

# HTML5新特性

## ( 1 ) 语义化标签

标签	描述
<code>&lt;header&gt;&lt;/header&gt;</code>	定义了文档的头部区域
<code>&lt;footer&gt;&lt;/footer&gt;</code>	定义了文档的尾部区域
<code>&lt;nav&gt;&lt;/nav&gt;</code>	定义文档的导航
<code>&lt;section&gt;&lt;/section&gt;</code>	定义文档中的节 ( section、区段 )
<code>&lt;article&gt;&lt;/article&gt;</code>	定义页面独立的内容区域
<code>&lt;aside&gt;&lt;/aside&gt;</code>	定义页面的侧边栏内容
<code>&lt;details&gt;&lt;/details&gt;</code>	用于描述文档或文档某个部分的细节
<code>&lt;summary&gt;&lt;/summary&gt;</code>	标签包含 details 元素的标题
<code>&lt;dialog&gt;&lt;/dialog&gt;</code>	定义对话框，比如提示框

语义化标签使得页面的内容结构化，见名知义

## ( 2 ) HTML5 Canvas

HTML5 `<canvas>` 元素用于图形的绘制，`<canvas>` 标签只是图形容器，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。

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

```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.fillStyle = "#FF0000";
ctx.fillRect(0,0,150,75);
```

## ( 3 ) HTML5 拖放

拖放是一种常见的特性，即抓取对象以后拖到另一个位置。在 HTML5 中，拖放是标准的一部分，任何元素都能够拖放。

```
<!-->
<div ondrop="drop(event)" ondragover="allowDrop(event)"></div>

```

被拖动的源对象可以触发的事件：

- ondragstart：源对象开始被拖动
- ondrag：源对象被拖动过程中(鼠标可能在移动也可能未移动)
- ondragend：源对象被拖动结束

拖动源对象可以进入到上方的目标对象可以触发的事件：

- ondragenter：目标对象被源对象拖动进入
- ondragover：目标对象被源对象拖动悬停在上方
- ondragleave：源对象拖动离开了目标对象
- ondrop：源对象拖动在目标对象上方释放/松手

#### ( 4 ) HTML5 地理位置

```
var x = document.getElementById("demo");
function getLocation(){
    if(navigator.geolocation){
        navigator.geolocation.getCurrentPosition(showPosition);
    }else{
        x.innerHTML="该浏览器不支持获取地理位置。";
    }
}
function showPosition(position){
    x.innerHTML="Latitude: "+
        position.coords.latitude + " Longitude: "+ position.coords.longitude;
}
```

#### ( 5 ) HTML5 Audio ( 音频 ) , Video ( 视频 )

HTML5 规定了在网页上嵌入音频元素的标准，即使用 `<audio>` 元素。

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  您的浏览器不支持 audio 元素。
</audio>
```

HTML5 规定了一种通过 `<video>` 元素来包含视频的标准方法。

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  您的浏览器不支持Video标签。
</video>
```

#### ( 6 ) HTML5 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。

color、date、datetime、datetime-local、email、month、number、range、search、tel、time、url、week

```
<input type="range" name="points" min="1" max="10">
Search Google: <input type="search" name="googlesearch">
电话号码: <input type="tel" name="usrtel">
```

## (7) HTML5 表单元素

标签	描述
<code>&lt;datalist&gt;</code>	<code>&lt;input&gt;</code> 标签定义选项列表。请与 <code>input</code> 元素配合使用该元素，来定义 <code>input</code> 可能的值。
<code>&lt;keygen&gt;</code>	<code>&lt;keygen&gt;</code> 标签规定用于表单的密钥对生成器字段。
<code>&lt;output&gt;</code>	<code>&lt;output&gt;</code> 标签定义不同类型的输出，比如脚本的输出。

`<datalist>` 元素规定输入域的选项列表。`<datalist>` 属性规定 `form` 或 `input` 域应该拥有自动完成功能。当用户在自动完成域中开始输入时，浏览器应该在该域中显示填写的选项：使用 `<input>` 元素的列表属性与 `<datalist>` 元素绑定。

```
<input list="browsers">
<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

## (8) HTML5 表单属性

HTML5 的 `<form>` 和 `<input>` 标签添加了几个新属性。

`<form>` 新属性：

- `autocomplete`、`novalidate`

`<input>` 新属性：

- `autocomplete`、`autofocus`、`form`、`formaction`、`formenctype`、`formmethod`、`formnovalidate`、`formtarget`、`height` and `width`、`list`、`min` and `max`、`multiple`、`pattern (regex)`、`placeholder`、`required`、`step`

## (9) HTML5 Web存储

Web Storage DOM API 为Web应用提供了一个能够替代 cookie 的 Javascript 解决方案

- `sessionStorage`—客户端数据存储，只能维持在当前会话范围内。

`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。

- `localStorage`—客户端数据存储，能维持在多个会话范围内。

`localStorage` 对象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。对于大量复杂数据结构，一般使用 `IndexedDB`。

## ( 10 ) HTML5 离线Web应用 ( 离线缓存 )

HTML5 引入了应用程序缓存，这意味着 web 应用可进行缓存，并可在没有因特网连接时进行访问。应用程序缓存为应用带来三个优势：

1. 离线浏览 - 用户可在应用离线时使用它们
2. 速度 - 已缓存资源加载得更快
3. 减少服务器负载 - 浏览器将只从服务器下载更新过或更改过的资源

```
<!-- 下面的例子展示了带有 cache manifest 的 HTML 文档 ( 供离线浏览 ) : -->
<!DOCTYPE HTML>
<html manifest="demo.appcache">
<body>
    The content of the document.....
</body>
</html>
```

**manifest 文件**是简单的文本文件，它告知浏览器被缓存的内容（以及不缓存的内容）。

manifest 文件可分为三个部分：

- **CACHE MANIFEST** - 在此标题下列出的文件将在首次下载后进行缓存
- **NETWORK** - 在此标题下列出的文件需要与服务器的连接，且不会被缓存
- **FALLBACK** - 在此标题下列出的文件规定当页面无法访问时的回退页面（比如 404 页面）

```
CACHE MANIFEST
# 2012-02-21 v1.0.0
/theme.css
/logo.gif
/main.js
NETWORK:
login.php
FALLBACK:
/html/ /offline.html
```

## ( 11 ) HTML5 Web Workers

当在 HTML 页面中执行脚本时，页面的状态是不可响应的，直到脚本已完成。web worker 是运行在后台的 JavaScript，独立于其他脚本，不会影响页面的性能。您可以继续做任何愿意做的事情：点击、选取内容等等，而此时 web worker 在后台运行。（相当于实现多线程并发）

## ( 12 ) HTML5 S-SSE

Server-Sent 事件指的是网页自动获取来自服务器的更新。以前也可能做到这一点，前提是网页不得不询问是否有可用的更新。通过服务器发送事件，更新能够自动到达。例子：Facebook/Twitter 更新、估价更新、新的博文、赛事结果等。EventSource 对象用于接收服务器发送事件通知：

```
var source = new EventSource("demo_sse.php");
source.onmessage = function(event){
    document.getElementById("result").innerHTML+=event.data + "<br>";
};
```

```
<?php
    header('Content-Type: text/event-stream');
    header('Cache-Control: no-cache');
    $time = date('r');
    echo "data: The server time is: {$time}\n\n";
    flush();
?>
```

## ( 13 ) WebSocket