# 一 HTML5简介

## 1 HTML5的诞生

HTML5并不仅仅只是做为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，还提供了一些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。

广义概念：HTML5代表浏览器端技术的一个发展阶段。在这个阶段，浏览器呈现技术得到了一个飞跃发展和广泛支持，它包括：HTML5，CSS3，Javascript，API在内的一套技术组合。

H5范称：HTML + CSS3 + JS

## 2 HTML5新特性

从HTML5诞生到HTML5规范制定完成，花费了10年时间，HTML5为我们的网页更新或新增了如下特性：

新的**语义化标签**，新的**表单**，**多媒体(视频与音频)**，**canvas绘图**，**酷炫的CSS特效**，**数据存储**，**离线应用**，**地理定位**等等一些好玩的特性。

## 3 HTML5语法变化

1-标签不再区分大小写

2-元素可以省略结束标签

空元素语法的元素：area，base，br，col，hr，img，input，link，mata等

可以写为：<img /> <img> 不允许写为<img></img>

可以省略结束标签的元素：dt，dd，li，option，p，thead，tbody，tr，td，t等

可以省略全部标签的元素：html，head，body，tbdy

3-允许省略属性名

4-允许省略属性值的引号

5-删除了少量的元素和属性，主要删除的是文档样式相关的各种元素和属性，如 font，width，HTML5推荐使用CSS来控制。删除了<table>和<td>原有的align，bgcolor，valign等属性，删除了元素<frame><frameset><noframes>，但是保留了<iframe>。

## 4 HTM5新增的元素

1-文档结构元素：

<article>:代表一片文章，内部可包含<article><header><footer><section>

<section>:页面内容进行分块，内部可包含：<h1><article><section>

<nav>：导航条

<aside>:侧边栏

<header><footer>:头部和脚注

<hgroup>:组织多个<h1>这样的标题

<figure>:表示一个独立的图片区域，包含多个图片标签，<figcaption>表示图片区域标题

2-语义元素

<mark>:显示页面中重点关注的内容，浏览器通常使用黄色显示

<time>：标注时间，向用户展示，datatime属性：标注时间，向计算机展示，如果<time>所包含的内容本身就满级标准的时间格式，那么可以不指定datatime属性

<time datetime="2012-04-01">2012年4月</time>

<time>2012-02-03</time>

3-Chrome独有的元素：<details><summary>前者显示一个详细信息，后者显示摘要

3-两个特殊功能元素：

<meter>:表示一个已知最大值和最小值的技术仪表，如电池剩余电量。

<progress>:表示进度条

## 5 HTML5新增的属性

1-HTML5为大部分元素增加了contentEditable属性，如果将该属性设为true，那么浏览器会允许开发者直接编辑该HTML元素里的内容，且如果一个HTML元素的父元素是可编辑的，那么他自己默认也是可编辑的。此外，还提供了isContentEditable属性，用返回值true、false来判断。

2-designMode属性相当于一个全局的contentEditable属性，如果把整个页面的designMode设置为on，那么所有支持编辑的元素都会变为可编辑状态，除非设置为off，如下所示：

<body ondbclick="document.designMode='on'">

3-HTM5为所有元素都提供了hidden属性。hidden属性可以代替CSS样式中的display属性，设置hidden=true相当于display：none

4-spellcheck属性：HTML为<input><textarea>等元素增加了spellcheck属性，支持true、false属性值，以判断是否需要浏览器对文本进行校验，如：对拼错的单词进行提示。

## 6 HTML5结构

文档结构：

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Document</title>

</head>

<body>

</body>

</html>

文档类型声明：



字符编码：



此外头部还有:<base>标签，如下，设置了所有链接的开头都是以下网址

<base target="\_blank" href="http://www.aaa.com/">

设置多个meta标签

<meta http-equiv="Expires" content="Sat Sep 27 16:12:33 CST 2009">网页过期时间

<meta http-equiv="Pragma " content="no-cache ">禁止从缓存中读取页面，即无法脱机使用

<meta http-equiv="Refresh" content="2" URL="http://www.a.com"> 一段时间后刷新

<meta http-equiv="Set-cookie" content="name=value expires=Sat Sep 27 16:12:33 CST 2009,path=/">网页过期删Cookie

