Prikaz multimedijskog sadržaja na webu pomoću HTML5 standarda

Devčić, Ema

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:122:581965

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

University North Digital Repository



Sveučilište
Sjever

Završni rad br. 646/MM/2019

Prikaz multimedijskog sadržaja na webu pomoću HTML5 standarda

Ema Devčić, 0825/336

Varaždin, rujan 2019. godine



Sveučilište
Sjever

Odjel za Multimediju, oblikovanje i primjenu

Završni rad br. 646/MM/2019

Prikaz multimedijskog sadržaja na webu pomoću HTML5 standarda

Student

Ema Devčić, 0825/336

Mentor

Vladimir Stanisavljević, mr.sc.

Varaždin, rujan 2019. godine

Sveučilište Sjever
Sveučilišni centar Varaždin
ul. braće Radića 31, HR-42000 Varaždin

Prijava završnog rada

Definiranje teme završnog rada i povjerenstva

ODJEL: Odjel za multimediju

STUDIJ: preddiplomski stručni studij Multimedija, oblikovanje i primjena

PRISTUPNIK: Ema Devčić | MATIČNI BROJ: 0825/336

DATUM: 16.09.2019. | KOLEGIJ: Programski alati 2

NASLOV RADA: Prikaz multimedijskog sadržaja na webu pomoću HTML5 standarda

NASLOV RADA NA ENGL. JEZIKU: Rendering of a multimedia content on web within HTML5 standard

MENTOR: mr.sc. Vladimir Stanisavljević | ZVANJE: Viši predavač

ČLANOVI POVJERENSTVA:
1. doc.dr.sc. Dean Valdec - predsjednik
2. doc.dr.sc. Andrija Bernik, pred. - član
3. mr.sc. Vladimir Stanisavljević, v.pred. - mentor
4. pred. Snježana Ivančić Valenko, dipl.graf.ing. - rezervni član
5.

Zadatak završnog rada

BROJ: 046/MM/2019

OPIS:

Prikaz multimedijskih sadržaja jedan je od osnovnih zadataka pri izradi suvremenih web aplikacija. Dosad su se u tu svrhu koristili programski dodaci za preglednike poput Adobe Flash-a ili Microsoft Silverlighta ili dodatne JavaScript programske zbirke. Internetski standard HTML5 donosi nove mogućnosti neposrednog ugrađivanja multimedijskih sadržaja u stranice korištenjem novih standardiziranih oznaka.
U radu je potrebno:
• analizirati mogućnost prikaza osnovnih multimedijskih sadržaja u HTML5 standardu
• izraditi pregled mogućnosti pojedinih oznaka za multimedijske sadržaje i prikazati njihovo korištenje u demo stranicama
• analizirati mogućnosti i kompatibilnost suvremenih internetskih preglednika za samostalni prikaz multimedijakih sadržaja i korištenjem standardnih oznaka
Detaljno opisati sve korištene tehnologije i korake u radu potrebne da bi se ostvarilo traženo te detaljno opisati stečena iskustva i postignute rezultate.

ZADATAK URUČEN: 20.09.2019.

Predgovor

Vladimiru Stanisavljeviću mr.sc. zahvaljujem na vrijednim savjetima, za vrijeme i napor koji je uložen u kritičko čitanje ovog uratka.

Zahvaljujem od srca svojim roditeljima Bosiljki i Alfredu te bratu Aronu za ljubav i pomoć koju su mi pružili.

Zahvaljujem svojim profesorima i kolegama Sveučilišta Sjever na stečenom znanju i iskustvu tijekom studiranja.

Sažetak

HTML5 standard je najnovija verzija inačice HTML jezika za opisivanje web aplikacija. Omogućio je veliki napredak prikaza multimedijskog sadržaja te pridonio samoj funkcionalnosti weba. Zamijenio je tehnologije kao što je *Flash* i *Silverligh,* koje su zahtjevale ugradnju vanjskih dodataka. Bitan aspekt HTML-a 5 su njegove oznake pomoću kojih možemo sadržaj nativno ugraditi na web stranice, a da time bude dostupan i isti neovisno o tome koju platformu koristimo. HTML5 obuhvaća tri jezika; HTML koji pruža strukturu stranice, CSS koji definira izgled te JavaScript koji upravlja događajima. Ova inačica donosi poboljšanje starih i unosi nove mogućnosti upravljanja multimedijskim sadržajem, one ne samo da nude bogatu medijsku podršku, već i poboljšanu podršku za stvaranje web aplikacija koje mogu lakše komunicirati s korisnicima.

Multimedija kao integracija različitih medijskih sadržaja je danas svepristuna, svakodnevno smo izloženi multimedijskom sadržaju kao što je tekst, slike, zvuk, audio, video i grafika. Samim time dolazi potreba za *online* distribucijom medijskog sadržaja a odnosi se na sve informacije koje se prikazuju u različitim medijskim formatima koje gledamo, slušamo ili preuzimamo s internetskih stranica.

Ključne riječi: HTML5, multimedija, medij, audio, video, embed, source, picture, track, svg, canvas

Abstract

HTML5 standard is the latest version of the HTML language used to describe web applications. It has made great progress in the display of multimedia content and has contributed to the functionality of the web. It replaced technologies such as *Flash* and *Silverligh*, which requires the installation of external add-ons. An important aspect of HTML5 is its tags that allow us to natively embed content on web pages, making it accessible regardless of which platform we use. HTML5 includes three languages; HTML that provides page structure, CSS that defines layout, and JavaScript that manages events. This version brings improvements to the old and introduces new multimedia content management capabilities, not only offering rich media support but also improved support for creating web applications that can communicate with users more easily.

Multimedia as an integration of various media contents is ubiquitous today, we are exposed to multimedia content such as text, images, sound, audio, video and graphics on a daily basis. Therefore, there is a need for online distribution of media content and refers to all information that is presented in different media formats that we view, listen to or download from web applications.

Keywords: HTML5, multimedia, media, audio, video, embed, source, picture, track, svg, canvas

Popis korištenih kratica

HTML HyperText Markup Language
Sintaksa za obilježavanje hipertektualnih dokumenata.

XML eXtensible Markup Language
Jezik koji definira set pravila za dekodiranje dokumenata na način koji je razumljiv ljudima i uređajima

Web World Wide Web
Usluga na internetu koja omogućuje povezivanje dokumenata i njihovih hiperlinkova

CSS Cascading Style Sheets
Jezik koji opisuje prezentaciju (izgled) HTML dokumenta

API Application Programming Interface
Sučelje koje omogućuje da dva neovisna softverska komponenta izmjenjuju informacije

DOM Document Object Model
API za HTML i XML dokumente, definira logičku strukturu, pristup i manipulaciju dokumenta

DRM Digital Rights Management
Sistematski pristup *copyright* zaštite za digitalne medije

IOS iPhone OS
Operacijski sustav na mobilnim uređajima razvijenih od strane Apple Inc

URL Uniform Resource Locator
Adresa resursa na internetu

MIME Multipurpose Internet Mail Extension
Internet standard koji proširuje podršku formata e-pošte

SVG Scalable Vector Graphics
Format vektorske slike temeljen na XML-u za dvodimenzionalnu grafiku

SWF Shockwave Flash Movie (Small Web Format)
Animacija napravljena u Adobe Flash programu

Sadržaj

1. Uvod 4
2. HTML5 standard 5
   2.1. Povijesni razvoj HTML jezika 6
   2.2. Značajne prednosti 7
3. Multimedija na webu 8
   3.1. HTML5 multimedijske tehnike 8
   3.2. DRM 9
   3.3. HTML5 kao zamjena za Flash i Silverlight 9
4. Prikaz teksta 11
   4.1. Semantički elementi 11
   4.2. Emotikoni 13
      4.2.1. Emotikoni u HTML-u 13
      4.2.2. Emotikoni u CSS-u 15
      4.2.3. Emotikoni u JavaScript-u 15
      4.2.4. Sažetak 15
5. Prikaz različitih medijskih sadržaja 17
   5.1. Embed 17
      5.1.1. Atributi 17
      5.1.2. Konačni kod 17
      5.1.3. Tablica kompatibilnosti 18
   5.2. Source 18
      5.2.1. Atributi 18
      5.2.2. Konačni kod 18
      5.2.3. Tablica kompatibilnosti 19
   5.3. Usporedba 19
6. Prikaz slika 22
   6.1. Picture 22
      6.1.1. Atributi 22
      6.1.2. Tablica kompatibilnosti 22
      6.1.3. Konačni kod 22
7. Prikaz audio i video sadržaja 24
   7.1. Video 24
      7.1.1. Atributi 24
      7.1.2. Formati videa 25
      7.1.3. Odabir video player-a 26
      7.1.4. Tablica kompatibilnosti 28
      7.1.5. Konačni kod 28
   7.2. Audio 34
      7.2.1. Atributi 34
      7.2.2. Tablica kompatibilnosti 34

7.2.3. Formati zvuka ... *35*
7.2.4. Konačni kod ... *36*
7.2.5. Odabir audio player-a ... *37*
7.3. Track ... 38
7.3.1. Atributi ... *39*
7.3.2. Tablica kompatibilnosti ... *39*
7.3.3. Formati kodiranja podnaslova ... *39*
7.3.4. WebVTT struktura podnaslova ... *40*
7.3.5. Konačni kod ... *41*
7.3.6. Uređivači podnaslova ... *41*
7.4. sažetak ... 42
8. Prikaz grafičkog sadržaja ... 43
8.1. SVG ... 43
8.1.1. Prednosti i nedostaci ... *43*
8.1.2. Tablica kompatibilnosti ... *44*
8.1.3. Atributi ... *44*
8.1.4. Konačni kod ... *44*
8.2. Canvas ... 46
8.2.1. Upotreba ... *47*
8.2.2. Atributi ... *47*
8.2.3. Tablica kompatibilnosti ... *47*
8.2.4. Konačni kod ... *48*
8.3. Usporedba ... 50
9. Zaključak ... 51
10. Literatura ... 53
Popis slika ... 55

1. Uvod

HTML5 je najnovija verzija HTML-a (eng. *Hypertext Markup Language*), koda koji opisuje web stranice. 3 vrste koda čine HTML5 - HTML pruža strukturu i arhitekturu web stranice; CSS (eng. *Cascading Style Sheets*) je za prezentaciju (izgled) i JavaScript za događaje. HTML5 je standard koji nam pokazuje kako bi HTML jezik trebao biti oblikovan i korišten u dostavi teksta, slike, multimedije, web aplikacija, pretraživanja i drugih mogućnosti vidljivih u web pregledniku. U danima prije HTML5, osnovni HTML mogao je isporučiti siromašni i jednostavni izgleda web stranica. Ranih devedesetih na webu nije bilo multimedije. Kasnijom pojavom audio i video datoteka internetska tehnologija nije imala mogućnost dostavljanja multimedijalnog sadržaja zbog brzine prijenosa. Daljnji razvoj donekle je ublažio element sporog dostavljanja sadržaja međutim jednom kad je sadržaj stigao morali smo pokrenuti vanjski *player* koji je bio neovisan o web pregledniku. HTML5 se pokazao kao najbolje rješenje za razvoj aplikacija koje rade na svim platformama (eng. *Cross platform development*). Cilj nativne multimedije je da se objekti više ne ugrađuju u web dokument već da se on tretira kao izvorni odnosno nativni kod. Kada govorimo o web-u (eng. *World Wide Web*), danas bilo bi ispravnije upotrebljavati pojam hipermedia umjesto pojma hipertekst s obzirom da su web stranice najčešće multimedijalni, a ne samo tekstualni dokumenti.

HTML5 je u mogućnosti isporučiti nevjerojatnu grafiku, videozapise visoke razlučivosti i još mnogo toga. Jednostavnost uporabe i činjenica da je od početka bio zamišljen kao besplatan i tako dostupan svima razlog je njegove opće prihvaćenosti i popularnosti. Tehnologije povezane s HTML-om 5 čine sadržaj dinamičnijim i interaktivnim te poboljšavaju korisničko iskustvo weba.

2. HTML5 standard

HTML5 je najnoviji standard u razvoju HTML jezika, čija je glavna odlika prikaz multimedijskog sadržaja bez upotrebe programskih dodataka. Omogućuje sve od animacije do aplikacija, glazbe do filmova, a može se koristiti i za izradu nevjerojatno složenih aplikacija koje se pokreću u našem pregledniku. Dizajniran je kako bi bio dostupan svim razvojnim programerima otvorenog weba.

HTML5 sastoji se od tri vrste koda, odnosno jezika:

1. HTML, koji pruža strukturu
2. CSS3, koji vodi računa o prezentaciji
3. JavaScript, koji upravlja događajima [1]

HTML5 ne pripada niti jednoj tvrtki niti pojedinom pregledniku, koriste ga sve velika poduzeća poput *Google*-a, *Microsoft*-a, *Apple*-a i drugih. Uvođenjem nove verzije HTML-a, web je dobio brojne nove mogućnosti, te poboljšanja starih. Donosi nove vizualne, audio i video mogućnosti, može koristiti web kameru i mikrofon za hvatanje slike i zvuka i omogućava manipulaciju grafike. Nove značajke pružaju ne samo bogatu medijsku podršku, već i poboljšanu podršku za stvaranje web aplikacija koje mogu lakše i učinkovitije komunicirati s korisnicima, lokalnim podacima i poslužiteljima. Tolerantniji je na kodne pogreške, može raditi i bez spajanja na internet te postavlja interakciju na potpuno novu razinu [2, 1].

