

# HTML5 新特性 Unit01

## 1. 视频

### 1.1 视频格式

浏览器支持的视频格式：`mp4`、`webm`、`ogg`

浏览器	MP4	WebM	Ogg
IE9+	Yes	No	No
Chrome	Yes	Yes	Yes
Firefox	No	Yes	Yes
Safari	Yes	No	No
Opera	No	Yes	Yes

### 1.2 视频标签

#### • 简写方式

```
<video src="" width="宽度" height="高度">
  视频格式不支持的提示文本
</video>
```

#### • 标准方式

```
<video width="宽度" height="高度">
  <source src="视频文件URL地址"/>
  <source src="视频文件URL地址"/>
  ...
  视频格式不支持的提示文本
</video>
```

### 1.3 视频属性

- `controls`，是否显示视频控制按钮，布尔属性
- `autoplay`，控制视频是自动播放（**必须与 `muted` 属性组合使用**），布尔属性
- `muted`，视频播放时是否静音，布尔属性
- `poster`，海报帧的 `URL` 地址

- `loop`，是否循环播放，布尔属性
- `preload`，控制视频的预载入的方式
  - `none`，不缓存视频文件，以节省流量
  - `metadata`，播放前只加载视频的时长、高度、宽度等信息
  - `auto`，浏览器尽可能地下载视频（默认）

## 2. 音频

### 2.1 音频格式

浏览器支持的音频格式

浏览器	mp3	wav	ogg
IE9	Yes	No	No
Chrome	Yes	Yes	Yes
Firefox	No	Yes	Yes
Safari	Yes	Yes	NO
Opera	No	Yes	Yes

### 2.2 音频标签

#### • 简写方式

```
<audio src="音频文件URL地址">
  音频格式不支持的提示文本
</audio>
```

#### • 标准方式

```
<audio>
  <source src="音频文件URL地址">
  ...
  音频格式不支持的提示文本
</audio>
```

### 2.3 音频属性

- `controls`，是否显示音频控制按钮，布尔属性
- `autoplay`，控制音频是自动播放（**必须与 `muted` 属性组合使用**），布尔属性
- `muted`，音频播放时是否静音，布尔属性

- `loop`，是否循环播放，布尔属性
- `preload`，控制音频的预载入的方式
  - `none`，不缓存音频文件，以节省流量
  - `metadata`，播放前只加载音频的时长、高度、宽度等信息
  - `auto`，浏览器尽可能地下载音频（默认）

