

Uvođenje multimedijских elemenata na web stranicu

Babić, Martina

Undergraduate thesis / Završni rad

2015

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Graphic Arts / Sveučilište u Zagrebu, Grafički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:216:651421>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-12-21**



Repository / Repozitorij:

[Faculty of Graphic Arts Repository](#)



SVEUČILIŠTE U ZAGREBU

GRAFIČKI FAKULTET

ZAVRŠNI RAD

Martina Babić



Sveučilište u Zagrebu
Grafički fakultet

Smjer: tehničko-tehnološki

ZAVRŠNI RAD
UVOĐENJE MULTIMEDIJSKIH ELEMENATA
NA WEB STRANICU

Mentor:

doc.dr.sc. Tajana Koren Ivančević

Student:

Martina Babić

Zagreb, 2015.

SAŽETAK

U radu se prikazuju načini i mogućnosti uvođenja različitih multimedijских elemenata na *web* stranicu. Analizira se prikladnost pojedinih formata audio i video datoteka te formata za uvođenje slika na *web* stranicu. Uspoređuje se kompatibilnost pojedinih formata s najpoznatijim *web* preglednicima kao što su *Google Chrome*, *Mozilla Firefox* i *Internet Explorer* te kompatibilnost formata i HTML *tagova* za uvođenje multimedijских elemenata. Rezultati su prikazani na *web* stranici, koja sadržava sve multimedijские elemente, dakle tekst, slike, audio i video datoteke te animaciju. *Web* stranica s multimedijским elementima izrađena je isključivo pomoću HTML-a i CSS-a. Time se prikazuje na koje se načine multimedijским elementima može utjecati na korisnika. Isto tako, prikazuje se kako i koji od tih elemenata doprinose vizualnom dojmu, interaktivnosti i dinamičnosti *web* stranice, odnosno kako se pomoću njih može upravljati različitim informacijama s ciljem kvalitetnijeg prijenosa istih.

KLJUČNE RIJEČI: multimedijски elementi, *web*, HTML, CSS

ABSTRACT

This paper shows all the ways and possibilities of introducing different multimedia elements to a web page. The suitability of individual formats of audio and video files are analysed, along with the formats for implementing images to a website. The compatibility of certain formats with the most common web browsers are compared, with an emphasis on the compatibility of HTML tags for each multimedia element. Results are displayed on a web page, which contains all the multimedia elements, i.e. text, images, audio and video files and an animation. The web page with multimedia elements is made exclusively in HTML and CSS. The multimedia elements and their effects on the user are presented. Additionally, the effects of these elements on the visual appearance are analyzed, i.e. how to use them to manage a different types of information, with the aim for their better transmission.

KEY WORDS: multimedia elements, *web*, HTML, CSS

SADRŽAJ

1. UVOD	1
2. TEORIJSKI DIO.....	2
2.1 HTML	2
2.2 CSS.....	5
2.3 Uvođenje teksta	8
2.3.1 Uvođenje linkova	14
2.3.2 Uvođenje liste.....	16
2.4 Uvođenje slika.....	18
2.5 Uvođenje zvuka/audia.....	21
2.6 Uvođenje videa	23
2.7 Uvođenje animacije.....	26
3. EKSPERIMENTALNI DIO.....	30
4. ZAKLJUČAK.....	45
5. LITERATURA	46

1. UVOD

U teorijskom dijelu ovoga rada predstavljaju se HTML i CSS, opisni jezici koji se koriste za izradu *web* dokumenta. Unutar HTML dokumenta, uz korištenje CSS-a moguće je uvesti sve multimedijske elemente istovremeno, dakle tekst, slike, animacije, video i audio datoteke te interaktivnost. Predstavljaju se osnovne tehničke značajke multimedijskih elemenata, te koji su načini njihovog uvođenja u HTML dokument.

U eksperimentalnom dijelu rada prikazan je razvoj *web* stranice korištenjem HTML-a i CSS-a. Ponajprije, prikazana je primjena elemenata spomenutih u teorijskom dijelu. Ispituje se funkcionalnost elemenata i *tagova* kojima se prikazuju uspoređujući najpoznatije *browsers* kao što su *Google Chrome*, *Mozilla Firefox* i *Internet Explorer*. Također, analiziraju se i naglašavaju problemi koji mogu nastati u procesu izrade *web* stranice.

Cilj ovoga rada je izradom funkcionalne *web* stranice prikazati svrhu i mogućnosti multimedijskih elemenata na internetu.

2. TEORIJSKI DIO

2.1 HTML

Razvojem interneta došlo je do pojave nove vrste dokumenta – hiperteksta. Hipertekst je skup dokumenata u elektroničkom obliku, uglavnom tekstualnog i slikovnog sadržaja, međusobno povezanih elektroničkim poveznicama (*linkovima*). Aktiviranjem *linkova* korisniku je omogućeno prelaženje s jednog dokumenta na drugi, koji se na prvi sadržajno nastavlja, a time i oblikovanje slijeda sadržaja prema vlastitom interesu. Hipertekstualni dokumenti izrađuju se pomoću HTML-a.

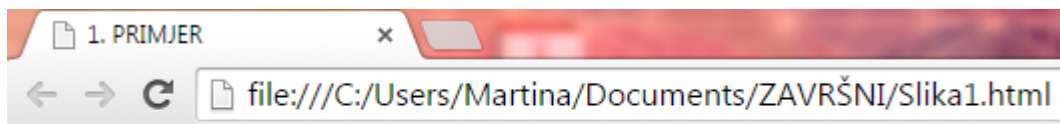
HTML je skraćenica od *HyperText Markup Language*, a to je opisni jezik koji se koristi za oblikovanje *web* dokumenata. Taj se jezik sastoji od niza oznaka (eng. *tags*) kojima se opisuje struktura pojedine *web* stranice, odnosno dokumenta. *Web* preglednici kao što su *Google Chrome*, *Internet Explorer*, *Mozilla Firefox* i *Safari* koriste upravo te *tagove* za određivanje načina prikaza sadržaja HTML stranice. Dakle, njihova je uloga čitanje HTML dokumenata i prikazivanje istih kao *web* stranice. Svaki HTML dokument mora imati ekstenziju `.html` ili `.htm`.

Svi *tagovi* pišu se unutar znakova “<” i “>”. Prilikom njihova pisanja nije važno koriste li se kurenti ili verzali zato što HTML nije *case sensitive* jezik. Unatoč tome, preporučuje se korištenje kurenata budući da su neki drugi programski jezici koji se koriste u kombinaciji s HTML-om *case sensitive* i podržavaju samo kurente. Većina HTML *tagova* piše se u parovima npr. `<html>` i `</html>`, pri čemu je prvi *tag* početni *tag* koji daje *web* pregledniku informaciju o tome gdje neki element počinje. Drugi je *tag* završni *tag* i vrlo lako se prepoznaje po “/” ispred njegovoga imena. On označava mjesto na kojemu prestaje djelovanje početnog *taga*. Također, postoje i tzv. samostalni *tagovi*, odnosno prazni elementi koji nemaju sadržaj i koriste samo jedan *tag*, a primjenjuju se, primjerice, kod ubacivanja praznih redaka (`</br>`). Sami *tagovi*, kao i sve ono što se nalazi između početnog i završnog *taga*, nazivaju se HTML elementima.

Izrada HTML stranice moguća je u bilo kojem tekstualnom editoru kao što su, primjerice, *Notepad* ili *Wordpad*. Osim tekstualnih editora, moguće je koristiti i vizualne editore za izradu *web* stranica kao što su *Macromedia Dreamweaver* ili *Microsoft Frontpage*. Takvi se editori još nazivaju i *WYSIWYG* (*What You See Is What You Get*) editorima. Oni imaju gotovu strukturu tablica i formi koje se mogu koristiti tijekom rada.

Sve HTML stranice imaju zajedničku osnovnu strukturu koja je prikazana na slici 1. Ta osnovna struktura započinje i završava `<html>` *tagom*. Unutar tog *taga* se nalazi kompletan sadržaj pojedine *web* stranice, a tim je sadržajem *web* pregledniku dana informacija kako se unutar njega nalazi HTML kod stranice. Već spomenuti sadržaj sastoji se od dvije cjeline, zaglavlja i tijela stranice. Zaglavljje se označava `<head>` *tagom* dok se tijelo označava `<body>` *tagom*. *Head* element smješta se prije *body* elementa te sadrži sve informacije o stranici. Informacije koje se pišu u *head* dijelu ne prikazuju se unutar prozora preglednika. Jedna od mogućih informacija unutar *heada* jest *title* element. Njegov se sadržaj tako prikazuje u naslovnoj traci dok je sadržaj naveden unutar *body* elementa vidljiv unutar prozora preglednika [1, 2].

```
<html>
  <head>
    <title> 1. PRIMJER </title>
  </head>
  <body>
    Ovo je prikaz osnovne strukture HTML stranice.
  </body>
</html>
```



Ovo je prikaz osnovne strukture HTML stranice.

Slika 1. Osnovna struktura HTML stranice

Razvoj tehnologije kroz povijest rezultirao je razvojem šest verzija HTML-a (HTML 1991., HTML+ 1993., HTML 2.0 1995., HTML 3.2 1997., HTML 4.01 1999. i HTML5 2012.). Pojavom HTML-a 3.2 pojavili su se *tagovi* poput fonta, a samim time i *color* atributi. Valja napomenuti kako atributi, uz elemente i tekstualne znakove koji ne pripadaju skupini ASCII (eng. *American Standard Code for Information Interchange*) znakova, predstavljaju jednu od komponenti HTML-a koja daje dodatne informacije o korištenim HTML elementima. Pojavom te verzije, promjena boje ili fonta morala se raditi za svaki element posebno. Pri tome je došlo do niza problema, a izrada *web* stranice postala je vrlo opsežan, skup i dugotrajan proces. Taj je problem riješen kreiranjem CSS-a.

