

SVEUČILIŠTE U ZAGREBU
FILOZOFSKI FAKULTET
ODSJEK ZA INFORMACIJSKE I KOMUNIKACIJSKE ZNANOSTI
Ak. god. 2017./ 2018.

Matija Marohnić

Responzivne slike u *web*-dizajnu

Diplomski rad

Mentor: dr.sc. Kristina Kocijan, doc.

Zagreb 2018.

Sadržaj

| | | |
|--------|---|----|
| 1. | Uvod..... | 4 |
| 2. | Uloga slika i potreba za optimizacijom..... | 5 |
| 3. | Primjeri upotrebe..... | 12 |
| 3.1. | Okvir prikaza..... | 12 |
| 3.2. | Gustoća zaslona..... | 15 |
| 3.3. | Likovno usmjeravanje..... | 17 |
| 3.4. | Značajke i tip medija..... | 19 |
| 3.5. | Slikovni formati..... | 20 |
| 3.6. | Kontrola nad odabirom veličine..... | 22 |
| 4. | Put prema standardizaciji..... | 24 |
| 4.1. | Nedostaci zaobilaznih rješenja..... | 24 |
| 4.2. | Standardizirana rješenja..... | 24 |
| 4.2.1. | HTML-atributi „srcset“ i „sizes“..... | 25 |
| 4.2.2. | HTML-element „picture“..... | 30 |
| 4.2.3. | CSS funkcija „image-set“..... | 33 |
| 4.2.4. | Klijentske HTTP-naznake..... | 35 |
| 4.2.1. | HTML-atributi „src-N“..... | 37 |
| 5. | Pristupačnost..... | 38 |
| 5.1. | Slabovidnost..... | 40 |
| 5.2. | Daltonizam..... | 43 |
| 6. | Automatizacija..... | 46 |
| 6.1. | Analiza koda..... | 46 |
| 6.2. | Kompresija slika..... | 46 |
| 6.3. | Web-servisi za slike..... | 48 |
| 6.4. | Korak dalje..... | 50 |
| 7. | Usporedba novinskih portala..... | 51 |
| 7.1. | Metoda..... | 51 |
| 7.2. | Rezultati..... | 53 |
| 8. | Zaključak..... | 57 |
| 9. | Literatura..... | 59 |
| 10. | Popis slika..... | 66 |

Sažetak

Raznolikost uređaja na kojima se danas pregledava *web* potaknula je programiranje responzivnih *web*-stranica. Zbog porasta broja mobilnih uređaja i velike cijene internetskog prometa u mnogim dijelovima svijeta počela je rasti potreba za dostavljanjem manjih slika na manjim uređajima. Programeri su se dosjetili raznih zaobilaznih rješenja za responzivne slike, najčešće pomoću JavaScripta. Međutim, svako je rješenje imalo nepremostive nedostatke. Bilo je potrebno izraditi specifikaciju za standardno rješenje koje bi *web*-preglednici znali prepoznati. Rješenje je moralo zadovoljavati najčešće načine prilagođavanja slika na *webu*, npr. prema gustoći zaslona, tipu medija, slikovnom formatu itd. Ovaj je izazov okupio grupu programera pod nazivom *Responsive Issues Community Group* (RICG), koji su tijekom nekoliko godina iteracija dovršili prijedlog za specifikaciju u obliku dodatnih HTML-atributa i novog elementa. Tu su specifikaciju nakon nekog vremena implementirali svi glavni *web*-preglednici, što znači da su ju programeri mogli početi koristiti. Osim na internetski promet važno je i osigurati pristupačnost slika slabovidnim osobama i daltonistima. Svakoj slici potrebno je dodati tekst koji opisuje nju ili njezinu funkciju i učiniti ju razumljivom onima koji ne mogu raspoznati sve tonove boja. Nakon pojave responzivnog *web*-dizajna održavanje slika postalo je znatno teže, zbog čega je korisno sagledati opcije za automatizaciju tog posla. Rješenja variraju od ručne kompresije slika do korištenja *web*-servisa specijaliziranih za slike. U radu se zatim uspoređuje stanje slika na najpopularnijim domaćim i stranim novinskim portalima na mobilnim i stolnim uređajima. Sagledavaju se ukupne težine slika kako bi se izračunao postotak optimizacije slika na mobilnim uređajima i kako bi se novinski portali usporedili međusobno. S obzirom da je težinu prosječne *web*-stranice čine većinom slike, važno je odrediti kad su slike doista potrebne, a kada nisu.

Ključne riječi: *responzivnost, web, slike, pristupačnost, automatizacija*

Responsive Images in Web Design

Abstract

The diversity of web-capable devices prompted the start of responsive web design. Because of the boom of mobile devices and the high prices of internet connection in many parts of the world, the need for delivering smaller images on smaller devices was rising. Programmers have thought of various workarounds for responsive images, most often using JavaScript. However, each solution suffered from insurmountable shortcomings. It was necessary to make a specification for a standard solution which web browsers could understand. The solution needed to satisfy most common use cases of responsive images, e. g. screen density, media type, image format etc. A group of people got together to tackle this challenge, they called themselves *Responsive Issues Community Group* (RICG), and after several years of iterations they came up with a proposal for a specification in the form of additional HTML attributes and a new element. Eventually all major web browsers have implemented this specification, which meant that programmers could start using it. Other than saving bandwidth it's also important to ensure that images are accessible to people who are visually impaired and colorblind. Each image needs to have a text which describes it or its function and make it easy to understand to people who can't tell some of its colors apart. After responsive web design maintaining images became a lot harder, which is why it's useful to look for options to automatize a part of that job. Solutions vary from manually compressing images to using web services specialized for images. In order to gain a sense of images on the web, images on popular local and foreign news sites are being compared on mobile and desktop devices. Total image weights are being calculated in order to see how much images are optimized for mobile devices and how news sites compare to each other. Considering that the total weight of an average web page consists mostly of images, it's important to decide when images are actually needed, and when they aren't.

Key words: *responsive, web, images, accessibility, automatization*

1. Uvod

World Wide Web izvorno je zamišljen kao mreža tekstualnih dokumenata (Wikipedia contributors, 2018g), ali kroz godine je postao i mreža mnogih drugih izvora poput slika, audio i video snimaka i sl. Još i danas upotrebljavamo termine poput „web-stranica“ i „HTML-dokument“, što asocira na tiskani medij.

Međutim, *web*-stranica nije papir A4 formata. Danas jedna osoba može pregledavati *web* na stolnom računalu, druga na e-čitaču, a treća na pametnom satu. Dolaskom mobilnih uređaja na tržište počelo je rasti i očekivanje za fluidnijim *webom* jer je porastom mobilnih aplikacija neprilagođenost *weba* kao medija postala sve vidljivija. Ethan Marcotte je 2010. godine skovao termin „responzivni *web*-dizajn“, koji je predstavljao fluidniji pristup programiranju *web*-stranica (Marcotte, 2010), što je vrlo važno s obzirom na to da većinu internetskog prometa danas čine upravo mobilni uređaji (Chaffey, 2018).

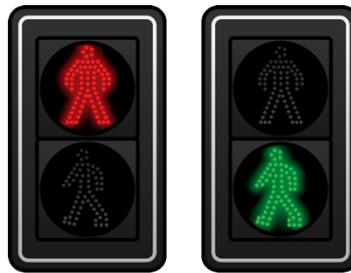
Osim dimenzija uređaja, bitno je razmišljati i o internetskom prometu, koji se najvećim dijelom sastoji od slika (HTTP Archive, 2018). Odabrao sam ovu temu jer su slike najizazovniji aspekt responzivnog *web*-dizajna. Ne radi se samo o rasporedu i dimenzijama elemenata dizajna, već je cilj dostaviti manje slike manjim uređajima kako bismo uštedjeli na internetskom prometu, koji je često ograničen na mobilnim uređajima.

Za početak ću opisati ulogu koju slike imaju na *webu* i općenito, a potom ću navesti i obrazložiti najčešće situacije u kojima je dobro prilagođavati slike. U trećem ću poglavlju navesti razne načine na koje su se programeri snalazili prije nego što su standardizirane responzivne slike, nakon čega ću objasniti kako se standardiziranim rješenjima doskače nedostacima zaobilaznih rješenja. U petom ću se poglavlju fokusirati na pristupačnost slika slabovidnim ljudima i daltonistima, a u šestom poglavlju navest ću neke metode i alate pomoću kojih možemo automatizirati dio posla oko dodavanja i održavanja responzivnih slika. Zatim ću u sedmom poglavlju analizirati slike na poznatim novinskim portalima i usporediti ih na stolnim i mobilnim uređajima. Rad ću zaključiti osvrtom na važnost slika i na važnost uštede internetskog prometa, ocijenit ću koliko su današnja rješenja za responzivne slike dorašla izazovima te završiti s pogledom u budućnost slika na *webu*.

2. Uloga slika i potreba za optimizacijom

Slike su neprocjenjivo sredstvo prijenosa informacija. Njima možemo efikasnije komunicirati, lakše izazvati željenu emocionalnu reakciju, a k tome ih pamtimo i znatno bolje od teksta (Seymour, 2015).

Upotrebu slikovnih elemenata možemo vidjeti u svakidašnjem životu. One mogu imati strogo definiranu ulogu, npr. funkcija semafora na Slika 1 jednoznačna je uglavnom zato što crvenu i zelenu boju univerzalno asociramo s lošim i dobrim i zato što jedna ikona prikazuje stajanje, a druga hodanje. U ovom je primjeru jednoznačnost slike izuzetno važna jer pogrešna interpretacija simbola može uzrokovati prometnu nesreću.



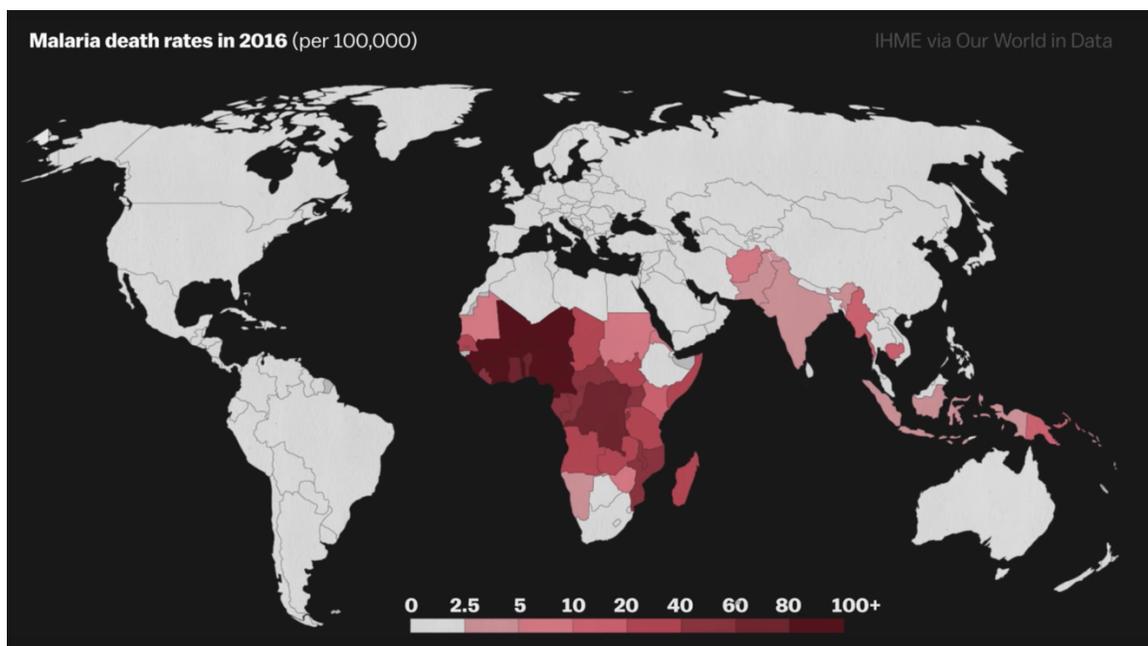
Slika 1: Semafor na pješačkom prijelazu.

Značenje nekih ikona nije jasno samo po sebi, nego ga usvojimo isto kao što i učimo jezik. Npr. ikona diskete na Slika 2 obično se koristi kao simbol za pohranjivanje datoteke. Međutim danas rijetko kada pohranjujemo datoteke na CD-ove, a kamoli diskete, tako da će s vremenom biti sve manje ljudi koji su ikad vidjeli disketu i ta će ikona postati apstraktna.



Slika 2: Disketa.

Kao potpuniji primjer uzmimo nedavno objavljeni video o borbi protiv malarije. Slika 3 predstavlja dio videa koji prikazuje stope smrtnosti ljudi oboljelih od malarije u svijetu (Vox, 2018). Ilustracija efikasno komunicira važne informacije: crvena boja, što je tamnija, uzrokuje neugodniji osjećaj jer asocira na krv, a koncentrirana je na područjima gdje je najveća stopa smrtnosti. Nadalje, unatoč tome što ne znamo napamet sve države, vrlo brzo možemo vidjeti da je stopa smrtnosti najveća u središnjem dijelu Afrike. Iste informacije izložene jedino u tekstovnom formatu ne bismo nužno toliko brzo usvojili.



Slika 3: Stopa smrtnosti ljudi oboljelih od malarije u svijetu (preuzeto iz Vox, 2018).

Funkcija Slika 3 je informiranje. No, slike možemo koristiti i kako bismo izazvali emocionalnu reakciju. U nastavku videa znanstvenik objašnjava kako se točno može koristiti genetska modifikacija za istrjebljivanje komaraca koji prenose malariju te koje su uopće implikacije genetske modifikacije potomaka. Postoji velik pritisak da se to istraživanje privede kraju kako bi se rezultati mogli provesti u praksi. Požurivanje istraživanja moglo bi poći po zlu upravo zato što se radi o genetskoj modifikaciji potomaka, ali što se dulje čeka, to više ljudi umire. Kevin Esvelt, znanstvenik sa sveučilišta Massachusetts Institute of Technology navodi: „Po pitanju malarije postoji snažan etički pritisak da se napravi nešto sada, danas. Mnogo je djece umrlo od malarije tijekom ovog

razgovora. Rekao bih, barem 20“ (Vox, 2018). Esvelt je ovom izjavom u potpunosti prenio željenu informaciju, njegovu poantu nije potrebno ilustrirati grafom, ali slikom možemo efikasno izazvati emocionalnu reakciju, kao što to postiže Slika 4.



Slika 4: Dijete u Africi oboljelo od malarije (preuzeto iz Vox, 2018).

Od slika smo dosad nabrojali ikone, grafove i fotografije i sve su imale jasnu funkciju. Međutim, to ne mora uvijek biti slučaj. Slika 5 naslikao je španjolski umjetnik Pablo Picasso nakon bombardiranja španjolskog sela Guernice. Interpretacije elemenata ove slike znatno variraju i mnoge su u sukobu jedna s drugom. Mnogi su od Picassa tražili objašnjenje, ali on je rekao da nije na njemu da svakom elementu pridaje značenje. Prema njegovom mišljenju, mnogo su važniji zaključci koje ljudi instinktivno i nesvjesno donose gledajući sliku, a da on slika elemente takvima kakvi jesu (Wikipedia contributors, 2018d).



Slika 5: Guernica, djelo španjolskog umjetnika Pabla Picassa (preuzeto iz Wikipedia contributors, 2018c).

Uloga slika u našem životu stvarno je raznolika i nezamjenjiva. Zato je već 1993. godine Marc Andreessen, koji je tada radio na *web*-pregledniku Mosaic, predložio uvođenje HTML-elementa „img“ u *web* u sličnom obliku u kojem postoji i danas (Andreessen, 1993).

S obzirom na to da prosječan posjetitelj provede 10-20 sekundi na *web*-stranici (Seymour, 2015), važno je zadržati pozornost i efikasno komunicirati. Slikama se na *webu* koristimo u razne svrhe:

- kao ukras, npr. morski valovi u pozadini *web*-stranica za tečajeve ronjenja;
- kao dio sadržaja, npr. graf koji prikazuje trend opisan u tekstu;
- kao način da efikasno pojasnimo funkciju elemenata korisničkog sučelja, npr. ikona koša za smeće ukazuje na brisanje.

Spektar uređaja na kojemu pregledavamo *web*-stranice raste velikom brzinom (vidi Slika 6). Prije 10 godina koristili smo se uglavnom stolnim računalima, a danas *web*-stranice pregledavamo na tabletima, televizorima, pa čak i na e-čitačima, koji sadržaj ispisuju pomoću elektroničke tinte.



Slika 6: Dio spektra uređaja kojima danas pregledavamo web-stranice (preuzeto iz Frost, 2012).

Nagli porast očituje se najviše kod uređaja s operacijskim sustavom Android, koji se proizvode 6 puta većom brzinom od uređaja s operacijskim sustavom iOS te ih danas ima u nebrojenim oblicima i veličinama (Frost, 2012).

Kad je rasprava o responzivnom *web*-dizajnu počela, spektar mobilnih uređaja bio je znatno manji i predvidljiviji. Dizajn *web*-stranica mogli smo prilagođavati pomoću predodređenih medijskih upita¹ u CSS-u baziranih na svojstvima postojećih uređaja. Na primjer, sljedeći medijski upit tada je pouzdano obuhvaćao samo iPhone 4 i 4s u okomitom i vodoravnom položaju (Graham, 2015):

```
@media only screen  
  and (min-device-width: 320px)  
  and (max-device-width: 480px)  
  and (-webkit-min-device-pixel-ratio: 2) { ... }
```

¹ Popis predloženih prijevoda manje poznatih termina u hrvatskom jeziku nalazi se u Prilogu 1.

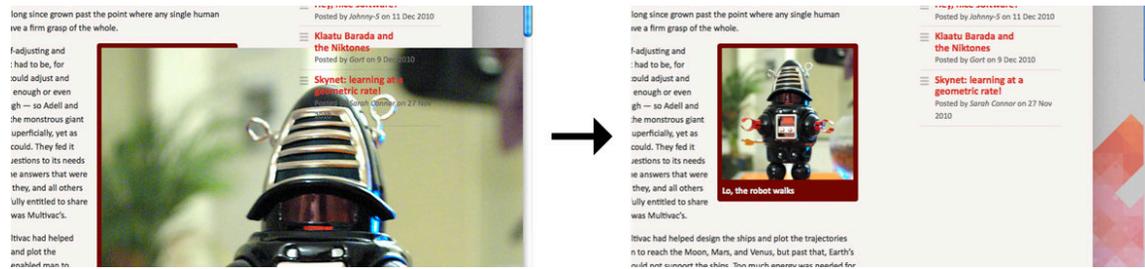
Taj način primjene preinaka u dizajnu bio je sigurniji jer smo pomoću ciljanih medijskih upita mogli pretpostaviti kapacitete uređaja. Takav pristup programiranju *web*-stranica zove se prilagodljivi web-dizajn (Graham, 2015). Glavna razlika između prilagodljivog i responzivnog web-dizajna upravo je fokus na pojedine uređaje. Međutim, točne definicije i razlike tih pojmova variraju u *web*-zajednici, npr. neki smatraju da je responzivni *web*-dizajn dio prilagodljivog. Ethan Marcotte koristio se terminom „prilagodljivi raspored elemenata“ kako bi opisao raspored elemenata koji izmjenjuje fiksne širine na određenim točkama loma, ali mnogi su počeli upotrebljavati termin „prilagodljivi dizajn“ kao sinonim za takav pristup (Graham, 2015; Frost, 2013). S vremenom se ustalio termin „responzivni *web*-dizajn“, pa tako i „responzivne slike“, zbog čega sam se u ovom radu odlučio koristiti tim terminima.

U svakom slučaju, voditi se pretpostavkama o uređajima korisnika postupno je počelo gubiti smisao. Zbog današnjeg spektra uređaja više ne možemo pouzdano ustvrditi kojim se uređajem naš korisnik služi niti bismo trebali pokušavati s obzirom na toliki spektar uređaja. Umjesto toga bismo trebali prigrliti rastuću raznolikost u tehnologiji i truditi se programirati što univerzalnije *web*-stranice.

Ljudi su nedostatak kontrole nad time kako će se *web*-stranice prikazivati u usporedbi s, na primjer, dizajnom ispisa, dugo smatrali kao manom tog medija. Međutim, to je zato što na *web* pokušavamo preslikavati ono što smo naučili iz drugih medija. To je ponekad korisno, a ponekad samo obrazac koji smo naučili. Umjesto da neizvjesnost interpretiramo kao manu medija, možemo to početi promatrati kao jedinstvenu *fleksibilnost* (Allsopp, 2000). Ubrzo nakon što je Ethan Marcotte skovao termin „responzivni *web*-dizajn“, demonstrirao je i prvi korak prema responzivnim slikama u obliku jedne linije koda u CSS-u:

```
max-width: 100%;
```

Matija Marohnić



Slika 7: Prije i nakon primjene CSS pravila „max-width: 100%“ na sliku (prilagođeno od Marcotte, 2011).