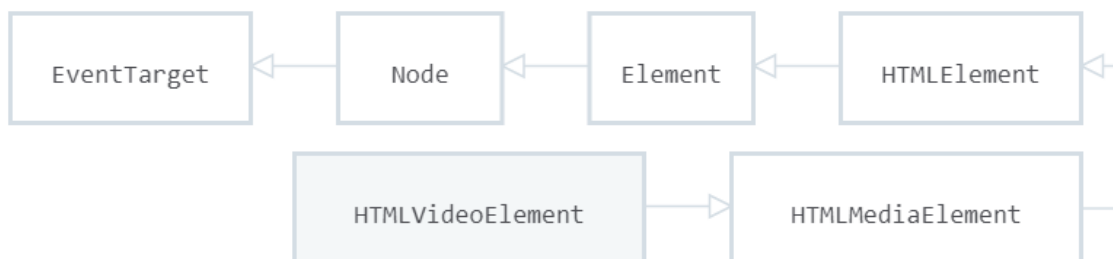
## 2.4 webp 图像格式

webp 图像格式是 Google 开发的图像格式，其体积为 JPG 格式的 2/3，目前 Google、Firefox(>=65) 支持该图像格式。

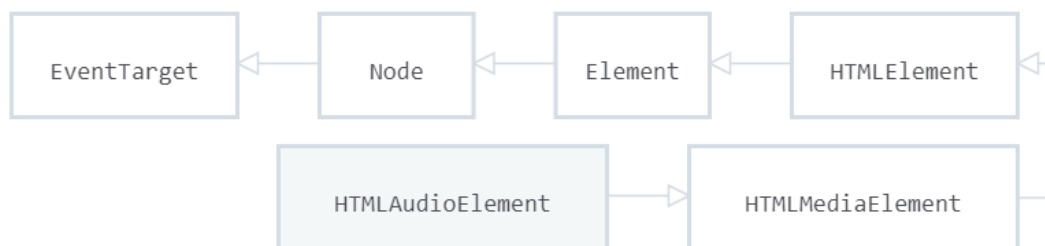
# 3.HTMLMediaElement

## 3.1 基础

HTMLVideoElement 接口提供访问 `<video>` 元素的方法，继承自 HTMLMediaElement 接口



HTMLAudioElement 接口提供访问 `<audio>` 元素的方法，继承自 HTMLMediaElement 接口



## 3.2 属性

### • DOM编程规范(补充知识)

1. 大部分 HTML 标记的属性即 DOM 对象的属性
2. HTML 元素的 `class` 属性在 DOM 编程时采用 `className` 取代（因为 `class` 是 ES 的关键字）
3. HTML 元素中的合成词属性，如 `bgcolor`、`cellpadding` 等，在 DOM 编程时采用驼峰标记法命名，如 `bgColor`、`cellPadding`
4. HTML 元素的 `style` 属性在 DOM 编程时将返回 `CSS2Properties` 或 `CSSStyleDeclaration` 对象
5. 某些 HTML 元素在 DOM 编程时，有自己特有的属性或方法，如 `HTMLSelectElement` 的 `add()` 方法、`remove()` 方法、`value` 属性、`selectedIndex` 属性等
6. HTML 元素的布尔属性在 DOM 编程时采用 `Boolean` 类型表示

HTML tag	DOM interface, properties, and methods
all tags	HTMLElement*: id, title, lang, dir, <b>className</b>
<a>	HTMLAnchorElement*: accessKey, charset, coords, href, hreflang, name, rel, rev, shape, [long] tabIndex, target, type, blur ( ), focus ( )
<applet>	HTMLAppletElement*: align**, alt**, archive**, code**, codeBase**, height**, hspace**, name**, object**, vspace**, width**
<area>	HTMLAreaElement: accessKey, alt, coords, href, [boolean] noHref, shape, [long] tabIndex, target
<base>	HTMLBaseElement: href, target
<basefont>	HTMLBaseFontElement*: color**, face**, size**
<blockquote>, <q>	HTMLQuoteElement: cite
<body>	HTMLBodyElement*: aLink**, background**, <b>bgColor**</b> , link**, text**, vLink**
 	HTMLBRElement: clear**
<button>	HTMLButtonElement: [readonly HTMLFormElement] form, accessKey, <b>[boolean] disabled</b> , name, [long] tabIndex, [readonly] type, value
<caption>	HTMLTableCaptionElement*: align**
<col>, <colgroup>	HTMLTableColElement*: align, ch, chOff, [long] span, vAlign, width
<del>, <ins>	HTMLModElement: cite, dateTime
<dir>	HTMLDirectoryElement*: [boolean] compact**
<div>	HTMLDivElement: align**
<select>	HTMLSelectElement*: [readonly] type, <b>[long] selectedIndex</b> , value, [readonly long] length, [readonly HTMLFormElement] form, [readonly HTMLCollection of HTMLOptionElement] options, [boolean] disabled, [boolean] multiple, name, [long] size, [long] tabIndex, <b>add ( ), remove ( ), blur ( ), focus ( )</b>

## • autoplay

autoplay 属性用于控制媒体对象是否自动播放，语法结构是：

```
//获取媒体对象的autoplay属性
variable = HTMLMediaElement.autoplay

//设置媒体对象的autoplay属性
HTMLMediaElement.autoplay = boolean value
```

## • controls

controls 属性用于控制媒体对象是否显示媒体控制按钮，语法结构是：

```
//获取媒体对象的controls属性
variable = HTMLMediaElement.controls

//设置媒体对象的controls属性
HTMLMediaElement.controls = boolean value
```

## • muted

muted 属性控制媒体对象是否在播放时静音，其语法结构是：

```
//获取媒体对象的muted属性
variable = HTMLMediaElement.muted

//设置媒体对象的muted属性
HTMLMediaElement.muted = boolean value
```

