

Nastavni predmet	SKRIPTNI JEZICI I WEB PROGRAMIRANJE
Naslov cjeline	CSS – uređivanje web sjedišta
Naslov jedinice	Vježba: CSS izgled stranice (max-width, overflow, float, inline-block, align)

Teo Šiletić, Filip Škrabec, 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 celu širinu roditeljskog elementa i počinju na novom redu (npr. <div>, <p>).

Inline elementi zauzimaju samo prostor svog sadržaja i ne prelaze u novi red (npr. , <a>).

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

Padding je prostor unutar elementa, između njegovog sadržaja i ivica (granica) elementa.

Margin je prostor spoljašnji, između ivica elementa i drugih elemenata oko njega.

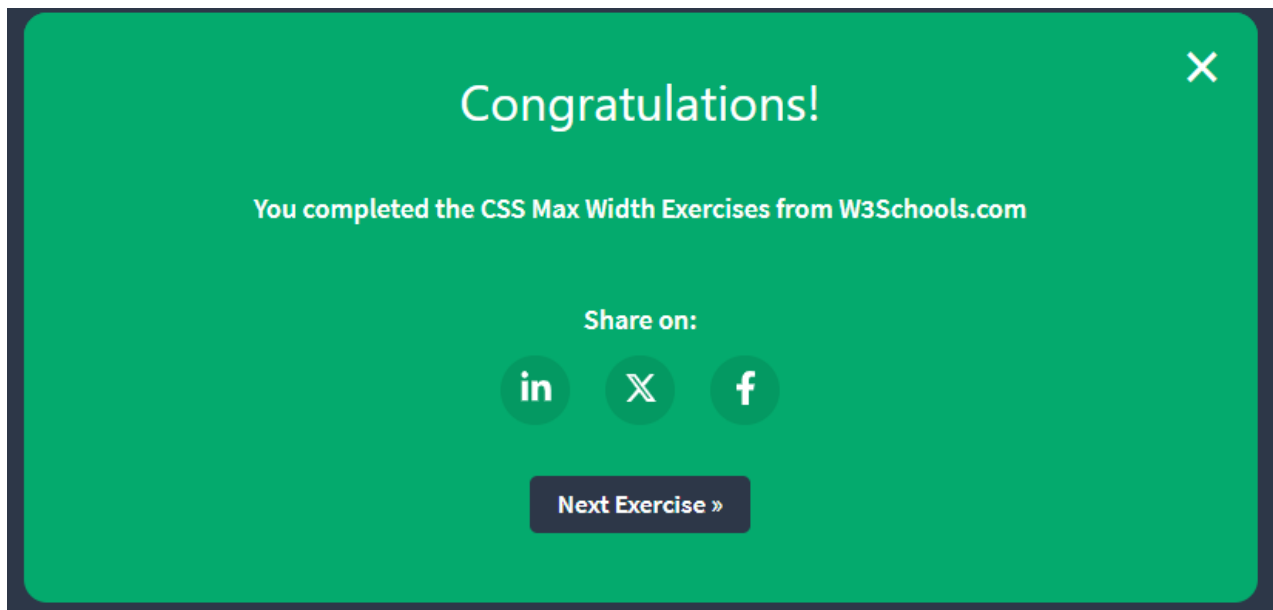
IZVOĐENJE VJEŽBE

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 centralo 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

