

# Responsive slike

Pored teksta, svakako najznačajniji tip sadržaja web sajtova jesu slike. Stoga su adekvatno rukovanje slikama i njihova optimizacija unutar HTML dokumenata apsolutni imperativ. Loša optimizacija slika može stvoriti brojne funkcionalne probleme na web sajtovima, a u ekstremnim situacijama može dovesti i do potpuno nefunkcionalnog sajta. Zbog toga su različite tehnike za pripremu i kreiranje responsive ponašanja slika veoma važne prilikom web razvoja.

## Slike na webu

Unutar HTML dokumenata, slike se mogu pojaviti u dva različita oblika:

- slike predstavljene `img` elementom;
- pozadinske slike koje se dokumentu dodaju korišćenjem CSS svojstava `background` ili `background-image`.

Osnovni element za uključivanje slika u HTML dokumente jeste element `img`. Reč je o elementu koji je veoma upotrebljiv kada je dokumentu potrebno dodati neku fotografiju, ilustraciju ili crtež. Ipak, pored `img` elementa, postoji još jedan način koji je moguće koristiti za dodavanje slika HTML dokumentima. Reč je o načinu koji podrazumeva korišćenje CSS jezika. Korišćenjem takvog pristupa moguće je sliku postaviti kao pozadinu bilo kojeg HTML elementa koji postoji u dokumentu. Bitno je znati da pozadinske slike nisu deo HTML strukture, dok slike koje se dodaju korišćenjem `img` elementa to jesu.

## Priprema slika za web

Slike za web zahtevaju posebnu pažnju kako bi na adekvatan način bile prezentovane korisnicima. Loša optimizacija slika na webu, pored lošeg korisničkog doživljaja, može prouzrokovati i veoma loše performanse, odnosno suviše dugo vreme učitavanja stranice. Zbog toga je prilikom uključivanja slika u HTML dokument potrebno voditi računa o sledećem:

- **format fajla** – tri su standardna formata slika za koje je garantovano da će ispravno funkcionisati na svim web pregledačima: JPG za fotografije, PNG za slike sa transparentijom i GIF za mape, crteže, ilustracije ili kratke animacije;
- **fizička veličina fajla koji predstavlja sliku** – neko nepisano pravilo nalaže da je veličina od 200 KB maksimum za sliku koja se postavlja online; naravno, nije reč o striktnom pravilu ili restrikciji, ali je dobro truditi se da veličina slike ne prelazi 200 KB;
- **stvarna, fizička veličina slike koja se izražava u pikselima** – veličina slike u pikselima (1920x1080px ili 800x600px, na primer) treba da odgovara prostoru koji će slika zauzeti na stranici; drugim rečima, nije dobro da izvorna veličina slike bude veća, ali ni manja od veličine koju će takva slika imati na stranici; u prvom slučaju, veličina slike u KB će bespotrebno biti veća nego što je potrebno, dok će u drugom slučaju doći do degradacije kvaliteta prikaza (takozvane pikselizacije); kako bi se u svim situacijama i na svim uređajima postigao optimalni prikaz slika, neophodno je pribegli tehnici koja podrazumeva korišćenje nekoliko verzija jedne iste slike; svaka takva verzija poseduje drugačije dimenzije, pa se u zavisnosti od situacije bira ona koja najviše odgovara osobinama displeja.

Upravo navedeni postulati koji se tiču ispravnog rukovanja slikama za web odnose se na operacije koje je potrebno sprovesti pre nego što se slike postave na web sajt. Drugim rečima, oni se tiču pripremnih radnji koje je potrebno obaviti nad slikama. Iako veoma značajne, ove operacije nisu jedine koje se primenjuju kako bi na web sajtu slike reagovala na promene osobina displeja na kojima se prikazuju.

