

只要7个问题，帮助企业找到一个**的HTML5开发者

产品名称	只要7个问题，帮助企业找到一个**的HTML5开发者
公司名称	东莞市数云网络科技有限公司
价格	.00/件
规格参数	
公司地址	广东省东莞市松山湖园区瑞和路1号2栋501室02
联系电话	18002820787 18002820787

产品详情

1.问：既然HTML5被设计用来替换HTML4和XHTML，那么请简短解释HTML5有哪些新特性？

新的HTML5规范的主要目标是向客户提供丰富的内容（如图形和视频），而不需要额外的插件（即Flash和Silverlight），通过引入新的结构元素标签，为网页结构提供更好的语义支持，提供更严格的解析标准以简化错误处理，并简化与写入旧标准的文档的向后兼容性。z后，现代世界z重要的目标是提供更好的跨平台支持，并确保无论在台式电脑、笔记本电脑、平板电脑还是智能机上都能运行良好。

为了实现这一点，HTML5引入了许多新的特性。新的HTML5通过新的<canvas>，<audio>和<video>标签嵌入了图形、音频和视频内容的改进支持。引入了Web workers，添加了JavaScript API的新扩展，如地理位置、新的拖放特性以及本地存储和缓存特性。引入了许多新的语义标签和新的表单控件来补充现代Web应用程序的结构逻辑。

问：请简短解释HTML是什么，HTML5元素中有哪些新的命名规范？

语义HTML是一种HTML，标记或标签显示的是意义，而不是纯粹的呈现或外观。例如，HTML5建议使用标签而不是用于粗体文本，而使用代替斜体文本<i>。这些语义标记将生成相同的粗体和斜体文本，但是它们也提供了含义，而不是纯格式化信息。

HTML5规范定义了全新的语义元素，如：

```
<article><aside><details><figcaption><figure><footer><header><main><mark><nav><section><summary><time>
```

新的HTML5窗体控件包括：

<calendar><date><time><email><url><search>

2.问：请问在用HTML5开发应用时，经常会遇到哪些陷阱？

许多开发人员通过使用旧的规范在HTML5中编写现代Web应用程序。突出的例子是，当开发人员使用<table>标记来设置布局时，或者或<div>时，新的语义标记如<header>、<footer>、<article>或<aside>标记会更合适。或者如前面所讨论的，它们不使用新的语义标签或而不是使用和<i>标签。这个老方法的结果是跨不同浏览器行为不一致的过于复杂的标记。在编写新的HTML5文档时，开发人员应该采用新的语义标记，享受HTML5规范带来的所有好处，并让旧习惯留在过去。

3.问：请问如何在不同的组合中使用新的HTML5元素，即<header>和<footer>，以及<section>和<article>？

新的HTML5元素开辟了一种全新的可能性和有趣的用法组合。

例如，尽管经典的知识是只有一个页眉和只有一个页脚，但是新的HTML5文档可以包含多个<header>和<footer>元素。这两个新的语义标签被设计成服务于它们各自的目的相对于它们的父元素。这意味着不仅页面<body>可以包含页眉和页脚，而且每个<article>和<section>元素都可以。

另一个例子是新的<section>和<article>元素的使用，它们的组合：<section>可以包含<article>元素，而<article>可以包含<section>元素。用一个例子来说明这一点：在你的网页中，你可以有一个带有<section>的用于社交网络交互的仪表板页面，和一个<section>的用于包含几个<article>元素的z新新闻文章的仪表板页面。相反，<article>结尾可能包含一个<section>供读者评论。

4.问：请解释HTML5是如何简化HTML结构？提供实例。

首先，HTML5规范简化了DOCTYPE声明。DOCTYPE或文档类型声明告诉浏览器他们可以期望的文档类型。旧的XHTML DOCTYPE声明如下：

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

而HTML4过渡性DOCTYPE看起来像这样：

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

HTML5的新DOCTYPE声明非常简单：

```
<!DOCTYPE html>
```

HTML5引入的另一个简化是我们如何声明文档的字符编码或字符集。HTML5中的字符集声明如下：

```
<!DOCTYPE html><html><head><meta charset="UTF-8">...</head><body>...</body></html>
```

这在HTML4中比以前没有CARSET元属性要简单得多：

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd"><html><head><meta http-equiv="Content-Type" content="text/html;
charset=utf-8">...</head><body>...</body></html>
```

HTML5规范在简化过程中进行得如此之远，以至于标记<html>、<body>和<head>对于HTML5文档的有效性不是强制性的。下面的简单示例将通过W3C标记验证服务：

