

**Web前端项目实战**

**HTML5篇**

目录

[目录 2](#_Toc453687691)

[第1章 前言 3](#_Toc453687692)

[1.1 认识HTML5 3](#_Toc453687693)

[1.2 语法规范 4](#_Toc453687694)

[第2章 语义化 6](#_Toc453687695)

[2.1 语义化标签 6](#_Toc453687696)

[2.1.1 常用语义标签 6](#_Toc453687697)

[2.1.2 兼容处理 7](#_Toc453687698)

[2.2 微数据 8](#_Toc453687699)

[2.3 ARIA 8](#_Toc453687700)

[第3章 表单 8](#_Toc453687701)

[3.1 输入类型 9](#_Toc453687702)

[3.2 表单元素 10](#_Toc453687703)

[3.3 表单属性 10](#_Toc453687704)

[3.4 表单事件 10](#_Toc453687705)

[第4章 多媒体 11](#_Toc453687706)

[4.1 音频 11](#_Toc453687707)

[4.2 视频 12](#_Toc453687708)

[第5章 DOM扩展 13](#_Toc453687709)

[5.1 获取元素 13](#_Toc453687710)

[5.2 类名操作 13](#_Toc453687711)

[5.3 自定义属性 14](#_Toc453687712)

# 前言

## 认识HTML5

HTML5并不仅仅只是作为HTML标记语言的一个最新版本，更重要的是它制定了Web应用开发的一系列标准，成为第一个将Web做为应用开发平台的HTML语言。

HTML5定义了一系列新元素，如新语义标签、智能表单、多媒体标签等，可以帮助开发者创建富互联网应用，还提供了一系列Javascript API，如地理定位、重力感应、硬件访问等，可以在浏览器内实现类原生应用，甚至结合Canvas我们可开发网页版游戏，同时结合CSS3的过渡、转换、动画等特性，可以极大的增强用户体验，提升开发功能的可应用性。

我们日常讨论的H5其实是一个泛称，它指的是由HTML5 + CSS3 + Javascript等技术组合而成的一个应用开发平台。

## 语法规范

随着Web技术的更新，HTML也先后经历了HTML 4.01、XHTML 1.0、HTML5几个重要的版本，在版本的演变过程中新增或废弃了一些属性，同时对语法规范也做了一些调整，为了能够保证浏览器可以兼容不同版本语法规范的，我们可以使用<!DOCTYPE>指示浏览器应该如何处理我们的HTML。

见示例代码1-1.html

HTML5在语法规范上也做了比较大的调整，去除了许多冗余的内容，书写规则更加简洁、清晰。

见示例代码1-2.html

通过以上的比较我们可以总结得出HTML5在语法规范上的特点：

1、更简洁

2、更宽松

但是我们在实际开发中应规范书写，不建议太随意 ！并且现在基本所有网站都采用了HTML5的语法规范。

**W3C验证地址**

https://validator.w3.org/

# 语义化

所谓语义化是要使HTML标签具备很好的可读性，可以清晰传达每个标签所要表达的意义，以方便其被友好的处理和解析。

## 语义化标签

对于语义化标签我们并不陌生，如<p>表示一个段落、<ul>表示一个无序列表<h1> ~ <h6>表示一系列标题等，在此基础上HTML5增加了大量更有意义的语义标签，更有利于搜索引擎或辅助设备来理解HTML页面内容。

传统的做法我们通过添加类名如class="header"、class="footer"，使HTML页面具有语义性的，但是不具有通用性（如class="header"也可能被写成class="head"）。

见示例代码2-1.html

HTML5则是通过增加语义化标签的形式来解决这个问题，例如<header></header>、<footer></footer>等，这样就可以保证其具有通用性。

见示例代码2-2.html

此章节学习目的为了解增加语义标签的目的，以及各语义标签所表达的意义，在网页布局中能够合理使用标签。

### 常用语义标签

|  |  |
| --- | --- |
| 标签 | 语义 |
| <nav> | 导航 |
| <header> | 页眉 |
| <footer> | 页脚 |
| <section> | 区块 |
| <article> | 文章 |
| <aside> | 侧边栏 |
| <progress> | 进度条 |
|  |  |

