**1.HTML5简介**

**什么是 HTML5？**

HTML5 将成为 HTML、XHTML 以及 HTML DOM 的新标准。

HTML 的上一个版本诞生于 1999 年。自从那以后，Web 世界已经经历了巨变。

HTML5 仍处于完善之中。然而，大部分现代浏览器已经具备了某些 HTML5 支持。

HTML5 是 W3C 与 WHATWG 合作的结果。

为 HTML5 建立的一些规则：

* 新特性应该基于 HTML、CSS、DOM 以及 JavaScript。
* 减少对外部插件的需求（比如 Flash）
* 更优秀的错误处理
* 更多取代脚本的标记
* HTML5 应该独立于设备
* 开发进程应对公众透明

**新特性**

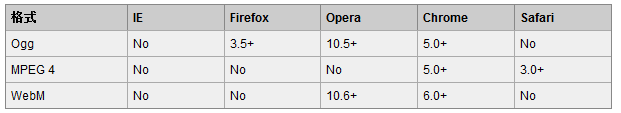
HTML5 中的一些有趣的新特性：

* 用于绘画的 canvas 元素
* 用于媒介回放的 video 和 audio 元素
* 对本地离线存储的更好的支持
* 新的特殊内容元素，比如 article、footer、header、nav、section
* 新的表单控件，比如 calendar、date、time、email、url、search

2.HTML5视频

## 视频格式

当前，video 元素支持三种视频格式：



### 实例

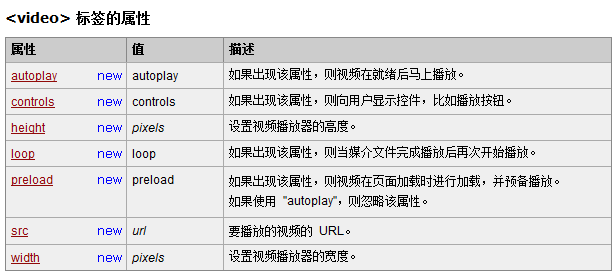
<video width="320" height="240" controls="controls">

<source src="movie.ogg" type="video/ogg">

<source src="movie.mp4" type="video/mp4">

Your browser does not support the video tag.

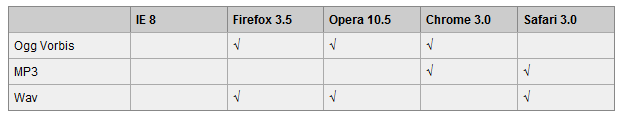
</video>



3.HTML5音频

## 音频格式

当前，audio 元素支持三种音频格式：



### 实例

<audio controls="controls">

<source src="song.ogg" type="audio/ogg">

<source src="song.mp3" type="audio/mpeg">

Your browser does not support the audio tag.

</audio>



4.HTML画布（Canvas）

HTML5 的 canvas 元素使用 JavaScript 在网页上绘制图像。

画布是一个矩形区域，您可以控制其每一像素。

canvas 拥有多种绘制路径、矩形、圆形、字符以及添加图像的方法。

## 创建 Canvas 元素

向 HTML5 页面添加 canvas 元素。

规定元素的 id、宽度和高度：

<canvas id="myCanvas" width="200" height="100"></canvas>

## 通过 JavaScript 来绘制

canvas 元素本身是没有绘图能力的。所有的绘制工作必须在 JavaScript 内部完成：

<script type="text/javascript">

var c=document.getElementById("myCanvas");

var cxt=c.getContext("2d");

cxt.fillStyle="#FF0000";

cxt.fillRect(0,0,150,75);

</script>

JavaScript 使用 id 来寻找 canvas 元素：

var c=document.getElementById("myCanvas");

然后，创建 context 对象：

var cxt=c.getContext("2d");

getContext("2d") 对象是内建的 HTML5 对象，拥有多种绘制路径、矩形、圆形、字符以及添加图像的方法。

下面的两行代码绘制一个红色的矩形：

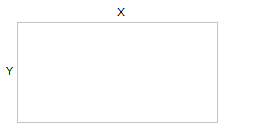
cxt.fillStyle="#FF0000";

cxt.fillRect(0,0,150,75);

fillStyle 方法将其染成红色，fillRect 方法规定了形状、位置和尺寸。

### 实例 - 线条

通过指定从何处开始，在何处结束，来绘制一条线：



JavaScript 代码：

<script type="text/javascript">

var c=document.getElementById("myCanvas");

var cxt=c.getContext("2d");

cxt.moveTo(10,10);

cxt.lineTo(150,50);

cxt.lineTo(10,50);

cxt.stroke();

</script>

canvas 元素：

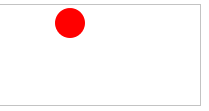
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">

Your browser does not support the canvas element.

</canvas>

### 实例 - 圆形

通过规定尺寸、颜色和位置，来绘制一个圆：



JavaScript 代码：

<script type="text/javascript">

var c=document.getElementById("myCanvas");

var cxt=c.getContext("2d");

cxt.fillStyle="#FF0000";

cxt.beginPath();

cxt.arc(70,18,15,0,Math.PI\*2,true);

cxt.closePath();

cxt.fill();

</script>

canvas 元素：

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">

Your browser does not support the canvas element.

</canvas>

### 实例 - 渐变

使用您指定的颜色来绘制渐变背景：



JavaScript 代码：

<script type="text/javascript">

var c=document.getElementById("myCanvas");

var cxt=c.getContext("2d");

var grd=cxt.createLinearGradient(0,0,175,50);

grd.addColorStop(0,"#FF0000");

grd.addColorStop(1,"#00FF00");

cxt.fillStyle=grd;

cxt.fillRect(0,0,175,50);

</script>

canvas 元素：

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">

Your browser does not support the canvas element.

