

# Elektroničke knjige sa namjenom za čitanje u web preglednicima

---

**Penava, Neda**

**Undergraduate thesis / Završni rad**

**2021**

*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:216:154523>

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

*Download date / Datum preuzimanja:* **2024-07-07**



*Repository / Repozitorij:*

[Faculty of Graphic Arts Repository](#)



SVEUČILIŠTE U ZAGREBU

GRAFIČKI FAKULTET

# ZAVRŠNI RAD

ELEKTRONIČKE KNJIGE S NAMJENOM ČITANJA  
U WEB PREGLEDNICIMA

Neda Penava



Sveučilište u Zagrebu  
Grafički fakultet

Smjer: Tehničko-tehnološki

# ZAVRŠNI RAD

ELEKTRONIČKE KNJIGE S NAMJENOM ČITANJA  
U WEB PREGLEDNICIMA

Mentorica:  
dr.sc. Maja Rudolf

Studentica:  
Neda Penava

Zagreb, 2021.



## **SAŽETAK**

Elektroničke knjige su namjenjene čitanju na različitim tehnologijama poput pametnih telefona, e-čitača te računala.

Ovaj završni rad se bazira na izradi elektroničke knjige čija je namjena primarno čitanje iz web preglednika, a ne namjenskih čitača poput Kindla i dr.

Za izradu e-knjige smo koristili postojeće web tehnologije kao što su HTML, označni jezik, te CSS, stilski jezik.

Tim tehnologijama smo ukomponirali glavne sastavnice e-knjige: naslovnicu, sadržaj, glavni tekst i multimediju zajedno sa samim izgledom e-knjige.

**Ključne riječi:** e-knjiga, HTML, multimedija, CSS

## SADRŽAJ

1. Uvod.....	1
2. Teorijski dio .....	2
2.1. E-knjiga.....	2
2.1.1. Prednosti i mane e-knjige.....	4
2.2. HTML .....	5
2.2.1. Struktura HTML-a .....	5
2.2.2. Oznake za multimediju .....	8
2.3. CSS.....	9
3. Izrada e-knjige.....	10
4. Zaključak.....	22
5. Literatura.....	23

## 1. Uvod

U današnje vrijeme tehnologija jako brzo raste i brzo se razvija. Putem interneta su nam dostupne sve informacije koje nas zanimaju.

Najbitnija te najviše korištena strategija prijenosa informacija na web-u je HTTP (engl. *HyperText Transfer Protocol*).

Elektroničke knjige su u zadnjih nekoliko godina postale popularne te su jednim dijelom zamijenile klasične tiskane knjige jer su ljudima dostupne u svakom trenutku na njihovim elektroničkim uređajima.

Pod pojmom elektroničke knjige spadaju online časopisi, multimedijske knjige te digitalne knjige čija je poanta da ih se sluša kao audioknjigu.

Tehnologija e-knjige se brzo razvija. Mnogi izdavači prilikom dizajna tiskane knjige odmah rade i pripremu za digitalnu verziju koja kasnije također ide u prodaju.

Postoji mogućnost izrade e-knjige skeniranjem putem Robotic Scannersa.

E-knjigu za čitanje u browserima radimo da bi bile dostupnije na svim uređajima, jer svaki elektronički uređaj ima pristup internetu i browser. Time smo dobili dvije prednosti, a to su da knjiga ne mora biti fizički pohranjena na uređaju, već je dostupna kao web stranica i drugo da nije kreirana u posebnom formatu kao neke e-knjige koje su prilagođene samo za određene čitače. Takva knjiga se radi u jeziku za prikaz i opis sadržaja – HTML-u i vizualno uređuje u CSS-u. Omogućavaju jasan pregled sadržaja, otvaranje i čitanje knjige na raznim uređajima te uredan prikaz multimedije zajedno sa tekstom, bez obzira na kojem je uređaju otvorena.

Kod PDF formata, datoteka nije jasno čitljiva korisniku kao elektronička knjiga te je potreban program za reprodukciju sadržaja. Navedeni formati ne pokrivaju u cijelosti zahtjeve koje pred njih stavlja elektronička knjiga, stoga se paralelno s razvojem pojedinih formata razvija i format zapisa koji bi trebao postati standard zapisa elektroničke knjige. [1]

## 2. Teorijski dio

### 2.1. E-knjiga

E-knjiga je elektronička datoteka koja je namjenjena prikazu na ekranu te se sastoji od teksta i multimedije, slika i videa.

Jedna od opcija stvaranja e-knjige je skeniranje tiskane knjige. Tim postupkom štedimo na vremenu te zahtjeva manje truda od tipkanja cijelog teksta knjige.

Skenirane slike se najčešće spremaju u formatu PDF-a (*Portable Document Format*), no dosta je popularan i format TIFF (*Tagged Image File Format*).

