



FER3

Preddiplomski studij

Računarstvo

Razvoj programske podpore za web

Prva laboratorijska vježba — HTML i
CSS

Ak. god. 2020./2021.

1 Uvod

U sklopu laboratorijskih vježbi izradit ćemo online dućan za cvijeće i potrepštine za vrt. Laboratorijske vježbe će se nadovezivati jedna na drugu. U sklopu ove vježbe bit će potrebno na osnovu priloženih primjera izraditi novu stranicu koju predajete kao rješenje laboratorijske vježbe. Na samoj vježbi će biti potrebno izmijeniti pripremu, odnosno bit će potrebno dodati nove elemente stranici ili modificirati postojeće. Potrebno je izraditi stilove samo za velike ekrane odnosno ekrane šire od 960px (uobičajene širine ekrana).

Svi primjeri dostupni su na ovoj **poveznici**.

1.1 Razvojno okruženje

Preporučuje se korištenje editora Visual Studio Code.

Korisne ekstenzije:

- Alat za automatsko formatiranje kôda Beautify
- Alat za jednostavnije pisanje HTML kôda Emmet — već dostupan u editoru.

Za debugging koristite:

- Chrome DevTools
- Firefox Developer Tools

1.2 Organizacija projekta

Slika 1 prikazuje strukturu projekta. Sve datoteke (odnosno projekt) osim datoteka **order.css** te **order.html** priložene su na **poveznici**.

2 Priprema za laboratorijsku vježbu

Vaš je zadatak izraditi datoteke koje nedostaju, odnosno **order.css** te **order.html**. Možete mijenjati ostale datoteke ako smatrate da je to potrebno. Od vas se očekuje izrada stranice **order.html** koja je prikazana na slici 2. Pri tome je dozvoljeno koristiti komponente priložene u projektu (npr. možete koristiti **main.css** prilikom izrade stranice **order.html**). Smatra se da razumijete cijeli projekt i njegov sadržaj.

Na laboratoriju se očekuje modifikacija gotove pripreme u skladu sa zadatkom koji će vam biti zadan.

2.1 Kako pristupiti rješavanju pripreme?

Potrebno je izraditi vjernu repliku stranice **order.html** koja je prikazana na slici 2. Postoji više ispravnih rješenja (npr. nije bitno koristite li grid ili flex ako stranica izgleda kako je prikazano na slici 2). Pri tome je potrebno osigurati intuitivnu funkcionalnost stranice te usvajanje dobre programerske prakse, kao npr:

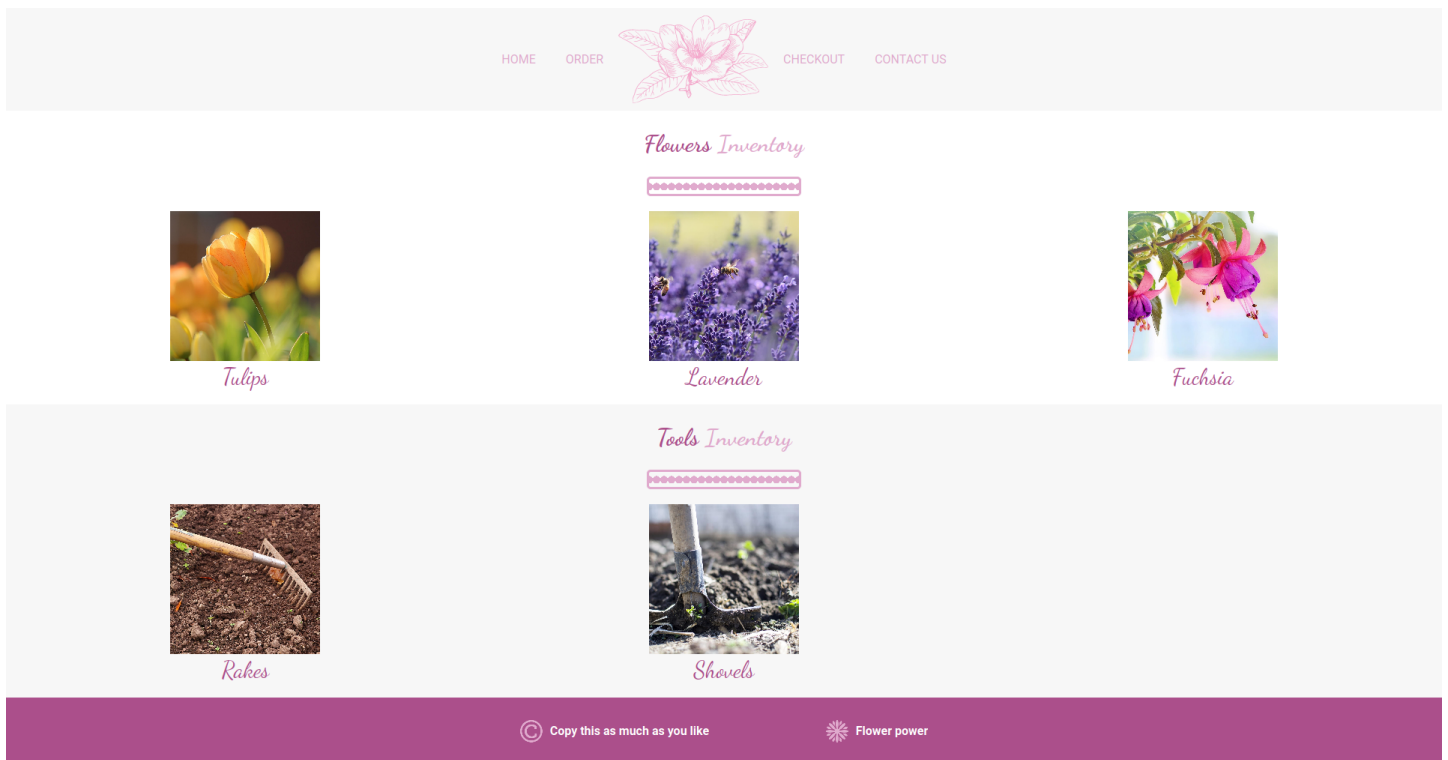
- Za stilove koristite samo klase, a ne HTML elemente.
- Horizontalne margine među slikama na stranici 2 nisu fiksno zadane — intuitivno je da su slike dalje jedna od druge na većim ekranima, a bliže na manjim.
- Sve slike prikazane na stranici **order.html** su fiksne veličine, no slike u folderu **images** su različitih veličina. Primijetite da se ne prikazuje cijela slika, već njen dio. Dodatno, prikazuje se centar slike.
- Manje pozornosti obratite na apsolutno pozicioniranje elemenata, a više na njihov relativni odnos. Iz primjera: tekst ispod slika je centriran ispod slike, naslov sekcije je horizontalno centriran u odnosu na viewport, itd.

```

root
├── fonts --> Fontovi
│   ├── DancingScript-Bold.woff
│   └── DancingScript-Bold.woff2
│
├── :
│
├── :
│
├── :
│   ├── Roboto-Thin.woff
│   └── Roboto-Thin.woff2
├── images --> Slike
│   ├── calendar-icon.png
│   └── chrysanthemum.jpg
│
├── :
│
├── :
│   ├── tulip.jpg
│   └── wild-flowers.jpg
├── index.html
├── order.html --> Vaša priprema
├── styles --> Vanjske knjižnice
│   ├── main.css
│   └── order.css --> Vaši stilovi
├── vendor --> Vanjske knjižnice
│   └── normalize.css --> Normalizira osnovne stilove neovisno pregledniku

```

Slika 1: Struktura projekta za 1. laboratorijsku vježbu

Slika 2: Prikaz stranice **order.html**

- Velika većina stilova se može primijeniti koristeći selektore specifičnosti 0|3|0 (IDs|Classes, attributes and

pseudo-classes|Elements and pseudo-elements). Ako pišete selektor dublji od toga, razmislite zašto je to potrebno.

- Ako vaše promjene trebaju obuhvatiti samo **order.html**, svakako se pobrinite da nehotice ne utječete na stranicu **index.html**. S druge strane, ako se neka vizualna komponenta koristi na više stranica, očekuje se da se stilovi primjene kroz web sjedište (kao npr. hover efekt na poveznice).
- Generalno nije dozvoljeno korištenje 'skrivenih' HTML elemenata (npr. nije ispravno dodati skriveni element da bi slike pod kategorijom 'Tools Inventory' poravnali sa slikama pod kategorijom 'Flowers Inventory')
- Podrazumijeva se da poveznice u headeru rade ako postoji odgovarajuća stranica na koju poveznica vodi.

