

Viewport

- Pre nego što uđemo dublje u strukturu, pojasnićemo termin **viewport**, ili u prevodu *prozor dokumenta*.
- Viewport je prozor kroz koji se posmatra sadržaj. Viewport kod računara je prozor browsera. Uglavnom se pojavljuju kontrole za scroll (sa desne strane), ukoliko je sadržaj duži od same visine viewpota. Mogu se pojaviti i horizontalne scroll kontrole ukoliko je širina sadržaja veća.
- Viewport može biti manji od ukupne dimenzije ekrana, ukoliko korisnik smanji prozor browsera. Kod mobilnih telefona, dimenzije viewpota su uglavnom preko celog ekrana.

Piksel nije piksel

- Danas, sa razvojem sve boljih i boljih ekrana, pogotovo na mobilnim uređajima, menja se i sam način rada sa njima i prikaz elemenata. Pojavljuju se termini ***Device pixel*** i ***CSS pixel***.

- *Device pixel* ili na srpskom pikseli uređaja su „pravi“ pikseli i realna rezolucija našeg monitora.
- Ako uzmemo za primer monitor koji prikazuje sliku u 1024x768 piksela, to znači da se na njemu nalazi 1024 tačaka po horizontalnoj i 768 tačaka po vertikalnoj osi.

U CSS-u, ako napišemo:

```
#element {  
  width:500px;  
}
```

Postavili smo određenom elementu 500px širine. Na malopre pomenutom monitoru ovakav element bi zauzeo nešto manje od polovine širine ekrana.

Ali šta se dešava ako korisnik promeni nivo uveličanja ekrana (engl. *Zoom*)? Na primer, na 200%

Šta bi se onda desilo sa našim elementom od malopre? Svakako, povećao bi se duplo i zauzeo bi nešto manje od cele širine ekrana ($500 \times 2 = 1000$).

Ali šta se dešava sa pikselima? Pikseli uređaja se nisu promenili (i dalje je ekran širok 1024px), ali se sada CSS pikseli (CSS pixel) razlikuju od piksela uređaja, od kojih su dva puta veći.

Kod računara, uveličanje – umanjenje slike se retko kad koristi, najčešće kada je možda sitan tekst i korisnik uveliča zbog čitljivosti.

Iz ugla piksela, pikseli uređaja se tada ne menjaju, ali se CSS pikseli povećavaju i možemo reći da se (uslovno) smanjuje širina monitora u CSS pikselima (manje ih staje jer su veći).

Viewport na mobilnim uređajima

- Šta je sa mobilnim i tablet uređajima? Kod njih se situacija dosta komplikuje. Ekрани takvih uređaja u suštini ne razlikuju mnogo od monitora računara, ali je njihova namena drugačija i drugačije se pristupa prikazu na njima. Prvi problem koji treba premostiti je sama dimenzija ekrana koja je kod telefona znatno manja od računara i manje elemenata može se prikazati odjednom.
- Dakle, jasno je da je prozor dokumenta (viewport) preuzak da bi adekvatno prikazao CSS layout sajta (koji je možda 960px širok). Logično rešenje je razdvajanje viewporta na dva, i njegovo virtuelno povećanje. Razlikujemo ***visual viewport*** i ***layout viewport***.

visual viewport i layout viewport

George Cummins u jednom tekstu kaže:

Zamislite layout viewport kao veliku sliku koja ne menja veličinu niti oblik. Sada zamislite i manji okvir kroz koji posmatrate tu veliku sliku. Taj manji okvir je okružen nekim neprovidnim materijalom koji zaklanja celu veliku sliku, osim jednog dela koji se vidi kroz okvir. Deo slike koji vidite kroz manji okvir je visual viewport. Možete se udaljiti od velike slike dok držite manji okvir. Odjednom ćete videti veći deo ili celu veliku sliku ukoliko se dovoljno udaljite (zoom out). Ili, ukoliko se približite, videćete još manji deo velike slike. Možete čak rotirati okvir, ali dimenzije i oblik velike slike (layout viewport) se nikad ne menjaju.