2.2 CSS

CSS je skraćenica za *Cascading Style Sheets*, a to su kaskadni stilovi koji definiraju na koji će se način prikazivati HTML elementi. Preciznije, CSS-om je moguće odvojiti HTML strukturu *web* stranice i pripadajućeg sadržaja od vizualnih i stilskih elemenata pri čemu opseg HTML koda postaje manji i znatno pregledniji. Na taj je način *web* elemente moguće prikazati na točno određenom mjestu i na točno određeni način. Njegovom upotrebom može se utjecati na različite segmente pojedine *web* stranice kao što su:

- opcije klasifikacija (mogućnost kontrole ponašanja i odnosa između različitih elemenata sadržanih na stranici)
- opcije okvira (mogućnost kontrole svih okvira na stranici poput listi, slika i tablica; npr. pozicija, širina, debljina)
- pozicioniranje elemenata
- podešavanje *paddinga* (mogućnost kontrole količine praznog prostora oko bilo kojeg elementa na stranici)
- podešavanje pozadine (npr. boja pozadine)
- uređivanje listi
- uređivanje margina
- uređivanje tablica
- uređivanje teksta
- veličina elemenata

Spomenuti kaskadni stilovi mogu se pisati na tri različita načina:

- eksterno
- interno
- *inline*

Eksterni stilovi predstavljaju zasebne dokumente sa .css ekstenzijom, a u HTML dokument ih uvodimo linkom. Eksterni stilovi mogu se napisati u bilo kojem tekstualnom editoru. Tako se editiranjem jednog dokumenta mijenja izgled cijele *web* stranice, a ona se može sastojati od mnogo podstranica. Eksterna CSS datoteka je odličan način za primjenu nekog stila na više različitih *web* stranica. To se postiže tako da svaka *web* stranica ostvari “vezu” do eksterne CSS datoteke `<link>` *tag* unutar `<head>` *taga*, što je prikazano na slici 2.

```
<head>
<link rel="stylesheet" type="text/css" href="s1.css">
</head>
```

Slika 2. Primjer eksternog stila

Na slici 3. nalazi se primjer internog stila. Taj je stil namijenjen korištenju u onim situacijama kada svaka stranica zahtjeva zaseban stil. Takav oblik CSS-a definira se unutar `<head>` i `<style>` *taga*.

```
<head>
<style>
hr {color:cyan;}
p {
font-family:calibri;
font-size:12px;
}
</style>
</head>
```

Slika 3. Primjer internog stila

Inline način uređivanja CSS-a bi se, u pravilu, trebao izbjegavati ili barem koristiti u što manjoj mogućoj mjeri. U tom se slučaju stil piše u liniji elementa, točnije u *style* atributu. Primjer pisanja takvog stila prikazan je na slici 4 [3].

```
<p style="margin-left:25px;">
ovaj je paragraf pomaknut u lijevo.
</p>
```

Slika 4. Primjer inline stila

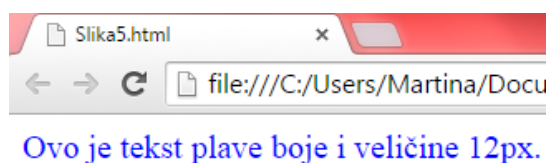
CSS se sastoji od tri dijela, a to su:

- selektori
- svojstva
- vrijednosti

Selektori su HTML elementi koje želimo stilizirati. Svojstva i vrijednosti zajedno čine deklaracije. Svojstva su stilski atributi koje možemo mijenjati. Svako svojstvo mora sadržavati vrijednost. Svaka deklaracija završava sa “;”, a grupe deklaracija počinju i završavaju vitičastim zagradama. Preporučuje se da se svaka deklaracija stavlja u novi redak kako bi CSS bio što pregledniji. Na slici 5. nalazi se primjer u kojemu je *p* selektor, a `{color:blue; font-size:20px;}` deklaracija. Pritom je *color* svojstvo, a *blue* vrijednost. Stoga će svi paragrafi ovog HTML dokumenta biti plave boje i veličine fonta 20 px. Osim selektora koji su HTML elementi, kroz CSS možemo specificirati i vlastite selektore kroz “*id*” i “*class*”. “*id*” selektor koristi se za specificiranje stila za jedan, unikatni element. On koristi *id* atribut HTML elementa i definira se sa “#”. “*id*” se nikada ne započinje brojkom. Dakle, `id="1p"` nije dobar način zapisa, a `id="p1"` je. “*Class*” selektor se upotrebljava za specificiranje stila za grupu elemenata. Na taj način, jedan stil možemo primijeniti na razne HTML elemente koji imaju istu klasu. “*Class*” selektor koristi *class* atribut HTML elementa i definira se sa “.”. Svrha i uporaba ovih selektora bit će vidljiva na primjerima kroz sljedeća poglavlja [4].

```
<html>
<head>
<style>
p {
color:blue;
font-size:20px;
}
</style>
</head>

<body>
<p> Ovo je tekst plave boje i veličine 12px. </p>
</body>
</html>
```



Slika 5. Primjer strukture i sintakse CSS-a

2.3 Uvođenje teksta

Tekst je neizbježna stavka svake *web* stranice. Prilikom uvođenja teksta na *web* na raspolaganju se nalazi pregršt mogućnosti i opcija zbog čega će okosnica ovog poglavlja biti prikaz upravljanja tekstom na primjeru izrađene *web* stranice.

Najjednostavniji način uvođenja teksta na *web* stranicu je putem paragraf *taga*, pri čemu se željeni tekst unese unutar znakova `<p>` i `</p>`, a njegov se stil definira u CSS-u. Ako se u CSS-u stil definira kao na slici 6., svakim pozivanjem paragraf *taga* u *body* dijelu HTML dokumenta na *web* stranici pojaviti će se tekst s točno navedenim svojstvima.

```
p {
  position:absolute;
  margin:50px 50px;
  font-family:Calibri;
  font-size:20px;
  color:gray;
  text-align:justify;
}
```

Slika 6. Primjer stilizacije teksta pomoću paragraf *taga*

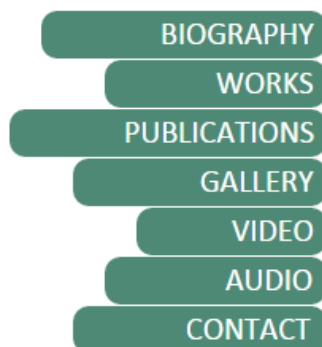
Drugi način uvođenja teksta, prikazan na slici 7., je preko *div* elemenata što je vrlo korisno i praktično ako se unutar jednog HTML dokumenta nalazi više tekstova koji se stilski međusobno razlikuju. Pritom je za svaki tekst posebno moguće definirati željeni font, veličinu fonta, boju, poziciju itd.

```
#imeip {
  position:absolute;
  top:20px;
  left:400px;
  font-family:Calibri;
  font-size:80px;
  font-weight:500;
  color:gray;
}

#komp {
  position:absolute;
  top:20px;
  left:875px;
  font-family:Calibri;
  font-size:80px;
  font-weight:100;
  color:gray;
}

.tekst {
  position:absolute;
  top:2px;
  font-family:Calibri;
  font-size:20px;
  color:white;
}
```

antonio babic | composer



Slika 7. Uvođenje teksta preko *div* elemenata

Za odabir fonta kojim će biti ispisan tekst, važno je da je upravo taj font sadržan na računalu posjetitelja određene *web* stranice. Većina poznatih *browsersa* koristi *Times New Roman* kao *defaultni* font. Ako se dizajner neke stranice odluči za onaj font koji posjetitelj nema instaliran na računalu, *browser* će automatski ispisati takav tekst svojim *defaultnim* fontom. Kako bi se takvo što izbjeglo, najčešće postoji lista nekoliko fontova koji se mogu koristiti za svaku pojedinu stranicu kao npr. Arial, Helvetica i sans-serif što *browseru* daje mogućnost da prvi zamijeni drugim ili oba trećim u slučaju da ne može pronaći redom navedene. U CSS-u se naredbom *font-family* odabire željeni font pri čemu se, zapravo, definira obitelj u koju font pripada. To može biti:

- generička obitelj fontova koja predstavlja grupu obitelji fontova sličnog izgleda npr. serif ili sans-serif
- obitelj fontova kojom definiramo specifični font npr. Arial, Helvetica, Calibri

Oni fontovi čije se ime sastoji od više riječi pišu se pod navodnim znakovima.

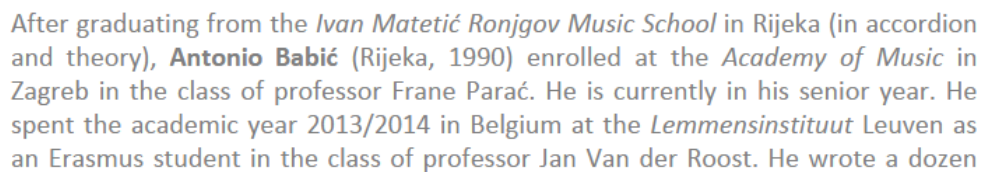
Na slici 7. može se uočiti sljedeća naredba, a to je *font-size*. Njome se definira veličina ispisanog teksta na stranici. Ta veličina može biti izražena na dva načina:

- apsolutno
- relativno

Apsolutna veličina je ona koja postavlja veličinu teksta na zadanu veličinu i korisnik ju ne može promijeniti u svim *browsersima*. Relativna veličina je ona koja je relativna ostalim elementima koji okružuju tekst i nju korisnik može promijeniti u svim *browsersima*. Veličinu fonta moguće je zadati u pikselima i em jedinicama pri čemu je 1em jednak 16px, tj. ako se veličina u pikselima podijeli sa brojem 16, dobiva se veličina u em jedinicama. Ukoliko veličina fonta nije zadana, ona će po *defaultu* iznositi 16px, odnosno 1em.

Naredbom *margin:50px 50px;* postignuta je udaljenost teksta za 50px sa svih strana od objekta unutar kojega se nalazi. Na slici 8., tekst je udaljen za 50px od rubova okvira u kojemu se prikazuje.

Za horizontalno pozicioniranje teksta koristi se naredba *text-align*. On može biti poravnat u lijevu ili desnu stranu, u sredinu te obostrano. Na slici 8. je tekst poravnat obostrano što je postignuto *text-align:justify;* svojstvom.

The image shows a rectangular box with a thin green border. Inside the box, there is a paragraph of text. The text is justified (aligned to both the left and right edges of the box) and has a consistent margin from all four sides of the box. The text reads: "After graduating from the Ivan Matetić Ronjgov Music School in Rijeka (in accordion and theory), Antonio Babić (Rijeka, 1990) enrolled at the Academy of Music in Zagreb in the class of professor Frane Parać. He is currently in his senior year. He spent the academic year 2013/2014 in Belgium at the Lemmensinstituut Leuven as an Erasmus student in the class of professor Jan Van der Roost. He wrote a dozen".

