**web前端面试题整理（HTML篇）**

**web前端面试题整理（HTML篇）**  
需要web前端课程工具和电子书，可以加: 33群105601600; 22群120342833  
**1. h5的改进:**  
**新元素**  
画布canvas: HTML5 <canvas> 元素用于图形的绘制，通过脚本 (通常是JavaScript)来完成  
音频audio  
视频video  
语义性: article,  nav ,  footer, section, aside, hgroup等.  
时间time  
  
**新属性**  
拖放: draggable   <img draggable="true">  
可编辑: contenteditable  
  
**新事件**  
拖放 ondrag ondrop  
关闭页面 onunload  
窗口大小改变 onresize  
取消了一些元素: font center等  
新的DOCTYPE声明  <!DOCTYPE html>   
完全支持CSS3  
Video和Audio  
2D/3D制图  
本地存储  
本地SQL数据  
Web应用  
  
  
**2. 什么是语义化的html?**  
**what?**  
根据内容的结构（内容语义化），选择合适的标签（代码语义化）便于开发者阅读和写出更优雅的代码的同时让浏览器的爬虫和机器很好地解析。  
  
**why?**  
为了在没有CSS的情况下，页面也能呈现出很好地内容结构、代码结构:为了裸奔时好看；  
用户体验：例如title、alt用于解释名词或解释图片信息、label标签的活用；  
有利于SEO：和搜索引擎建立良好沟通，有助于爬虫抓取更多的有效信息：爬虫依赖于标签来确定上下文和各个关键字的权重  
方便其他设备解析（如屏幕阅读器、盲人阅读器、移动设备）以意义的方式来渲染网页；  
便于团队开发和维护，语义化更具可读性，是下一步吧网页的重要动向，遵循W3C标准的团队都遵循这个标准，可以减少差异化。  
  
**how?**  
尽可能少的使用无语义的标签div和span；  
在语义不明显时，既可以使用div或者p时，尽量用p, 因为p在默认情况下有上下间距，对兼容特殊终端有利；  
不要使用纯样式标签，如：b、font、u等，改用css设置。  
需要强调的文本，可以包含在strong或者em标签中（浏览器预设样式，能用CSS指定就不用他们），strong默认样式是加粗（不要用b），em是斜体（不用i）；  
使用表格时，标题要用caption，表头用thead，主体部分用tbody包围，尾部用tfoot包围。表头和一般单元格要区分开，表头用th，单元格用td；  
表单域要用fieldset标签包起来，并用legend标签说明表单的用途；  
每个input标签对应的说明文本都需要使用label标签，并且通过为input设置id属性，在lable标签中设置for=someld来让说明文本和相对应的input关联起来。  
  
  
**3. 从前端角度出发谈谈做好seo需要考虑什么?**  
语义化html标签  
合理的title, description, keywords;  
重要的html代码放前面  
少用iframe, 搜索引擎不会抓取iframe中的内容  
图片加上alt  
 **4. 文档类型(DOCTYPE)**  
**作用:**  
doctype声明位于文档中最前面的位置，处于标签之前，告知浏览器使用的是哪种规范。  
**类型:**  
三种: Strict、Transitional 以及 Frameset   
如果不声明: 不写doctype，浏览器会进入quirks mode （混杂模式）。即，如果不声明doctype，浏览器不引入w3c的标准，那么早期的浏览器会按照自己的解析方式渲染页面。浏览器采用自身方式解析页面的行为称为"quirks mode（混杂模式也称怪异模式）"；采用w3c方式解析就是"strict mode（标准模式）"。 如果完全采用strictmode就不会出任何的差错，但这样会降低程序的容错率，加重开发人员的难度  
  
**用哪种:**  
没有doctype声明的采用quirks mode解析  
对于有doctype的大多数采用standard mord。  
  
**特殊情况：**  
对于那些浏览器不能识别的doctype ,浏览器采用quirks mode；  
没有声明DTD或者html版本声明低于4.0采用quirks mode，其他使用standard mode；  
ie6中，如果在doctype声明前有一个xml声明(比如:<?xml version="1.0" encoding="iso-8859-1"?>)，则采用quirks mode解析  
  
**标准模式与怪异模式的区别:**  
标准模式：浏览器根据规范呈现页面  
混杂模式（怪异模式）：页面以一种比较宽松的兼容方式显示。  
他们最大的不同是对盒模型的解析。  
在strict mode中 ：width是内容宽度 ，也就是说,元素真正的宽度 = margin-left + border-left-width + padding-left + width + padding-right + border-right- width +  margin-right;  
在quirks mode中 ：width则是元素的实际宽度 ，内容宽度 = width - (margin-left + margin-right + padding-left + padding-right + border-left-width + 　border-right-width)  
  
