

Nastavni predmet	SKRIPTNI JEZICI I WEB PROGRAMIRANJE
Naslov cjeline	CSS – uređivanje web sjedišta
Naslov jedinice	LV20: CSS izgled stranice (max-width, overflow, float, inline-block, align)
Učenci	Marin Šupljika, 3.c

CILJ VJEŽBE

Učenik će demonstrirati i prikazati načine za poravnanje elemenata na stranici, odabrati i primijeniti način za poravnanje elemenata, odnosno njihovog sadržaja, prilagoditi vrstu poravnanja elementa, upotrebljavati svojstva za prikaz i poravnanje elemenata.

PRIPREMA ZA VJEŽBU

Prisjetite se i odgovorite pisano:

1. Koja je osnovna razlika između *inline* i *block-level* elemenata?

Block-level elementi zauzimaju cijelu širinu svog roditeljskog spremnika i uvijek počinju u novom redu. Primjeri block-level elemenata su <div>, <p>, <h1> do <h6>, <section>, itd.

Inline elementi ne zauzimaju cijelu širinu svog roditeljskog spremnika, već se smještaju jedan pored drugog. Također, ne počinju u novom redu, a primjer inline elementa je , <a>, , itd.

2. Što predstavlja *padding*, a što *margin*?

Padding je unutarnji razmak između sadržaja elementa i njegovih rubova. Padding se koristi za stvaranje prostora unutar samog elementa, oko njegovog sadržaja.

Margin je vanjski razmak između ruba elementa i drugih susjednih elemenata. Margin se koristi za raspored i udaljenost između različitih elemenata na stranici.

IZVOĐENJE VJEŽBE

CSS max-width

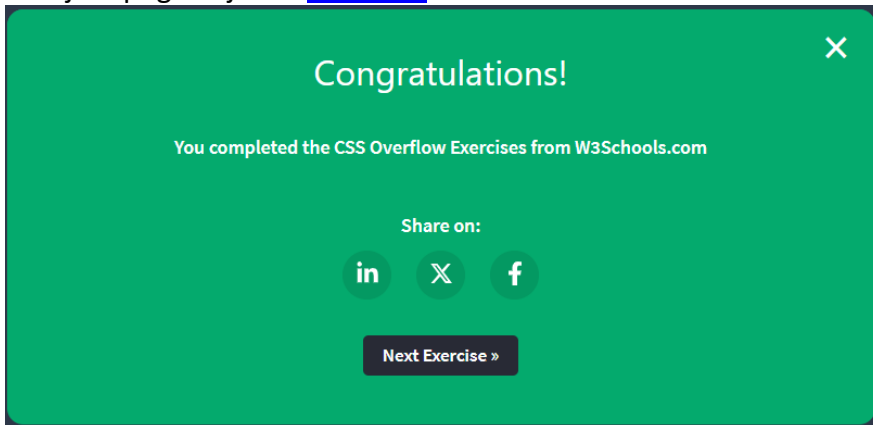
- *block-level* element uvijek zauzima **punu raspoloživu širinu**.
- Postavljanje širine *block-level* elementa spriječit će da se on raširi do rubova svog spremnika. U tom slučaju, margine se mogu postaviti na *auto* kako bi se **vodoravno centriralo element unutar njegovog spremnika**. Pri tom će element zauzeti određenu/navedenu širinu, a **preostali će prostor biti podijeljen jednako** između dviju margina.
- U slučaju da je prozor preglednika manji od širine elementa, može se pojaviti problem sa elementom <div> u smislu da će **preglednik dodati vodoravnu kliznu traku** na stranicu. U takvoj

situaciji upotreba *max-width* može **poboljšati rukovanje preglednika s malim prozorima**, što je važno kod izrade sjedišta koja trebaju biti upotrebljiva i na manjim uređajima.

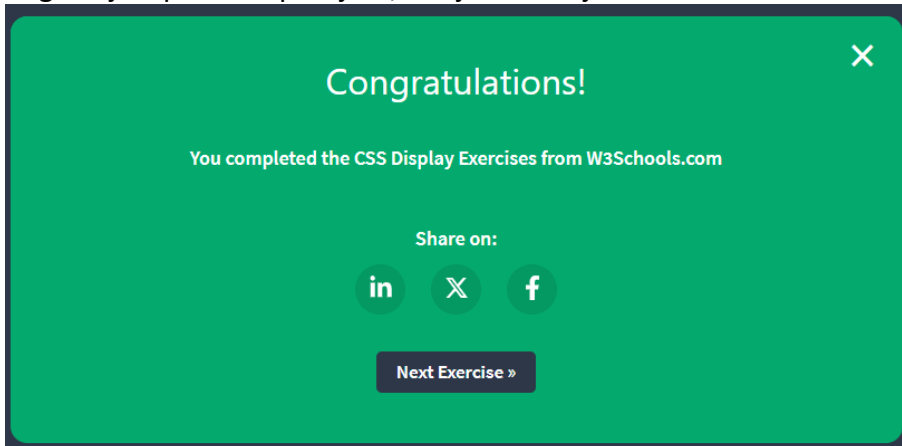
- Pogledajte i provjerite kako funkcionira primjer na [poveznici](#).
-