After graduating from the *Ivan Matetić Ronjgov Music School* in Rijeka (in accordion and theory), **Antonio Babić** (Rijeka, 1990) enrolled at the *Academy of Music* in Zagreb in the class of professor Frane Parać. He is currently in his senior year. He spent the academic year 2013/2014 in Belgium at the *Lemmensinstituut* Leuven as an Erasmus student in the class of professor Jan Van der Roost. He wrote a dozen

Slika 8. Primjer uporabe *margin* i *text-align* taga

U navedenim se primjerima koristi stilska deklaracija *font-weight*, a čija vrijednost može biti jedna od sljedećih:

- *bold* (*boldani* tekst)
- *bolder* (tekst koji je deblji od normalnog *bolda*)
- *lighter* (tekst koji je tanji od normalnog *bolda*)
- *normal* (tekst bez *boldanog* formatiranja)
- vrijednost od 100 do 900 (pri tome je 100 najtanji, a 900 najdeblji tekst)

Također postoje i još dvije opcije kojima se može postići *boldanje* teksta. Te su opcije praktične u onim situacijama u kojima je potrebno zadebljanje samo određenih segmenata teksta kao što se može vidjeti na slici 8. To se postiže

otvaranjem *bold* ili *strong taga* ispred onih dijelova teksta koje je potrebno *boldati* te zatvaranjem tih istih *tagova* na željenim mjestima čime će se prekinuti njihovo djelovanje. Kada se unutar nekog elementa nalazi drugi element potrebno je paziti na redoslijed zatvaranja. Primjer *boldanja* teksta spomenutim *tagovima* prikazan je na slici 9 [5, 6].

```
<b> Antonio Babić </b>  
<strong> Antonio Babić </strong>
```

Slika 9. Primjer boldanja teksta različitim tagovima

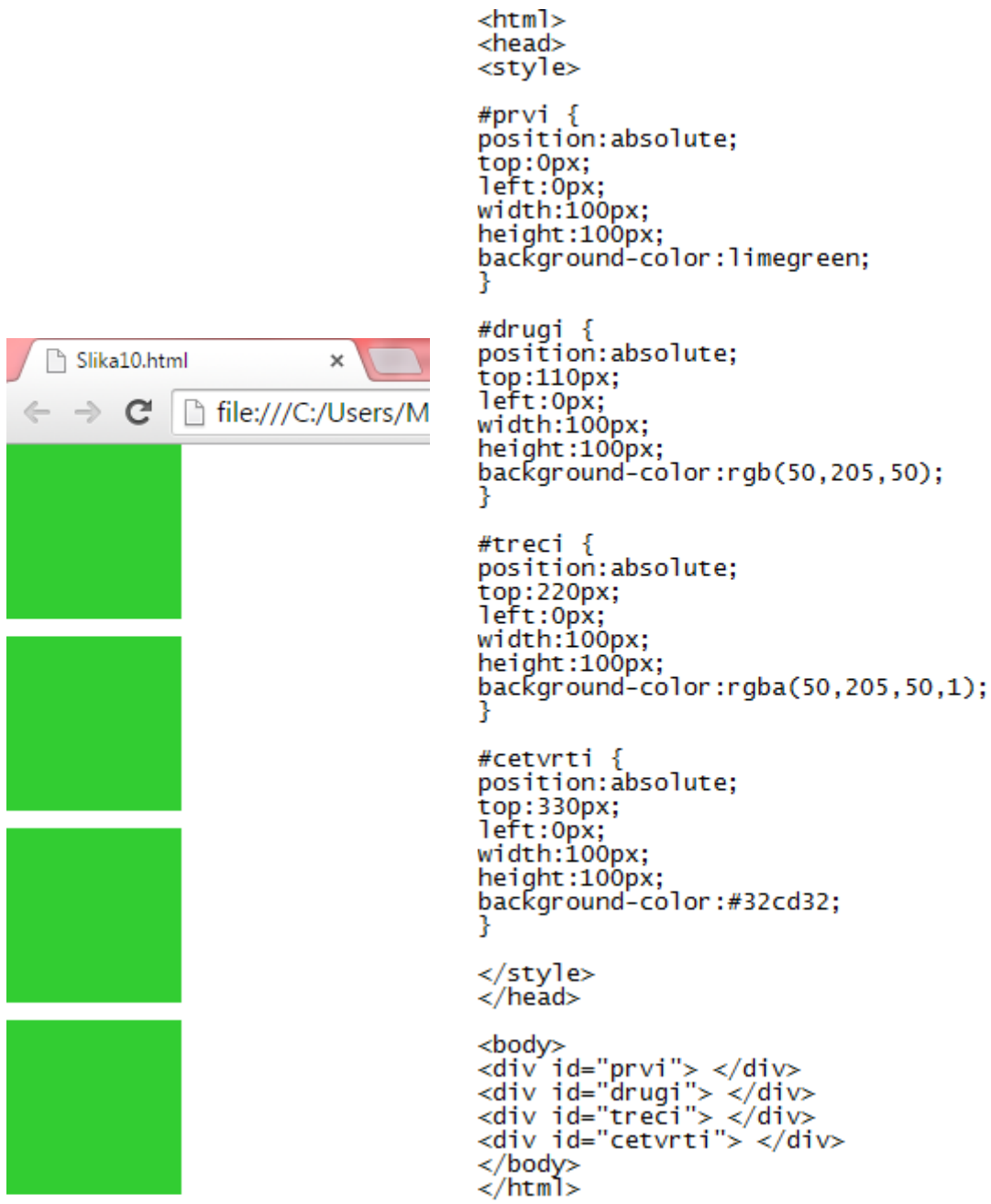
Pri uvođenju teksta vrlo je bitan i razmak između redaka. Njime se može regulirati čitljivost koja se znatno smanjuje ako je zadani razmak premalen ili prevelik. To se postiže pomoću *line-height* naredbe čija se vrijednost definira u postotcima, a ona po *defaultu* iznosi oko 110-120% veličine fonta. Također se može utjecati i na razmak između slova pomoću *letter-spacinga*, a isto tako i na razmak između riječi pomoću *word-spacinga*.

Kada je riječ o bojama, CSS-om se one mogu definirati na četiri različita načina što prikazuje slika 10. Prvi je način putem imena boja pri čemu se *color* atributu dodaje ime same boje. U početku se tako moglo pozvati samo šesnaest boja kao što su *aqua*, *black*, *blue*, *fuchsia*, *green*, *gray*, *lime*, *maroon*, *navy*, *olive*, *purple*, *red*, *silver*, *teal*, *white* i *yellow*. Razvojem interneta taj je broj znatno narastao pa ih danas na raspolaganju ima oko 140.

Drugi način je putem *RGB* vrijednosti. Riječ je o vrijednostima od 0 do 255 za svaku od tri boje *RGB* sustava, dakle, crvenu (*red*), zelenu (*green*) i plavu (*blue*) boju čime se izbor širi na 16.7 milijuna različitih boja.

Treći način je putem *RGBa* sustava koji uz *RGB* boje koristi i alfa kanal. Dodana komponenta *a* predstavlja transparentnost boje, a njene vrijednosti mogu biti decimalni brojevi od 0 (potpuno prozirno) do 1 (potpuno neprozirno).

Zadnji način je heksadecimalnim zapisom koji koristi šest znakova. Pritom prva dva znaka označavaju crvenu, druga dva zelenu, a zadnja dva znaka plavu boju, a vrijednosti su od 0 do 9 i od a do f, gdje je f najveća vrijednost. Da bi takav zapis bio potpun prije kombinacije slova i brojeva dodaje se znak "#".



Slika 10. Primjer različitih načina uvođenja boje u HTML dokument

Budući da je na većini *web* stranica upravo tekst onaj element koji je najzastupljeniji, vrlo je važno da je lako čitljiv. To se postiže odabirom prikladnih boja, fontova, isticanjem raznih segmenata teksta itd. Osim navedenog *boldanja*, dijelovi teksta mogu se isticati i na brojne druge načine.

Jedan od načina je pomoću *font-style* naredbe čije vrijednosti mogu biti sljedeće:

- *normal*
- *italic*
- *oblique*

Time se utječe na naglašenost, odnosno nenaglašenost dijelova teksta u vidu njegove nakošenosti. Kao i kod *boldanja*, kada je riječ samo o pojedinim dijelovima teksta koristit će se *italic tag* čijim ispisivanjem ispred određenih riječi ili rečenica, iste postaju ukošene što se može vidjeti na slici 11. Kada se unutar nekog elementa nalazi drugi element potrebno je paziti na redosljed zatvaranja.

```
<strong> When I Become Grass </strong><i>for baritone and chamber orchestra </i>(2013)
```

When I Become Grass for baritone and chamber orchestra (2013)

Slika 11. Primjer uporabe *italic taga*

Uporabom stilske deklaracije *text-decoration*, opcije su sljedeće:

- *underline* (podcrtani tekst)
- *overline* (linija iznad teksta)
- *line-through* (linija preko teksta)
- *blink* (tekst koji treperi)
- *none* (bez formatiranja)

Alternativa za neke od tih opcija je, recimo, otvaranje i zatvaranje `` taga za liniju preko teksta i `<ins>` taga za podcrtavanje teksta.

Također, dijelovi teksta mogu se naglasiti i malom kapitalom što ostvarujemo otvaranjem i zatvaranjem `<small>` taga ili stilskom deklaracijom *font-variant* čije vrijednosti mogu biti *normal* i *small-caps*.

Još jedna mogućnost jest *text-transform* naredba koja se koristi za prikazivanje teksta u verzalima (*uppercase*), kurentima (*lowercase*) ili za kapitaliziranje prvog slova u svakoj riječi (*capitalize*).

2.3.1 Uvođenje linkova

World Wide Web je mreža stranica koje se nalaze na *web* serverima diljem svijeta i međusobno su, na mnogo različitih načina, povezane. Ta je povezanost ostvarena pomoću *hyperlinkova*, odnosno poveznica. Poveznica (eng. *link*) najjednostavniji je oblik interaktivnosti na *webu*, a ona može biti riječ, rečenica, slika ili bilo što drugo. *Klikom* na nešto od navedenog, otvara se mogućnost prelaska s jedne *web* stranice na drugu (*link* koji vodi na stranicu *World Wide Web*), unutar određene domene (lokalno) ili pak unutar jedne *web* stranice (*internal*). Za uvođenje *linkova* na *web* koristi se `<a>` *tag*. Kako bi se točno definiralo kamo određeni *link* vodi, `<a>` *tagu* se dodaje *href* (engl. *Hypertext REFerence*) atribut. U slučaju u kojemu neki *link* vodi na novu *web* stranicu, unutar *href* atributa mora se nalaziti cijela *web* adresa željene stranice poznata pod nazivom apsolutni *URL* (engl. *Uniform Resource Locator*). To je *web* adresa koja se upisuje u *browser* i vodi na točno određenu *web* stranicu. Svaka stranica ili datoteka ima svoj jedinstveni *URL*. Slika 12. prikazuje i *target* atribut koji se nerijetko piše unutar `<a>` *taga*. Njime se, naime, određuje na koji način će se otvoriti dokument na koji *link* navodi. Mogućnosti su sljedeće:

- *target*="*_blank*" (dokument će se otvoriti u novom prozoru (eng. *tab*))
- *target*="*_self*" (dokument će se otvoriti u istom prozoru)
- *target*="*_parent*" (dokument će se otvoriti u okviru koji je "nadređen" linku)
- *target*="*_top*" (dokument će se otvoriti u trenutno otvorenom okviru, poništavajući sve ostale)

```
<a href="http://www.zamp.hr/clanak/pregled/810/antonio-babic-skladanje-je-poput-poniranja-u-vlastitu-nutrinu" target="_blank">Interview</a>
```

Slika 12. Primjer uporabe *href* i *target* atributa

Ako se na stranici nalazi dovoljno sadržaja, moguće je koristiti *linkove* koji vode na neki drugi dio HTML dokumenta (npr. *link* koji vodi na neki drugi paragraf ili neki drugi dio teksta). Za to je samo potrebno koristiti *id* attribute u kombinaciji sa sidrom (eng. *anchor*).