Još jedan od formata koji se često upotrebljava je EPUB *engl. Electronic Publication*, komprimirani (sažeti) skup XHTML-a, CSS-a i ostalih mogućih datoteka koje se koriste i kod izrade web stranica. To je format koji je poznat po tome što se prilagođava različitim dimenzijama ekrana. Najnovija verzija EPUB3 omogućava ubacivanje multimedije u sadržaj te je napredniji u prikazu teksta i grafike. [2]

S druge strane, e -knjige nastaju u formatu koji mijenja oblik ovisno o uređaju na kojem ga se čita. I dalje će postojati i imati poglavlja i odlomke, ali prijelomi redaka nisu prisilni: uvijek će savršeno ispuniti zaslon. Tehnologija koja može izmijeniti tekst sa učitane slike skenera tzv. optičko prepoznavanje znakova (*eng. Optical Character Recognition, OCR*).

Pretvara tiskani dokument ili sliku u elektroničku verziju s tekстом koja se može uređivati te pretraživati po ključnoj riječi. Poanta je stvaranje digitalne verzije tiskane knjige ili određenog teksta.

Sliku obrađuje lociranjem znakova poput slova, brojeva ili simbola koje pretvara u ASCII ili Unicode kod. [3] Većina modernih OCR programa skenira svaku stranicu, prepozna tekst na njoj, a potom skenira sljedeću. [4]

Pomoću ovih skenera se vrlo lako pretvara tiskana verzija knjige u elektroničku sa mogućnošću dodatne dorade teksta i slike.

Jedan od najpoznatijih projekata koji skeniraju knjige u većim količinama je „Project Gutenberg“. Cilj projekta je digitalizacija i pristup poznatim knjigama i tekstovima na računalu. [5]

Knjige koje se skeniraju moraju biti dostupne korisnicima na web preglednicima kako bi ih mogli iskoristiti kao „univerzalnu knjižnicu“.



Jedna od većih prednosti digitalne knjižnice je to što joj može pristupiti više korisnika u isto vrijeme. Svakom korisniku je ista kvaliteta materijala. [6]

Postoje tri glavna načina skeniranja, a to su: eksternaliziranje, skeniranje u „kući“ sa komercijalnim skenerima te skeniranje u „kući“ sa robotiziranim skenerima.

Kod eksternaliziranja, knjige se skeniraju na jeftinijim destinacijama poput Indije ili Kine.

Glavna poanta Projekta Gutenberg je dostupnost knjiga na webu zbog očuvanja jer papir na tiskanim knjigama propada.

Prodaja e-knjiga se prvi put pojavila krajem 90-ih godina sa tvrtkom Peanut Press koja je nudila knjige sa digitalnim ručnim uređajima koji su prethodili pametnim telefonima.



*Slika 1. Prikazuje korištenje osobnog digitalnog pomoćnika kojim se mogu uređivati podaci*

*Izvor:*

[https://www.google.com/search?q=pda&rlz=1C1CHBF\\_hrHR875HR875&sxsrf=AOaemvJ6G2uH4M0HqKjMeFXI89JbB2wV6w:1630753374880&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjSsoDdleXyAhVxhPOHHef3CTEO\\_AUoAXoECAEQAw&biw=824&bih=712#imgrc=AZLzg\\_8nhOCIIM](https://www.google.com/search?q=pda&rlz=1C1CHBF_hrHR875HR875&sxsrf=AOaemvJ6G2uH4M0HqKjMeFXI89JbB2wV6w:1630753374880&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjSsoDdleXyAhVxhPOHHef3CTEO_AUoAXoECAEQAw&biw=824&bih=712#imgrc=AZLzg_8nhOCIIM)

Prodaja e-knjiga je naglo porasla kada je 2006. godine Sony Corporation izradio softver za čitanje e-pošte, te kada je Amazon 2007. izradio e-čitač Kindle.[7]

Elektroničke knjige imaju opciju čitanja online i offline putem, a u današnje vrijeme se najčešće prodaju u SAD-u te Velikoj Britaniji, no i na ostalim tržištima poput Kine, Japana ili Njemačke. [8]

### 2.1.1. Prednosti i mane e-knjige

E-knjiga ima dosta prednosti u odnosu na tiskanu, no isto tako postoje i nedostaci koji čine tiskanu knjigu jednostavnijom.

Prednosti:

- sadržaj knjige se u bilo kojem trenutku može pretraživati
- mogu se čitati u mraku ili pri manje svjetla
- brzo ih možemo pretvoriti u tiskani materijal
- cijena distribuiranja nije visoka
- smanjena potrošnja papira i tinte
- na nju ne djeluje prolaznost vremena
- može sadržavati multimediju poput videa te audiozapisa
- moguća vizualna prilagodba
- ekološki prihvatljivija
- lako prenosiva

Mane:

- može biti neprihvaćena od strane hardwera i softwera
- može doći do greške u programu te spriječiti ispis
- ovisna o napajanju strujom (može nestati baterije)
- prosječna cijena je nekim korisnicima preskupa
- predugo gledanje u ekran može imati loše posljedice – problemi s vidom te glavobolje
- potreban izvor struje za računala

## 2.2.HTML

HTML je skraćenica za „*Hyper Text Markup Language*“. To je prezentacijski jezik koji služi izradi web stranica.

