

**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。

摇一摇 小球 地图定位

## 1.2 HTML5基本概念

01 html5 就是在 html4.的基础上新增了一新标签和新的API。

02 Html5具有非常松散的语法，有很多标签都可以省略不写，或者只写开始。

03 Html5新增标签具有很强的语义化。

## 1.3什么是HTML5

html:超文本标记语言

xhtml:严格型超文本标记语言（过渡版本）

狭义上来说：html5，就是超文本标记语言的第5次修订（标签的扩展）。

广义上来说：html5=h5新标签 + css3 + javascript新功能 + api（说明文档）

## 1.4 HTML5的兼容性

只有IE9及以上的现代浏览器支持，手机上的浏览器都属于现代浏览器。

目前市面上主流的标准浏览器都相应的支持或者部分支持HTML5，但是非标准（ IE6 7 8 ）不支持，所以目前我们在使用HTML5时只用它的骨架而不用具体的HTML5标签进行布局。

Ps：国内的大部分浏览器都是基于Chrome的内核.



## 1.5 HTML5的优点

（1）相较于xhtml而言，它的性能较高，功能更全面（可以做H5小游戏，如：微信打飞机）；

（2）新增了大量的语义化标签，利于SEO优化（搜索引擎优化，如：百度、谷歌）；

（3）h5中css3的发布，将页面表现带入了一个全新的高度（可以创建3D世界）；

（4）h5中的语法规范更加松散（如：所有的type可以不写，单标签可以不闭合....）.

例：

1. <script type="text/javascript"></script>
2. //下面的方式也可
3. <script></script>
4. //单标签可不闭合
5. <img src="" alt="" >

## 1.6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

## 1.7 HTML5的骨架

1. <!DOCTYPE html> //文档类型声明，就是跟W3C宣誓，该网站符合最新的html标准（即html5，第五代标准）
2. <html lang="en"> //文件命名空间，规定页面内所有的标签、class、Id等必须遵循英文命名规范
3. <head>
4. <meta charset="UTF-8" />
5. <title></title>
6. </head>
7. <body>
8. </body>
9. </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. XHTML：
3. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4. <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
5. <head>
6. <meta http-equiv="Content-Type" content="text/html;charset=UTF-8" />

2）HTML5:

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <meta charset="UTF-8">

2、更宽松(单标签可以不闭合、style可不写...)

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

1. <!-- 头部 -->
2. <div class="header">头部</div>
3. <!-- 主体部分 -->
4. <div class="container">
5. <div class="article"></div>
6. <div class="aside"></div>
7. </div>
8. <!-- 底部 -->
9. <div class="footer"></div>

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(定义网页中的导航内容) |
| **<header>** | 页眉(定义网页区域中的头部内容。) |
| **<footer>** | 页脚(定义区域底底部内容) |
| **<section>** | 区块(和之前的div 类似没有特殊语义,表示网页中的某一个完整区域。) |
| **<article>** | 文章(定义一块完整的文字内容) |
| **<aside>** | 侧边栏(定义区域中的侧边栏内容) |
| **<progress>** | 进度条(各个浏览器效果不同) |

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。

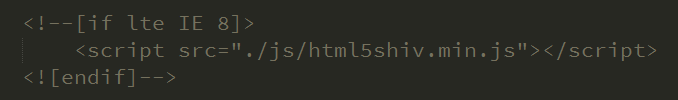


### 兼容处理

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，可以识别通过document.createElement('tagName')创建的自定义标签，使得IE低版本也能正常解析HTML5新标签了，

1. <head>
2. <meta charset="UTF-8">
3. <title>Document</title>
4. <style>
5. header {
6. display: block;
7. width: 980px;
8. height: 100px;
9. margin: 0 auto;
10. background-color: red;
11. }
12. footer {
13. display: block;
14. width: 980px;
15. height: 80px;
16. margin: 0 auto;
17. background-color: blue;
18. }
19. </style>
20. <script>
22. document.createElement('header');
23. document.createElement('footer');
24. //非H5标签也可通过此种方式创建
25. document.createElement('itcast');
26. </script>
27. </head>
28. <body>
29. <header></header>
30. <footer></footer>
31. </body>

