

1. (4 boda)

Na priloženoj slici ispod teksta zadatka dan je izgled HTML stranice čiji je sadržaj isključivo jedna tablica. Tablica sadrži sliku s adrese https://www.fer.unizg.hr/images/50029121/FER_logo_3.png. Za rješenje zadatka priložite HTML kod koji definira izgled ovakve stranice, također definirajući i njenu minimalnu strukturu (nužni elementi stranice). Širina slike mora biti 200 pixela. Za sve podatke koji su nužni za definiranje stranice, a koji nisu navedeni u tekstu zadatka, odaberite proizvoljne vrijednosti.



2. (4 boda)







Zadana je sljedeća HTML stranica s nizom CSS izraza koji, među ostalim, ciljaju drugi element poredane liste. Odredite kojim redom će biti primijenjeni ti izrazi na taj drugi element liste. Rješenje navedite tako da u tekst polje navedete boje odvojene s točka zarez pri čemu selektor koji će "pobijediti" navodite prvi, npr.:

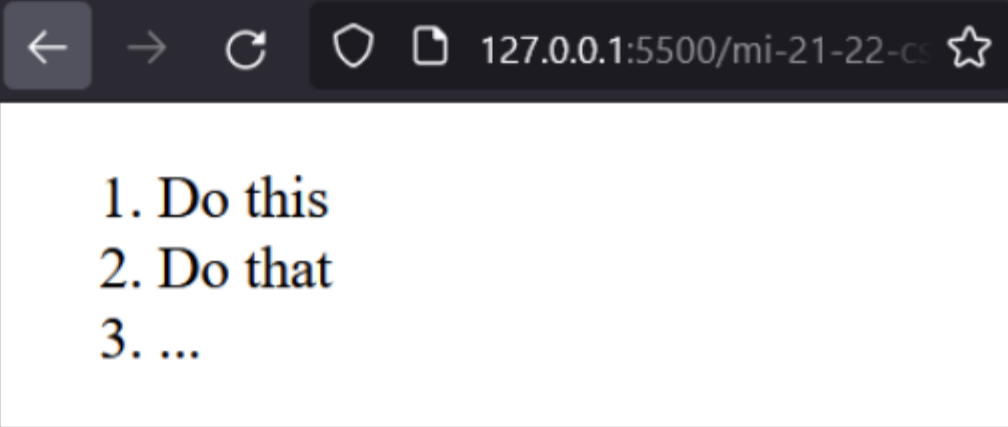
```
blue;green;gray;orange;red;yellow;
```

Tumačenje je sljedeće:

- drugi element će biti plave boje
- kada ne bi bilo prethodnog selektora, bio bi zelene boje,
- kada ne bi bilo prethodnih selektora, bio bi sive boje,
- kada ne bi bilo prethodnih selektora, bio bi narančaste boje,
- kada ne bi bilo prethodnih selektora, bio bi crvene boje,
- kada ne bi bilo prethodnih selektora, bio bi žute boje.

Preporuka: kako ne biste pogriješili neko slovo kod tipkanja, u tekst polje kopirajte primjer rješenja i drugačije poredajte boje. Pazite na `;` jer će zadatak biti ocijenjen automatski. Nemojte ništa drugo stavljati u tekst polje.

```
9 <style>
10     ol li {
11         color:  green;
12     }
13     .todos li {
14         color:  blue;
15     }
16     li {
17         color:  gray;
18     }
19     #item13{
20         color:  orange;
21     }
22     li {
23         color:  yellow;
24     }
25 </style>
26 <body>
27     <ol class="todos">
28         <li>Do this</li>
29         <li id="item13" class="important" style="color:  red;">Do that</li>
30         <li>...</li>
31     </ol>
32 </body>
```



3. (6 bodova)

Zadana su dva ulazna skupa podataka: skupa kategorija i skupa proizvoda. Napišite odsječak kôda u JavaScriptu kojim se određuje koja kategorija ima najviše proizvoda i koji tu informaciju ispisuje korisniku (na konzolu).

Ulazni skup kategorija je dan JSON datotekom **categories.json**:

```
[
  {"CategoryId": 1, "CategoryName": "IT Equipment"},
  {"CategoryId": 2, "CategoryName": "Drinks"},
  {"CategoryId": 3, "CategoryName": "Food"}
]
```

Napomena: atributi svake kategorije su redom: identifikator kategorije i naziv kategorije.

Ulazni skup proizvoda dan je JSON datotekom **products.json**:

```
[
  {"ProductName": "USB-C charger", "CategoryId": 1, "NumProd": 55},
  {"ProductName": "Monitor MT246", "CategoryId": 1, "NumProd": 22},
  {"ProductName": "Mineral water", "CategoryId": 2, "NumProd": 250}
]
```

Napomena: atributi svakog proizvoda su redom: naziv proizvoda, identifikator kategorije kojoj proizvod pripada i količina proizvoda u toj kategoriji.

Za navedene datoteke vidljivo je da kategorija "Drinks" ima 250 proizvoda te je time kategorija s najviše proizvoda. Program u ovome slučaju ispisuje na konzolu:

Category with most products: Drinks - 250

U programskom odsječku treba koristiti (već je napisana) funkciju **LoadData(fileName)** koja učitava datoteku s podacima u JSON-u i kao rezultat vraća obećanje (promise) sa sadržajem datoteke u JSON formatu.

```
//vec postoji - potrebno koristiti
async function LoadData(fileName){
    let promise = await fetch(fileName);
    if (!promise.ok) { throw new Error("Cannot load json file"); }
    else { return await promise.json();}
}
```

Napomene:

- za učitavanje datoteka categories.json i products.json potrebno je koristiti funkciju LoadData(). Nije dozvoljeno pisati vlastite pozive metode fetch
- u slučaju greške prekida se izvođenje programa, a sve greške prihvća u jednom glavnom catch bloku, te korisniku na konzolu (console) ispisuje "An error has occured and program stopped working".
- za svaku kategoriju koja nema pripadnih proizvoda postaviti NumProd na 0
- u slučaju da proizvodi sadrže identifikator nepostojeće kategorije, takve proizvode ignorirati i ne uključiti u rezultat
- ako dvije kategorije imaju isti broj proizvoda ispisati bilo koju od te dvije

4. (6 bodova)

Potrebno je napraviti **responzivno** zaglavlje (header) web-aplikacije kao na slici dolje.