Slijedi popis brojnih resursa HTML5 tehnologije, razvrstanih u nekoliko grupa na temelju njihove funkcije:

1. Semantika; omogućuje preciznije opisivanje sadržaja
2. Povezanost; omogućuje komunikaciju s poslužiteljem novim i inovativnim načinima
3. Izvanmrežnost i pohrana; dopušta web stranicama lokalno pohranjivanje podataka korisnika i učinkovitije upravljanje mrežom
4. Multimedija; stvaranje videa i zvuka na otvorenom webu
5. 2D/3D grafike i efekti; omogućuje puno veći i raznoliki raspon mogućnosti prezentacije
6. Izvedba i integracija; pruža veću brzinu optimizacije i bolju upotrebu računalnog hardvera
7. Pristup uređaju; dopušta korištenje različitih uređaja za unos i izlaz

8. Stiliziranje; dopušta autorima da pišu sofisticirane teme [3]

2.1. Povijesni razvoj HTML jezika

HTML je standard koji se s vremenom mijenja, mogućnosti se proširuju, dodaju se nove funkcionalnosti dok se neke stare ukidaju, standard se stalno razvija.

HTML 1.0 je prva objavljena verzija HTML-a. U to vrijeme nije bilo puno *web developer-a*, a jezik bio je vrlo ograničen, pružao je jedino ispis i manipulaciju tekstom.

HTML 2.0 sadržavao je sve originalne HTML 1.0 specifikacije, ali i nekoliko novih značajki. Koristio se za izradu web stranica do siječnja 1997., a definirao je neke od temeljnih značajki HTML-a po prvi put.

HTML 3.0 donosi mogućnosti definicije tablica uz doradu verzije koja je omogućila prihvaćanje specifičnih oznaka podržanih u tada najvećim i najprihvaćenijim web preglednicima. Sve više ljudi počinje se interesirati za HTML jezik i nastaje potreba za definiranjem novih elemenata. Nastale su brojne duplikacije i postojalo je više oznaka koje su imale istu funkciju.

HTML 3.2 je razvijen kao prvi službeni standard proveden od strane W3C-a (eng. *World Wide Web Consortium*), radne grupe koja je nastala kako bi standardizirala jezik i nastavila njegov daljnji razvoj.

HTML 4.0 predstavljen je 1997. god., nastavio je s prihvaćanjem oznaka nametnutih od strane proizvođača različitih web preglednika, no istovremeno je pokrenuto i "čišćenje" standarda, proglasivši neke od njih suvišnima. Manje promjene u specifikaciji ovog standarda predstavljene su 1999. god., kada je predstavljena verzija HTML 4.1.

HTML 4.1 predstavljao je veliku evolucija HTML standarda u kojem se po prvi put uvodi CSS (eng. *Cascading Style Sheets*) opisni jezik, zadužen za izgled i prezentaciju HTML-a.

XHTML 1.0 nastao je početkom 21. stoljeća i ubrzo postao standardom zajedno s HTML-om 4.1. XHTML kao nova grana HTML-a, dolazi sa setom pravila po kojima je kod morao biti strogo točno napisan da bi radio na preglednicima. Većina oznaka i atributa je ostalo isto, ali promijenile su se neke su rezultirale povećanjem pristupačnosti i funkcionalnosti.

HTML5 prva je nova revizija standarda nastala nakon inačice HTML 4.01. HTML5 donosi brojne nove mogućnosti koje prijašnje verzije standarda nisu imale. HTML5 je najnovija evolucija standarda koji definira HTML jezik [3, 4].

2.2. Značajne prednosti

HTML5 nije vlasnički, što znači da ne moramo platiti autorske naknade za upotrebu. Ovo i eng. *cross-platforma*, što znači da nije značajno upotrebljavamo li eng. *tablet* ili eng. *smartphone, eng. netbook*, prijenosno računalo ili eng. *ultrabook* ili eng. *Smart TV*. Ako naš preglednik podržava HTML5, za očekivati je besprijekorno funkcioniranje što je značajna prednost.

Gledajući unazad dok se HTML jedva mogao nositi s jednostavnim izgledom stranice, HTML5 se može koristiti za pisanje web aplikacija koje još uvijek funkcioniraju i kada nismo povezani s mrežom [5]. Svi veliki pretraživači kao što su Internet Explorer, Edge, Firefox, Chrome, Safari i Opera, Mobile Safari i preglednik Androida podržavaju HTML5, ali ne podržavaju sve iste stvari. Firefox općenito podržava najširi izbor HTML5 značajki, a slijede ga Chrome i Safari [3].

3. Multimedija na webu

Multimedija predstavlja integraciju različitih medijskih sadržaja koji su u osnovi samostalni elementi. Definira se kao informacija predstavljena u kombinaciji teksta, slike, zvuka i videa. Svrha multimedije je da što zanimljivije i realnije prikaže i prenese informaciju ili poruku sa ciljem da zaokupi što više naših osjetila kako bi ju doživjeli što potpunije [6].

Multimedijski pojam odnosi se na više multimedijskih formata, a online multimedija na sve informacije koje se prikazuju u različitim medijskim formatima tako koje gledamo, slušamo ili preuzimamo s internetskih stranica. Multimedija na webu obuhvaća sve slike, zvukove, videozapise, animacije ili grafike koje se šalju iz svijeta putem interneta u naše domove i urede (automobili, zrakoplovi i tako dalje) [5, 7]. Popularnost multimedije može se pripisati razvoju internet tehnologija koje su omogućile pristup širokom krugu korisnika. Multimedija na webu je način prezentacije sadržaja spajanjem različitih medija u jednu zajedničku smislenu cjelinu, s ciljem prenošenja informacije ili poruke na najutjecajniji način. Tako sklopljeni sadržaji daju bogate prezentacijske mogućnosti, korištenjem naprednih tehnoloških rješenja [8].

Multimedija se danas često koristi u promotivne svrhe, danas svaka organizacija može doprijeti do svojih kupaca ili klijenata kroz različite oblike medija te lakšeg objavljivanja i dijeljenja sadržaja, bio on informativan ili promotivan, kao što su e-trgovine, online katalozi i slično. Može se koristiti za razmjenu informacija, u zabavnoj industriji, za prikaz filmova i videa, igre, društvene mreže i u brojne druge svrhe [8, 9].

3.1. HTML5 multimedijske tehnike

Iako su zvuk i video već godinama dio popularnih web lokacija, kao što je npr. *YouTube*, oni su uvijek bili zanemareni i za reprodukciju oslanjali su se na programske dodatke neovisnih proizvođača (npr. *Flash*, *Adobe Flash Player*). Takvi programski dodaci nisu dugoročno najbolje rješenje za vlasnike web lokacija ili za proizvođače preglednika. Mogućnosti reproduciranja zvuka i videa najbolje je ažurirati zajedno s preglednikom, a posebno danas kada proizvođači preglednika sve češće nadograđuju preglednike i objavljuju nove verzije. Važnost HTML5 je uvođenje podrške za reprodukciju zvuka i videa u pregledniku preko elemenata. Budući da su ti elementi dio web platforme, oni će puno bolje funkcionirati nego što je to bilo moguće s dodacima treće strane, upravo to čini HTML5 multimedijske elemente građanima prvog reda na webu [5].

Cilj HTML5 multimedijskih tehnika integracije je pružanje alata koji omogućuju multimedijsko oblikovanje informacija u svojim projektima bez potrebe za podrškom treće strane. Uz sve veću potrebu za eng. *streamingom* videozapisa i glazbe, slijedi i potreba za boljom web integracijom istih.

Multimedija u HTML5 nudi mnoge prednosti, od kojih su glavni ovdje navedeni:

- Nema više potrebe za dodacima (eng. *plugins*)
- Brzina - sve što je prirodno integrirano u preglednik bit će izvedeno i izvršeno brže nego da je uvezeno preko treće strane
- Preglednik omogućava izvorne (ugrađene) kontrole
- Pristupne mogućnosti (tipkovnica, miš) automatski su ugrađene [7]

3.2. DRM

Pojam DRM (eng. *Digital Rights Management*) odnosi se na zbirku sustava koji se koriste za zaštitu autorskih prava elektroničkih medija. On uključuje digitalnu glazbu i filmove, kao i ostale podatke koji se digitalno prenose i pohranjuju. Mogućnost upravljanja digitalnim pravima značajna je izdavačima elektroničkih medija je im pomaže osigurati prikladan prihod za svoje proizvode. Upravljajući trgovinom, zaštitom, nadgledanjem i praćenjem digitalnih medija, DRM pomaže izdavačima ograničiti nezakonito širenje djela zaštićenog autorskim pravima. To se može postići korištenjem digitalnih vodenih žigova ili vlasničkih šifriranja datoteka na medijima koje distribuiraju. Bez obzira koji način izdavači odaberu, osigurano im je da njihov digitalni sadržaj koriste samo oni korisnici koji su platili njegovo korištenje [10].

3.3. HTML5 kao zamjena za Flash i Silverlight

Sredinom devedesetih godina većina web lokacija sadržavala je puno teksta i nekoliko slika. U to vrijeme u upotrebu dolazi *FutureSplash* dodatak koji stranice obogaćuje animacijama i grafikom. *FutureSplash*, međutim nije bio jedini, uskoro mu se pridružuju *Real Player* i *QuickTime*. Ipak budući da je riječ o vlasničkim, a ne otvorenim standardima (eng. *open source*), otvara se rasprava da li bi trebali formirati tako temeljni dio arhitekture weba. Kompanija Macromedia kupila je *FutureSplash* i preimenovala ga u *Flash*, koji uz kasniji *Silverlight*, postaje sveprisutan na webu [11].

Pojavom i sve većom popularnošću, HTML5 je polako započeo zamjenjivati *Flash* i *Silverlight*. Uređaji s IOS-om kao i mnoge druge web lokacije videozapisa prešle su sa *Flash*-a na HTML5 video element ili su barem nudile HTML5 kao opciju. Apple je razvio HTML5 izlog za svoju služenu stranicu, *Google* je razvio HTML5 video *player* za *YouTube*, *Adobe* je najavio da će prestati razvijati *Flash Player* za mobilne uređaje, samo su neki od primjera. Međutim, kako HTML5 ne uključuje tehnologiju upravljanja digitalnim pravima (DRM) kako bi spriječio kopiranje, mnogi vlasnici sadržaja preferiraju vlasničke, *DRM-friendly* formate kao što su upravo *Flash* ili *Silverlight* [1].

Nakon 2020. godine, *Adobe* će prestati objavljivati ažuriranja za *Flash* i mnogi web preglednici ga više neće podržavati. Tvrtke potiču razvojne programere da migriraju svoj softver na moderne programske standarde poput HTML5. U proteklih nekoliko godina, nekoliko web preglednika počelo je zahtijevati od korisnika da omoguće *Flash* prije pokretanja. Na pregledniku Chrom *Flash*-ova upotreba već je drastično pala. U 2014. godini *Flash* je svaki dan koristio 80% korisnika osobnih računala, taj broj je danas ispod 15% [11].

4. Prikaz teksta

4.1. Semantički elementi

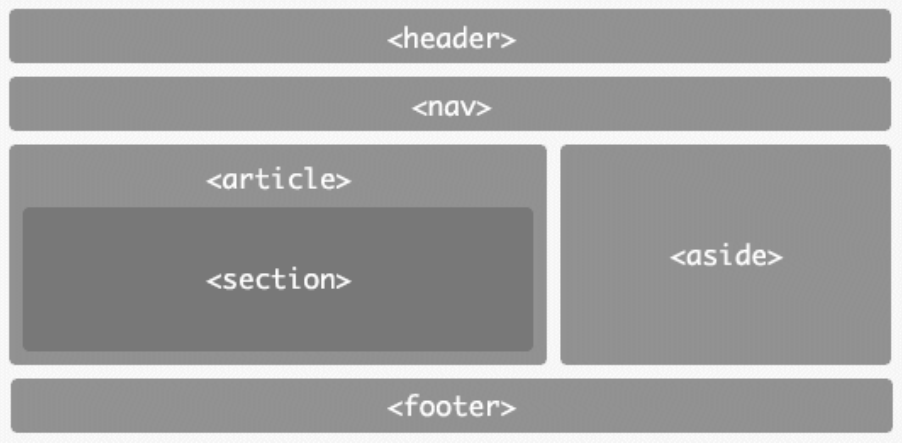
HTML5 uvodi mnoge nove oznake za definiranje semantičkih i strukturnih elemenata, upute za oblikovanje teksta, kontrole oblika, vrste unosa i još mnogo toga.

Popis semantičkih elemenata:

Struktura stranice	Manipulacija teksta
<article>	<details>
<aside>	<figcaption>
<header>	<figure>
<footer>	<mark>
<main>	<summary>
<nav>	<time>
<section>	

Semantički elementi za definiranje strukture i različitih dijelova stranice omogućuju nam lakše čitanje, snalaženje te konzistentniji kod. Isti kod čitaju i tražilice i pomoćne tehnologije (poput čitača zaslona za korisnike s oštećenjem vida) koje preko semantičkih oznaka mogu bolje razumjeti kontekst i sadržaj web lokacije, što znači bolje korisničko iskustvo.

Prikaz semantičkih elemenata za definiranje različitih dijelova web stranice:



Slika 4.1 oznake za strukturu stranice

Pregled oznaka i primjer njihova korištenja:

Article - definira prostor za članak. Lakše je upotrijebiti ovaj element umjesto standardnog p, kada trebamo određeno oblikovanje primijeniti na veću količinu teksta.

```
<article>
<p>Lorem ipsum dolor sit amet.<p>
<p>consectetur adipiscing elit.</p>
</article>
```

Aside - opisuje sadržaj koji je potrebno postaviti na stranu. Može se koristiti kao bočna traka ili dio dodatnog teksta.

```
<aside>
<p> Lorem ipsum dolor sit amet. </p>
</aside>
```

Figure – postavlja prostor za izolirani sadržaj. To mogu biti fotografije, dijagrami, kodovi, videozapisi i slično. ***Figcaption*** – definira naslov elemenata *figure*, s kojim se mora koristiti.

```
<figure>
  <figcaption>Logo</figcaption>
  <img src="image.png" alt="Logo" width="300" height="300">
</figure>
```

Footer – pomaže nam kod boljeg opisivanja sadržaja, omogućuje postavljanje podnožja.