Linkovi se mogu koristiti i za slanje *maila* pri čemu se, na mjesto gdje se obično upisuje *web* adresa, upisuje *mail* adresa npr. <mailto:neka.adresa@gmail.com>. Na taj se način sa stranice šalje *mail* na željenu adresu. Time je posjetitelju stranice na kojoj je sadržan takav link omogućeno vrlo brzo stupanje u kontakt s dotičnom osobom, tvrtkom itd.

Kao što je već navedeno, *linkovi* mogu biti i slike što je prikazano na slici 13. *Klikom* na njih također se ostvaruje prijelaz na neki drugi sadržaj.



```
<a href="https://www.facebook.com/disalteration?fref=ts" target="_blank"></a>
```

Slika 13. Primjer slike kao linka

Da taj sadržaj može biti bilo što, može se vidjeti i na primjeru izrađene *web* stranice gdje *link* vodi na *pdf* dokument kao na slici 14.

```
<a href="cantus.pdf" target="_top">Article</a>
```

Slika 14. Primjer linka koji vodi na pdf dokument

Linkovi se mogu stilizirati na različite načine. Prema stanju se dijele na:

- posjećene
- neposjećene
- aktivne
- *mouse over* (kada se strelicom miša pređe preko linka)

Ako se ne definira drugačije, u *browseru* će neposjećeni *linkovi* biti podcrtani i plave boje, posjećeni podcrtani i ljubičaste boje dok će aktivni *linkovi* također biti podcrtani, ali crvene boje. Kod njihova stiliziranja, potrebno je paziti na redoslijed pisanja naredbi. U slučaju ne pridržavanja tog redoslijeda, željeni stil neće biti efektan. Pravilo nalaže da se u CSS-u stil piše sljedećim redoslijedom: *a:link*, *a:visited*, *a:hover*, *a:active*.

2.3.2 Uvođenje liste

Većina *web* stranica sadrži liste. Navigacijski meni koji gotovo svaka stranica ima, nije ništa drugo nego lista *linkova* koja vodi na druge HTML dokumente. One služe organizaciji elemenata i omogućavaju korisniku lakši pregled informacija. Liste mogu biti:

- uređene (*ordered*)
- neuređene (*unordered*)
- definicijske (*definition*)

Uređene liste su one kod kojih je redoslijed stavki bitan. Ispred tih stavki mogu se nalaziti arapski ili rimski brojevi te slova. Ako se ne definira drugačije, uređene će liste po *defaultu* ispred stavki imati arapske brojeve. Označavaju se `` *tagom* unutar kojega se dodaje *type* kojim se naznači o kakvoj je listi riječ. Nakon što je to definirano, slijedi nabrojanje stavki koje se vrši tako da se ispred otvori, a nakon svake stavke zatvori `` *tag*. Primjeri uređenih listi:

- lista s verzalnim slovima

```
<ol type="A">  
</ol>
```
- lista s kurentnim slovima

```
<ol type="a">  
</ol>
```
- lista s velikim rimskim brojevima

```
<ol type="I">  
</ol>
```
- lista s malim rimskim brojevima

```
<ol type="i">  
</ol>
```

Neuređene se liste na *webu* češće upotrebljavaju. To su one čije se stavke ne trebaju prikazivati određenim redoslijedom. Ispred stavki takvih listi mogu se nalaziti puni i prazni kružići te kvadratići. Ako se ne postavi drugačije, one će po

defaultu ispred stavki imati pune kružice. Neuređene liste označavaju se `` *tagom* unutar kojega se dodaje *type* kojim se naznači o kakvoj je listi riječ. Nakon što je to definirano, slijedi nabrojanje stavki koje je identično kao i kod uređenih listi i vrši se tako da se ispred otvori, a nakon svake stavke zatvori `` *tag*. Primjeri neuređenih listi:

- lista s punim kružićima (diskovima)

```
<ul type="discs">
</ul>
```

- lista s praznim kružićima (circle)

```
<ul type="circle">
</ul>
```

- lista s kvadratićima (square)

```
<ul type="square">
</ul>
```

Definicijske liste koriste se pri grupiranju pojmova i njihovih definicija u listu, a način na koji se one mogu upotrebljavati prikazan je na slici 15. Takve liste se označavaju `<dl>` *tagom* i sadrže dvije stavke:

- `<dt>` *tag* koji definira pojam u listi
- `<dd>` *tag* definira definiciju pojma u listi

HTML	<pre><html> <head> </head> <body> <dl> <dt> HTML </dt> <dd> HyperText Markup Language </dd> <dt> CSS </dt> <dd> Cascading style sheets </dd> </dl> </body> </html></pre>
CSS	
HyperText Markup Language	
Cascading Style Sheets	

Slika 15. Primjer definicijske liste

Također, liste mogu bit i ugniježdene, odnosno mogu sadržavati liste unutar liste [7].

2.4 Uvođenje slika

Stara izreka da slika govori više od tisuću riječi sasvim sigurno vrijedi i na *webu*, primarno vizualnom mediju. Slike na *web* stranicama svakako idu u prilog njihovoj posjećenosti. One ih, vrlo vjerojatno, neće učiniti sadržajno boljima, ali će zasigurno izazvati veću zainteresiranost kod posjetitelja i ostaviti bolji dojam. U HTML dokument uvode se `` *tagom* koji sadrži samo atribute, odnosno nema zatvarajući *tag*. Da bi se slika prikazala na stranici koristi se *src* atribut. Tim atributom *browseru* je dana informacija gdje pronaći točno određenu sliku. Vrijednost *src* atributa može ukazivati na *URL* (*eng. Uniform Resource Locator*) gdje se nalazi željena slika ili, lokalno gledano, to može biti putanja do slike u istom *folderu*. Osim toga mogu se koristiti i drugi atributi, kao što su:

- *alt* atribut
- *title* atribut
- *align* atribut
- *width* i *height* atribut

Alt atribut sadrži opis slike koji će se prikazati umjesto same slike. To se može dogoditi ako se pogrešno definira putanja do slike, ako se ona više ne nalazi na tom određenom mjestu ili ako je, primjerice, njezino ime pogrešno napisano.

Title atribut omogućava dodatne informacije o slici. Većina će *browsersa* sadržaj ovog atributa prikazati kada korisnik mišem prijeđe preko slike.

Align atribut se koristi za definiranje pozicioniranja slike u odnosu na druge elemente na stranici. Vrijednosti tog atributa mogu biti:

- *top* (tekst oko slike u ravnini je s vrhom slike)
- *bottom* (tekst oko slike u ravnini je s dnom slike)
- *left* (slika na lijevoj strani, tekst na desnoj strani)
- *right* (slika na desnoj strani, tekst na lijevoj)
- *middle* (tekst oko slike u ravnini je sa sredinom slike)

Ako se ne definira drugačije, njegova će vrijednost biti *align="bottom"*.

Width i *height* atributi definiraju širinu i visinu slike. Slikama na *webu* je obično potrebno više vremena da se prikažu, nego što je potrebno ostatku HTML koda. Upravo zbog toga je dobro pisati dimenzije slika unutar *img taga*. Na taj će način *browser* prikazati ostatak HTML koda ostavljajući, pritom, dovoljno mjesta za slike koje se još nisu prikazale. Ti se atributi mogu pisati i unutar CSS-a što je puno češća opcija.

U *browseru* će se slika prikazati točno na onome mjestu na kojemu se u HTML dokumentu nalazi *img tag*. Svaka slika posjeduje određenu veličinu i težinu. Veličina slike određena je njezinom širinom i visinom, a te su dimenzije izražene u pikselima. Težina slike izražava se u kilobajtima (KB). Ta su dva svojstva vrlo važna prilikom prikaza na *webu*. Naime, što je slika većih dimenzija, to je ista teža. Što je slika teža, potrebno je više vremena da se ona učita. Stoga je prilikom uvođenja slike na *web* te brojke potrebno svesti na minimum i to tako da kvaliteta slike ne bude zamjetno narušena. Još jedno pravilo je da slike za *web* moraju biti definirane u *RGB* sustavu boja s rezolucijom od 72dpi (*dots per inch*). Formati u kojima mogu biti slike koje se prikazuju na *webu* su sljedeći:

- JPEG
- GIF
- PNG
- WEBP

JPEG (*Joint Photograph Experts Group*) je jedan od najčešće korištenih formata za pohranu slike. Ovaj je format izrazito popularan na *webu* upravo zbog toga što pruža dobar kompromis između kvalitete slike i veličine datoteke. Moguće je smanjiti veličinu slikovne datoteke, a da se pritom znatno ne narušava njezina kvaliteta. Što je veći stupanj kompresije podataka to će biti manja kvaliteta same slike. Nedostaci JPEG formata su dosta izražene nepravilnosti slike i nepodržavanje prozirnosti i animacije.

GIF (*Graphic Interchange Format*) format je koji je primarno namijenjen uporabi na *webu*. Za razliku od JPEG formata, GIF format koristi kompresiju podataka bez gubitaka na kvaliteti slike. Kompresijom se slikovni zapis pretvara u oblik

koji zauzima jako malo memorije, a da pritom ne narušava njegovu kvalitetu. GIF format ograničen je na 256 boja zbog čega se koristi za pohranu jednostavnijih slika koje ne sadrže puno prijelaza boja. Preporučljiv je za prikaz logotipa, dekorativnih linija i slično. Ovaj format podržava transparentnost i animaciju.

PNG (*Portable Network Graphics*) baš kao i GIF format koristi kompresiju podataka bez gubitaka. Prvotna ideja njegovog nastanka bila je unaprjeđenje GIF formata što je kasnije rezultiralo potpunom zamjenom GIF formata PNG formatom. Podržava transparentnost, ali ne i animaciju. Važno je napomenuti kako ovaj format ne podržava CMYK prostor boja zbog čega se ne primjenjuje u tisku. U RGB prostoru boja, najkvalitetniji je grafički format. Namijenjen je za prijenos slika prosječne kvalitete na *webu*.

WEBP format je *Googleova* JPEG alternativa. Njegova prednost u odnosu na JPEG je bolja kompresija bez gubitaka na kvaliteti slike. Uporabom tog formata, dodatno se ubrzava prijenos slika i učitavanje *web* stranica [8, 9].

