

## 米傻的博客

勿在浮沙筑高楼！

## HTML面试题

📅 2016-10-08 | 📁 [HTML](#)**doctype(文档类型)的作用是什么？**

声明文档的解析类型(document.compatMode)，避免浏览器的怪异模式。有以下两个值：

- 怪异模式，浏览器使用自己的怪异模式解析渲染页面。
- 标准模式，浏览器使用W3C的标准解析渲染页面。

这个属性会被浏览器识别并使用，但是如果你的页面没有DOCTYPE的声明,浏览器按照自己的方式解析渲染页面（怪异模式），在不同的浏览器就会显示不同的样式。

**注意以下几点：**

- <!DOCTYPE> 声明不是 HTML 标签；
- <!DOCTYPE> 声明必须是 HTML 文档的第一行，位于 标签之前。
- <!DOCTYPE> 不区分大小写
- HTML 5的doctype声明是<!DOCTYPE html>

在 HTML 4.01 中，<!DOCTYPE> 声明引用 DTD，因为 HTML 4.01 基于 SGML。DTD 规定了标记语言的规则，这样浏览器才能正确地呈现内容。HTML5 不基于 SGML，所以不需要引用 DTD。

**浏览器标准模式和怪异模式有什么区别？****两者的区别：**

- 盒模型的解析上：

在strict mode中：width是内容宽度

在quirks mode中：width则是元素的实际宽度，而内容宽度 = width - (margin-left + margin-right + padding-left + padding-right + border-left-width + border-right-width)

- 图片元素垂直对齐方式

在strict mode中：vertical-align 属性默认取值为 baseline

在quirks mode中：vertical align 属性默认为 bottom，因此，在图片底部会有几像素的空间。

- <table>元素中的字体

Quirks Mode 下，对于 table 元素，字体的某些属性将不会从 body 或其他封闭元素继承到 table 中，特别是 font-size 属性。

- 内联元素的尺寸

在 Standards Mode 下，non-replaced inline 元素无法自定义大小，而在 Quirks Mode 下，定义这些元素的 width 和 height 属性，能够影响该元素显示的大小尺寸。

- 元素的百分比高度

当一个元素使用百分比高度时，在 Standards Mode 下，高度取决于内容的变化，而在 Quirks Mode 下，百分比高度则被正确应用。

- 元素溢出的处理

在 Standard Mode 下，overflow 取默认值 visible，即溢出可见，这种情况下，溢出内容不会被裁剪，呈现在元素框外。而在 Quirks Mode 下，该溢出被当做扩展 box 来对待，即元素的大小由其内容决定，溢出不会被裁剪，元素框自动调整，包含溢出内容。

## HTML和XHTML的区别是什么？

主要的不同：

- XHTML可以理解为html+xml，就是用xml的语法来规范html。
- XHTML 元素必须被正确地嵌套。
- XHTML 元素必须被关闭。
- 标签名必须用小写字母。
- XHTML 文档必须拥有根元素。

## 如果网页内容需要支持多语言，你会怎么做？

考虑:

- 应用字符集的选择 utf-8
- 语言书写习惯&导航结构
- 数据库驱动型网站

具体做法：

- 静态：就是为每种语言分别准备一套页面文件，要么通过文件后缀名来区分不同语言，要么通过子目录来区分不同语言。
- 动态：站点内所有页面文件都是动态页面文件（PHP，ASP等）而不是静态页面文件，在需要输出语言文字的地方统一采用语言变量来表示，这些语言变量可以根据用户选择不同的语言赋予不同的值，从而能够实现在不同的语言环境下输出不同的文字

## data-属性的作用是什么？

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

## 如果把 HTML5 看作做一个开放平台，那它的构建模块有哪些？

开放网络平台（Open Web Platform）是一些开放的（免版权）技术的集合，这些技术激活了互联网。使用开放网络平台时，每个人都有权实现 Web 上的一个组件，而不用向任何人索取许可和证书。

构建模块，指的应该是开放网络平台这个技术集合中的技术：

HTML

DOM

CSS

SVG

MathML

Web APIs...