Tom linijom koda sprječava se prelijevanje slike izvan zadanih granica u dizajnu, kao što ilustrira Slika 7. Ta se tehnika zove „fluidne slike“ i predstavlja vrlo efikasno i minimalno rješenje za prilagođavanje dimenzija slike rasporedu elemenata okoline (Marcotte, 2011).

Međutim, tom se tehnikom doskočilo jedino problemu prikazivanja slika. Teške slike još se uvijek učitavaju jednakom brzinom i troše jednaku količinu podatkovnog prometa. Potreba za dostavljanjem lakših slika manjim uređajima time je još i više rasla.

Danas slike čine u prosjeku oko 60 % težine *web*-stranica (HTTP Archive, 2018) pa ulaganje vremena u optimizaciju slika može imati razne druge tehničke prednosti. Tako, na primjer, brzina učitavanja *web*-stranice znatno direktno utječe na njezinu iskoristivost s obzirom na pozornost korisnika, ograničenost podatkovnog prometa, pa čak i na rangiranje na *web*-tražilicama, kao što je Google objavio 2015. godine (Seymour, 2015).

S obzirom na toliki udio slika u težini *web*-stranice, svaka znatna ušteda dobro bi se odrazila na brzinu učitavanja, zbog čega se javila potreba za standardnim rješenjem. Kako bi to dokazao, Tim Kadlec je na uzorku od 471 *web*-stranice mjerio kolika bi otprilike bila ušteda kad bi postojalo takvo rješenje. Prema njegovim mjerenjima, na mobilnim uređajima bi se težina slika mogla smanjiti za oko 70 %, a na tabletima i prijenosnim računalima oko 50 %. Osim toga je dokazao koliko velike slike mogu negativno utjecati na performanse na manjim uređajima u pogledu klizanja kroz sadržaj, trošenja baterije i sl. (Kadlec, 2013a; Kadlec, 2013b).

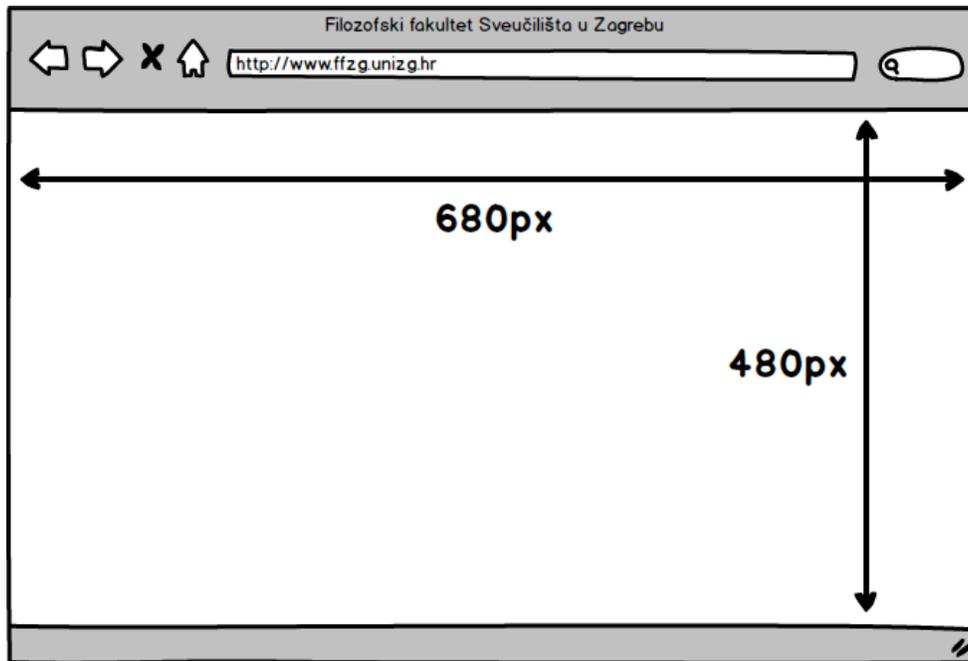
3. Primjeri upotrebe

Okolni uvjeti *web*-stranice očituju se primarno kroz značajke (npr. gustoća zaslona, orijentacija uređaja itd.) i tipove medija (npr. zaslona, pisač itd.). Mnoge su značajke medija dinamične, npr. u bilo kojem trenutku možemo na svom prijenosnom računaru promijeniti širinu preglednika, a pametne telefone okrenuti iz okomitog u vodoravni položaj. Neovisno o statičnosti ili dinamičnosti samog medija, responzivne slike prilagođavaju se okolnim uvjetima, što može uključivati prilagođavanje dimenzija, podrezivanje, pa čak i promjenu samog formata slike (Cáceres et al., 2013).

Ne možemo predvidjeti baš svaku situaciju u kojoj će se korisnik naći. No, glavni kriteriji za prilagođavanje slika jesu **okvir prikaza, gustoća zaslona, likovno usmjeravanje, točke loma, značajke i tip medija, slikovni formati i kontrola nad odabirom veličine**. Nešto više o njima reći ću u nastavku.

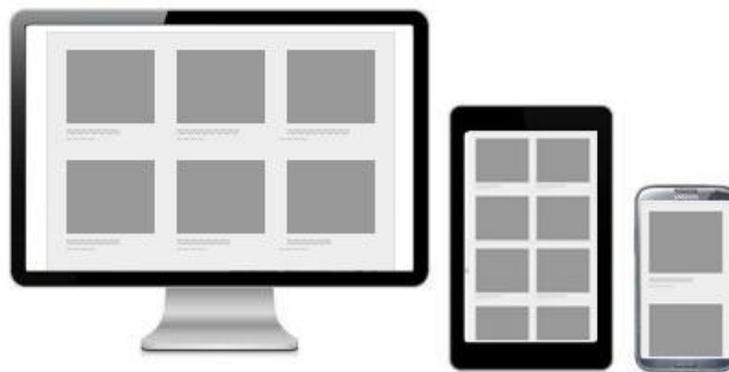
3.1. Okvir prikaza

Okvir prikaza jedan je od glavnih kriterija prema kojima prilagođavamo ne samo slike, nego i raspored elemenata na *web*-stranici. Dimenzije okvira prikaza često se podudaraju s dimenzijama zaslona, ali zapravo se odnose na dimenzije okvira *web*-preglednika u kojem se prikazuje *web*-stranica, ne uključujući komponente korisničkog sučelja kao što su izbornici, alatne trake i dr. (vidi Slika 8) (Cáceres et al., 2013).



Slika 8: Dimenzije okvira prikaza web-preglednika.

Okvir prikaza vrlo je pouzdan kriterij za responzivni *web*-dizajn jer pomoću njega znamo točno kojim prostorom raspolažemo. Slika 9 prikazuje tipičan primjer prilagođavanja rasporeda elemenata u galeriji slika. S obzirom na to da dimenzije okvira prikaza ovise o dimenzijama *web*-preglednika, kad bismo na stolnom računalu prikazanom s lijeve strane promijenili širinu *web*-preglednika npr. na širinu tableta, dobili bismo isti raspored elemenata kao i na slici tableta u sredini. Ovo je željeni rezultat jer je preporučljivo prilagođavati elemente *web*-stranice neovisno o vrsti uređaja (Cáceres et al., 2013).



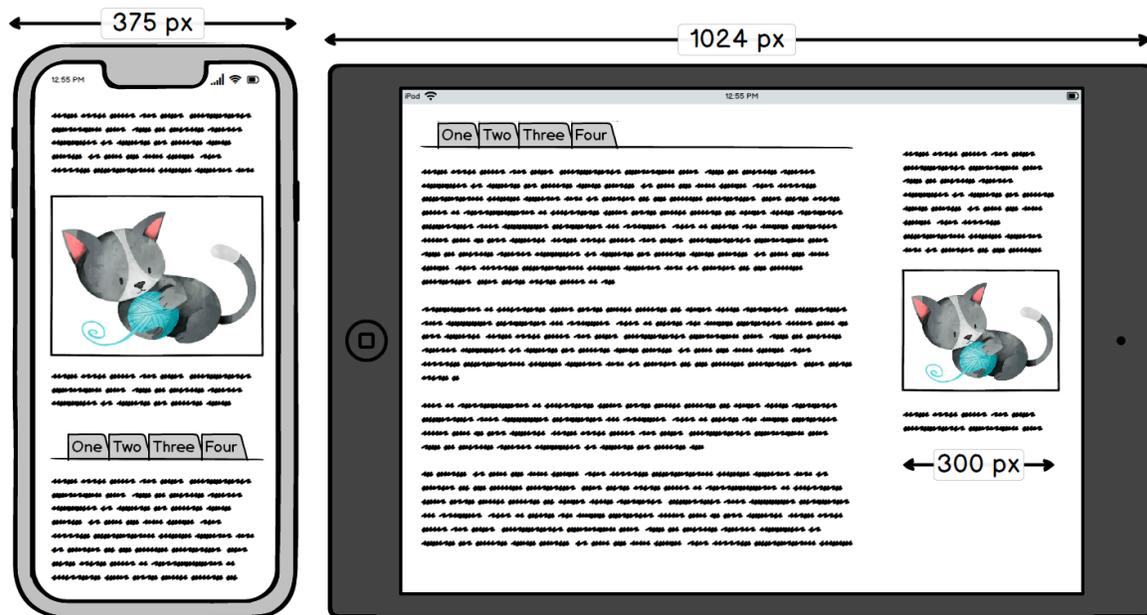
Slika 9: Različite veličine okvira prikaza (preuzeto iz Cáceres et al., 2013).

Kako se smanjuje širina okvira prikaza, tako se najčešće smanjuje i veličina prikazanih slika pa bi u tim slučajevima bilo idealno dostavljati manje slike radi uštede podatkovnog prometa i vremena potrebnog za učitavanje (Cáceres et al., 2013). Na primjer, ako u okviru prikaza širokom 2 560 piksela slike u galeriji imaju dimenzije 800x600 piksela, a u okviru prikaza širine 768 piksela slike imaju dimenzije 320x240 piksela, na manjem okviru prikaza nema smisla dostavljati slike dimenzija 800x600 piksela jer se te dimenzije neće iskoristiti. Umjesto toga možemo dostaviti manje slike kako bismo uštedjeli podatkovni promet.

Potreba za prilagođavanjem *web*-stranica prema okviru prikaza znatno je porasla 2007. godine nakon dolaska prvog iPhonea (Wikipedia contributors, 2018j), a idealno sredstvo za postizanje tog cilja bili su medijski upiti u CSS-u. Medijskim upitom navedenim u nastavku nastat će bočna traka širine 300 piksela ako širina okvira prikaza bude barem 512 piksela:

```
@media (min-width: 512px) {  
  #sidebar {  
    float: right;  
    width: 300px;  
  }  
}
```

Uvjet ili skup uvjeta medijskog upita naziva se točkom loma, u ovom slučaju to je minimalna širina okvira prikaza od 512 piksela. Slika 10 ilustrira taj primjer. S lijeve strane možemo vidjeti pametni telefon širine 375 piksela, što je manje od 512 piksela pa su elementi poredani jedan ispod drugoga. Međutim, s desne strane nalazi se tablet širine 1024 piksela, što zadovoljava uvjet medijskog upita, zbog čega se dio sadržaja pretvara u bočnu traku širine 300 piksela.



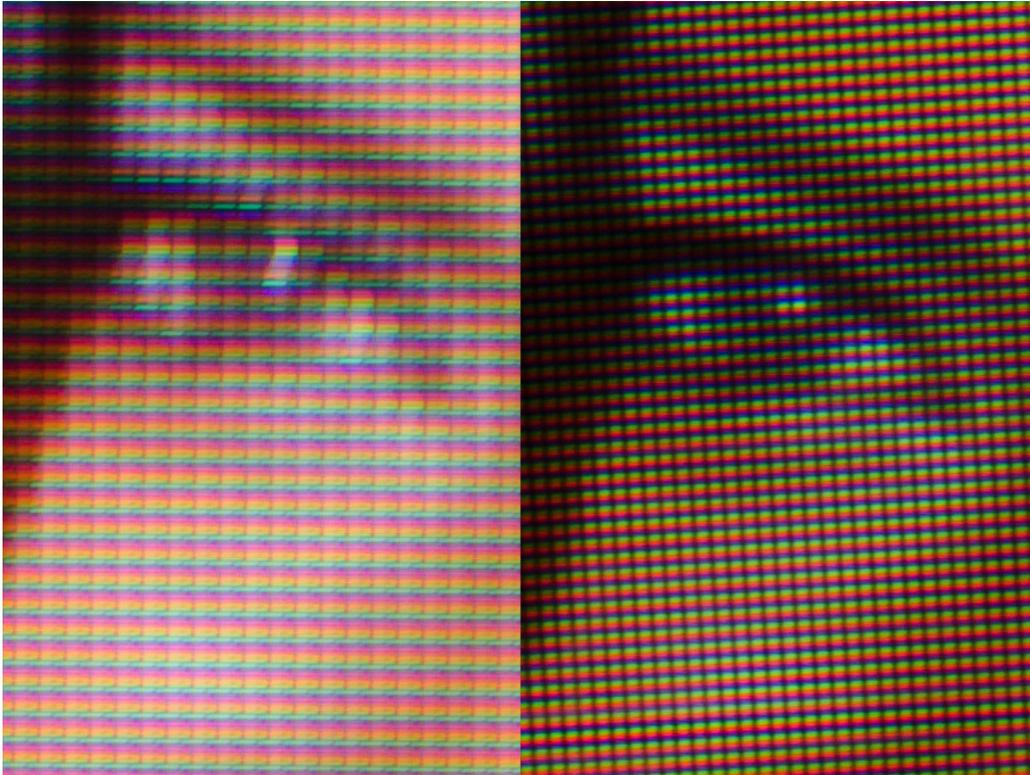
Slika 10: Primjer web-stranice koja mijenja izgled ovisno o širini okvira prikaza.

Danas možemo računati na podršku medijskih upita u *web*-preglednicima, ali to nije uvijek bio slučaj. Mozilla Firefox podržava medijske upite tek od 2009. godine, a Internet Explorer tek od 2011. godine objavom verzije 9 (Deveria, 2018). Međutim, čak ni tada nismo mogli računati na podršku za medijske upite jer su se mnogi korisnici interneta još uvijek služili sustavom Windows XP, koji ne podržava verzije Internet Explorera nakon verzije 8 (Wikipedia contributors, 2018f). Na sreću, Scott Jehl, zaposlenik tvrtke Filament Group, nedugo je nakon objave Internet Explorera 9 objavio biblioteku *Respond.js*, koja je nadomještavala funkcionalnost medijskih upita u preglednicima koji ih nisu podržavali (Jehl, 2018). Čak i ako netko ima isključen JavaScript, koji je potreban kako bi biblioteka mogla funkcionirati, u *web*-pregledniku koji ne podržava medijske upite, taj će ih *web*-preglednik naprosto ignorirati, a sadržaj *web*-stranice bit će dostupan.

3.2. Gustoća zaslona

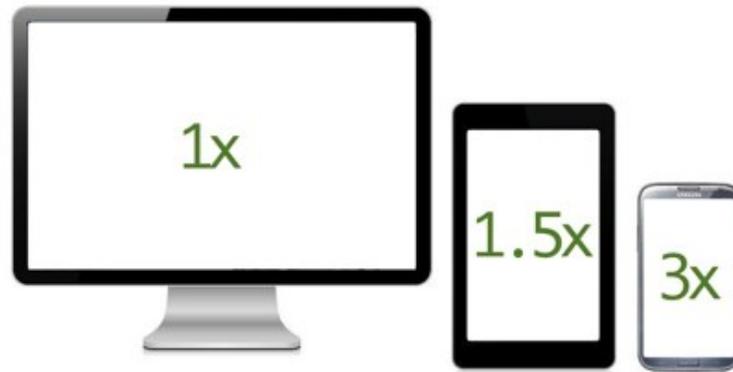
Tema gustoće zaslona dobila je zalet 2010. godine kada je Apple najavio iPhone 4 s Retina zaslonom, koji ima znatno veću gustoću od dotadašnjih zaslona (Wikipedia contributors, 2018c). Gustoća zaslona analogna je pisačima, npr. ispis laserskim pisačem

znatno je gušći od ispisa igličnim pisačem, koji se upotrebljava za ispis računa. Dok u tisku govorimo o gustoći točaka, kod zaslona govorimo o gustoći piksela, tj. gušći zaslon ima gušći raspored fizičkih piksela (Wikipedia contributors, 2018i). Gušći zaslon može prikazati veći broj detalja u istom prostoru, kao što možemo vidjeti na Slika 11.



Slika 11: Razlika između zaslona manje (lijevo) i veće (desno) gustoće (preuzeto iz Lein, 2012).

Kako bi slike na *webu* izgledale oštro na gušćem zaslonu potrebno je dostaviti više piksela, tj. uređajima s dvostruko gušćim zaslonom idealno bi bilo dostaviti dvostruko veću sliku (Cáceres et al., 2013). Različiti uređaji imaju zaslone različite gustoće (vidi Slika 12), potrebno je to uzeti u obzir kako ne bismo dostavili veće slike nego što je potrebno.



Slika 12: Gustoća zaslona različitih uređaja (preuzeto iz Cáceres et al., 2013).

3.3. Likovno usmjeravanje

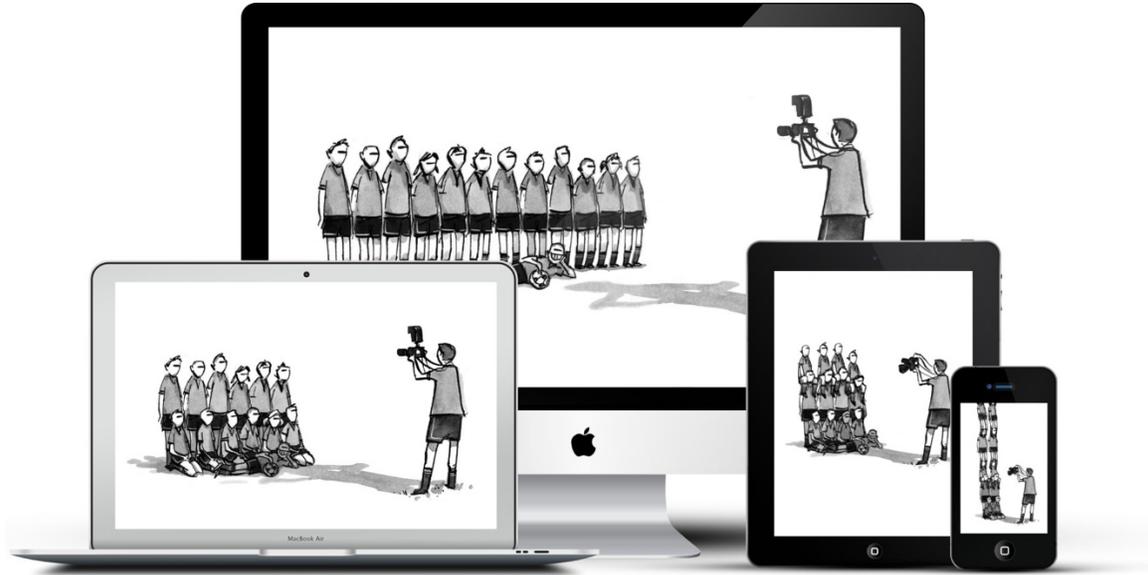
Kod velikih slika s mnogo detalja ponekad nije praktično dostaviti mobilnim uređajima smanjenu verziju iste slike jer može postati sve teže razaznati glavne značajke slike. U tim slučajevima može biti prikladnije podrezati sliku i tako istaknuti njezine glavne dijelove (vidi Slika 13) (Cáceres et al., 2013).



Slika 13: Podrezivanje slike kako bi pas bio prepoznatljiv na manjim zaslonima (preuzeto iz Cáceres et al., 2013).

U drugim slučajevima, kad podrezivanje ne bi znatno pomoglo, možemo dostaviti posve drugu sliku. Na primjer, možemo prilagoditi kompoziciju elemenata, kao na Slika 14 (Cáceres et al., 2013). Jedan od slučajeva u kojem bi to bilo uputno jest kada su nam svi

elementi važni. Prethodnu smo sliku podrezali jer smo odlučili da nam ostali elementi slike nisu ključni, ali na Slika 14 važno nam je da se vidi što više ljudi.



Slika 14: Likovno usmjeravanje promjenom kompozicije elemenata slike (prilagođeno od Marcotte, 2010).

Prilagođavanje kompozicije elemenata zahtijeva daleko više truda i planiranja unaprijed jer je potrebno donijeti tu odluku prilikom izrade same slike. To se isplati jedino za vrlo važne slike, npr. za vizualizaciju podataka (vidi Slika 15).



Slika 15: Likovno usmjeravanje vizualizacije podataka (preuzeto iz AmCharts, 2015).

3.4. Značajke i tip medija

Pisači obično gušće ispisuju sadržaj nego što su prikazani na zaslonu, zbog čega prilikom ispisivanja slika moraju nadomjestiti nedostatak slikovnih podataka polutonskom tehnikom² kako bi simulirali kontinuitet slike. Unatoč tome slike obično izgledaju mutnije nego tekst, kao što prikazuje Slika 16 (Cáceres et al., 2013).

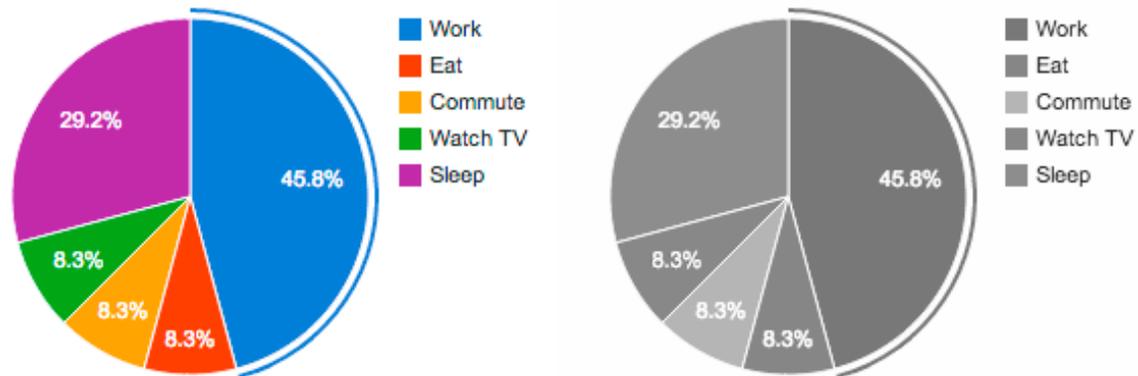


Slika 16: Primjer ispisa polutonskom tehnikom (prilagođeno od Cáceres et al., 2013).

Ako definiramo sliku u više različitih rezolucija, tada bi *web*-preglednici mogli odabrati sliku manje rezolucije radi uštede, a pisači bi npr. odabrali veću rezoluciju kako bi ispis bio što bolje kvalitete (Cáceres et al., 2013).

Drugi je izazov što sadržaj najčešće ispisujemo bez boja kako bismo uštedjeli toner. To ponekad može biti problematično kod slika na kojima je važna mogućnost razlikovanja boja. Kružni dijagram na Slika 17 ilustrira taj problem, gdje je na crno-bijelom dijagramu teško pridružiti boje dijagrama bojama u legendi. Taj problem ne može riješiti računalo, već je odgovornost na autorima da izrađuju slike koje mogu prenijeti glavne informacije i na monokromatskim medijima (Cáceres et al., 2013).

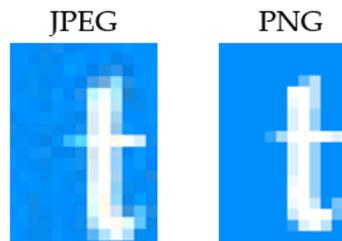
² Polutonska tehnika je grafička tehnika koja se koristi za simulaciju kontinuiteta tona slike pomoću točaka s varijacijama u veličini ili razmaku, čime se stvara privid prijelaza boje (Wikipedia contributors, 2018e; Wikipedia contributors, 2018a).



Slika 17: Dva primjera istog dijagrama, jedan je u boji, a drugi crno-bijeli (preuzeto iz Cáceres et al., 2013).

3.5. Slikovni formati

Različiti slikovni formati imaju različite prednosti i mana, npr. JPG je dobar za fotografije jer obično pruža dobru kompresiju bez vidljivog gubitka u kvaliteti (Cáceres et al., 2013). Međutim, pri kompresiji pravilnijih slika (npr. dijagram) lako se stvore artefakti³, kao što je vidljivo na Slika 18.



Slika 18: Usporedba slikovnih formata JPEG i PNG prilikom kompresije pravilne slike (preuzeto iz Green, 2013).

Slikovne formate ne bираmo samo prema sadržaju, nego i prema podržanosti u web-preglednicima. Primjer za to je WebP⁴ – jedan od novijih slikovnih formata koji trenutno podržavaju samo Chrome i Opera. S obzirom na to da WebP može biti do 25 % lakši u usporedbi s PNG-om i oko 25 – 35 % lakši u usporedbi s JPG-om (Seymour, 2015), mogli

³ Vidljivo izobličenje slike, audio ili video snimke i dr. uzrokovano sažimanjem s gubitkom podataka (Wikipedia contributors, 2018h).

⁴ Slovo „P“ u nazivu WebP predstavlja „Pictures“ ili „Photos“ (Massimino, 2018).

bismo npr. istu sliku dostaviti u WebP i JPG formatu na način da *web*-preglednici koji podržavaju WebP odaberu taj format, a ostali prikažu sliku kao JPG.

Na *webu* je najbolje težiti slikovnom formatu SVG kad god je to prikladno jer je iznimno lagan, fleksibilan i dobro podržan. S obzirom na to da je to vektorski format, izgleda oštro na bilo kojoj rezoluciji i gustoći zaslona, a moguće mu je i mijenjati izgled pomoću CSS-a, animirati ga i sl. (Cáceres et al., 2013; Coyier, 2016a). Idealan je za ikone jer možemo dodati element „svg“, koji sadržava sve željene ikone, direktno u tijelo našeg HTML-dokumenta (Coyier, 2014), npr.:

```
<svg style="display: none">
  <symbol id="plus" viewBox="0 0 16 16">
    <title>plus</title>
    <path d="M15 7h-6v-6h-2v6h-6v2h6v6h2v-6h6z"></path>
  </symbol>
</svg>
```

Ovim smo kodom definirali ikonu matematičkog simbola za zbrajanje, tj. „+“. Dodali smo stil „display: none“ kako taj element „svg“ ne bi zauzimao prostor na *web*-stranici jer ga upotrebljavamo isključivo za definiranje ikona. Možemo dodati ikona koliko god želimo, svaka mora biti zasebni element „symbol“ i imati jedinstvenu vrijednost atributa „id“.

Nakon što smo definirali ovu ikonu, možemo se njome koristiti pomoću elementa „use“ i mijenjati izgled pomoću atributa:

```
<svg width="32" height="32">
  <use xlink:href="#plus" />
</svg>

<svg width="64" height="64" transform="rotate(45)" fill="#00f"
fill-opacity="0.5">
  <use xlink:href="#plus" />
</svg>
```

Ovaj kod demonstrira samo dio fleksibilnosti koju pruža format SVG. *Web*-preglednik prikazat će dvije varijante iste ikone: prvu dimenzija 32x32 piksela i crne boje, a drugu dimenzija 64x64 piksela, okrenutu oko svoje osi za 45 stupnjeva, plave boje i s 50 % prozirnosti. Rezultat možemo vidjeti na Slika 19.



Slika 19: Ista ikona prikazana na dva različita načina pomoću koda SVG.

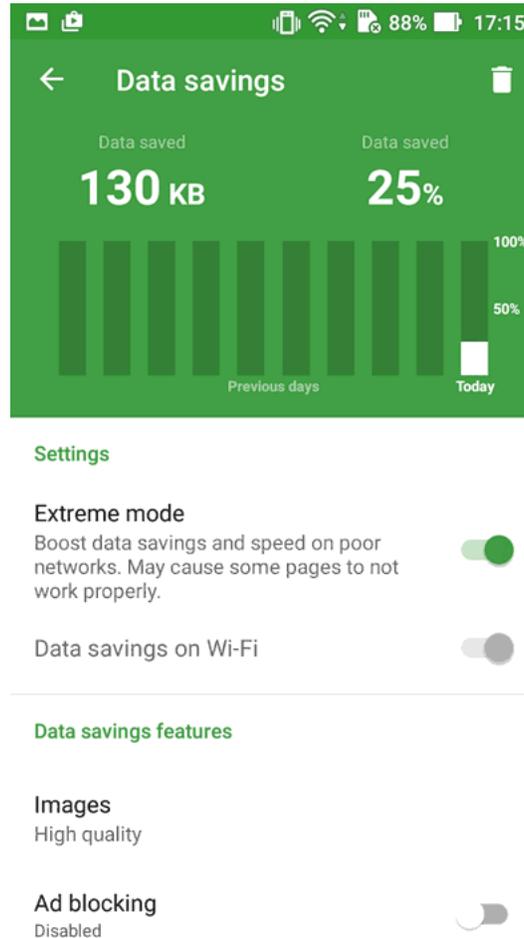
Osim fleksibilnosti prikaza, pomoću ovog pristupa korištenja SVG-a *web*-preglednik ne samo da ne mora trošiti vrijeme učitavajući datoteku, nego ne mora ni slati HTTP-zahtjev jer se sve informacije potrebne za prikaz slike nalaze u samom HTML-u (Coyier, 2016a).

3.6. Kontrola nad odabirom veličine

Mnogi od nas nisu svjesni koliko je Hrvatska dobro internetski umrežena država – imamo besplatni pristup internetu u gotovo svakom kafiću. Međutim, oni koji su putovali svijetom svjesni su da stanovnici mnogih drugih država nemaju takav luksuz. Stanovnicima Afrike i Azije internet je iznimno ograničen, a podatkovni promet skup (Hui Jing, 2017).

Zbog toga se služe mobilnim *web*-preglednikom Opera Mini jer pruža razne mogućnosti kompresije podataka pomoću svojih poslužitelja i omogućava uštedu podatkovnog prometa do 90 %. Slika 20 prikazuje jedan dio postavki Opere Mini na operacijskom sustavu Android, gdje možemo vidjeti informacije o uštedi podataka, mogućnost da isključimo prikazivanje slika i dr. (Hui Jing, 2017).

Operu Mini mnogi programeri zapostavljaju prilikom osiguravanja kompatibilnosti među *web*-preglednicima, unatoč tome što taj *web*-preglednik danas čini više internetskog prometa nego Edge i Safari zajedno (Hui Jing, 2017).



Slika 20: Postavke Opere Mini na operacijskom sustavu Android (preuzeto iz Hui Jing, 2017).

Opera Mini znatno lošije podržava mnoge moderne *web*-standarde u odnosu na ostale preglednike, ali činjenica da se njezini korisnici vjerno služe njima dobra je ilustracija toga koliko si ne mogu priuštiti učitavanje bilo kakvog viška podataka. Zbog toga je najbolje odabir veličine slike delegirati *web*-pregledniku, tako da korisnici imaju što veću kontrolu nad time koliko će podatkovnog prometa potrošiti (Cáceres et al., 2013). Međutim, Opera Mini komprimira slike pomoću svojih poslužitelja kao dio svoje usluge, ne nudi uslugu ručnog odabira manje slike. Trenutno nijedan *web*-preglednik nema takvu mogućnost temeljenu na specifikaciji responzivnih slika, koju ću obrazložiti u sljedećem poglavlju.

4. Put prema standardizaciji

4.1. Nedostaci zaobilaznih rješenja

Potreba za standardiziranim rješenjem za responzivne slike sve je više rasla, ali programeri nisu čekali, već su na razne načine pokušavali smanjiti težinu *web*-stranica pomoću tehnologija koje su imali na raspolaganju. Izniklo je mnogo rješenja, obično u obliku biblioteka u JavaScriptu. Bilo je potrebno pisati HTML-kod za slike na određen način, a neka su rješenja uključivala i skripte na poslužitelju. Svako rješenje ima svoje prednosti i nedostatke, koje su programeri trebali imati na umu prilikom odabira rješenja za svoj projekt (Coyier, 2012).

U svrhu bržeg učitavanja *web*-stranice mnogi *web*-preglednici prilikom obrađivanja HTMLkoda počinju učitavati slike čim dođu do elementa „img“. Ne postoji način da to spriječimo pomoću JavaScripta u svrhu učitavanja prikladnije veličine slike, što rezultira time da se učitaju dvije slike. Jedini način kako to možemo izbjeći je da izostavimo atribut „src“, čime se u potpunosti oslanjamo na JavaScript (Coyier, 2012). Međutim, to nije pouzdan pristup jer su neki korisnici isključili JavaScript u svom *web*-pregledniku⁵, a onima koji ga imaju uključenog može se dogoditi greška prilikom izvršavanja koda, što sprječava daljnje izvršavanje. Tako se može dogoditi da se ne izvrši dio koda zadužen za slike (Cáceres et al., 2013; Coyier, 2012; Marquis, 2014).

Moguće je umjesto toga postavljati pozadinske slike pomoću medijskih upita u CSS-u, ali tu nailazimo na slične negativne strane: ne koristimo se HTML-om na pravilan način i opet zaobilazimo mehanizam za učitavanje slika unaprijed.

4.2. Standardizirana rješenja

Zaobilazna rješenja imala su dugoročno neprihvatljive mane. Bilo je potrebno smisliti rješenja koje bi se elegantno nadograđivalo na postojeće standarde za sliku, na

⁵ Neki od razloga zašto ljudi biraju isključiti JavaScript su ušteda podatkovnog prometa, sigurnost, izbjegavanje reklama i dr.

način da su preglednici koji nisu implementirali tu specifikaciju sposobni prikazati pričuvenu sliku (Cáceres et al., 2013).

Problem responzivnih slika toliko je kompleksan da je bilo potrebno oformiti skupinu ljudi fokusiranih samo na to. Ta se skupina zove „*Responsive Images Community Group*“ ili, skraćeno, RICG (Seymour, 2015; Cáceres et al., 2013). RICG predvodi Mat Marquis i trenutno broji preko 400 sudionika, a neki od poznatijih su Marcos Cáceres iz Mozille, Jason Grigsby iz Cloud Foura i Mathias Byens iz Googlea. Misiju RICG-a čini osmišljavanje načina kako pomoću HTML-koda dostaviti slike u skladu s mogućnostima uređaja kako bi se izbjeglo suvišno trošenje podatkovnog prometa i optimizirao prikaz za zaslone i ispis (Community and Business Groups, 2018).

Sudionici RICG-a objavili su 2012. prvi nacrt specifikacije za responzivne slike (W3C Team, 2012), a 2014. godine ju je W3C (World Wide Web Consortium) pripojio službenoj specifikaciji za HTML (Seymour, 2015). Iste je godine RICG proširio svoju misiju i promijenio naziv u „*Responsive Issues Community Group*“, čime je uspio zadržati isti akronim (Community and Business Groups, 2018).

4.2.1. HTML-atributi „srcset“ i „sizes“

Osim navedenih mana, zaobilazna rješenja stavljaju previše odgovornosti na programera, zbog čega je bilo lako napraviti pogreške. Nijedno se zaobilazno rješenje nije isticalo jer je bilo nemoguće dobro implementirati responzivne slike bez novih specifikacija. To je zato što postoji veliki jaz između informacija koje programer ima tijekom kodiranja i informacija koje *web*-preglednik ima tijekom učitavanja *web*-stranice, kao što prikazuje Tablica 1 (Portis, 2014).

Jasno je vidljivo iz Tablica 1 da nijedna od varijabli nije istovremeno poznata oboma. Zato je bilo potrebno pronaći rješenje koje bi premostilo taj jaz. Pomoću novih atributa „srcset“ i „sizes“ programer može *web*-pregledniku dostaviti informacije koje mu nedostaju i prebaciti teret donošenja odluke o odabiru slike (Portis, 2014).

| varijabla | programer zna tijekom kodiranja | web- preglednik zna tijekom učitavanja web- stranice |
|-----------------------------|------------------------------------|--|
| dimenzije okvira prikaza | Ne | da |
| dimenzije prikazane slike | Da | ne |
| gustoća zaslona | Ne | da |
| dimenzije slikovne datoteke | Da | ne |

Tablica 1: Koje varijable o okolini poznaje programer, a koje web-preglednik.

Funkcioniranje tih atributa najbolje se može objasniti primjerom. Recimo da izrađujemo web-stranicu na kojoj jedna slika zauzima punu širinu okvira prikaza, dok god je ona ispod 480 piksela. Od 480 piksela nadalje slika zauzima širinu od 300 piksela. S pretpostavkom da smo implementirali željeni dizajn u CSS-u, novim atributima bismo to mogli izraziti na sljedeći način:

```

```

Atribute „alt“ i „src“ već poznajemo, atribut „src“ koristit će preglednici koji nisu implementirali ovu specifikaciju, a nove attribute „srcset“ i „sizes“ će ignorirati. Pomoću atributa „srcset“ definirali smo dvije veličine slikovnih datoteka koje imamo na raspolaganju. U atributu „sizes“ definirali smo da želimo da slika zauzima punu širinu okvira prikaza („100vw“⁶), ali od 480 piksela nadalje slika će zauzimati 300 piksela.

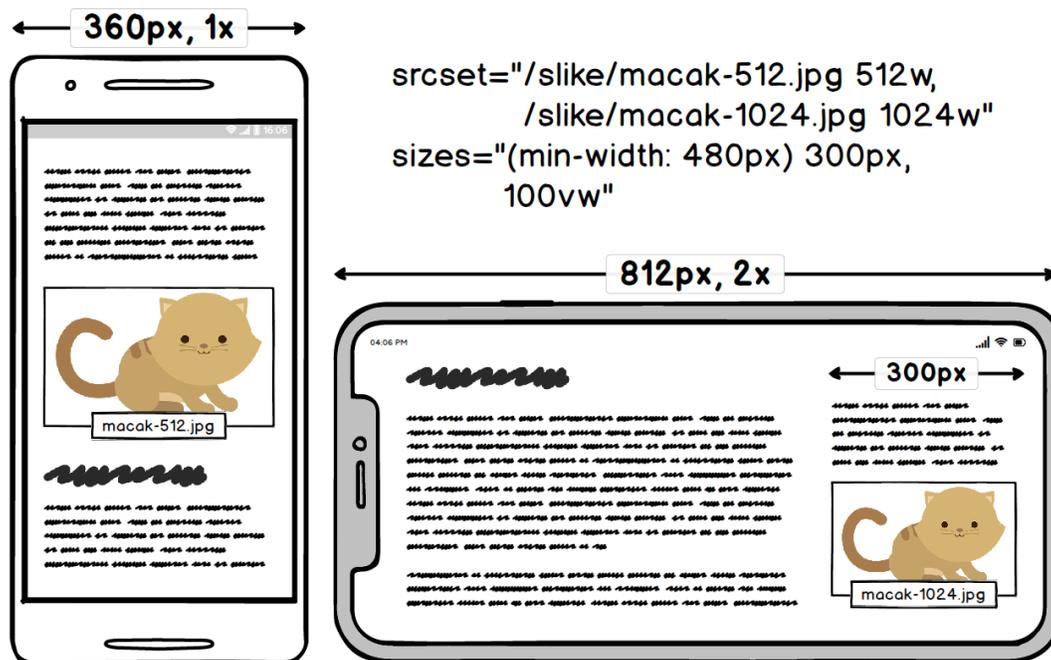
Sadržaj atributa „srcset“ je popis članova odvojenih zarezom, a članovi se sastoje od putanje slike i oznakom veličine ili gustoće zaslona. Atribut „sizes“ također je popis

⁶ Mjerna jedinica „vw“ (engl. *viewport width*) predstavlja postotak širine okvira prikaza.

članova odvojenih zarezom, čiji se članovi sastoje od medijskog upita i širine. Kad izostavimo medijski upit, taj se član primjenjuje ako nijedan od prethodnih članova nije odgovarao okolinskim uvjetima. S obzirom na to da dva medijska uvjeta mogu biti istinita istovremeno, *web*-preglednik čita popis po redu i uzima u obzir prvi izraz koji odgovara okolnim uvjetima (Portis, 2014).

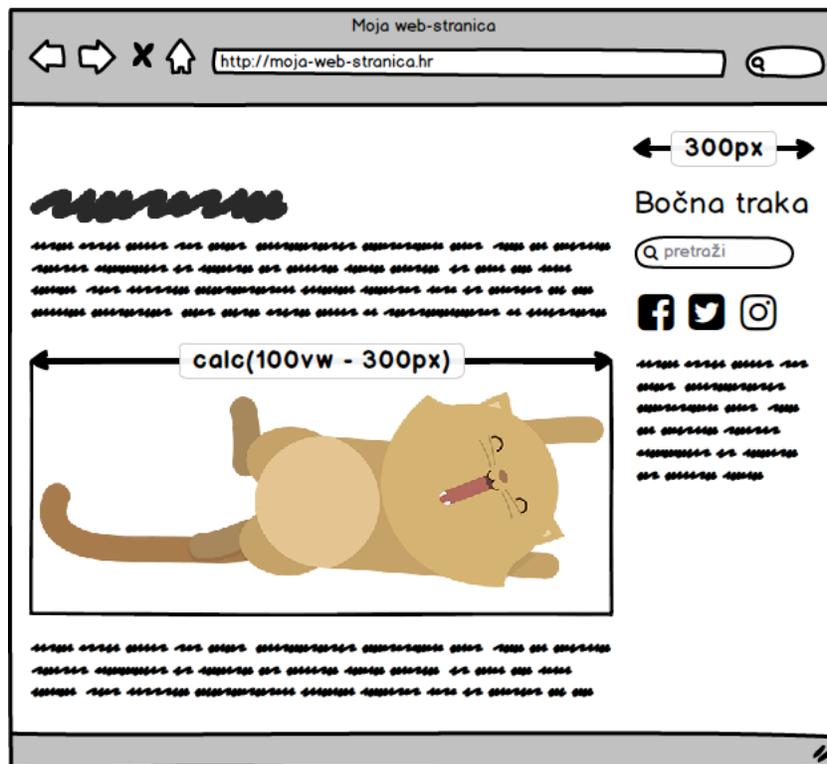
Ovo će raditi dobro i na zaslonima veće gustoće unatoč tome što nismo nigdje u kodu opisali gustoću. To je jedna od prednosti delegiranja odgovornosti *web*-pregledniku, jer on zna koja je gustoća zaslona uređaja te će prema tome odabrati veličinu slike koja ima najviše smisla.

Na primjer, *web*-preglednik na uređaju Android širine 360 piksela s jednostrukom gustoćom zaslona odabrat će sliku širine 512 piksela, ali iPhone s Retina zaslonom, koji ima širinu 812 piksela jer je okrenut vodoravno, odabrat će sliku širine 1024 piksela jer je širina prešla točku loma od 480 piksela, što znači da je sada slika široka 300 piksela, ali s obzirom na to da je zaslon dvostruko gušći, treba nam slika od barem 600 piksela, a taj uvjet zadovoljava slika od 1024 piksela (vidi Slika 21). Ovo je primjer koliko se računanje može lako zakomplicirati te koliko je korisno taj teret delegirati *web*-pregledniku.



Slika 21: Ilustracija kako nas atributi „srcset“ i „sizes“ mogu osloboditi kompliciranog računanja.

Atribut „sizes“ ne podržava samo mjerne jedinice „vw“ i „px“. Veličine u tom atributu izražavamo jednako kao i u CSS-u, što znači da možemo koristiti i funkcije poput „calc“, pomoću koje možemo definirati dinamične veličine koje drugačije ne bismo mogli; npr. „calc(100vw - 300px)“ predstavlja cijelu širinu okvira prikaza umanjenu za 300 piksela, kao što je ilustrirano na Slika 22 (Portis, 2014).



Slika 22: Primjer kada funkcija „calc“ može biti korisna.

U dosadašnjim sam primjerima radi jednostavnosti upotrebljavao samo dvije slikovne datoteke u atributu „srcset“. Na programerima je još uvijek izbor koliko će točno veličina ponuditi i koje. U ovom sam primjeru rastavio slike po dimenzijama u potencijama broja 2, ali često se preporučuje rastavljanje prema težini slike, jer je ušteda na težini svrha responzivnih slika (Grigsby, 2013; Grigsby, 2015b).

Još jedan parametar za upotrebu atributa „srcset“ je gustoća zaslona. To je korisno kada su dimenzije prikazane slike statične, ali želimo na uređajima s gušćim zaslonom

dostaviti veće slike kako bi izgledale oštrije. Tipičan je primjer logo jer on najčešće ostaje iste veličine (vidi Slika 23):

```

```



Slika 23: Prilagođavanje loga prema gustoći zaslona.

Oznake „1x“, „2x“ i „3x“ označavaju gustoću zaslona. Npr. s obzirom na to da je Retina zaslon *dvostruko* gušći od dotadašnjih zaslona, na njega će se odnositi oznaka „2x“ i *web*-preglednik će prikazati sliku čija je putanja „/slike/logo@2x.png“.

U ovom primjeru nema smisla koristiti atribut „sizes“. Naime, *web*-preglednik ne zna veličine slikovnih datoteka jer mu u ovom slučaju nismo mogli dati te informacije. Atribut „srcset“ ne podržava oznaku širine i gustoće zaslona istovremeno, može biti samo

jedno od toga. S obzirom na to da nema te informacije, ne bi ih mogao iskoristiti u kombinaciji s atributom „sizes“. Neovisno o dimenzijama prikazane slike, *web*-preglednik će odabrati sliku ovisno o gustoći zaslona (Kirtani, 2015).

Atribute „srcset“ i „sizes“ podržavaju svi *web*-preglednici osim Internet Explorera⁷ i Opere Mini⁸. Za te *web*-preglednike moguće je koristiti JavaScript biblioteku „*Picturefill*“ koja nadomješta tu mogućnost. S obzirom na to da je to zaobilazno rješenje, to će najčešće uzrokovati učitavanje dviju slika, a *web*-preglednici koji nemaju uključen JavaScript upotrijebit će atribut „src“. Sve to demonstrira koliko se ta specifikacija elegantno razgrađuje.

4.2.2. HTML-element „picture“

Element „img“ s atributima „srcset“ i „sizes“ dovoljan je za osnovnu implementaciju responzivnih slika, ali element „picture“ omogućava mnogo veću kontrolu. Koristan je npr. za likovno usmjeravanje, kad želimo koristiti novije slikovne formate i dr. (Grigsby, 2014). Sintaksa je slična kao i kod elementa „video“, tj. sadržava niz elemenata „source“, završavajući s elementom „img“:

```
<picture>  
  <source />  
  <source />  
  <source />  
    
</picture>
```

Elementi „source“ podržavaju razne attribute, uključujući „srcset“ i „sizes“, ali prije nego što uđemo u detalje toga važno je napomenuti da poredak elemenata „source“ nije nebitan jer će *web*-preglednici odabrati prvi po redu koji odgovara okolnim uvjetima.

⁷ Microsoft je već 2015. godine objavio prvu verziju svog novog *web*-preglednika Edge (Wikipedia contributors, 2018k), nakon čega je prestao dodavati nove značajke za Internet Explorer (Microsoft, 2018).

⁸ S obzirom na to da Operu Mini možemo koristiti jedino na mobilnim uređajima, nedostatak podrške ne predstavlja značajan problem ako atribut „src“ sadržava mobilnu inačicu slike.

Oni se ne ispisuju na *web*-stranici nego su način kojim definiramo kada želimo da se prikaže koja slika, a *web*-preglednik će izvor te slike primijeniti na posljednji element „img“, koji ne smijemo izostaviti jer se u suprotnom slika neće prikazati. Slika u atributu „src“ na elementu „img“ služi kao pričuvna u slučaju da nijedan element „source“ ne odgovara okolnim uvjetima ili ako *web*-preglednik uopće ne podržava element „picture“ (Chen, 2014).

Uzmimo kao primjer upotrebu novih slikovnih formata JPEG 2000 i WebP. S obzirom na to da JPEG 2000 podržava jedino Safari, a WebP podržavaju jedino Chrome i Opera Mini (Deveria, 2018), možemo pomoću elementa „picture“ definirati pričuvni format JPEG, koji podržavaju svi *web*-preglednici:

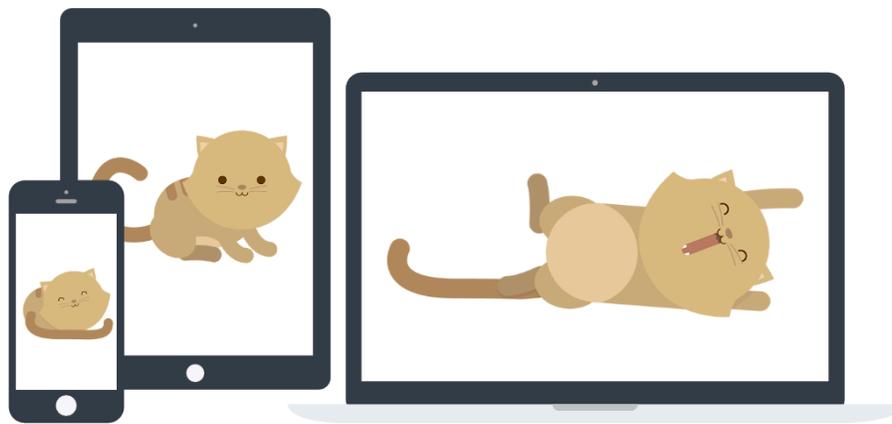
```
<picture>
  <source srcset="/slike/macak.jp2" type="image/jp2" />
  <source srcset="/slike/macak.webp" type="image/webp" />
  
</picture>
```

Web-preglednici koji ne podržavaju nijedan od formata definiranih u elementima „source“ koristit će atribut „src“.

Kompleksniji primjer čini likovno usmjeravanje; element „source“ podržava atribut „media“ kojim možemo odrediti točke loma:

```
<picture>
  <source
    media="(min-width: 2048px)"
    srcset="/slike/macak-ispruzeni.jpg" />
  <source
    media="(min-width: 1024px)"
    srcset="/slike/macak-sjedeci.jpg" />
  
</picture>
```

Ako je okvir prikaza širok barem 2048 piksela, prikazat će se slika ispruženog mačka, ako je širok barem 1024 piksela, prikazat će se mačak u sjedećem položaju, a u suprotnom će se prikazati mačak u skupljenom položaju, kao što prikazuje Slika 24. Kao što sam prije napomenuo, poredak elemenata „source“ je važan; kad bismo u ovom primjeru obrnuli njihov redoslijed, nikad se ne bi prikazao ispruženi mačak! *Web*-preglednik s okvirom prikaza od 3000 piksela odabrao je sliku mačka u sjedećem položaju jer je uvjet širine od barem 1024 piksela zadovoljen.



Slika 24: Mačak u različitim položajima ovisno o širini zaslona (prilagođeno od Chen, 2014).

Atribut „media“ ne mora se odnositi samo na dimenzije okvira prikaza; recimo da izrađujemo *web*-stranicu koju će naši posjetitelji često htjeti ispisati, ali da je na toj *web*-stranici slika koja sadržava osjetljive informacije (npr. lozinku). Ne možemo računati na to da će ljudi biti oprezni kako njihova lozinka ne bi dospjela u pogrešne ruke, pa možemo sakriti tu sliku ako je medij pisar:

```
<picture>
  <source
    media="print"
    srcset="/slike/osjetljive-informacije-skrivene.png" />
  
</picture>
```

Iako je ovo vrlo nevjerojatna situacija, dobra je ilustracija koje nam sve mogućnosti pruža element „picture“, te da se medijski upiti u atributu „media“ mogu kombinirati isto kao i u CSS-u.

Ovi su primjeri pojednostavljeni za potrebe demonstracije, ali u pravim projektima upotreba elementa „picture“ može znatno povećati količinu koda, a time i posao održavanja. Kad bismo u zadnjem primjeru s mačkom u različitim položajima dodali nekoliko varijacija za svaki položaj za različite veličine i gustoće zaslona te dodali atribut „sizes“ na elemente „source“ i „img“, količina koda narasla bi stvarno brzo. Zbog toga je zasad najbolje držati se elementa „img“ kada je to moguće (Grigsby, 2014).

Element „picture“ jednako je dobro podržan u *web*-preglednicima kao atributi „srcset“ i „sizes“, a za ostale *web*-preglednike možemo upotrebljavati biblioteku *Picturefill*.

4.2.3. CSS funkcija „image-set“

Specifikacija za responzivne slike u HTML-u bila je veći prioritet nego u CSS-u jer u CSS-u responzivnost možemo postići medijskim upitima. Međutim, za razliku od medijskih upita, atributi „srcset“ i „sizes“ ujedno služe i kao sugestije prema kojima *web*-preglednici mogu odabrati najprikladniju sliku (Coyier, 2016b). Uzmimo kao primjer sljedeći HTML-kod:

```

```

Pretpostavimo jednostruku gustoću zaslona (tj. 1x) i širinu okvira prikaza od barem 1024 piksela. Kad bismo ovaj kod prikazali u *web*-pregledniku Chrome, on bi učitao sliku

mačka širine 1024 piksela. Međutim, ako zatim širinu okvira prikaza smanjimo na 512 piksela, Chrome neće učitati sliku od 512 piksela jer je već učitao i prikazao veću sliku. Učitavanje manje slike bilo bi nepotrebno trošenje podatkovnog prometa jer korisnik ne bi primijetio razliku. Na ovom primjeru možemo uočiti kako su neki *web*-preglednici već počeli raditi na optimizaciji jer je uspostavljen standard za responzivne slike.

S druge strane, sljedeći CSS-kod uvijek će učitati manju sliku ako smanjimo veličinu okvira prikaza:

```
.slika {  
    background-image: url(/slike/macak-512.jpg);  
}  
@media (min-width: 1024px) {  
    .slika {  
        background-image: url(/slike/macak-1024.jpg);  
    }  
}
```

To je zato što *web*-preglednici ne mogu znati koja je namjera iza naših medijskih upita. Na primjer, možda koristimo tehniku likovnog usmjeravanja i u tom je slučaju važno prikazati sliku „macak-512.jpg“ na manjoj širini okvira prikaza. Međutim, glavni nedostatak toga što *web*-preglednici slijepo slijede naše medijske upite čini to što je teret računanja dimenzija responzivnih slika na nama.

Dobra vijest je što postoji specifikacija za CSS funkciju „image-set“ koja je slična atributu „srcset“. Loša vijest je što ona trenutno podržava samo gustoću zaslona (Atkins, Verou, and Etemad, 2018; Grigsby, 2015a; Coyier, 2016b). U sljedećem CSS kodu možemo vidjeti primjer korištenja te funkcije:

```
background-image: image-set(  
    planine@1x.jpg 1x,  
    planine@2x.jpg 2x,  
    planine@3x.jpg 3x);
```

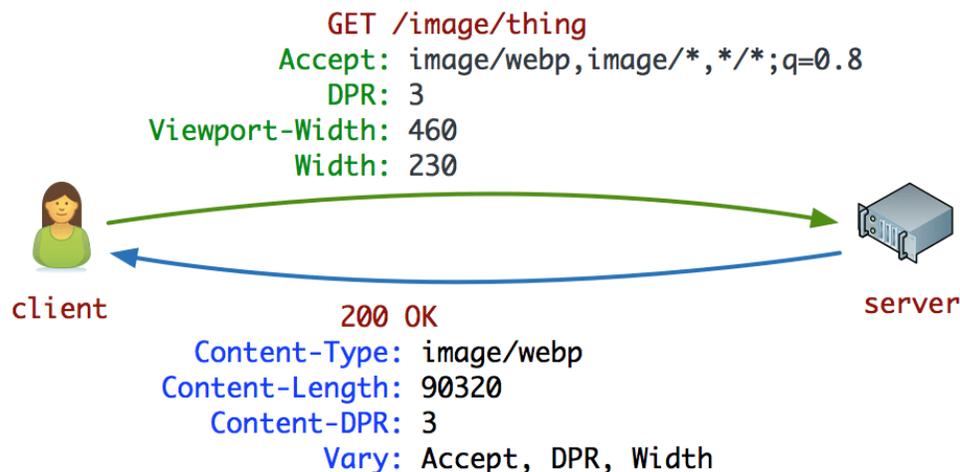
Specifikacija za funkciju „image-set“ nije još dovršena, trenutno postoji samo kao nacrt, a dodavanje podrške za širinu je u planu. Zbog tih razloga mnogi *web*-preglednici uopće je ne podržavaju, na primjer Internet Explorer, Edge i Firefox (Coyier, 2016b; Deveria, 2018; Atkins, Verou, and Etemad, 2018; Grigsby, 2015a).

4.2.4. Klijentske HTTP-naznake

Atributi „srcset“ i „sizes“ znatno olakšavaju responzivnost slika. Međutim, odluka na koliko različitih datoteka trebamo podijeliti sliku doista je proizvoljna i najčešće rezultira učitavanjem teže slike nego što je potrebno. Teret te odluke još je uvijek na *web*-programerima.

Klijentske HTTP-naznake su specifikacija koja propisuje kako *web*-preglednik može u HTTP-zaglavlju poslati poslužitelju metapodatke o okolini kao što su širina okvira pregleda, širina slike (prema atributu „sizes“), gustoća zaslona i sl. *Web*-preglednici koji podržavaju klijentske naznake ne šalju ih sami od sebe. Potrebno je dodati liniju koda u zaglavlju HTML-dokumenta kojim specificiramo koje informacije želimo slati, npr. gustoću zaslona i širinu (Grigsby, 2016):

```
<meta http-equiv="Accept-CH" content="DPR, Width">
```



Slika 25: Primjer metapodataka koje klijent može poslati poslužitelju.

Pomoću tih podataka poslužitelj koji podržava tu specifikaciju može dostaviti prikladnu sliku. Na taj način mogli bismo u potpunosti eliminirati atribut „srcset“, čime bismo znatno uštedjeli trud i smanjili količinu koda (Grigsby, 2016; Portis, 2016). Uz to bi korištenje elementa „picture“ bilo znatno lakše i pristupačnije.

Međutim, klijentske naznake zasad podržavaju jedino *web*-preglednici Chrome i Opera. Proizvođači *web*-preglednika često nisu skloni implementirati novu specifikaciju dok se ne uvjere da je to u praksi dobra ideja, a programeri nisu skloni koristiti nove implementacije dok nisu dobro podržane u svim preglednicima. Takva je dinamika analogna zagonetki „što je bilo prije, kokoš ili jaje?“.

Kad se programeri koriste novim specifikacijama, obično pronadu neki način da nadomjeste nedostatak podrške (najčešće pomoću JavaScripta), ali u ovom slučaju to nije moguće jer se radi o zahtjevu za sliku na razini *web*-preglednika, na koji nikako ne možemo utjecati jer, kao što sam opisao u zaobilaznim rješenjima, *web*-preglednici izvrše zahtjeve za slike prije nego što počnu izvršavati JavaScript (Portis, 2016).

Web-servis *Cloudinary* podržava klijentske naznake i pruža mogućnost pričuvne slike zadanih dimenzija, tj. možemo zatražiti sliku u skladu s klijentskim naznakama, a ako *Cloudinary* nije dobio klijentske naznake, poslat će nam sliku širine koju mu zadamo, npr. 800 piksela. U nekim situacijama to može biti prihvatljivo rješenje (Portis, 2016). Ovo je primjer kako bi takav zahtjev izgledao:

```

```

Ključan dio URL-a je „w_auto:128:512“, koji šalje *Cloudinary*ju tri informacije: signalizira mu da dostavi sliku u skladu s klijentskim naznakama, specificira veličinu skokova u dimenzijama slikovne datoteke i definira dimenzije pričuvne slike u slučaju da *web*-preglednik nije poslao širinu u klijentskim naznakama (zato što smo mi napravili neku grešku ili zato što *web*-preglednik uopće ne podržava klijentske oznake). Ako je veličina

skokova „1“, tada će Cloudinary za svaku jedinstvenu širinu okvira prikaza trebati stvoriti novu sliku, pa se malo iskorištavaju slike u pričuvnoj memoriji (Portis, 2016).

Drugi važan dio koda je atribut „sizes“, koji opisuje koju će širinu *web*-preglednik slati u HTTP-zaglavlju. U ovom slučaju to će biti širina okvira prikaza u trenutku kada *web*-preglednik šalje zahtjev.

4.2.1. HTML-atributi „src-N“

Specifikacija za attribute „src-N“ predložena je 2013. godine kao zamjena za attribute „srcset“ i element „picture“ (Cáceres, 2013), a predstavljala je pristup problemu responzivnih slika korištenjem specijaliziranih HTML-atributa „src-1“, „src-2“, „src-3“ itd. Vrijednosti tih atributa mogu sadržavati razne kombinacije putanja do slika, oznaka za širinu slika i gustoću zaslona i medijskih upita. Službeni nacrt specifikacije sadržava sljedeće primjere upotrebe atributa „src-N“ (Atkins, 2014):

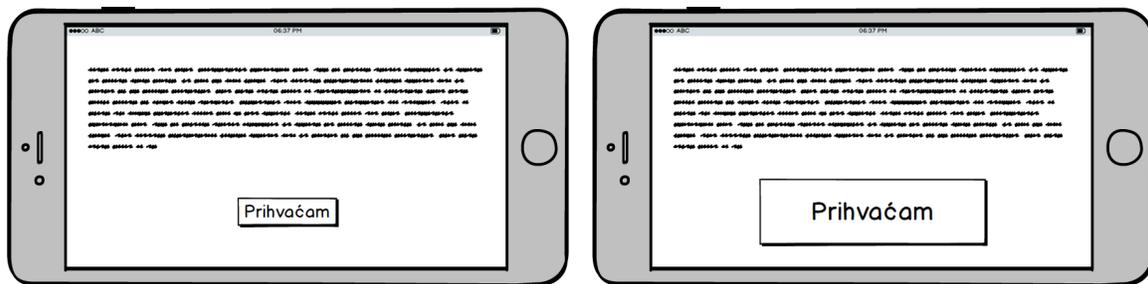
```
<img src-1="100%; pic1.png 160, pic2.png 320, pic3.png 640,  
pic4.png 1280, pic5.png 2560" />
```

```
<img  
  src-1="(max-width: 400px) pic1.png .5x, pic2.png 1x, pic3.png 2x"  
  src-2="(max-width: 800px) pic2.png .5x, pic3.png 1x, pic4.png 2x"  
  src-3="(max-width: 1600px) pic3.png .5x, pic4.png 1x, pic5.png 2x"  
  src-4="pic4.png .5x, pic5.png 1x"  
>
```

Međutim, taj je prijedlog nailazio na velik otpor zato što se numerirani atributi nisu dobro uklapali u HTML kao jezik, a njihove vrijednosti bile su previše fleksibilne i komplicirane. Zato je 2014. godine ovaj prijedlog **napušten**, a iz njega se izrodila ideja za atribut „sizes“ kao dio slagalice koji je nedostajao atributu „srcset“ i elementu „picture“ (Cáceres, Marquis, and Weiss, 2014).

5. Pristupačnost

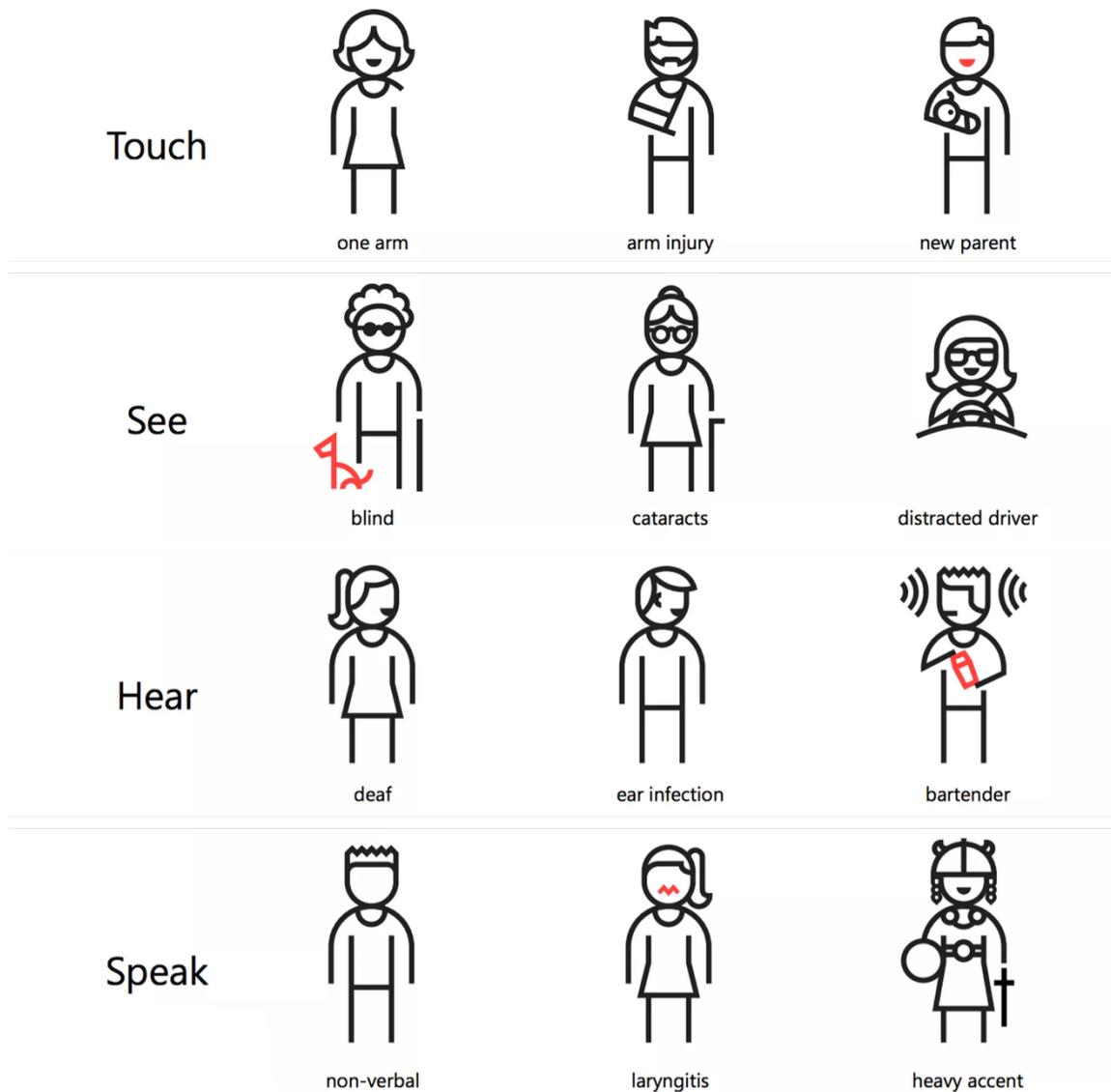
Ne možemo predvidjeti kojim će uređajima i softverom korisnici pregledavati našu *web*-stranicu, a kamoli potencijalno otežavajuće zdravstvene okolnosti kao što su slabovidnost, multipla skleroza, artritis itd. Na primjer, ljudima s motoričkim poteškoćama kao što je Parkinsonova bolest ponekad je teško naciļjati i pritisnuti gumb na *web*-stranici. Takve probleme možemo izbjeći ako povećamo površinu interaktivnim elementima (gumbima, poveznicama i sl.) (WebAIM, 2012), kao što demonstrira Slika 26.



Slika 26: Primjer iste *web*-stranice na mobitelu s manjim i većim gumbom.

Međutim, mnogima je potrebna pomoćna tehnologija kako bi se mogli koristiti računalom. Vrste pomoćne tehnologije ovise o poteškoćama, to mogu biti razne alternative za miš i tipkovnicu, senzor koji prati oči, softver za prepoznavanje govora, povećala za zaslon i dr. (Kalbag, 2017).

Upotreba računala nije otežana samo onima koji imaju trajnu ili privremenu zdravstvenu poteškoću. Korisnik može biti posve zdrav, samo u vrlo specifičnoj situaciji, npr. pod utjecajem alkohola, pod stresom, smeten brigom o svojem djetetu itd. Slika 27 prikazuje neke primjere konteksta, situacija i razdoblja u životu koji korisniku mogu otežati upotrebu nekog proizvoda. Pristupačnost *web*-stranice ne odnosi se na određenu skupinu ljudi, već je cilj da je može upotrebljavati što veći broj ljudi. Zato tvrtka Vox Media inzistira na terminu „univerzalni dizajn“ kao idealnom pristupu izrade proizvoda (Tanaka, Hearn, and Scherer, 2016).



Slika 27: Otežavajući konteksti, situacije i razdoblja u životu (prilagođeno od Tanaka, Hearn, and Scherer, 2016).

U kontekstu slika na *webu* važno je spomenuti dvije najvažnije zdravstvene poteškoće koje utječu na percepciju slika: **slabovidnost** i **daltonizam**. O njima ću govoriti u nastavku.

5.1. Slabovidnost

Preko 285 milijuna ljudi je slabovidno do mjere gdje im je potrebna pomoćna tehnologija. Slabovidnost se dijeli na tri razine: umjerena slabovidnost, teška slabovidnost i slijepost (Tarnoff, 2013). Kada govorimo o responzivnim slikama, važno je pričati o tome kako možemo pružiti opis onima koji ih imaju poteškoća vidjeti.

U kontekstu slabovidnosti, glavnu vrstu pomoćne tehnologije čine čitači zaslona, koji su postali simbol pristupačnosti na *webu*. U početku su čitači zaslona bili vrlo skupi i malo ljudi si ih je moglo priuštiti. Prva inačica čitača zaslona JAWS (*Job Access With Speech*) izrađena je 1995. godine, a još je i danas najpoznatiji i najnapredniji čitač zaslona. Međutim, vrlo je skup i cijena mu je oko 900 USD, zbog čega je rasla potreba za jeftinijim rješenjem (Kalbag, 2017). Apple je 2005. predstavio *VoiceOver*, čitač zaslona koji je bio dio samog operacijskog sustava (Wikipedia contributors, 2018). Tijekom zadnjih nekoliko godina na tržištu su se pojavili i besplatni čitači zaslona otvorenog koda kao NVDA (*NonVisual Desktop Access*), LSR (*Linux Screen Reader*) i dr. Microsoft je objavio svoj čitač zaslona, *Narrator*, koji je sličan Appleovom *VoiceOveru*. Osim čitača zaslona, slabovidni ljudi koriste se još i softverom za povećanje zaslona, a primjer djelomičnog i potpunog povećanja zaslona možemo vidjeti na Slika 28 (Kalbag, 2017).



Slika 28: Primjer djelomičnog (lijevo) i potpunog (desno) povećanja zaslona.

Zahvaljujući razvoju pomoćnih tehnologija *web* je postao znatno pristupačniji. Zato je nužno da se *web*-programeri informiraju o mogućnostima i ograničenjima pomoćnih tehnologija. U primjerima koda u prethodnom poglavlju upotrebljavali smo atribut „alt“.

On je najčešće najviše što možemo učiniti da bi naše slike bile pristupačnije ljudima koji se koriste čitačima zaslona i onima koji su isključili slike u svom *web*-pregledniku. Međutim, treba znati kako ga pravilno koristiti. Vrijednost atributa „alt“ najčešće bi trebala biti tekst koji opisuje funkciju slike u sadržaju. Ako slika ima tekstualni sadržaj, npr. logo, sadržaj atributa „alt“ trebao bi sadržavati taj tekstualni sadržaj. Također, alternativni tekst ne bi trebao počinjati sa „slika“ ili „fotografija“, npr. „fotografija planina“ jer pomoćne tehnologije same već najave da je to slika, pa je ponavljanje iste informacije u atributu „alt“ redundantno (The Accessibility Project, 2017; Kalbag, 2017).

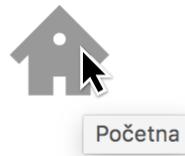
Ako je slika ukrasna, odnosno nije dio sadržaja, izostavljanje atributa „alt“ neće postići željeni cilj jer će neke pomoćne tehnologije pročitati ime datoteke kao alternativni tekst, npr. „IMG_0123.jpg“, što je zbunjujuće i može oduzeti mnogo vremena. Kako bismo pomoćnim tehnologijama naznačili da se radi o prezentacijskoj slici dovoljno je dodati prazan atribut „alt“ (The Accessibility Project, 2017; Kalbag, 2017), npr.:

```

```

Ne mora ista slika uvijek imati isti atribut „alt“. Ponekad ista slika na različitim mjestima ima različitu funkciju. Zato je važno fokusirati se na kontekst slike, a ne samo na sadržaj. Dobro pravilo pri odabiru teksta koji će poslužiti kao alternativni tekst slike je način na koji bismo ju opisali (ako uopće) nekome preko telefona (Kalbag, 2017).

Element „img“ samo je jedan od načina dodavanja slikovnog sadržaja na *web*-stranicu. U prijašnjim primjerima koristili smo se i elementom „svg“. Postoje dva elementa kojima možemo postići veću pristupačnost elementa „svg“, a to su „title“ i „desc“. Element „title“ trebao bi sadržavati kratki naslov slike, koji će se prikazati u oblačiću ispod pokazivača miša ako ga malo dulje zadržimo na slici, kao na Slika 29. Po potrebi možemo dodati i neobavezan element „desc“ koji bi trebao sadržavati dulji opis slike. Prema W3C-ovoj specifikaciji to bi trebalo biti sve što je potrebno kako bismo učinili element „svg“ pristupačnim.



Slika 29: Oblačić sa sadržajem elementa „title“.

No, neki *web*-preglednici još ne podržavaju ove elemente, zbog čega je potrebno koristiti atribute iz specifikacije WAI-ARIA (Web Accessibility Initiative – Accessible Rich Internet Applications) kako bismo elemente „title“ i „desc“ jednoznačno povezali kao naslov i opis slike. Specifikacija WAI-ARIA poznata je i pod skraćenim nazivom ARIA i njome se možemo koristiti za opisivanje uloga, stanja i drugih svojstava elemenata u aplikaciji za pomoćne tehnologije (Migliorisi, 2016; Kalbag, 2017). Atributi koji nam trebaju u ovom slučaju su „aria-labelledby“ i „role“:

```
<svg
  role="img"
  aria-labelledby="macakNaslov macakOpis"
  viewBox="0 0 720 800"
>
  <title id="macakNaslov">Smeđi mačak</title>
  <desc id="macakOpis">Smeđi mačak sa žutim očima i bijelim
  šapama koji leži na travi</desc>
  <!-- ... -->
</svg>
```

Pomoću atributa „role“ naznačujemo čitačima zaslona da element „svg“ predstavlja običnu sliku koja ima samo naslov i eventualno opis, što je prikladno ako slika ne sadržava tekst. Upotrebom atributa „aria-labelledby“ pridružili smo elemente „title“ i „desc“ kao naslov i opis slike, koristeći vrijednosti njihovih atributa „id“ (Migliorisi, 2016).

5.2. Daltonizam

Daltonizam je poremećaj raspoznavanja boja, a zahvaća čak jednog od 12 muškaraca (8 %) i jednu od 200 žena (0,5 %) (Colour Blind Awareness, 2018a). Mnoge su poznate osobe daltonisti, poput Marka Twaina i Billa Clintona (HRFnd, 2013). Daltonizam se prenosi najčešće genetski, ali može se razviti kao posljedica bolesti poput dijabetesa i multiple skleroze ili kao posljedica starenja, uzimanja određenih lijekova i dr. (Colour Blind Awareness, 2018a). Daltonizam zahvaća pretežito muškarce zato što se prenosi preko kromosoma X, a s obzirom na to da žene imaju dva kromosoma X, drugi kromosom najčešće nadomještava taj nedostatak (Jager, 2017).

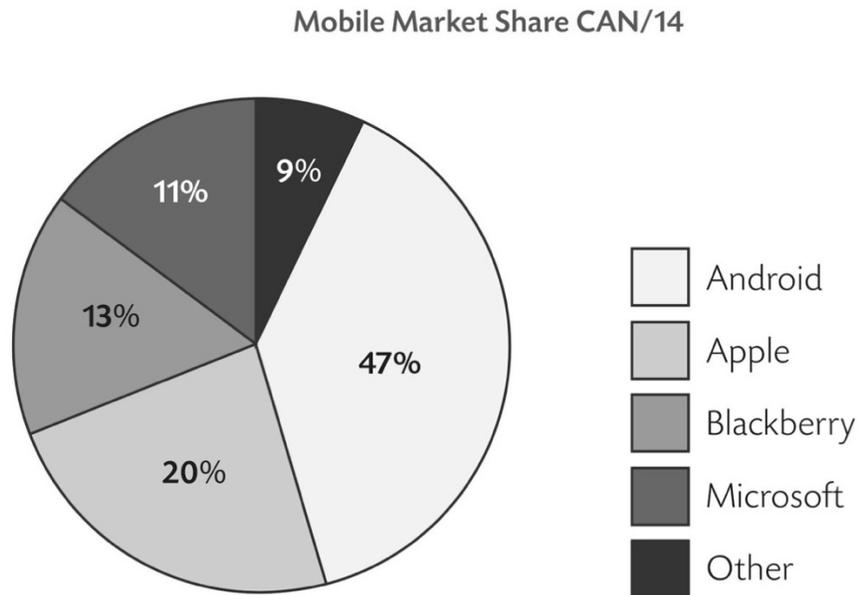
Česta je zablude da daltonisti **uopće** ne mogu raspoznati boje, ali to je iznimno rijetko. Većina daltonista u manjoj ili većoj mjeri imaju poteškoća u raspoznavanju određenih **tonova** u spektru, najčešće crvenih i zelenih. Razliku u percepciji iste slike između ljudi s normalnim vidom i daltonista koji ne raspoznaju crvene tonove možemo vidjeti na Slika 30 (Colour Blind Awareness, 2018b).



Slika 30: Normalan vid (lijevo) i nemogućnost raspoznavanja crvenih tonova (desno) (prilagođeno od Colour Blind Awareness, 2018b).

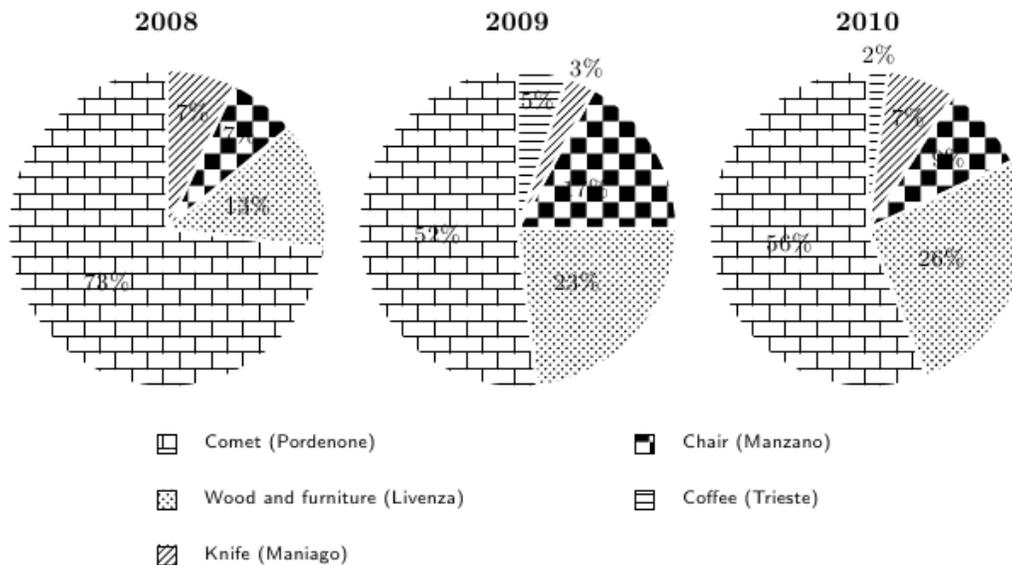
Mogućnost raspoznavanja boja nije uvijek važna, ali može biti ključna u vizualizaciji podataka, gdje su grupe podataka često označene određenom bojom. S obzirom na to da postoje različiti tipovi daltonizma, važno je ne donositi pretpostavke o tome kojem tipu pripada naš posjetitelj i učiniti svoj graf razumljivim čak i onima koji uopće ne raspoznaju boje (Jager, 2017). Uostalom, korisnik s normalnim vidom koji ispiše

naš graf crno-bijelom tehnikom također bi ga trebao moći razumjeti. Primjer čitljivog crno-bijelog grafa možemo vidjeti na Slika 31.



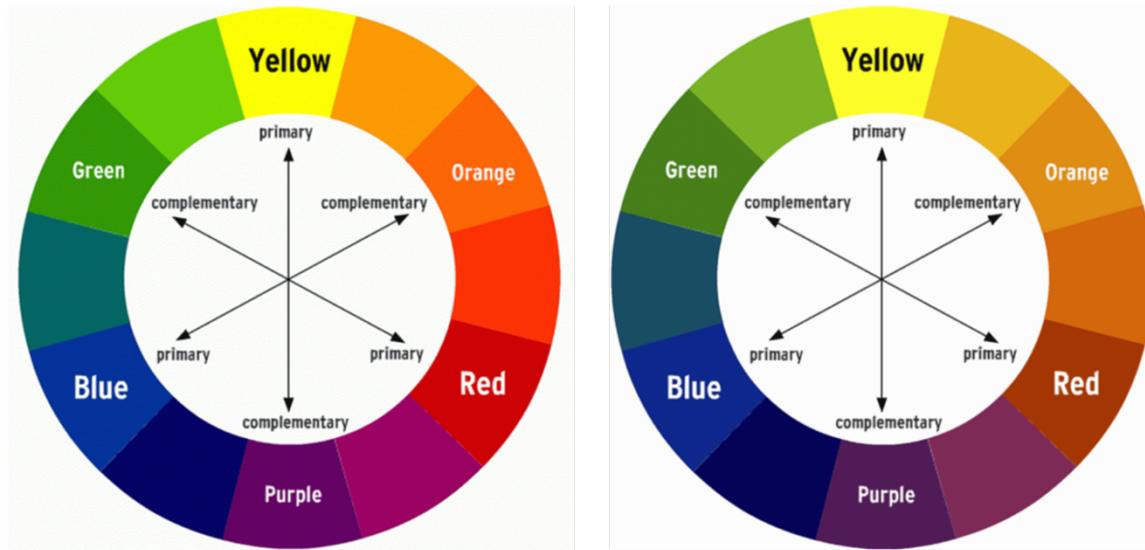
Slika 31: Primjer razumljivog crno-bijelog grafa.

Još jedan način kako učiniti crno-bijeli graf prepoznatljivim je pomoću uzoraka. Ovakve grafove je znatno teže izraditi i nije uvijek praktično, npr. na Slika 32 se na desnom grafu isječak grafa od 2 % jedva vidi.



Slika 32: Označavanje setova podataka u grafu pomoću uzoraka (prilagođeno od TeX - LaTeX Stack Exchange, 2014).

Kako bi grafovi u boji bili što pristupačniji daltonistima, potrebno je unijeti određenu količinu redundancije u sliku zato da razumijevanje grafa ne ovisi isključivo o nijansi boje. Boje imaju tri svojstva: nijansu, zasićenost i svjetlinu. Naučiti baratare tim svojstvima korisno je prilikom odabira sheme boja za vizualizaciju podataka, npr. na Slika 31 svjetlina je jedino svojstvo po kojem možemo raspoznati grupe podataka, tj. svjetlinom postizemo kontrast. Važno je upotrebljavati boje velike zasićenosti jer time olakšavamo raspoznavanje boja daltonistima koji djelomično mogu raspoznavati tonove. Na Slika 33 možemo vidjeti primjer grafa s pažljivo odabranom shemom boja kako ga vide ljudi s umjerenim crveno-zelenim daltonizmom (Jager, 2017).



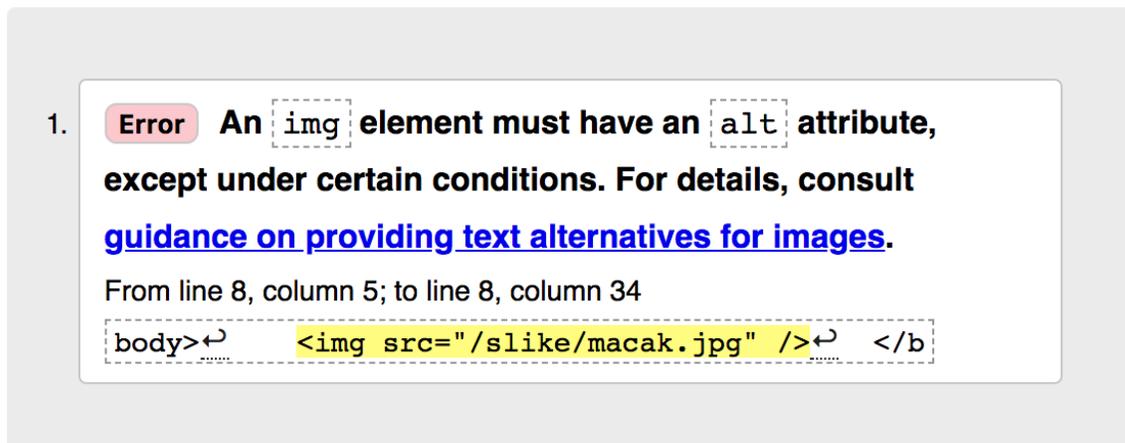
Slika 33: Razlika kako isti graf vide ljudi s normalnim vidom (lijevo) i oni s umjerenim crveno-zelenim daltonizmom (preuzeto iz Jager, 2017).

6. Automatizacija

Upoznali smo se s raznim tehnikama implementacije responzivnih slika, međutim što više slika imamo, to nam je posao teži. Stoga ćemo u ovom poglavlju sagledati načine na koje možemo automatizirati dio posla oko optimizacije i održavanja responzivnih slika.

6.1. Analiza koda

Za početak možemo provjeriti jesmo li ispravno kodirali slike u HTML-u koristeći W3C-ov servis za validaciju HTML-a. Tako možemo saznati jesmo li slučajno izostavili neki atribut kao „alt“, pogriješili u tipkanju i/ili jesu li vrijednosti korištenih atributa u ispravnom formatu. Slika 34 prikazuje primjer greške prilikom validacije HTML-koda gdje elementu „img“ nedostaje atribut „alt“ (W3C, 2018).



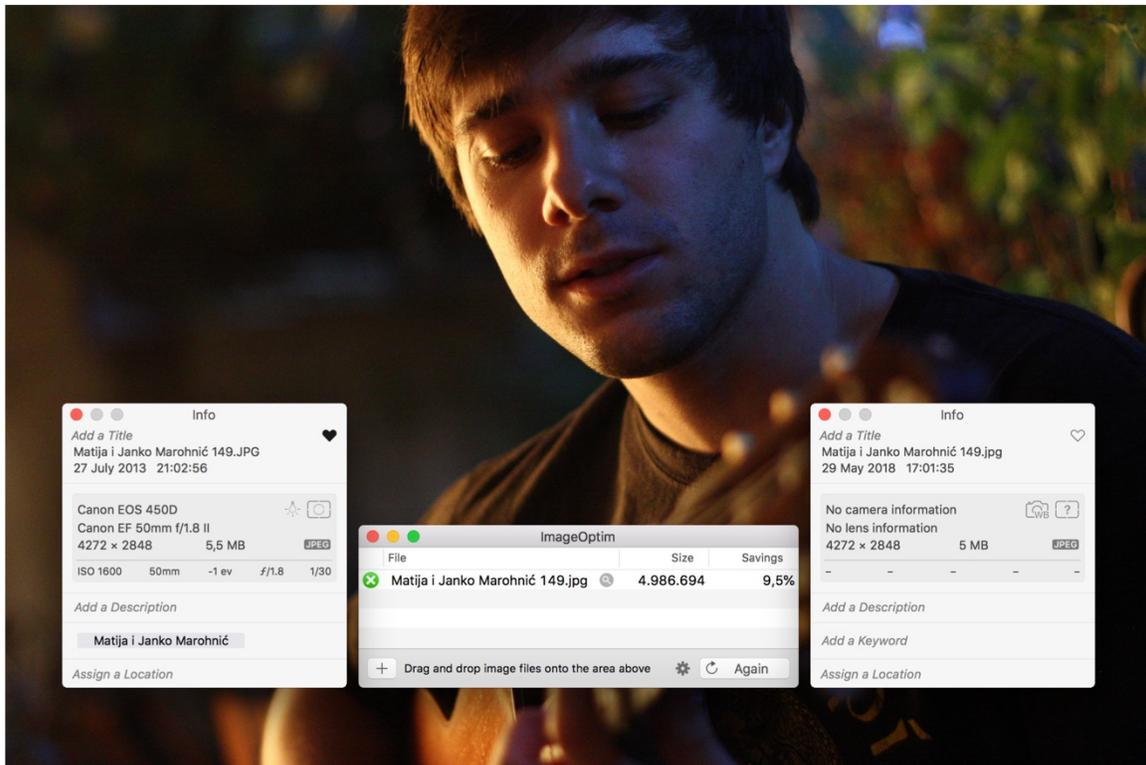
Slika 34: Greška u validaciji HTML-koda.

6.2. Kompresija slika

Osim informacija o prikazu, slike mogu sadržavati razne druge podatke. Tako na primjer, prilikom fotografiranja ne pohranjujemo samo piksele, već i informacije o

fotoaparatu, vremenu fotografiranja, lokaciji i sl. (Wikipedia contributors, 2018b). Međutim, ti nam podaci najčešće nisu potrebni, a čine sliku težom.

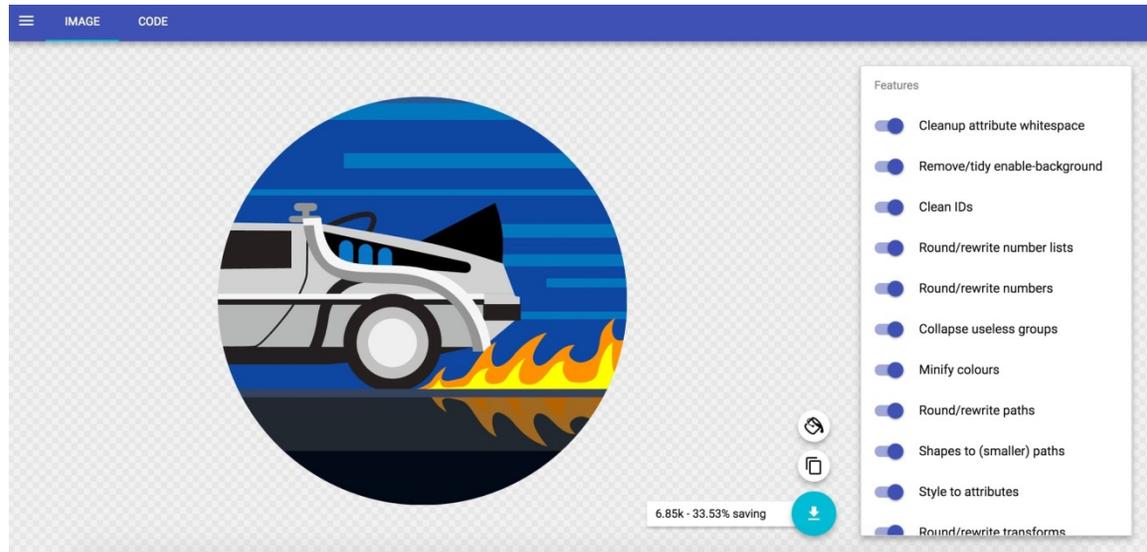
Aplikacija ImageOptim jedna je od aplikacija koje mogu ukloniti te metapodatke i dodatno komprimirati slike bez gubitka na kvaliteti (Osmani, 2018; Lesiński, 2018). JPEGmini popularna je aplikacija specijalizirana za format JPEG i može smanjiti težinu slike do 80 % bez gubitka na kvaliteti (Beamr Ltd., 2018). Slika 35 prikazuje kako smo pomoću aplikacije ImageOptim uklonili metapodatke slike i tako smanjili njezinu težinu za 500 kilobajta, tj. za gotovo 10 %.



Slika 35: Uklanjanje metapodataka slike pomoću aplikacije ImageOptim.

Za slike formata SVG možemo koristiti SVGO, vrlo sofisticiranu biblioteku napisanu u JavaScriptu koja uklanja nepotrebne dijelove koda kao što su komentari, neki atributi, prazni elementi itd. Jedan od kriterija optimizacije zove se „preciznost“. Na taj način SVGO pojednostavljuje sliku što može uzrokovati neprihvatljive promjene ako nismo oprezni. Kako ne bismo slučajno uništili sliku, možemo pomoću sučelja SVGOMG

vidjeti rezultat svake optimizacije koju napravi SVGO (vidi Slika 36) (Osmani, 2018; Belevich and Solntsev, 2018).



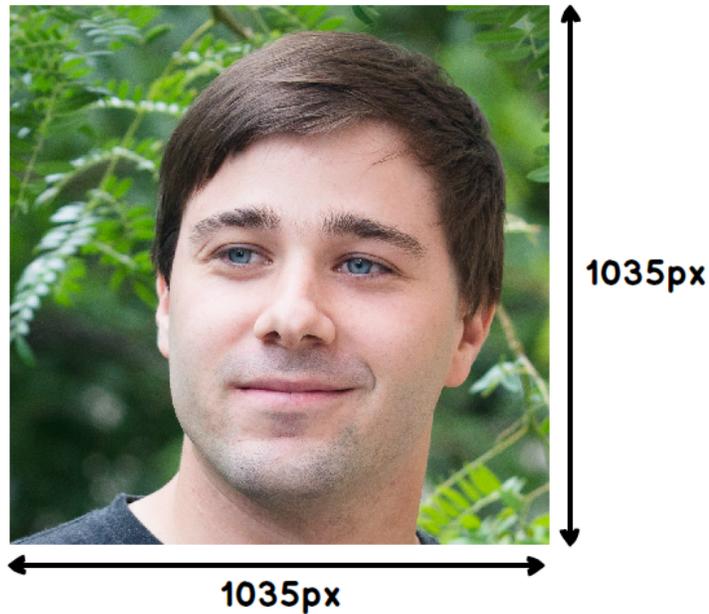
Slika 36: SVGOMG, grafičko sučelje za optimizaciju slika formata SVG.

6.3. Web-servisi za slike

Načini optimizacije koje sam dosad spomenuo samo su vrh ledenjaka. Postoji mnoštvo načina optimizacije kojima možemo pribjeći kako bismo smanjili težinu slika. Zato postoje razni *web*-servisi specijalizirani upravo za slike, npr. *imgix* i *Cloudinary*, koji na mnoge načine mogu obavljati optimizaciju umjesto nas (Osmani, 2018). *Cloudinary* sam spomenuo već prije, kad sam pričao o klijentskim naznakama pa ću ga nastaviti koristiti za primjere.

Kao što smo vidjeli dosad, *Cloudinary* nam omogućava da pomoću URL-a definiramo razne transformacije slike. Na taj si način možemo znatno ubrzati radni proces. Slika 37 prikazuje našu početnu sliku bez ikakvih transformacija, koju ćemo u daljnjim primjerima transformirati tako što ćemo izmjenjivati URL.

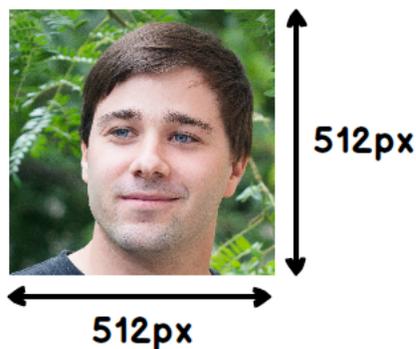
<https://res.cloudinary.com/silvenon/image/upload/v1510308691/avatar.jpg>



Slika 37: Dohvaćanje slike bez transformacija.

Slika 38 pokazuje primjer, gdje dodavanjem „c_scale,w_512“ u URL-u dohvaćamo sliku proporcionalno smanjenu na širinu od 512 piksela.

https://res.cloudinary.com/silvenon/image/upload/c_scale,w_512/v1510308691/avatar.jpg

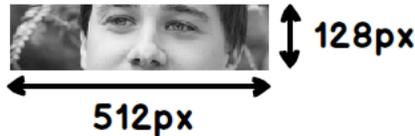


Slika 38: Dohvaćanje slike širine 512 piksela.

Slika 39 predstavlja znatno kompleksniji zahtjev, koji demonstrira fleksibilnost i snagu servisa. Pomoću njega sliku podrezujemo na dimenzije 512x128 piksela prema licu

(koje Cloudinary može automatski detektirati), primjenjujemo crno-bijeli filter i koristimo format WebP s kvalitetom od 65 %.

```
https://res.cloudinary.com/silvenon/image/upload/  
c_fill,e_grayscale,g_face,h_128,q_65,w_512  
/v1510308691/avatar.webp
```



Slika 39: Dohvaćanje crno-bijele slike podrezane na dimenzije 512x128 piksela u WebP formatu kvalitete 65 %.

Prednost korištenja *web*-servisa čini to što je dovoljno pohraniti slike u izvornoj veličini na njihovom poslužitelju. Stoga ne moramo održavati mnogo datoteka i brinuti se oko optimizacije memorije i dr.

6.4. Korak dalje

Optimiziranje postojećih slika vrlo je važan korak, ali u dugoročnom smislu dobro je razmišljati i o tome koje nam slike doista trebaju. Zaslونima veće gustoće trebaju veće slike kako bi izgledale oštro, što povećava težinu stranice više nego što bi mnogi od nas očekivali – dvostruko veća slika nije dvostruko teža, nego je oko tri do četiri puta teža (Rupert, 2012).

Nije dobro uklanjati slike koje imaju važnu ulogu na *web*-stranici u svrhu umanjivanja cijene održavanja, ali nije ni korisno imati slike koje ne pridonose sadržaju samo zato što možemo. Takve su odluke znatno subjektivnije i osjetljivije od optimizacije slika, ali dobro je imati na umu da više piksela sa sobom povlači više problema.

7. Usporedba novinskih portala

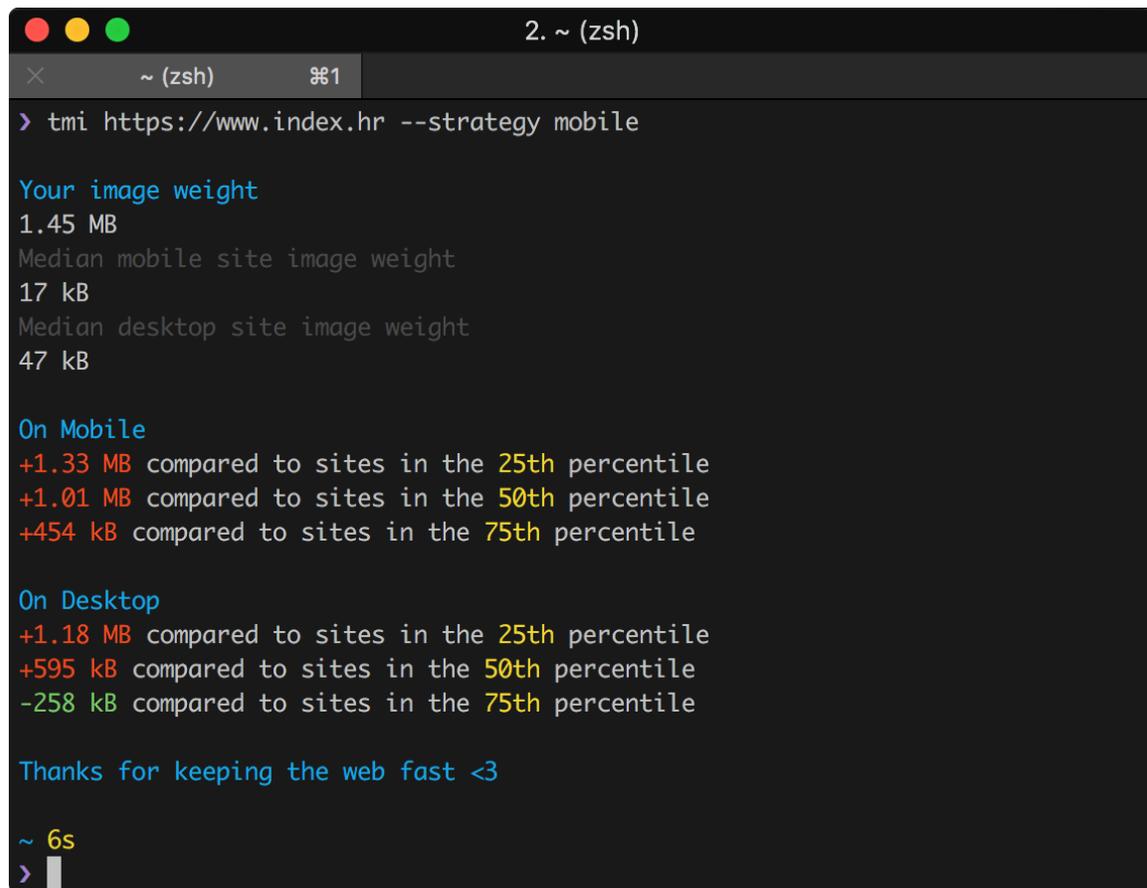
Kao praktični dio ovog rada odlučio sam usporediti slike na početnoj stranici najpopularnijih domaćih i stranih novinskih portala, tj. odrediti ukupnu težinu slika na stolnim i mobilnim uređajima i usporediti ih međusobno. S obzirom na to da su to stranice s velikim brojem posjeta, važno je da ne troše mnogo internetskog prometa te da optimiziraju slikovni sadržaj na mobilnim uređajima.

7.1. Metoda

Popularnost novinskih portala odredio sam pomoću Amazonovog *web*-servisa Alexa, koji, među ostalim uslugama, pruža pristup statistikama o internetskom prometu. Najpopularniji domaći novinski portali su Index.hr, Jutarnji.hr, 24sata, Net.hr i Večernji.hr (Alexa, 2018b), a najpopularniji strani novinski portali su CNN International, The New York Times, Google News, The Guardian i Yahoo News (Alexa, 2018a). Odabrao sam pet najpopularnijih portala kako bismo mogli primijetiti najvažnije informacije, a da nas ne smetu nijanse.

Slike sam analizirao koristeći biblioteku TMI⁹ (*Too Many Images*), koju je napisao Googleov zaposlenik Addy Osmani u JavaScriptu (Seymour, 2015). Biblioteka dobiva informacije o slikama na *web*-stranici koristeći Googleov *web*-servis PageSpeed Insights, a uz to dohvaća globalne statističke podatke o slikama sa servisa HTTP Archive, koja periodički analizira najpopularnije *web*-stranice i te podatke stavlja na raspolaganje svima. Biblioteka TMI zatim obrađuje te informacije i prikazuje ih u komandnoj liniji (Osmani, 2017). Slika 40 prikazuje primjer upotrebe te biblioteke u komandnoj liniji za analizu početne stranice novinskog portala Index.hr na mobilnim uređajima.

⁹ <https://github.com/addyosmani/tmi/tree/adedbc383aaadf415725445f5f3ac40e6dd60dbd>



```
2. ~ (zsh)
~ (zsh) ㉿1
> tmi https://www.index.hr --strategy mobile

Your image weight
1.45 MB
Median mobile site image weight
17 kB
Median desktop site image weight
47 kB

On Mobile
+1.33 MB compared to sites in the 25th percentile
+1.01 MB compared to sites in the 50th percentile
+454 kB compared to sites in the 75th percentile

On Desktop
+1.18 MB compared to sites in the 25th percentile
+595 kB compared to sites in the 50th percentile
-258 kB compared to sites in the 75th percentile

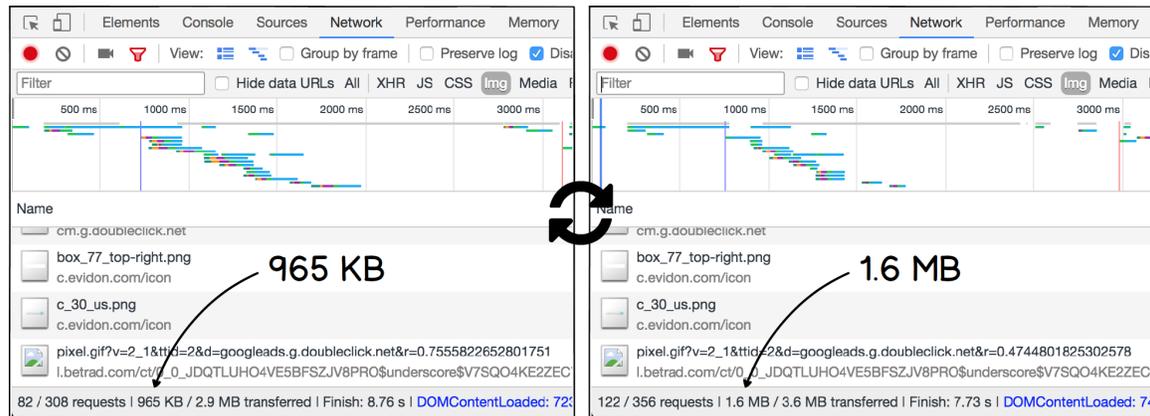
Thanks for keeping the web fast <3

~ 6s
> |
```

Slika 40: Primjer korištenja biblioteke TMI u komandnoj liniji.

Važno je imati na umu da težine slika izmjerene tom bibliotekom u većoj ili manjoj mjeri odstupaju od veličine u stvarnim *web*-preglednicima. Razlozi tome mogu biti mnogobrojni i teško ih je predvidjeti: možda *web*-stranica prikazuje drugačije slikovne oglase ovisno o lokaciji korisnika, možda je *web*-stranica učitala neke slike nakon što je *PageSpeed Insights* završio s mjerenjem itd.

Težina često varira i prilikom svakog osvježavanja stranice. Slika 41 ilustrira taj problem na primjeru hrvatskog novinskog portala tportal. To možemo vidjeti uz pomoć razvojnih alata kao što je Developer Tools u *web*-pregledniku Chrome.



Slika 41: Varijacija u težini slika na početnoj stranici portala nakon osvježavanja web-stranice.

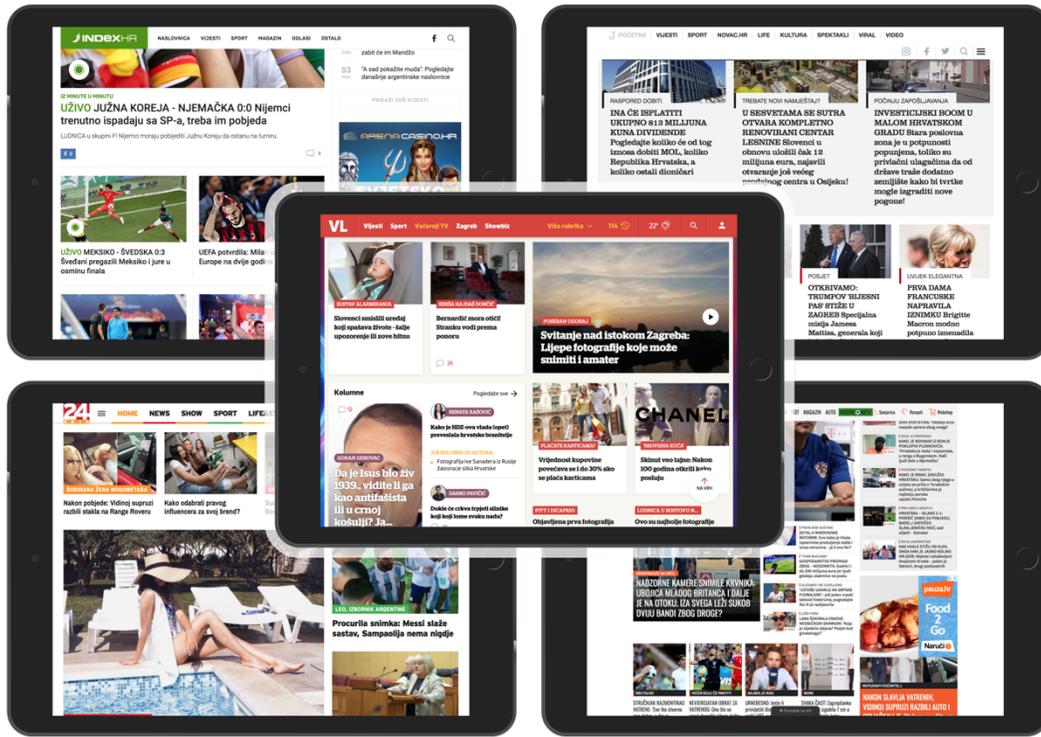
Teško je odrediti koji su točno razlozi varijacija u težini slika bez detaljne analize svakog novinskog portala. Otežavajuća je okolnost to što je JavaScript na *web*-stranicama obično komprimiran kako bi bio što lakši, zbog čega ga vrlo teško čitati.

Jedan od faktora za koje možemo biti sigurni da utječu na ukupnu težinu slika jesu oglasi jer su oni obično glavni izvor zarade. S obzirom na to da te slike dostavljaju servisi treće strane, nemamo mnogo kontrole nad optimizacijom. Najviše što možemo učiniti jest birati servise koji upotrebljavaju lakše slike.

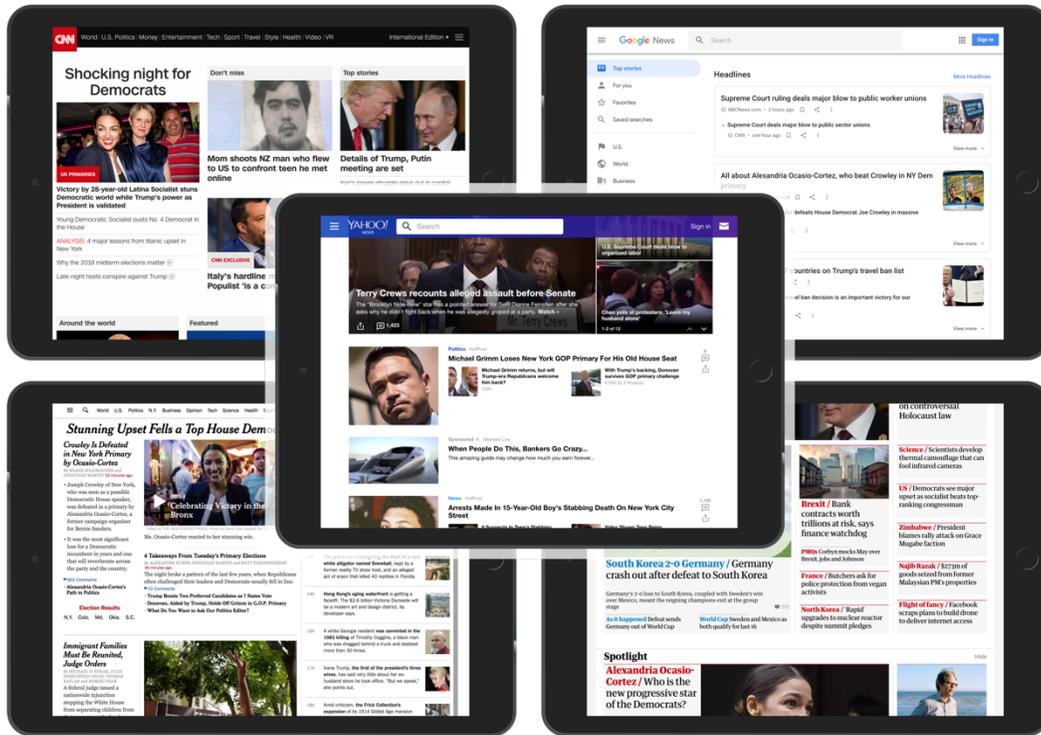
7.2. Rezultati

S obzirom na varijacije u težini slika, konačne sam brojeve odabrao tako da sam biblioteku TMI pokrenuo nekoliko puta na istim novinskim portalima te odabrao najviši broj, vođen pretpostavkom da su analize koje su rezultirale manjim brojevima završile prerano.

Podatke koje sam dobio pokazuju da domaći novinski portali imaju uglavnom teže slike na stolnim i mobilnim uređajima u odnosu na strane. Slika 42 i Slika 43 prikazuju početne stranice odabranih domaćih i stranih novinskih portala na tabletu. Možemo vidjeti da strani portali više štede na slikama, što je jedan od razloga zašto im je ukupna težina slika znatno manja. Još jedan od razloga, koji nije vidljiv na slikama, jest što se na mnogim stranim portalima dijelovi sadržaja učitavaju po potrebi, tj. kad se korisnik klizanjem kroz sadržaj približi dnu. To se u *web*-programiranju naziva beskonačnim učitavanjem.

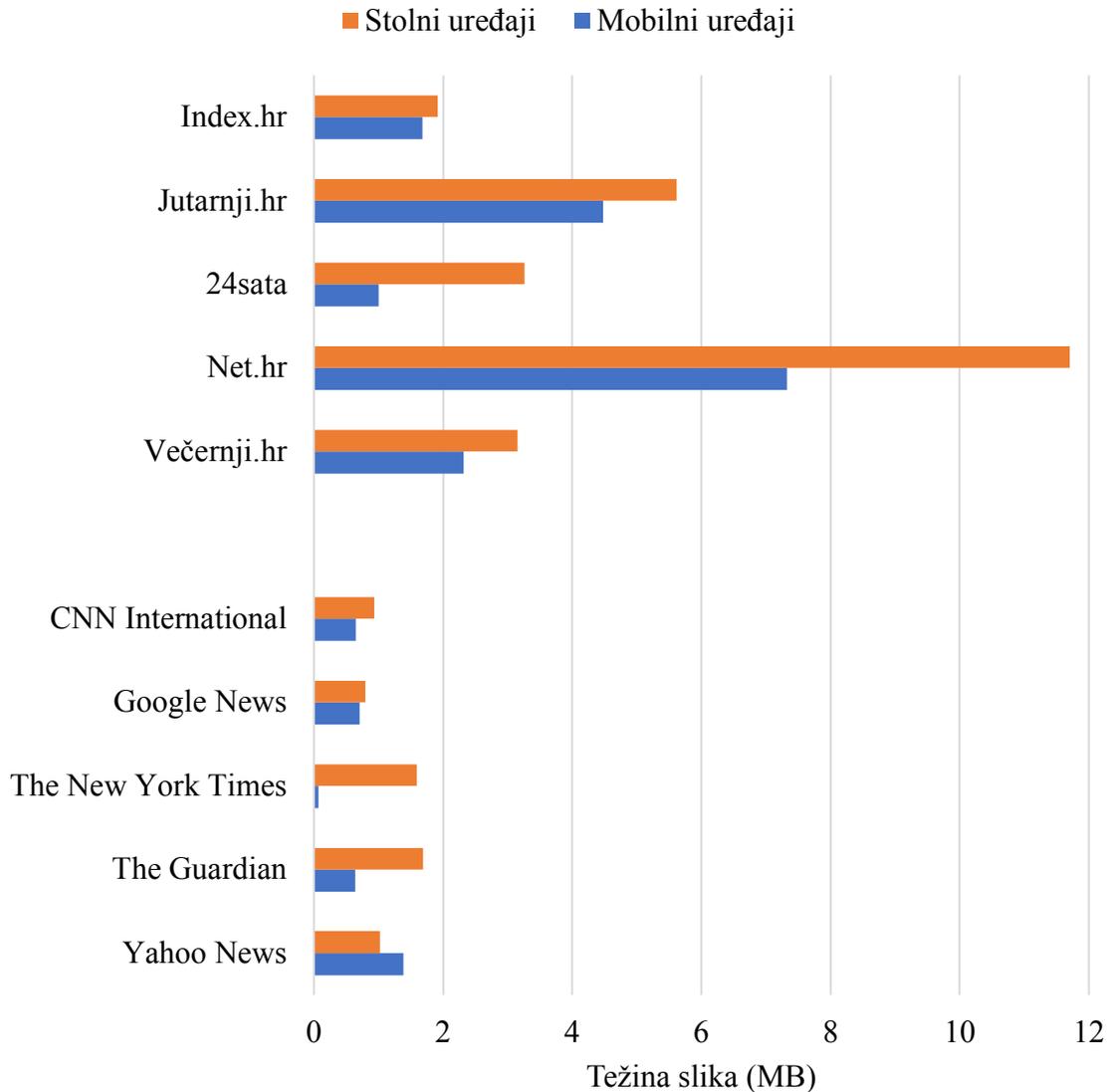


Slika 42: Početna stranica domaćih novinskih portala.



Slika 43: Početna stranica stranih novinskih portala.

Slika 44 predstavlja težinu slika na stolnim i mobilnim uređajima na popularnim novinskim portalima. Portali su poredani prema popularnosti, počevši od najposjećenijeg.



Slika 44: Težina slika na početnoj stranici popularnih novinskih portala (MB).

Možemo vidjeti da Net.hr ima daleko najteže slike i na stolnim i na mobilnim uređajima, s tim da je na stolnima težina veća od čak 11 megabajta! S druge strane, težina slika na najposjećenijem domaćem portalu Index.hr najmanja je u odnosu na ostale domaće portale na stolnim uređajima. To je vrlo dobro s obzirom na njegovu popularnost, ali na mobilnim uređajima slike nisu znatno optimizirane. Portal 24sata demonstrira izvrstan

stupanj optimizacije težine slika na mobilnim u odnosu na stolne uređaje, gdje je težina na mobilnim uređajima smanjena za čak 70 %. Zahvaljujući tome portal 24sata najlakši je od svih domaćih portala na mobilnim uređajima.

Prema servisu DotMetrics, Hrvati provode nešto više vremena na internetu na mobilnim uređajima i tabletima (Dotmetrics, 2018). Stoga bi se moglo očekivati da će se jednaka količina truda ulagati u optimizaciju iskustva na mobilnim uređajima kao i na stolnim. No, izgleda da će za to biti potrebno još malo vremena.

Strani novinski portali optimiziraju svoje slike znatno bolje od svih naših. Dva najpopularnija portala, CNN International i Google News, ujedno imaju najlakše slike na stolnim uređajima. Ovo je snažna indikacija da optimizacija slika ima utjecaj na popularnost *web*-stranice. Gotovo svaki od stranih portala na neki način optimizira slike na mobilnim uređajima, a jedini koji odstupa od toga je Yahoo News, čija je ukupna težina slika na mobilnim uređajima veća nego na stolnim. S druge strane, The New York Times najbolje optimizira slike na mobilnim uređajima. Težina slika iznosi samo 60-ak kilobajta, što je svega 4 % težine na stolnim uređajima.

8. Zaključak

Slike na *webu* u početku su bile luksuz. Postupno smo ih počeli upotrebljavati sve češće, ne samo kao dio sadržaja, nego i kao ukras. Danas ne možemo zamisliti *web* bez slika.

Nakon pojave prvih pametnih telefona nismo više mogli ignorirati činjenicu da mobilni uređaji moraju učitavati prevelike slike i nepotrebno trošiti podatkovni promet. Međutim, *web* je bio prepun slika, bilo je potrebno pronaći tehnike kako barem one velike prilagoditi za mobilne uređaje, tj. kako na manjim uređajima učitati manje slike.

U početku nije postojalo standardno rješenje za responzivne slike, što je znatno opterećivalo programere da se snalaze s tehnologijama koje imaju na raspolaganju. Bilo je potrebno na razne načine navesti *web*-preglednike da učitaju željene slike, a greške je nerijetko bilo teško uočiti. Rasla je potreba za standardnim rješenjem koje ćemo svi moći primjenjivati i koje će *web*-preglednici razumjeti.

Taj izazov preuzela je grupa RICG, koja je kroz nekoliko godina marljivog rada i iteracija došla do stabilnog rješenja u obliku specifikacije. To su rješenje postupno implementirali svi glavni *web*-preglednici, a u onima koji nemaju podršku moguće je to nadomjestiti u JavaScriptu pomoću biblioteke Picturefill. Velik broj dotadašnjih zaobilaznih rješenja stvorio je znatan jaz u rješavanju problema sa slikama pa je standardizacija tog rješenja bila ključni trenutak u povijesti responzivnih slika jer smo se napokon svi počeli koristiti istom sintaksom.

Osim osnovne potrebe dostavljanja manjih slika na manjim uređajima, ta nam je specifikacija otvorila i mnoga druga vrata, npr. likovno usmjeravanje slika, upotrebu novih slikovnih formata i dr. Sljedeći velik pomak na području responzivnih slika nazire se u obliku klijentskih HTTP-naznaka pomoću kojih ćemo moći znatno smanjiti količinu koda. Implementaciju klijentskih naznaka u *web*-preglednicima ohrabruju Google, koji ih je prvi implementirao u Chromeu, i Cloudinary, koji ih podržava u svom *web*-servisu.

Nakon što sam opisao trenutno stanje responzivnih slika na *webu*, htio sam vidjeti koliko dobro popularne *web*-stranice optimiziraju težinu slika, pogotovo na mobilnim uređajima. Usporedio sam po pet najpopularnijih domaćih i stranih novinskih portala i

otkrio sam da gotovo svaki barem malo optimizira slike za mobilne uređaje, ali i da su neki domaći portali znatno teži nego što bi mogli biti. Uostalom, cilj je novinskih portala prenijeti vijesti, oni nisu galerije slika.

Standardizacija responzivnih slika prebacila je velik dio tereta s programera na *web*-preglednike. To potiče natjecanje među *web*-preglednicima da implementiraju dodatne načine optimizacije. Chrome je to već demonstrirao tako da je izbjegao učitati manju sliku, a u budućnosti bi preglednici mogli birati slike prema brzini internetske veze, stanju baterije na mobilnom uređaju i sl.

To rasterećivanje omogućava nam da se usredotočimo na druge važne probleme u pogledu slika na *webu*, kao što su pristupačnost, automatizacija itd. Međutim, ponekad se trebamo zapitati kada nam slike doista trebaju, a kada ne. Više piksela uvijek povlači sa sobom više problema.

9. Literatura

1. Allsopp, John. A Dao of Web Design. A List Apart. 2000. URL: <http://alistapart.com/article/dao>. (05. 21. 2018.)
2. Andreessen, Marc. Proposed New Tag: IMG. WWW-Talk. 1993. URL: <http://1997.webhistory.org/www.lists/www-talk.1993q1/0182.html>. (06. 02. 2018.)
3. Atkins, Tab, Lea Verou, and Erika Etemad. CSS Images Module Level 4. W3C. 2018. URL: <https://drafts.csswg.org/css-images-4/#image-set-notation>. (05. 24. 2018.)
4. Atkins, Tab. Proposal for RespImg Syntax. 2014. URL: <http://tabatkins.github.io/specs/respimg/>. (06. 28. 2018.)
5. Beamr Ltd. JPEGmini - Your Photos on a Diet! JPEGmini. 2018. URL: <http://www.jpegmini.com/>. (06. 03. 2018.)
6. Belevich, Kir, and Lev Solntsev. Node.js Tool for Optimizing SVG Files. GitHub. 2018. URL: <https://github.com/svg/svgo>. (06. 03. 2018.)
7. Cáceres, Marcos, Mat Marquis, and Yoav Weiss. A Q&A on the Picture Element. A List Apart. 2014. URL: <http://alistapart.com/blog/post/picture-element-qa>. (06. 28. 2018.)
8. Cáceres, Marcos, Mat Marquis, Yoav Weiss, and David Newton. Use Cases and Requirements for Standardizing Responsive Images. W3C. 2013. URL: <http://usecases.responsiveimages.org/>. (05. 21. 2018.)
9. Cáceres, Marcos. Reasoning behind SrcN Replacing Srcset and Picture. Responsive Issues Community Group. 2013. URL: <https://www.w3.org/community/respimg/2013/10/14/reasoning-behind-srcn-replacing-srcset-and-picture/>. (06. 28. 2018.)
10. Chaffey, Dave. Mobile Marketing Statistics Compilation. Smart Insights. 2018. URL: <https://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/>. (05. 21. 2018.)

11. Chen, Pearl. Built-in Browser Support for Responsive Images. HTML5 Rocks. 2014. URL: <https://www.html5rocks.com/en/tutorials/responsive/picture-element/>. (05. 21. 2018.)
12. Colour Blindness. Colour Blind Awareness. 2018a. URL: <http://www.colourblindawareness.org/colour-blindness/>. (06. 04. 2018.)
13. Coyier, Chris. Practical SVG. New York, New York: Jeffrey Zeldman. 2016a.
14. Coyier, Chris. Responsive Images in CSS. CSS-Tricks. 2016b. URL: <https://css-tricks.com/responsive-images-css/>. (06. 28. 2018.)
15. Coyier, Chris. SVG `symbol` a Good Choice for Icons. CSS-Tricks. 2014. URL: <https://css-tricks.com/svg-symbol-good-choice-icons/>. (06. 02. 2018.)
16. Coyier, Chris. Which Responsive Images Solution Should You Use? CSS-Tricks. 2012. URL: <https://css-tricks.com/which-responsive-images-solution-should-you-use/>. (05. 21. 2018.)
17. Deveria, Alexis. Can I Use... Support Tables for HTML5, CSS3 Etc. 2018. URL: <https://caniuse.com/>. (05. 27. 2018.)
18. Dotmetrics. DotMetrics - Daily Fresh Big Audience Data. 2018. URL: <https://www.dotmetrics.net/>. (05. 29. 2018.)
19. Famous People with Colour Blindness. HRFnd. 2013. URL: <https://healthresearchfunding.org/famous-people-color-blindness/>. (06. 13. 2018.)
20. Frost, Brad. Device Fatigue. Brad Frost. 2012. URL: <http://bradfrost.com/blog/post/device-fatigue/>. (05. 21. 2018.)
21. Frost, Brad. The Many Faces of 'Adaptive Design.' Brad Frost. 2013. URL: <http://bradfrost.com/blog/post/the-many-faces-of-adaptive-design/>. (06. 14. 2018.)
22. Graham, Geoff. The Difference Between Responsive and Adaptive Design. CSS-Tricks. 2015. URL: <https://css-tricks.com/the-difference-between-responsive-and-adaptive-design/>. (06. 13. 2018.)
23. Green, Dan. Image Optimization Part 1: File Formats. LoadStorm. 2013. URL: <https://loadstorm.com/2013/05/image-optimization-part-1-file-formats/>. (06. 17. 2018.)

24. Grigsby, Jason. Don't Use <picture> (Most of the Time). Cloud Four. 2014. URL: <https://cloudfour.com/thinks/dont-use-picture-most-of-the-time/>. (05. 21. 2018.)
25. Grigsby, Jason. Responsive Images 101, Part 8: CSS Images. Cloud Four. 2015a. URL: <https://cloudfour.com/thinks/responsive-images-101-part-8-css-images/>. (05. 24. 2018.)
26. Grigsby, Jason. Responsive Images 101, Part 9: Image Breakpoints. Cloud Four. 2015b. URL: <https://cloudfour.com/thinks/responsive-images-101-part-9-image-breakpoints/>. (05. 24. 2018.)
27. Grigsby, Jason. Responsive Images 201: Client Hints. Cloud Four. 2016. URL: <https://cloudfour.com/thinks/responsive-images-201-client-hints/>. (05. 21. 2018.)
28. Grigsby, Jason. Sensible Jumps in Responsive Image File Sizes. Cloud Four. 2013. URL: <https://cloudfour.com/thinks/sensible-jumps-in-responsive-image-file-sizes/>. (05. 22. 2018.)
29. Hui Jing, Chen. We Need to Talk about Opera Mini. Chen Hui Jing. 2017. URL: <https://www.chenhuijing.com/blog/we-need-to-talk-about-opera-mini>. (05. 22. 2018.)
30. Jager, Tom. How to Optimize Charts For Color Blind Readers. Venngage. 2017. URL: <https://venngage.com/blog/color-blind-friendly-palette/>. (06. 04. 2018.)
31. Jehl, Scott. Respond: A Fast & Lightweight Polyfill for Min/Max-Width CSS3 Media Queries (for IE 6-8, and More). JavaScript. 2018[2011]. URL: <https://github.com/scottjehl/Respond>. (06. 18. 2018.)
32. Kadlec, Tim. Why We Need Responsive Images: Part Deux. Tim Kadlec. 2013b. URL: <https://timkadlec.com/2013/11/why-we-need-responsive-images-part-deux/>. (05. 21. 2018.)
33. Kadlec, Tim. Why We Need Responsive Images. Tim Kadlec. 2013a. URL: <https://timkadlec.com/2013/06/why-we-need-responsive-images/>. (05. 21. 2018.)
34. Kalbag, Laura. Accesssibility for Everyone. New York: Jeffrey Zeldman. 2017.
35. Kirtani, Saurabh. How to Build Responsive Images with Srcset. SitePoint. 2015. URL: <https://www.sitepoint.com/how-to-build-responsive-images-with-srcset/>. (05. 21. 2018.)

36. Lein, Adam Z. How Important Are Those Pixels Per Inch, Really? Pocketnow. 2012. URL: <http://pocketnow.com/2012/12/12/how-important-are-ppi>. (06. 18. 2018.)
37. Lesiński, Kornel. ImageOptim — Better Save for Web. ImageOptim. 2018. URL: <https://imageoptim.com/>. (06. 03. 2018.)
38. Making Charts and Maps Responsive. AmCharts. 2015. URL: <https://www.amcharts.com/kbase/making-charts-responsive/>. (06. 28. 2018.)
39. Marcotte, Ethan. Fluid Images. A List Apart. 2011. URL: <http://alistapart.com/article/fluid-images>. (05. 21. 2018.)
40. Marcotte, Ethan. Responsive Web Design. A List Apart. 2010. URL: <http://alistapart.com/article/responsive-web-design>. (05. 21. 2018.)
41. Marquis, Mat. So, You're Writing A Responsive Images Script. Filament Group Blog. 2014. URL: https://www.filamentgroup.com/lab/respimg_scripts.html. (05. 21. 2018.)
42. Massimino, Pascal. @silvenon WebP Was Created Just after the WebM Format. And Where "M" in WebM Was for "Movie" (or, Moving Pictures), the "P" in WebP Was for "Pictures" or "Photos". We Never Quite Decided Which One :). Tweet. Twitter. 2018. URL: <https://twitter.com/PascalMassimino/status/1008831648513777664>. (06. 19. 2018.)
43. Migliorisi, Heather. Accessible SVGs. CSS-Tricks. 2016. URL: <https://css-tricks.com/accessible-svgs/>. (06. 02. 2018.)
44. Osmani, Addy. Automating Image Optimization. Google Developers. 2018. URL: <https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/automating-image-optimization/>. (05. 29. 2018.)
45. Osmani, Addy. TMI (Too Many Images) - Discover Your Image Weight on the Web. GitHub. 2017. URL: <https://github.com/addyosmani/tmi>. (06. 01. 2018.)
46. Portis, Eric. Automatic Responsive Images with Client Hints. Cloudinary Blog. 2016. URL:

- https://cloudinary.com/blog/automatic_responsive_images_with_client_hints. (05. 21. 2018.)
47. Portis, Eric. Srcset and Sizes. Ericportis.Com. 2014. URL: <http://ericportis.com/posts/2014/srcset-sizes/>. (05. 21. 2018.)
48. Quick Tip: Using Alt Text Properly. The Accessibility Project. 2017. URL: <https://a11yproject.com/posts/alt-text/>. (05. 21. 2018.)
49. Responsive Issues Community Group. Community and Business Groups. 2018. URL: <https://www.w3.org/community/respimg/>. (06. 19. 2018.)
50. Rupert, Dave. Mo' Pixels Mo' Problems. A List Apart. 2012. URL: <http://alistapart.com/article/mo-pixels-mo-problems>. (05. 21. 2018.)
51. Seymour, Ben. Practical Responsive Images. Cardiff: Five Simple Steps. 2015.
52. State of Images. HTTP Archive. 2018. URL: https://httparchive.org/reports/state-of-images?start=2017_05_01&end=2018_05_01#bytesImg. (05. 23. 2018.)
53. Support for Older Versions of Internet Explorer Ended. Microsoft. 2018. URL: <https://www.microsoft.com/en-us/windowsforbusiness/end-of-ie-support>. (06. 28. 2018.)
54. Tanaka, Sanette, Winston Hearn, and Kelsey Scherer. Accessibility Matters—and Here's What We're Doing about It. Vox Product. 2016. URL: <https://product.voxmedia.com/2016/5/11/11612516/accessibility-matters-and-heres-what-were-doing-about-it>. (05. 21. 2018.)
55. Tarnoff, Greg. A Primer To Visual Impairment. The A11Y Project. 2013. URL: <https://a11yproject.com/posts/understanding-visual-impairment/>. (05. 21. 2018.)
56. Tikz Pgf - Pie Chart Slice with Pattern Background. TeX - LaTeX Stack Exchange. 2014. URL: <https://tex.stackexchange.com/questions/162961/pie-chart-slice-with-pattern-background>. (06. 18. 2018.)
57. Top Sites by Category: News. Alexa. 2018a. URL: <https://www.alexa.com/topsites/category/Top/News>. (06. 01. 2018.)
58. Top Sites in Croatia. Alexa. 2018b. URL: <https://www.alexa.com/topsites/countries/HR>. (05. 30. 2018.)

59. Types of Colour Blindness. Colour Blind Awareness. 2018b. URL:
<http://www.colourblindawareness.org/colour-blindness/types-of-colour-blindness/>.
(06. 04. 2018.)
60. Types of Motor Disabilities. WebAIM. 2012. URL:
<https://webaim.org/articles/motor/motordisabilities>. (06. 11. 2018.)
61. Vox. Scientists Are Remaking Mosquitoes to Fight Malaria. 2018. URL:
<https://youtu.be/P0HPHUzsHbI>. (05. 31. 2018.)
62. W3C Team. First Draft of Picture Element Proposal Published by Responsive Images Community Group. Responsive Issues Community Group. 2012. URL:
<https://www.w3.org/community/respimg/2012/07/02/first-draft-of-picture-element-proposal-published-by-responsive-images-community-group/>. (06. 19. 2018.)
63. W3C. The W3C Markup Validation Service. W3C. 2018. URL:
<https://validator.w3.org/>. (06. 03. 2018.)
64. Wikipedia contributors. Compression Artifact. Wikipedia, The Free Encyclopedia. 2018h. URL:
https://en.wikipedia.org/w/index.php?title=Compression_artifact&oldid=844527481. (06. 17. 2018.)
65. Wikipedia contributors. Exif. Wikipedia, The Free Encyclopedia. 2018b. URL:
<https://en.wikipedia.org/w/index.php?title=Exif&oldid=836272129>. (06. 03. 2018.)
66. Wikipedia contributors. Guernica (Picasso). Wikipedia, The Free Encyclopedia. 2018d. URL:
[https://en.wikipedia.org/w/index.php?title=Guernica_\(Picasso\)&oldid=842059838](https://en.wikipedia.org/w/index.php?title=Guernica_(Picasso)&oldid=842059838).
(06. 02. 2018.)
67. Wikipedia contributors. Halftone. Wikipedia, The Free Encyclopedia. 2018e. URL: <https://en.wikipedia.org/w/index.php?title=Halftone&oldid=842857039>.
(06. 17. 2018.)
68. Wikipedia contributors. Internet Explorer Version History. Wikipedia, The Free Encyclopedia. 2018f. URL:

- https://en.wikipedia.org/w/index.php?title=Internet_Explorer_version_history&oldid=842977751. (06. 18. 2018.)
69. Wikipedia contributors. iPhone (1st Generation). Wikipedia, The Free Encyclopedia. 2018j. URL:
[https://en.wikipedia.org/w/index.php?title=IPhone_\(1st_generation\)&oldid=846400520](https://en.wikipedia.org/w/index.php?title=IPhone_(1st_generation)&oldid=846400520). (06. 18. 2018.)
70. Wikipedia contributors. iPhone 4. Wikipedia, The Free Encyclopedia. 2018c. URL: https://en.wikipedia.org/w/index.php?title=IPhone_4&oldid=841317121. (05. 25. 2018.)
71. Wikipedia contributors. Mac OS X Tiger. Wikipedia, The Free Encyclopedia. 2018l. URL:
https://en.wikipedia.org/w/index.php?title=Mac_OS_X_Tiger&oldid=846683696. (06. 28. 2018.)
72. Wikipedia contributors. Microsoft Edge. Wikipedia, The Free Encyclopedia. 2018k. URL:
https://en.wikipedia.org/w/index.php?title=Microsoft_Edge&oldid=846446761. (06. 28. 2018.)
73. Wikipedia contributors. Pixel Density. Wikipedia, The Free Encyclopedia. 2018i. URL:
https://en.wikipedia.org/w/index.php?title=Pixel_density&oldid=846156638. (06. 18. 2018.)
74. Wikipedia contributors. Reprography. Wikipedia, The Free Encyclopedia. 2018a. URL: <https://en.wikipedia.org/w/index.php?title=Reprography&oldid=828206530>. (06. 17. 2018.)
75. Wikipedia contributors. World Wide Web. Wikipedia, The Free Encyclopedia. 2018g. URL:
https://en.wikipedia.org/w/index.php?title=World_Wide_Web&oldid=843817554. (06. 01. 2018.)

10. Popis slika

| | |
|--|----|
| Slika 1: Semafor na pješačkom prijelazu..... | 5 |
| Slika 2: Disketa..... | 5 |
| Slika 3: Stopa smrtnosti ljudi oboljelih od malarije u svijetu (preuzeto iz Vox, 2018)..... | 6 |
| Slika 4: Dijete u Africi oboljelo od malarije (preuzeto iz Vox, 2018). | 7 |
| Slika 5: Guernica, djelo španjolskog umjetnika Pabla Picassa (preuzeto iz Wikipedia contributors, 2018c)..... | 8 |
| Slika 6: Dio spektra uređaja kojima danas pregledavamo web-stranice (preuzeto iz Frost, 2012). | 9 |
| Slika 7: Prije i nakon primjene CSS pravila „max-width: 100%“ na sliku (prilagođeno od Marcotte, 2011). | 11 |
| Slika 8: Dimenzije okvira prikaza web-preglednika..... | 13 |
| Slika 9: Različite veličine okvira prikaza (preuzeto iz Cáceres et al., 2013)..... | 13 |
| Slika 10: Primjer web-stranice koja mijenja izgled ovisno o širini okvira prikaza. | 15 |
| Slika 11: Razlika između zaslona manje (lijevo) i veće (desno) gustoće (preuzeto iz Lein, 2012). | 16 |
| Slika 12: Gustoća zaslona različitih uređaja (preuzeto iz Cáceres et al., 2013)..... | 17 |
| Slika 13: Podrezivanje slike kako bi pas bio prepoznatljiv na manjim zaslonima (preuzeto iz Cáceres et al., 2013)..... | 17 |
| Slika 14: Likovno usmjeravanje promjenom kompozicije elemenata slike (prilagođeno od Marcotte, 2010). | 18 |
| Slika 15: Likovno usmjeravanje vizualizacije podataka (preuzeto iz AmCharts, 2015).. | 18 |
| Slika 16: Primjer ispisa polutonskom tehnikom (prilagođeno od Cáceres et al., 2013).. | 19 |
| Slika 17: Dva primjera istog dijagrama, jedan je u boji, a drugi crno-bijeli (preuzeto iz Cáceres et al., 2013)..... | 20 |
| Slika 18: Usporedba slikovnih formata JPEG i PNG prilikom kompresije pravilne slike (preuzeto iz Green, 2013)..... | 20 |
| Slika 19: Ista ikona prikazana na dva različita načina pomoću SVG koda..... | 22 |
| Slika 20: Postavke Opere Mini na operacijskom sustavu Android (preuzeto iz Hui Jing, 2017). | 23 |
| Slika 21: Ilustracija kako nas atributi „srcset“ i „sizes“ mogu lišiti kompliciranog računanja. | 27 |
| Slika 22: Primjer kada funkcija „calc“ može biti korisna..... | 28 |
| Slika 23: Prilagođavanje loga prema gustoći zaslona..... | 29 |
| Slika 24: Mačak u različitim položajima ovisno o širini zaslona (prilagođeno od Chen, 2014). | 32 |
| Slika 25: Primjer metapodataka koje klijent može poslati poslužitelju..... | 35 |
| Slika 26: Primjer iste web-stranice na mobitelu s manjim i većim gumbom..... | 38 |
| Slika 27: Otežavajući konteksti, situacije i razdoblja u životu (prilagođeno od Tanaka, Hearn, and Scherer, 2016)..... | 39 |
| Slika 28: Primjer djelomičnog (lijevo) i potpunog (desno) povećanja zaslona. | 40 |
| Slika 29: Oblačić sa sadržajem elementa „title“..... | 42 |

| | |
|--|----|
| Slika 30: Normalni vid (lijevo) i nemogućnost raspoznavanja crvenih tonova (desno) (prilagođeno od Colour Blind Awareness, 2018b)..... | 43 |
| Slika 31: Primjer razumljivog crno-bijelog grafa..... | 44 |
| Slika 32: Označavanje setova podataka u grafu pomoću uzoraka (prilagođeno od TeX - LaTeX Stack Exchange, 2014)..... | 44 |
| Slika 33: Razlika kako isti graf vide ljudi s normalnim vidom (lijevo) i oni s umjerenim crveno-zelenim daltonizmom (preuzeto iz Jager, 2017)..... | 45 |
| Slika 34: Greška u validaciji HTML-koda..... | 46 |
| Slika 35: Uklanjanje metapodataka slike pomoću aplikacije ImageOptim..... | 47 |
| Slika 36: SVGOMG, grafičko sučelje za optimizaciju slika formata SVG..... | 48 |
| Slika 37: Dohvaćanje slike bez transformacija..... | 49 |
| Slika 38: Dohvaćanje slike širine 512 piksela..... | 49 |
| Slika 39: Dohvaćanje crno-bijele slike podrezane na dimenzije 512x128 piksela u WebP formatu kvalitete 65 %..... | 50 |
| Slika 40: Primjer korištenja biblioteke TMI u komandnoj liniji..... | 52 |
| Slika 41: Varijacija u težini slika na početnoj stranici tportal nakon osvježavanja web-stranice..... | 53 |
| Slika 42: Početna stranica domaćih novinskih portala..... | 54 |
| Slika 43: Početna stranica stranih novinskih portala..... | 54 |
| Slika 44: Težina slika na početnoj stranici popularnih novinskih portala (MB)..... | 55 |

11. Popis priloga

11.1. Terminologija

Moj popis predloženih prijevoda manje poznatih termina u hrvatskom jeziku:

| | |
|--------------------------|-------------------------|
| <i>viewport</i> | okvir prikaza |
| <i>media query</i> | medijski upit |
| <i>art direction</i> | likovno usmjeravanje |
| <i>breakpoint</i> | točka loma |
| <i>screen density</i> | gustoća zaslona |
| <i>HTTP client hints</i> | klijentske HTTP naznake |