Prva verzija jezika je napravljena 1993. godine, no tada još nije bilo moguće dodati slike u dokument.

Sljedeća verzija je nastala 1995., a ona donosi mogućnost stvaranja tablice.

HTML4 je nastao u prosincu 1997., a 1999. nova verzija koja se zvala HTML4.01.

Posljednja verzija je HTML5 koja je modificirana; uvode sučelje označavanja te API za složenije aplikacije. API je sučelje koje služi programiranju aplikacija. [9]

### 2.2.1. Struktura HTML-a

Svaki HTML se sastoji od HTML tagova koji mogu imati atribute koji opisuju značenje tog elementa.

Prije početka kreiranja samog HTML dokumenta je poželjno upisati oznaku

`<!DOCTYPE>` kojom se formulira točan standard koji se koristi za izradu HTML dokumenta.

Sljedeći korak je otvoriti `<html>` tag unutar kojeg se nalaze svi ostali elementi poput `<head>`, u kojem se definira naslov, ali i sam stil i izgled datoteke, i `<body>`, u kojem se nalazi sav sadržaj koji želimo imati u dokumentu. Sve elemente na kraju obavezno moramo zatvoriti inače datoteka nije valjana.

Tagovi se zatvaraju koristeći kosu zagradu (engl. *slash*) ispred imena unutar početne i završne oznake kao npr. zatvoreni `<head>` tag: `</head>`.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 |   <title>Document</title>
5 </head>
6 <body>
7 |   <h1>Naslov</h1>
8 |   <p>Tekst u obliku paragrafa</p>
9 </body>
10 </html>
```

Slika 2. Primjer osnovnog HTML dokumenta

Izvor: autor

Unutar HTML dokumenta možemo unositi i komentare kako bi se lakše mogli snaći, a oni neće biti vidljivi na samoj stranici.

Izgled komentara: `< ! - - komentar - - >`.

Može se reći da je svaka HTML datoteka „obiteljsko stablo“.

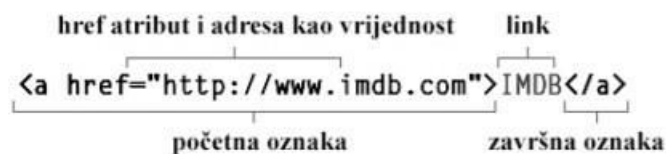
Glavne sastavnice za izradu teksta su naslov i paragraf. Naslovi imaju 6 razina, od `<h1>` do `<h6>`, a paragraf se označava oznakom `<p>`.

Unutar paragrafa možemo ubaciti razne elemente kojima uređujemo tekst poput `<blockquote>` u kojem uređujemo duže citate dok za kraće koristimo `<q>`.

U HTML-u možemo dodati liste, a postoje dvije vrste: uređene i neuređene.

Uređene ćemo koristiti kada imamo već redoslijed kojim lista treba ići, `<ol>`, a neuređene kada nemamo redoslijed, `<ul>`. Svaki dio liste se definira `<li>` tagom.

Pomoću elementa `<a>` ugrađujemo linkove u HTML datoteku, a sve što se nalazi unutar oznaka označava ono što će na stranici djelovati kao sam link, bila to slika ili tekst, odvesti će na adresu navedenog linka jednim klikom.



Slika 3. Sastavnice linka u HTML-u

Izvor: [https://eprints.grf.unizg.hr/2293/1/Z611\\_Prelec\\_Andrija.pdf](https://eprints.grf.unizg.hr/2293/1/Z611_Prelec_Andrija.pdf)

Linku možemo omogućiti otvaranje u novom prozoru korištenjem atributa `target` kojemu vrijednost definiramo kao `_blank`.

Linku također možemo dodati atribut `id` preko kojeg nas može navesti na dio naše stranice, a često se koristi i atribut `class` (klasa, kategorija).

Neki sadržaji internetskih stranica poput sportskih rezultata ili rasporeda tramvajskih stanica se prikazuju u tablicama. Tablice se sastoje od stupaca i redova. U HTML-u

tablice označavamo `<table>` elementom unutar kojeg se može sastojati bezbroj `<tr>` i `<td>` elemenata.

Redovi se označavaju `<tr>` elementom, a `<td>` stupci, dok se `<th>` element koristi za naslov retka i stupca.

Elementom `<form>` prikupljamo unose korisnika, a `<input>` služi za unos podataka.

`<iframe>` element koristimo ukoliko unutar naše stranice želimo prikazati drugu stranicu. Jedan od najboljih primjera gdje se to koristi je prikaz Google Maps karte s određenom lokacijom unutar stranice koju trenutno posjećujemo. Za točan prikaz `<iframe>` elementa trebamo nekoliko atributa, a to su `src` za specificiranje željene URL adrese, te `height` i `width` (visina i širina). [9]

