爬虫对文档内容解读的，从而有利于我们网站的 SEO，从 html 5 我们可以看出，标准是倾向于以语义化的方式来构建网页的，比如新增了 header、footer、aside 这些语义化的标签，删除了 big、font 这些没有语义的标签。html 5 新增的内容远远不止于此。

3. HTML5有哪些新特性

表示结构的标签：

| | |
|------------------------------|------------------------|
| <code><header></code> | 头部内容 |
| <code><nav></code> | 导航 |
| <code><main></code> | 主要内容 |
| <code><article></code> | 独立的内容 |
| <code><section></code> | 某个块的内容 |
| <code><aside></code> | 表示与其余页面几乎没有关系---> 侧边栏等 |
| <code><footer></code> | 底部内容 |

表示文字形式

```
<data value="1111">产品1</data>
<data value="2222">产品1</data>
```

- 表示日期和时间值， 通过读取 `datetime` 属性来指定

```
<time datetime="2001-05-15 19:00">May 15</time>.
```

- 高亮文本

```
<mark>高亮文本</mark>
```

嵌入内容

- video** 定义视频

```
<video src="https://www.w3school.com.cn/i/movie.ogg"/
controls="controls"
autoplay="autoplay"
loop="loop"
preload="preload"
muted="muted"
poster="http://p6.qhimg.com/t01f5a5270b7946b505.jpg"
height="100px"
width="200px">

</video>
```

src: 视频的地址

controls: 出现控制条（播放，静音等控制）

autoplay: 自动播放

loop: 循环播放

muted: 静音播放

poster: 视频下载时显示的图片

preload: 页面加载时加载视频

height: 视频的高

width: 视频的宽

- **audio** 定义视频

```
<audio src="https://www.w3school.com.cn/i/movie.ogg"
      loop="loop"
      controls="controls"
      preload="preload"
      autoplay="autoplay">
</audio>
```

src: 音频的地址

controls: 出现控制条（播放，静音等控制）

autoplay: 自动播放

loop: 循环播放

muted: 静音播放

preload: 页面加载时加载视频

- **canvas** 定义图形， 标只是图形容器，必须使用脚本来绘制图形

[canvas的学习及其使用](#)

这篇文章写的很清楚: <https://www.cnblogs.com/vicky1018/p/7705223.html>

4. meta viewport 是做什么用的，怎么写

Step 1：使用目的

meta viewport 是为了让用户在移动端不让用户缩放页面

Step 2：怎么写

Step 3：解释每个单词的含义

- width=device-width： 页面布局的宽度等于设备分辨率的宽度
- initial-scale=1： 初始缩放比例为设备的分辨率的大小
- maximum-scale =1: 最大的缩放比例
- minimum-scale =1: 最小的缩放比例

5.H5 是什么？

移动端页面（微信上的移动端页面）

总之 跟 html5 是没什么关系的

6.label 标签的作用

label标签是用来定义表单控件之前的关联关系的。如果点击该标签，那么相关联的控件就会哦的焦点。lable 有for属性，值是要关联的元素的id。

如果lable 标签把控件包裹起来，那么就不需要for属性。点击label 标签就会自定获得焦点。

7.行内元素有哪些？ 块级元素有哪些？ 空(void)元素有那些？

html的元素都有默认的display值，默认值为block的是块级元素。默认值为inline的为行内元素。

- 块级元素：独占一行，可以设置宽高，设置了宽高还是会占一行。
div, ul, lo, li, h1 ~ h6, table, section, main等
- 行内标签：会与其他行内标签并排在一行，不可以设置宽高。

span, a, img, input, i, b, strong等

- 空元素：没有闭合标签的标签的标签。
- img, input, br, meta

8. a标签中 如何禁用href 跳转页面 或 定位链接

```
<a href="javascript:void(0)"></a>
```

9. html5新特性之：localStorage, sessionStorage

- 共同点

存储大小为5MB，都保存在客户端，不与服务器进行交互通信，有相同的Web API

- 不同点：

localStorage长期存储数据，浏览器关闭后数据不丢失；sessionStorage的数据在浏览器关闭后自动删除；

- 共同的api

setItem(key,value)：为指定key值设置一个对应的value值

```
// 把name值存储到name的键上
sessionStorage.setItem('name', 'jacky'); // 法1
// sessionStorage.name = 'jacky'; // 法2
// sessionStorage['name'] = 'jacky'; // 法3
```

getItem (key)：根据指定的key值获取对应的value值

```
// 获取存储到 name 的键上的值
var name = sessionStorage.getItem('name');
// var name = sessionStorage.name;
// var name = sessionStorage['name'];
```

removeItem (key) : 删除指定的key值对应的value值

```
var name = sessionStorage.getItem('name'); // "jacky"
sessionStorage.removeItem('name');
name = sessionStorage.getItem('name'); // null
```

clear(): 删除所有存储的内容

```
// 清除 localStorage
sessionStorage.clear();
var len = sessionStorage.length; // 0
//length属性用于获取 sessionStorage 中键值对的数量。
```

key(index): 在指定的数字位置获取该位置的名字

```
sessionStorage.setItem('name','jacky');
var key = sessionStorage.key(0); // 'name'
sessionStorage.setItem('age', 10);
key = sessionStorage.key(0); // 'age'
key = sessionStorage.key(1); // 'name'
```

| 特性 | cookie | localStorage | sessionStorage |
|--------|---|--------------|-----------------------|
| 数据生命周期 | 一般由服务器生成，可以设置过期时间 | 除非被清理，否则一直存在 | 仅在当前会话有效，关闭页面或浏览器后被清除 |
| 数据存储大小 | 4K | 5M | 5M |
| 与服务端通信 | 每次都会携带在同源 的http请求头中,如果使用cookie保存过多 数据会带来性能问题的 | 不参与 | 不参与 |
| 用途 | 服务端生成，用于表示用户身份 | 用于浏览器缓存数据 | 用于浏览器缓存数据 |

10. html5 新的技术：webworker,websockt、Geolocation；

webworker, websockt, Geolocation

11. src 和 href的区别

- src是指向外部资源的位置，指向的内容会嵌入到文档中当前标签所在的位置，在请求src资源时会将其指向的资源下载并应用到文档内，如js脚本，img图片和frame等元素。当浏览器解析到该元素时，会暂停其他资源的下载和处理，知道将该资源加载、编译、执行完毕，所以一般js脚本会放在底部而不是头部。
- href是指网络资源所在位置，建立和当前元素（锚点）或当前文档（链接）之间的链接，用于超链接。

12. 什么叫优雅降级和渐进增强？

- 渐进增强：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：一开始就构建完整的功能，然后再针对低版本的浏览器进行兼容。

13. defer和async的区别

- defer 要等到整个页面在内存中正常渲染结束（DOM结构完全生成，以及其他脚本执行完成），才会执行。多个defer脚本会按照它们在页面出现的顺序加载。==“渲染完再执行”==
- async 一旦下载完，渲染引擎就会中断渲染，执行这个脚本以后，再继续渲染。多个async脚本是不能保证加载顺序的。==“下载完就执行”==

14. 各个浏览器的内核

- IE:trident
- 谷歌： 以前是webkit， 现在是Blink
- 火狐： gecko
- 欧朋： 开始是自己的presto， 后改称webkit, 现在是Blink
- 360: IE—chrome 双内核