EcmaScript / JavaScript

HTTP

URI

Media Accessibility Checklist

请描述 cookies、sessionStorage 和 localStorage 的区别。

**共同点：**都是保存在浏览器端，且同源的。

**区别：**

- 大小  
cookie 最多只有 4kb，而 sessionStorage 和 localStorage 大小一般可以有 5M
- 生命周期  
cookie 的生命周期由服务器控制，默认是关闭浏览器后删除；sessionStorage 仅在当前的窗口有效，localStorage 除非手动删除否则一直存在。
- http 通信  
浏览器每次向服务器发送请求的时候都要带上该域的 cookie，而 sessionStorage 和 localStorage 仅存在于浏览器端。
- 作用域  
cookie 和 localStorage 在同个域名下的多个窗口都有效，sessionStorage 只在一个窗口有效，不能跨窗口共享。
- 易用性  
sessionStorage 和 localStorage 属于 HTML5 的 Web Storage 的 API，更加灵活易用。

**Web Storage带来的好处：**

- 减少网络流量：一旦数据保存在本地后，就可以避免再向服务器请求数据，因此减少不必要的请求，减少数据在浏览器和服务器间不必要的来回传递。
- 快速显示数据：性能好，从本地读数据比通过网络从服务器获得数据快得多，本地数据可以即时获得。再加上网页本身也可以有缓存，因此整个页面和数据都在本地的话，可以立即显示。
- 临时存储：很多时候数据只需要在用户浏览一组页面期间使用，关闭窗口后数据就可以丢弃了，这种情况使用sessionStorage非常方便。

**服务端存储**

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。

- 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。
- 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。

**参考：**

[H5本地存储](#)

[session原理](#)

**请解释 <script>、<script async> 和 <script defer> 的区别**

通过script标签加载js文件时，如果加载过慢，或者出错，有可能会阻塞整个页面的加载，这时就需要js的异步加载，以下两个属性可支持异步加载：

defer 和 async 的**共同点**：都是可以并行加载JS文件，不会阻塞页面的加载，

defer 和 async 的**不同点**：defer的加载完成之后，JS会等待整个页面全部加载完成了再执行，而async是加载完成之后，会马上执行JS，所以假如对JS的执行有严格顺序的话，那么建议用 defer加载。

**为什么通常推荐将 CSS <link> 放置在 <head></head> 之间，而将 JS <script> 放置在 </body> 之前？你知道有哪些例外吗？**

**css放在head中：**

- css放在head中，是因为浏览器解析html文档是自上而下的，如果放底部的话，页面结构出来了，css还没开始渲染，可能会看到只有结构的页面。

- CSS 应当写在 head 中，以避免页面元素由于样式确实造成瞬间的白页或者给用户闪烁感。

js放在/body之前：

- JS可能会改变DOM树，也可能依赖css样式。如果放在前面，那么DOM和css可能还未加载，这样容易报错。
- 性能：js放前面，页面会先去加载他，拖慢了时间，使用户在一定时间内看不到网页内容。

**例外：**js如果需要先加载后运行可以写在头里（当脚本使用 defer 方式加载的时候可以不用约束放置的位置。）；

### 如果页面使用‘application/xhtml+xml’会有什么问题吗？

首先这是个MIME类型，意思就是告诉浏览器，要用xhtml的格式来解析我发给你的页面；

xhtml 语法要求严格，必须有head、body 每个dom 必须要闭合。空标签也必须闭合。例如<img />, <br/>, <input />等。另外要在属性值上使用双引号。一旦遇到错误，立刻停止解析，并显示错误信息。

如果页面使用‘application/xhtml+xml’,一些老的浏览器会不兼容。

IE6，7，8不支持，IE6，7，8支持text/html。

### 什么是渐进式渲染 (progressive rendering) ？

渐进式渲染是指浏览器不用等待所有页面资源都渲染好之后再呈现给用户看，而是边下载边渲染，所以用户打开一个网页的时候往往不能第一时间看到所有的内容，但是能够看到一个大概的样子，后续的内容浏览器会慢慢补上形成一个完整的页面。这个有点像 bigpipe。

### 请解释下什么事语义化的HTML？

语义化的好处：

- 去掉或样式丢失的时候能让页面呈现清晰的结构：html本身是没有表现的，我们看到例如<h1>是粗体，字体大小2em，加粗；<strong>是加粗的，不要认为这是html的表现，这些其实html默认的css样式在起作用，所以去掉或样式丢失的时候能让页面呈现清晰的结构不是语义化的HTML结构的优点，但是浏览器都有默认样式，默认样式的目的也是为了更好的表达html的语义，可以说浏览器的默认样式和语义化的HTML结构是不可分割的。
- 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。
- PDA、手机等设备可能无法像普通电脑的浏览器一样来渲染网页（通常是因为这些设备对CSS的支持较弱）。
- 搜索引擎的爬虫也依赖于标记来确定上下文和各个关键字的权重。
- 你的页面是否对爬虫容易理解非常重要,因为爬虫很大程度上会忽略用于表现的标记,而只注重语义标记。
- 便于团队开发和维护

语义化的HTML就是：标题用h1-h6，文字段落用p，列表用ul li，大致如此

# HTML面试题

◀ JS面试题

性能优化之一减少HTTP请求 ▶