将HTML5的新标签全部通过document.createElement('tagName')来创建一遍，较麻烦，实际开发中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，无需再给新标签添加display:block。



见示例代码2-4.html

PS:Sublime中cc:ie6+tab 可得到：

1. <!--[if lte IE 6]>
3. <![endif]-->

## 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

1. <body>
2. <!-- 传统做法 -->
3. <!-- <div>
4. <span>Kayo</span>
5. <span>2012年2月6日 22:30</span>
6. <span>评论内容</span>
7. </div> -->
9. <!-- 友好做法，可以让机器也能识别内容 -->
10. <div itemscope itemtype="http://schema.org/Movie">
11. <span itemprop="actor">刘德华</span>
12. <span itemprop="director" datatime="2012-02-06T22:30:30.50+08:00">2012年2月6日 22:30</span>
13. <a href="" itemprop="url"></a>
14. <p itemprop="url" >http://www.baidu.com</p>
15. <span itemprop="description">评论内容</span>
16. </div>
17. </body>

[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/)

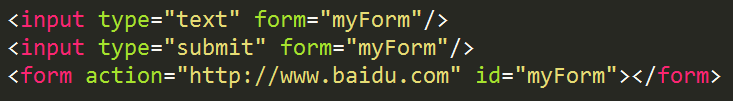
# 表单

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。

## 3.1智能表单

网站中所有的数据都是存储在远程服务器（超级计算机）上的，用户在页面上输入的值最终需要传递到服务器，以便服务器做筛选，进而返回不同的数据给用户。

通过<form action=”服务器地址”>可以将表单中的数据传递到后台，但在xhtml年代，form和表单元素必须是包含关系，十分不利；在h5中，form和表单元素可以不再是包含关系，只需要给form添加ID，在表单元素身上新增form=”formID名称”属性即可。