### 2.2.2. Oznake za multimediju

Za umetanje slika u dokument koristimo oznaku `<img>` koja u sebi mora imati atribut `src` koji nam objašnjava gdje se slika nalazi. Atribut `alt` nam daje detaljniji opis slike ukoliko se slika ne može prikazati ili ukoliko se koristi čitač za slijepe osobe.

Možemo dodati i naslov slike atributom `title` koji nam se prikazuje prelaskom miša preko slike u obliku oblačića.

Izgled slike možemo manipulirati definiranjem visine i širine, odnosno `width` i `height` atributima, a najčešća vrijednost im je u pikselima.

Za vektorsku grafiku upotrebljavamo `<svg>` element.

Oznakom `<video>` dodajemo videozapis u dokument, a unutar njega se nalazi `<source>` koji nam objašnjava izvor videa te format.

Atribut `src` nas točno navodi gdje je video smješten, `poster` stvara sliku na videu u vremenu dok ga korisnik ne skine na računalo ili dok se video još nije pokrenio.

`Autoplay` učitava video odmah pri ulasku na odabranu stranicu, `loop` ga iznova učitava nakon završetka te `preload` koji navodi na korake pri učitavanju web dokumenta.

Glazbu možemo dodati putem `<embed>` ili `<audio>` oznake.

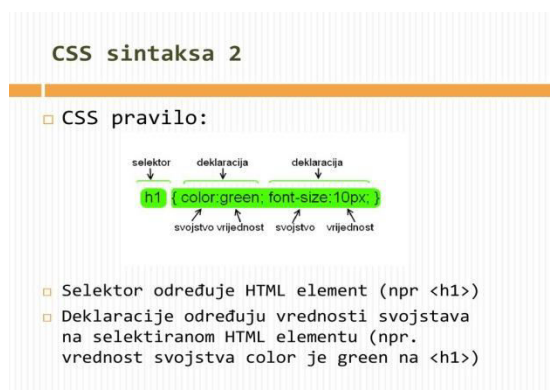
Element `<audio>` je uglavnom jednak kao `<video>`, ali nema attribute poput `poster`, `width` i `height`. [9]

## 2.3. CSS

Proizvod bez dizajna i pripadajućeg izgleda, na današnjem tržištu, nema neku vrijednost. Jako je bitno prezentirati proizvod, bilo to web stranica, aplikacija ili e-knjiga, na najbolji mogući način, odnosno na način da je primamljiv korisniku. U slučaju e-knjige, korisniku bi, osim pristupačnosti, bilo primamljivo da je taj sadržaj praktičan odnosno da ima opciju lakog prebacivanja stranica i mogućnost odlaska na željenu stranicu te mogućnost pretraživanja željene riječi ili pojma. Uz sve te prednosti koje bi imala e-knjiga, ona ne bi bila dobar proizvod ukoliko njen dizajn i izgled ne bi donekle privukao korisnika. Nije dobro koristiti jarke boje, nečitljiv ili premali font, kao ni prevelik. Odnos boje pozadine i boje teksta također treba uzeti u obzir. Vizualni dio e-knjige je gotovo jednako bitan kao i sam sadržaj.

Izgled proizvoda, ako koristimo jezik HTML ili JavaScript, određuje CSS engl. *cascading style sheets*. To je jezik stilske tablice koji se koristi za opisivanje prezentacije dokumenta napisanog označnim jezikom koji je najčešće HTML. CSS opisuje kako se elementi trebaju prikazati na ekranu, papiru, u govoru ili na drugim medijima.

Prije CSS -a, gotovo svi prezentacijski atributi HTML dokumenata bili su sadržani u HTML oznakama. Sve boje fontova, stilovi pozadine, poravnanja elemenata, obrubi i veličine morali su biti eksplicitno opisani u HTML -u. CSS omogućuje autorima premještanje većeg dijelatih podataka u drugu datoteku, stilsku tablicu, što rezultira znatno jednostavnijim HTML -om. [10]



Slika 4. Sintaksta CSS-a

Izvor: <https://slideplayer.com/slide/14007015/>

### 3. Izrada e-knjige

U istraživačkom dijelu završnog opisuje se postupak izrade e-knjige namijenjene čitanju iz preglednika. Odabrani sadržaj knjige je skripta sažetih tekstova iz predavanja kolegija Industrija i Okoliš, namijenjena studentima kao ispitna literatura. Upravo zato je kreirana kao web stranica koja se prikazuje iz web browsera kako bi bila dostupnija studentima tijekom učenja. Tekstualni sadržaj je obogaćen dodatnim multimedijским elementima za bolje razumijevanje sadržaja.

Nadalje, u radu su opisani svi segmenti korištenih tehnologija, HTML-a i CSS-a.

Za svaki dio postoje određene oznake koje se koriste za stvaranje sadržaja.

Kod izrade same e-knjige je potrebno imati dobar dizajn koji privlači korisnika te je ugodan za duže čitanje.

Posebno je rađen HTML dio koji predstavlja sadržaj knjige te je u njega linkom ugrađen CSS kojim se sadržaj stilizira.

Prilikom izrade naslovne stranice nije bilo potrebno koristiti velik opus oznaka u HTML kodu, već je bitnije se fokusirati na CSS i dobro komponiranje boja zajedno sa fontom te pozicijom samog teksta.