## 7 HTML语法验证

验证地址：<https://validator.w3.org/>

# 二 语义化标签

## 1 语义化标签作用

语义标签对于我们并不陌生，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理解HTML页面内容。

传统的做法我们或许通过增加类名如class="header"、class="footer"，使HTML页面具有语义性，但是不具有通用性。

HTML5则是通过新增语义标签的形式来解决这个问题，例如<header></header>、<footer></footer>等，这样就可以使其具有通用性。

通过标签就能明白标签中所包含的内容的这样的标签。

1. 可以让文档更加清晰简洁
2. 可以让开发者更加容易修改和维护
3. 可以让索搜引擎和残障人士更好的获取网页信息

## 2 常见语义化标签

header 定义 section 或 page 的页眉。页面的头部

nav 定义导航链接。一般定义导航

main 定义主要区域

section 定义文档中的节

aside 定义内容之外的内容 侧边栏

footer 定义 section 或 page 的页脚。

article 定义文章。

mark 定义记号文本，自带可改变的背景色，带UI，不常用

figure 定义媒介内容的分组，以及它们的标题。

figcaption 定义 figure 元素的标题

details 定义元素的细节

summary 定义可见的 <details> 元素的标题。

progress 定义任何类型的任务的进度，带UI，不常用

本质上新语义标签与<div>、<span>没有区别，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>。尽量避免全局使用header、footer、aside等语义标签。

## 3 语义化标签兼容处理

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

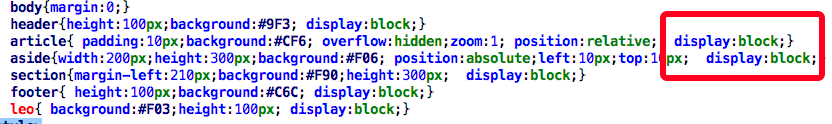
document.createElement('tagName')创建的自定义标签，于是我们的解决方案就是将HTML5的新标签全部通过document.createElement('tagName')来创建一遍，这样IE低版本也能正常解析HTML5新标签了。

注意：创建出来的元素是行内元素，因此一般情况下要给：display:block

<script>

document.createElement('header');

</script>



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三方的一个JS库来解决兼容问题。

引入兼容文件

<script src="html5shiv.js"></script>

<!--[if lte IE 8]>

<script type="text/javascript" src="htmlshiv.js"></script>

<![endif]-->

# 三 表单

## 1 表单标签

<datalist> 数据列表，与input 配合使用

<input type=”text” list=”data”>

<datalist id=”data”>

<option>男</option>

<option>女</option>

<option>不详</option>

</datalist>

<keygen> 生成加密字符串

**keygen 元素**   
  
 keygen 元素的作用是提供一种验证用户的可靠方法。   
 keygen 元素是密钥对生成器（key-pair generator）。当提交表单时，会生成两个键， 一个是私钥，一个公钥。   
  
 私钥（private key）存储于客户端，公钥（public key）则被发送到服务器。公钥可用于之后验证用户的客户端证书（client certificate）。

<output> 不可当做数据提交？

<meter> 表示度量器，不建议用作进度条

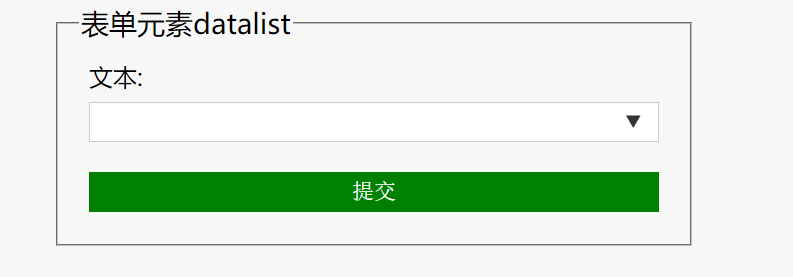
<progress></progress>

<**meter value="81" min="0" max="100" low="60" high="80"** />

Max-width

Min-width





如果添加成功，鼠标移动上去会出现下拉列表

## 2 表单type

email 输入email格式

tel 手机号码，移动设备上得到焦点后会弹出键盘

url 只能输入url格式

number 只能输入数字

search 搜索框，输入内容时候会出现清除按钮

range 范围 滑动条

color 拾色器

time 时间，可控制时间，且有增减按钮

date 日期，可控制日期，且有日期控件！

month 月份，同上

week 星期，同上

--datetime 时间日期

部分类型是针对移动设备生效的，且具有一定的兼容性，在实际应用当中可选择性的使用。

使用案例：

<form>

<label>

邮箱：<input type="email" name="email" class="email">

</label>

<input type="submit" value="提交">

</form>





## 3 表单属性

placeholder 占位符

autofocus 获取焦点

multiple 文件上传多选或多个邮箱地址

autocomplete 自动完成，用于表单元素，也可用于表单自身(on/off)