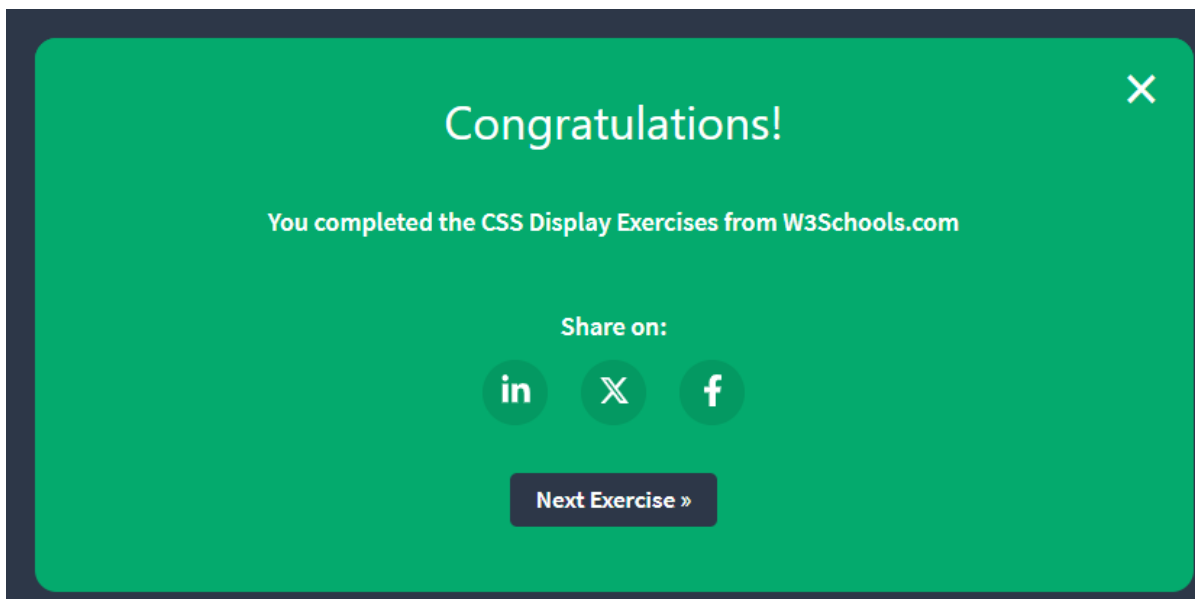
važno kod izrade sjedišta koja trebaju biti upotrebljiva i na manjim uređajima.

- 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%;
}
```

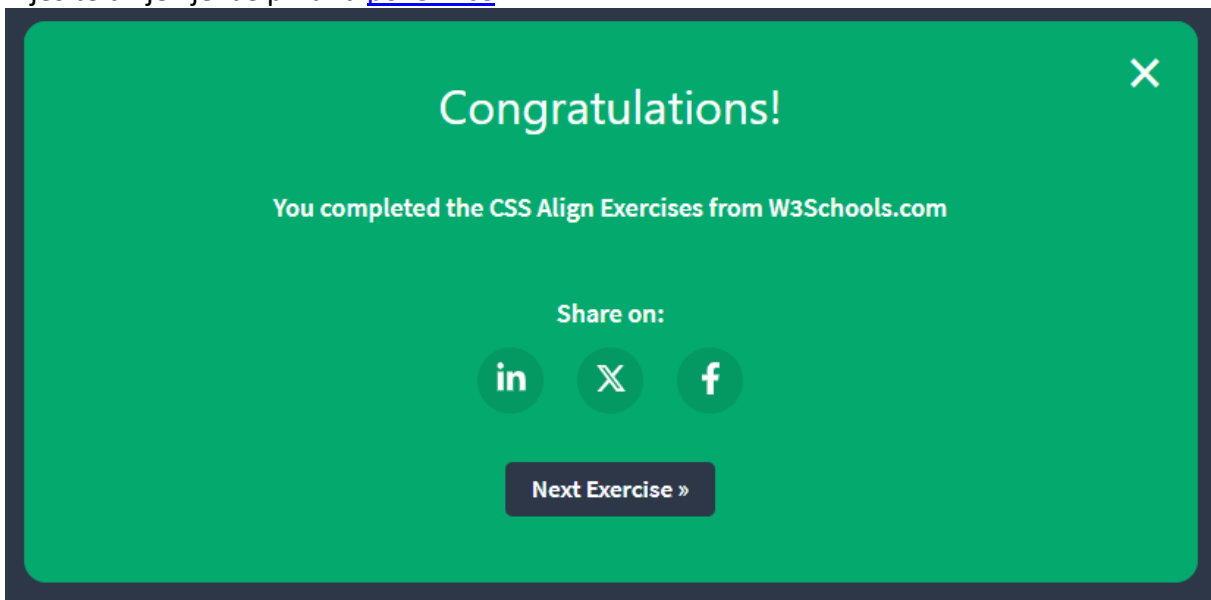
- mogućnost za poravnanje elemenata je korištenje: *position: absolute*:

```
.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.

Uvod u prometne znakove

Prometni znakovi su važan dio cestovne sigurnosti. Razlikujemo znakove opasnosti, izričitih naredbi i obavijesti.

Znakovi Opasnosti Znakovi Izričitih Naredbi Znakovi Obavijesti

Znakovi Opasnosti



Znakovi opasnosti upozoravaju vozače i pješake na potencijalne opasnosti na cesti. Ovi znakovi obično prikazuju ikone ili simbole koji označavaju specifične opasnosti, poput zavoja, klizavih cesta, proklizavanja, ili opasnih križanja. Važno je obratiti pažnju na ove znakove jer oni omogućuju vozaču da unaprijed prilagodi brzinu i ponašanje kako bi izbjegao nesreće. Znakovi opasnosti uvijek zahtijevaju veću pozornost i opreznost na cesti.