```
110
111 /*NASLOVNIA*/
112 #naslovnica {width:100%;
113 background: linear-gradient(
114   #234F1E 0%,
115   #234F1E 90%,
116   rgb(250, 192, 129) 98%,
117   rgb(250, 192, 129) 100%
118 );
119 }
120 #pocetna{
121 width: 70%;
122 margin: auto;
123 font-size: 7em;
124 /*word-spacing: -0.125em; */
125 line-height: 1;
126 padding-bottom: 1em;
127 padding-top: 0.5em;
128 font-family: 'Playfair_BoldItalic', serif;
129 color: #wheat;
130 text-shadow: 2px 2px 5px #red;
131 font-style: italic;
132 }
```

Slika 5. prikazuje izgled HTML koda za izradu naslovnice

Izvor: autor

Kod izrade sadržaja, bitan dio je bio postaviti link na naslov koji nas samo jednim klikom odvede na traženi paragraf.



Da bi link funkcionirao, mora se dodati atribut `id` naslovu paragrafa te se njegovo ime koristi u oznaci `href` u linku te ispred njega moramo staviti znak `#`.

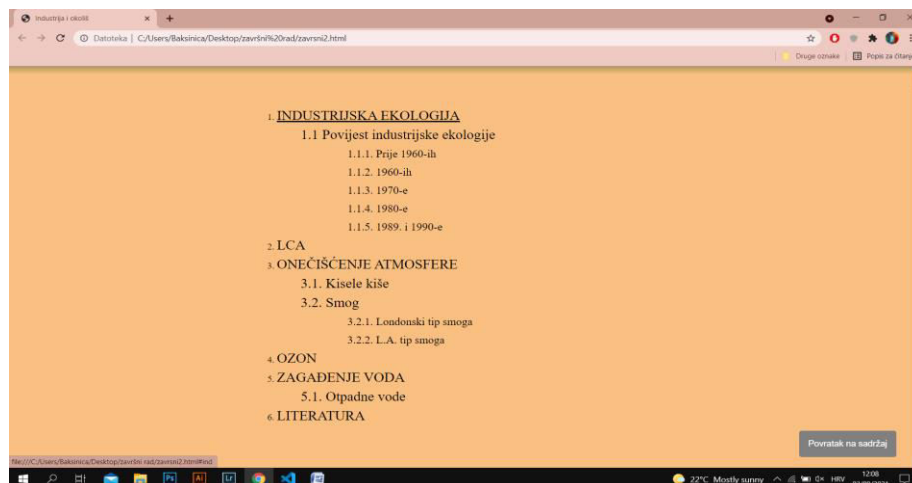
Sadržaj je rađen u obliku liste koja ima tri razine: određena, neodređena i stavka unutar popisa. Određenom stavkom definirano je glavno poglavlje oznakom `<ol>` kojom se postiže automatska numeracija komponenti koje imaju određenu poziciju točno jedna ispod druge. Unutar te oznake postavljena je oznaka `<ul>` koja opisuje neodređenu listu gdje svaka stavka unutar, definirana oznakom `<li>`, sadrži poseban link sa posebnim imenom atributa. Tom oznakom definirala su se sva podpoglavlja u sadržaju. Neodređena listama grafičku oznaku u obliku točkica koji možemo ukloniti definiranjem stila u CSS-u: `ul{list-style-type:none;}`.

Podpoglavlja su se posebno uredila u CSS-u kako bi sadržaj bio pregledniji; pomaknuta su ulijevo naredbom `padding-left`.

```
16 <div id="sadržaj">
17 <ol>
18 <li><a href="#ind" class="naslovi">INDUSTRIJSKA EKOLOGIJA</a>
19 <ul>
20 <li><a href="#povijest" class="pov">1.1 Povijest industrijske ekologije</a>
21 <ul>
22 <li><a href="#prije_1960-ih" class="podpodnaslovi">1.1.1. Prije 1960-ih</a></li>
23 <li><a href="#1960-ih" class="podpodnaslovi">1.1.2. 1960-ih</a></li>
24 <li><a href="#1970-e" class="podpodnaslovi">1.1.3. 1970-e</a></li>
25 <li><a href="#1980-e" class="podpodnaslovi">1.1.4. 1980-e</a></li>
26 <li><a href="#1989_i_1990-e" class="podpodnaslovi">1.1.5. 1989. i 1990-e</a></li>
27 </ul>
28 </li>
29 </ul>
30 </li>
```

Slika 6. prikazuje primjer HTML koda liste i linkova za izradu sadržaja

Izvor:autor



Slika 7. Izgled sadržaja na web-u

Izvor:autor

Klikom na prvi naslov „Industrijska ekologija“ se odmah otvori poglavlje kojem smo dodali određeni id koji se nalazi u linku.



*Slika 8. Izgled prvog poglavlja u web pregledniku*

*Izvor:autor*

Svako poglavlje sadrži segmente: naslov <h> i paragraf <p>. U paragraf <p> se ugradio glavni tekst koje to poglavlje ima. Osim paragrafa, poglavlje može sadržavati i multimediju poput slika i videa te ostalih elemenata.

U ovom primjeru e-knjige, za umetanje multimedije, se koristio se element <figure> unutar kojeg bi se nalazili svi potrebni elementi određene slike odnosno videa.

Elementu <figure> se može pridružiti i opis kojeg postavljamo unutar

<figcaption> oznake kao što je prikazano na slici 9.