visual viewport i layout viewport

- Sigurno ste i sami puno puta na svom telefonu pomerali neki sajt, uveličavali, smanjivali prikaz. Tada se upravo dešava ono što je rečeno u ovom prethodnom slajdu.
- Ekran telefona je naš manji okvir koji držimo, dok je sam sajt ta velika slika koju posmatramo. Kada zumiramo analogno je tome kao da smo se približili velikoj slici (sajtu).
- I na kraju, **visual viewport** je deo stranice koji je trenutno prikazan na ekranu, dok je *layout viewport* ceo prikaz sajta.

visual viewport i layout viewport

- U realnim uslovima, šta se onda dešava sa prikazom na telefonima? Koliko je širok layout viewport?
- **On se razlikuje od browsera do browsera** i zavisi od uređaja. Na primer, Android browser (uglavnom) postavlja širinu layout viewports na 800px. Sama širina ekrana je možda realno 320px, ali će telefon učitati stranicu kao da je širok 800px, što znači da će u startu CSS pikseli biti manji od realnog piksela uređaja.

Gustina piksela

- Povrh svega, ovde se komplikacija sa mobilnim prikazom ne završava. U poslednje vreme sve je više uređaja sa većom gustinom piksela kod kojih se ne poklapaju pikseli uređaja (koje smo pominjali do sad) i realni pikseli na ekranu.
- A tu su i CSS pikseli.

Gustina piksela

- Na primer, telefon Apple iPhone 7 ima ekran čije su realne dimenzije 750x1334 piksela. Ali, to ne znači da je kod tog telefona broj **piksela uređaja** tačno toliki.
- Zbog velike gustine piksela, on se ponaša kao da je njegov ***device-width* 375px**, a ***device-height* 667px**, odnosno polovina od realne dimenzije. To je zato što ima odnos 2/1 px i veću gustinu ekrana koja iznosi 326ppi i realan piksel je previše sitan za udoban rad kada bi odnos bio 1/1.

Gustina piksela

- Uprošćeno rečeno, po horizontali dva piksela ekrana se koriste za prikaz jednog piksela uređaja, koji se opet razlikuju od CSS piksela.
- Šta dobijamo time? Sa većom gustinom i upotrebom više realnih piksela za jedan piksel uređaja dobijamo kvalitetniju sliku, oštrije fontove i sl.

Gustina piksela

- Ako pogledamo još jedan primer, telefon HTC One, ima ekran čije su dimenzije realno 1080x1920 piksela, ali je device width 360x640px.
- Sa razvojem boljih i kvalitetnijih ekrana sve je više ovakvih primera. Na linku <http://mydevice.io/devices/> možete naći tabelu koja prikazuje trenutni popularne telefone i tablet uređaje i karakteristike njihovih ekrana. Na istom sajtu, na početnoj stranici možete testirati svoj uređaj, tj. prikazati info o svom uređaju.

Gustina piksela

- Ali, sve u svemu, detalji sa gustinom ekrana i sa ovim novijim (boljim) telefonima ne treba da nas brinu.
- Što se tiče browsera, malopre pomenuti HTC One se ponaša kao telefon sa ekranom 360x640px (iako realno ima 1920x1080), što je isto ili slično kao mnogi drugi telefoni sa različitim realnim rezolucijama.

Gustina piksela

- Većina uređaja ima slične dimenzija po pitanju device pixela. To znači da se svi oni ponašaju slično.
- Na primer, LG G4 i Sony Xperia P se ponašaju, što se tiče browsera, kao da imaju istu rezoluciju od 360x640 piksela, iako su njihovi ekrani znatno različiti i prvi će svakako korisniku pružiti bolju sliku.

Piksel, piksel i piksel

Uzevši u obzir sve ranije navedeno dolazimo do konstatacije:

REALNI PIKSEL EKRANA \neq PIKSEL UREĐAJA \neq CSS PIKSEL

Odnosno, ne možemo poistovetiti i pričati o jednim “pikselima” u sferi web dizajna. Konkretno:

Piksel, piksel i piksel

- **Realni piksel ekrana** - fizički, realni piksel ekrana, odnosno najmanja, nedeljiva tačka ekrana. Jedini pravi piksel u ovoj grupi. Ipak, on nas ovde najmanje zanima.
- **Piksel uređaja (device-width)** – piksel koji uređaj prijavljuje browseru. Uglavnom je kod starijih uređaja: piksel uređaja = realni piksel ekrana, dok se kod novih pojavljuje odnos 1.5/1, 2/1 ili 3/1 i sl. Dakle, može biti istovetan realnim pikselima uređaja, ali uglavnom nije.
- **CSS piksel** – relativni piksel koji zavisi od nivoa uvećanja (zooma). Ovaj piksel se koristi u CSS opisima.