## Osnovne tehnike za kreiranje responsive slika

Pojam *responsive slika* odnosi se na njihov prikaz unutar HTML dokumenata, koji je zavisen od brojnih parametara, odnosno osobina displeja. Responsive slike reaguju na promene osobina displeja na kojima se prikazuju. U narednim redovima, prvo će biti prikazane osnovne tehnike za dobijanje responsive slika, dok će u narednom segmentu biti ilustrovani i neki napredniji pristupi.

Tehnike za kreiranje responsive slika razlikuju se u zavisnosti od njihovog tipa. Tako će prvo biti ilustrovane osnovne tehnike za kreiranje responsive slika koje se dokumentu dodaju korišćenjem `img` elementa.

### img slike

Kada se neka slika postavi unutar HTML dokumenta korišćenjem `img` elementa, ona podrazumevano ima svoje izворne dimenzije. Na primer, ukoliko je slika veličine 600x400 piksela, ona će upravo toliku veličinu imati i unutar HTML dokumenta. Takva slika ne reaguje na promenu veličine vidnog polja (*viewport*), pa se ne može nazvati responsive slikom.

Kako bi se slika prilagodila veličini elementa unutar koga se nalazi, dovoljno je napisati sledeće:

```
img.product {  
    width: 100%;  
}
```

U primeru je širina slike postavljena na 100%. Ovakva slika će uvek biti jednaka širini svog roditelja. To praktično znači da će slika moći da se poveća ili smanji, a sve u zavisnosti od veličine roditelja. Tako će prikaz slike na stranici u nekim situacijama moći da bude i veći od fizičke veličine slike, što svakako nije dobro, jer dovodi do prikaza degradiranog kvaliteta. Kako bi se ovakav problem prevazišao, dovoljno je formirati nešto drugačiji CSS opis:

```
img.product {  
    max-width: 100%;  
}
```

Prikazanim CSS opisom, rečeno je da će maksimalna veličina slike biti ona koju slika izvorno ima. Ipak, ukoliko prostor za prikaz slike nije dovoljan kako bi se ona prikazala u svojoj izvornoj veličini, slika će se smanjiti. Drugim rečima, slika će moći samo da se smanjuje, ali ne i da se poveća preko svojih fizičkih dimenzija.

Na kraju, prilikom definisanja veličine slike na stranici, moguće je otići i korak dalje, i eksplicitno kontrolisati maksimalnu veličinu koju slika može imati:

```
img.product {  
    width: 100%;  
    max-width: 500px;  
}
```

Na ovaj način je maksimalna širina slike postavljena na 500px. Čak i kada na stranici ima više prostora za njen prikaz, ovakva slika će imati širinu od 500px. S obzirom na to da je širina (*width*) postavljena na 100%, slika će adekvatno pratiti smanjivanje dostupnog prostora.

## Pozadinske slike

Pozadinske slike se HTML dokumentima dodaju korišćenjem `background` ili `background-image` CSS svojstava. Za njih važi identično pravilo kao i za `img` slike: podrazumevano imaju svoju izvornu veličinu. Ukoliko je pozadinska slika veća od elementa čiju pozadinu predstavlja, ona će biti isečena (*crop*) veličinom elementa. Kada je izvorna veličina slike manja od veličine elementa za čiju pozadinu je postavljena, podrazumevano će pozadinska slika biti replicirana (ponovljena) i po visini i po širini, sve dok ne popuni pozadinu kompletnog elementa. Naravno, korišćenjem ovakvog podrazumevanog ponašanja ne dobijaju se pozadinske slike koje mogu reagovati na promenu osobina vidnog polja.

Kako bi pozadinska slika pratila promenu elementa nad kojim je definisana, moguće je iskoristiti CSS svojstvo `background-size` i njegove vrednosti:

- `contain` – slika će biti uvećana ili umanjena tako da popuni element bez odsecanja ili promene odnosa stranica;
- `cover` – slika će popuniti kompletan prostor unutar elementa, ali u slučaju nepodudaranja odnosa stranica elementa i pozadinske slike, doći će do odsecanja njenih delova.