## 3.2 H5中常用的新增表单类型

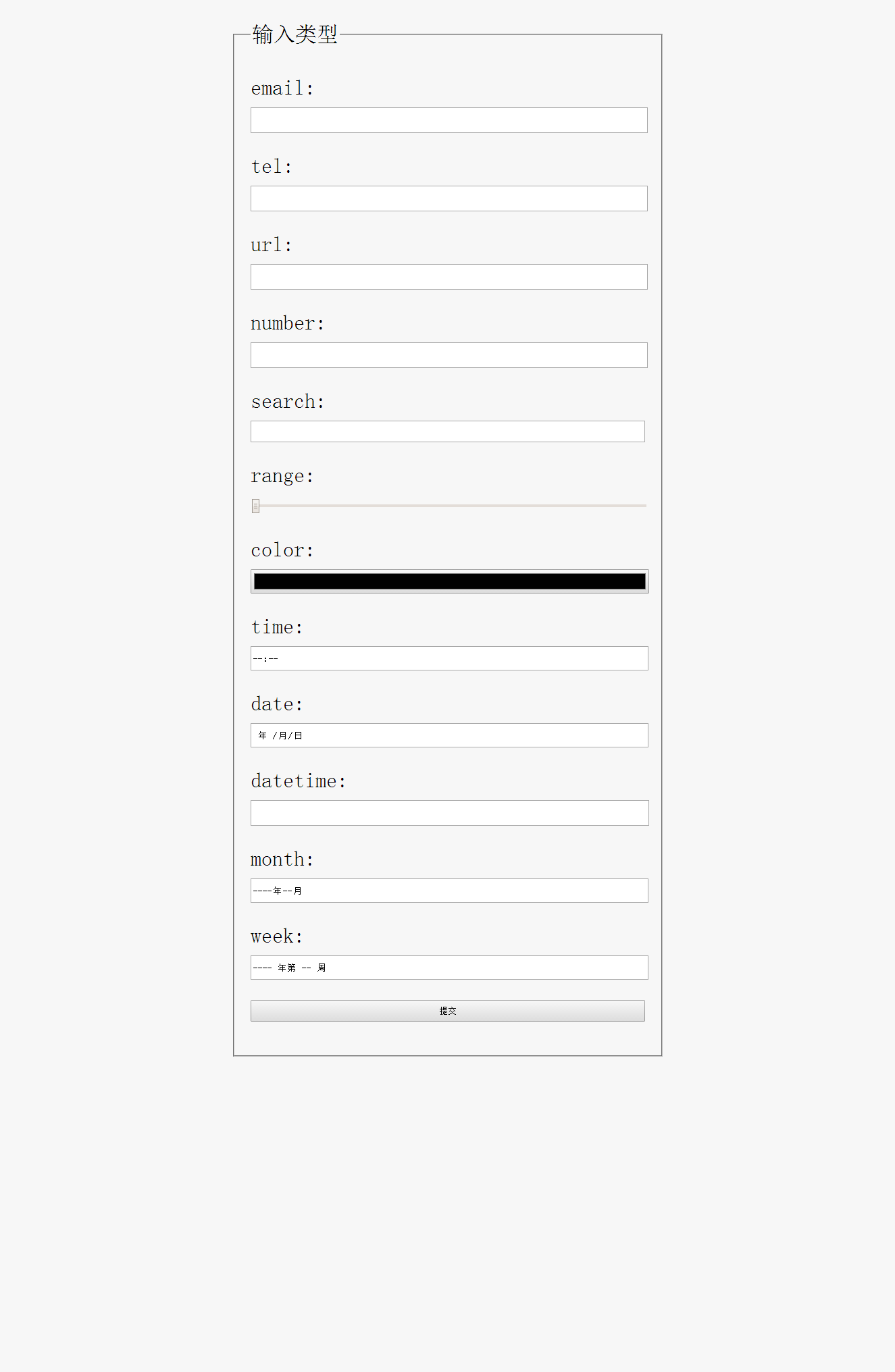
已经学过的表单类型：

text textarea password checkbox radio submit reset file....

|  |  |  |
| --- | --- | --- |
| **类型** | **使用示例** | **含义** |
| **email** | <input type="email"> | 输入邮箱格式(限制用户输入必须是合法的邮箱格式，必须带@符号) |
| **tel** | <input type="tel"> | 输入手机号码格式（手机上会弹出数字键） |
| **url** | <input type="url"> | 输入url格式（限制用户输入必须是以http://打头的合法网址） |
| **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"> | 生成日期控件,并且限制用户输入为日期格式 |
| **datetime** | <input type="datetime"> |  |
| **month** | <input type="month"> | ...限制用户输入为月格式 |
| **week** | <input type="week"> | ...限制用户输入为周格式 |

见示例代码3-1.html

1. <form action="">
2. <fieldset>
3. <legend>输入类型</legend>
4. //label可扩大点击范围，即使点击文字也可获取焦点
5. <label for="demo">
6. email: <input type="email" id="demo">
7. </label>
8. <label for="">
9. tel: <input type="tel">
10. </label>
11. <label for="">
12. url: <input type="url">
13. </label>
14. <label for="">
15. number: <input type="number" step="4" min="8" max="16">
16. </label>
17. <label for="">
18. search: <input type="search">
19. </label>
20. <label for="">
21. range: <input type="range" value="0">
22. </label>
23. <label for="">
24. color: <input type="color">
25. </label>
26. <label for="">
27. time: <input type="time">
28. </label>
29. <label for="">
30. date: <input type="date">
31. </label>
32. <label for="">
33. datetime: <input type="datetime">
34. </label>
35. <label for="">
36. month: <input type="month">
37. </label>
38. <label for="">
39. week: <input type="week">
40. </label>
41. <label for="">
42. <input type="submit" value="提交">
43. </label>
44. </fieldset>
45. </form>



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。



## 3.3 表单属性

见示例代码3-3.html

|  |  |  |
| --- | --- | --- |
| **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(默认="on" 也可写在form标签上 全部关闭) |
| **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 |
| placeholder=”....” 该属性对应的值默认会在input中占位，在用户不需要时自动隐藏 | | |
| required 填写了该属性的表单元素，会在提交form时做验证，必须不为空，才能顺利提交数据 | | |
| autofocus 使用了该属性的表单元素会在页面一加载时，自动获得焦点，省去用户单击文本获取焦点这一步，提升用户体验 | | |