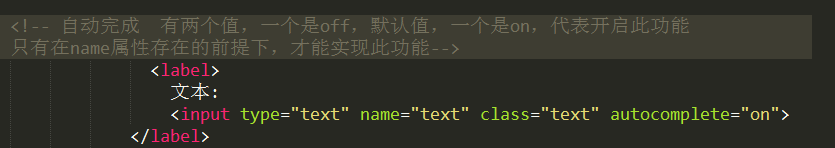
form 指定表单项属于哪个form，处理复杂表单时会需要

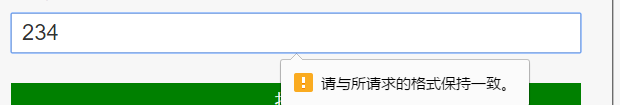
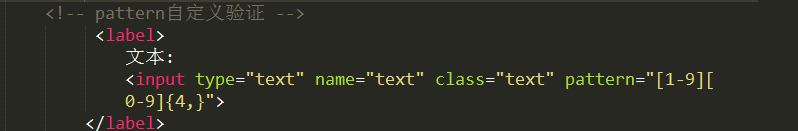
novalidate 关闭验证，可用于<form>标签

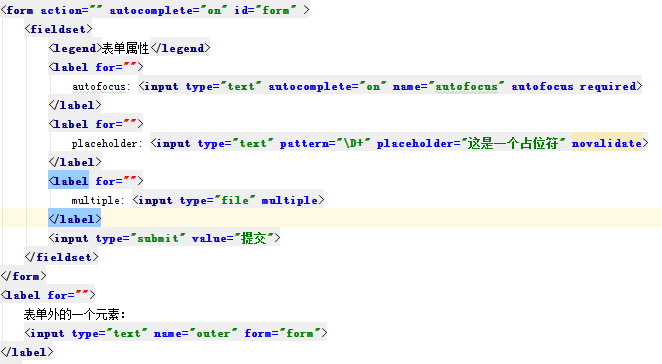
required 必填项

pattern 正则表达式 验证表单

一些讲解：



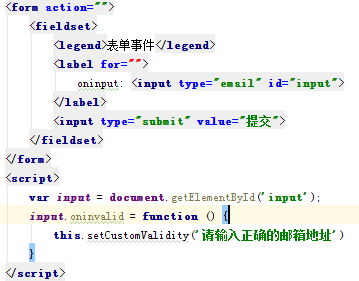




## 4 表单事件

oninput 用户输入内容时触发，可用于移动端输入字数统计

oninvalid 验证不通过时触发

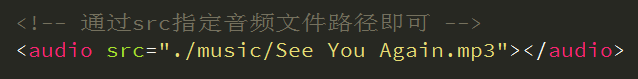


# 四 多媒体

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。

## 1 音频

HTML5通过<audio>标签来解决音频播放的问题，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：