Na slici 16. prikazano je uvođenje navedenih slikovnih formata u HTML dokument i njihov prikaz u *Google Chrome* browseru.



```
<html>
<head>
</head>

<body>

</br>
</br>

</br>
</br>

</br>
</br>

</body>
</html>
```

Slika 16. Primjer uvođenja različitih slikovnih formata u HTML dokument

2.5 Uvođenje zvuka/audia

Svrha multimedije jest da na što zanimljiviji način predoči i prenese neku informaciju. To se svakako može postići korištenjem zvuka. Zvuk je jedan od temeljnih elemenata multimedije koji je po svojoj prirodi zvučni val, tj. analogni signal. Da bi neko računalo moglo prepoznati zvuk, zvučna kartica ima zadatak pretvoriti analognu (zvučni val) u digitalnu informaciju (binarnu). Nakon toga, digitalna informacija ostaje pohranjena na računalu u obliku datoteke. Postoje različiti formati audio datoteka, a neki od poznatijih su [10]:

- *MP3 (.mp3)*
- *WAV (.wav)*
- *WMA (.wma)*

MP3 je zvučni dio *MPEG* filea. *MPEG* je stvoren za video, a razvila ga je *Moving Pictures Expert Group*. To je najpoznatiji format za audio, a odlikuje se dobrom kompresijom (fileovi su mali) i visokom kvalitetom.

WAV format je razvijen od strane *IBM*-a i *Microsoft*a, a podržavaju ga različiti operativni sustavi poput *Windowsa*, *Macintosha* i *Linuxa*.

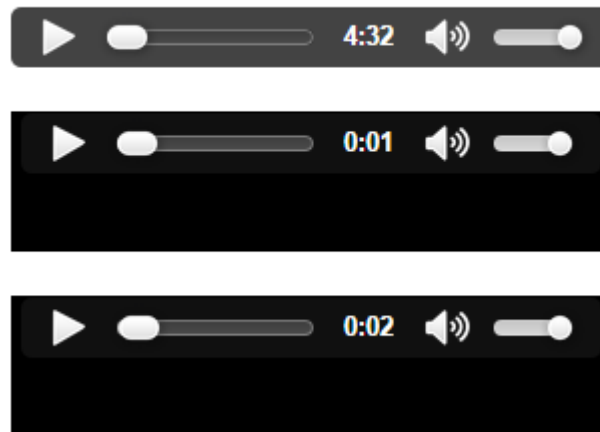
WMA je skraćenica za *Windows Media Audio* i po kvaliteti se može usporediti s *MP3* formatom. Kompatibilan je s većinom *audio* playera.

HTML5 inačica omogućila je uvođenje *audio* i video datoteka preko zasebnih tagova.

Zvuk se danas u HTML dokument može uvesti pomoću više različitih tagova što je prikazano na slici 17. Za uvođenje *audio* datoteka preporuča se `<audio>` tag, no one se mogu uvesti i preko:

- `<a>` tag
- `<object>` tag
- `<embed>` tag

Uvođenje zvuka pomoću a taga



```
<html>
<head>
</head>

<body>

<a href="Kad Budem trava.mp3">Uvođenje zvuka pomoću a taga</a>
</br>
</br>

<audio src="Queentet.mp3" type="audio/mp3" controls></audio>
</br>
</br>

<object height="70" width="300" data="E1.mp3"></object>
</br>
</br>

<embed height="70" width="300" src="Tić.mp3"></embed>

</body>
</html>
```

Slika 17. Primjer uvođenja zvuka različitim tagovima

2.6 Uvođenje videa

Unatoč tome što zahtjeva najviše memorijskog prostora, video će prenijeti najviše informacija i pri tome ostvariti najveći utjecaj. Upravo zato će se željena informacija puno uspješnije prenijeti videom nego što bi to uspjela nekim drugim multimedijским elementom. Kako kod audio datoteka, tako i kod video datoteka postoje različiti formati koji se koriste [10]. Najčešći su:

- *MP4 (.mp4)*
- *MPEG (.mpg, .mpeg)*
- *AVI (.avi)*
- *WMW (.wmw)*
- *QUICKTIME (.mov)*
- *FLASH (.swf, .flv)*
- *OGV (.ogv)*
- *WEBM (.webm)*

MP4 razvila je *Moving Pictures Expert Group*. To je noviji format na *webu* kojega preporučuje *YouTube*. Na *YouTube* je moguće postaviti razne formate koji se konvertiraju u *.mp4* ili *.flv*.

MPEG (Moving Pictures Expert Group) jedan je od najpopularnijih formata na *webu* i podržava ga većina *browsera*.

AVI (Audio Video Interleave) je razvio *Microsoft* te ga podržavaju sva računala koja kao operativni sustav koriste *Windows* i gotovo svi najpoznatiji *browseri*. U odnosu na *MPEG* ima puno lošiju kvalitetu i razinu komprimiranja.

WMW (Windows Media Video) je razvio *Microsoft* i podržavaju ga računala sa *Windows* operativnim sustavom, također. *Linux* i *Mac* operativni sustavi ga ne podržavaju. Vrlo je čest format na *webu*. Kvaliteta videa nije visoka, ali sadrži vrlo malo količinu podataka.

QUICKTIME je također vrlo čest video format na *webu*. Razvio ga je *Apple*. Računala koja koriste *Windows* operativni sustav zahtijevaju dodatnu instalaciju za pokretanje *QuickTime* videa.

Zahvaljujući *QuickTime Appleovoj* aplikaciji koja sadržava *browser plugin*, u *Mozilli Firefoxu* je moguća reprodukcija video i audio zapisa.

FLASH je razvila *Macromedia*. Isto kao i *QuickTime* zahtjeva instalaciju dodatne komponente za prikazivanje. *Flash player* je uglavnom unaprijed instaliram u poznatijim browserima.

OGV je video format koji je razvio *Xph.Org Foundation*. Može se reproducirati u najpoznatijim *browserima* i *playerima*, iako za neke zahtjeva *plugin*.

WEBM je video format za *web* namijenjen HTML5 inačici. Za reprodukciju videa takvog formata nisu potrebni dodatni *pluginovi*, već samo moderni *web browseri* koji podržavaju HTML5.

Tagovi pomoću kojih se neki video uvodi u HTML dokument su sljedeći, a kao i kod zvuka preporuča se `<video>` tag:

- `<iframe>` *tag*
- `<video>` *tag*
- `<object>` *tag*
- `<embed>` *tag*

Primjer uvođenja videa pomoću tih *tagova* prikazuje slika 18.

Uvođenje zvuka pomoću a taga



```
<html>
<head>
</head>
<body>
<a href="variations.mp4">Uvođenje videa pomoću a taga</a>
</br>
</br>
<audio src="variations.mp4" type="audio/mp4" controls></audio>
</br>
</br>
<object height="70" width="300" data="variations.mp4"></object>
</br>
</br>
<embed height="70" width="300" src="variations.mp4"></embed>
</body>
</html>
```

Slika 18. Primjer uvođenja videa različitim tagovima

2.7 Uvođenje animacije

Animacija je grafička kreacija sastavljena od više grafičkih elemenata koji se izmjenjuju određenim redoslijedom u određenom vremenskom periodu. Upravo dimenzija vremena predstavlja bitnu razliku između animacije i drugih grafičkih elemenata poput teksta ili, primjerice, fotografije. Samim time se ističe jedna od dvije osnovne stavke koje svaka animacija mora sadržavati, a to je vrijeme trajanja. Ime animacije, bez kojega ista ne bi mogla funkcionirati, jest druga stavka. Kada je riječ o uvođenju animacije u HTML dokument potrebno je spomenuti dvije komponente. Jedna komponenta je stil koji opisuje animaciju, a druga niz okvira (engl. *keyframes*). *Keyframeovi* određuju početno stanje animacije, međustanja te završno stanje animacije. Također je vrlo važno da se određena animacija postavi unutar onoga elementa kojega želimo animirati. Valja spomenuti što se sve animacijama može postići. Pa su tako, neke od opcija sljedeće:

- promjena boje
- promjena pozicije
- promjena oblika
- promjena prozirnosti

Te promjene mogu biti prezentirane na različite načine, pri čemu razlikujemo:

- *ease-in* animacije (na početku se odvijaju sporije nego kasnije kada postepeno ubrzavaju)
- *ease-out* animacije (na početku su brže nego pri kraju kada usporavaju)
- *ease-in-out* animacije (na početku i na kraju postepeno usporavaju)
- linearne animacije (odvijaju se linearno tj. jednako tijekom cijele duljine njihova trajanja)

Naravno, moguće je i odrediti što će se dogoditi s animacijama kada dođu u završnu, odnosno krajnju fazu. Stoga postoje:

- one koje su beskonačne (*infinite*)
- one koje se mogu ponavljati unatrag (*alternate*)

- one koje se mogu pokretati sa zakašnjenjem (to se zapisuje pomoću *delay* naredbe; npr. *animation-delay:2s*)
- one koje mogu zadržati završno stanje (*animation-fill-mode:forwards*)

Keyframeovi animacija sastoje se od ključnih trenutaka *@keyframes*, imena animacije i niza pravila (napisanih unutar vitičastih zagrada) koji ju definiraju. *Keyframeovi* koriste postotke kako bi odredili u koje vrijeme će se određeni dio animacije prikazati. No, kada postoji samo početno i završno stanje animacije, *keyframeovi* se mogu definirati i sa “*from*” i “*to*”. *Keyframeovi* animacije koja se nalazi na izrađenoj stranici prikazani su na slici 19.

```

@keyframes s3
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-webkit-keyframes s3
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-moz-keyframes s3
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-ms-keyframes s3
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}

@keyframes s7
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-webkit-keyframes s7
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-moz-keyframes s7
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-ms-keyframes s7
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}

@keyframes s6
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-webkit-keyframes s6
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-moz-keyframes s6
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-ms-keyframes s6
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}

```



```

@keyframes s4
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-webkit-keyframes s4
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-moz-keyframes s4
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-ms-keyframes s4
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}

@keyframes s2
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-webkit-keyframes s2
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-moz-keyframes s2
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-ms-keyframes s2
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}

@keyframes s8
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-webkit-keyframes s8
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-moz-keyframes s8
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-ms-keyframes s8
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}

@keyframes s9
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-webkit-keyframes s9
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-moz-keyframes s9
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-ms-keyframes s9
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}

@keyframes s1
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-webkit-keyframes s1
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-moz-keyframes s1
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-ms-keyframes s1
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}

@keyframes s5
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-webkit-keyframes s5
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-moz-keyframes s5
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}
@-ms-keyframes s5
{
0% {opacity:0;}
99% {opacity:0;}
100% {opacity:1;}
}

```

Slika 19. Keyframeovi animacije

Na kraju, u *body* dijelu dokumenta potrebno je pozvati one *div* elemente kojima je pridružena animacija kao što je prikazano na slici 20. Na taj će se način ista prikazivati u odgovarajućem *browseru*.

```
<div id="s1"> </div>  
<div id="s2"> </div>  
<div id="s3"> </div>  
<div id="s4"> </div>  
<div id="s5"> </div>  
<div id="s6"> </div>  
<div id="s7"> </div>  
<div id="s8"> </div>  
<div id="s9"> </div>
```

Slika 20. Pozvani elementi u *body* dijelu

3. EKSPERIMENTALNI DIO

Sadržaj prethodnih poglavlja u ovome će dijelu rada biti prikazan na primjeru izrađene *web* stranice. Cilj izrade *web* stranice jest prikaz i obrazloženje primjene pojedinih HTML elemenata te analiza ponašanja pojedinih formata za uvođenje slike, audia i videa u HTML dokument kroz ispitivanje njihove kompatibilnosti s najpoznatijim *browserima*.