## 

Ps:

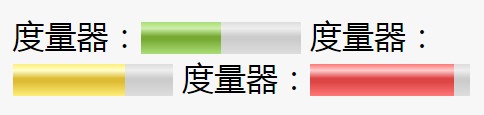
1. xx.select(); //可以全选表单内容
2. xx.focus(); //可以让表单元素聚焦

## 3.4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

1. <form action="">
2. <label for="">
3. 姓名：<input type="text" name="name">
4. </label>
5. <label for="">
6. 省份：<select name="area" id="">
7. <option value="1">河北省</option>
8. <option value="2">山西省</option>
9. <option value="3">山东省</option>
10. </select>
11. </label>
12. <label for="">
13. 个人简介：<textarea name="" id="" cols="30" rows="10"></textarea>
14. </label>
15. <label for="">
16. 课程：<input type="text" list="course">
17. </label>
18. <datalist id="course">
19. <option value="php1"></option>
20. <option value="python"></option>
21. <option value="photoshop"></option>
22. <option value="java"></option>
23. <option value="javascrit"></option>
24. <option value="front end"></option>
25. <option value="写代码"></option>
26. </datalist>
27. <label for="">
28. 加密类型：<keygen name="anquan">
29. </label>
30. <label for="">
31. 结果：<output>一些结果</output>
32. </label>
33. <label for="">
34. 度量器：<meter min="0" max="100" low="60" high="80" value="50"></meter>
35. 度量器：<meter min="0" max="100" low="60" high="80" value="70"></meter>
36. 度量器：<meter min="0" max="100" low="60" high="80" value="90"></meter>
37. </label>
38. <label for="">
39. <input type="submit" value="保存">
40. </label>
41. </form>



# 3.5 选项列表(兼容性还存在很多问题)

通过 Input 配合 datalist option 结构可以很方便的实现下拉列表功能。需要注意的是在 Input 元素的身上通过 list 属性指向 datalist 标签（ 给datalist 添加ID属性 ）.

需要使用如下结构创建一个选项列表：

1. <datalist id=”名称”>
2. <option></option>
3. .....
4. </datalist>

再在表单元素身上通过list=”选项列表ID名”让两者产生关系。



04选项列表-104选项列表-2

# 3.6 change事件

当表单元素内的值发生改变时，松开滑动条或者失去焦点时，会触发的事件。

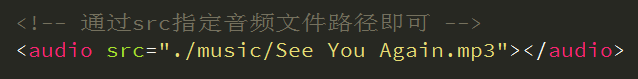
# 多媒体

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



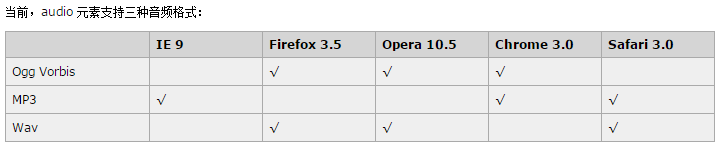
并且可以通过附加属性可以更友好控制音频的播放，如：

autoplay 自动播放

controls 显示控制按钮（默认不显示）

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



## 注意：

（1）对于音频，浏览器只支持3种格式：mp3、wav、ogg；

Mp3兼容最好，不支持mp3的，一定只支持另两个，

所以，要兼容所有浏览器，就准备：

1. 1个mp3格式 + wav|ogg中任一个格式
2. 兼容性写法，也是使用source标签，如：
3. <audio>
4. <source src=”...”/>
5. </audio>
6. 如果需要兼容IE678播放音频，就使用embed标签。
7. <embed src="source/music.mp3"></embed>
8. 隐藏
9. <embed src="source/music.mp3" width="0" height="0"></embed>