Piksel, piksel i piksel

- Sve ovo verovatno deluje konfuzno, ali srećom postoji elegantno rešenje. Apple je uveo pre nekoliko godina meta tag ***viewport***, koji su kasnije svi prihvatili.
- Pomoću njega možemo da utičemo na prozor dokumenta i prikaz. Dovoljno je da postavimo:
`<meta name="viewport" content="width=device-width">`

Piksel, piksel i piksel

```
<meta name="viewport" content="width=device-width">
```

Ovim tagom govorimo browseru da prilikom učitavanja stranice izjednači:

- ***width*** - predstavlja širinu stranice, odnosno ranije pomenuti **layout viewport i CSS piksele**
- ***device-width*** – predstavlja širinu ekrana u **pikselima uređaja**.

Ukoliko postavimo ovakav meta tag, mi praktično izjednačavamo širine i postavljamo piksele na svoje mesto.

Primer

- Hajde da pogledamo jedan primer.
- Na ranije pomenutom HTC One telefonu, rezolucija jeste 1080x1920, ali znamo da se ponaša kao 360x640.
- **360 piksela je širina ekrana** (u vertikalnom položaju)

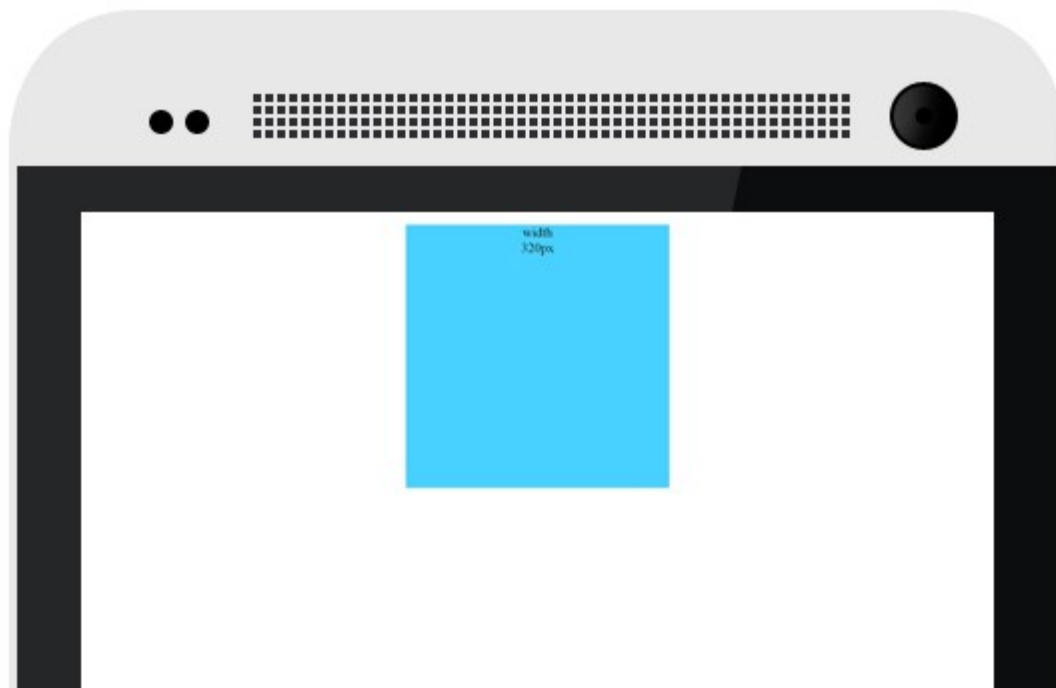
Postavićemo u body delu HTML-a:

```
<div id="test">  
    <p>width 320px</p>  
</div>
```