```
<figure>
  <video width="700" height="380" controls>
    <source src="Industrial Ecology-(480p).mp4" type="video/mp4">
  </video>
  <figcaption>Video 1. Objašnjava značenje industrijske ekologije.</figcaption>
</figure>
```

*Slika 9. Primjer korištenja oznake <figure> i <figcaption> za prikaz multimedije*

*Izvor:autor*

Svako poglavlje je postavljeno u poseban <div> element, a svaki citat unutar paragrafa se nalazi odvojeno u <blockquote> oznaci koja čini taj tekst izraženijim od ostalog te nije potrebno dodatno stiliziranje CSS stilovima.

U CSS-u se u jednom koraku napravio izgled svih paragrafa, naslova te podnaslova. Svaki paragraf ima isti id, kao i naslov i podnaslov svoj. Isto tako slike i videi imaju svoje attribute koje se jednim upisom u CSS file mogu upravljati.

```
/*GLAVNI TEKST*/
/*stil glavnog kontejnera i paragrafa su definirani u @media */
#h{
  font-size: 2em;
  font-weight: bold;
  padding-top: 0.5em;
  padding-left: 40px;
}
#h2{
  font-size: 18pt;
  font-weight: bold;
  margin-left: 1em;
}

figcaption {
  width:100%;
  margin: 20px auto 50px auto;
  font-family: 'Playfair_Italic', serif;
  text-align: center;
}
```

Slika 10. Prikaz koda u CSS-u za naslov, podnaslov i opis multimedije

Izvor:autor

Pomoću @media pravila se može uvesti mogućnost prikaza različitih fontova različitih veličina na određenoj vrsti ekrana, što bi se nazvalo responzivnim web dizajnom.

Definira za kakvu rezoluciju ekrana želimo kakav stil.

Specifično za ovaj slučaj su se dodale 3 vrste @media pravila: za ekrane manje od 600px, između 600 i 900px i ekrane iznad 900px. Važno je bilo podesiti izgled i veličinu teksta za bolju čitljivost na različitim veličinama ekrana. Za manje ekrane odabran je font bez serifa Roboto, a na većim ekranima prikazuje se tekst u fontu Playfair. Također, poravnavanje margina teksta na malim ekranima je postavljeno na obostrano poravnanje (*justify*) kako bi se prostor za ispis teksta bolje iskoristio, a tekst

bio uredniji i čitljiviji. Za veće ekrane poravnanje redaka je prebačeno na lijevo jer se smatra da je takav tekst čitljiviji.

Stilovi `#container` i `#p` se pojavljuju više puta u svakoj vrsti `@media` koje su navedene. Na početku izrade skripte se nalazi poseban div sa id-om `#container` koji definira širinu i margine glavnog sadržaja knjige. Širina sadržaja je na ekranima između 600px i 900px postavljena na 90% širine ekrana kako bi se što više iskoristio prostor ekrana. Font, razmaci između teksta i naslova, margine, razmaci između riječi te sve što ima utjecaja na tekst se dodalo u stilu za element paragraf `#p` (slika 11.).

```
/*prikazivanje za ekrane između 600 i 900px*/
#container {
  width: 90%;
  margin: 0 auto;
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
  box-sizing: border-box;
}
p{
  font-family:Playfair; /*na tabletima uredajima ce font paragrafa biti drugačiji nego na mobilnim*/
  margin:0;
  margin-bottom: 2em;
  word-spacing: normal;
  line-height: 1.5em;
  font-size: 1.5em;
  text-indent: 0;
  padding-left:50px;
}
```

*Slika 11. #container i #p za ekrane između 600 i 900px*

*Izvor:autor*