Header – pomaže nam kod boljeg opisivanja sadržaja, omogućuje postavljanje zaglavlja.

Main – opisuje prostor za glavni sadržaj web stranice.

```
<main id="content">
```

Mark – određuje istaknuti tekst ili dio teksta na web stranici.

```
<p>Element mark je koristan kada treba <mark>naglasiti</mark>
važne informacije.</p>
```

Nav – opisuje poseban prostor za navigacijske veze.

```
<nav>
  <ul>
  <li><a href="https://www.w3schools.com/">w3schools</a></li>
  <li><a href="https://stackoverflow.com/">stackoverflow</a></li>
  </ul>
</nav>
```

Section – koristi se za odvajanje pojedinih dijelova sadržaja.

```
<section>
```

```
  <h1></h1>
  <p></p>
</section>
```

Details - opisuje detalje web stranice ili sadržaja općenito. Pojedinosti mogu biti vidljive svima ili skrivene. ***Summary*** – definira naslov za *details* element.

```
<details>
      <summary>Copyright</summary>
      <p>by your company</p>
      <p>Content governed by copyright</p>
</details>
```

Time - služi za određivanje vremena i datuma.

```
<h2>Snimka uživo započinje u <time>21:00</time></h2>
```

[13]

4.2. Emotikoni

Od svojih skromnih početaka 1999. godine emotikoni su danas sveprisutni. Koristimo ih cijelo vrijeme, a gotovo svaka aplikacija za *chat* ili slanje poruka pruža veliku podršku emotikona. Emotikoni su male šarene ikone koje mogu davati dojam da su slike u tradicionalnom smislu, ali u suštini su slova, brojevi, interpunkcijski znakovi i simboli na koje možemo gledati kao tekst.

Pregled osnovnih značajki:

- Emotikoni su znakovi (eng. *charachters*)
- Možemo ih selektirati, kopirati, zalijepiti, prilagoditi njihovu veličinu i sl.
- Imaju primitivni numerički prikaz koji možemo koristiti da bi ih prikazali

4.2.1. Emotikoni u HTML-u

Da bi koristili emotikone u HTML-u, prvi korak koji moramo napraviti jest postaviti kodiranje znakova dokumenta UTF-8. Na taj način smo osigurali konstantni prikaz emotikona na različitim preglednicima i uređajima. Unutar glavne HTML oznake uključimo slijedeću meta oznaku:

```
<meta charset="UTF-8">.
```

Za prikaz emotikona možemo koristit dva načina. Jedan od načina je korištenje emotikona direktno u HTML kodu. Drugi način je specificiranje pojedinog emotikona pomoću primitivnog numeričkog prikazivanja.

1. Direktno korištenje

Direktno korištenje je ujedno i najjednostavniji način prikaza jer uključuje kopiranje i lijepljenje. Potrebna je samo aplikacija ili web stranica koja nam omogućuje da kopiramo odabrani emotikon u svom nativnom obliku. Jedno od tih mjesta je *Emojipedia* (*https://emojipedia.org*)



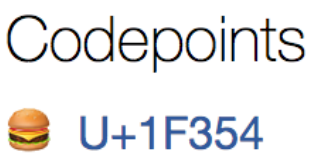
Slika 4.2 opis emotikona Hamburger

Nakon što odaberemo kopiranje, jednostavno ga zalijepimo u HTML kod:

```
<p>  </p>
```

2. Specificiranje koda

Kada izravno određivanje ne radi, tada koristimo nešto zahtjevniji postupak upotrebe numeričkog (eng. *codepoint*) prikaza pojedinog emotikona.



Slika 4.3 numerički kod emotikona Hamburger

Numerički kod kopiramo sa iste stranice koji smo koristili u prvom koraku. Za hamburger emotikon, to je U+1F354. Da bismo ovakvu vrijednost mogli koristiti u kodu, potrebno je malo izmijeniti vrijednosti. Na početak se dodaju znakovi *& # x,* umjesto *U + 1*. Ovo je dobivena forma koda:

```
<p> &#x1F354 </p>
```

4.2.2. Emotikoni u CSS-u

Emotikone možemo koristiti i u CSS-u, na sličan način kao pomoću HTML-a.

1. Direktno korištenje:

```
h1::before  {
  content: "🍔";
}
```

2. Specificiranje koda:

Umjesto *U+* dodajemo /0 (kosa crta i znak nula).

```
h1::before  {
  content: "\01F354";
}
```

4.2.3. Emotikoni u JavaScript-u

Za početak potrebno je definirat da se emotikoni tretiraju kao tekst: `document.body.innerText = "😀";`

Nadalje, koristi se metoda *String.fromCodePoint* koja uzima vrijednost za kodiranje kao argument.

```
document.body.innerText = String.fromCodePoint(0x1F354);
```

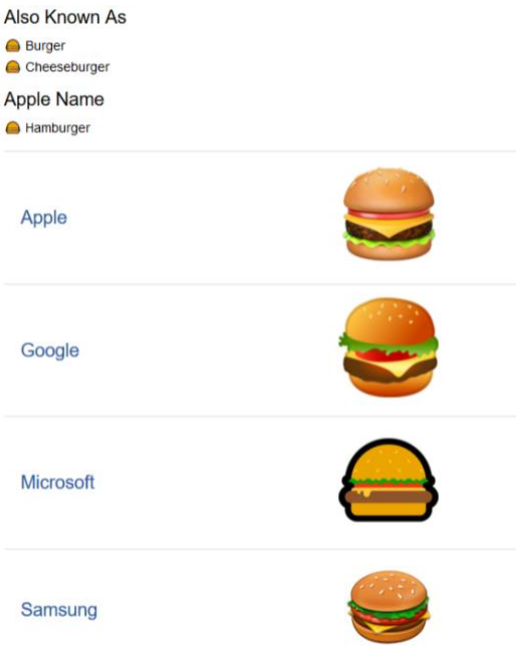
Povratna vrijednost biti će znak na toj lokaciji. Kao što smo napravili izmjene kod HTML-a i CSS-a, umjesto *U + 1* dodajemo *0x* (nula i slovo x).

4.2.4. Sažetak

Može biti zbunjujuće razmišljati o emotikonima kao o znakovima (tekstu), jedan od razloga je taj što mislimo da su znakovi samo ono što nam tipkovnica pruža, zapravo ona prikazuje samo mali postotak ukupnog broja znakova koji su nam dostupni. Emotikoni nisu

jedini koji koriste numeričko predstavljanje, svi znakovi koje koristimo imaju sličnu reprezentaciju. Na zaslonu vidimo znakove kao što su A, B, C, D, kada u stvarnosti oni izgledaju ovako: U+0041, U+0042, U+0043, U+0044. Spomenuli smo da se ovakva reprezentacija naziva *codepoint*, no precizniji naziv bi bio *Unicode Codepoint*. Može se definirati kao industrijski standard koji osigurava da tekst koji vidimo na zaslonu je isti bez obzira na jezik, mogućnosti sustava, operacijski sustav i slično.

Kopiranje i lijepljenje je zasigurno najjednostavniji način implementiranja emotikona u HTML-u, CSS-u I JavaScriptu. Ipak, postojati će situaciju u kojima to neće biti moguće, pa se koriste metode pristupa preko *codepoint*-a. Također treba imati na umu da budući da su emotikoni nativni elementi aplikacije ili platforme, što znači da će izgledati drugačije za različite korisnike.



Slika 4.4 emotikon Hamburger na različitim platformama

Unicode standard osigurava da odgovarajući *codepoint* predstavlja, u ovom slučaju hamburger. Svaka platforma, uređaj ili aplikacija ima potpunu kreativnu slobodu tumačenja istog na koji god način žele. Za neke ovako nešto nije poželjno, pa su umjesto toga svaki emotikon ponovno kreirali u SVG ili u PNG obliku kako bi mogli osigurati dosljednost. Primjer nekoga tko to radi je društvena mreža *Twitter* [14].

5. Prikaz različitih medijskih sadržaja

5.1. Embed

Element *embed* definira spremnik za vanjsku aplikaciju ili interaktivni sadržaj. Upotrebljava se za umetanje sadržaja nestandardnih elemenata, obično multimedijskih elemenata kojih HTML ne podržava. Može se koristiti za dodavanje slika, audio zapisa, video zapisa, grafike i sl. Mnogi web preglednici već duže vrijeme podržavaju oznaku *embed*, međutim, ona nije bila dio HTML4 specifikacije, dodana je kao dio HTML5 standarda i prikazati će se samo na HTML5 stranici [17, 18]. Treba imati na umu da suvremeni preglednici imaju zastarjelu i uklonjenu podršku za dodatke preglednika pa koristimo *embed* oznaku samo kada sadržaj ne možemo prikazati na niti jedan drugi način [3]. Na primjer videozapis u formatu *quicktime*, nije podržan od strane *video* oznake pa ćemo koristiti *embed*.

5.1.1. Atributi

- *Height* - određuje visinu ugrađenog sadržaja u pixselima, vrijednost mora biti apsolutna; postotak nije dopušten
- *Width* - određuje širinu ugrađenog sadržaja u pixelima, vrijednost mora biti apsolutna; postotak nije dopušten
- *Src* - određuje adresu vanjske datoteke za ugrađivanje
- *Type* - određuje vrstu medija ugrađenog sadržaja [3, 17]

5.1.2. Konačni kod

Primjer 1. dodavanje grafike, odnosno medija koji zahtjeva vlasnički dodatak, u ovom slučaju *Flash*. Ako korisnik nema dodatak, sadržaj se neće prikazati.

```
<embed src="helloworld.swf">
```

Primjer 2. dodavanje videozapisa, video će se prikazati samo ako preglednik sadrži dodatak za *quicktime.*

```
<embed type="video/quicktime" src="horses.mov" width="600"
height="400">
```

[3]

5.1.3. Tablica kompatibilnosti

	Google Chrome	Microsoft Edge	Mozilla Firefox	Safari	Opera
<embed>	2.0+	5.5+	1.0+	1.3+	9.2+

[18]

5.2. Source

Element *source* omogućava određivanje više alternativnih medijskih resursa na elementima medija. Odnosno određivanje alternativnih video i audio datoteka koje preglednik može odabrati na temelju svoje podrške vrste medija ili kodeksa. Iako je najčešća upotreba sa *audio* i *video* oznakama može se koristit i uz *picture* te *track* [20].

5.2.1. Atributi

- *Src* - određuje lokaciju audio ili video datoteke, atribut mora biti naveden da bi oznaka bila važeća
- *Srcset* – koristi se uz oznaku *picture*, određuje lokaciju slike koji se može koristiti u različitim situacijama
- *Type* - specificira vrstu ugrađenog sadržaja, odnosno vrstu medijskog resursa. Ako je navedeno, vrijednost mora biti MIME vrsta
- *Media* - određuje vrstu resursa medija, tako da preglednik može utvrditi može li ga reproducirati ili ne. Ako ne, može odabrati da se ne preuzima. Ako je navedeno, vrijednost mora biti važeći medijski upit [20, 21].

5.2.2. Konačni kod

Primjer 1. prikaz videa

```
<video controls>
  <source src="horses.mp4" type="video/mp4">
  <source src="horses.ogg" type="video/ogg">
  Vaš preglednik ne podržava HTML5 video oznaku.
</video>
```

Primjer 2. prikaz slike

```
<picture>
  <source media="(min-width: 650px)" srcset="horses01.jpg">
```

```
  <source media="(min-width: 400px)" srcset="horses02.jpg">
  <img src="horses03.jpg" alt="horses" style="width:auto;">
</picture>
```

[17]

5.2.3. Tablica kompatibilnosti

	Google Chrome	Microsoft Edge	Mozilla Firefox	Safari	Opera
<source>	4.0+	9.0+	3.5+	4.0+	10.5+

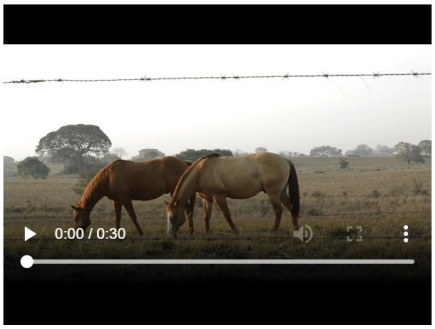
[18]

5.3. Usporedba

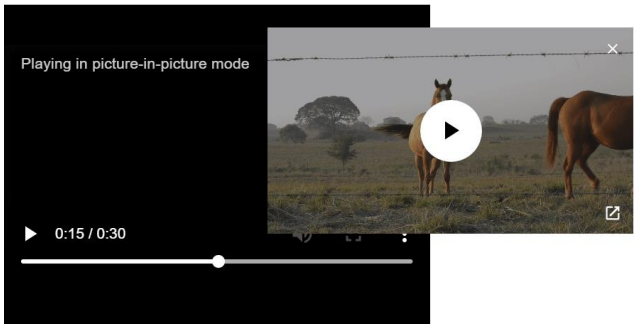
Slijedi prikaz videozapisa od strane *embed* i *source*, u različitim preglednicima:

Chrome

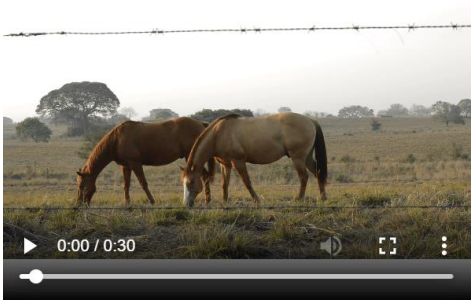
Embed video *player* nudi opciju *picture-in-picture*, koja omogućava manipulaciju videa na način da se prikazuje preko stranice i ostalih aplikacija, te da ga možemo premještati na ekranu. Druge opcije nisu dostupne. Kod *source player*-a moguće je povećati video, odnosno omogućiti prikaz preko cijelog ekrana te uključiti opciju *picture-in-picture*.



