

# Spracovanie médií

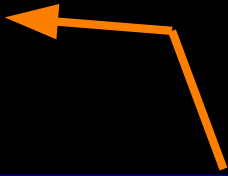
28.04.2022

Marek Nagy

# HTML Media Elementy

- audio

```
<audio controls>
  <source src="zvuk.ogg" type="audio/ogg">
  <source src="zvuk.mp3" type="audio/mpeg">
  Your browser does not support the audio tag.
</audio>
```



„source“ elementy  
dávajú prehliadaču  
výber na rozdiel  
od atribútu „src“.

- video

```
<video width="320" height="240" controls>
  <source src="film.mp4" type="video/mp4">
  <source src="film.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

# Atribúty elementu <audio>

**src** (alebo vnorené elementy **source**)

- zdrojový súbor (URL)
- väčšinou podpora mp3 (WAV, OGG)

**controls**

- zobrazenie ovládača

**autoplay**


- či hneď spustiť

**loop**

- po skončení začať odznovu

**muted**

- či je utlmený zvuk



Príliš sa nedá  
nakonfigurovať vzhľad.  
Je to dané  
prehliadačom.

# Atribúty elementu <video>

**src** (alebo vnorené elementy **source**)

- zdrojový súbor (URL)
- väčšinou podpora formátu MP4 (WebM, Ogg), kodeky mp4: H264, WebM: VP8

**controls**

- zobrazenie ovládača

**autoplay**

- či hneď spustiť

**loop**

- po skončení začať odznovu

**muted**


- či je utlmený zvuk

**height, width**

- výška, šírka elementu prehrávača

**poster**

- obrázok prednáhl'adu (pred nahraním dát videa) (URL)



Niektoré prehliadeče vyžadujú zapnuté **muted** ak má fungovať **autoplay**.



# DOM elementy audio/video

- trieda **HTMLMediaElement** rozširuje **HTMLElement**
- vlastnosti napr.:  
**style, title, classList, ...**
- metódy napr.:  
**addEventListener(), focus(), appendChild(), ...**
- udalosti napr.:  
**'keydown', 'mousemove', ...**

# DOM audio/video

- mapujú sa HTML atribúty:

**src** = url

**controls** = true | false

**autoplay** = true | false

**loop** = true | false

**muted** = true | false

- naviac metódy:

**play()**

- spustí audio/video

**pause()**

- pozastaví prehrávanie

**load()**

- znovu načíta dátový obsah

# DOM video

- vlastnosti:

**height, width**

- rozmery prehrávača v pixeloch

**poster**

- prednáhľad

**videoHeight, videoWidth**

- skutočné rozmery videa v súbore



# Ďalšie vlastnosti DOM audio/video

## duration

- celková dĺžka [sec]

## currentTime

- aktuálna pozícia prehrávania [sec]

## ended

- či skončil [true/false]

## paused

- či je pozastavený [true/false]

## playbackRate

- ako rýchlo sa má prehrávať [1.0 → normálna rýchlosť]

## buffered

- aktuálne načítané časové úseky (TimeRanges)
- sú nenulové, utriedené, neprekrývajúce sa, nedotýkajúce sa

```
let tr = video.buffered;  
for (let i=0; i < tr.length; i++)  
    console.log ('Od=', tr.start(i), 'do=', tr.end(i));
```

# Ďalšie udalosti DOM audio/video

- 'loadedmetadata' - načítané základné parametre/rozmery
- 'progress' - periodicky pri nahrávaní dát
- 'canplay' - možno začať prehrávať  
(dostatočné množstvo načítaných dát)
- 'play' - začalo sa prehrávanie
- 'pause' - pozastavené
- 'ended' - ukončené
- 'timeupdate' - zmena pozície currentTime



# Trieda **MediaStream**

- reprezentuje prúd multimedialneho obsahu
  - zahŕňa viacero video a audio stôp
- vlastnosti:
  - id** – 36 znakový UUID identifikátor objektu
- metódy:
  - getTracks()** – zoznam stôp
  - getAudioTracks()** – zoznam iba audio stôp
  - getVideoTracks()** – zoznam iba video stôp
  - getTrackById(id)** – konkrétna stopa

# Trieda **MediaStreamTrack**

- reprezentuje stopu
- vlastnosti:
  - id** - jednoznačný UUID stopy
  - kind** - 'audio' / 'video'
  - label** - informačný reťazec
- metódy:
  - stop()** - zastavenie stopy

# navigator.mediaDevices

- prístup ku mediálnym zariadeniam
- zoznam možností pomocou enumerateDevice()
  - úplné info až po povolení používateľa

```
try {  
  
  let devlist = await navigator.mediaDevices.enumerateDevices();  
  
  for (let d of devlist) {  
    console.log (`${d.deviceId}, ${d.label}, ${d.kind}`);  
  }  
  
}  
catch (err) {  
  console.log ("The following error occurred: " + err.name);  
}
```

# getUserMedia()

- záznam (MediaStream) z kamery a mikrofónu