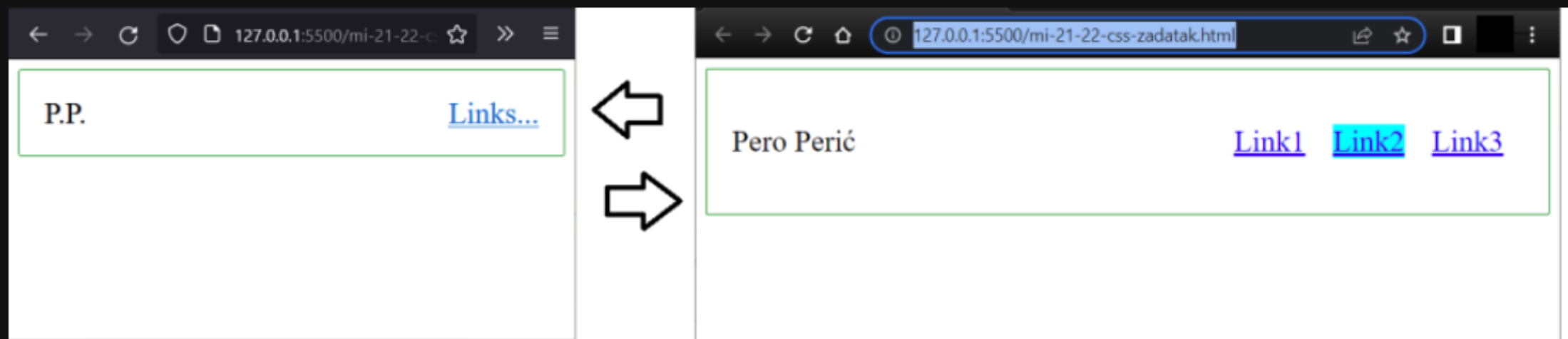
Na rezolucijama **manjim od 500px** umjesto imena prikazati inicijale, a umjesto Link1-3 prikazati jednu poveznicu (koja predstavlja *dropdown menu* koji nije potrebno implementirati). Poveznice Link1-3 izvedite pomoću neporedane liste i `li` elemenata (Links... ne treba, on je obična poveznica).

Ime (i inicijale) je potrebno zamijeniti **vlastitim imenom tj. inicijalima**. Veličina fonta u zaglavlju je **1.5 puta** veća od korijenske veličine fonta dokumenta.

Primijetite:

- Zaglavlje mora imati **obli rub zelene boje** (`green`) i sadržaj je odmaknut od ruba
 - konkretne vrijednosti u pixelima postavite otprilike
- Ime/inicijali i poveznice su:
 - smješteni na dva različita kraja zaglavlja (skroz lijevo i desno)
 - vertikalno ispravno poravnati
- Postoji horizontalni razmak između Link1 - Link3
- Kada se mišem pređe preko Link1-Link3 pozadina postane `aqua` boje
 - Links... ne treba imati nikakav efekt

Zadatak riješite s **jednom HTML datotekom** u koju ćete smjestiti i CSS. Predajte **sadržaj** (c/p) te datoteke u tekst polje.



5. (10 bodova)

Potrebno je napraviti web-stranicu (**HTML, CSS i JS datoteke**) koja omogućuje ostavljanje povratne informacije (feedback) korisnika putem formulara (form). Povratna informacija se sastoji od **imena korisnika, e-mail adrese korisnika i teksta** kojeg korisnik upisuje. Sučelje stranice prikazano je slikom:

Uz gumb za predaju povratne informacije (Submit), postoji i gumb za pohranu (Save Draft) kojim se skica povratne informacije može privremeno pohraniti u cilju kasnije predaje. Skica se nakon klika na Save Draft pohranjuje u **lokalni spremnik (local storage)**, te se **automatski učitava** prilikom ponovnog pokretanja stranice.

Važno: prilikom klika na Submit potrebno je obaviti validaciju sva tri tekstualna polja. Ako nisu popunjena i ako polje za e-mail nije formata x@y (pri čemu se x i y znakovni nizovi dugi barem jedan znak), korisnika je potrebno o tome obavijestiti validacijskim porukama.

Napomene:

- veličine tekstualnih polja postaviti po vlastitom nahođenju kako bi najbolje odgovarala slici
- attribute method, target, action itd. može se postaviti na vrijednosti po želji
- nazive polja lokalnog spremnika odaberite po želji
- rozi pravokutnik forme ima boju pink, te je od lijevog i desnog ruba odmaknut za 15 viewport jedinica u odnosu na širinu stranice; padding rozog pravokutnika postavite na 20px

Zadatak riješite s **tri datoteke: s po jednom HTML, CSS i JS datotekom**. Datoteke arhivirajte u **zip arhivu** i predajte je na sustav (uploadajte).

Flower Power Customer Feedback Form

How was your experience with the Flower Power web shop? We want to hear from you!

Name:

E-mail:

Feedback: