

Sveučilište J. J. Strossmayera u Osijeku  
Filozofski fakultet  
Preddiplomski studij Informatologije

Iva Peranović

**CSS rešetke: implementacija vizualnih elemenata koristeći stilski jezik**

Završni rad

izv.prof.dr.sc. Boris Badurina

dr. sc. Tomislav Jakopec, sumentor

Osijek, 2018

Sveučilište J. J. Strossmayera u Osijeku

Filozofski fakultet

Preddiplomski studij Informatologije

Iva Peranović

**CSS rešetke: implementacija vizualnih elemenata koristeći stilski jezik**

Završni rad

Društvene znanosti, informacijske i komunikacijske znanosti, informacijski sustavi i  
informatologija

izv.prof.dr.sc. Boris Badurina

dr. sc. Tomislav Jakopec, sumentor

Osijek, 2018

## Sažetak

Glavni cilj ovoga rada je smjestiti tehnologiju CSS rešetki u vremenski kontekst računalnih tehnologija posljednja dva desetljeća, objasniti detalje rada navedene tehnologije kao i temeljne tehnologije na kojima je izgrađena i demonstrirati jedinstvene prednosti koje tehnologija CSS rešetki pruža. Od svoga postanka, CSS nije posjedovao jednoznačne metode brzog i efikasnog pozicioniranja velikog broja vizualnih elemenata na temelju vizualnih odrednica elemenata čiji su oni sastavni dijelovi. Razvojni inženjeri i mrežni dizajneri koristili su ugrađene odrednice HTML-a kako bi postigli specifičan vizualni razmještaj elemenata prezentacije, a upotreba opisanih ugrađenih odrednica HTML-a otežavala je razdvajanje razvoja sadržaja od prezentacije. Spomenuti profesionalci često su se nalazili u situaciji gdje je kompromis kvalitete prezentacije i kvalitete sadržaja bio neizbježan. Razvoj CSS rešetki kao proširenja CSS-a tijekom ovog desetljeća predstavio je novi i jedinstveni pristup realizaciji dvodimenzionalnih razmještaja elemenata unutar HTML dokumenta u obliku rešetke. U ovome radu bit će pobliže istraženi i demonstrirani neki od mnogih oblika razmještaja dvodimenzionalnih elemenata unutar nadležnih elemenata HTML dokumenta. Također, bit će navedena i sama tehnička osnova ove tehnologije, njena raspoloživost i podržanost u kontekstu različitih pretraživača Interneta kao i opisan mehanizam razlomačkih jedinica koje također pretpostavlja CSS Grid i objašnjena njihova relacija u odnosu na tradicionalne modele dimenzija elemenata.

**KLJUČNE RIJEČI:** HTML, CSS, CSS Grid, mrežni pretraživač, mrežni dizajn

# Sadržaj

1. Uvod.....	1
2. Cascading Style Sheet (CSS).....	1
2.1 HTML.....	2
2.2 CSS i mrežni dizajn.....	3
3. CSS rešetke.....	5
3.2 <i>Fractional units</i> – razlomačke jedinice.....	13
3.3 Primjeri nekih od atributa propisanih specifikacijom CSS rešetaka.....	15
4. Zaključak.....	23
5. Literatura.....	24

# 1. Uvod

S vremenom, razvojem vizualnih odrednica HTML-a, kao i mrežnih pretraživača, postalo je jasno kako tadašnje mrežne stranice postupno evoluiraju iz jednostavnih, mirujućih i nepromjenjivih razmještaja vizualnih elemenata u itekako živa, kvalitetna i informativna grafička remek-djela. Također, postalo je jasno kako dotadašnje metode postizanja određenog razmještaja elemenata na raspoloživoj vidljivoj površini mrežne stranice, prvenstveno svojstva imenom *float*, više nisu optimalne metode za realiziranje željenog dizajna. Postići realizaciju sve specifičnijih zahtjeva na dizajn značilo je koristiti ugrađene odrednice HTML-a na sve kreativnije načine. Opisani načini često su uključivali kreativno baratanje HTML tablicama, piksel-preciznim izračunima dimenzija i JavaScript-om, s ciljem inovativnog razmještanja elemenata. Nijedna od navedenih tehnologija nije točno osmišljena za ovu namjenu, a nuspojava te okolnosti je neočekivan rad mrežne stranice ukoliko raspoloživ prostor na zaslonu postane neočekivano preuzak, prenizak ili čak i obrnuto. Postala je jasna potreba za brzim, jednostavnim i efikasnim sustavom prostornog razmještaja (engl. *layout*) elemenata na raspoloživoj površini.

CSS rešetke predstavljaju jedinstveno utaženje ove potrebe. CSS rešetke nude brz i predvidiv način podjele raspoložive vidljive površine HTML dokumenta radi potreba razmještaja elemenata, bez kreiranja tablica kao HTML elemenata ili ikakvih sličnih elemenata. CSS rešetke nude kontrolu nad dvodimenzionalnim razmještajem prema modelu tablica, ali ne kreiraju nikakve tablice. Cijela funkcionalnost CSS rešetki ostaje zaokružena unutar granica upravljanja prezentacijom HTML dokumenta. Mrežni dizajneri mogu koristiti CSS rešetke za precizno i kvalitetno upravljanje rasporedom elemenata.

## 2. Cascading Style Sheet (CSS)

CSS (“Cascading style sheet”) računalna je tehnologija za upravljanje prezentacijom elemenata mrežnih stranica i dokumenata koji su stvoreni koristeći označiteljski programski jezik (engl. *markup language*). Takav jezik elemente nekog dokumenta opisuje slijednim navođenjem svakog pojedinog

elementa preko posebno rezerviranih i unaprijed poznatih oznaka (engl. *tags*), a navedene oznake precizno su definirane pripadajućim tehničkim specifikacijama i standardima. U praksi, u posljednja dva desetljeća, ulogu jezika označavanja mrežnih stranica uvijek preuzima jezik HTML, a CSS predstavlja izuzetno moćan sustav upravljanja prezentacijom svih dokumenata napisanih prema pravilima jezika HTML. HTML, CSS i JavaScript predstavljaju trojac tehnologija koje čine osnovu vizualnog aspekta Interneta kakav znamo danas.

## 2.1 HTML

Prije svega, HTML (“Hyper-text Markup Language”) je tehnologija za prijenos semantike, odnosno značaja i konteksta. HTML predstavlja standardni način definiranja strukturiranih dokumenata namijenjenih za prijenos Internetom. Program koji može razumjeti i prikazati značaj definiran pomoću HTML-a zove se mrežni pretraživač (engl. “web browser”) i općenito se smatra osnovnim programom za pretraživanje i vizualizaciju Interneta. Internet predstavlja implementaciju mrežnog modela podataka, a svaki pojedini stalni i statički čvor tog mrežnog modela predstavlja jedan dokument. U praksi, u posljednjih dva desetljeća, opisani je dokument u gotovo svim slučajevima – HTML dokument. Mrežni pretraživač je u tom kontekstu program koji korisniku omogućava da putuje mrežnim modelom podataka pojedinačnim vizualiziranjem svakog pojedinog čvora opisanog mrežnog modela podataka i ostavljanjem slobode korisniku da odabere koju će mrežnu stranicu iduće prikazati. Razvojni inženjeri internetskog (u daljnjem tekstu, “mrežnog”) sadržaja brinu o veoma bitnom preduvjetu prikazivanja interaktivnog internetskog sadržaja – međusobnoj povezanosti različitih mrežnih resursa. Čvor nekakvog mrežnog modela podataka efektivno je nepostojeći ako je nevidljiv – ako preko ostalih čvorova nije moguće do njega doći. U tome smislu, razvojni inženjeri mrežnog sadržaja brinu da svaki bitan čvor neke mreže nekamo vodi, što je preduvjet da navedeni sadržaj bude opisan riječju “interaktivan”. Drugim riječima, kada dizajniraju mrežni sadržaj, razvojni inženjeri koriste standardizirane elemente HTML jezika kako bi korisniku omogućili korištenje URL-ova (engl. “Uniform Resource Locator”) – svojevrsnih veza na druge mrežne resurse. Uloga mrežnog pretraživača je da, razumjevši odnos ishodište – odredište, korisniku dostavi idući mrežni resurs kojeg on zatraži i u tom smislu se može opisati riječju “pretraživač”. Internet, iz perspektive korisnika mrežnog pretraživača, jest vrlo velika mreža dokumenata napisanih u HTML-u.

HTML je prvobitno bio asociran uz jezik SGML 1991. godine. Jezik SGML („Standard Generalized Markup Language“), definiran dokumentom ISO 8879:1986 jezik je za definiranje označiteljskih jezika, a HTML predstavlja jednu „primjenu“ SGML-a.<sup>1</sup> Ovaj HTML kao podvrsta SGML-a predstavlja HTML verziju 1.0. Zatim, tehnički dokument RFC 1866<sup>2</sup> koji biva objavljen 24.11.1995. godine od strane CERN-ovog fizičara Tima Berners-Leeja opisuje tadašnju verziju HTML-a 2.0. Tim Berners-Lee je također osnivač udruge World Wide Web Consortium (W3C) koja predstavlja glavno tijelo standardizacije World Wide Web tehnologija – primarno HTML-a. Standardizacija HTML-a i WWW-a općenito predstavila je osnovu za nastanak mrežnih pretraživača – bez jedinstvene interpretacije značaja prenesenog jezikom HTML nije moguće razvijati mrežne stranice na način da oni tvore smislene i korisne cjeline.

Kroz godine, HTML je doživio mnogo dorada i novih tehničkih inovacija, opisa, standarda i definicija. Trenutna verzija HTML-a jest HTML5 i obuhvaća XHTML standard.

## **2.2 CSS i mrežni dizajn**

CSS je jezik za opis prezentacije mrežnih stranica koja uključuje boje, razmještaj i prikaz teksta. Omogućava pojedincu prilagodbu prezentacije različitim vrstama uređaja, poput velikim zaslonima, malim zaslonima ili printerima.<sup>3</sup> U praksi, CSS se koristi za upravljanje prezentacijom HTML dokumenata. Najvažnija svrha CSS-a jest odvajanje pisanja sadržaja i prezentacije tog sadržaja. Ukoliko je sadržaj napisan prema nekim dogovorenim pravilima, CSS će se moći iskoristiti kao izuzetno prilagodljiva tehnologija koju je vrlo lako proširiti i nanovo iskoristiti u svrhu upravljanja prezentacijom više različitih HTML dokumenata, bez pisanja posebnih, odvojenih CSS dokumenata za svaki pojedini HTML dokument. Navedeni dogovori prilikom pisanja mrežnih dokumenata nisu standardizirani, ali čine velik dio onoga što zovemo inženjerskom vještinom razvoja programa. Vješti programeri su oni programeri koji poznaju velik broj različitih kvalitetnih, kako kratkoročno tako i dugoročno, postupaka prilikom programiranja koji osiguravaju dugovječnost, kvalitetu, efikasnost i efektivnost programa kojeg stvaraju. Pojava CSS-a omogućila je jedan iskorak u smjeru povećanja svega navedenog, a prvenstveno na temelju koncepata modularnosti, ponovne iskoristivosti i

---

<sup>1</sup> A brief SGML tutorial. URL: <https://www.w3.org/TR/WD-html40-970708/intro/sgmltut.html>

<sup>2</sup> Usp. Berners-Lee, T.; Connolly, D. Hypertext Markup Language - 2.0, 1995. URL: <https://tools.ietf.org/html/rfc1866>

<sup>3</sup> HTML & CSS. URL: <https://www.w3.org/standards/webdesign/htmlcss>

skalabilnosti mrežnog sadržaja stvorenog pomoću HTML-a i CSS-a. S vremenom, CSS kroz rastući broj slučajeva korištenja na Internetu zauzeo je svoje mjesto među osnovnim tehnologijama koje napajaju Internet. CSS je omogućio razvoj dizajna mrežnih stranica na svakim trenutkom većoj skali i s vremenom, kroz mnoštvo dorada, implementacija novih funkcionalnosti, standardizacija i proširenja postao veoma moćna tehnologija za upravljanje svim aspektima prikaza mrežnih stranica. CSS je svojim napredovanjem u funkcionalnosti na neki način stvorio industriju mrežnog dizajna. Razvoj sadržaja na takav je način odvojen od razvoja prezentacije da današnji mrežni dizajneri ne moraju mnogo baratati razvojem sadržaja, a da i dalje mogu razvijati programe čiji su izgled i interaktivnost dosad neviđeni.<sup>4</sup>

Moć kvalitetno razvijenog CSS-a najviše je bila izražena tijekom posljednjeg desetljeća kada su uređaji korišteni za pretraživanje Interneta naglo počeli imati čudne, nestandardizirane i neuobičajene razlučivosti ekrana. Najčešće, radilo se o pametnim telefonima koji se većinu vremena u ruci drže u portretnoj orijentaciji. S obzirom da su ljudi veoma utjecajni od strane vizualnog, ono što izgleda dobro i primamljivo na ekranu računala (u *pejzažnoj* orijentaciji), ne izgleda nužno ni blizu toliko dobro i primamljivo na ekranu pametnog telefona – koji ponekad također ima i daleko manju razlučivost. Prava korisnost CSS-a došla je do izražaja kada je kolektiv mrežnih dizajnera ovog svijeta u svojoj svakodnevnici vrlo jednostavno iskoristio funkcionalnosti CSS-a kako bi ovaj nagli skok u neodređenosti osobina ekrana uređaja na kojem se pretražuje Internet lako “pripitomio” i za svaki pojedini slučaj opisanog kaosa jednim potezom pružio ne optimalan, ali dovoljno dobar kompromis. Naime, s vremenom CSS je omogućio dizajneru da prezentaciju mrežnog dokumenta realizira uvjetno - da se elementi međusobno pozicioniraju i poprime neke određene dimenzije i ostale parametre ovisno o osobinama ekrana uređaja. Granice ovih osobina zapravo ne moraju biti jasno određene, osobine uređaja mogu u potpunosti kontinuirano utjecati na konačnu prezentaciju sadržaja. Tako će nekakva *ecommerce* mrežna trgovina na velikom zaslonu imati mnoštvo tipki, veza, ikona, prikaza i svega vizualnog, a na pametnom telefonu, samo one najbitnije i to točno one koje je dizajner smatrao najbitnijima. Mrežni dizajn ide ruku pod ruku s psihologijom čovjeka i svaka je najmanja odluka prilikom pisanja CSS koda veoma utjecajna od strane dobrih i efektivnih principa prodaje, marketinga, pričanja priče (engl. “storytelling”) i općenito kroz vrijeme naučenih efektivnih principa utjecanja na

---

<sup>4</sup> Usp. Pluralsight, 2015. What’s the difference between the front-end and the back-end? URL: <https://www.pluralsight.com/blog/film-games/whats-difference-front-end-back-end>



korisnika. Mrežni dokumenti nisu dizajnirani samo da budu međusobno drukčiji – dizajnirani su s jasnim ciljevima na umu, a među glavnim ciljevima jest da pretraživanje Interneta predstavlja korisniku jedno toplo i ugodno interaktivno iskustvo koje će ga potaknuti da se eventualno na isti dokument opet vrati. Mrežni dizajner je na Internetu često ono što je dizajner interijera u, primjerice, nekom salonu automobila.

CSS specifikacije također predvodi W3C, ista udruga koja predvodi HTML.<sup>5</sup> Prvi radni nacrt CSS-a formirao je Håkon Wium Lie već u listopadu 1994. godine<sup>6</sup>, devet dana nakon nastanka organizacije W3C. Zatim, W3C je javno objavio svojevrsan prijedlog CSS specifikacije 1996. godine, pod vodstvom radnog kolege Tima Berners-Leeja, Håkona Wium Liea, u suradništvu s nizozemskim doktorom matematike Bertom Bosom. Håkon Wium Lie i Bert Bos smatraju se suosnivačima jezika CSS. U početku, CSS nije naišao na stopostotnu podršku od strane svih mrežnih pretraživača, nego su različiti mrežni pretraživači imali različite jednostavno netočne implementacije nekih CSS funkcionalnosti. Rezultat toga bio je da je bilo nemoguće kreirati jednu kolekciju CSS datoteka koji bi jednoznačno dizajnirali mrežni dokument na svim pretraživačima, nego bi dokument različito izgledao na različitim pretraživačima. W3C je nastavio izdavati nove specifikacije i standarde i s vremenom, mrežni pretraživači dosegli su zadovoljavajuću razinu podrške koja bi približila interpretaciju CSS-a jednoznačnosti, no razlike u prikazu nekih CSS opisa i dalje postoje. Ako ove neodređenosti mrežni inženjeri ne trivijaliziraju odvojenim kolekcijama CSS datoteka, realizirat će ih pisanjem uvjetnih CSS svojstava unutar istog dokumenta.

### 3. CSS rešetke

CSS rešetke nastale su uslijed potrebe za jedinstvenim, preciznim i efikasnim sustavom dvodimenzionalnog razmještaja elemenata na raspoloživoj površini HTML dokumenta. Također, nastanak CSS rešetaka korelirao je s porastom u popularnosti pametnih telefona i sličnih uređaja čija je najupečatljivija razlika u odnosu na računalne zaslone bila činjenica da su većinu vremena promatrani u portretnoj orijentaciji zaslona, u odnosu na računalne zaslone koji uglavnom pokazuju

---

<sup>5</sup> Usp. A Brief history of CSS until 2016. URL <https://www.w3.org/Style/CSS20/history.html>

<sup>6</sup> Usp. W Lie, Håkon, Cascading HTML style sheets –a proposal, 1994. URL: <https://www.w3.org/People/howcome/p/cascade.html>

*pejzažnu* orijentaciju. Bilo je veoma potrebno osmisliti sustav koji programeru i dizajneru pruža kvalitetan i nepogrešiv način dizajniranja mrežnog dizajna koji bi bili izuzetno prilagodljivi na promjene dimenzija dokumenta koje opisuju.

CSS rešetke moderan su sustav za brzo, lako i moćno kreiranje dvodimenzionalnih struktura mrežnih dokumenata koristeći CSS. CSS rešetke podsustav su CSS-a i kao takve su tek nedavno dodane u jezik CSS. CSS rešetke veoma su korisne za pozicioniranje, poravnavanje i podešavanje dimenzija elemenata nekog HTML dokumenta, a prije CSS rešetaka, uglavnom su se koristile funkcionalnosti jezika CSS imenom *float* i *flexbox*. Smatra se da, iako i te funkcionalnosti imaju svoju idealnu svrhu korištenja, one nisu toliko jednostavne i moćne za realizaciju dvodimenzionalnih struktura HTML dokumenata na baš toj razini.

CSS rešetke razvijene su od strane Microsoft-ovog razvojnog inženjera Phila Cuppa, a prva radna verzija<sup>7</sup> CSS rešetaka bila je predstavljena pred udugom W3C tijekom 2012. godine. Navedena radna verzija nije bila prva verzija nekog grafičkog sustava rešetke, kakve su CSS rešetke, koji bi bio predstavljen pred W3C-om, čak štoviše, bilo ih je mnogo sugeriranih prethodno. No, W3C tvrdi kako je ovaj put razliku učinila činjenica da su CSS rešetke zapravo i imale eksperimentalnu implementaciju koju je bilo moguće testirati i koristiti u radu. S vremenom, CSS Radna Grupa za rešetke je osmislila tri primarna cilja:<sup>8</sup>

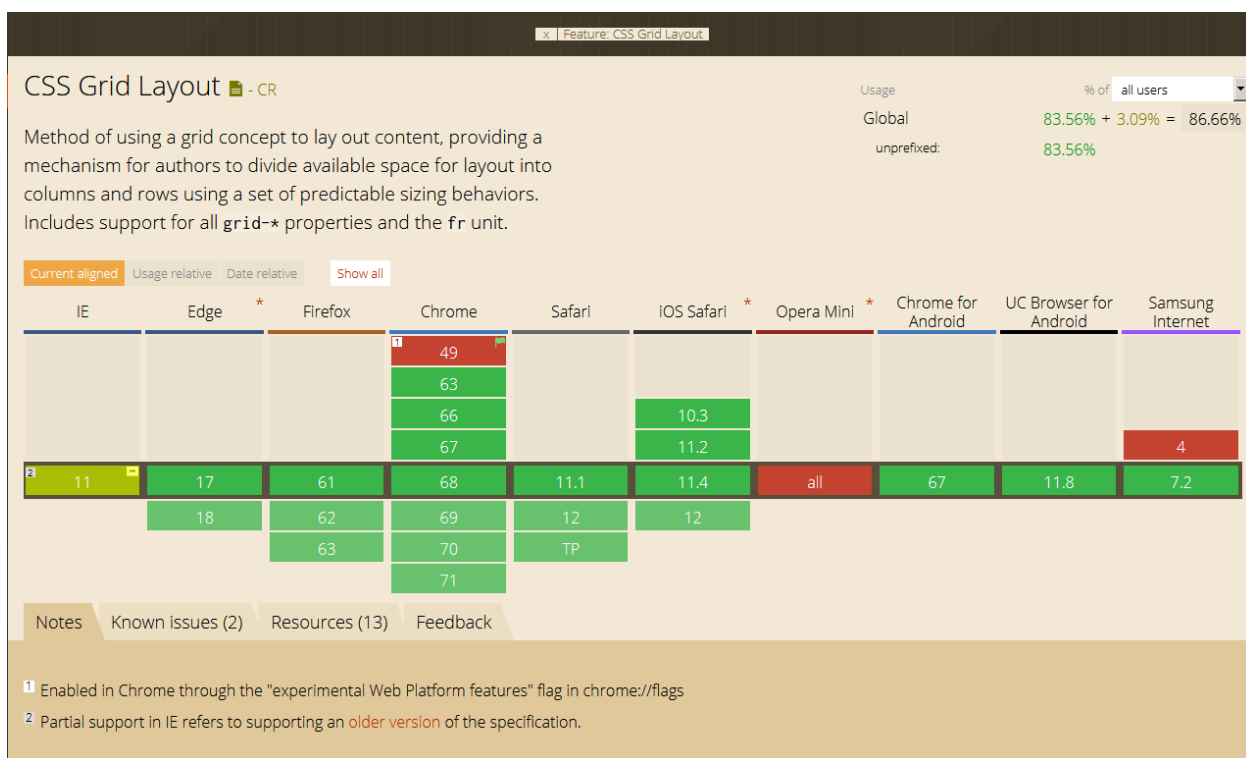
1. Kvaliteta – od CSS rešetaka se očekivalo da budu dovoljno kvalitetne da jednostavne odrednice dizajna učine lakim za implementaciju, a kompleksne – uopće mogućim.
2. Robusnost – od CSS rešetaka se očekivalo da budu dovoljno sigurne u radu da spriječe nagle i neobjašnjene gubitke sadržaja, nemogućnosti pretraživanja ili neočekivano ponašanje prikaza.
3. Performansa – ako algoritam rada CSS rešetaka ne bude bio dovoljno efikasan da pravovremeno brine o situacijama poput promjene dimenzija prozora mrežnog pretraživača ili o dinamičkom učitavanju sadržaja, CSS rešetke kreirale bi frustracije među korisnicima.

---

<sup>7</sup> Usp. Moglievsky, Alex; Cupp, Phil; Mielke, Markus; Glazman, Daniel. Grid Layout: W3C Working Draft 7 April 2011, 2011. URL: <https://www.w3.org/TR/2011/WD-css3-grid-layout-20110407/>

<sup>8</sup> Gustafson, Aaron. The Story of CSS Grid, from Its Creators, 2018. URL: <https://alistapart.com/article/the-story-of-css-grid-from-its-creators>

Da su CSS rešetke “nedavno dodane u CSS” je zapravo samo onoliko točno koliko ih moderni mrežni pretraživači podržavaju. Danas, CSS rešetke podržane su u svim trenutnim verzijama najpopularnijih pretraživača, osim u pretraživaču Opera Mini. Na Internetu je moguće pronaći jedan koristan alat imenom Caniuse<sup>9</sup> koji za CSS rešetke posjeduje sljedeće podatke o podržanosti od strane popularnih mrežnih pretraživača.

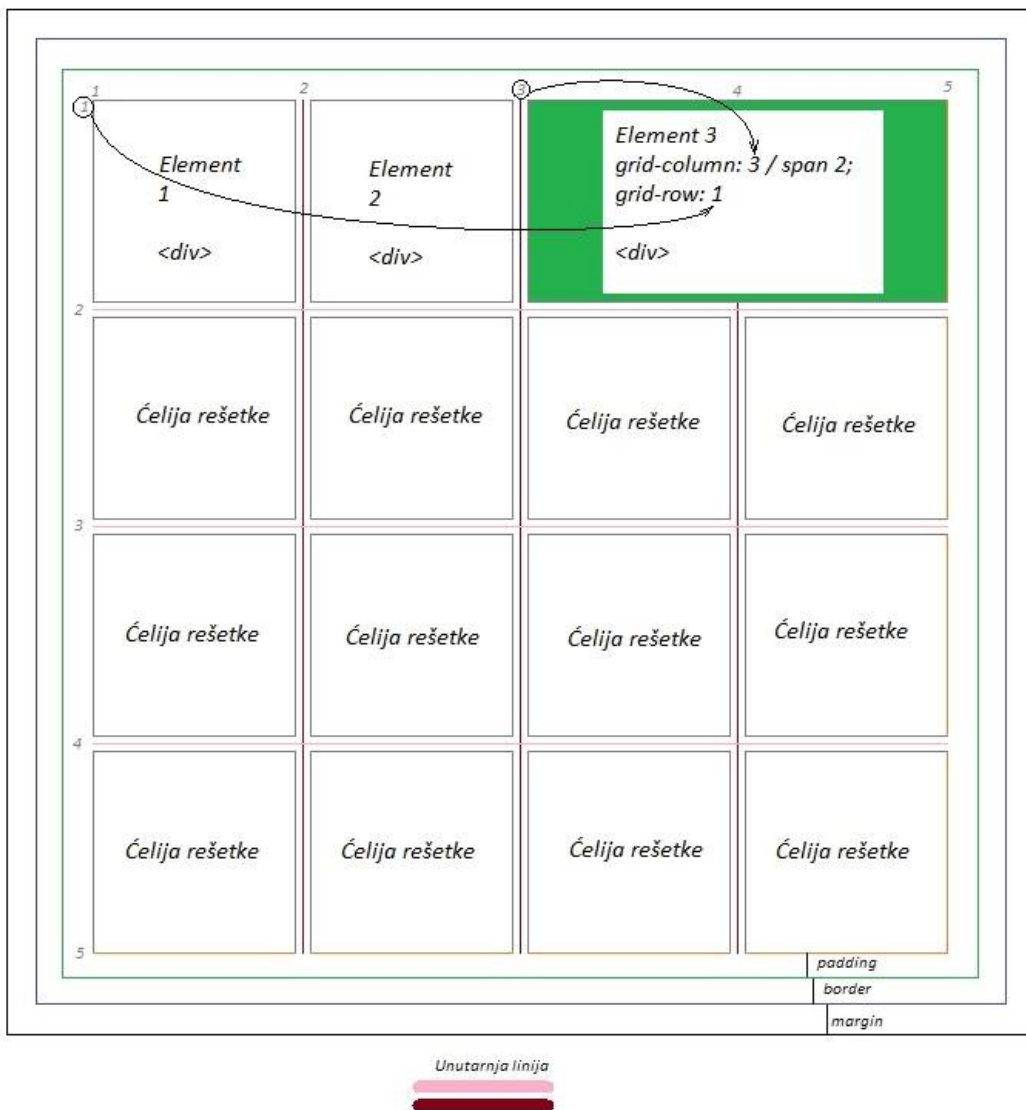


Slika 1: podržanost funkcionalnosti CSS rešetaka među najpopularnijim mrežnim pretraživačima na svijetu.

CSS rešetke izgrađene su na osnovama CSS-a. Među osnove CSS-a spada i takozvani “box model” – model kutije prema kojem su realizirani svi dokumenti na Internetu. Svaki element HTML dokumenta posjeduje *margin* atribut i *padding* atribut, kao i složeni atribut *border* koji opisuje vizualna svojstva granice elementa. *Margin* predstavlja minimalnu brojčanu udaljenost svih drugih elemenata od vanjskog ruba promatranog elementa, a *padding* predstavlja minimalnu brojčanu udaljenost svih drugih elemenata koji su u *child* odnosu s promatranim elementom, od njegovog unutarnjeg ruba.

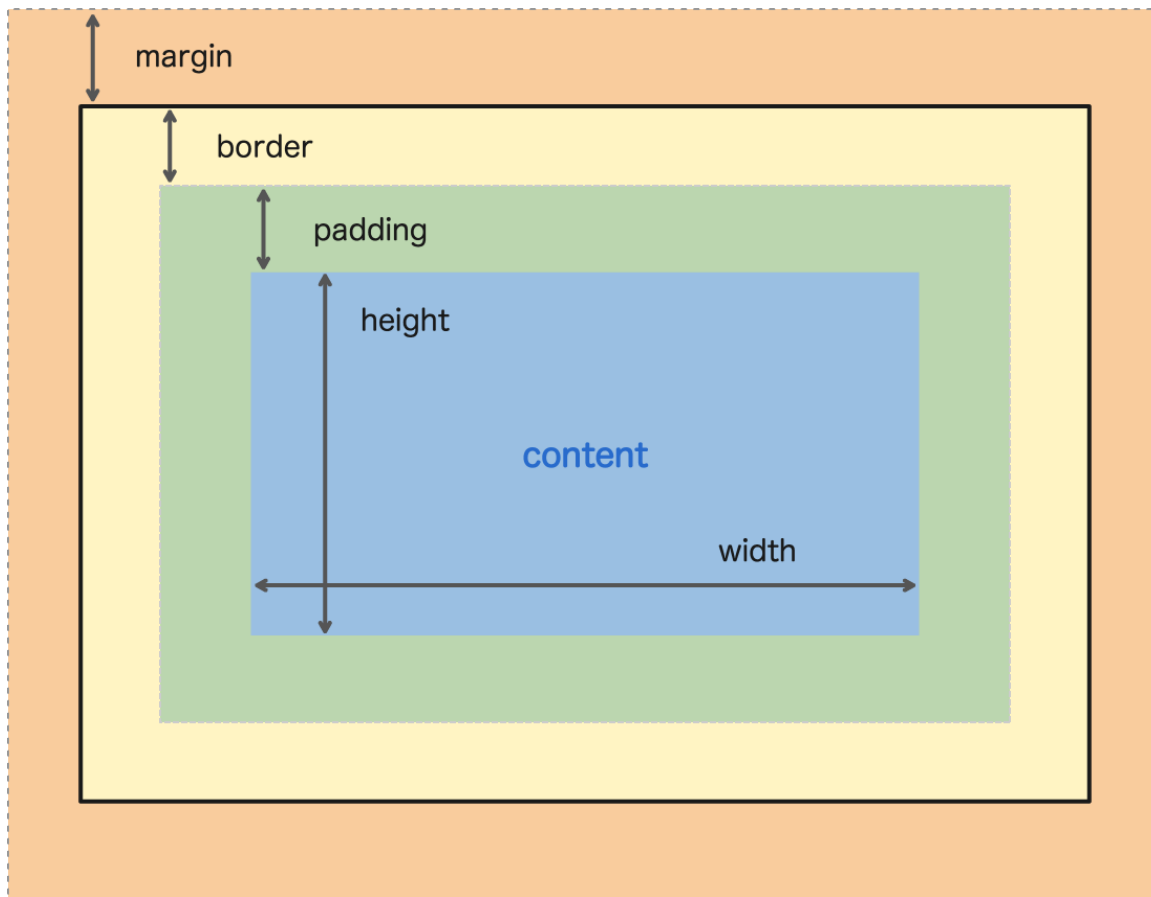
<sup>9</sup> Can I use..., URL: <https://caniuse.com/>

Slika 2 predstavlja kratko vizualno objašnjenje osnova rada CSS rešetaka, a slika 2a predstavlja CSS model kutije.



Slika 2: vizualna reprezentacija CSS *box modela* i osnova rada CSS rešetaka.<sup>10</sup>

<sup>10</sup> Usp. JavaScript Teacher. CSS Grid – The Beginner’s Guide, 2018. URL: <https://medium.freecodecamp.org/css-grid-the-beginners-guide-45998e6f6b8>



Slika 2a: CSS model kutije <sup>11</sup>

Najčešći element kojeg će se stilizirati koristeći CSS rešetke bit će `<div>` element. `<div>` element općenito je samo apstraktni element namijenjen logičkom odvajanju nekih drugih elemenata od svih ostalih. Elementi se pomoću `<div>` elementa odvajaju od ostalih kroz dodavanje navedenih elemenata u *parent-child* odnos u odnosu na navedeni `<div>` element. U kodu, to izgleda ovako:

---

<sup>11</sup> Okafuji, It's Not Too Late to Learn the Box-Model! An Introduction to CSS for Engineers, 2016. URL: <https://nulab-inc.com/blog/nulab/css-basics-for-engineer-boxmodel/>

```
<div class="asd" id="asd_2">
  <a href="registracija_1.php">
    <div id="Registracija">
      <span>Registracija</span>
    </div>
  </a>
</div>
<div class="asd" id="asd_3">
  <a href="prijava_1.php">
    <div id="Prijava">
      <span>Login</span>
    </div>
  </a>
</div>
```

Slika 3: primjer postavljanja nekih elemenata u *parent-child* odnos s nekim *<div>* elementom.

Konceptualni model prikazan na slici 2, s obzirom na elemente ćelija stvorene pomoću funkcionalnosti CSS rešetke postigao bi se sljedećim programskim kodom:

```

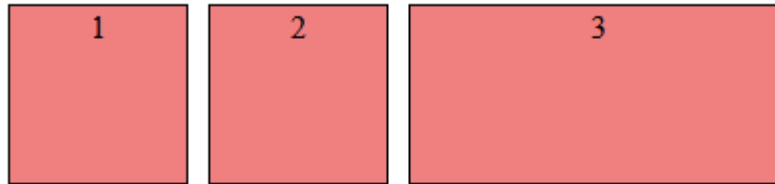
<html>
  <head>
    <style>
      #container{
        display: grid;
        grid-template-columns: 100px 100px 100px 100px;
        grid-template-rows: 100px 100px 100px;
      }
      #asd{
        grid-column: 3 / span 2;
        grid-row: 1;
      }
      #container > div{
        margin: 5px;
        background: lightcoral;
        border: 1px solid black;
        text-align: center;
      }
    </style>
  </head>
  <body>
    <div id="container">
      <div>1</div>
      <div>2</div>
      <div id="asd">3</div>
    </div>
  </body>
</html>

```

Slika 4: Programski kod pomoću kojeg je realiziran 2D raspored HTML elemenata na slici 2.<sup>12</sup>

Slika 2 je konceptualni model, a stvarni prikaz razmještaja elemenata kojeg kreira kod sa slike 2 vidljiv je na slici 5:

<sup>12</sup> Usp. JavaScript Teacher. CSS Grid – The Beginner’s Guide, 2018. URL: <https://medium.freecodecamp.org/css-grid-the-beginners-guide-45998e6f6b8>



Slika 5: Stvarni prikaz razmjesta elemenata kojeg kreira kod sa slike 9. Mozilla Firefox.

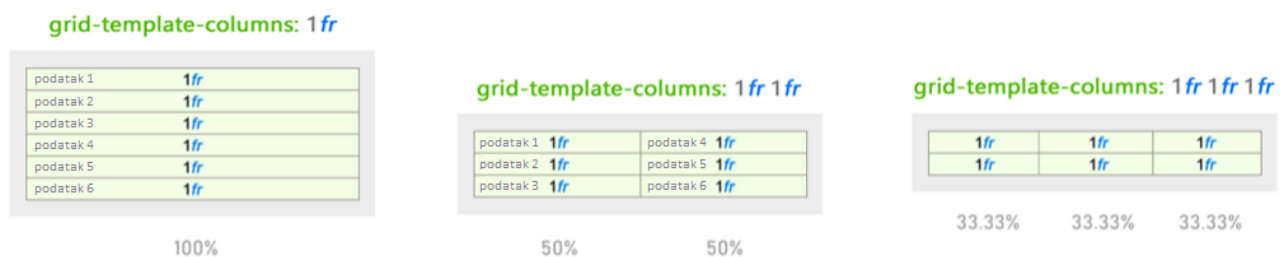
Ključan dio postavljanja nekog `<div>` elementa u kontekst pravila CSS rešetaka jest atribut `display: grid` na slici 4. Pomoću tog koda, mrežnom pretraživaču poručuje se da sve *child* elemente tog `<div>` elementa pretraživač tretira kao svojevrsnu rešetku. Za takvu rešetku definiraju se stupci i redovi, a preko toga i sve vizualne odrednice koje stupci i redovi mogu imati, poput širina, visina, broja i ostalih veličina.

Prema slici 4 (na prethodnoj stranici) vidljivo je da CSS rešetke automatski pretpostavljaju broj stupaca i redova prema broju veličina koje su navedene, veličina koje u slučaju slike 2 opisuju dimenzije svakog od redova i stupaca.



## 3.2 Fractional units – razlomačke jedinice

Jedna jedinstvena i posebna funkcionalnost CSS rešetaka jest funkcionalnost relativnih jedinica. CSS rešetke omogućavaju korištenje tih relativnih jedinica koje same po sebi nemaju jasne dimenzije, ali CSS rešetke imaju sposobnost otkriti koliko ih je sveukupno korišteno u nekom kontekstu i raspoloživu površinu podijeliti među elementima ovisno o tome koliko relativnih jedinica koji element posjeduje duž neke dimenzije. Jedna relativna jedinica označava se kao  $1fr$ .



Slika 6: jednostavna realizacija nekoliko rešetki uz korištenje relativnih jedinica  $fr$  i attribute CSS rešetaka.<sup>13</sup>

Na slici 6 vidljivo je kako CSS rešetke pozicioniraju i dimenzioniraju elemente kada se koriste relativne jedinice. CSS rešetke će prepoznati koliko ukupno jedinica ima duž neke dimenzije i pravilno i proporcionalno raspodijeliti površine među elementima, ovisno o koliko koji element ima jedinica vezanih uz svoje dimenzije. CSS rešetke također omogućavaju dodatno automatsko sažimanje elemenata ovisno o povećanju praznina između elemenata. Praznine se definiraju korištenjem idućih atributa:

<sup>13</sup> “JavaScript Teacher”. CSS Grid – The Beginner’s Guide, 2018. URL: <https://medium.freecodecamp.org/css-grid-the-beginners-guide-45998e6f6b8>

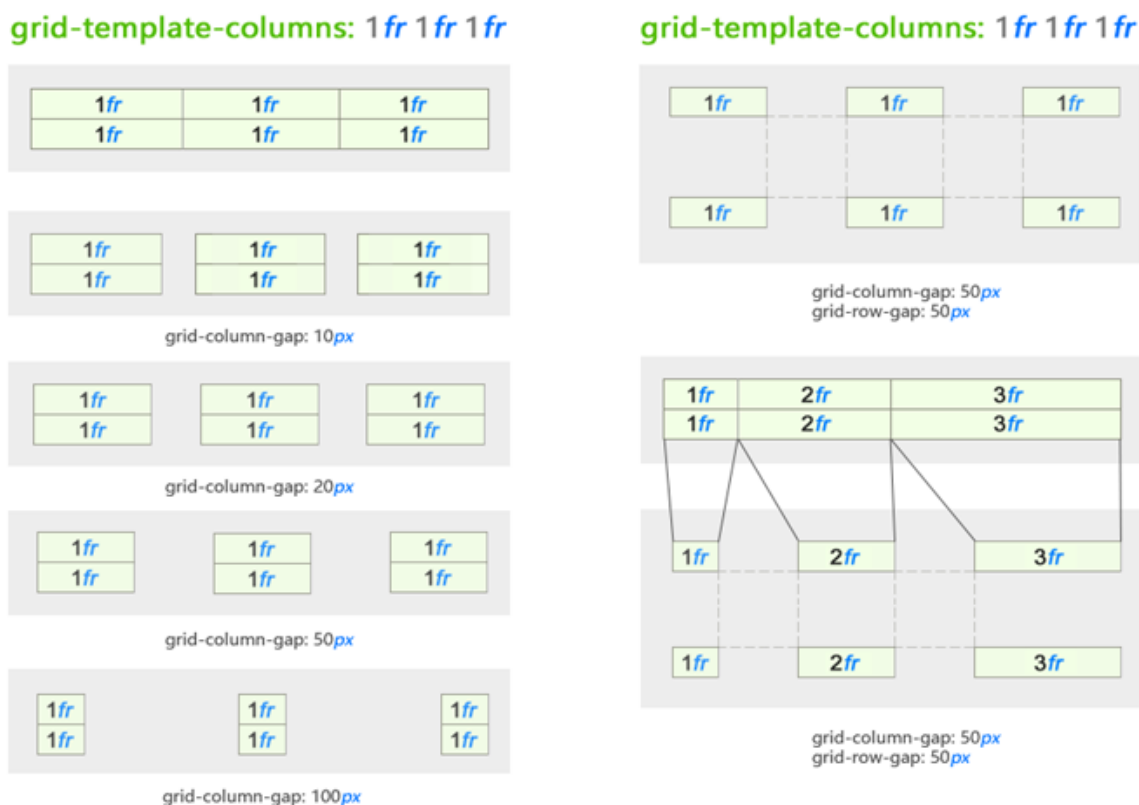
```

div{
  //...
  grid-column-gap: 10px;
  grid-row-gap: 10px;
  //...
}

```

Slika 7: atributi pomoću kojih se postavljaju dimenzije praznina između pojedinih elemenata u rešetci:

Kada se jedinice *fr* kombiniraju s navodima poput onih sa slike 7, rezultat izgleda kao na slici 8:

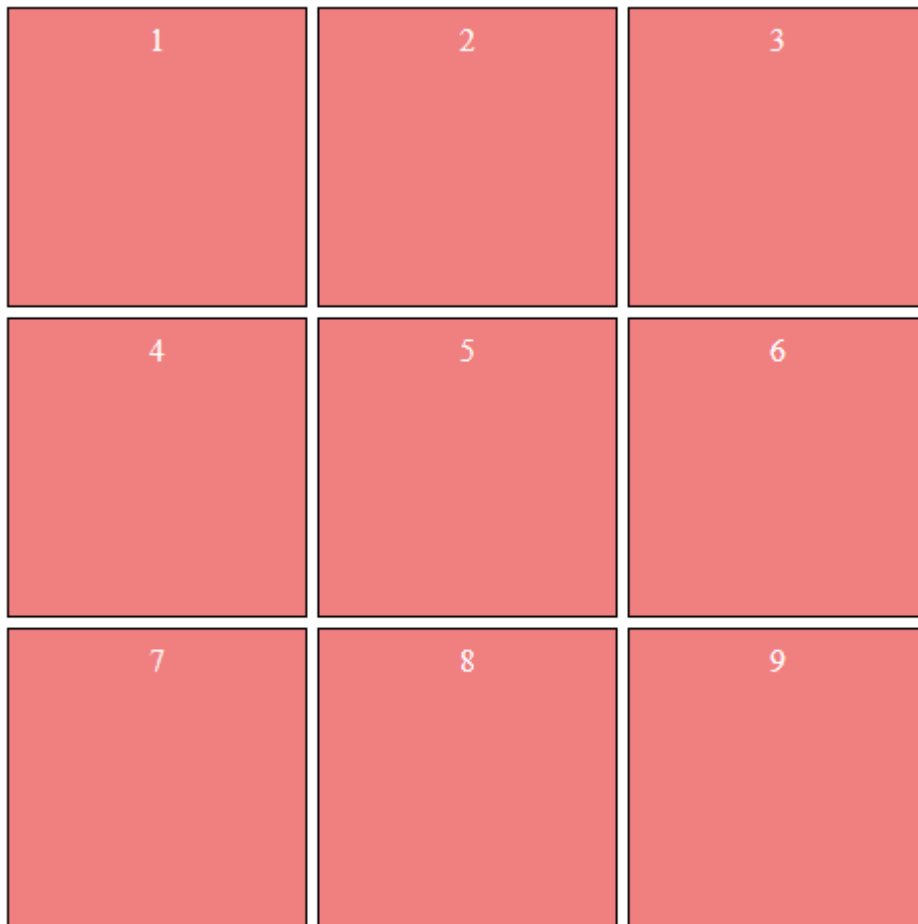


Slika 8: vizualni rezultat modificiranja neke rešetke atributima CSS rešetaka sa slike 7 <sup>14</sup>

<sup>14</sup> JavaScript Teacher. CSS Grid – The Beginner’s Guide, 2018. URL: <https://medium.freecodecamp.org/css-grid-the-beginners-guide-45998e6f6b8>

### 3.3 Primjeri nekih od atributa propisanih specifikacijom CSS rešetaka

U ovom poglavlju nalazi se nekoliko primjera atributa koje predviđa specifikacija CSS rešetaka i njihov utjecaj na prezentaciju elementa nad kojim rade. Prikazani atributi nisu svi korišteni CSS atributi, s obzirom da su polja obojana i popunjena tekстом, ali ostatak koda izostavljen je radi praktičnosti.



Slika 9: Rezultat primjene koda sa slike 9a.<sup>15</sup>

---

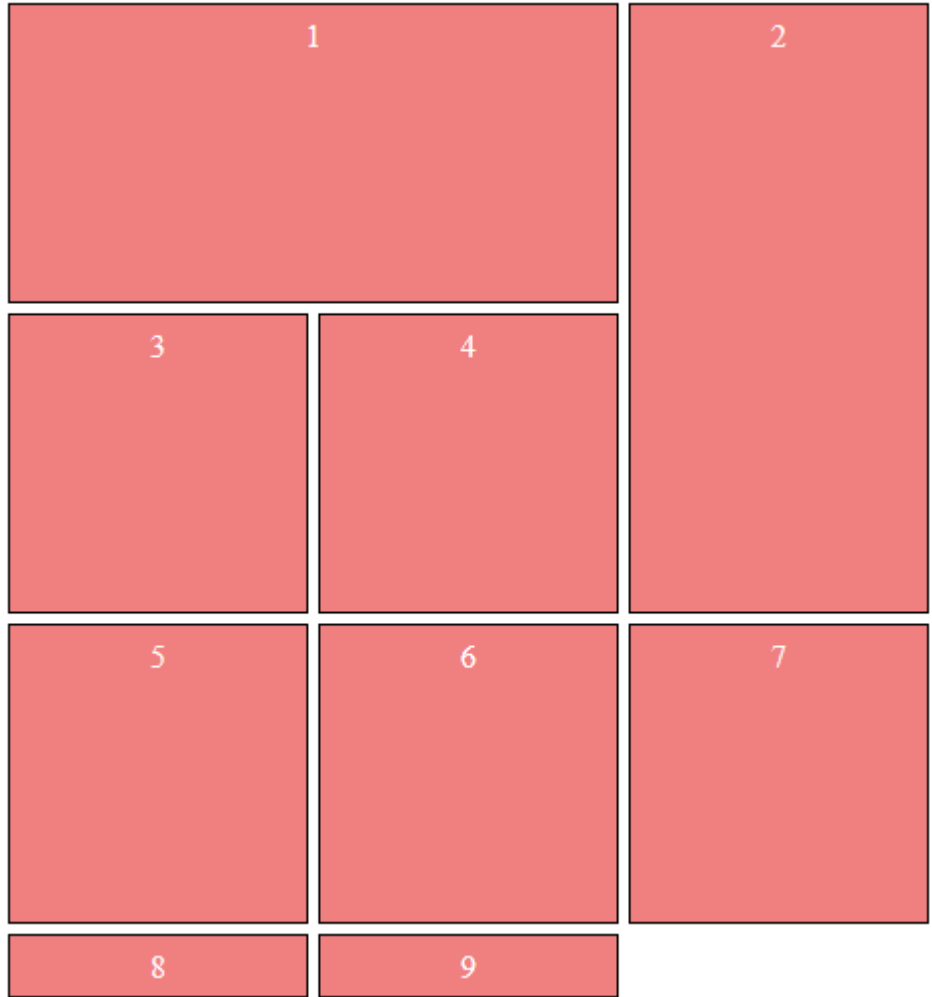
<sup>15</sup> Usp. Grid by Example: Everything you need to learn CSS Grid Layout. URL: <https://gridbyexample.com/examples/>

```

<html>
  <head>
    <style>
      body{
        margin: 5px;
      }
      .kontejner{
        display: grid;
        grid-template-columns: 150px 150px 150px;
        grid-template-rows: 150px 150px 150px;
        grid-gap: 5px;
        background-color: white;
        color: grey;
      }
      .kontejner > div{
        background-color: lightcoral;
        color: white;
        border: 1px solid black;
        padding: 5px;
        text-align: center;
      }
    </style>
  </head>
  <body>
    <div class="kontejner">
      <div class="a">1</div>
      <div class="b">2</div>
      <div class="c">3</div>
      <div class="d">4</div>
      <div class="e">5</div>
      <div class="f">6</div>
      <div class="g">7</div>
      <div class="h">8</div>
      <div class="i">9</div>
    </div>
  </body>
</html>

```

Slika 9a: Kod korišten za postizanje rezultata sa slike 9.



Slika 10: Rezultat primjene koda sa slike 10a. <sup>16</sup>

---

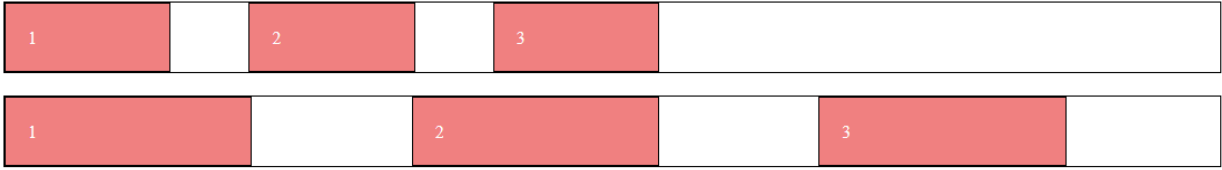
<sup>16</sup> Usp. Grid by Example: Everything you need to learn CSS Grid Layout. URL: <https://gridbyexample.com/examples/>

```

<html>
  <head>
    <style>
      body{
        margin: 5px;
      }
      .kontejner{
        display: grid;
        grid-template-columns: 150px 150px 150px;
        grid-template-rows: 150px 150px 150px;
        grid-gap: 5px;
        background-color: white;
        color: grey;
      }
      .kontejner > div{
        background-color: lightcoral;
        color: white;
        border: 1px solid black;
        padding: 5px;
        text-align: center;
      }
      .a{
        grid-column: 1 / span 2;
      }
      .b{
        grid-column: 3;
        grid-row: 1 / span 2;
      }
      .c{
        grid-column: 1;
        grid-row: 2;
      }
      .d{
        grid-column: 2;
        grid-row: 2;
      }
    </style>
  </head>
  <body>
    <div class="kontejner">
      <div class="a">1</div>
      <div class="b">2</div>
      <div class="c">3</div>
      <div class="d">4</div>
      <div class="e">5</div>
      <div class="f">6</div>
      <div class="g">7</div>
      <div class="h">8</div>
      <div class="i">9</div>
    </div>
  </body>
</html>

```

Slika 10a: Kod korišten za postizanje rezultata sa slike 10.



Slika 11: Rezultat primjene koda sa slike 11a.<sup>17</sup>

---

<sup>17</sup> Usp. Grid by Example: Everything you need to learn CSS Grid Layout. URL: <https://gridbyexample.com/examples/>

```

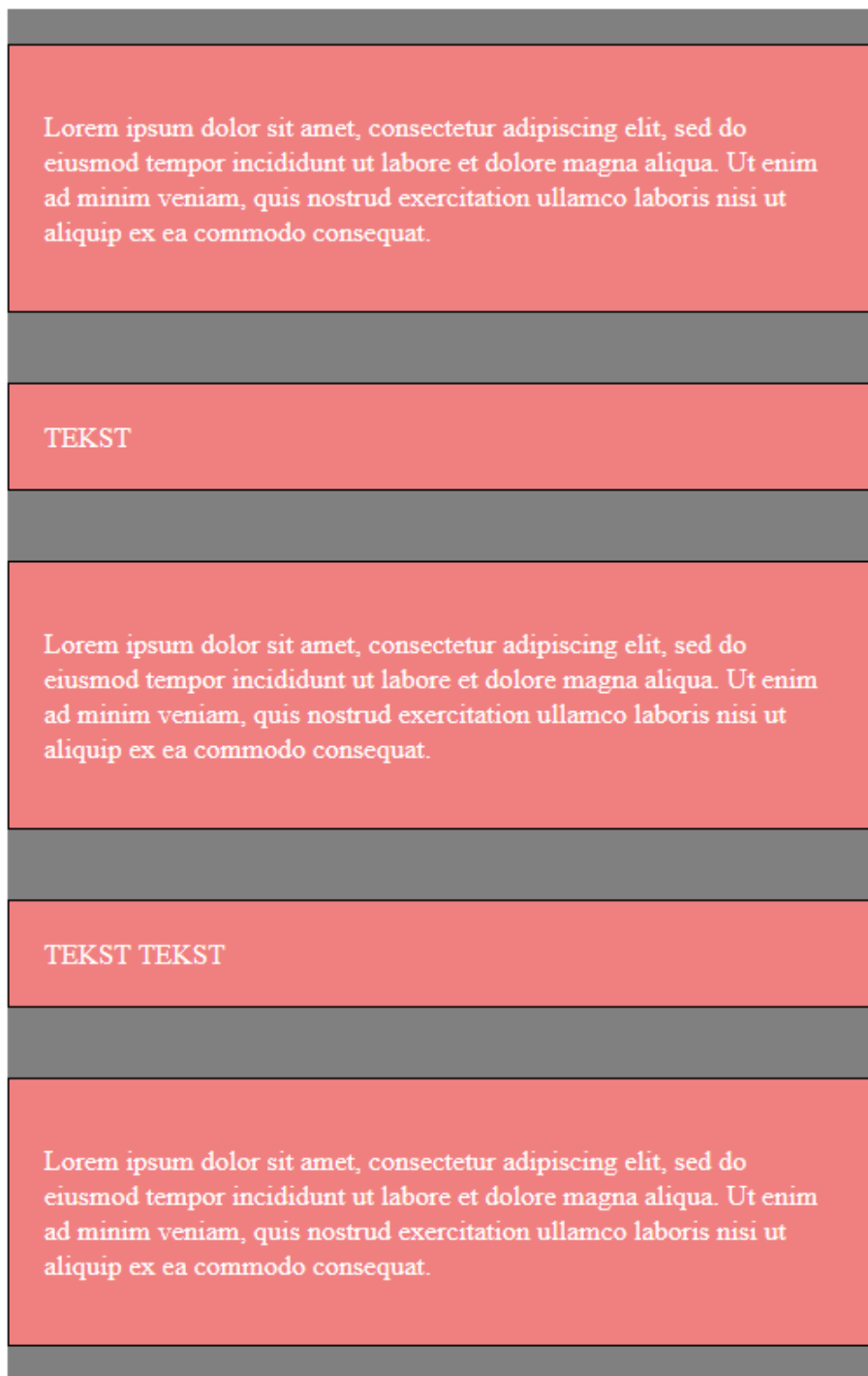
<html>
  <head>
    <style>
      body{
        margin: 50px;
      }
      .element{
        background-color: lightcoral;
        color: white;
        border: 1px solid black;
        padding: 20px;
        width: 50%;
      }
      .kontejner1{
        display: grid;
        border: 1px solid black;
        grid-gap: 5px;
        grid-template-columns: repeat(auto-fill, minmax(200px,1fr));
        margin-bottom: 20px;
        width: 60%;
      }
      .kontejner2{
        display: grid;
        border: 1px solid black;
        grid-gap: 5px;
        grid-template-columns: repeat(auto-fit, minmax(200px,1fr));
        width: 60%;
      }
    </style>
  </head>
  <body>
    <div class="kontejner1">
      <div class="element">1</div>
      <div class="element">2</div>
      <div class="element">3</div>
    </div>

    <div class="kontejner2">
      <div class="element">1</div>
      <div class="element">2</div>
      <div class="element">3</div>
    </div>
  </body>
</html>

```

Slika 11a: Kod korišten za postizanje rezultata sa slike 11.





Slika 12: Rezultat primjene koda sa slike 12a.<sup>18</sup>

---

<sup>18</sup> Usp. Grid by Example: Everything you need to learn CSS Grid Layout. URL: <https://gridbyexample.com/examples/>

```

<html>
<head>
<style>
body{
margin: 50px;
}

.kontejner > div{
background-color: lightcoral;
color: white;
border: 1px solid black;
padding: 20px;
margin: 20px 0 20px 0;
}

.kontejner{
display: grid;
width: 500px;
grid-template-columns: auto;
grid-template-rows: auto;
background-color: grey;
color: white;
}

.dio1{
grid-row: 1 / 2;
}

.dio2{
grid-row: 3 / 4;
}

.dio3{
grid-row: 5 / 6;
}

.A{
grid-row: 2 / 3;
}

.B{
grid-row: 4 / 5;
}
</style>
</head>
<body>
<div class="kontejner">
<div class="dio1">
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
</div>
<div class="dio2">
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
</div>
<div class="dio3">
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
</div>
<div class="A">TEKST</div>
<div class="B">TEKST TEKST</div>
</div>
</body>
</html>

```

Slika 12a: Kod korišten za postizanje rezultata sa slike 12.

## 4. Zaključak

CSS rešetke predstavljaju proširenje CSS-a prvenstveno namijenjeno usavršavanju sposobnosti CSS-a za razmještanje elemenata u dvodimenzionalnu strukturu rešetke. CSS rešetke građene su na temelju HTML-a i CSS-a koji, unatoč činjenici da bi sami po sebi trebali biti dovoljni za realizaciju baš svih dizajnerskih koncepata, ipak ne nude praktičnost navedene realizacije na nivou koje CSS rešetke nude u kontekstu realizacije struktura rešetke. Prije nastanka CSS rešetaka, razvojni inženjeri oslanjali su se na svojevrstne trikove u osnovnim mrežnim tehnologijama kako bi postigli željene rezultate, a rezultati su bili često neuračunljivi i samo privremena rješenja, ako i to. Rješenja realizirana korištenjem opisanih snalažljivih, ali nestabilnih tvorevina bila su neotporna na neočekivane izmjene dimenzija nadležnih elemenata i sličnih globalnih i trenutnih izmjena nad scenom, uslijed kojih bi najčešće doživljavale neočekivano ponašanje, ako ne i nestanak funkcionalnosti i prikaza. CSS rešetke predstavljaju kvalitetan i robustan sustav visokih performansi. Ono što se prije realiziralo tabličnom implementacijom, danas je moguće vizualizirati korištenjem CSS rešetaka. Izuzetna implicitna priroda CSS rešetaka (elementi mogu poprimiti osobine i bez da ih programer posebno imenuje – na temelju okolnih uvjeta) dozvoljava veoma brzu implementaciju raznih dizajnerskih ideja koja je također otporna na iznenadno dodavanje novih elemenata u postojeće kolekcije elemenata. CSS rešetke predstavljaju adekvatnu prilagodbu na ekosustav internetski sposobnih uređaja kakav danas uvelike definiraju mobilni uređaji. U budućnosti mogu se očekivati daljnje verzije i specifikacije kao i popravke nenamjernih grešaka u programiranju, ali i poboljšanja u performansama.

Agilan i prilagodljiv način rada ovog proširenja CSS-a također će služiti kao osnova za razvoj nekog budućeg proširenja neke druge tehnologije. U tehnološkoj eri sve veće prilagodljivosti svih internetski sposobnih uređaja, ova filozofija razvoja tehnologije predstavlja najveći prioritet – tehnologije niskih performansi nisu niti kandidat za vodeće tehnologije u industriji.

## 5. Literatura

1. A Brief history of CSS until 2016. URL: <https://www.w3.org/Style/CSS20/history.html> (2018-09-05)
2. A brief SGML tutorial. URL: <https://www.w3.org/TR/WD-html40-970708/intro/sgmltut.html> (2018-07-09)
3. Basic concepts of grid layout, 2018. URL: [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Grid\\_Layout/Basic\\_Concepts\\_of\\_Grid\\_Layout](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout) (2018-08-23)
4. Berners-Lee, T.; Connolly, D. Hypertext Markup Language - 2.0, 1995. URL: <https://tools.ietf.org/html/rfc1866> (2018-22-08)
5. Can I use..., URL: <https://caniuse.com/> (2018-08-26)
6. CSS Grid Layout Terminology: Explained, 2017. URL: <https://bitsofco.de/css-grid-terminology/> (2018-09-01)
7. Cuppens, Martijn. The div that looks different in every browser, 2018. URL: [https://twitter.com/martijn\\_cuppens/status/1015169981368225793?lang=en](https://twitter.com/martijn_cuppens/status/1015169981368225793?lang=en) (2018-27-08)
8. Grid by Example: Everything you need to learn CSS Grid Layout. URL: <https://gridbyexample.com/examples/> (2018-04-09)
9. Gustafson, Aaron. The Story of CSS Grid, from Its Creators, 2018. URL: <https://alistapart.com/article/the-story-of-css-grid-from-its-creators> (2018-27-08)
10. HTML & CSS. URL: <https://www.w3.org/standards/webdesign/htmlcss> (2018-07-09)
11. JavaScript Teacher. CSS Grid – The Beginner’s Guide, 2018. URL: <https://medium.freecodecamp.org/css-grid-the-beginners-guide-45998e6f6b8> (2018-29-08)
12. Moglievsky, Alex; Cupp, Phil; Mielke, Markus; Glazman, Daniel. Grid Layout: W3C Working Draft 7 April 2011, 2011. URL: <https://www.w3.org/TR/2011/WD-css3-grid-layout-20110407/> (2018-01-09)

13. Okafuji, It's Not Too Late to Learn the Box-Model! An Introduction to CSS for Engineers, 2016. URL: <https://nulab-inc.com/blog/nulab/css-basics-for-engineer-boxmodel/> (2018-05-09)
14. Pluralsight, 2015. What's the difference between the front-end and the back-end? URL: <https://www.pluralsight.com/blog/film-games/whats-difference-front-end-back-end> (2018-07-09)
15. Practical CSS Grid: Adding Grid to an Existing Design, 2017. URL: <https://alistapart.com/article/practical-grid> (2018-08-29)
16. Wium Lie, Håkon; Bos, Bert. Cascading Style Sheets, level 1: W3C Recommendation 17 Dec 1996, revised 11 Apr 2008, 2008. URL: <https://www.w3.org/TR/CSS1/> (2018-26-08)