</canvas>

### 实例 - 图像

把一幅图像放置到画布上：



JavaScript 代码：

<script type="text/javascript">

var c=document.getElementById("myCanvas");

var cxt=c.getContext("2d");

var img=new Image()

img.src="flower.png"

cxt.drawImage(img,0,0);

</script>

canvas 元素：

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">

Your browser does not support the canvas element.

</canvas>

4.HTML5 Web存储

**在客户端存储数据**

HTML5 提供了两种在客户端存储数据的新方法：

* localStorage - 没有时间限制的数据存储
* sessionStorage - 针对一个 session 的数据存储

之前，这些都是由 cookie 完成的。但是 cookie 不适合大量数据的存储，因为它们由每个对服务器的请求来传递，这使得 cookie 速度很慢而且效率也不高。

在 HTML5 中，数据不是由每个服务器请求传递的，而是只有在请求时使用数据。它使在不影响网站性能的情况下存储大量数据成为可能。

对于不同的网站，数据存储于不同的区域，并且一个网站只能访问其自身的数据。

HTML5 使用 JavaScript 来存储和访问数据。

## localStorage 方法

localStorage 方法存储的数据没有时间限制。第二天、第二周或下一年之后，数据依然可用。

如何创建和访问 localStorage：

### 实例

<script type="text/javascript">

localStorage.lastname="Smith";

document.write(localStorage.lastname);

</script>

[亲自试一试](mk:@MSITStore:E:\WEB\W3school.chm::/tiy/t.asp-f=html5_webstorage_local.htm)

下面的例子对用户访问页面的次数进行计数：

### 实例

<script type="text/javascript">

if (localStorage.pagecount)

{

localStorage.pagecount=Number(localStorage.pagecount) +1;

}

else

{

localStorage.pagecount=1;

}

document.write("Visits "+ localStorage.pagecount + " time(s).");

</script>

[亲自试一试](mk:@MSITStore:E:\WEB\W3school.chm::/tiy/t.asp-f=html5_webstorage_local_pagecount.htm)

## sessionStorage 方法

sessionStorage 方法针对一个 session 进行数据存储。当用户关闭浏览器窗口后，数据会被删除。

如何创建并访问一个 sessionStorage：

### 实例

<script type="text/javascript">

sessionStorage.lastname="Smith";

document.write(sessionStorage.lastname);

</script>

[亲自试一试](mk:@MSITStore:E:\WEB\W3school.chm::/tiy/t.asp-f=html5_webstorage_session.htm)

下面的例子对用户在当前 session 中访问页面的次数进行计数：

### 实例

<script type="text/javascript">

if (sessionStorage.pagecount)

{

sessionStorage.pagecount=Number(sessionStorage.pagecount) +1;

}

else

{

sessionStorage.pagecount=1;

}

document.write("Visits "+sessionStorage.pagecount+" time(s) this session.");

</script>

[亲自试一试](mk:@MSITStore:E:\WEB\W3school.chm::/tiy/t.asp-f=html5_webstorage_session_pagecount.htm)

5.HTML5输入类型

**HTML5 新的 Input 类型**

HTML5 拥有多个新的表单输入类型。这些新特性提供了更好的输入控制和验证。

本章全面介绍这些新的输入类型：

* email
* url
* number
* range
* Date pickers (date, month, week, time, datetime, datetime-local)
* search
* color

## Input 类型 - email

email 类型用于应该包含 e-mail 地址的输入域。

在提交表单时，会自动验证 email 域的值。

## Input 类型 - url

url 类型用于应该包含 URL 地址的输入域。

在提交表单时，会自动验证 url 域的值。

## Input 类型 - number

number 类型用于应该包含数值的输入域。

## Input 类型 - range

range 类型用于应该包含一定范围内数字值的输入域。

range 类型显示为滑动条

**Input 类型 - Date Pickers（数据检出器）**

HTML5 拥有多个可供选取日期和时间的新输入类型：

* date - 选取日、月、年
* month - 选取月、年
* week - 选取周和年
* time - 选取时间（小时和分钟）
* datetime - 选取时间、日、月、年（UTC 时间）
* datetime-local - 选取时间、日、月、年（本地时间）

## Input 类型 - search

search 类型用于搜索域，比如站点搜索或 Google 搜索。

search 域显示为常规的文本域。

6.HTML5表单元素

**HTML5 的新的表单元素：**

HTML5 拥有若干涉及表单的元素和属性。

本章介绍以下新的表单元素：

* datalist
* keygen
* output

## datalist 元素

datalist 元素规定输入域的选项列表。

列表是通过 datalist 内的 option 元素创建的。

如需把 datalist 绑定到输入域，请用输入域的 list 属性引用 datalist 的 id：

**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）。

目前，浏览器对此元素的糟糕的支持度不足以使其成为一种有用的安全标准。

## output 元素

output 元素用于不同类型的输出，比如计算或脚本输出

7．HTML5表单属性

## HTML5 的新的表单属性

### 新的 form 属性：

* autocomplete
* novalidate

### 新的 input 属性：

* autocomplete
* autofocus
* form
* form overrides (formaction, formenctype, formmethod, formnovalidate, formtarget)
* height 和 width
* list
* min, max 和 step
* multiple
* pattern (regexp)
* placeholder
* required