```
/*stilovi koji ce se primijeniti za ekrane veće od 900px*/
@media only screen and (min-width: 900px ) {
  #container {
    width: 70%;
    margin: 0 auto;
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box;
  }
  p{
    font-family:Playfair;/*na većim ekranima font se mijenja iz Roboto u Playfair*/
    margin:0;
    margin-bottom: 2em;
    word-spacing: normal;
    line-height: 1.5em;
    font-size: 18pt;
    text-indent: 0;
  }
}
```

*Slika 12. #container i #p za ekrane veće od 900px*

*Izvor:autor*

Na ekranima većima od 900px širina sadržaja od 90% bi proizvela predugačke redke koji bi bili nečitljivi, stoga je za velike ekrane iznad 900px širina sadržaja smanjena na 70% ekrana (slike 12 i 13).



Slika 13. Izgled paragrafa pri većim ekranima

Izvor:autor

Korištenjem JavaScript jezika postoji mogućnost izrade tzv. *Image comparison slider-a* preko `for` petlje te uvjetovanja `if` koji služi za usporedbu dviju slika. Na sredini se nalazi „gumb“ kojim možemo povećati jednu, a pri tom smanjiti drugu sliku i obratno.

Time se dobila mogućnost interakcija korisnika i knjige što u konačnici može značiti povećano zanimanje za sadržaj i duže zadržavanje pažnje korisnika.

U slučaju da neko poglavlje sadrži mnogo multimedije ili slika, moguće je "uštedjeti" prostor na način da se izradi, galerija slika u kojoj postoji mogućnost jednim klikom lijevo ili desno prijeći na sljedeću sliku. To se također postiglo uz pomoć vrsta petlji, `if` i `for`.

JavaScript kod za ove dvije vrste galerija je prikazan na slici 15.



*Slika 14. Prikazuje primjer Image comparison slider-a u web pregledniku*

*Izvor:autor*

```

<script>
function initComparisons() {
  var x, i;
  x = document.getElementsByClassName("img-comp-overlay");
  for (i = 0; i < x.length; i++) {
    compareImages(x[i]);
  }
  function compareImages(img) {
    var slider, img, clicked = 0, w, h;
    w = img.offsetWidth;
    h = img.offsetHeight;
    img.style.width = (w / 2) + "px";

    slider = document.createElement("DIV");
    slider.setAttribute("class", "img-comp-slider");

    img.parentElement.insertBefore(slider, img);

    slider.style.top = (h / 2) - (slider.offsetHeight / 2) + "px";
    slider.style.left = (w / 2) - (slider.offsetWidth / 2) + "px";

    slider.addEventListener("mousedown", slideReady);
    window.addEventListener("mouseup", slideFinish);

    slider.addEventListener("touchstart", slideReady);
    window.addEventListener("touchend", slideFinish);
    function slideReady(e) {
      e.preventDefault();
      clicked = 1;

      window.addEventListener("mousemove", slideMove);
      window.addEventListener("touchmove", slideMove);
    }
    function slideFinish() {
      clicked = 0;
    }
    function slideMove(e) {
      var pos;

      if (clicked == 0) return false;

      pos = getCursorPos(e);

      if (pos < 0) pos = 0;
      if (pos > w) pos = w;

      slide(pos);
    }
    function getCursorPos(e) {
      var a, x = 0;
      e = (e.changedTouches) ? e.changedTouches[0] : e;

      a = img.getBoundingClientRect();
      x = e.pageX - a.left;
      x = x - window.pageXOffset;
      return x;
    }
    function slide(x) {
      img.style.width = x + "px";
      slider.style.left = img.offsetWidth - (slider.offsetHeight / 2) + "px";
    }
  }
}
</script>

```

Slika 15. Prikaz javascript koda za Image comparison slider

Izvor:autor



*Slika 16. Prikazuje slike postavljene u galeriju napravljenu javascript kodom*

*Izvor:autor*



```

<div class="slideshow-container">
  <div class="mySlides fade">
    <div class="numbertext">1 / 3</div>
    
  </div>

  <div class="mySlides fade">
    <div class="numbertext">2 / 3</div>
    
  </div>

  <div class="mySlides fade">
    <div class="numbertext">3 / 3</div>
    
  </div>

  <a class="prev" onclick="plusSlides(-1)">&#10094;</a>
  <a class="next" onclick="plusSlides(1)">&#10095;</a>

</div>
<br>

<script>
var slideIndex = 1;
showSlides(slideIndex);

function plusSlides(n) {
  showSlides(slideIndex += n);
}

function currentSlide(n) {
  showSlides(slideIndex = n);
}

function currentSlide(n) {
  showSlides(slideIndex = n);
}

function showSlides(n) {
  var i;
  var slides = document.getElementsByClassName("mySlides");
  var dots = document.getElementsByClassName("dot");
  if (n > slides.length) {slideIndex = 1}
  if (n < 1) {slideIndex = slides.length}
  for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
  }
  for (i = 0; i < dots.length; i++) {
    dots[i].className = dots[i].className.replace(" active", "");
  }
  slides[slideIndex-1].style.display = "block";
  dots[slideIndex-1].className += " active";
}
</script>

```

*Slika 17. Prikazuje javascript kod za izradu galerije slika*

*Izvor:autor*

U CSS-u se može stvoriti i album slika različitih veličina i pozicije. Stvore se redak i stupac u kojem se odredi pozicija slika koje se dodaju u HTML-u.



*Slika 18. prikazuje izrađeni album slika*

*Izvor:autor*