Obe prikazane vrednosti garantuju da će pozadinska slika pratiti promenu veličine elementa. Ipak, CSS ne poznaje svojstvo kojim je moguće postaviti maksimalnu veličinu pozadinskih slika, te je stoga potrebno voditi računa o tome da pozadinska slika na stranici ne zauzme veći prostor od svojih fizičkih dimenzija, čime bi se izazvala pikselizacija.

### Pitanje

Podrazumevana veličina slika koje se dodaju HTML dokumentima je:

- **izvorna fizička veličina slike**
- veličina roditeljskog elementa
- veličina HTML dokumenta
- veličina prozora web pregledača

### Objašnjenje:

*Bilo da je reč o slikama koje se HTML dokumentu dodaju korišćenjem `img` elementa ili o pozadinskim slikama, njihova podrazumevana veličina unutar dokumenta je identična njihovoj izvornoj veličini u pikselima.*

## Napredne tehnike za kreiranje responsive slika

Upravo prikazane osnovne tehnike kojima se mogu dobiti prilagodljive (*responsive*) slike veoma su jednostavne za realizaciju. Ipak, iako obezbeđuju brzinu i jednostavnost, prikazana rešenja imaju jedan veliki problem.

Iako prikaz slika na stranici prati promenu veličine vidnog polja, fizička veličina slika je uvek ista. Drugim rečima, skaliranje slika je prepušteno web pregledaču. Tako svi uređaji, bez obzira na osobine displeja, dobijaju jednu istu varijantu slike. Fizička veličina takve slike može biti odgovarajuća za jedan tip uređaja, ali nikako za sve. Kako bi se postigao optimalan prikaz slika na svim uređajima i uštedeli resursi, pribegava se tehnici koja podrazumeva korišćenje većeg broja verzija jedne iste slike.

### Više varijanti jedne iste pozadinske slike

Definisanje više verzija jedne iste slike najlakše je postići sa pozadinskim slikama. Najjednostavniji primer implementacije takvog ponašanja može da izgleda ovako:

```
#hero {  
  background-image: url("tokyo-2048w.jpg");  
}  
  
@media screen and (max-width: 1024px) {  
  #hero {  
    background-image: url("tokyo-1024w.jpg");  
  }  
}  
  
@media screen and (max-width: 768px) {  
  #hero {  
    background-image: url("tokyo-768w.jpg");  
  }  
}  
  
@media screen and (max-width: 414px) {  
  #hero {  
    background-image: url("tokyo-414w.jpg");  
  }  
}
```

Ovakvom stilizacijom definisane su četiri različite varijante jedne iste slike, koje će se koristiti u različitim situacijama u zavisnosti od osobina vidnog polja uređaja (tabela 16.1).

Širina vidnog polja	Naziv slike	Fizička veličina slika
veća od 1024px	tokyo-2048w.jpg	1920x1080px
769–1024px	tokyo-1024w.jpg	1024x576px
415–768px	tokyo-768w.jpg	768x432px
manja od 415px	tokyo-414w.jpg	414x233px

Tabela 16.1. Efekat medija upita za kreiranje responsive pozadinskih slika

Unutar tabele 16.1. prikazana je kompletna logika po kojoj funkcionišu upravo prikazani medija upiti za kreiranje responsive pozadinskih slika. Jednostavno, u zavisnosti od širine vidnog polja, učitava se odgovarajuća slika. Slike su imenovane tako da poseduju slikovite sufikse koji govore kolika je fizička širina slike (na primer, 1024w znači da je slika široka 1024px).

Ipak, i ovakav pristup poseduje jedan nedostatak – šta raditi sa uređajima koji poseduju displeje velike gustine (kompanija Apple takve displeje naziva Retina)?

Danas su displeji na gotovo svim pametnim telefonima i tabletima displeji velikih gustina piksela. Ukoliko tome dodamo činjenicu koja je poznata iz prethodne lekcije, a tiče se postojanja softverskih i hardverskih piksela, jasno je da se stvari komplikuju.

