

Nastavni predmet	SKRIPTNI JEZICI I WEB PROGRAMIRANJE
Naslov cjeline	CSS – uređivanje web sjedišta
Naslov jedinice	Vježba: CSS - prikaz, pozicioniranje i prozirnost

## CILJ VJEŽBE

Učenik će demonstrirati način i posljedice skrivanja elementa na mrežnoj stranici, odabrati način prikaza elemenata, prilagoditi vrstu pozicioniranja elementa, upotrebljavati svojstva za pozicioniranje elemenata, demonstrirati i primjenjivati različita pozicioniranja, prilagoditi prozirnost elementa.

## PRIPREMA ZA VJEŽBU

Odgovorite pisano:

- ▶ 1. Što znate o elementu `<div>`, a što o elementu `<span>`?
- ▶ 2. Koji se element koristi za postavljanje slike na stranicu? Napišite primjer koda.

## IZVOĐENJE VJEŽBE

### CSS prikaz (`display`)

- Svojstvo `display` najvažnije je CSS svojstvo **za upravljanje oblikovanjem** (layout). To svojstvo opisuje **hoće li se i kako** element ispisati.
- Svaki HTML element ima podrazumijevanu vrijednost za prikaz ovisno o tipu elementa. Podrazumijevana vrijednost prikaza **za većinu elemenata** je `block` ili `inline`.
- Elementi tipa `block-level` **uvijek započinju u novom retku i zauzimaju punu raspoloživu širinu** stranice. Primjeri ovih elemenata su: `<div>`, `<h1>` – `<h6>`, `<p>`, `<form>`, `<header>`, `<footer>`, `<section>`
- Elementi tipa `inline` **ne počinju u novom retku i zauzimaju tek toliko širine koliko je potrebno**. Primjeri takvih elemenata su: `<span>`, `<a>`, `<img>`
- `display: none;` obično se koristi u JavaScriptu kako bi se **sakrili i pokazali elementi bez da ih se briše ili ponovno kreira**.
- Podrazumijevana vrijednost elementa može se promijeniti tako da im se **promijeni tip**. Uobičajeni primjer za to je kad se `<li>` element koristi za vodoravni izbornik, pa pišemo:  
`<style> li { display:inline; } </style>`

- Pri tome je važno naglasiti da to samo mijenja **kako se element prikazuje**, a ne mijenja tip elementa.
- **Skriavanje elemenata** – `display:none` ili `visibility:hidden`?
  - Skriavanje elemenata može se postići postavljanjem svojstva `display` na `none`. U tom će slučaju element biti skriven, a stranica će se prikazati kao da element ne postoji. Takav element **ne zauzima mjesto**.
  - `visibility:hidden` također skriva element, no u ovom će slučaju tako skriven element **zauzimati mjesto** i time utjecati na izgled stranice.
- **Pogledajte i proučite primjere, te riješite četiri vježbe zadane na stranici ([vježbe](#)).**