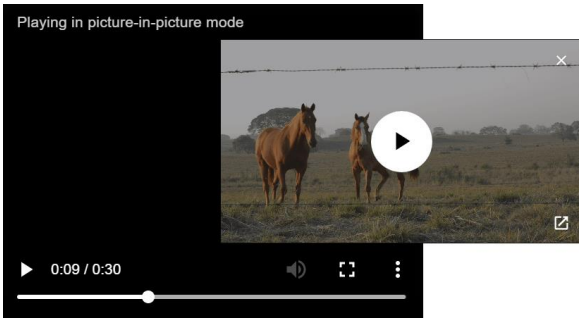
Slika 5.1 embed video player u Chrom-u



Slika 5.2 embed opcija picture-in-picture



Slika 5.4 source video player u Chrom-u



Slika 5.3 source opcija picture-in-picture

Microsoft Edge

Embed video se automatski reproducira dolaskom na stranicu. Može se manipulirati zvukom, odnosno mijenjati glasnoću. Prelaskom kursora na prikaz vremena, video se može ubrzati za 30 sekundi ili vratit unazad 30 sekundi. Kod *source* moguće je namještati jačinu zvuka, te postaviti prikaz preko cijelog ekrana. U ovom slučaju moguće je samo vratiti se za 30 sekundi, ne i ubrzati.



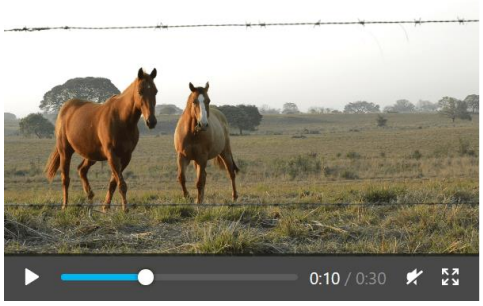
Slika 5.5 embed video player u Edgeu



Slika 5.6 source video player u Edgeu

Mozilla Firefox

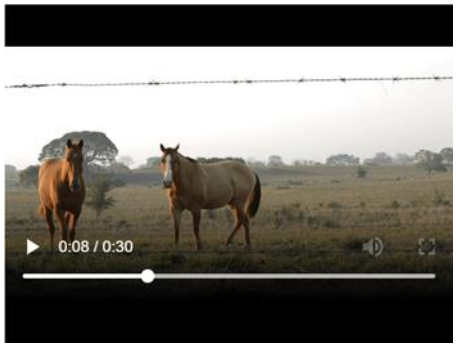
Kod *source player*-a zvuk je automatski postavljen na bešumno i video se može povećati.



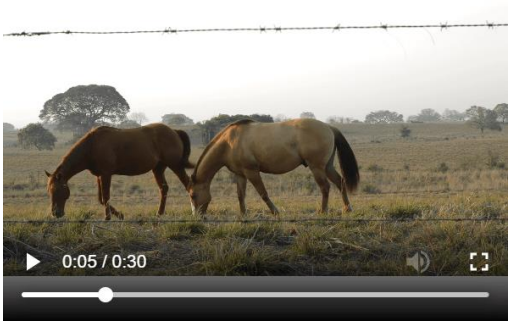
Slika 5.7 source video player u Firefox-u

Opera

Embed video se automatski reproducira i nije omogućena nijedna opcija. Kod *source* jedina radnja koja je dozvoljena je prikaz preko cijelog ekrana.



Slika 5.8 embed video player u Operi



Slika 5.9 source video player u Operi

6. Prikaz slika

6.1. Picture

Oznaka *picture* daje više mogućnosti i veću fleksibilnost kod postavljanja slika na web mjesto. Najčešće se upotrebljava kod prilagodljivih slika, umjesto da se jednu sliku skalira na temelju širine okvira prikaza (eng. *viewport width*), pomoću ove oznake jedna slika može se definirati na više načina i odabrati onaj koji bolje ispunjava digitalni prostor.

6.1.1. Atributi

- *Srcset* – definira lokaciju slike, obavezan atribut
- *Media* – prihvaća sve valjane medijske upite koji bi se inače definirali u CSS-u
- *Sizes* – definira širinu slike
- *Type* – definira vrstu MIME

6.1.2. Tablica kompatibilnosti

	Google Chrome	Microsoft Edge	Mozilla Firefox	Safari	Opera
<picture>	38.0+	13.0+	38.0+	9.1+	25.0+

6.1.3. Konačni kod

Element *picture* sastoji se od dvije različite oznake: jedne ili više *sourc*e i jedne *img* oznake. Prvi *source* element koji će se prikazati ovisi o kompatibilnosti preglednika i općenito postavkama prilagodbe, nakon što prikaže sliku, preglednik će ignorirati sve ostale *source* oznake. Na kraju, *img* se koristi kao osiguranje, da se slika prikaže u slučaju da preglednik ne podržava oznaku *picture* ili ako ne može prikazati sliku prema definiranim postavkama širine.

```
<picture>
   <source media="(min-width: 600px)" srcset="running_horse.jpg">
   <source media="(min-width: 400px)" srcset="white_horse.jpg">
   <img src="brown_horse.jpg" alt="horses" style="width:auto;">
 </picture>
```

[17]



Slika 6.1 prikaz slika u različitim širinama preglednika

Smanjenjem prozora preglednika (što u ovom slučaju imitira prikaz na različitim uređajima), prikazati će se ona slika koja odgovara postavkama dužine koju smo odredili (*min-width*). Nakraju kada više ne odgovara nijedna od opcija, prikazati će se slika u *img* oznaci. Prikaz će biti isti u svim preglednicima koji podržavaju *picture* element.

7. Prikaz audio i video sadržaja

Prije HTML5 nije bilo standarda za reprodukciju audio i video datoteka na web stranici koja općenito zahtijeva dodatke za različite formate audio i video sadržaja. HTML5 pruža standardni način ugradnje takvih datoteka koji sadrže ne vlasničke formate na web stranici. Oznake *audio*, *video* i *source* govore pregledniku da će se povezane informacije dijeliti kao HTML5 kompatibilni *stream*-ovi. To korisnicima omogućuje pregled i slušanje audio i video zapisa ugrađenog na stranicu bez upotrebe dodatka treće strane. Ovi elementi su međusobno povezani i moramo ih promotriti pojedinačno te kao cjelinu da bismo ih razumjeli.

7.1. Video

Video element ugrađuje medijski *player* koji podržava reprodukciju videozapisa na stranici. Osim za video, može se koristit i za ugradnju audio sadržaja, iako *audio* oznaka pruža prikladnije korisničko iskustvo [3]. Korištenjem *video* oznake na stranici stvara se izvorni video *player* koji omogućuje dodatne opcije prilagodbe i prikaza sadržaja [23].

7.1.1. Atributi

- *Autoplay* - ako je navedeno (čak i ako je vrijednost lažna) sadržaj će se automatski reproducirati od trenutka dostupnosti, osim kod mobilnih preglednika koji obično ne podržavaju ovaj atribut. Stranice koje automatski reproduciraju video mogu korisnicima biti neugodne pa je bolje izbjegavati ovaj atribut kada je to moguće
- *Buffered* - atribut kojim se određuju vremenski rasponi medija, sadrži *TimeRanges* objekt
- *Controls* - ako je ovaj atribut prisutan, preglednik će ponuditi kontrole koji dopuštaju korisniku da kontrolira reprodukciju videa, uključujući glasnoću, pretragu, pauziranje i nastavak reprodukcije
- *Height* - postavlja visinu videozapisa
- *Width* - postavlja širinu videozapisa
- *Loop* - ako je specifirano, preglednik će se automatski vratiti na početak videozapisa svaki put kada završi reprodukcija istog
- *Muted* - ako je podešeno, zvuk će u početku biti stišan, odnosno postavljen na bešumno
- *Poster* - pruža sliku koja će se prikazati tijekom učitavanja videa, ako ovaj atribut nije naveden, ništa se ne prikazuje dok prvi kadar ne bude dostupan

- *Preload* - namijenjen je pružanju savjeta pregledniku o tome što korisnik smatra da će dovesti do najboljeg korisničkog iskustva s obzirom na sadržaj koji se učitava prije reprodukcije videa. Može imati jednu od sljedećih vrijednosti:
    - *None* - označava da se video ne treba unaprijed učitati
    - *Metadata* - označava da se preuzimaju samo meta podaci videa, npr. duljina
    - *Auto* - označava da se može preuzeti cijela video datoteka, čak i ako se od korisnika ne očekuje da ju koristi
    - *Autovalue* - sinonim automatske vrijednosti
- *Playsinline* - upućuje na to da će se videozapis reproducirati unutar područja reprodukcije elementa. Izostanak ovog atributa ne znači da će se videozapis uvijek reproducirati preko cijelog zaslona
- *Src* - URL audio datoteke za ugradnju, ovaj atribut je opcionalan, umjesto njega se može koristiti *source* [3, 24].

7.1.2. Formati videa

Na žalost nitko se ne bi mogao složiti o tome koji format podržava, što znači da različiti preglednici podržavaju različite HTML5 video formate. Uz pomak podrške iz dodataka preglednika, preglednik je postao odlučujući čimbenik u podržavanju formata za video izvor. U početku, preglednici otvorenog koda, Chrome, Firefox i Opera svi su koristili WebM format koji je besplatan. Safari, Internet Explorer i Chrome su široko prihvaćeni, ali ne i besplatni MP4 format. S vremenom se to promijenilo i sada svi glavni preglednici pružaju podršku za MP4 format.

MP4 - MP4 ili MPEG-4 je komprimirani video format dizajniran od strane grupe *Moving Picture Experts Group* koja je izvorno razvijena kako bi osiguravala kvalitetu DVD-a komprimiranih video i audio informacija. Ovaj format MP4 datoteke podijeljen je na 17 različitih profila od kojih svaki ima različite karakteristike, a time i više ili manje prikladne za HTML5. Za HTML5 MP4 videozapise koriste se profili pod nazivom 'Osnovica' (eng. *baseline*) i 'Glavna' (eng. Main) s MIME tipom video / mpeg i tipom atributa video / mp4.

WEBM - WebM je komprimirani format koji su razvili i podržali vođe web industrije, kao što su Mozilla, Opera, Adobe i Google. Video sadržaj komprimira se VP8 kodekom, koji je

razvio On2 Technologies (Google je kupio). WebM format koristi video / webm vrste MIME tipa i video / webm vrste atributa.

Ogg - Theora Ogg je otvoreni format spremnika, prema programerima Ogg formata to je vrlo dovoljno *streaming* format i omogućuje kvalitetnu manipulaciju multimedijskim podacima. OGG Theora – pozitivno je *royalty-free* kodek i nije opterećen bilo kojim poznatim patentima osim izvornih VP3 patenata, koji su licencirani bez naknade. Negativno - Theora generira videozapise visoke kvalitete s relativno većim veličinama datoteka. Osim toga, puno je teže pronaći alate za pretvorbu u OGG Theoru [3].

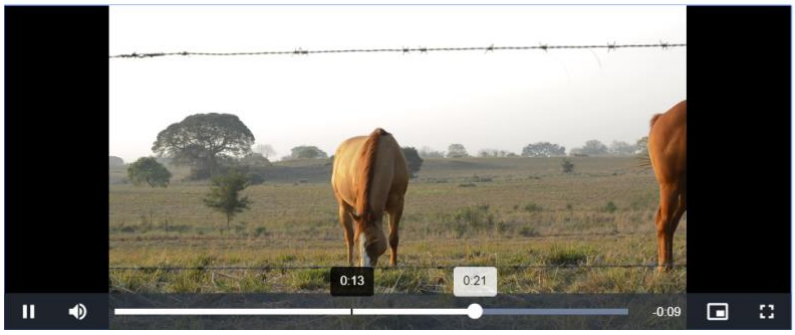
Tablica kompatibilnosti video formata s različitim preglednicima: [17]

	Google Chrome	Microsoft Edge	Mozilla Firefox	Safari	Opera
MP4	Da	Da	Da	Da	Da (25+)
WebM	Da	Ne	Da	Ne	Da
Ogg	Da	Ne	Da	Ne	Da

7.1.3. Odabir video player-a

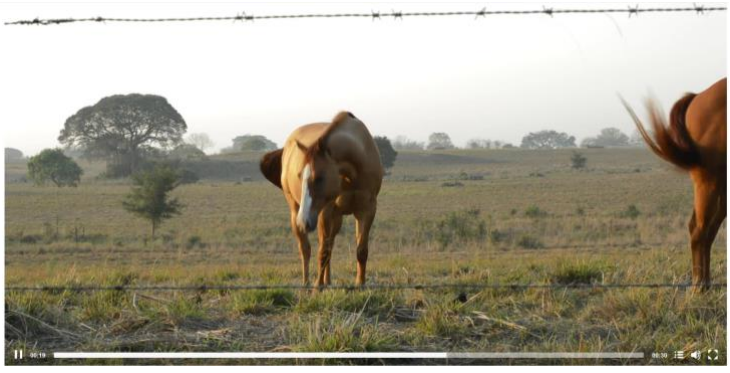
Open source video *player*-i nude neke prednosti u odnosu na uređaje razvijene s vlastitim softverom te ih mnoge firme i individualni korisnici koriste. Video *player*-i izgrađeni s vlasničkim softverom koriste privatni izvorni kod i nude se putem licenciranja gdje korisnici trebaju platiti i prihvatiti različite uvjete prije upotrebe *player*-a. S obzirom na znatno poboljšanu provjeru sigurnosti, dostupnosti ispravljanja pogrešaka (eng. *bug*s) video *player*-i otvorenog koda su slobodni za upotrebu bez licenciranih naknada i ugovora. Potpuno su besplatni za preuzimanje i korištenje za širu javnost. Mogu se prilagoditi i izmijeniti kako bi se zadovoljili specifični zahtjevi organizacija ili pojedinca. Bez obzira na to jesu li trenutno u fazi evaluacije ili se komercijalno integriraju, njihova globalna zajednica programera i korisnika je uvijek dostupna odgovoriti na pitanja i dati savjet. Ovakva podrška je potpuno besplatna i može sadržavati detaljnu dokumentaciju, vijesti, forume, popise e-pošte, *chat* uživo i slično [26]. S obzirom na sve navedeno, možemo zaključiti da su takvi *player*-i uvijek dobra odluka, u nastavku ćemo opisati dva popularna i na webu česta *open source* video *player*-a.