CSS overflow (preljev)

- CSS svojstvo *overflow* kontrolira što se događa sa sadržajem koji **ne stane u neko područje**. Ovo svojstvo određuje hoće li se **odsjeći sadržaj ili će se dodati klizne trake**. Funkcionira samo za blok elemente specificiranih dimenzija i ima slijedeće vrijednosti: *visible* – podrazumijevano, sadržaj koji se prelijeva ne odsjeca se i prikazuje se izvan okvira elementa, *hidden* – preljev se odsjeca, ostatak sadržaja će biti skriven, *scroll* – preljev se odsjeca i dodaje se klizna traka kako bi se vidio ostatak sadržaja, *auto* – slično kao *scroll*, ali dodaje kliznu traku samo ako je potrebno. Primjere pogledajte na [poveznici](#).



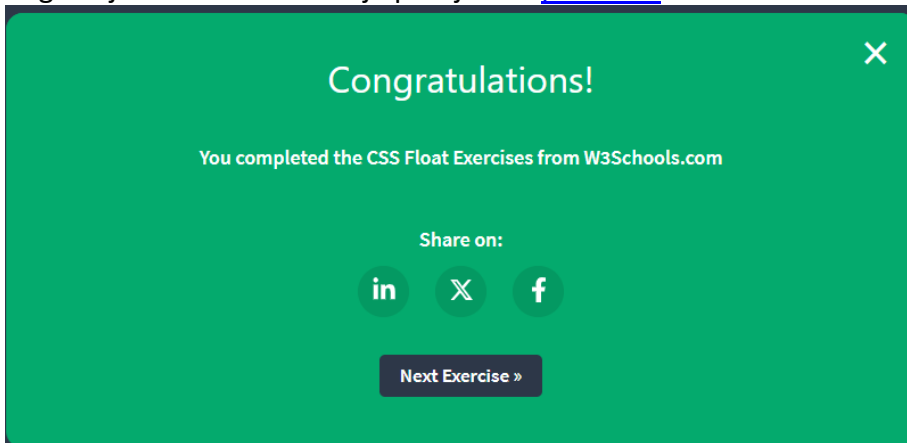
- Svojstva *overflow-x* i *overflow-y* određuju hoće li se primijeniti preljev sadržaja samo vodoravno, okomito ili oboje: *overflow-x* određuje što će biti sa lijevim/desnim rubom sadržaja, *overflow-y* određuje što će biti sa gornjim/donjim rubom sadržaja.
- Pogledajte i proučite primjere, te riješite tri vježbe zadane na stranici ([vježbe](#)).



CSS svojstva float i clear

- CSS svojstvo *float* određuje kako bi element trebao “plutati”. CSS svojstvo *clear* određuje koji elementi mogu plutati osim obrisanog elementa i na kojoj strani.
- Svojstvo *float* koristi se za **pozicioniranje i oblikovanje sadržaja** (npr. može omogućiti da slika pluta lijevo u odnosu na tekst u spremniku). To svojstvo može imati vrijednosti: *left* – element pluta na lijevom dijelu svojeg spremnika, *right* – element pluta na desnoj strani svojeg spremnika, *none* - element ne pluta (ispisuje se točno tamo gdje se u tekstu pojavljuje - podrazumijevano), *inherit* – element nasljeđuje vrijednost smještaja od roditeljskog elementa.

- Svojstvo *float* može se koristiti kako bi omotalo (*wrap*) tekst oko slika.
- U normalnoj situaciji *div* elementi biti će ispisani jedan iznad drugog, a ako koristimo *float* možemo dozvoliti elementima da plutaju **jedan pored drugog**.
- Pogledajte kako funkcioniraju primjeri na [poveznici](#).