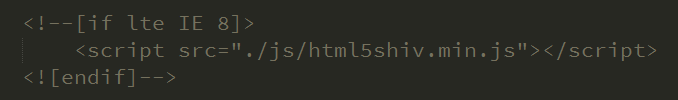
见示例代码2-3.html

本质上新语义标签与<div>、<span>没有区别，只是其具有语义性，使用时除了在HTML结构上需要注意外，其它和普通标签的使用无任何差别，可以理解成<div class="nav"> 相当于 <nav>。**不要好奇，它只是一个标签！**

尽量避免全局使用header、footer、aside等语义标签。

### 兼容处理

在不支持HTML5新标签的浏览器里，会将这些新的标签解析成行内元素(inline)对待，所以我们只需要将其转换成块元素(block)即可使用，但是在IE9版本以下，并不能正常解析这些新标签，但是却可以识别通过document.createElement('tagName')创建的自定义标签，于是我们的解决方案就是将HTML5的新标签全部通过document.createElement('tagName')来创建一遍，这样IE低版本也能正常解析HTML5新标签了，但在实际开发中我们更多采用的是通过检测IE浏览器的版本来加载第三方的一个JS库来解决兼容问题，这个库文件会帮自动通过document.createElement('tagName')创建所有HTML5的新标签。



见示例代码2-4.html

## 微数据

微数据是在如 span、div 的标签内添加属性，让机器（如搜索引擎）识别其含义，某些特定类型的信息，例如评论、人物信息或事件都有相应的属性，用来描述其含义，可以理解成新语义标签的一种补充。

见示例代码2-5.html

[更多学习资料](http://www.zhangxinxu.com/wordpress/2011/12/html5%E6%89%A9%E5%B1%95-%E5%BE%AE%E6%95%B0%E6%8D%AE-%E4%B8%B0%E5%AF%8C%E7%BD%91%E9%A1%B5%E6%91%98%E8%A6%81/)

## WAI-ARIA

WAI-ARIA**[1]**, 是Web Accessibility Initiative - Accessible Rich Internet Applications 的缩写，她主要解决的一个问题：让残障人士能无障碍地访问网页上的动态内容。

[见参考资料](http://www.zhangxinxu.com/wordpress/2012/03/wai-aria-%E6%97%A0%E9%9A%9C%E7%A2%8D%E9%98%85%E8%AF%BB/)

# 表单

伴随着互联网富应用以及移动开发的兴起，传统的Web表单已经越来越不能满足开发的需求，所以HTML5在Web表单方向也做了很大的改进，如拾色器、日期/时间组件等，使表单处理变的更加高效。

## 输入类型

|  |  |  |
| --- | --- | --- |
| 类型 | 使用示例 | 含义 |
| email | <input type="email"> | 输入邮箱格式 |
| tel | <input type="tel"> | 输入手机号码格式 |
| url | <input type="url"> | 输入url格式 |
| number | <input type="number"> | 输入数字格式 |
| search | <input type="search"> | 搜索框（体现语义化） |
| range | <input type="range"> | 自由拖动滑块 |
| color | <input type="color"> | 拾色器 |
| time | <input type="time"> |  |
| date | <input type="date"> |  |
| datetime | <input type="datetime"> |  |
| month | <input type="month"> |  |
| week | <input type="week"> |  |

见示例代码3-1.html

部分类型是针对移动设备生效的，且具有一定的兼容性，在实际应用当中应选择性的使用，扫描二维码可以对比测试。



## 表单元素

|  |  |
| --- | --- |
| 元素 | 含义 |
| <datalist> | 数据列表 |
| <keygen> | 生成加密字符串 |
| <output> | 输出结果 |
| <meter> | 度量器 |

见示例代码3-2.html

## 表单属性

|  |  |  |
| --- | --- | --- |
| 属性 | 用法 | 含义 |
| placeholder | <input type="text" placeholder="请输入用户名"> | 占位符 |
| autofocus | <input type="text" autofocus> | 自动获得焦点 |
| multiple | <input type="file" multiple> | 多文件上传 |
| autocomplete | <input type="text" autocomplete="off"> | 自动完成 |
| form | <input type="text" form="某表单ID"> |  |
| novalidate | <form novalidate></form> | 关闭验证 |
| required | <input type="text" required> | 必填项 |
| pattern | <input type="text" pattern="\d"> | 自定义验证 |