Na primer, popularni uređaj kompanije Apple iPhone XS poseduje displej fizičke rezolucije 1125x2436px. Ipak, njegova softverska, odnosno CSS rezolucija iznosi 414x896px. To praktično znači da se na iPhone XS telefonu svaki CSS piksel predstavlja korišćenjem devet hardverskih piksela.

Iako je postojanje CSS rezolucije veoma zgodno kada se kreira layout ili optimizuje tekst, najbolji (najoštrij) prikaz slika na displejima velike gustine podrazumeva iskorišćenje svih dostupnih hardverskih piksela. Na kraju, sve ovo implicira da se medija upiti za odabir slike odgovarajuće veličine ne mogu isključivo kreirati testiranjem širine vidnog polja. Stoga u priču o medija upitima uvodimo još nekoliko kriterijuma:

- resolution
- min-resolution
- max-resolution

### resolution CSS tipovi

CSS poznaje tri različite ključne reči koje je moguće koristiti prilikom formiranja uslova unutar medija upita: `resolution`, `min-resolution` i `max-resolution`. Ovakve ključne reči se odnose na gustinu piksela displeja, što je hardverska karakteristika koja se drugačija naziva **rezolucija**.

Rezolucija se može iskazati na nekoliko načina, odnosno korišćenjem različitih jedinica:

- **dpi** (*dots per inch*) – broj tačaka po inču
- **dpcm** (*dots per centimeter*) – broj tačaka po centimetru
- **dppx** (*dots per pixel*) – broj tačaka po pikselu

Jedinica `dppx` posebno je korisna, zato što omogućava definisanje rezolucije displeja kao odnosa softverskih i hardverskih piksela. Broj tačaka po pikselu (*dots per pixel*) zapravo predstavlja broj hardverskih piksela unutar jednog CSS piksela.

Sada kada znamo za resolution CSS tipove, nije teško dodatno optimizovati medija upite za dobijanje responsive pozadinskih slika:

```
@media screen and (max-width: 414px) {
  #hero {
    background-image: url("tokyo-414w.jpg");
  }
}
@media only screen and (min-resolution: 2dppx) and (max-width: 414px) {
  #hero {
    background-image: url("tokyo-414w@2.jpg");
  }
}
@media only screen and (min-resolution: 3dppx) and (max-width: 414px) {
  #hero {
    background-image: url("tokyo-414w@3.jpg");
  }
}
```

Za maksimalnu širinu vidnog polja od 414px sada su dodata još dva medija upita, unutar kojih se proverava minimalna rezolucija displeja. Prvi medija upit aktiviraće se na svim uređajima sa vidnim poljem maksimalne širine 414px. Pri tome je neophodno da odnos između hardverskih i CSS piksela bude manji od 2. Preostala dva medija upita takođe se aktiviraju za CSS širinu vidnog polja do 414px, ali samo onda kada je odnos između hardverskih i CSS piksela 2 ili veći i 3 ili veći, respektivno.

### Više varijanti jedne iste img slike

Kreiranje responsive slika koje se HTML dokumentima dodaju korišćenjem `img` elementa nešto je drugačije nego pristup koji je upravo iskorišćen za optimizaciju pozadinskih slika. Osnovna razlika se ogleda u nemogućnosti upotrebe CSS-a za definisanje većeg broja varijanti jedne iste slike. Naime, putanja do `img` slike definiše se u HTML kodu, pa tako nema mogućnosti za njenu izmenu korišćenjem CSS jezika. Ipak, rešenje nudi sam HTML jezik:

- `srcset` i `sizes` HTML atributi na `img` elementu;
- `picture` HTML element.

Prvo će biti prikazan primer korišćenja `srcset` i `sizes` atributa na `img` elementu:

```

```

Iako `img` element u ovakvom obliku na prvi pogled može delovati vrlo zbunjujuće, sve funkcioniše veoma jednostavno. Krenućemo od atributa `srcset`.