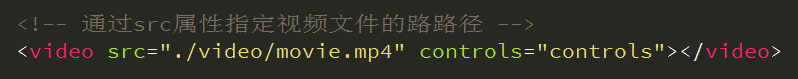
见示例代码4-1.html

1. <!-- html也提供了一个标签来播放音乐 -->
2. <!-- 此种方式↓会导致页面中出现多个控件 不友好 -->
3. <!-- <audio src="./music/See You Again.mp3" controls autoplay loop="3"></audio>
4. <audio src="./music/See You Again.wav" controls autoplay loop="3"></audio>
5. <audio src="./music/See You Again.ogg" controls autoplay loop="3"></audio> -->
6. <!-- 此种方式只会出现1个控件 浏览器自动识别可播放的格式 -->
7. <audio controls autoplay>
8. <source src="./music/See You Again.mp3">
9. <source src="./music/See You Again.wav">
10. <source src="./music/See You Again.ogg">
11. 浏览器不支持HTML5音频播放
12. </audio>

## 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



见示例代码4-2.html

## 注意：

1. h5中视频支持的格式仅有3种：mp4、ogg、webm.
2. Mp4兼容性最好，不支持mp4的浏览器一定支持ogg|webm中任一种，所以，要想视频兼容所有浏览器，就需要准备：
3. Mp4格式 + ogg|webM格式中任一种

## Video标签兼容写法：

1. <video autoplay controls loop>
2. <source src=”...mp4” />
3. <source src=”...webm” />
4. </video>

Ps：当第一个资源不支持时，就去读取第二个资源。

## 控制音、视频的JS函数：

xx.pause(); //暂停

xx.play(); //播放

xx.load(); //重载

Ps：以上函数并没有包含入jquery体系中，所以jq对象无法调用该函数，可以使用

$(...).get(下标)；

的形式将jq对象转换成js对象，使用该函数。

# 键盘按下事件

Keydown

事件对象中存在keyCode属性，使用event.keyCode可以返回一个键值，用以区分不同按键。

# 本地存储

定义：用户可以通过某个浏览器在本地存储数据，目的是为了后面方便使用这些数据，提高用户体验。

h5的本地存储分为：

1. localStorage 和 sessionStorage 两个对象（区域）

区别：

1. localStorage存储的数据会永久驻留在浏览器中，除非你人为删除；
2. sessionStorage存储的数据会在关闭浏览器时，被清空。

存数据：

1. localStorage.setItem(键，值);

取数据：

1. localStorage.getItem(键);

删除数据：

1. localStorage.removeItem(键);

Ps：键和值都可以自定义。

# 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

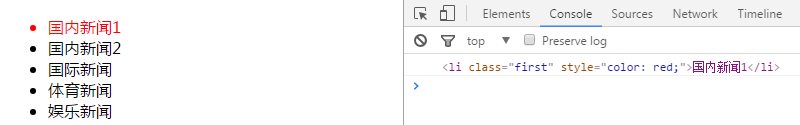
1. <head>
2. <meta charset="UTF-8">
3. <title>获取元素</title>
4. <style>
5. .active {
6. color: red;
7. }
8. </style>
9. </head>
10. <body>
11. <ul>
12. <li>国内新闻</li>
13. <li class="active">国际新闻</li>
14. <li class="active">体育新闻</li>
15. <li>娱乐新闻</li>
16. </ul>
17. <script>
18. // document.getElementById();
19. // document.getElementsByTagName(); 伪数组
20. // 以前是可以获取元素的，但是获取的条件比较单一
21. // HTML5 提供了一些新的DOM获取元素的方法
22. // document.getElementsByClassName(); 伪数组
23. var active = document.getElementsByClassName('active');
24. console.log(active);
25. active[0].style.backgroundColor = 'skyblue';
26. active[1].style.backgroundColor = 'blue';
27. </script>
28. </body>