- Svojstvo *clear* određuje koji elementi mogu plutati uz obrisani element te na kojoj strani. Ovo svojstvo može imati vrijednosti: *none* – omogućuje plutajuće elemente na obje strane (podrazumijevano), *left* – plutajući elementi nisu dozvoljeni na lijevoj strani, *right* – plutajući elementi nisu dozvoljeni na desnoj strani, *both* – plutajući elementi nisu dozvoljeni ni lijevo ni desno, *inherit* – vrijednost za *clear* element nasljeđuje od svojeg roditelja. Najčešći način za korištenje svojstva *clear* je **nakon što se primijeni svojstvo *float*** na neki element.
- Prilikom brisanja svojstva *float*, trebao bi se *clear* podudarati sa *float*, npr. ako je element plutajući lijevo, treba primijeniti *clear* na lijevoj strani. Plutajući element će nastaviti plutati, ali obrisani element pojaviti će se ispod njega na mrežnoj stranici.
- Ako je element viši od elementa koji ga sadrži te ako je plutajući, on će se prelijati izvan spremnika, kako bi prevladali taj problem možemo mu dodati *overflow: auto;* :

```
.clearfix {
  overflow: auto;
}
```

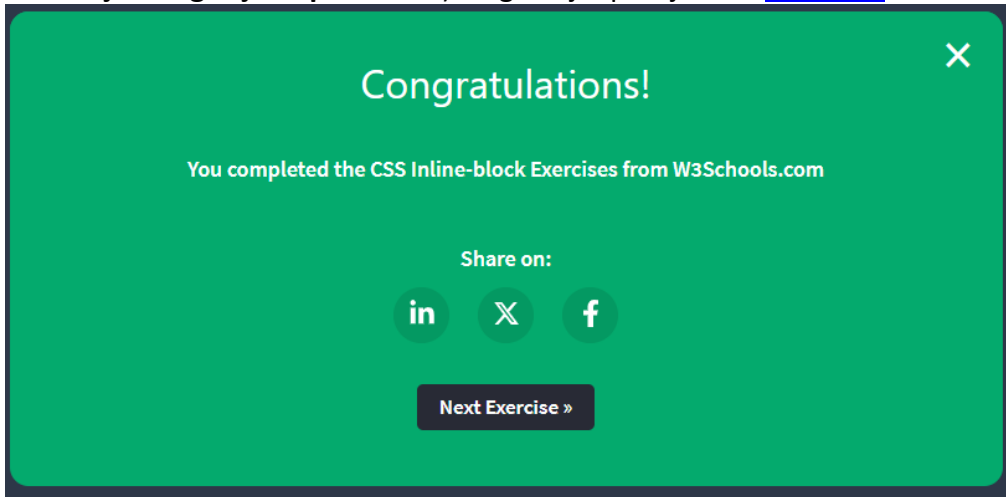
- *overflow: auto clearfix;* radi dobro u slučajevima u kojima je moguće kontrolirati margine i odmak (*padding*), inače se mogu vidjeti klizne trake. **Novi i moderniji *clearfix hack*** je sigurniji za upotrebu, pa je primijenjen na većini mrežnih stranica:

```
.clearfix::after {
  content: "";
  clear: both;
  display: table;
}
```

- Primjere pogledajte na [stranici](#). Na [stranici](#) pogledajte i proučite primjere za *float*: mrežu okvira sa sadržajima jednake širine, niz slika smještenih jedna uz drugu, okvire jednake visine, primjer za *flexbox*, navigacijski izbornik, kompletan izgled stranice

CSS inline-block

- Usporedi li se sa *display: inline*, glavna razlika je da *display: inline-block* omogućava postavljanje širine i visine elementa, a poštuju se i gornje i donje margine/padding.
- Usporedi li se sa *display: block*, glavna razlika je da *display: inline-block* ne dodaje novu liniju nakon elementa, tako da se element može smjestiti uz druge elemente.
- ***display: inline-block* se uobičajeno koristi za ispis liste vodoravno umjesto okomito** (npr. kod stvaranja navigacijskih poveznica). Pogledajte primjere na [poveznici](#).