Video.js je posebno kreiran za HTML5, podržava HTML5 *Flash Video* (što može biti korisno kada ga koristimo u preglednicima koji nemaju HTML5 podršku). Trenutačno ga aktivno koristi oko 400,000 web stranica. Kompatibilan je s mobilnim uređajima. Potpuno je prilagodljiv na stiliziranje, npr. na njega možemo dodati svoj logotip. Nudi veliku podršku za dodatke, kao što je analitika, oglašavanje i napredni formati (npr. *HLS, DASH*).



Slika 7.1 Video.js player

MediaElement.js danas je jedan od najboljih media *player*-a otvorenog koda za web stranice. Podržava audio i video datoteke te koristi *Silverlight* i *Flash*, što mu omogućuje dosljedno korisničko sučelje u svim preglednicima. Umjesto da nudi HTML5 *player* modernim preglednicima i samostalni *Flash player* starijim, *MediElement.js* pruža nadogradnju s prilagođenim *Silverlight* i *Flash* dodacima koji imitiraju HTML5 *MediaElement API*. Može reproducirati medije iz lokalnih i drugih izvora kao što su *SoundCloud*, *YouTube*, *DailyMotion*, *Vimeo* i *Twitch*. Može se konfigurirati s *Vanilla JavaScript*, *jQuery*, *Node*, *Meteor*, *Require.js* i *React*. Jedna od najvećih prednosti su njegova česta ažuriranja i unaprjeđivanja, što znači da će se problemi rješavati brže i ažurirati u skladu s najnovijim HTML5 standardima i podrškama preglednika [25, 26, 27].



Slika 7.2 MediaElement.js player

7.1.4. Tablica kompatibilnosti

	Google Chrome	Microsoft Edge	Mozilla Firefox	Safari	Opera
<video>	4.0+	9.0+	3.5+	4.0+	10.5+

[19]

7.1.5. Konačni kod

Atribut *controls* dodaje video kontrole, kao što su reprodukcija, pauza i glasnoća. Uvijek je dobro uključiti atribute širine i visine, ako nisu postavljeni stranica može treptati dok se video učitava. Tekst između početne i završne oznake prikazivat će se samo u preglednicima koji ne podržavaju *video* element. Može se napisati neki tekst za slučaj da se video ne može vidjeti, kao što je u ovom slučaju: „vaš preglednik ne podržava HTML5 video oznaku.“ Svi preglednici ne podržavaju iste video formate pa se definira više izvora unutar ugniježđenih *source* elemenata, a preglednik će tada upotrijebiti prvi priznati oblik [17, 3].

Primjer 1. Izgled player-a i prikaz na različitim preglednicima opisan je u poglavlju 5.

```
<video controls width="320" height="240">
  <source src="horses.mp4" type="video/mp4">
  <source src="horses.webm" type="video/webm">
  Vaš preglednik ne podržava HTML5 video oznaku.
</video>
```

[17]

Primjer 2. Osim što možemo preuzeti već razvijen player, možemo napraviti i svoj - prilagođeni video player. Jedan od razloga zašto je HTML5 video tako popularan je jednostavnost stvaranja vlastitih prilagođenih kontrola i njihovog stiliziranja. To se sve radi putem JavaScript API-ja [26]. U nastavku ćemo napraviti jedan jednostavni prilagođeni video *player*.

U HTML kodu ćemo implementirati videozapis i njegove kontrole; gumb za reprodukciju (*play / pause*), bešumno (*mute*), način punog zaslona (*full screen*) te dva klizača. Jedan koji će se koristiti za preskakanje kroz video, a drugi za kontrolu glasnoće.

```
<div id="video-container">
  <!-- Video -->
    <video id="video" width="640" height="360">
```

```
    <source src="horses.mp4" type="video/mp4">
    <source src="horses.webm" type="video/webm">
    Vaš preglednik ne podržava HTML5 video oznaku.
  </video>

  <!-- kontrole -->
    <div id="video-controls">
    <button type="button" id="play-pause" class="play">Play</button>
    <input type="range" id="seek-bar" value="0">
    <button type="button" id="mute">Mute</button>
     <input type="range" id="volume-bar" min="0" max="1" step="0.1"
     value="1">
    <button type="button" id="full-screen">Full-Screen</button>
   </div>
</div>
```

U JavaScript dodajemo varijable za video, gumbe i klizače.

```
window.onload = function() {

  // Video
  var video = document.getElementById("video");

  // Gumbi
  var playButton = document.getElementById("play-pause");
  var muteButton = document.getElementById("mute");
  var fullScreenButton = document.getElementById("full-screen");

  // Klizači
  var seekBar = document.getElementById("seek-bar");
  var volumeBar = document.getElementById("volume-bar");

}
```

Implementacija gumba za reprodukciju / pauzu - postaviti ćemo događaj koji će u petlji provjeravati da li se video trenutno reproducira i zatim na odgovarajući način ili zaustaviti video ili ga reproducirati.

```
// Događaj za play/pause dugme
playButton.addEventListener("click", function() {
  if (video.paused == true) {
    // reproduciraj video
    video.play();

    // Promjena teksta gumba u 'Pause'
    playButton.innerHTML = "Pause";
  } else {
    // Pauziraj video
    video.pause();

    // Promjena teksta gumba u 'Play'
    playButton.innerHTML = "Play";
  }
```

```
});
```

Implemetacija gumba za bešumno - da bi pokrenuli gumb za isključivanje zvuka, slijediti ćemo slični postupak kao i kod gumba za reprodukciju / pauzu. Napraviti ćemo petlju koja provjerava da li ima zvuka i sukladno tome mijenja njegovu vrijednost.

```
/ Događaj za gumb 'mute'
muteButton.addEventListener("click", function() {
  if (video.muted == false) {
    // bešumno
    video.muted = true;

    // Promjena teksta gumba
    muteButton.innerHTML = "Unmute";
  } else {
    // isključi bešumni način
    video.muted = false;

    // Promjena teksta gumba
    muteButton.innerHTML = "Mute";
  }
});
```

Prikaz preko cijelog zaslona - suvremeni preglednici implementirali su *FullScreen API* koji omogućuje prikaz cijelog zaslona. Postaviti ćemo događaj koji će pritiskom na gumb pozvati funkciju *requestFullScreen()*. Da bi osigurali da će raditi u svim preglednicima, provjeriti ćemo je li *requestFullScreen()* dostupan u svim verzijama, odnosno uključiti funkcije *mozRequestFullScreen()* i *webkitRequestFullscreen()* ako nije.

```
// Događaj za prikaz preko cijelog zaslona
fullScreenButton.addEventListener("click", function() {
  if (video.requestFullscreen) {
    video.requestFullscreen();
  } else if (video.mozRequestFullScreen) {
    video.mozRequestFullScreen(); // Firefox
  } else if (video.webkitRequestFullscreen) {
    video.webkitRequestFullscreen(); // Chrome i Safari
  }
});
```

Klizač na vremenskoj traci - napraviti ćemo događaj koji će povlačenjem klizača promjeniti položaj videzapisa i izračunati vrijednost trenutnog vremena trajanja videa. Izračun ćemo napraviti tako da ukupno vrijeme trajanja videa pomnožimo s vrijednošću koji ćemo dobiti

kada trenutno stanje vremena podijelimo s brojem 100. Da bi se promijenila pozicija u videozapisu funkciji *currentTime()* dodijeliti ćemo izračunatu vrijednost.

```
// Događaj za klizač
seekBar.addEventListener("change", function() {
  // Izračun novog vremena
  var time = video.duration * (seekBar.value / 100);

  // Promjena trenutnog vremena
  video.currentTime = time;
});
```

Nakon što smo promjenili poziciju u videozapisu, ostaje nam da promjenimo sam položaj klizača u vremenskoj traci. Da bi to napravili postaviti ćemo događaj koji će se izvršiti kada se aktivira događaj *timeupdate.* Obrnuti ćemo prethodno korišten izračun da bi dobili vrijednost na koju se klizač treba postaviti u vremenskoj traci.

```
// Promijena vrijednosti klizača za vrijeme trajanja videa
video.addEventListener("timeupdate", function() {
  // Izračun položaja klizača
  var value = (100 / video.duration) * video.currentTime;

  // Promijena vrijednost klizača
  seekBar.value = value;
});
```

Klizač glasnoće zvuka - Dodajemo događaj *change,* kojim ćemo uzeti vrijednost klizača i promijeniti vrijednost glasnoće videa. Svojstvo volumena bi uvijek trebalo biti između 0 i 1. Prisjetit ćemo se da smo u HTML kodu koristili atribute *min* i *max* pomoću kojih će funkcionirati događaj.

```
// Događaj za klizač glasnoće
volumeBar.addEventListener("change", function() {
  // Promjena glasnoće videa
  video.volume = volumeBar.value;
});
```

Programski dio je gotov, ostaje da u CSS-u uredimo izgled *player*-a.

```
/* video player*/
     video-container {
     width: 640px;
     height: 365px;
     position: relative;
```

```
}

/* kontrole */
#video-controls {
     position: absolute;
     bottom: 0;
     left: 0;
     right: 0;
     padding: 5px;
     opacity: 0;
     -webkit-transition: opacity .3s;
     -moz-transition: opacity .3s;
     -o-transition: opacity .3s;
     -ms-transition: opacity .3s;
     transition: opacity .3s;


     background-image: -webkit-gradient(
          linear,
          left bottom,
          left top,
          color-stop(0.13, rgb(3,113,168)),
          color-stop(1, rgb(0,136,204))
     );
}

#video-container:hover #video-controls {
     opacity: .9;
}

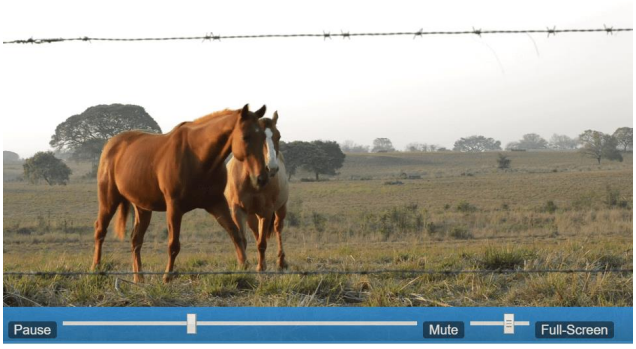
/* gumbi */
button {
     background: rgba(0,0,0,.5);
     border: 0;
     color: #EEE;
     -webkit-border-radius: 3px;
     -moz-border-radius: 3px;
     -o-border-radius: 3px;
     border-radius: 3px;
}

button:hover {
     cursor: pointer;
}

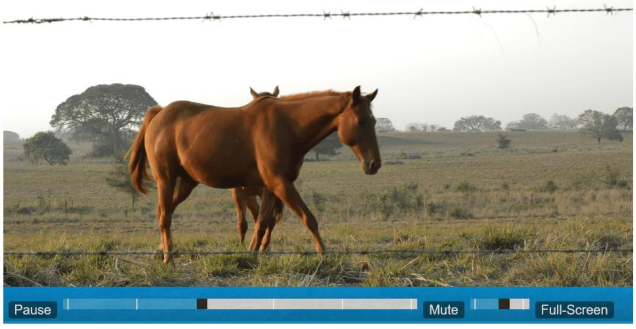
/* klizači */
#seek-bar {
     width: 360px;
}

#volume-bar {
     width: 60px;
}
```

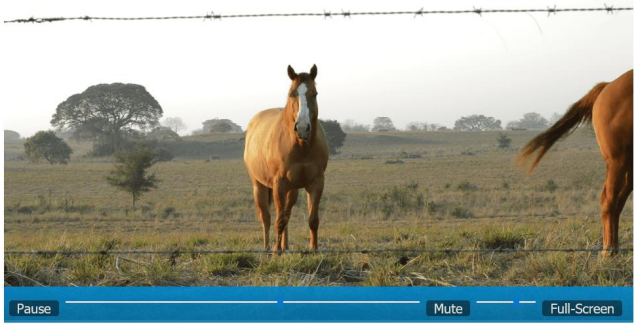
[27]



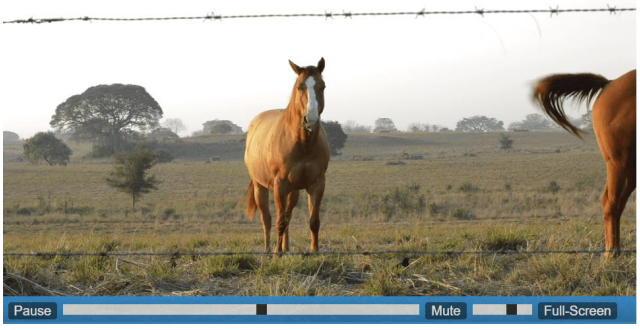
Slika 7.3 prilagođeni video player u Chrom-u



Slika 7.4 prilagođeni video player u Edge-u



Slika 7.5 prilagođeni video player u Firefox-u



Slika 7.6 prilagođeni video player u Operi

7.2. Audio

Audio element omogućuje ugradnju zvučnih datoteka na stranici, može se koristiti za dodavanje glazbe, e-knjiga, *podcast-a* i ostalog audio sadržaja. Način funkcioniranja je vrlo sličan kao i kod *video* oznake. Pruža podršku za tri audio formata; MP3, OGG i WAV, od kojih je MP3 najčešći i kompatibilan sa svim preglednicima.

7.2.1. Atributi

- *Autoplay* - ako je naveden, audiozapis će odmah započeti s reprodukcijom, bez čekanja da cijela audio datoteka završi s preuzimanjem
- *Controls* - ako je ovaj atribut prisutan, preglednik će ponuditi kontrole koji dopuštaju korisniku da kontrolira reprodukciju zvuka, uključujući glasnoću, pretragu, pauziranje i nastavak reprodukcije
- *Loop* - ako je specifirano, preglednik će se automatski vratiti na početak audiozapisa svaki put kada završi reprodukcija istog
- *Muted* - ako je podešeno, zvuk će u početku biti stišan, odnosno postavljen na bešumno
- *Preload* - namijenjen je pružanju savjeta pregledniku o tome što korisnik smatra da će dovesti do najboljeg korisničkog iskustva s obzirom na sadržaj koji se učitava prije reprodukcije videa. Može imati jednu od sljedećih vrijednosti:
    - *None* - označava da se zvuk ne treba unaprijed učitati
    - *Metadata* - označava da se preuzimaju samo metapodaci audiozapisa, npr. duljina
    - *Auto* - označava da se može preuzeti cijela audio datoteka, čak i ako se od korisnika ne očekuje da ju koristi
    - *Autovalue* - sinonim automatske vrijednosti
- *Src* - URL audio datoteke za ugradnju, ovaj atribut je opcionalan, umjesto njega se koristi *source* [3]

7.2.2. Tablica kompatibilnosti

	Google Chrome	Microsoft Edge	Mozilla Firefox	Safari	Opera
<audio>	4.0+	9.0+	3.5+	4.0+	10.5+

[19]

7.2.3. Formati zvuka

MP3 (eng. *MPEG-1 Audio Layer 3*) pušten je u upotrebu devedesetih godina, nakon čega mu je ubrzo porasla popularnost i postao je najkorišteniji audio format za glazbene datoteke. Skoro svaki digitalni uređaj na svijetu sa audio podrškom može čitati i reproducirati MP3 datoteke. Unatoč svemu navedenome, kreatori MP3 formata su objavili da će ga prestati razvijati. Iako se sigurno neće prestati koristiti kada su u pitanju krajnji korisnici, s vremenom će i s njihove strane biti zaboravljen. Razlog je dolazak boljeg i kvalitetnijeg **AAC** (eng. *Advanced Audio Coding*) formata, koji je na putu da postane standardom kada je u pitanju računalna glazba ili video glazbena podloga, kao na računalima tako i na mobilnim uređajima. Algoritam kompresije koji koristi AAC mnogo je napredniji od MP3 formata, kada usporedimo istu snimku u oba formata istih *bitrate*-inga, ACC snimak će općenito imati bolju kvalitetu zvuka. Upotreba ACC formata sve je šira, te je postao standardni način kompresije zvuka koji koriste *YouTube, Android, IOS, iTunes, Nintendo, PlayStation* te mnogi drugi. **WAV** (eng. *Waveform Audio File Format*) format kreiran je od strane *Microsoft*-a i IBM-a devedesetih godina. Često se krivo pretpostavlja da su sve WAV datoteke nekomprimirane audio datoteke ali to nije sasvim točno. WAV je zapravo *Windows* kontejner za različite audio formate, što znači da takva datoteka potencionalno može sadržavati komprimirani zvuk, ali se za to rijetko koristi. **OGG** je multimedijski kontejner koji može sadržavati sve vrste formata kompresije, ali najčešće se koristi za pohranu *Vorbis* datoteka. *Vorbis* je dobio na popularnosti jer se pridržava principa *open source* softvera, te ostvaruje znatno bolje rezultate od većine ostalih formata kompresije zvuka (stvara manju veličinu datoteke za ekvivalent kvalitete zvuka). MP3 i ACC imaju tako snažna uporišta da se OGG teško probijao do središta pažnje, mnogi uređaji ga ne podržavaju u potpunosti, ali to se s vremenom poboljšava. Za sada ga najviše koriste zagovornici softvera otvorenog koda [28].

Tablica kompatibilnosti audio formata s različitim preglednicima:

	Google Chrome	Microsoft Edge	Mozilla Firefox	Safari	Opera
MP3	Da	Da	Da	Da	Da
WAV	Da	Ne	Da	Da	Da
OGG	Da	Ne	Da	Ne	Da
ACC	12+	12+	Djelomično*	Ne	15+

[17, 29]

*ACC format je djelomično podržan od strane Firefox verzije 22+. Djelomična podrška znači da je ACC podržan samo unutar MP4 kontejnera i to samo kada operativni sustav već ima instalirane potrebne kodeke.

7.2.4. Konačni kod

Atributom *controls* dodajemo kontrole kao što su reprodukcija / pauza i volumen zvuka. *Source* omogućuje određivanje alternativnih audio datoteka, preglednik će izabrati i koristiti prvi priznati format. Tekst između početne i završne *audio* oznake prikazati će se samo u preglednicima koji ne podržavaju HTML5 *audio* [17].

Primjer 1. Izgled audio player-a biti će drukčiji u svakom pregledniku.

```
<audio controls>
  <source src="sounds_horse.mp3" type="audio/mp3">
  <source src="sounds_horse.ogg" type="audio/ogg">
  Vaš preglednik ne podržava HTML5 audio oznaku.
</audio>
```

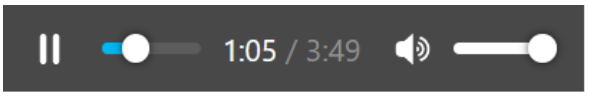
[3]



Slika 7.7 HTML5 audio player u Chrom-u



Slika 7.8 HTML5 audio player u Edge-u



Slika 7.9 HTML5 audio player u Firefox-u



Slika 7.10 HTML5 audio player u Operi

7.2.5. Odabir audio player-a

Jedna od prednosti *streaming*-a je da široke mase mogu uživati u svojoj omiljenoj glazbi i on-line emisijama bez brige oko prostora za pohranu na svojim uređajima. Sve veći broj ljudi, posebice mlađe generacije, otkrivaju ovu pogodnost jer dostupnost pristupačnog interneta doseže sve veće razmjere. Prilikom odabira audio *player*-a za web stranicu možemo pratiti koliko često se pojedini *player* ažurira. Posljedica stalnih ažuriranja modernih preglednika je što ponekad prekida određene značajke koje koristi sam *player*. Licenca otvorenog koda omogućuje nam da u potpunosti prilagodimo značajke dodavanjem novih kodova ili koristimo prijedloge drugih korisnika kako bi stvorili novu vrstu audio uređaja s jedinstvenim značajkama. U nastavku slijedi opis nekoliko najčešće korištenih audio *player*-a otvorenog koda.

MediaElement.js je popularan *player* kao za video tako i za audio datoteke. Opisan u poglavlju 7.1.3.

SoundManager 2 je pripremljen kako bi olakšao isporuku zvuka na desktopu i mobilnim platformama. Ima snažan API koji koristi HTML5 audio, gdje podržava i opcionalno vraća natrag u *Flash* gdje je to potrebno. Može se koristiti s drugim JavaScript *framework*-ima kao što je *jQuery*. Dokumentacija je vrhunska, omogućujući programerima da ga konfiguriraju za svoje potrebe. Podržava popise za reprodukciju (eng. *playlists*). Ažuriranja ovog audio *player*-a nešto su sporija, no globalna zajednica ulaže napore u rješavanju pojedinih problema i pitanja koje prijavljuju korisnici. Vrhunski je dizajniran i kompatibilan s najnovijim standardima preglednika [30].



Slika 7.11 SoundManager 2 audio player



Slika 7.12 SoundManager 2 povećana slika audio kontrola

Amplitude.js omogućuje programerima razvoj potpuno prilagođenog sučelja *player*-a s brojnim mogućnostima i funkcionalnostima. Modernog je izgleda i privlači sve veći broj korisnika, kompatibilan je s brojim platformama kao što su *Angular*, *React*, *Vue*, *vanilla JS*, *jQuery ninja* i mnoge ostale. *Amplitude.js* pruža raznolike načine prikaza i omogućava odabir načina prikazivanja; samo jedne pjesme, *podcast* kataloga, popisa za reprodukciju, pa čak i *streaming*-a uživo [31].



Slika 7.13 Amplitude.js audio player

7.3. Track

Mnogi korisnici ne mogu ili ne žele čuti medijski sadržaj koji pronađu na web-u ili to ne mogu u određenom vremenu. Razlozi mogu biti raznoliki:

- Osobe sa oštećenjem sluha
- Neke osobe možda neće moći čuti zvuk jer su u bučnom okruženju (poput festivala) ili možda ne žele ometati druge ako su na mirnom mjestu (poput knjižnice)
- Osobe koje ne govore jezik medijskog zapisa i imaju potrebu za transkripcijom teksta ili prijevodom kako bi mogli razumjeti sadržaj medija [3]

Track pruža jednostavan, standardizirani način za određivanje tekstualnih zapisa za audio i video sadržaj. Upotrebljava se za određivanje podnaslova (titlova), naslova datoteka (eng. *caption*), opisa ili drugih datoteka koje sadrže tekst, a koje bi trebale biti vidljive tijekom reprodukcije medija [17, 7]. Omogućuje upotrebu vremenskih metapodataka i dinamičkog sadržaja povezanog s reprodukcijom medija i time dodaje vrijednost *audio i video* elementima. S obzirom na njegovu snagu, fleksibilnost i jednostavnost, *track* predstavlja veliki korak u povećanju i dinamičnosti medija na webu [24].

7.3.1. Atributi

- *Src* – definira lokaciju datoteke s tekstualnim zapisom u obliku poveznice (URL-a), datoteka mora biti u određenom formatu (npr. WebVTT). Ovaj atribut mora biti naveden da bi oznaka bila važeća
- *Srclang* – označava jezik zapisa teksta
- *Kind* – specificira vrstu podataka koje track omogućuje za medijsku oznaku, dopuštene su sljedeće vrijednosti:
    - *Captions* – može biti puna transkripcija dijaloga, zvučni efekt ili bilo koji drugi valjani podaci, koristiti će se kada audio/video zapis nije dostupan. Prikladan je za osobe s oštećenjem sluha.
    - *Chapters* – određuje naslove poglavlja (ukoliko poglavlja postoje), može biti prikladan za navigaciju medijskim sadržajem
    - *Subtitles* – pruža potpunu transkripciju dijaloga i koristi se kada je zvuk medija nerazumljiv. Ovo je zadana vrijednost (eng. *default*).
    - *Descriptions* – pruža tekstualni opis medijskog sadržaja i koristi se kada vizualne informacije nisu dostupne
    - *Metadata* – sadrži podatke o korištenom izvoru medija
- *Label* – definira naslov zapisa koji će biti vidljiv od strane korisnika
- *Default* – preglednik će reći korisniku da zapis mora biti omogućen (osim u slučaju kada je u korisničkim postavkama definirano drugačije) [7]

7.3.2. Tablica kompatibilnosti

	Google Chrome	Microsoft Edge	Mozilla Firefox	Safari	Opera
<track>	18.0+	10.0+	31.0+	6.0+	15.0+

[18]

7.3.3. Formati kodiranja podnaslova

Postoji nekolicina formata za kodiranje podnaslova, ali dva najčešća su WebVTT (eng. *Web Video Text Track Format*) i WebSRT (eng. *Web Sutible Resource Tracks*). Metode kodiranja su dosta slične, glavna razlika je što za kodiranje WebSRT formata potreban *SubRip* program, a kod WebVTT-a poslužiti će jednostavan uređivač teksta (npr. *WordPad*, *Notepad*).

WebVTT ima HTML5 podršku, pa ćemo ga koristiti za postavljanje titlova za *video* i *audio* elemente [32].

WebVTT je format za pisanje tekstualnih datoteka koje sadrže više nizova teksta uz metapodatke kao što je vrijeme na videozapisu za koje želimo da se svaki tekstni niz prikazuje, pa čak i ograničene informacije o stiliziranju i pozicioniranju. Tu je zastupljen i API dostupan za predstavljanje i upravljanje podatcima potrebnim za reprodukciju teksta u točno vrijeme. Mora biti kodiran pomoću UTF-8. Pruži mogućnost uređivanja boje i fonta teksta, što ga čini jednim od najboljih dostupnih formata [3].

7.3.4. WebVTT struktura podnaslova

WebVTT format iako pisan regularnih tekstom mora slijediti određeni redoslijed i raspored. Na početku se velikim tiskanim slovima napiše „WEBVTT“, te pusti prazan red. Prilikom pisanja vremenskog okvira, započeti ćemo vremenom kada kojem želimo da se podnaslov pojavi i završiti kada bi treba nestati. Slijedi sam tekst podnaslova [32]. Svaka stavka u *track* datoteci naziva se *cue*. Svaki *cue* ima vrijeme početka i završetka odvojeno strelicom, s tekstom u donjem retku. *Cue*-ovi mogu po želji imati i identifikacijske brojeve, a odvojene su praznom linijom. Duljina vremena izražena je u satima, minuta, sekunde i milisekundi formatu. Ako je potrebno, brojevi moraju biti prazni: sati, minute i sekunde moraju imati dvije znamenke („00“ za nulu), a milisekundi moraju imati tri znamenke („000“ za nulu) [26].

Slijedeći primjer prikazuje tipičnu .vtt datoteku: [7]

```
WEBVTT

00:00:05.000 --> 00:00:10.000
Konji su tipično životinje koje žive u krdima.

00:00:12.000 --> 00:00:17.000
U krdu vlada jasno uspostavljen hijerarhijski red.
```

7.3.5. Konačni kod

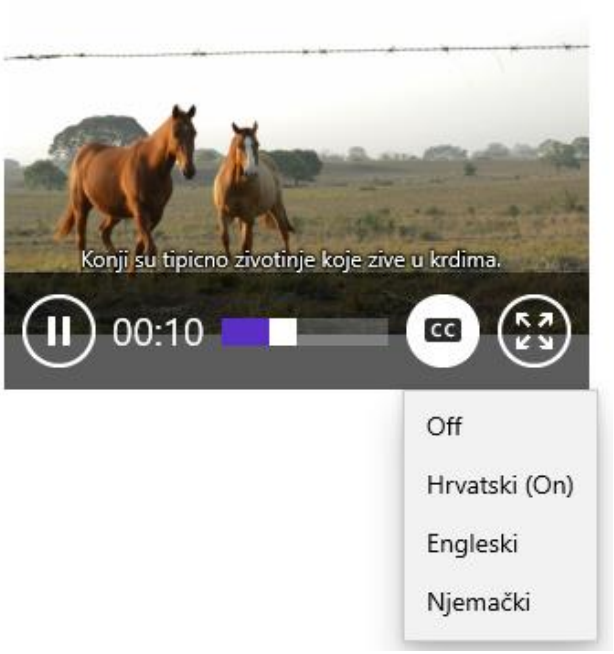
Primjer 1. Osnovi kod u kojem ćemo postaviti hrvatski podnaslov unutar *video* oznake. Također ćemo definirati više podnaslova na više različitih jezika, svaki put dodavanjem nove *track* oznake.

```
<video width="320" height="240" controls>
  <source src="horses.mp4" type="video/mp4">
  <source src="horses.ogg" type="video/ogg">

  <track src="subtitle_hr.vtt" kind="subtitles" srclang="hr"
  label="Hrvatski" default>
  <track src="subtitle_eng.vtt" kind="subtitles" srclang="eng"
  label="Engleski">
  <track src="subtitle_de.vtt" kind="subtitles" srclang="de"
  label="Njemački">

</video>
```

[17]

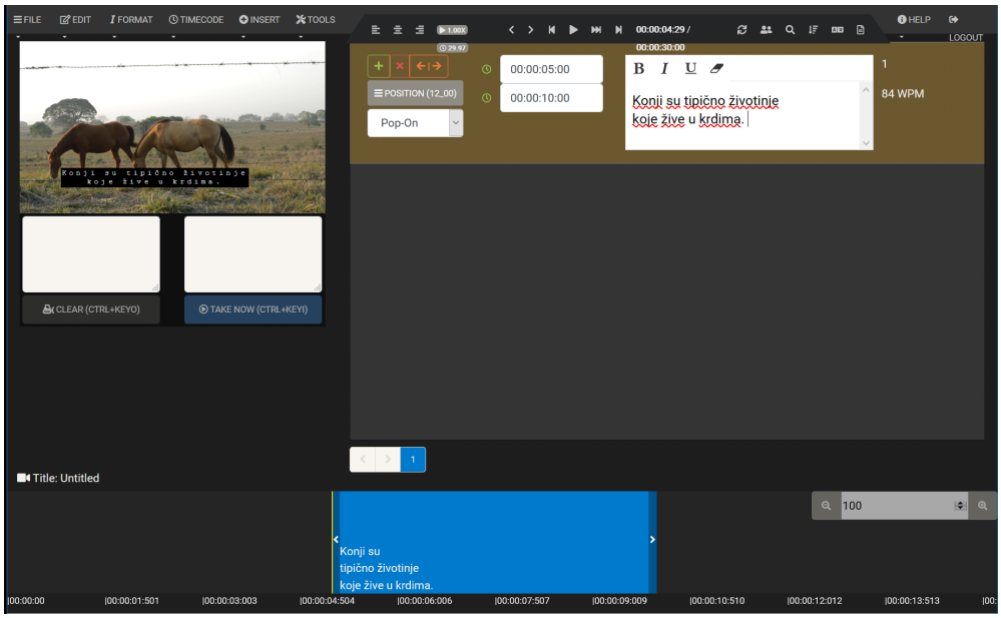


Slika 7.14 odabir podnaslova u video player-u

7.3.6. Uređivači podnaslova

Osim ručnog pisanja podnaslova u nekom od tekstualnih editora, postoje softverski i *online* editori koji to rade umjesto nas. Jedan od takvih je *Closed Caption Creator*.

Online uređivač *Closed Caption Creator* je besplatna online aplikacija za stvaranje, uređivanje i isporuku titlova. Namijenjen je pružanju naprednih alata, i nudi podršku za više formata titlova, među kojima je i WebVTT. Pruža pregledno sučelje te jednostavne metode uređivanja i stiliziranja.



Slika 7.15 uređivač podnaslova Closed Caption Creator

7.4. sažetak

HTML5 daje prvu vjerodostojnu alternativu dodataka treće strane. Prednost je da su *video* i *audio* sada redoviti elementi nativno podržani od preglednika i nude snažan API i vrlo su jednostavni za kontrolu putem JavaScript-a. Razvojni programeri mogu lako izraditi vlastite prilagođene kontrole ili napraviti sve vrste video manipulacija sa samo nekoliko redaka koda. Kao sigurnosna mreža za preglednike koji se ne mogu nositi mogu se preuzeti video i audio datoteke izvan samih elemenata. Imamo brojne gotove skripte koje nam omogućuju da jednostavno koristimo HTML5 sinergije na svojim stranicama koje su vrhunskog dizajna i funkcionalnosti [33].

8. Prikaz grafičkog sadržaja

Dugo vremena jedini način za prikazivanje slika na Webu bilo je korištenje slika u formatima JPG za prikaz fotografija i GIF za ikone i crteže, te kasnije i PNG format. Sve su to interni formati slika, osim jednostavnih animacija oni ne mogu ponuditi dinamičke ili interaktivne slike ili grafiku, za što su se uglavnom koristili dodaci za *Flash*. HTML5 uvodi dva nova elementa za korištenje grafike na Webu: SVG, vektorski format s kojim se može manipulirati kroz CSS i JavaScript te *canvas*, element koji omogućuje crtanje korištenjem API-ja. Između ova dva formata otvorio se cijeli novi raspon grafičkih mogućnosti, od interaktivnih dijagrama do raznih mogućnosti manipulacije slikama [5].

8.1. SVG

SVG (eng. *Scalable Vector Graphics*) je jezik za opisivanje 2D grafike i grafičkih aplikacija u XML-u. Vektori mogu biti jednostavni oblici, putanje ili bilo koji objekt koji možemo napraviti u npr. *Adobe Illustrator*-u. SVG je format slike ima više zajedničkog s web stranicom nego primjerice s JPG formatom. SVG je mnogo moćniji od ostalih formata slika koje možemo koristiti na Webu jer ga možemo manipulirati kodom (CSS, JS). HTML5 element *svg* služi kao kontejner za SVG grafiku [34, 17].

8.1.1. Prednosti i nedostaci

Glavni razlog popularnosti SVG formata je njegova skalabilnost, odnosno ne gubi nikakvu kvalitetu ako se zumira ili promijeni veličina, što je vrlo važno za responzivni dizajn modernog Weba. SVG slike mogu se izraditi i uređivati s bilo kojim programom za uređivanje teksta, mogu se pretraživati, indeksirati i komprimirati. Kod printanja slike se mogu ispisati s visokom kvalitetom pri bilo kojoj razlučivosti. SVG je otvoreni standard. Jedna od najvećih prednosti SVG slika – umetanje u DOM, ujedno je i jedna od najvećih slabosti. Budući da SVG postaje dio stranice, ona će se sporije učitavati jer se renderira kompleksnija stranica i troši se više memorije. Radi toga treba razmisliti o korištenju složenih SVG objekata na stranici, jer će to nepovoljno utjecati na korisnike koji će stranicu posjećivati preko mobilnih telefona ili prijenosnih uređaja [5, 17]. U tijeku je razvoj nove SVG inačice. Nakon što je W3C SVG grupa odustala od razvoja prije par godina, ponovno je obnovljena s misijom da završe SVG 2, koja bi trebala bolje funkcionirati i imati veći broj mogućnosti [35].

8.1.2. Tablica kompatibilnosti

	Google Chrome	Microsoft Edge	Mozilla Firefox	Safari	Opera
<svg>	4.0+	9.0+	3.0+	3.2+	10.1+

[17]

8.1.3. Atributi

- *BaseProfile* - minimalni profil SVG jezika koji dokument zahtjeva (ne koristi se od SVG 2)
- *ContentScriptType* - zadani skriptni jezik koji koristi SVG (ne koristi se od SVG 2)
- *ContentStyleType* - zadani opisni jezik koji koristi SVG (ne koristi se od SVG 2)
- *Hight* - visina pravokutnog okvira za prikaz (eng. *viewport*)
- *PreserveAspectRatio* - definira kako se *svg* fragment mora prilagoditi ako se prikazuje s različitim omjerima (eng. *aspect ratio*)
- *Version* - definira koja će se inačica koristiti za unutarnji sadržaj elementa (ne koristi se od SVG 2)
- *ViewBox* - SVG kordinate okvira za prikaz za trenutni SVG fragment
- *Width* - prikazana širina pravokutnog okvira za prikaz
- *X* - prikazana x kordinate *svg* kontejnera
- *Y* - prikazana y kordinata *svg* kontejnera [3]

8.1.4. Konačni kod

Primjer 1. prikaz gotove SVG datoteke na stranici može se napraviti pomoću *img*, *picture* ili *embed*. Prikaz će biti isti neovisno o pregledniku ili koju od oznaka odaberemo.

```
<p> oznaka img </p>
<img src="horse_clipart.svg" alt="Horse Running Silhouette">

<p> oznaka picture </p>
<picture>
  <source media="(min-width: 600px)" srcset="horse_clipart.svg">
  <source media="(min-width: 400px)" srcset="horse2_clipart.svg">
  <img src="horse3_clipart.svg" alt="Horse Running Silhouette"
style="width:auto;">
</picture>

<p> oznaka embed </p>
<embed src="horse_clipart.svg">
```

oznaka img



oznaka picture



oznaka embed



Slika 8.1 prikaz SVG datoteke pomoću različitih oznaka

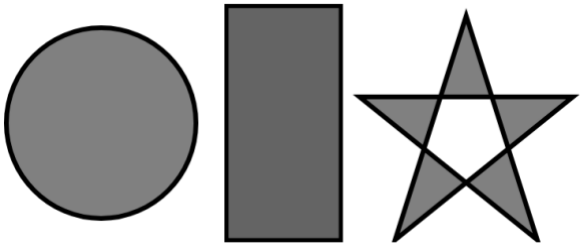
Primjer 2. Izrada SVG objekata na stranici; krug, pravokutnik i zvijezda.

```
<!-- krug -->
<svg width="200" height="200">
   <circle cx="100" cy="100" r="80" stroke="black" stroke-width="4"
fill="gray" />
   Vaš preglednik ne podržava SVG.
</svg>

<!-- pravokutnik -->
<svg width="100" height="200">
  <rect width="100" height="200" style="fill:rgb(100,100,100);stroke-
width:7;stroke:rgb(0,0,0)" />
  Vaš preglednik ne podržava SVG.
</svg>

<!-- zvijezda -->
<svg width="300" height="200">
  <polygon points="100,10 40,198 190,78 10,78 160,198"
  style="fill:gray;stroke:black;stroke-width:4;fill-rule:evenodd;" />
</svg>
```

[17]



Slika 8.2 izgled kreiranih SVG oblika

Primjer 3. Za dodavanje SVG animacije koristit ćemo *animateMotion* oznaku i njene atribute *path*, *dur* i *repeatCount*. Rezultat će biti animirani krug.

```
<svg witdh="300" height="200">
<circle cx="50" cy="90" r="20" fill="gray"></circle>
<animateMotion path="M 0 0 V 0 H 300 Z" dur="3s"
repeatCount="indefinite" />
</svg>
```

[36]



Slika 8.3 izgled SVG animacije

8.2. Canvas

HTML5 definira element eng. *canvas* kao rastersku površinu za crtanje koja ovisi o rezoluciji koja se može koristit za iscrtavanje grafova, grafike igara i drugih slika tokom izvršavanja. Crtanje se odvija pomoću JavaScripta korištenjem posebnog API-ja kako bi se dobila prilagođena rasterska grafika. Naziv *canvas* (hrv. platno) nazvan je tako što se po njemu može crtati kao po praznom slikarskom platnu, isto tako kada nešto na njemu nacrtamo, s tim sadržajem više ne možemo manipulirati. Dobiveni elementi nisu poput SVG elemenata, već samo pikseli na zaslonu. Omogućeno je samo crtanje preko već nastalih elemenata [5].

8.2.1. Upotreba

Canvas se može koristiti za mnogo stvari za koje su se prethodno koristili dodaci, poput: dinamične grafike, *online* i izvanmrežne igre, animacije, interaktivni video i audio te druge vizualne prikaze. Iako se ovakvi interaktivni elementi ugraditi pomoću *Flash*-a, glavna razlika je što on ugrađena vanjska datoteka i ne može izravno komunicirati s ostalim HTML elementima. Budući da je *canvas* dio DOM-a on omogućuje upravo tu komunikaciju, primjerice možemo stvoriti animaciju koja se mijenja kada postoji interakcija nekog drugog djela stranice, poput obrasca koji se ispunjava možemo pretvoriti polje obrasca u animaciju. Pomoću *canvas* elementa možemo jednostavno pretvoriti običnu stranicu u dinamičnu web aplikaciju, rezultat će biti interaktivnost grafike, slike i teksta kojeg manipuliramo JavaScript-om. Postoje slučajevi u kojima se ne preporučuje upotreba *canvas*-a. Treba uzeti u obzir ciljanu publiku, ako su nam ciljana publika prvenstveno korisnici koji koriste Windows XP ili starije verzije Internet Explorer-a tada se *canvas* neće prikazati jer ga ti preglednici ne podržavaju. Međutim ako razvijamo aplikaciju koja će se gledati preko mobilnih uređaja (Android i iOS), kao i na modernim desktop kompjuterima tada će korištenje *canvas*-a biti dobar izbor. Pogotovo jer upotreba ovog elementa sadrži povratne mogućnosti poput statičkih slika za starije preglednike koji ga ne podržavaju. Upotreba *canvas*-a ne preporučuje se za neke objekte kao što su logo, glavni naslov, navigacija ili u slučaju kada stvaramo web stranicu ili aplikaciju koja bi se mogla koristiti i u ne-interaktivnom mediju kao što je ispis (printanje) [37].