```
try {  
  let stream = await navigator.mediaDevices.getUserMedia(  
    {audio:true, video:{width:320,height:240}});  
  
  let video = $('video');  
  video.onloadedmetadata = e => video.play ();  
  
  //v.src = URL.createObjectURL (stream);  
  video.srcObject = stream;  
  
}  
catch (err) {  
  console.log ("The following error occurred: " + err.name);  
}
```

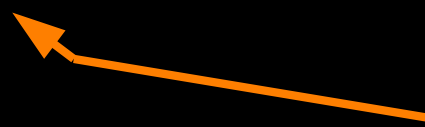
Je možné vyžiadať  
aj konkrétne  
zariadenie cez  
deviceId

Špeciálna  
alternatíva „src“ iba  
pre MediaStream.

# getDisplayMedia()

- záznam (MediaStream) **zdieľaného okna a mikrofónu**

```
try {  
  let stream = await navigator.mediaDevices.getDisplayMedia(  
    {audio:false, video:{cursor:'always'}});  
  
  let video = $('video');  
  video.onloadedmetadata = e => video.play ();  
  
  //v.src = URL.createObjectURL (stream);  
  video.srcObject = stream;  
  
}  
catch (err) {  
  console.log ("The following error occurred: " + err.name);  
}
```



Kedy je viditeľný  
kurzor: always,  
motion, never



# captureStream()

- **canvas** (prípadne **video** / **audio**)
  - vytváranie MediaStream z obsahu elementu za behu
- **canvas**
  - navyše parameter **frameRate**

```
stream = canvas.captureStream (25); //25 FPS
```

# Zastavenie MediaStream-u

- je potrebné zastaviť každú stopu zvlášť

```
let tracks = stream.getTracks ();  
for (let t of tracks) t.stop ();
```



# Trieda **MediaRecorder**

- nahrávanie a kódovanie údajov z **MediaStream-u**
- metódy:
  - start (timeSlice)** - odštartuje „kódovanie“ záznamu v daných časových kúskoch v ms
  - pause ()** - pozastavenie záznamu
  - resume ()** - obnovenie záznamu
  - stop ()** - ukončenie
- udalosti:
  - 'dataavailable'** - sú k dispozícii dáta
  - 'stop'** - bol zastavený nahrávač alebo prerušený stream

Napojenie  
potrebného  
**MediaStream-u**.

```
let buf = [];  
let mime = 'video/webm; codecs="vp8"';  
  
let mr = new MediaRecorder (stream, {mimeType: mime});  
mr.addEventListener ('dataavailable', e => {  
    if (e.data.size > 0) buf.push (e.data);  
});  
  
mr.start (100);
```

Data treba  
potom zlepiť  
dohromady.

# Trieda **MediaSource**

- vytvorenie zdroja dát pre audio/video element
  - postupne možno pridávať dáta
  - nie je dopredu známe trvanie (duration)

- udalosť:

**'sourceopen'**

- objekt je nainicializovaný

- metóda:

**addSourceBuffer()**

- vytvorí buffer kam sa budú ukladať dáta

```
let v = $('video');

let msrc = new MediaSource ();
v.src = URL.createObjectURL (msrc);

msrc.addEventListener ('sourceopen', e => {
    sBuffer = msrc.addSourceBuffer (mime);
});
```

# Trieda SourceBuffer

- metóda:

**appendBuffer (data)**

- zaradí dáta na rozkódovanie
- chvíľu to trvá a tak treba čakať na udalosť

- udalosť:

**'updateend'**

- udalosť po spracovaní dátového balíčka

```
let i = 0;
sBuffer.appendBuffer (data[i]);

// Pridávať možno ďalšie dáta až po udalosti
sBuffer.addEventListener ('updateend', () => {
  sBuffer.appendBuffer (data[++i]);
});
```



# Spracovanie videa v prehliadači

- skopírovať video element do canvas-u

```
ctx.drawImage (video, 0, 0);
```

- získať dáta z canvas-u

```
d = ctx.getImageData (0, 0, 320, 240);
```

- spracovať pixle (R,G,B,A)

```
d.data[i]
```

- niekam inam nakopírovať

```
ctx2.putImageData (d, 0, 0);
```





# HTML5 Web Storage API

- uloženie dát na strane klienta (prehliadača)
- spoločné dáta podľa rovnakej domény
- meno → hodnota (uložená ako utf-16 reťazec)
- cca > 5MB

# Životnosť

- bez expirácie

```
window.localStorage.setItem("lastname", "Smith");  
localStorage.getItem("lastname")  
localStorage.removeItem("lastname")  
localStorage.clear()      // vymaže všetky záznamy  
  
localStorage.lastname = "Smith";  
localStorage.lastname
```

- po uzavretí stránky (záložky prehliadača) sú dáta zrušené

```
window.sessionStorage.setItem("lastname", "Smith");  
sessionStorage.getItem("lastname")  
sessionStorage.removeItem("lastname")  
sessionStorage.clear()    // vymaže všetky záznamy  
  
sessionStorage.lastname = "Smith";  
sessionStorage.lastname
```



Ďakujem za pozornosť