Atribut **`srcset`** koristi se kako bi se definisao skup različitih verzija jedne iste slike. U primeru je takvih verzija ukupno četiri. Svaka verzija slike definiše se nazivom fajla praćenim odrednicom koja definiše fizičku širinu slike u pikselima.

Korišćenjem atributa **`sizes`** definišu se smernice koje sugerišu web pregledaču koju verziju slike je najbolje koristiti, a sve u zavisnosti od postavljenog uslova. Tako atribut `sizes` omogućava definisanje neke vrste medija upita, i to unutar samog `img` elementa. Nakon uslova koji se definiše unutar zagrada, navodi se širina slike koja bi najviše odgovarala različitim stanjima koje pokriva definisani uslov. Na primer, ukoliko razmotrimo prikazani `img` element, kada širina vidnog polja nije veća od 414px, web pregledaču se sugeriše da je najbolje koristiti sliku fizičke širine 414px. Web pregledač zatim, na osnovu različitih verzija koje su definisane atributom `srcset`, bira najprikladniju verziju slike. Naravno, to je prva definisana varijanta slike, fizičke širine od 414px.

Ukoliko iz nekog razloga web pregledač na podržava `srcset` atribut, biće učitana slika koja je definisana standardnim `src` atributom.

Upravo prikazani primer ilustrovao je kreiranje responsive `img` slike, pri čemu se za formiranje uslova koristila širina vidnog polja. Kada je optimizaciju slike potrebno vršiti na osnovu gustine piksela, odnosno odnosa između fizičkih i CSS piksela, moguće je koristiti nešto drugačiji oblik `img` elementa:

```

```

Ovoga puta su vrednosti `srcset` atributa navedene na nešto drugačiji način. Nakon naziva fajla koji predstavlja jednu verziju slike, navodi se vrednost koja predstavlja odnos između hardverskih i CSS piksela. Na taj način se različite verzije jedne iste slike vezuju za gustinu piksela displeja. Takođe, u ovakvoj konstrukciji nema potrebe za korišćenjem atributa `sizes`, pošto su i putanje i uslovi sadržani unutar vrednosti atributa `srcset`.

Za kraj ove lekcije biće ilustrovan još jedan pristup koji je moguće koristiti za dobijanje responsive slika koje su sastavni deo HTML strukture (*čitaj: nisu pozadinske*). Pristup podrazumeva korišćenje `picture` HTML elementa.

HTML element `picture` zapravo je omotač koji unutar sebe može sadržati jedan `img` element i nijedan ili više `source` elemenata. Korišćenjem `source` elemenata, definišu se različite verzije jedne slike. Primer korišćenja `picture` elementa može da izgleda ovako:

```
<picture>
  <source srcset="tokyo-414w.jpg" media="(max-width: 414px)">
  <source srcset="tokyo-768w.jpg" media="(max-width: 768px)">
  <source srcset="tokyo-1024w.jpg" media="(max-width: 1024px)">
  <source srcset="tokyo-2048w.jpg" media="(max-width: 2048px)">
  
</picture>
```

Na ovaj način postignut je identičan efekat kao i prvim primerom korišćenja `srcset` atributa na `img` elementu.

Unutar `picture` elementa nalaze se četiri `source` elementa, od kojih svaki poseduje `srcset` i `media` atribut. Atribut `srcset` definiše putanju do slike, a atribut `media` uslov koji je potrebno zadovoljiti kako bi se slika definisana atributom `srcset` prikazala. Na kraju, unutar `picture` elementa nalazi se i standardni `img` element koji će biti iskorišćen ukoliko web pregledač na podržava `picture` element.

#### Da li koristiti `picture` element ili `img` sa atributom `srcset`?

Iako na prvi pogled proizvode identičan efekat, pristupi koji podrazumevaju korišćenje `picture` elementa i elementa `img` sa atributom `srcset` poseduju jednu značajnu razliku.