## CSS svojstvo `position`

- CSS svojstvo `position` određuje način pozicioniranja koji će se primijeniti na element. Postoji pet različitih vrijednosti za `position`: `static`, `relative`, `fixed`, `absolute` i `sticky`. Nakon što je određena i izabrana vrijednost za to svojstvo, element se dodatno može pozicionirati pomoću svojstva `top`, `bottom`, `left` i `right`. Ovisno o vrijednosti svojstva `position` i ove će vrijednosti funkcionirati različito.
  - `position: static`; Svi HTML elementi podrazumijevano su pozicionirani kao `static` pa na njih ne utječu svojstva `top`, `bottom`, `left` i `right`. Takvi elementi u stvari nisu pozicionirani, odnosno uvijek su pozicionirani samim tijekom elemenata na stranici.
  - `position: relative`; određuje da je element pozicioniran relativno na svoju normalnu poziciju. Postavljanje svojstava `top`, `right`, `bottom` i `left` na relativno pozicioniran element ima za posljedicu da će biti odmaknut od svojeg normalnog položaja, pri čemu se preostali sadržaj neće prilagoditi kako bi popunio bilo kakvu prazninu koju je element ostavio zbog svojeg pomaka.
  - `position: fixed`; element koji je pozicioniran na ovaj način, pozicionira se relativno prema tzv. `viewportu`, što znači da uvijek ostaje na istom mjestu bez obzira na `skrolanje` stranice. Svojstva `top`, `right`, `bottom` i `left` koriste se kako bi pozicionirali element. Fiksiran element ne ostavlja prazninu u stranici gdje bi inače normalno bio smješten.
  - `position: absolute`; Element će biti pozicioniran relativno u odnosu na najbliže pozicioniranog pretka (kad apsolutno pozicioniran element nema pozicioniranih predaka, on koristi tijelo dokumenta i kod skrolanja se miče uzduž stranice). (Pozicioniran element je onaj koji nije statičan.)
  - `position: sticky`; Element će biti pozicioniran na temelju pozicije korisničkog `skrolanja`. Takav se element prebacuje između relativnog i fiksnog, ovisno o poziciji `skrola`. Pozicioniran je relativno sve dok se ne dostigne dana `offset` pozicija u `viewport-u`, nakon toga se "zalijepi" na lokaciju (kao `position:fixed`). Safari preglednik zahtjeva `-webkit-` prefix. Potrebno je specificirati najmanje jedno od svojstava `top`, `right`, `bottom` ili `left` kako bi `sticky` pozicioniranje funkcioniralo.
- Preklapanje elemenata: prilikom pozicioniranja elemenata, oni mogu prekrivati druge elemente. **Svojstvo `z-index`** određuje stog (`stack`) međusobni poredak elemenata. Element može imati pozitivan ili negativan poredak: ako je vrijednost za `z-index` `-1`, bit će smješten iza drugog elementa, npr.:
 

```
img { position: absolute; left: 0px; top: 0px; z-index: -1; }
```

- Element sa većom vrijednošću uvijek je ispred (iznad) elementa sa manjom vrijednošću poretka. Ako se dva pozicionirana elementa preklapaju bez da je specificiran *z-index*, element koji je u HTML kodu pozicioniran posljednji bit će prikazan na vrhu.
- Pozicioniranje teksta preko slike može biti *top left*, *bottom left*, *centered*, *top right*, *bottom right*.
- **Riješite pet vježbe pri dnu [stranice](#).**

### CSS prozirnost (opacity/transparency)

- CSS svojstvo *opacity* određuje prozirnost elementa. Može imati vrijednost od 0.0 do 1.0. Što je vrijednost manja, transparentnost je veća. Svojstvo *opacity* često se koristi zajedno sa *:hover* selektorom kako bi se transparentnost/prozirnost promijenila prilikom prelaska pokazivača miša (*mouse over*): `img { opacity: 0.5; } img:hover { opacity: 1.0; }`
- Transparentni okviri: Kada se svojstvo *opacity* koristi kako bi dodali prozirnost pozadini elementa, svi elementi kojima je on predak nasljeđuju istu prozirnost, pa tekst unutar potpuno prozirnog okvira može postati nečitak. Kako bi se spriječilo da se prozirnost nasljeđuje, potrebno je koristiti RGBA vrijednosti boja koje uključuju i podatak za prozirnost, npr.: `div { background: rgba(76, 175, 80, 0.3) /* zelena pozadina sa 30% prozir. */ }`
- **Riješite dvije vježbe pri dnu [stranice](#).**

### PONAVLJANJE I PRAKTIČNA VJEŽBA:

1. U online [editoru](#) provježbajte naučeno.
2. Prikažite neki tekst upozorenja (npr. "ZABRANJEN PROLAZ" i sl.) u transparentnom okviru, čitko i preko cijele širine ekrana. Primijenite pri tome neki od **google fontova** pogodan za naslove (pogledajte sličan primjer na stranici w3schools).
3. Dodajte stranici dva niza od po tri slike. Sve slike trebaju biti **jednake** veličine i neizobličene. Prikaz treba reagirati na *mouse-over* **promjenom prozirnosti**. Na slikama mogu biti vama zanimljiva prijevozna sredstva ili građevine i sl. Ovisno o veličini ekrana slike se prikazuju jedna pored druge ili jedna ispod druge (mobilni uređaji). Preko svake slike stavite njezin kratak opis, odnosno napišite što prikazuje.
4. Na vrh stranice dodajte vodoravni izbornik koji će prikazivati ovisno o odabiru korisnika, tekst ili slike

**TEKST**

**SLIKE**

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