Prikaz teksta, najjednostavnijeg HTML elementa, u različitim *browserima* ne predstavlja veliki problem. Kod uvođenja teksta na *web* stranicu, važno je paziti na jezik kojim je sadržaj iste napisan. Iako je na primjeru izrađene stranice tekst na engleskom jeziku, korištena su i neka vlastita imena na hrvatskome jeziku koja sadrže dijakritičke znakove. Da bi ti znakovi u pojedinim *browserima* na računalu bili vidljivi, potrebno je HTML dokument spremirati u odgovarajućem znakovnom skupu. Konkretno za hrvatski jezik, to je UTF-8. Time je problem riješen lokalno, no da bi ti isti znakovi bili vidljivi na svim računalima u svim *browserima* diljem svijeta potrebno je koristiti `<meta charset>` tag. Taj se tag piše u *head* dijelu i njime se kodiraju znakovi odabranog jezika za HTML dokument. Velika prednost HTML5 inačice jest i ta što je *defaultni charset* upravo UTF-8, što rješava mnoge probleme prikazivanja specifičnih znakova.

Kategoriji teksta pripadaju i liste. Na *web* stranici je lista upotrijebljena za izradu navigacijskog menija, a način na koji je ona uvedena u dokument prikazuje slika 21. Pritom je najprije definiran stil tog menija. *Classom* su definirana ona svojstva koja će se primijeniti na sve *id* elemente od kojih, onda, svaki element posjeduje još neka dodatna svojstva. Tako je *classom* definirana visina svake stavke menija, boja, *border-radius* i *list-style:none* čime su uklonjeni kružići i kvadratići koji se, po *defaultu*, nalaze ispred stavki neuređene liste. *Hoverom* je omogućeno da prelaskom miša preko navigacijskog menija svaka pojedina stavka liste mijenja boju u sivu. Zatim je za svaku od njih definirana točna pozicija i širina kroz *id* elemente (*#nm1*, *#nm2*, *#nm3*, *#nm4*, *#nm5*, *#nm6*, *#nm7* i *#nm8*). Na isti način je definiran i stil teksta koji je ispisan u meniju. Dakle, *classom* je određena boja, pozicija, font i njegova veličina što se odnosi na cijelu listu. Ovisno o širini *div* elemenata stavki unutar liste nije jednaka

mijenja se pozicija teksta zbog čega je ona definirana kroz *id* elemente *classe* (#t1, #t2, #t3, #t4, #t5, #t6, #t7 i #t8).

```
.navmenu {
position:absolute;
height:30px;
border-radius:10px;
background-color:#4E8975;
list-style:none;
}
.navmenu:hover {
background-color:gray;
}

#nm1 {
top:375px;
left:250px;
width:100px;
}

#nm2 {
top:406px;
left:170px;
width:180px;
}

#nm3 {
top:437px;
left:210px;
width:140px;
}

#nm4 {
top:468px;
left:150px;
width:200px;
}

#nm5 {
top:499px;
left:190px;
width:160px;
}

#nm6 {
top:530px;
left:230px;
width:120px;
}

#nm7 {
top:561px;
left:210px;
width:140px;
}

#nm8 {
top:592px;
left:190px;
width:160px;
}

.tekst {
position:absolute;
top:2px;
font-family:calibri;
font-size:20px;
color:white;
}

#t1 {left:40px;}
#t2 {left:75px;}
#t3 {left:70px;}
#t4 {left:72px;}
#t5 {left:80px;}
#t6 {left:60px;}
#t7 {left:76px;}
#t8 {left:71px;}
```

Slika 21. Navigacijski meni u CSS-u

U *body* dijelu dokumenta, kao što je prikazano na slici 22. ispisuje se neuređena lista sa već spomenutim *tagovima*. Pritom je u svaku stavku liste dodan i *link* na drugi HTML dokument te ispisan željeni tekst pozivanjem *id* elemenata.

```

<ul>
<div class="navmenu" id="nm1">
<li>
<a href="index.html">
  <div class="tekst" id="t1">
    HOME
  </div>
</a>
</li>
</div>
<div class="navmenu" id="nm2">
<li>
<a href="biography.html">
  <div class="tekst" id="t2">
    BIOGRAPHY
  </div>
</a>
</li>
</div>
<div class="navmenu" id="nm3">
<li>
<a href="works.html">
  <div class="tekst" id="t3">
    WORKS
  </div>
</a>
</li>
</div>
<div class="navmenu" id="nm4">
<li>
<a href="publications.html">
  <div class="tekst" id="t4">
    PUBLICATIONS
  </div>
</a>
</li>
</div>
<div class="navmenu" id="nm5">
<li>
<a href="gallery.html">
  <div class="tekst" id="t5">
    GALLERY
  </div>
</a>
</li>
</div>
<div class="navmenu" id="nm6">
<li>
<a href="video.html">
  <div class="tekst" id="t6">
    VIDEO
  </div>
</a>
</li>
</div>
<div class="navmenu" id="nm7">
<li>
<a href="audio.html">
  <div class="tekst" id="t7">
    AUDIO
  </div>
</a>
</li>
</div>
<div class="navmenu" id="nm8">
<li>
<a href="contact.html">
  <div class="tekst" id="t8">
    CONTACT
  </div>
</a>
</li>
</div>
</ul>

```

Slika 22. Navigacijski meni u *body* dijelu HTML dokumenta

Slika 23. konkretno, na primjeru izrađene *web* stranice, prikazuje kako uvedeni tekst (uključujući u to listu i linkove) u konačnici izgleda.



Slika 23. Tekst na izrađenoj web stranici

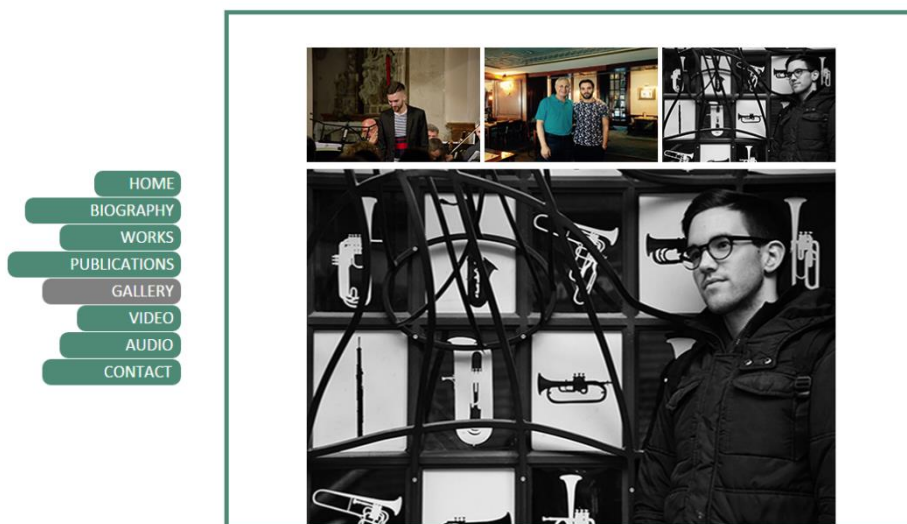
Kompatibilnost pojedinih slikovnih formata s najpoznatijim browserima

Sljedeći primjeri prikazuju rezultate testiranja različitih slikovnih formata u tri *browsera* (*Google Chrome*, *Mozilla Firefox* i *Internet Explorer*). Tablica 1. objedinjuje rezultate analize iz kojih se može primijetiti da se slikovni formati kao što su JPEG, PNG i GIF uredno prikazuju u sva tri *browsera*. Time je dokazano kako ovi formati uistinu jesu namijenjeni *webu*. S druge strane, WEBP format podržava jedino *Google Chrome*. Pritom se, sukladno prednostima i nedostacima i onome što se njima želi prikazati odabire odgovarajući slikovni format. Galeriju slika na *web* stranici prikazuje slika 24.

SLIKOVNI FORMAT	TAG	BROWSER		
		CHROME 44	FIREFOX 38	EXPLORER 11
JPEG	img	+	+	+
PNG		+	+	+
GIF		+	+	+
WEBP		+	-	-

Tablica 1. Rezultati ispitivanja kompatibilnosti pojedinih slikovnih formata s najpoznatijim browserima

antonio babiĆ | composer



Slika 24. Galerija slika na izrađenoj web stranici

Kompatibilnost pojedinih audio formata i tagova s najpoznatijim browserima

AUDIO FORMAT	TAG	BROWSER		
		CHROME 44	FIREFOX 38	EXPLORER 11
MP3	a	+	+	+
	audio	+	+	+
	object	+	+	+
	embed	+	+	+
WAV	a	+	+	+
	audio	+	+	-
	object	+	+	-
	embed	+	+	+
WMA	a	+/-	+/-	+/-
	audio	-	-	-
	object	-	-	+
	embed	-	-	+

Tablica 2. Rezultati ispitivanja kompatibilnosti pojedinih audio formata i tagova s najpoznatijim browserima

Tablica 2. prikazuje rezultate testiranja audio formata s različitim *tagovima* (<a> tag, <audio> tag, <object> tag i <embed> tag) u tri *browsera* (Google Chrome, Mozilla Firefox i Internet Explorer).

MP3 format se u *Google Chromeu* uredno prikazuje, neovisno u *tagu* koji se koristi. U *Mozilli Firefoxu* audio s <object> i <embed> *tagom* zahtjeva *QuickTime plugin*. *Pluginovi* su maleni programi koji mogu biti dodani u *browser* kako bi upravljali onim internet sadržajem za čiju obradu *browseri* nisu programirani. Tako se *QuickTimeom* mogu reproducirati medijski formati kao što su MPEG, WAV i MP3. Aktivacijom *QuickTime plugina* MP3 format će se u *Firefoxu* reproducirati bez obzira koji *tag* se koristi. U *Internet Exploreru* moguće je reproducirati sve *audio* snimke bez obzira koji *tag* se koristi.