2.2 Primjer zadatka kojeg dobivate na licu mjesta

Ovaj zadatak nije potrebno implementirati u sklopu pripreme! Služi tome da dobijete dojam kompleksnosti zadataka koje dobivate na licu mjesta, a trebate ga moći implementirati za vrijeme trajanja laboratorija. Bit će vam pružena slika stranice onako kako bi trebala izgledati po završetku laboratorija te sve odgovarajuće slike ako ih budete trebali koristiti.

Vaš je zadatak implementirati *floating social sidebar*, odnosno traku s ikonama — poveznicama na stranice društvene mreže koja stoji na desnom rubu stranice.

Obratite pozornost:

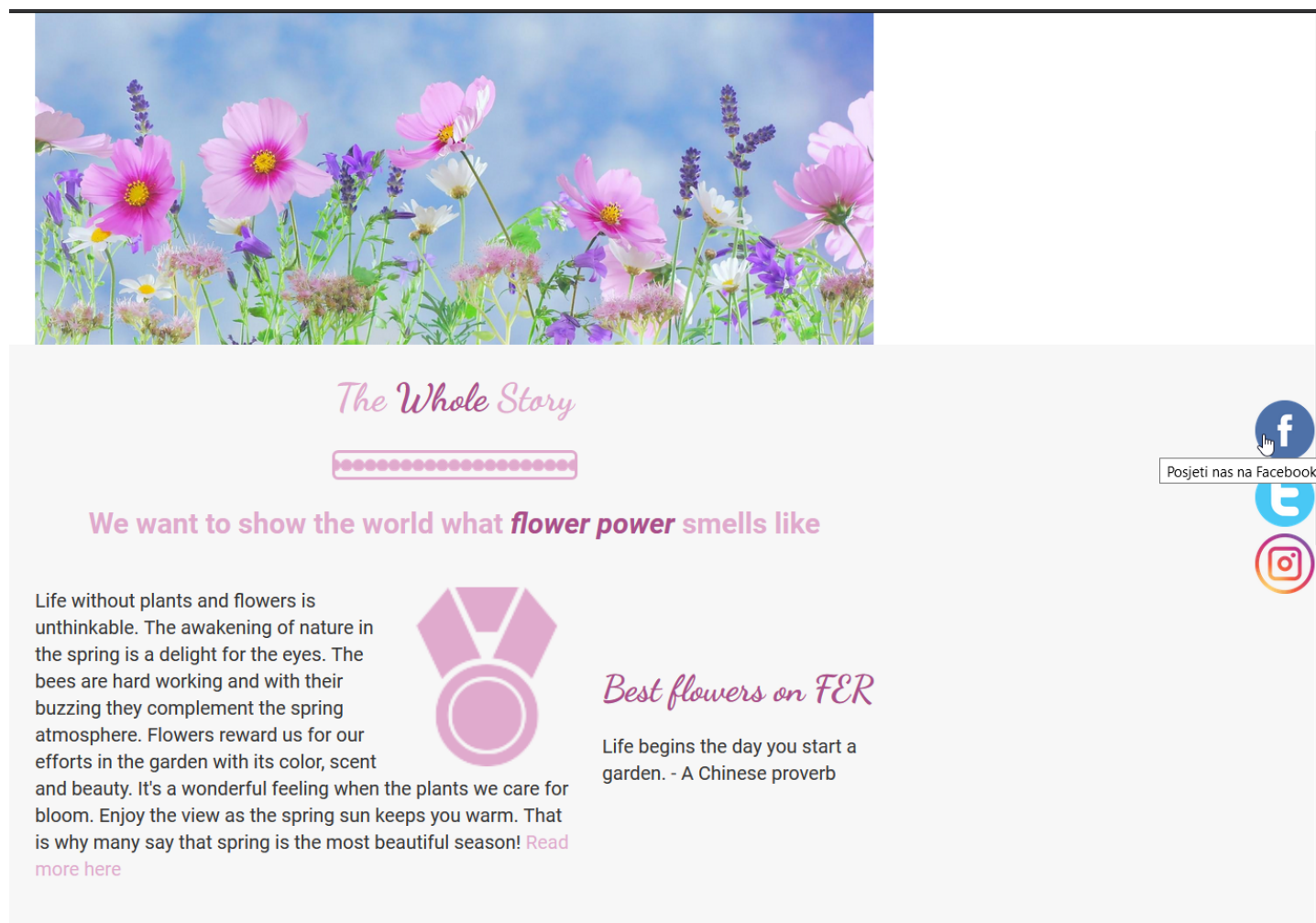
- Sidebar je fiksna na desnom rubu stranice neovisno o vidljivom dijelu stranice, odnosno *scrollanju*.
- Slike su ujedno i poveznice i moraju biti iste veličine i jedna ispod druge.
- Pozicioniranjem miša na sliku/poveznicu potrebno je prikazati poruku s opisom.