8.2.2. Atributi

- *Height* - definira visinu kordinatnog sustava u CSS pikselima (zadana vrijednost - 150px)
- *Width* - definira širinu kordinatnog sustava u CSS pikselima (zadana vrijednost - 300px)
- *Moz-opaque* - definira prozirnost (eng. *transluency*), kada nema prozirnosti performanse crtanja mogu se optimizirati. Ovu opciju podržava jedino Mozilla preglednik, pa se umjesto toga može definirati slijedeća linija koda:

```
canvas.getContext ('2d', {alfa:false})
```

[3]

8.2.3. Tablica kompatibilnosti

	Google Chrome	Microsoft Edge	Mozilla Firefox	Safari	Opera

<canvas>	4.0+	9.0+	2.0+	3.1+	9.0+

[17]

8.2.4. Konačni kod

Korisnik ima nekoliko načina za crtanje putanja, okvira, krugova, teksta, dodavanja slika s gradijentom, višekratnih pravokutnika ili višebojnog teksta. Prema zadanim postavkama, *canvas* nema granice i nema sadržaja. Oznaka izgleda ovako:

```
<canvas id="myCanvas" width="300" height="300"></canvas>
```

Uvijek je potrebno navesti id atribut, atribut širine i visine kako bi definirali veličinu platna.

Canvas može nacrtati obojene tekstove, sa ili bez animacije. Ima izvrsne značajke za grafičko prikazivanje podataka sa slikama grafikona te grafikona. Objekti *canvas*-a mogu se pomicati. Sve je moguće: od jednostavnih do složenih animacija. Može odgovoriti na bilo koju korisničku akciju (ključne klikove, klikovi mišem, klikanje tipkovnice, pokretanje prsta).

Primjer 1. Izrada jednostavnih oblika; linije i kruga te prikaz teksta.

HTML:

```
<canvas id="myCanvas" width="600" height="100" style="border:1px
solid;"></canvas>
```

JavaScript:

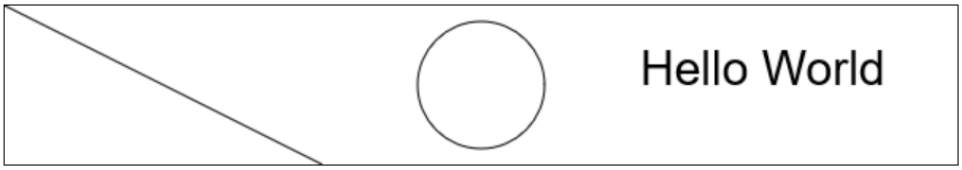
```
// linija
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.moveTo(0, 0);
ctx.lineTo(200, 100);
ctx.stroke();

// krug
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(300, 50, 40, 0, 2 * Math.PI);
ctx.stroke();

// tekst
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
```

```
ctx.font = "30px Arial";
ctx.fillText("Hello World", 400, 50);
```

[19, 17]



Slika 8.4 prikaz dobivenih objekata u canvas-u

Primjer 2. Dodavanje slike u *canvas.*

HTML:

```
<img id="horse" width="285" height="428" src="img/horse03.jpg"
alt="Brown Horse">

<canvas id="myCanvas" width="285" height="428" style="border:2px
solid;">
```

JavaScript:

```
window.onload = function() {
    var canvas = document.getElementById("myCanvas");
    var ctx = canvas.getContext("2d");
    var img = document.getElementById("horse");
   ctx.drawImage(img, 10, 10);
};
```

[17]



Slika 8.5 prikaz dodane slike u canvas-u

8.3. Usporedba

SVG je jezik za opisivanje 2D grafike u XML-u, a canvas za crtanje 2D grafike pomoću JavaScripta. SVG je temeljen na XML-u, što znači da je svaki element dostupan unutar SVG DOM-a. Svaki opisani oblik u SVG-u pamti se kao objekt, ukoliko promijenimo atribute objekta, preglednik će automatski prikazati novonastali objekt. Crtaća površina canvas-a određena je pikselima, jednom nacrtani grafički oblik biti će zaboravljen od strane preglednika. Kod promjena položaja oblika, cijela scena treba biti ponovno nacrtana, uključujući sve eventualno prethodno nacrtane oblike koji se preklapaju s grafičkim prikazom.

SVG	**Canvas**
Baziran na vektorskoj grafici	Baziran na rasterskoj grafici
Ne ovisi o rezolucij	Ovisan o rezoluciji
Modificira se putem JavaScript-a i CSS-a	Modificira se putem JavaScripta
Ima podršku za HTML događaje	Nema podrške za HTML događaje
Odlična prilagodba za aplikacije s velikim područjem rendeiranja (npr. *Google Maps*)	Slabe mogućnosti renderiranja teksta
Sporo renderiranje kod kompleksnijih projekata (jer koristi DOM)	Dobivena slika može se spremiti samo kao .png ili .jpg
Nije prikladan za igre	Dobra prilagodba za igre
Daje bolje rezultate s manjim brojem objekata ili većom površinom, ili oboje	Daje bolje rezultate s manjom površinom ili većim brojem objekata ili oboje

[5, 17]

9. Zaključak

Iako su neki mediji, kao naprimjer video i zvuk sveprisutni sastavni dio web aplikacija, oni su se uvijek morali oslanjati na programske dodatke treće strane. Važnost HTML5 je upravo uvođenje podrške za reprodukciju zvuka i videa u pregledniku preko HTML elemenata. Budući da su ti elementi dio web platforme, oni puno bolje funkcioniraju i poboljšavaju korisničko iskustvo samog weba.

Pomoću HTML-a 5 distribucija multimedijskog sadržaja je znatno olakšana i nudi brojne mogućnosti distribucije, gledanja, slušanja te pridonosi samoj dinamičnosti weba aplikacija.

Veliki utjecaj na HTML5 u budućnost imati će rast mobilnih uređaja i pad tradicionalnih računala. Programeri provoditi će više vremena razvijajući aplikacije za mobilne uređaje nego za tradicionalna računala. HTML5 će transformirati model distribucije aplikacija. Omogućiti će web programerima trenutačno ažuriranje aplikacija bez potrebe odobravanja od neke treće strane.

HTML5 se pokazao kao najbolji način za predstavljanje multimedijalnog sadržaja bliže korisniku i ključan je za budući razvoj interneta kao platforme za distribuciju sadržaja, multimedije i aplikacija.

Mjesto i datum završetka rada: U Varaždinu, 07.10.2019.

Vlastoručni potpis kandidata: Ivan Đuka

Sveučilište
Sjever

SVEUČILIŠTE
SJEVER

IZJAVA O AUTORSTVU
I
SUGLASNOST ZA JAVNU OBJAVU

Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, Ema Devčić *(ime i prezime)* pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica završnog/diplomskog (*obrisati nepotrebno*) rada pod naslovom Prikaz multimedijskog sadržaja na webu pomoću HTML5 standarda *(upisati naslov)* te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:
(upisati ime i prezime)

Ema Devčić
(vlastoručni potpis)

Sukladno Zakonu o znanstvenoj djelatnost i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Ja, Ema Devčić *(ime i prezime)* neopozivo izjavljujem da sam suglasan/na s javnom objavom završnog/diplomskog (*obrisati nepotrebno*) rada pod naslovom Prikaz multimedijskog sadržaja na webu pomoću HTML5 standarda *(upisati naslov)* čiji sam autor/ica.

Student/ica:
(upisati ime i prezime)

Ema Devčić
(vlastoručni potpis)

10. Literatura

[1] https://www.techradar.com/, dostupno 02.05.2019

[2] http://www.webtech.com.hr/, dostupno 02.05.2019

[3] https://developer.mozilla.org/en-US/, dostupno 01.09.2019

[4] http://www.yourhtmlsource.com/starthere/historyofhtml.html, dostupno 02.05.2019

[5] Peter Gasston: Moderni Web: responzivni Web dizajn uz HTML5, CSS3 i JavaScript, Zagreb, 2013.

[6] http://www.pmf.ni.ac.rs/pmf/predmeti/4604/doc/08-Multimedija.pdf, dostupno 05.05.2019

[7] http://www.brenkoweb.com/tutorials/html5/html5-media-elements, dostupno 05.05.2019

[8] http://www.dimedia.hr/multimedija, dostupno 05.05.2019

[9] http://www.oakton.edu/user/2/rjtaylor/CIS101/Text/PowerPointPresentations/morley15e__ppt_ch10%20REV.pdf, dostupno 06.05.2019

[10] https://techterms.com/definition/drm, dostupno 15.05.2019

[11] https://www.alvinisd.net/cms/lib03/TX01001897/Centricity/Domain/1077/beginning_html5_and_css3.pdf, dostupno 15.05.2019

[12] https://www.reuters.com/article/us-adobe-systems-flash/adobe-to-pull-plug-on-flash-ending-an-era-idUSKBN1AA22R, dostupno 05.06.2019

[13] https://www.htmlgoodies.com/tutorials/html5/new-tags-in-html5.html, dostupno 05.06.2019

[14] https://www.kirupa.com/html5/emoji.htm, dostupno 02.05.2019.

[15] http://thenewcode.com/957/Native-Modal-Windows-in-HTML5-Using-the-dialog-Element, dostupno 08.06.2019

[16] https://www.bitdegree.org/learn/html5-tags/, dostupno 14.07.2019

[17] https://www.w3schools.com/html/html5_intro.asp, dostupno 14.07.2019

[18] https://www.c-sharpcorner.com/uploadfile/667ddf/the-embed-tag-in-html5/, dostupno 15.07.2019

[19] https://www.tutorialrepublic.com/, dostupno 01.09.2019

[20] https://www.quackit.com/html_5/tags/html_source_tag.cfm, dostupno 16.07.2019

[21] https://www.html.am/tags/html-source-tag.cfm, dostupno 16.07.2019

[22] https://html.spec.whatwg.org, dostupno 16.07.2019

[23] https://www.dacast.com/blog/optimizing-html5-video-streaming/, dostupno 20.07.2019

[24] https://www.html5rocks.com/en/tutorials/video/basics/, dostupno 20.07.2019

[25] https://www.sitepoint.com/5-great-html5-video-players/, dostupno 22.07.2019

[26] https://blog.bitsrc.io/5-open-source-html5-video-players-for-2018-38fa85932afb, dostupno 01.08.2019

[27] https://blog.teamtreehouse.com/building-custom-controls-for-html5-videos, dostupno 01.08.2019.

[28] https://www.makeuseof.com/tag/audio-file-format-right-needs/, dostupno 01.08.2019.

[29] https://www.lambdatest.com/aac-audio-file-format, dostupno 02.08.2019.

[30] https://www.sitepoint.com/10-jquery-html5-audio-players/, dostupno 03.08.2019

[31] https://serversideup.net/what-is-amplitudejs/, dostupno 03.08.2019

[32] https://coconut.co/webvtt, dostupno, 04.08.2019

[33] https://code.tutsplus.com/tutorials/html5-audio-and-video-what-you-must-know--net-15545, dostupno 10.08.2019.

[34] https://svgontheweb.com/, dostupno 10.08.2019

[35] https://www.patreon.com/posts/svg-2-status-19078074, dostupno 17.08.2019.

[36] https://tutorial.techaltum.com/html5-svg.html, dostupno 18.08.2019.

[37] https://www.lifewire.com/why-use-html5-canvas-3467995, dostupno 18.08.2019.

Popis slika

Slika 4.1 oznake za strukturu stranice 11
Slika 4.2 opis emotikona Hamburger 14
Slika 4.3 numerički kod emotikona Hamburger 14
Slika 4.4 emotikon Hamburger na različitim platformama 16
Slika 5.1 embed video player u Chrom-u 19
Slika 5.2 embed opcija picture-in-picture 19
Slika 5.3 source opcija picture-in-picture 19
Slika 5.4 source video player u Chrom-u 19
Slika 5.5 embed video player u Edgeu 20
Slika 5.6 source video player u Edgeu 20
Slika 5.7 source video player u Firefox-u 20
Slika 5.8 embed video player u Operi 21
Slika 5.9 source video player u Operi 21
Slika 6.1 prikaz slika u različitim širinama preglednika 23
Slika 7.1 Video.js player 27
Slika 7.2 MediaElement.js player 27
Slika 7.3 prilagođeni video player u Chrom-u 33
Slika 7.4 prilagođeni video player u Edge-u 33
Slika 7.5 prilagođeni video player u Firefox-u 33
Slika 7.6 prilagođeni video player u Operi 33
Slika 7.7 HTML5 audio player u Chrom-u 36
Slika 7.8 HTML5 audio player u Edge-u 36
Slika 7.9 HTML5 audio player u Firefox-u 36
Slika 7.10 HTML5 audio player u Operi 36
Slika 7.11 SoundManager 2 audio player 37
Slika 7.12 SoundManager 2 povećana slika audio kontrola 37
Slika 7.13 Amplitude.js audio player 38
Slika 7.14 odabir podnaslova u video player-u 41
Slika 7.15 uređivač podnaslova Closed Caption Creator 42
Slika 8.1 prikaz SVG datoteke pomoću različitih oznaka 45
Slika 8.2 izgled kreiranih SVG oblika 46
Slika 8.3 izgled SVG animacije 46
Slika 8.4 prikaz dobivenih objekata u canvas-u 49

Slika 8.5 prikaz dodane slike u canvas-u.....49

Prilozi

Prikaz HTML5 elementa može se vidjeti na web adresi:

http://arwen.unin.hr/~edevcic/HTML5%20multimedija/index.html