```
<!DOCTYPE html><title>My title</title><header>My header</header><p>Wall of text</p>
```

这个示例中唯一可用的重要标记是HTML5的新doctype，如果没有它，浏览器将无法检测到它是HTML5文档。

问：为什么需要所有的浏览器都遵循相同的HTML5规范呢？

HTML5规范是一组定义有效文档的规则，还提供了关于浏览器必须如何解释和呈现此类文档的指令。不幸的是，实际情况是没有浏览器仍然支持HTML5规范定义的所有规则。大多数主要浏览器都支持大多数规范，但是HTML5规范的浏览器解释仍然存在差异。因此，开发人员有必要确认他们使用的方面是否将得到他们希望显示内容的所有浏览器的支持。这就是为什么跨浏览器支持对于开发者来说仍然是一个头疼的问题，尽管改进了规范。

5.问：请问你对HTML5的可访问性方面有什么看法？特别是它在现实世界中使用时的局限性和问题。

今天，网页和应用程序被越来越多地使用，使得依赖辅助技术的人们能够访问网页和应用程序变得比以前更加重要。HTML已经走了很长的路，HTML5引入了新的用户界面特性，使得那些依赖辅助技术的人更容易使用网络。尽管存在一些问题。

问题之一是，开发人员和设计人员以前没有过多关注他们的网页或应用程序的这方面。正如前面提到的，他们辩护的一个主要原因是之前的HTML5没有任何可访问性接口特性。现在，即使开发人员希望在他们应用程序中实现附加的可访问性选项，任何新的和新兴的Web技术都面临着一个经常性的问题：浏览器支持。HTML5可访问性是跟踪浏览器上z常见的可访问性特征。为了实现可访问性特征并覆盖所有浏览器，开发人员需要额外的时间投资。如果客户希望自己的Web应用程序和页面处于当今技术所能达到的边缘，这可能是客户需要预期的关键因素。

6.问：请问HTML5 Web存储安全性吗？并解释local Storage和Web Storage之间有什么区别？

HTML5使Web页面能够使用Web存储在用户浏览器中本地存储数据。在早期版本中，开发人员只能使用Cookie。新的Web存储更加安全和快捷。与Cookie不同，Web存储的数据不包含在每个服务器请求中，只在被请求时才使用。数据存储为名称和值对。cookie的其他好处是存储限制。Web存储可以达到5MB大，并且其内容从不传输到服务器。网页只能访问自身存储的数据，因为每个原点是有限的。

重要的是要注意，虽然Web存储比Cookie更安全，但有些事情需要牢记。它比使用Cookie更好，因为内容不是通过线传输的，而是本地存储没有加密。出于这个原因，像安全令牌这样的敏感数据永远不应该存储在那里。Web应用程序不应该依赖于存储在Web Storage中的数据，因为恶意用户可以随时轻松地修改Local Storage和会话Storage值中的数据。

说到Session Storage，Local Storage和会话Storage之间的差别涉及存储的生存期和范围。通过Local Storage存储的数据是永久性的：它不会过期，并且一直存储在用户的计算机上，直到Web应用程序删除它，或者用户请求浏览器删除它。另一方面，Session Storage与浏览器选项卡相同的生存期，其中存储了它的脚本正在运行。当选项卡关闭时，通过Session Storage的任何数据都会被删除。

与本地存储的起始限制不同，Session Storage是窗口范围。例如，如果用户有两个浏览器选项卡，显示来自相同来源的文档，那么这两个选项卡具有单独的Session Storage数据。在一个选项卡中运行的脚本不能读取或覆盖由另一个选项卡中的脚本编写的数据，即使两个选项卡访问完全相同的页面并且运行完全相同的脚本。

7.问：请问你是如何看待WEB开发者的？

JavaScript是一个单线程语言，所以多个脚本不能同时运行。WebWorker是一种新的API，它独立于其他脚本，在其自身的线程中在浏览器后台运行脚本。结果是网页不需要等待Web工作者完成，这提高了性能和响应能力，因为用户可以在Web工作者还在后台运行的时候与页面交互。例如，Web工作者对于执行大量计算的长时间运行的脚本来说是完美的。

只要7个问题，帮助企业找到一个优秀的HTML5开发者

只要7个问题，帮助企业找到一个优秀的HTML5开发者