3 Kako pisati CSS

Imajući na umu da sva CSS pravila djeluju globalno, ovo poglavlje navodi uobičajene probleme koji nastaju kao posljedica neopreznog pisanja CSS kôda te nudi neka od mogućih rješenja.

3.1 Primjer 1 — jednostavan CSS scope

Pretpostavimo da postoji stranica definirana kôdom na slici 4, a izgleda kako je prikazano na slici 5. Na stranici prikazujemo novosti te sliku. I novosti i slika imaju naslov. Potrebno je naslov novosti pozicionirati centralno.



Slika 3: Zadatak na laboratoriju — ishod

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta http-equiv="content-type" content="text/html; charset=utf-8" />
5   <style>
6     .image {
7       display: block;
8       max-width: 200px;
9       margin: auto;
10    }
11
12    .title {
13      font-size: 25px;
14    }
15  </style>
16 </head>
17 <body>
18   <div>
19     <h1 class="title">Butterflies</h1>
20     <p>Are incoming</p>
21     <hr>
22     <div>
23       <h1 class="title">A butterfly</h1>
24       
25     </div>
26   </div>
27 </body>
28 </html>

```

Slika 4: Primjer 1 — kôd

Butterflies

Are incoming

A butterfly



Slika 5: Primjer 1 — stranica

Jedan od pokušaja rješenja prikazan je kôdom na slici 6. Rješenje je loše zato što osim novosti i slika ima definiran element s klasom **title**, pa se centrira i naslov slike, što se nije trebalo dogoditi. Bolje rješenje prikazano je kôdom na slici 7. Razlika je u tome što smo novosti okružili klasom **news**, a sliku klasom **card**. Sada možemo definirati selektor koji centrira samo onaj **.title** koji se nalazi unutar HTML elementa s klasom **news**. Možemo reći da smo dodavanjem klasa **news** i **card** stvorili *container* za njihove unutarnje *elemente* s klasama **title**.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta http-equiv="content-type" content="text/html; charset=utf-8" />
5   <style>
6     .image {
7       display: block;
8       max-width: 200px;
9       margin: auto;
10    }
11
12    .title {
13      font-size: 25px;
14      text-align: center;
15    }
16  </style>
17 </head>
18 <body>
19   <div>
20     <h1 class="title">Butterflies</h1>
21     <p>Are incoming</p>
22     <hr>
23     <div>
24       <h1 class="title">A butterfly</h1>
25       
26     </div>
27   </div>
28 </body>
29 </html>

```

Slika 6: Primjer 1 — loše rješenje

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta http-equiv="content-type" content="text/html; charset=utf-8" />
5   <style>
6     .image {
7       display: block;
8       max-width: 200px;
9       margin: auto;
10    }
11
12    .title {
13      font-size: 25px;
14    }
15
16    .news .title {
17      text-align: center;
18    }
19  </style>
20 </head>
21 <body>
22   <div class="news">
23     <h1 class="title">Butterflies</h1>
24     <p>Are incoming</p>
25   </div>
26   <hr>
27   <div class="card">
28     <h1 class="title">A butterfly</h1>
29     
30   </div>
31 </body>
32 </html>

```

Slika 7: Primjer 1 — bolje rješenje

3.1.1 Zaključak

Dodavanjem klasa **news** i **card** ograničili smo *scope* naših selektora, odnosno klasama smo stvorili *container* oko *elemenata*. Ako trebate ograničiti *scope* selektora tako da on djeluje samo unutar *containera*, koristite sljedeću sintaksu:

```
.container .element { ... }
```

Imajte na umu: kada bi bilo potrebno centrirati sve naslove na stranici, rješenje 6 je sasvim prikladno.

3.2 Primjer 2 — složen CSS scope

Slično prethodnom primjeru, pretpostavimo da postoji stranica definirana kôdom na slici 8, a izgleda kako je prikazano na slici 9. Za razliku od prethodnog primjera, u HTML kôdu slika s naslovom se nalazi **unutar** novosti. Potrebno je naslov novosti centrirati lijevo. Prividno se čini da je dovoljno liniju 17 na slici 8 prepraviti u **text-align: left**, kako je prikazano kôdom na slici 10. Ali ovo nije moguće napraviti jer unutar klase **news** postoje dva elementa s klasom **title**, pa bi onda oba elementa pomakli lijevo.

Jedno od mogućih rješenja prikazano je na slici 11. Dodali smo prefiks **news-** na klasu **title** unutar **news containera**. Rješenje radi, no postoji i bolji način. Selektor definiran na slici 11 na liniji 17 je nepotrebno specifičan (020). Dovoljno je napisati selektor kako je prikazano na slici 12 na liniji 17. Zbog prefiksa selektora intuitivno razumijemo da se radi samo o naslovima unutar *containera* s klasom **news**, a selektor je manje specifičan (010) pa je lakše uvesti promjene ako to bude bilo potrebno. Napomena: Iako se elementi unutar HTML dokumenta nalaze u nekoj vrsti hijerarhije, nije potrebno da i CSS prati istu hijerarhiju.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta http-equiv="content-type" content="text/html; charset=utf-8" />
5   <style>
6     .image {
7       display: block;
8       max-width: 200px;
9       margin: auto;
10    }
11
12    .title {
13      font-size: 25px;
14    }
15
16    .news .title {
17      text-align: center;
18    }
19  </style>
20 </head>
21 <body>
22   <div class="news">
23     <h1 class="title">Butterflies</h1>
24     <p>Are incoming</p>
25     <hr>
26     <div class="card">
27       <h1 class="title">A butterfly</h1>
28       
29     </div>
30   </div>
31 </body>
32 </html>

```

Slika 8: Primjer 2 — kôd

Butterflies

Are incoming

A butterfly



Slika 9: Primjer 2 — stranica

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta http-equiv="content-type" content="text/html; charset=utf-8" />
5   <style>
6     .image {
7       display: block;
8       max-width: 200px;
9       margin: auto;
10    }
11
12    .title {
13      font-size: 25px;
14    }
15
16    /*Bad*/
17    .news .title {
18      text-align: left;
19    }
20  </style>
21 </head>
22 <body>
23   <div class="news">
24     <h1 class="title">Butterflies</h1>
25     <p>Are incoming</p>
26     <hr>
27     <div class="card">
28       <h1 class="title">A butterfly</h1>
29       
30     </div>
31   </div>
32 </body>
33 </html>

```

Slika 10: Primjer 2 — loše rješenje

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta http-equiv="content-type" content="text/html; charset=utf-8" />
5   <style>
6     .image {
7       display: block;
8       max-width: 200px;
9       margin: auto;
10    }
11
12    .title {
13      font-size: 25px;
14      text-align: center;
15    }
16
17    .news .news-title {
18      text-align: left;
19    }
20  </style>
21 </head>
22 <body>
23   <div class="news">
24     <h1 class="news-title">Butterflies</h1>
25     <p>Are incoming</p>
26     <hr>
27     <div class="card">
28       <h1 class="title">A butterfly</h1>
29       
30     </div>
31   </div>
32 </body>
33 </html>

```

Slika 11: Primjer 2 — dobro rješenje

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta http-equiv="content-type" content="text/html; charset=utf-8" />
5   <style>
6     .image {
7       display: block;
8       max-width: 200px;
9       margin: auto;
10    }
11
12    .title {
13      font-size: 25px;
14      text-align: center;
15    }
16
17    .news-title {
18      text-align: left;
19    }
20  </style>
21 </head>
22 <body>
23   <div class="news">
24     <h1 class="news-title">Butterflies</h1>
25     <p>Are incoming</p>
26     <hr>
27     <div class="card">
28       <h1 class="title">A butterfly</h1>
29       
30     </div>
31   </div>
32 </body>
33 </html>

```

Slika 12: Primjer 2 — bolje rješenje

3.2.1 Zaključak

Dodavanjem prefiksa **news-** nad klase *elemenata containera* s klasom **news** definirali smo jasan scope selektora **unutar** containera. Ako smatrate da je potrebno ostvariti hijerarhijski odnos između *containera* i njegovih *elemenata*, koristite sljedeću sintaksu:

```

.container { ... }
.container—element { ... }

```

Imajte na umu: kada bi bilo potrebno centrirati sve naslove unutar containera **news**, ili ste sigurni da će *container news* uvijek imati samo jedan *element* s klasom **title**, rješenje 10 je sasvim prikladno.

3.3 Primjer 3 — modifikacija postojećeg kôda

U ovom primjeru na stranici imamo dva **news containera**, kako je prikazano kôdom na slici 13, a prikazano u pregledniku slikom 14. Želimo jasnije istaknuti **news-title element** drugih novosti tako da za naslov koristimo tamnije boje. Slično kao u prethodnim primjerima, loše rješenje prikazano kôdom na slici 15 će promijeniti boju teksta svim elementima s klasom **news-title**, za koje može vrijediti da se nalaze i u drugim HTML elementima — takve je greške jako teško otkriti. Dobro rješenje je prikazano na slici 16. Umjesto da mijenjamo sadržaj klase **news-title**, *modificirat* ćemo element tako da na njega dodajemo klasu **news-title-emphasized**.

Selektor **news-title-emphasized** je specifičnosti 010, a iz prefiksa **news-title-** znamo da se radi o *modifikaciji elementa title* koji pripada *containeru news*.


```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta http-equiv="content-type" content="text/html; charset=utf-8" />
5   <style>
6     .news {
7       color: #e0aacd;
8     }
9     .news-title {
10      text-align: left;
11    }
12  </style>
13 </head>
14 <body>
15   <div class="news">
16     <h1 class="news-title">Butterflies</h1>
17     <p>Are incoming</p>
18   </div>
19   <hr>
20   <div class="news">
21     <h1 class="news-title">Butterflies</h1>
22     <p>Are here!</p>
23   </div>
24 </body>
25 </html>

```

Slika 13: Primjer 3 — kôd

Butterflies

Are incoming

Butterflies

Are here!

Slika 14: Primjer 3 — stranica

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta http-equiv="content-type" content="text/html; charset=utf-8" />
5   <style>
6     .news {
7       color: #e0aacd;
8     }
9     .news-title {
10      text-align: left;
11      color: #ab4f8b;
12    }
13  </style>
14 </head>
15 <body>
16   <div class="news">
17     <h1 class="news-title">Butterflies</h1>
18     <p>Are incoming</p>
19   </div>
20   <hr>
21   <div class="news">
22     <h1 class="news-title">Butterflies</h1>
23     <p>Are here!</p>
24   </div>
25 </body>
26 </html>

```

Slika 15: Primjer 3 — loše rješenje

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta http-equiv="content-type" content="text/html; charset=utf-8" />
5   <style>
6     .news {
7       color: #e0aacd;
8     }
9     .news-title {
10      text-align: left;
11    }
12     .news-title-emphasized {
13      color: #ab4f8b;
14    }
15  </style>
16 </head>
17 <body>
18   <div class="news">
19     <h1 class="news-title">Butterflies</h1>
20     <p>Are incoming</p>
21   </div>
22   <hr>
23   <div class="news">
24     <h1 class="news-title-emphasized">Butterflies</h1>
25     <p>Are here!</p>
26   </div>
27 </body>
28 </html>

```

Slika 16: Primjer 3 — dobro rješenje

3.3.1 Zaključak

Kada je potrebno promijeniti samo neke elemente od postojećih na stranici, ne treba olako brisati ili modificirati postojeće klase. Bolje je dodati novu klasu nad *elemente* koje mijenjamo. Prefikse koristimo iz istog razloga kao u prethodnom primjeru. Ovim pristupom je kôd pregledniji, modularniji i lakše se održava. Ako mijenjate samo neke elemente od postojećih na stranici, koristite sljedeću sintaksu:

```

.container {...}
.container-modifier {...}
.container-element {...}
.container-element-modifier {...}

```

Imajte na umu: ako znate točno što će se dogoditi modifikacijom postojećih klasa, modificirajte ih. Konstantno dodavanje novih klasa bez refaktoriranja kôda učinit će vaš kôd teškim za pregled i održavanje.

3.4 Koji pristup je ispravan?

Teško je reći što trebate koristiti. Svaki pristup ima svoje prednosti i mane, a primjena ovisi o problemu koji rješavate. Ako niste sigurni, vodite se konzervativnim pristupom, navedenom u primjeru 3.3.

Za one koji žele znati više, neke od formalnih metodologija pisanja CSS-a su:

- SMACSS
- BEM