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。

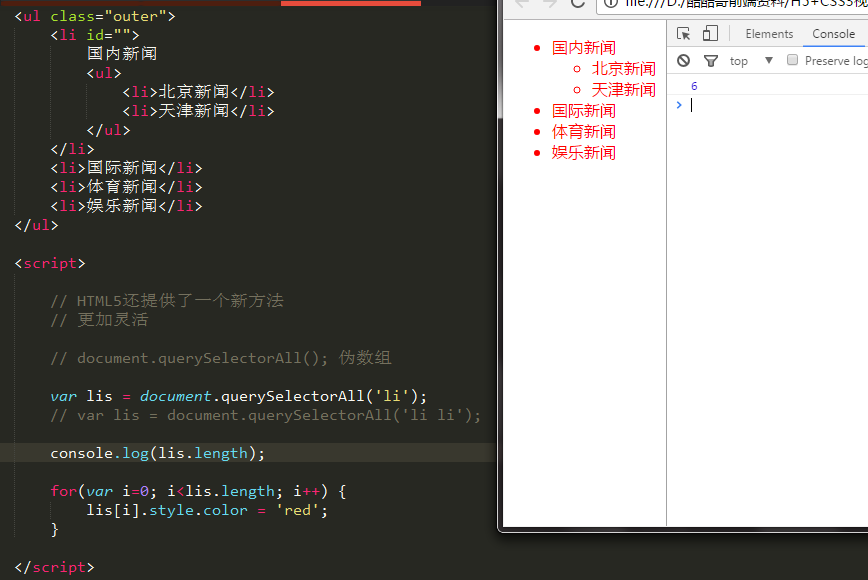
见示例代码5-2.html

1. <ul>
2. <li class="first">国内新闻1</li>
3. <li class="first">国内新闻2</li>
4. <li id="second">国际新闻</li>
5. <li>体育新闻</li>
6. <li>娱乐新闻</li>
7. </ul>
8. <script>
9. // HTML5还提供了更加灵活的获取方式
10. // doucment.querySelector();
11. // 通过CSS选择器来获取元素
12. // 标签、类、ID、后代、子代....
13. // 获取到是符合条件（选择器）的第一个元素
14. var lis = document.querySelector('.first');
15. lis.style.color = 'red';
16. console.log(lis);
17. </script>

效果图：



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

1. <script>
3. // jQuery中 addClass() removeClass() hasClass() toggleClass()
4. // 加载jQuery 是需要时间的
5. // HTML5提供了“类似jQuery”中操作类名的方法
6. // Node.classList.add() // 添加一个类名
7. // Node.classList.remove() // 删除一个类名
8. // Node.classList.contains() // 检测是否存在一个类名
9. // Node.classList.toggle() // 切换一个类名(有则删除，无则添加)
10. // Node指的是一个DOM节点
11. var navs = document.querySelectorAll('nav a'); // 伪数组
12. for(var i=0; i<navs.length; i++) {
13. // 绑订事件
14. navs[i].onclick = function () {
15. var active = document.querySelector('.active');
16. // 移除一个类名
17. active.classList.remove('active');
18. // $('.active').removeClass('active');
19. // 添加一个类名
20. this.classList.add('active');
21. // 相当于下面写法
22. // $(this).addClass('active');
23. //上面removeClass+addClass相当于
24. //$(this).addClass('active').siblings('nav a').removeClass('active')
25. // 切换类名
26. // this.classList.toggle('active');
27. // $(this).toggleClass('active');
28. // 检测是否包含一个类名
29. // alert(this.classList.contains('active'));
30. // $(this).hasClass('active');
31. }
32. }
33. </script>