见示例代码3-3.html

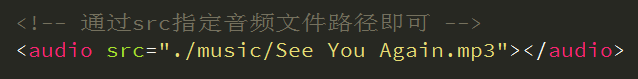
# 多媒体

在HTML5之前，在网页上播放音频/视频的通用方法是利用Flash来播放，但是大多情况下，并非所有用户的浏览器都安装了Flash插件，由此使得处理音频/视频播放变的非常复杂，并且移动设备的浏览器并不支持Flash插件。

## 音频

HTML5通过<audio>标签来解决音频播放的问题。

使用相当简单，如下图所示



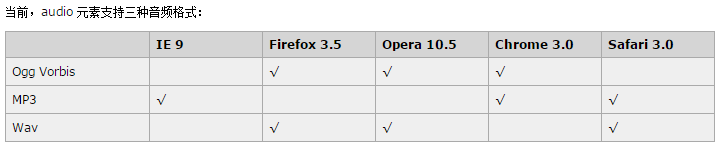
并且可以通过附加属性可以更友好控制音频的播放，如：

autoplay 自动播放

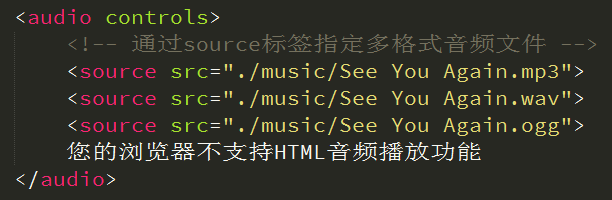
controls 是否显不默认播放控件

loop 循环播放

由于版权等原因，不同的浏览器可支持播放的格式是不一样的，如下图供参考



**多浏览器支持的方案，如下图**

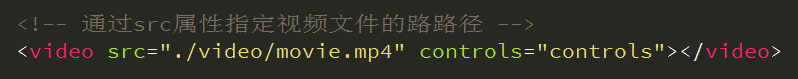


见示例代码4-1.html

## 视频

HTML5通过<video>标签来解决音频播放的问题。

同音频播放一样，<video>使用也相当简单，如下图



同样，通过附加属性可以更友好的控制视频的播放

autoplay 自动播放

controls 是否显示默认播放控件

loop 循环播放

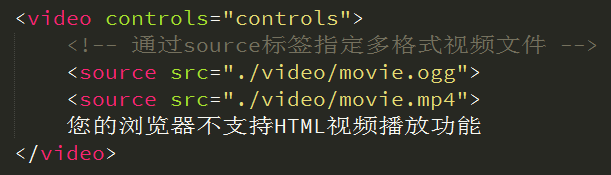
width 设置播放窗口宽度

height 设置播放窗口的高度

由于版权等原因，不同的浏览器可支持播放的格式是不一样的，如下图供参考



**多浏览器支持的方案，如下图**



见示例代码4-2.html

# DOM扩展

## 获取元素

1、document.getElementsByClassName ('class') 通过类名获取元素，以伪数组形式存在。

见示例代码5-1.html

2、document.querySelector('selector') 通过CSS选择器获取元素，符合匹配条件的第1个元素。

见示例代码5-2.html

3、document.querySelectorAll('selector') 通过CSS选择器获取元素，以伪数组形式存在。

见示例代码5-3.html

## 类名操作

1、Node.classList.add('class') 添加class

2、Node.classList.remove('class') 移除class

3、Node.classList.toggle('class') 切换class，有则移除，无则添加

4、Node.classList.contains('class') 检测是否存在class

Node指一个有效的DOM节点，是一个通称。

见示例代码5-4.html

## 自定义属性

在HTML5中我们可以自定义属性，其格式如下data-\*=""，例如

data-info="我是自定义属性"，通过Node.dataset['info'] 我们便可以获取到自定义的属性值。

Node.dataset是以对象形式存在的，当我们为同一个DOM节点指定了多个自定义属性时，Node.dataset则存储了所有的自定义属性的值。

假设某元素 <div id="demo" data-name="itcast" data-age="10">

var demo = document.querySelector('#demo');

1、读取 demo.dataset['name'] 或者 demo.dataset['age']

2、设置demo.dataset['name'] = 'web developer'

见示例代码5-5.html

注：当我们如下格式设置时，则需要以驼峰格式才能正确获取

<div data-my-name="itcast"> 这样获取Node.dataset['myName']