CSS align (vodoravno i okomito poravnanje)

- **Kako bi vodoravno centralizirali *block* element (npr. <div>), koristi se *margin: auto***; Postavljanje širine elementa spriječit će njegovo rastezanje izvan rubova njegovog spremnika. Element će tada zauzeti specificiranu širinu, a preostali će prostor biti podijeljen jednako između dviju margina (**centrirano poravnanje nema učinka ako *width* nije postavljeno ili je postavljeno na 100%**):

```
.center {  
margin: auto;  
width: 60%;  
border: 3px solid #73AD21;  
padding: 10px;  
}
```

- Kako bi samo **centralizirali tekst unutar elementa**, koristimo *text-align: center*;
- Kako bismo **centralizirali sliku**, potrebno je postaviti lijevu i desnu marginu na *auto* i napraviti *block* element:

```
img {  
display: block;  
margin-left: auto;  
margin-right: auto;  
width: 40%;  
}
```

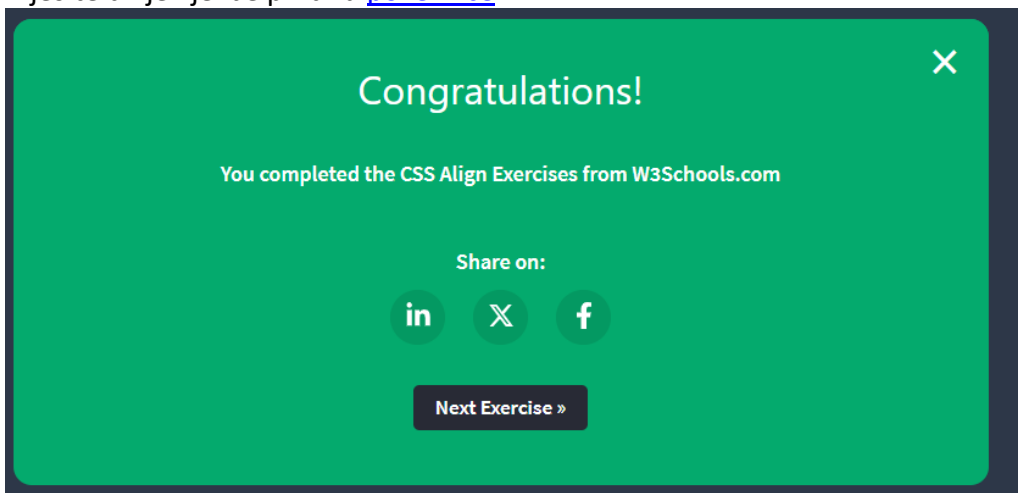
- mogućnost za poravnanje elemenata je korištenje: *position: absolute*:

```
.right {  
position: absolute;  
right: 0px;  
width: 300px;  
border: 3px solid #73AD21;  
padding: 10px;  
}
```

- Apsolutno pozicionirani elementi pomaknuti su iz normalnog tijeka i mogu prekrivati druge elemente.
- Mogućnost za poravnanje elemenata je i korištenje svojstva *float* (pri tom treba na preporučeni način kontrolirati element koji je viši od elementa koji ga sadrži):

```
.right {  
float: right;  
width: 300px;  
border: 3px solid #73AD21;  
padding: 10px; }  
}
```

- **Okomito centriranje** može se u CSS-u riješiti na više načina. Jednostavno rješenje je koristiti *top* i *bottom padding* (npr. *padding: 50px 0;*). Kako bi se centriralo i okomito i vodoravno, može se upotrijebiti *padding* i *text-align: center*; Druga je mogućnost koristiti svojstvo *line-height* sa vrijednošću jednakoj svojstvu *height* (pogledajte primjer na [stranici](#)).
- **Okomito centriranje** može se izvesti pomoću *position* i *transform*. Ako *padding* i *line-height* nisu opcije, rješenje je koristiti pozicioniranje i svojstvo *transform*. Centriranje se može izvesti i pomoću *flexbox* (pogledajte primjere na [stranici](#)).
- Riješite dvije vježbe pri dnu [poveznice](#).



PONAVLJANJE I PRAKTIČNA VJEŽBA:

1. U online [editoru](#) ili editoru koji uobičajeno koristite provježbajte naučeno.
- 2.

Vrste promet. znakova				
Uvodni tekst...				
Z1	Z2	Z3
☺	Opis skupine Z1			
Opis skupine Z2				☺
☺	Opis skupine Z3...			

Napravite mrežnu stranicu na kojoj ćete nakon kratkog uvodnog teksta o prometnim znakovima navesti (kao izbornik) skupine prometnih znakova. Tekst treba biti napisan tako da o svakoj vrsti ima napisan jedan do dva odlomka. Odgovarajuće slike trebaju biti pozicionirane naizmjenično lijevo-desno. Dakle, izgled stranice treba biti otprilike kao prikaz lijevo. Izbornik treba omogućiti pomak stranice na odabranu vrstu. Svi odlomci teksta trebaju imati isto oblikovanje.

Napomena: ove praktične vježbe kad je to moguće izvodi svaki učenik samostalno.