## 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']

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <meta charset="UTF-8">
5. <title>Tab 标签</title>
6. <style>
7. body {
8. margin: 0;
9. padding: 0;
10. background-color: #F7F7F7;
11. }
13. .tabs {
14. width: 400px;
15. margin: 30px auto;
16. background-color: #FFF;
17. border: 1px solid #C0DCC0;
18. box-sizing: border-box;
19. }
20. .tabs nav {
21. height: 40px;
22. text-align: center;
23. line-height: 40px;
24. overflow: hidden;
25. background-color: #C0DCC0;
26. display: flex;
27. }
28. nav a {
29. display: block;
30. width: 100px;
31. border-right: 1px solid #FFF;
32. color: #000;
33. text-decoration: none;
34. }
35. nav a:last-child {
36. border-right: 0 none;
37. }
38. nav a.active {
39. background-color: #9BAF9B;
40. }
41. .cont {
42. overflow: hidden;
43. display: none;
44. }
45. .cont ol {
46. line-height: 30px;
47. }
48. </style>
49. </head>
50. <body>
51. <div class="tabs">
52. <nav>
53. <!-- 自定义属性格式 data-\* -->
54. <a href="javascript:;" data-name="local" data-my-name="itcast" data-id="1" data-info="test">国内新闻</a>
55. <a href="javascript:;" >国际新闻</a>
56. <a href="javascript:;" >体育新闻</a>
57. <a href="javascript:;" >娱乐新闻</a>
58. </nav>
59. <section class="cont" id="local">
60. <ol>
61. <li>河感在生矿难，死伤在全10</li>
62. <li>禽流感在感在广1处继续蔓延，温家宝指示</li>
63. <li>南方大旱，农作物减产绝收面积上亩</li>
64. <li>猪流感在广在全国暴发</li>
65. <li>禽流感在全国多处继续蔓延，温家宝指示</li>
66. <li>南方大旱，农作物减产绝收面积上亩</li>
67. <li>猪流感在广东群体性暴发</li>
68. </ol>
69. </section>
70. <section class="cont" id="global">
71. <ol>
72. <li>河南再次发生矿难，死伤人数超过100</li>
73. <li>禽流感次发生蔓延，温家宝指示</li>
74. <li>南方大旱，农作物减产绝收面积上亩</li>
75. <li>猪流感在广减产绝收发</li>
76. <li>禽流感在全国多作物减产绝收面积上亩</li>
77. <li>猪流感在广东群体性暴发</li>
78. </ol>
79. </section>
80. <section class="cont" id="sports">
81. <ol>
82. <li>河南再次发生矿难，死伤人数超过100</li>
83. <li>禽流感在全国多处农作物农延，温家宝指示</li>
84. <li>南方大旱，农作物减产绝收面积上亩</li>
85. <li>猪流感在广东群体性暴发</li>
86. <li>禽流感在全农作物继续蔓延，温家宝指示</li>
87. <li>南方大农作物减产绝收面积上亩</li>
88. <li>猪流感在广东群体性暴发</li>
89. </ol>
90. </section>
91. <section class="cont" id="funny">
92. <ol>
93. <li>河南再次发生矿难，死伤人数超过100</li>
94. <li>禽流感在全国多处农作物农延，温家宝指示</li>
95. <li>南方大旱，农作物减产绝收面积上亩</li>
96. <li>猪流感在广东群体性暴发</li>
97. <li>禽流感在全农作物继续蔓延，温家宝指示</li>
98. <li>南方大农作物减产绝收面积上亩</li>
99. <li>猪流感在广东群体性暴发</li>
100. </ol>
101. </section>
102. </div>
103. <img src="" alt="">
104. <input type="text" id="demo" value="123">
105. <div id="demo1" data-info="demo1" data-id="10" data-name="itcast"></div>
106. <script>
107. // $('div').data();
108. // 1、什么叫自定义属性
109. // 2、我们可以自定来定义一些属性，格式data-\*
110. // 3、特点，自定义的属性全部必须以data-开头，只要符合这种格式就称为自定义属性
111. var nav = document.querySelector('nav a');
112. // 读取一个自定义属性值
113. // alert(nav.getAttribute('data-name'));
114. // alert(nav.getAttribute('data-id'));
115. // 设置一个自定义属性
116. // nav.setAttribute('data-demo', '123');
117. // **HTML5提供了一个对象dataset，可以获取所有的自定义属性**
118. console.log(nav.dataset);
119. var customData = nav.dataset;
120. // console.log(customData.name);
121. // console.log(customData.id);
122. // var obj = {
123. // name: 'local',
124. // id: 1
125. // }
126. // // obj.name = 'local1';
127. // obj.demo = '123';
128. customData.name = 'local1';
129. // customData.myName;
130. customData.demo = 123;
131. customData.testDemoFab = 'test-demo';
132. // 总结：
133. // 1、获取一个DOM节点，并且这个DOM节点定义了一些自定义属性
134. // 2、通过Node.dataset，也管理(读取/设置)自定义属性
135. // 3、Node.dataset.属性
136. // 4、Node.dataset.属性 = '属性值'
137. </script>
138. </body>
139. </html>