```
/*album slika*/
* {
  box-sizing: border-box;
}
.header {
  text-align: center;
  padding: 32px;
}

.row {
  display: -ms-flexbox;
  display: flex;
  -ms-flex-wrap: wrap;
  flex-wrap: wrap;
  padding: 0 4px;
}

.column {
  -ms-flex: 25%;
  flex: 25%;
  max-width: 25%;
  padding: 0 4px;
}

.column img {
  margin-top: 8px;
  vertical-align: middle;
  width: 100%;
}
```

*Slika 19. Prikazuje stil za album slika izrađen u CSS-u*

*Izvor:autor*

Prilikom listanja i proučavanja skripte sa širokim rasponom materijala i velikim opusom stranica, trebala bi postojati mogućnost lakšeg povratka s jednog dijela skripte na početni sadržaj. U HTML-u se taj dio također postiže određenim petljama, a to je `else` petlja koja se svrsta u poseban `<div>` element.

```
<div class="button">
  <button onclick="topFunction()" id="myBtn" title="Go to top">Povratak na sadržaj</button>

  <script>
    var mybutton = document.getElementById("myBtn");

    window.onscroll = function() {scrollFunction()};

    function scrollFunction() {
      if (document.body.scrollTop > 20 || document.documentElement.scrollTop > 20) {
        mybutton.style.display = "block";
      } else {
        mybutton.style.display = "none";
      }
    }

    function topFunction() {
      document.body.scrollTop = 500;
      document.documentElement.scrollTop = 500;
    }
  </script>

```

*Slika 20. Prikazuje kod za stvaranje gumba za povratak iz bilo kojeg dijela skripte natrag na sadržaj*

*Izvor:autor*

#### 4. Zaključak

U ovom radu je zaključeno da je jedna od glavnih prednosti elektroničkih knjiga u odnosu na tiskane dostupnost više korisnika u isto vrijeme. Za razliku od tiskane knjige, uvijek ostaje ista i lakše je prenosiva.

Korištenje elektroničkog uređaja dovodi do loših posljedica kao što je problem s vidom ili glavobolja i isto tako može doći do gubitka baterije jer je elektroničkom uređaju potrebno napajanje strujom.

Teško je odrediti jesu li e-knjige bolje u odnosu na tiskane jer ovisi od osobe do osobe i subjektivno je, no zasigurno da su jeftinije i ekološki povoljnije jer je smanjena potrošnja papira i tinte.

Ovim radom prikazana je izrada interaktivne skripte koja je dostupna na web-u, a služi studentima za lakši način pristupanja materijalima te lakši način korištenja istih.

Skripta je optimizirana za jednostavnije čitanje na web-u, se prilagođava različitim dimenzijama ekrana. Studentima olakšava učenje jer sadrži multimediju koju mogu pregledati u samoj skripti (npr.pokretanje videa) kako bi bolje shvatili i usvojili gradivo. Isto tako, postoji mogućnost prelaska s jednog dijela skripte na drugi jednim klikom, čime se može uštedjeti na vremenu.

Time je prikazan moderniji način dostupnosti potrebne literature u bilo kojem trenutku s bilo kojeg uređaja - računala, tableta ili mobilnih telefona.

## 5. Literatura

1. Andrea Sudarević (2019.), Elektronička knjiga i marketing elektroničkog nakladništva u Hrvatskoj, Stručni rad
2. E-knjige i EPUB format, članak dostupan na:  
<https://loomen.carnet.hr/mod/book/tool/print/index.php?id=133556>, 31.8.2021.
3. What is OCR?, članak dostupan na: <https://www.necc.mass.edu/wp-content/uploads/accessible-media-necc/uncategorized/resources/What-is-OCR.pdf>, 31.8.2021.
4. Optical character recognition (OCR), članak dostupan na:  
<https://www.explainthatstuff.com/how-ocr-works.html>, 31.8.2021.
5. Marica Šapro-Ficović (2008.), Masovna digitalizacija knjiga: utjecaj na knjižnice, Stručni rad, Dubrovnik
6. Lea Walz (2019.), Knjižnica u budućnosti- digitalna i (li) multimedijalna knjižnica, diplomski rad, Sveučilište u Osijeku, Akademija za umjetnost i kulturu
7. What's an Ebook? Beginner's Guide to Ebook Creation and Distribution, članak na publishdrive, dostupno na: <https://publishdrive.com/whats-an-ebook.html>, 1.9.2021.
8. E-book, definicija dostupna na: <https://www.britannica.com/technology/e-book>, 1.9.2021.
9. Andrija Prelec (2015.), Izrada responzivne internetske stranice upotrebom html5 i css3 tehnologija, završni rad, Sveučilište u Zagrebu, Grafički fakultet

10.HTML 5. A vocabularyandassociatedAPIs for HTML and XHTML, rad dostupan na: HTML 5. A vocabularyandassociatedAPIs for HTML and XHTML". World Wide Web Consortium, 2.9.2021.