

Vježba 19: CSS prikaz, pozicioniranje i prozirnost

Antonio Matek, Mihael Modrić 3.B

Odgovorite pisano u bilježnicu:

- ▶ 1. Što znate o elementu `<div>`, a što o elementu ``?

`<div>`: Ovaj element se koristi za definiranje odjeljka ili dijela dokumenta s određenim stilovima. Obično se koristi za grupiranje blokovskih elemenata kako bi se primijenili stilovi na cijelu grupu. `<div>` je blokovski element, što znači da automatski zauzima cijelu širinu roditeljskog elementa, osim ako se ne promijeni stilovima CSS-a.

``: Ovaj element se koristi za označavanje dijela teksta ili elemenata unutar teksta s posebnim stilovima. Za razliku od `<div>`, `` je inline element, što znači da zauzima samo prostor koji je potreban za sadržaj koji sadrži, bez prekida linije.

- ▶ 2. Koji se element koristi za postavljanje slike na stranicu? Napišite primjer.

Za postavljanje slike na stranicu koristi se element ``. Primjer:

```

```

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: ``, `<a>`, ``.

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 prikaza ovisno o njegovom tipu. 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: ``, `<a>`, ``
- `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 `` 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.
- Skrivanje elemenata – `display:none` ili `visibility:hidden`? Skrivanje 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](#)).

1 Exercise: Antonio Matek, Mihael Modrić 3.B

Hide the <h1> element. It should still take up the same space as before.

```
<style>
h1 {
  visibility: hidden;
}
</style>

<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph</p>
  <p>This is a paragraph</p>
</body>
```

2 Exercise: Antonio Matek, Mihael Modrić 3.B

Hide the <h1> element. It should NOT take up any space.

```
<style>
h1 {
  display: none;
}
</style>

<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph</p>
  <p>This is a paragraph</p>
</body>
```

3Exercise: Antonio Matek, Mihael Modrić 3.B

Display the list items as inline elements.

```
<style>
li {
  display: inline;
}
</style>

<body>
<ul>
  <li>Apple</li>
  <li>Orange</li>
  <li>Pear</li>
</ul>
</body>
```

4Exercise: Antonio Matek, Mihael Modrić 3.B

Display the elements as block elements.

```
<style>
strong {
  display: block;
}
</style>

<body>
<h1>This is a Heading</h1>
<p>This is a <strong>paragraph</strong></p>
</body>
```

CSS svojstvo position

- **CSS svojstvo position određuje tip metode pozicioniranja koja 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 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 prilagođen odmaknuto 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](#).

# 1 Exercise: Matek, Modrić 3.B

Position the <h1> element to always be 50px from the top, and 10px from the right, r

```
<style>
h1 {
 position: fixed;
 top: 50px;
 right: 10px;
}
</style>

<body>
 <h1>This is a heading</h1>
 <p>This is a paragraph</p>
 <p>This is a paragraph</p>
</body>
```

## 2Exercise: Matek, Modrić 3.B

Position the <h1> element 50px from the top, relative to its normal position.

```
<style>
h1 {
 position: relative;
 top: 50px;
}
</style>

<body>
 <h1>This is a heading</h1>
 <p>This is a paragraph</p>
 <p>This is a paragraph</p>
</body>
```

# 3Exercise: Matek, Modrić 3.B

Position the `<h1>` element 50px from the top, relative to the HTML page.

```
<style>
h1 {
 position: absolute;
 top: 50px;
}
</style>

<body>
 <h1>This is a heading</h1>
 <p>This is a paragraph</p>
 <p>This is a paragraph</p>
</body>
```



# 4Exercise: Matek, Modrić 3.B

Position the `<h1>` element 50px from the top, by referring to its class name.

```
<style>
.myheader {
 position: relative;
 top: 50px;
}
</style>

<body>
 <h1 class="myheader">This is a heading</h1>
 <p>This is a paragraph</p>
 <p>This is a paragraph</p>
</body>
```

## 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 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) /* Green background with 30% opacity */
}
```
- Riješite dvije vježbe pri dnu [stranice](#).

# 1Exercise: Matek, Modrić 3.B

Use CSS to set the transparency of the image to 50%.

```
<style>
img {
 opacity: 0.5;
}
</style>

<body>

</body>
```

# 2Exercise: Matek, Modrić 3.B

Use CSS to "remove" the transparency of the image when the user mouse over the image.

```
<style>
img {
 opacity: 0.4;
}
img:hover {
 opacity: 1.0;
}
</style>

<body>

</body>
```

## PONAVLJANJE I PRAKTIČNA VJEŽBA:

1. U online [editoru](#) provježbajte naučeno.
2. Napišite neki tekst u transparentnom okviru, ali tako da bude čitak. Primijenite pri tome neki od google fontova pogodan za naslove (pogledajte primjer na stranici).

Ovaj blok koda definira font Roboto za latinično pismo s normalnim stilom i težinom 400.

Ovaj font se koristi za latinicu i ostale znakove koji nisu pokriveni drugim unicode-range definicijama. Ovdje se koristi verzija fonta koja se nalazi na Google Fonts i može se učitati putem URL-a.

```
/* latin */
@font-face {
 font-family: 'Roboto';
 font-style: normal;
 font-weight: 400;
 font-display: swap;
 src: url(https://fonts.gstatic.com/s/roboto/v30/KFOmCnqEu92Fr1u4mxK.woff2) format('woff2');
 unicode-range: U+0000-00FF, U+0131, U+0152-0153, U+02BB-02BC, U+02C6, U+02DA, U+02DC, U+0304, U+0308, U+0329, U+2000-206F, U+2074, U+20AC, U+2122, U+2191, U+2193, U+2212, U+2215, U+FEFF, U+
}
```

URL: <https://fonts.googleapis.com/css?family=Roboto&display=swap>

```
<style>
```

```
.transparentni-okvir {
```

```
 background-color: rgba(255, 255, 255, 0.7); /* Bijela boja s 70% prozirnosti */
```

```
 padding: 20px;
```

```
 border-radius: 10px;
```

```
}
```

color: #ff69b4;: Postavlja boju teksta na ružičastu koristeći HEX kod boje. Možete promijeniti ovu vrijednost da biste postavili bilo koju boju po vašem izboru.

font-style: italic;: Dodaje nagib teksta kako bi izgledao ukošeno.

text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);: Dodaje sjenu teksta kako bi se istaknuo na pozadini. Ovdje su specificirane vrijednosti za horizontalni i vertikalni pomak sjene te zamaglivanje i boja sjene.

3. Napravite mrežnu stranicu na kojoj ćete prikazati dva niza od po tri slike. Sve slike trebaju biti jednake veličine i neizobličene. Prikaz treba reagirati na mouse-over promjenom

```
.slika {
```

```
 width: 100%; /* Sve slike su jednake veličine */
```

```
 height: auto; /* Očuvaj omjer širine i visine */
```

```
 transition: opacity 0.3s ease; /* Prijelaz prozirnosti */
```

```
}
```

```
.slika:hover {
```

```
 opacity: 0.7; /* Prozirnost na hover */
```

transition: opacity 0.3s ease;: Ovo svojstvo primjenjuje prijelaz na promjenu svojstva

slika:hover { opacity: 0.7; }: Ovo pravilo primjenjuje se kada korisnik pređe mišem preko elementa s klasom .slika. Kada se to dogodi, prozirnost slike postavlja se na 0.7 (70%), čime postaje nešto prozirnija.