Korišćenjem `picture` elementa, definiše se obaveza koju web pregledač mora ispoštovati, što nije slučaj kod pristupa koji podrazumeva korišćenje `srcset` atributa na `img` elementu. Drugim rečima, pristup koji podrazumeva korišćenje `srcset` atributa na `img` elementu definiše **smernice** za odabir odgovarajuće varijante jedne iste slike. Odabir konkretne verzije slike koja će biti učitana zavisi od brojnih faktora. Na primer, ukoliko web pregledač već poseduje sliku veće rezolucije unutar svog keša, on će takvu verziju i koristiti, bez obzira na to što možda neka druga verzija slike više odgovara datoj situaciji.



Sa druge strane, korišćenje `picture` elementa podrazumeva drugačije ponašanje. Web pregledač mora odabrati varijantu slike sa uslovom koji najviše odgovara trenutnim osobinama uređaja.

Zbog upravo iznete osobine, može se reći da je pristup koji podrazumeva korišćenje `img` elementa sa `srcset` atributom bolji izbor kada je potrebno dobiti responsive sliku čije se različite verzije razlikuju isključivo po veličini. Ipak, na webu se neretko javlja potreba za različitim kadriranjem jedne slike u zavisnosti od veličine i odnosa stranica displeja. U takvim situacijama je korišćenje `picture` elementa idealan izbor, zato što garantuje da će najbolja verzija slike biti upravo ona koja će se korisniku i prikazati.

## Rezime

- Adekvatno rukovanje slikama i njihova optimizacija unutar HTML dokumenata su apsolutni imperativ.
- Unutar HTML dokumenata, slike se mogu dodati korišćenjem `img` elementa ili CSS svojstva `background` ili `background-image`.
- Loša optimizacija slika na webu, pored lošeg korisničkog doživljaja, može prouzrokovati i veoma loše performanse, odnosno suviše dugo vreme učitavanja stranice.
- Prilikom pripreme slika za web, potrebno je voditi računa o formatu, veličini fajla i fizičkoj veličini slike izraženoj u pikselima.
- Tri su standardna formata slika za koje je garantovano da će ispravno funkcionisati na svim web pregledačima: JPG, PNG i GIF.
- Nepisano pravilo nalaže da je veličina od 200 KB maksimum za sliku koja se postavlja online.
- Veličina slike u pikselima treba da odgovara prostoru koji će slika zauzeti na stranici.
- Kako bi se `img` slika prilagodila veličini elementa unutar koga se nalazi, dovoljno je `width` postaviti na 100%.
- Kako `img` slika ne bi mogla da se poveća preko svoje fizičke veličine, koristi se svojstvo `max-width` sa vrednošću 100%.
- Kako bi pozadinska slika pratila promenu elementa nad kojim je definisana, moguće je iskoristiti CSS svojstvo `background-size` i njegove vrednosti `contain` i `cover`.
- Kako bi se postigao optimalan prikaz slika na svim uređajima i uštedeli resursi, napredni pristup kreiranja responsive slika podrazumeva korišćenje većeg broja verzija jedne iste slike.
- Definisanje više verzija jedne iste pozadinske slike postiže se korišćenjem medija upita.
- `resolution`, `min-resolution` i `max-resolution` su ključne reči koje je moguće koristiti prilikom definisanja medija upita, a odnose se na gustinu piksela displeja, što je hardverska karakteristika koja se drugačija naziva rezolucija.
- Rezolucija se može iskazati na nekoliko načina, odnosno korišćenjem različitih jedinica: `dpi`, `dpcm`, `dppx`.
- `dppx` jedinica omogućava definisanje rezolucije displeja kao odnosa softverskih i hardverskih piksela.
- Definisanje više varijanti jedne iste `img` slike moguće je postići na dva načina: korišćenjem `srcset` i `sizes` HTML atributa na `img` elementu i upotrebom `picture` HTML elementa.