A u povezanom CSS-u:

```
body {  
    margin:0;  
}  
  
#test {  
    width:320px;  
    height:320px;  
    background:#3CF;  
    margin:0 auto;  
    text-align:center;  
}
```

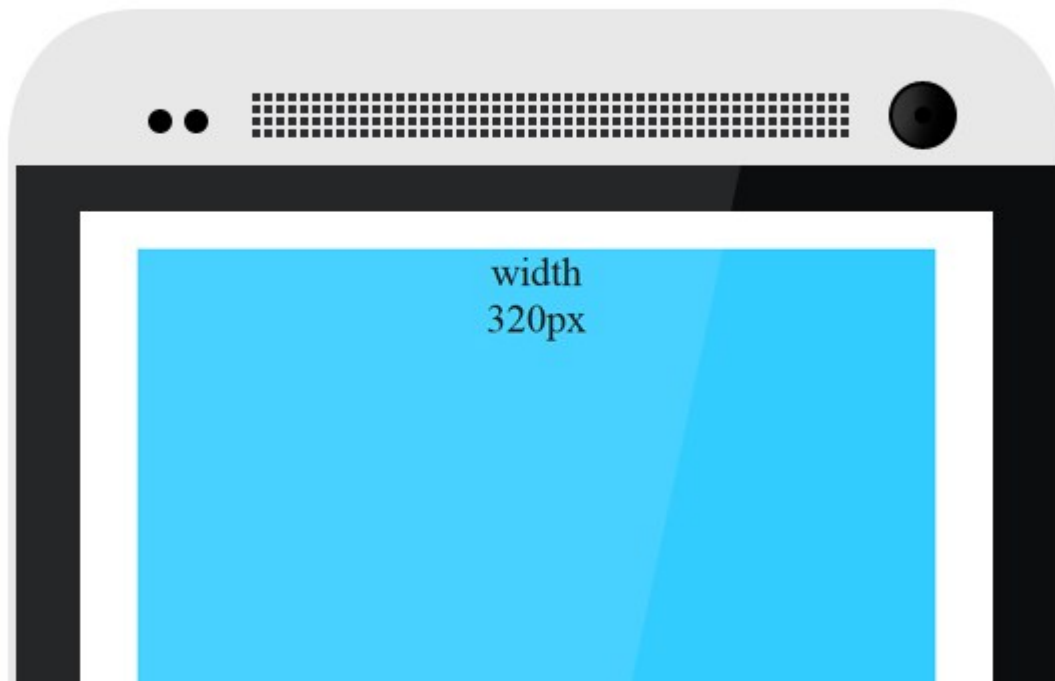
Kreirali smo plavi kvadrat na vrhu i horizontalnoj sredini stranice. Prikaz na računalu nas sada ne zanima, već samo prikaz na telefonu. Na pomenutom HTC One telefonu će prikaz biti ovakav:

*temy*

Što je drastično manje od onoga što želimo da postignemo. Ako dodamo sada:

`<meta name="viewport" content="width=device-width">`

Prikaz će se promeniti i dobićemo ovo:



Academy

- Sada smo najzad dobili željeno. CSS dimenzije, odnosno **CSS pikseli odgovaraju pikselima uređaja**. Kvadrat je 320px širok u CSS pikselima, a toliko prostora zauzima i u pikselima uređaja (širina ekrana je 360px i zbog toga je prostor levo/desno od po 20px).
- Pritom, u primeru **bez** meta viewport taga, ukoliko zumiramo do samog kvadrata, ostaće dosta prostora desno i levo i zato će bespotrebno moći da se pomera viewport desno i levo van sadržaja. Sa druge strane, **sa** meta viewport tagom nema tih problema.

- Ali (uvek postoji neko ali), treba biti oprezan. Ukoliko je sajt u potpunosti optimizovan za mobilne uređaje, odnosno ukoliko smo kreirali posebne media upite za različite rezolucije i prilagodili elemente, onda ima smisla postavljati meta viewport tag. Ukoliko sajt ne sadrži media upite, onda nemojte postavljati viewport tag.
- Dakle, u ovom primeru je dostupna širina uređaja max 360 piksela i svakako da moramo postaviti odgovarajuće media upite, ako postavljamo meta viewport tag.