## • loop

loop 属性控制媒体对象是否循环播放，其语法结构是：

```
//获取媒体对象的loop属性
variable = HTMLMediaElement.loop

//设置媒体对象的loop属性
HTMLMediaElement.loop = boolean value
```

- **volume**

`volume` 属性用于控制媒体音量，其值从 0.0 到 1.0，其语法结构是：

```
//获取媒体对象的volume属性
variable = HTMLMediaElement.volume

//设置媒体对象的volume属性
HTMLMediaElement.volume = double value
```

- **currentTime**

`currentTime` 属性用于获取媒体当前的播放时间，单位为秒，其语法结构是：

```
variable = HTMLMediaElement.currentTime
```

- **duration**

`duration` 属性用于获取媒体的总时长，单位为秒，语法结构是：

```
variable = HTMLMediaElement.duration
```

- **paused**

`paused` 属性用于获取媒体是否正在暂停，其语法结构是：

```
variable = HTMLMediaElement.paused
```

- **playbackRate**

`playbackRate` 属性用于控制媒体的播放速度，1.0 表示正常速度，其语法结构是：

```
//获取媒体的播放速度
variable = HTMLMediaElement.playbackRate

//设置媒体的播放速度
HTMLMediaElement.playbackRate = double value
```

## 3.3 方法

- **play()**

`play()` 方法用于实现媒体的播放，其语法结构是：

```
HTMLMediaElement.play()
```

- **pause()**

`pause()` 方法用于实现媒体暂停，其语法结构是：

```
HTMLMediaElement.pause()
```

## 3.4 事件

- **canplaythrough**

`canplaythrough` 事件在媒体的 `readyState` 变成 `CAN_PLAY_THROUGH` 时触发，其语法结构是：

```
HTMLMediaElement.addEventListener('canplaythrough',function(){
    ...
})
```

- **timeupdate**

`timeupdate` 事件在媒体元素的 `currentTime` 属性发生改变时触发，其语法结构是：

```
HTMLMediaElement.addEventListener('timeupdate',function(){
    ...
})
```

- **play**

`play()` 事件在媒体播放时触发，其语法结构是：

```
HTMLMediaElement.addEventListener('play',function(){
    ...
});
```

- **pause**

pause() 事件在媒体暂停时触发，其语法结构是：

```
HTMLMediaElement.addEventListener('pause',function(){
    ...
});
```



## 4. Canvas

### 4.1 什么是 canvas?

canvas 是一个可以通过脚本语言（主要指 JavaScript）来绘制图形的 HTML 元素，语法结构是：

```
<canvas width="宽度" height="高度">
</cavas>
```

<canvas> 元素默认的宽度为 300 像素，高度为 150 像素

### 4.2 canvas 有什么用?

- 数据可视化，如百度的 Echarts

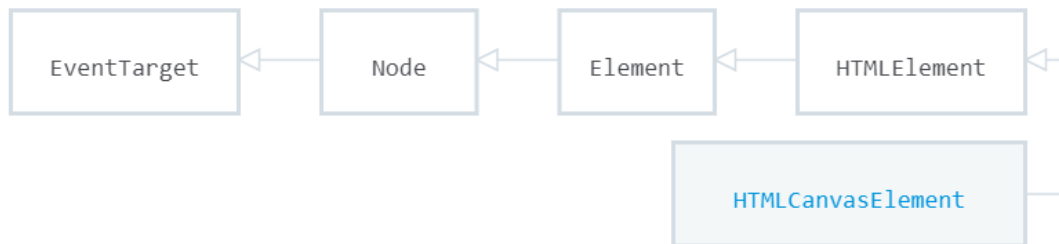
- 游戏画面
- Banner 广告

## 4.3 HTMLCanvasElement

### • 概述

HTMLCanvasElement 接口提供用于操纵 canvas 元素布局的属性和方法

HTMLCanvasElement 接口继承自 HTMLElement 接口



### • 属性

#### • width

width 属性用于获取/设置 <canvas> 元素的宽度，其语法结构是：

```
//获取
variable = HTMLCanvasElement.width

//设置
HTMLCanvasElement.width = number value
```

#### • height

height 属性用于获取/设置 <canvas> 元素的高度，其语法结构是：

```
//获取
variable = HTMLCanvasElement.height

//设置
HTMLCanvasElement.height = number value
```