HTML

```

<!DOCTYPE html>
<html lang="hr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Prometni Znakovi</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <h1>Uvod u prometne znakove</h1>
    <p>Prometni znakovi su važan dio cestovne sigurnosti. Razlikujemo znakove opasnosti, izričitih naredbi i obavijesti.</p>
  </header>

  <nav>
    <ul>
      <li><a href="#Znakovi0">Znakovi Opasnosti</a></li>
      <li><a href="#ZnakoviIN">Znakovi Izričitih Naredbi</a></li>
      <li><a href="#ZnakoviOB">Znakovi Obavijesti</a></li>
    </ul>
  </nav>

  <section id="Znakovi0">
    <h2>Znakovi Opasnosti</h2>
    <div class="content">
      

```



```
<p>Znakovi opasnosti upozoravaju vozače i pješake na potencijalne opasnosti na cesti. Ovi znakovi obično prikazuju ikone ili simbole koji označavaju specifične opasnosti, poput zavoja, klizavih cesta, proklizavanja, ili opasnih križanja. Važno je obratiti pažnju na ove znakove jer oni omogućuju vozaču da unaprijed prilagodi brzinu i ponašanje kako bi izbjegao nesreće. Znakovi opasnosti uvijek zahtijevaju veću pozornost i opreznost na cesti.</p>
```

```
</div>
```

```
</section>
```

```
<section id="ZnakoviIN">
```

```
<h2>Znakovi Izričitih Naredbi</h2>
```

```
<div class="content">
```

```

```

```
<p>Znakovi izričitih naredbi obvezuju vozače na određeno ponašanje na cesti, a njihovo nepoštivanje može rezultirati kaznom. Ovi znakovi uključuju zabrane (npr. zabrana skretanja, zabrana ulaska u određenu ulicu) i naredbe poput obavezne vožnje u određenom smjeru ili usmjerenja na određeni način. Ovi znakovi osiguravaju sigurnost i učinkovito upravljanje prometom, jer jasno definiraju što vozači smiju i ne smiju činiti u određenim situacijama.</p>
```

```
</div>
```

```
</section>
```

```
<section id="ZnakoviOB">
```

```
<h2>Znakovi Obavijesti</h2>
```

```
<div class="content">
```

```

```

```
<p>Znakovi obavijesti pružaju vozačima korisne informacije o uvjetima na cesti, kao što su informacije o smjeru, udaljenostima, ili oznakama za parkirališta. Ovi znakovi mogu uključivati nazive gradova, kilometražu, ili upozorenja o dolaznim usmjerenjima ili izlazima. Iako ovi znakovi nisu obavezni za vozače, oni su ključni za orijentaciju i planiranje rute, posebno na nepoznatim ili dužim putovanjima.</p>
```

```
</div>
```

```
</section>
```

```
</body>
```

```
</html>
```

CSS

```
body {  
  font-family: 'Arial', sans-serif;  
  background-color: #fafafa;  
  margin: 0;  
  padding: 0;  
  color: #333;  
}
```

```
header {  
  background-color: #2a9d8f;
```

```

    color: white;
    text-align: center;
    padding: 40px 20px;
    font-size: 1.8em;
}

header h1 {
    margin-bottom: 10px;
}

header p {
    font-size: 1.1em;
    line-height: 1.6;
}

nav {
    background-color: #264653;
    padding: 15px 0;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}

nav ul {
    list-style-type: none;
    padding: 0;
    margin: 0;
    text-align: center;
}

nav ul li {
    display: inline-block;
    margin: 0 20px;
}

nav ul li a {
    color: white;
    text-decoration: none;
    font-weight: bold;
    font-size: 1.2em;
    transition: color 0.3s ease, transform 0.3s ease;
}

nav ul li a:hover {
    color: #e9c46a;
    transform: scale(1.1);
}

section {
    margin: 30px auto;
    padding: 30px;
    background-color: white;
    border-radius: 8px;
}

```

```
max-width: 900px;
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

section h2 {
  font-size: 2.2em;
  margin-bottom: 20px;
  color: #2a9d8f;
}

.content {
  display: flex;
  justify-content: space-between;
  align-items: center;
  gap: 20px;
  margin-top: 20px;
}

.content img {
  max-width: 45%;
  border-radius: 8px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}

.content p {
  flex: 1;
  font-size: 1.1em;
  line-height: 1.6;
  color: #555;
}

.image-left {
  order: -1;
}

.image-right {
  order: 1;
}
```