Što se tiče *WAV* formata, rezultat ispitivanja u *Google Chromeu* i *Mozilli Firefoxu* (uz aktivaciju *QuickTime plugina* za *audio* s <object> i <embed>

tagom) identičan je kao i kod *MP3* formata, odnosno moguće je pokrenuti *audio*. U *Internet Exploreru* jedino funkcionira *audio* s `<a>` i `<embed>` *tagom* (uz aktivaciju *ActiveXa*).

WMA format se u *Google Chromeu* ne prikazuje ukoliko se koristi `<audio>`, `<object>` ili `<embed>` *tag*. *Chrome* podržava `<a>` *tag*, ali *audio* s tim *tagom* ne može se reproducirati na samoj stranici nego isključivo onda kada se isti preuzme na osobno računalo. U *Mozilli Firefoxu* također je problem u istim *tagovima*, ali u slučaju korištenja `<a>` *taga* *browser* otvara prozor u kojemu korisnik može izabrati želi li samo otvoriti dokument ili ga i pohraniti na računalo. Ako ga želi samo otvoriti, tada će morati i izabrati u kojem *audio playeru* će se taj zvuk reproducirati. U *Internet Exploreru* je isti slučaj s `<a>` *tagom* kao i u *Firefoxu*. *Audio* s `<object>` i `<embed>` *tagom* moguće je reproducirati uz aktivaciju *ActiveXa*, a onaj s `<audio>` *tagom* uopće nije moguće pokrenuti.

Slika 25. prikazuje na koji je način *audio* uveden na *web* stranicu.



Slika 25. *Audio* na izrađenoj *web* stranici

Kompatibilnost pojedinih video formata i tagova s najpoznatijim browserima

VIDEO FORMAT	TAG	BROWSER		
		CHROME 44	FIREFOX 38	EXPLORER 11
MP4	iframe	+	+	+
	video	+	+	+
	object	+	+	-
	embed	+	+	+
MPEG	iframe	+/-	+	+
	video	-	-	-
	object	-	+	-
	embed	-	+	+
AVI	iframe	+/-	+/-	+/-
	video	-	-	-
	object	-	-	-
	embed	-	-	-
WMV	iframe	+/-	+/-	+/-
	video	-	-	-
	object	-	-	-
	embed	-	-	-
MOV	iframe	+/-	+	+
	video	-	-	-
	object	-	+	-
	embed	-	+	+
FVL	iframe	+/-	+/-	+/-
	video	-	-	-
	object	-	-	-
	embed	-	-	-
WEBM	iframe	+	+	+/-
	video	+	+	-
	object	+	-	-
	embed	+	-	-
OGV	iframe	+	+	+/-
	video	+	+	-
	object	+	-	-
	embed	+	-	-

Tablica 3. Rezultati ispitivanja kompatibilnosti pojedinih video formata i tagova s najpoznatijim browserima

Tablica 3. prikazuje rezultate testiranja video formata s različitim *tagovima* (<a> tag, <audio> tag, <object> tag i <embed> tag) u tri *browsera* (*Google Chrome*, *Mozilla Firefox* i *Internet Explorer*).

MP4 format se u *Google Chromeu*, neovisno o tagu, uredno prikazuje. U *Mozilli Firefoxu* također je moguća reprodukcija svih videa bez obzira na tag (uz aktivaciju *QuickTime plugina* za videe s <object> i <embed> tagom). U *Internet Exploreru* se prikazuje video s <iframe> tagom, videi s <video> i <embed> tagom uz aktivaciju *ActiveX* programa, dok se onaj s <object> tagom uopće ne prikazuje.

MPEG format se u *Google Chromeu* ne prikazuje, ali se pri otvaranju HTML dokumenta automatski pokreće preuzimanje videa s <iframe> tagom sa stranice. *Firefox* jedino video s <video> tagom ne prikazuje. Ostali se prikazuju uz aktivaciju *QuickTime plugina*. U *Internet Exploreru* je moguća reprodukcija videa s <iframe> i <embed> tagom uz aktivaciju *ActiveX* programa, dok se ostali ne prikazuju.

AVI format se u *Google Chromeu* ne prikazuje, ali se pri otvaranju HTML dokumenta automatski pokreće preuzimanje videa s <iframe> tagom sa stranice baš kao i u slučaju s MPEG formatom. *Mozilla Firefox browser* će za video s <iframe> tagom otvoriti prozor u kojemu korisnik može izabrati želi li samo otvoriti dokument ili ga i pohraniti na računalo. Ako ga želi samo otvoriti, tada će morati i izabrati u kojem *video playeru* će se taj zvuk reproducirati. Za video s <video> tagom, *Firefox* ne pronalazi podršku. Za videe s <object> i <embed> tagom zahtjeva *plugin*. Kod *Internet Explorera* je isti slučaj kao i u *Firefoxu* kada je riječ o <iframe> tagu, dok se ostali videi uopće ne prikazuju.

Uvođenje WMV formata u HTML dokument različitim *tagovima* identično je uvođenju AVI formata.

MOV format se u *Google Chromeu* ne prikazuje ako se koristi <object> i <embed> tag. Pri otvaranju HTML dokumenta automatski pokreće preuzimanje videa s <iframe> tagom sa stranice, a za video s <video> tagom dostupan je

samo zvuk. *Mozilla Firefox* uz aktivaciju *QuickTime* *plugina* ne prikazuje jedino video s `<video>` *tagom*. U *Internet Exploreru* funkcioniraju videi s `<iframe>` i `<embed>` *tagom* uz aktivaciju *ActiveXa*.

Uvođenje FLV formata u HTML dokument različitim *tagovima* identično je uvođenju AVI i WMV formata.

WEBM format se u *GoogleChromeu* uredno prikazuje neovisno o *tagu* koji se koristi. U *Mozilli Firefoxu* videi s `<iframe>` i `<video>` *tagom* se prikazuju, video s `<embed>` *tagom* zahtjeva aktivaciju *plugina*, a onaj s `<object>` *tagom* se uopće ne prikazuje. U *Internet Exploreru* se jedino video s `<iframe>` *tagom* može pohraniti na računalo ili reproducirati u nekom od *playera* na računalo.

Uvođenje OGV formata u HTML dokument različitim *tagovima* identično je uvođenju WEBM formata.

Postoji još jedan, nešto jednostavniji, način uvođenja videa u HTML dokument koji je prikazan na slici 26., a to je pomoću *linka*. To je vrlo vjerojatno najlakši način na koji se neki video može uvesti na *web* stranicu budući da različite verzije različitih *browsera* podržavaju različite formate. Vjerojatnost prikazivanja videa u različitim *browserima* je, u tom slučaju, veća nego onda kada se video uvodi kao u prethodnim primjerima. U tom slučaju, testiranjem kompatibilnosti pojedinih *tagova* (`<iframe>`, `<object>` i `<embed>` *taga*) i poznatih *browsera* (*Google Chrome*, *Mozilla Firefox* i *Internet Explorer*) dobivaju se sljedeći rezultati: u *Chromeu* i *Exploreru* je moguće pokrenuti sve video formate neovisno o *tagu*, a u *Firefoxu* se ne prikazuje jedino video s `<embed>` *tagom*.

```

<html>
<head>
</head>

<body>

<iframe width="400" height="300" frameborder="0"
src="https://www.youtube.com/embed/wwf5BY6-w1E" ></iframe>
</br>
</br>

<object data="https://www.youtube.com/embed/wwf5BY6-w1E" width="400"
height="300"></object>
</br>
</br>

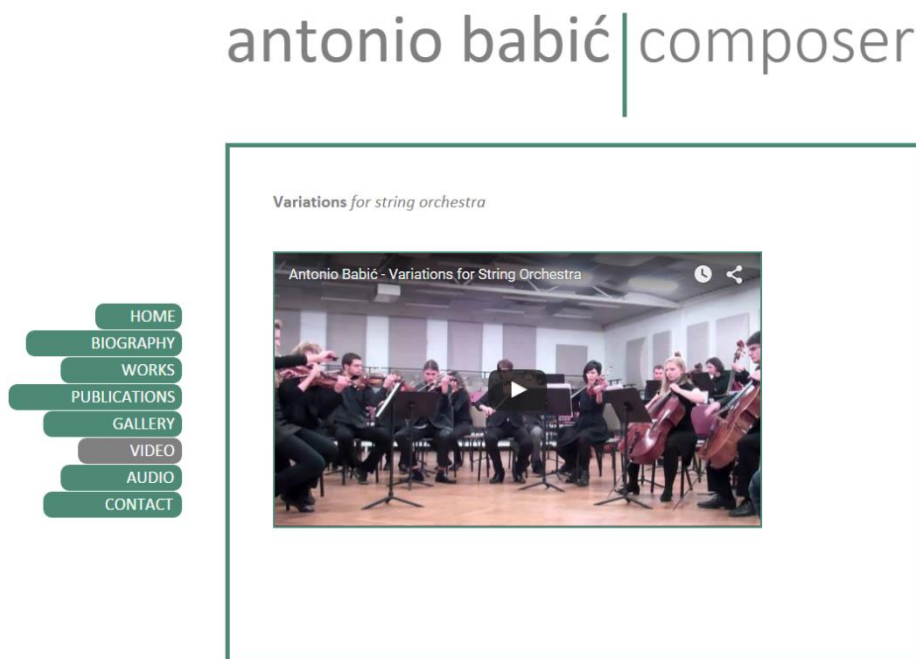
<embed src="https://www.youtube.com/embed/wwf5BY6-w1E" width="400"
height="300"></embed>

</body>
</html>

```

Slika 26. Primjer uvođenja videa različitim tagovima pomoću linka

Slika 27. prikazuje na koji je način video uveden na web stranicu.



Slika 27. Video na izrađenoj web stranici

Na kraju preostaje analiza uvođenja animacije. Na izrađenoj stranici riječ je o postepenom prikazivanju pojedinih dijelova fotografije što je postignuto promjenom opaciteta. Tako, svaki *div* element mijenja svoj opacitet iz 0 u 1, odnosno iz potpuno prozirnog u potpuno neprozirno stanje. Animacija ima početno stanje (0%), međukorak (99%) i završno stanje (100%). Međukorakom se postiže oštro otvaranje fotografija.