  
**5.使用XHTML的局限有哪些?**  
XHTML较为严格，标签必须闭合，必须要body，head等  
如果页面使用 'application/xhtml+xml' 一些老的浏览器并不兼容  
  
  
**6. 如果网页内容需要多语言,要怎么做?**  
采用统一编码utf-8模式  
  
 **7. data-\*属性的作用**  
html5规范里增加了一个自定义data属性  
为前端开发者提供自定义的属性，这些属性集可以通过对象的dataset属性获取，不支持该属性的浏览器可以通过 getAttribute方法获取

<div data-author="david" data-time="2011-06-20" data-comment-num="10">...</div>

div.dataset.commentNum; // 可通过js获取 10

data-为前端开发者提供自定义的属性，这些属性集可以通过对象的dataset属性获取，不支持该属性的浏览器可以通过 getAttribute方法获取。ppk提到过使用rel属性，lightbox库推广了rel属性，HTML5提供了data-做替代，这样可以更好 地使用自定义的属性  
需要注意的是，data-之后的以连字符分割的多个单词组成的属性，获取的时候使用驼峰风格。  
  
  
**8. 如果把 HTML5 看作做一个开放平台，那它的构建模块有哪些？**  
1）Web Storage API  
2）基于位置服务LBS  
3）无插件播放音频视频  
4）调用相机和GPU图像处理单元等硬件设备  
5）拖拽和Form API  
  
  
**9. 请描述一下 cookies，sessionStorage 和 localStorage 的区别？**  
**描述:**  
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  
  
**作用域不同:**  
sessionStorage不在不同的浏览器窗口中共享，即使是同一个页面；  
localStorage 在所有同源窗口中都是共享的；cookie也是在所有同源窗口中都是共享的。  
Web Storage 支持事件通知机制，可以将数据更新的通知发送给监听者。  
Web Storage 的 api 接口使用更方便。  
  
  
**10. 浏览器本地存储与服务器端存储之间的区别**  
其实数据既可以在浏览器本地存储，也可以在服务器端存储。  
浏览器端可以保存一些数据，需要的时候直接从本地获取，sessionStorage、localStorage和cookie都由浏览器存储在本地的数据。  
服务器端也可以保存所有用户的所有数据，但需要的时候浏览器要向服务器请求数据。  
1.服务器端可以保存用户的持久数据，如数据库和云存储将用户的大量数据保存在服务器端。  
2.服务器端也可以保存用户的临时会话数据。服务器端的session机制，如jsp的 session 对象，数据保存在服务器上。实现上，服务器和浏览器之间仅需传递session id即可，服务器根据session id找到对应用户的session对象。会话数据仅在一段时间内有效，这个时间就是server端设置的session有效期。  
服务器端保存所有的用户的数据，所以服务器端的开销较大，而浏览器端保存则把不同用户需要的数据分布保存在用户各自的浏览器中。  
浏览器端一般只用来存储小数据，而服务器可以存储大数据或小数据。  
服务器存储数据安全一些，浏览器只适合存储一般数据。  
  
 **11. sessionStorage和页面js数据对象的区别**  
页面中一般的 js 对象或数据的生存期是仅在当前页面有效，因此刷新页面或转到另一页面这样的重新加载页面的情况，数据就不存在了。  
而sessionStorage 只要同源的同窗口（或tab）中，刷新页面或进入同源的不同页面，数据始终存在。也就是说只要这个浏览器窗口没有关闭，加载新页面或重新加载，数据仍然存在。  
  
 **12. canvas和svg的区别?**  
**svg:**  
SVG是一种使用XML描述2D图形的语言  
SVG基于XML, 这意味着SVG DOM中的每个元素都是可用的. 所以可以为每个元素附加JavaScript事件处理器  
在SVG中, 每个被绘制的图像均被视为对象. 如果SVG对象的属性发生变化, 那么浏览器能够自动重现图像  
  
**Canvas**  
Canvas通过js来绘制2D图形  
Canvas是逐像素进行渲染的  
在Canvas中, 一旦图形被绘制完成, 它就不会继续得到浏览器的关注. 如果其位置发生变化, 那么整个场景也需要重新绘制, 包括任何或许已被图形覆盖的对象.  
  
**区别**  
Canvas支持分辨率, SVG不支持  
Canvas不支持事件处理器, SVG支持  
Canvas只有弱的文本渲染能力, 而SVG最适合带有大型渲染区域的应用程序(比如谷歌地图)  
Canvas能够以.png或.jpg格式保存结果图像  
SVG的复杂度过高的话会减慢渲染速度(任何过度使用DOM的应用都不快)  
Canvas最适合图像密集型的游戏, 其中的许多对象会被频繁重绘. 而SVG不适合游戏应用  
Canvas是基于位图的图像,它不能够改变大小, 只能缩放显示; SVG是基于矢量的, 所以它能够很好地处理图形大小的改变　　　　  
Canvas提供的功能更原始, 适合像素处理, 动态渲染和大数据量绘制; SVG功能更完善, 适合静态图片显示, 高保真文档查看和打印的应用场景  
绘制Canvas对象后, 不能使用脚本和CSS对它进行修改; 而SVG对象是文档对象模型的一部分, 所以可以随时使用脚本和CSS修改它们  
  
  
**13. src和href的区别?**  
src指向外部资源的位置, 用于替换当前元素, 比如js脚本, 图片等元素  
href指向网络资源所在的位置, 用于在当前文档和引用资源间确定联系, 加载css

【内容展示有限，可以加群下载，群文件会定期更新学习资料，以及练手小案例】