controls

autoplay 自动播放

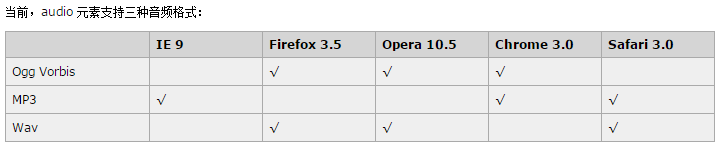
controls 是否显不默认播放控件

loop 循环播放

preload 预加载 同时设置autoplay时些属性失效

auto：加载；none：不加载；metadata：加载元数据（封面、歌词、作者）

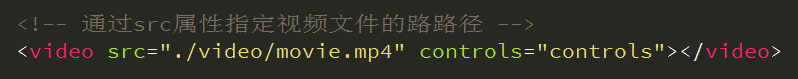
由于版权等原因，不同的浏览器可支持播放的格式是不一样的



**多浏览器支持的方案，如下图**

## 2 视频

HTML5通过<video>标签来解决音频播放的问题。



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

preload 预加载，同时设置了autoplay，此属性将失效

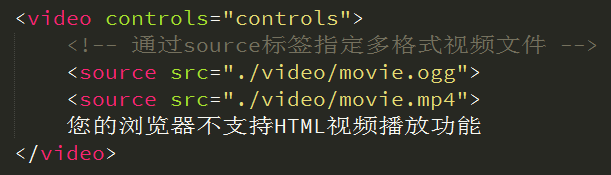
width 设置播放窗口宽度

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



**多浏览器支持的方案，如下图**



## 3 多媒体控制

方法：

load() 加载

play() 播放

pause() 暂停

属性：

currentTime 视频播放的进度

paused 视频播放的状态

duration 视频总时长

事件：

oncanplay 是否能播放

ontimeupdate 播放时触发，报告当前的播放进度

onended 播放完时触发

## 4 全屏

全屏：video.webkitRequestFullScreen();

webkit: webkit RequestFullscreen()

moz: moz RequestFullScreen()

ms: ms RequestFullscreen()

其他: RequestFullscreen()

出现浏览器前缀的原因：为了兼容低版本的自己的浏览器，比如说webkit为了兼容低版本 的chorme。

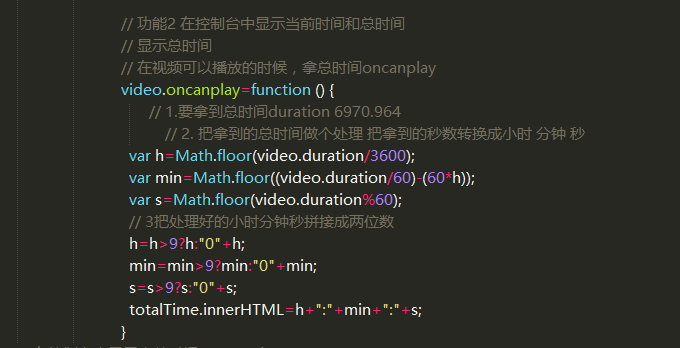
## 5 代码展示

功能1 视频的播放与暂停 播放按钮的样式

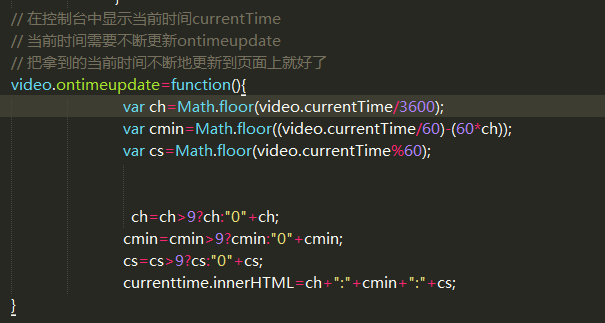


功能2 在控制台中显示当前时间和总时间

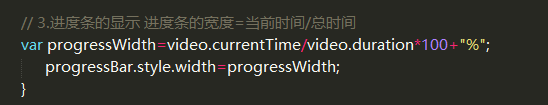
总时间



当前时间



进度条的显示



# 五 DOM扩展

## 1 获取元素

document.getElementsByClassName ('class')

通过类名获取元素，以类数组形式存在。

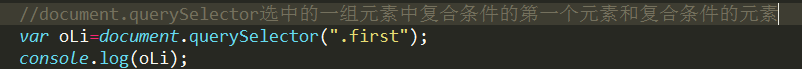
document.querySelector(‘div’)

通过CSS选择器获取元素，符合匹配条件的第1个元素。

document.querySelectorAll('selector')

通过CSS选择器获取元素，以类数组(伪数组)形式存在。

document.querySelector()查询复合该选择器的单个元素





## 2 类名操作

Node.classList.add('class') 添加class

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

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

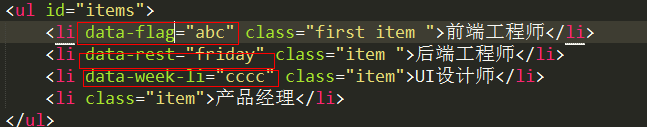
Node.classList.contains('class') 检测是否存在class

Node指一个有效的DOM节点，是一个通称。

## 3 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'] 获取到自定义的属性值。

Node.dataset是以类对象形式存在的

当我们如下格式设置时，则需要以驼峰格式才能正确获取

data-my-name="mine"，获取Node.dataset['myName']



## 4 案例练习 (tab 切换)