Konkretno, na primjeru izrađene *web* stranice mogu se vidjeti neki od spomenutih obilježja animacija. Na slici 28. nalazi se stil kojim je opisana animacija. Fotografija čija je izvorna veličina 800x600px podijeljena je na devet jednakih dijelova. Svakom dijelu dodijeljene su osnovne informacije, kao što su pozicija (pomoću naredbe *top* i *left*) te širina i visina svakog pojedinog dijela. Kako ove fotografije, tako i svi ostali elementi na stranici pozicionirani su apsolutno što znači da je točka ishodišta u gornjem lijevom kutu. Svemu navedenom, dodana je i informacija o animaciji iz koje se može iščitati njezino ime (*s1*, *s2*, *s3*, *s4*, *s5*, *s6*, *s7*, *s8*, *s9*), vrijeme trajanja (2s), način na koji se odvija (linearna je), zakašnjenje (svaki od devet dijelova kasni za prethodnim 0.5s) te da zadržava svoje završno stanje (*forwards*). Animacija je u ovome slučaju definirana za *Google Chrome*, *Mozillu Firefox* i *Internet Explorer*. To se postiže pridruživanjem odgovarajućeg prefiksa animaciji. Tako se onoj koja će funkcionirati u *Google Chromeu* dodaje prefiks *-webkit-*, onoj u *Mozilli Firefoxu* *-moz-*, dok se onoj u *Internet Exploreru* dodaje *-ms-*. Dakle, kako bi se animacije, tranzicije ili transformacije na ispravan način prikazivale u odabranim preglednicima potrebno je koristiti spomenute prefikse. No nažalost, to ne mora nužno biti garancija ispravnog prikaza upravo zbog različite podrške koju pružaju pojedini *web* preglednici.

```
#s1 {
position:absolute;
width:267px;
height:200px;
top:190px;
left:400px;
background:url(1.jpg);
opacity:0;
animation:s1 2s linear 3.5s forwards;
-webkit-animation:s1 2s linear 3.5s forwards;
-moz-animation:s1 2s linear 3.5s forwards;
-ms-animation:s1 2s linear 3.5s forwards;
}
```

```

#s2 {
position:absolute;
width:267px;
height:200px;
top:190px;
left:668px;
background:url(2.jpg);
opacity:0;
animation:s2 2s linear 2s forwards;
-webkit-animation:s2 2s linear 2s forwards;
-moz-animation:s2 2s linear 2s forwards;
-ms-animation:s2 2s linear 2s forwards;
}

#s3 {
position:absolute;
width:267px;
height:200px;
top:190px;
left:936px;
background:url(3.jpg);
opacity:0;
animation:s3 2s linear forwards;
-webkit-animation:s3 2s linear forwards;
-moz-animation:s3 2s linear forwards;
-ms-animation:s3 2s linear forwards;
}

#s4 {
position:absolute;
width:267px;
height:200px;
top:391px;
left:400px;
background:url(4.jpg);
opacity:0;
animation:s4 2s linear 1.5s forwards;
-webkit-animation:s4 2s linear 1.5s forwards;
-moz-animation:s4 2s linear 1.5s forwards;
-ms-animation:s4 2s linear 1.5s forwards;
}

#s5 {
position:absolute;
width:267px;
height:200px;
top:391px;
left:668px;
background:url(5.jpg);
opacity:0;
animation:s5 2s linear 4s forwards;
-webkit-animation:s5 2s linear 4s forwards;
-moz-animation:s5 2s linear 4s forwards;
-ms-animation:s5 2s linear 4s forwards;
}

```

```

#s6 {
position:absolute;
width:267px;
height:200px;
top:391px;
left:936px;
background:url(6.jpg);
opacity:0;
animation:s6 2s linear 1s forwards;
-webkit-animation:s5 2s linear 1s forwards;
-moz-animation:s5 2s linear 1s forwards;
-ms-animation:s5 2s linear 1s forwards;
}

#s7 {
position:absolute;
width:267px;
height:200px;
top:592px;
left:400px;
background:url(7.jpg);
opacity:0;
animation:s7 2s linear 0.5s forwards;
-webkit-animation:s7 2s linear 0.5s forwards;
-moz-animation:s7 2s linear 0.5s forwards;
-ms-animation:s7 2s linear 0.5s forwards;
}

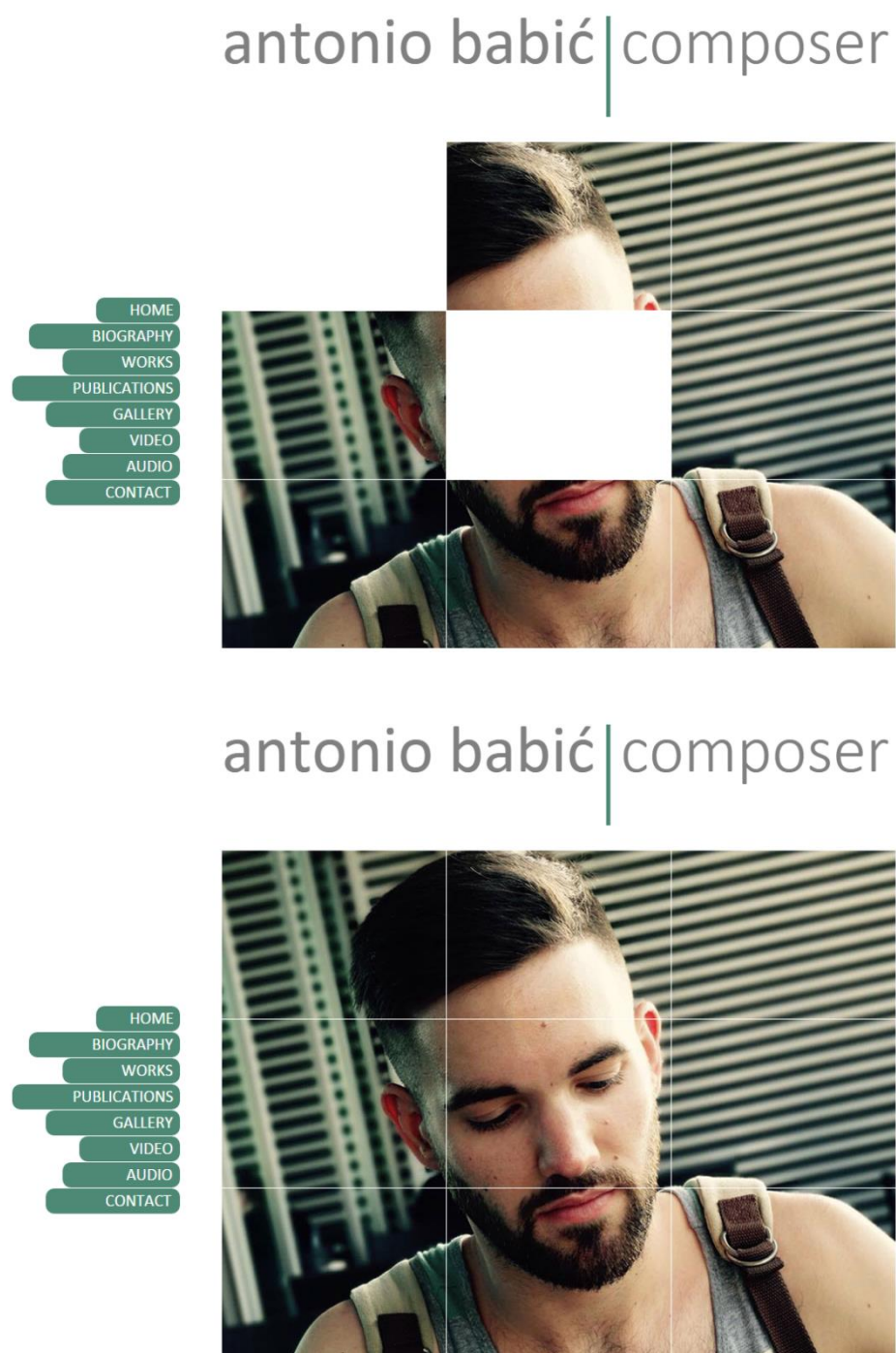
#s8 {
position:absolute;
width:267px;
height:200px;
top:592px;
left:668px;
background:url(8.jpg);
opacity:0;
animation:s8 2s linear 2.5s forwards;
-webkit-animation:s8 2s linear 2.5s forwards;
-moz-animation:s8 2s linear 2.5s forwards;
-ms-animation:s8 2s linear 2.5s forwards;
}

#s9 {
position:absolute;
width:267px;
height:200px;
top:592px;
left:936px;
background:url(9.jpg);
opacity:0;
animation:s9 2s linear 3s forwards;
-webkit-animation:s9 2s linear 3s forwards;
-moz-animation:s9 2s linear 3s forwards;
-ms-animation:s9 2s linear 3s forwards;
}

```

Slika 28. Stil animacije

Slika 29. prikazuje izgled animacije na izrađenoj web stranici.



Slika 29. Animacija na izrađenoj web stranici

4. ZAKLJUČAK

Iz teorijskog i eksperimentalnog dijela može se zaključiti da je moguće izraditi funkcionalnu *web* stranicu pomoću HTML-a i CSS-a. Funkcionalna *web* stranica omogućuje korisniku jednostavno i intuitivno korištenje sadržaja kojim ista raspolaže. Kroz ovaj rad to je postignuto uvođenjem različitih multimedijских elemenata i mogućnostima koje oni pružaju. Njihova zastupljenost na *web* stranici ovisi prvenstveno o informacijama koje ona posjeduje. Pri tome se svakako se može izdvojiti tekst kao neizbježna stavka svake *web* stranice dok se ostali multimedijски elementi mogu, a i ne moraju upotrebljavati. S jedne strane, ti elementi znatno pridonose prijenosu neke izložene informacije. Osim što će u njemu izazvati veću zainteresiranost, korisnik će ju neusporedivo bolje zapamtiti nego kada se ona interpretira samo u tekstualnom obliku. S druge strane, za takve elemente na raspolaganju je mnoštvo mogućnosti, poput upotrebe različitih *tagova* i formata zapisa, koje nerijetko mogu izazvati problem. Do toga dolazi pri otvaranju HTML dokumenta u nekom od *browsera* zbog podrške koju oni ne pružaju određenim formatima tih elemenata. Stoga je potrebno procijeniti kojim će se elementom željena informacija najbolje prenijeti i nakon toga, sukladno podršci *browsera*, izabrati najprikladniji format i *tag*.

5. LITERATURA

1. *** <http://www.w3.org/standards/webdesign/>, 10.08.2015.
2. *** <http://www.bug.hr/pretraga/?s=bug&q=html>, 11.08.2015.
3. *** <http://www.w3schools.com/css/default.asp>, 10.08.2015.
4. *** <http://www.bug.hr/pretraga/?s=bug&q=css>, 11.08.2015.
5. Duckett J. (2011). *HTML & CSS Design and build websites*, John Wiley & Sons, Indianapolis
6. E. Robson, E. Freeman (2012). *Head First HTML and CSS*, O'Reilly Media, Sebastopol
7. *** <http://www.w3schools.com/html/default.asp>, 10.08.2015.
8. C. Musciano, B. Kennedy (2006). *HTML & XHTML The Definitive Guide, Sixth Edition*, O'Reilly Media, Sebastopol
9. R. Huddelston (2008). *HTML, XHTML & CSS: Your visual blueprint™ for designing effective Web pages*, Wiley Publishing, Indianapolis
10. J. Meloni, M. Morrison (2009). *Sams Teach Yourself HTML & CSS in 24 Hours, Eighth Edition*, Sams Publishing, Indianapolis