Stilizovanje u CSS-u i modeli okvira

2.1 CSS i HTML

Nova specifikacija za HTML ne podrazumeva samo oznake i sam jezik HTML. Veb zahteva odgovarajući dizajn i funkcionalnost, a ne samo organizaciju strukture i definisanje odeljaka. U takvom novom modelu, HTML se integriše sa CSS-om i JavaScriptom u jedinstvenu novu tehnologiju. Do sada smo govorili o mogućnostima svakog pojedinačnog jezika i uveli nove elemente strukture HTML dokumenta; sada ćemo proučavati važnost CSS-a u ovom strateškom savezu, i njegov uticaj na prikazivanje HTML dokumenata.

Zvanično, CSS nema nikakve veze s jezikom HTML5. CSS nije deo specifikacije, niti je to ikada bio: on je dopuna koja treba da prevaziđe ograničenja HTML-a i smanji njegovu složenost. U početku, atributi unutar HTML oznaka definisali su osnovne stilove svih elemenata, ali kako se jezik razvijao, kôd je postajao sve komplikovaniji za pisanje i održavanje, pa sâm HTML više nije mogao da zadovolji zahteve veb dizajnera. Zbog toga je usvojen CSS, kao način razdvajanja strukture od prezentacije. Od tada se CSS razvija uporedo, usmeren pre svega na dizajnere i njihove potrebe, a ne na to da po svaku cenu bude deo razvoja HTML-a.

Najnovija verzija CSS-a, CSS3, sledi isti put, ali uz mnogo više kompromisa. Specifikacija za HTML5 razvila se tako što se implicitno podrazumevalo da je CSS zadužen za dizajn. Zbog toga je integracija jezika HTML i CSS3 sada ključna za razvoj veba, pa svako pominjanje jezika HTML5 istovremeno upućuje i na CSS3, iako su to zvanično dve nezavisne tehnologije.

Mogućnosti jezika CSS3, zajedno sa ostatkom specifikacije, danas se implementiraju i ugrađuju u čitače veba kompatibilne sa HTML5. U ovom poglavlju naučićete osnovne koncepte CSS-a i nove metode jezika CSS3 za predstavljanje i strukturiranje. Upoznaćete i nove selektore i pseudoklase pomoću kojih se HTML elementi lakše biraju i identifikuju.

VAŽNO Sledi kratak uvod u CSS stilove. Pominjemo samo one metode i svojstva koji su neophodni za razumevanje primera u ovoj knjizi. Ako nemate nikakvog iskustva sa CSS-om, mogu vam pomoći priručnici na vebu. Naći ćete ih na našoj veb lokaciji ako pratite veze za ovo poglavlje.

2.2 Kratak uvod u CSS

CSS je jezik koji zajedno sa HTML-om definiše vizuelne stilove za elemente dokumenta, kao što su veličina, boja, pozadine, ivice itd. Iako čitači veba podrazumevano dodeljuju stilove HTML elementima, ti stilovi ne moraju odgovarati idejama dizajnera ili željama korisnika. Dizajneri i programeri često moraju da primenjuju sopstvene stilove kako bi veb lokacija imala odgovarajući izgled na ekranu.

U ovom delu poglavlja, prikazaćemo stilizovanje pomoću CSS-a i objasniti osnovne metode za definisanje strukture dokumenta.

VAŽNO Neki koncepti koji će biti objašnjeni u narednim pasusima treba da uvedu početnike u CSS i osnove stilizovanja. Ako su vam ove informacije poznate, slobodno preskočite delove koje ste već savladali.

Pravila CSS-a

CSS definiše kako će se HTML elementi prikazivati na ekranu. Za prikazivanje stilova u CSS-u koriste se svojstva (engl. *properties*) i vrednosti (engl. *values*). Ta sintaksa se zove **deklaracija** (engl. *declaration*), a sadrži dvotačku iza imena svojstva, i red zatvoren tačkom i zarezom.

Listing 2-1: *Deklarisanje svojstava CSS-a.*

```
color: #FF0000;
```

U primeru iz listinga 2-1, svojstvu **`color`** dodeljena je vrednost **`#FF0000`**. Ako se ovo svojstvo primeni na HTML element, on će se na ekranu prikazati u crvenoj boji.

Podsetite se osnova Boje u CSS-u definišu se pomoću kombinacije tri osnovne boje: crvene, zelene i plave. Možemo ih predstaviti pomoću heksadecimalnih brojeva (od 00 do FF) ili decimalnih brojeva (od 0 do 255). Ako boju izražavamo heksadecimalnim brojem, na početak stavljamo znak za povisilicu (tarabu), kao u primeru iz listinga 2-1. Za decimalne brojeve dostupna je funkcija **`rgb`**, a boje se deklarišu pomoću sintakse **`rgb(255, 0, 0)`**. O funkciji **`rgb()`** i sličnim funkcijama govorićemo u nastavku knjige.

Svojstva se mogu grupisati pomoću vitičastih zagrada. Takva grupa od jednog ili više svojstava naziva se **pravilo** (engl. *rule*) i identifikuje se pomoću imena, tj. *selektora* (engl. *selector*), predstavljajući element ili grupu elemenata na koje će pravilo uticati. Broj pravila koje možemo napisati je neograničen.

Listing 2-2: *Deklarisanje pravila CSS-a.*

```
p {
  color: #FF0000;
  font-size: 24px;
}
```

U listingu 2-2 vidite dva svojstva (**`color`** i **`font-size`**) navedena u zagradama, sa odgovarajućim vrednostima. Ovo pravilo se identifikuje pomoću imena **`p`**. U ovom slučaju, ime pravila referencira elemente **`<p>`**. Ako ga primenimo na naš dokument, sadržaj svakog elementa **`<p>`** biće crvene boje i veličine 24 piksela.

U CSS pravilu možemo deklarisati neograničen broj selektora. Treba samo da navedemo njihova imena i razdvojimo ih zarezima.

Listing 2-3: *Deklarisanje pravila CSS-a.*

```
p, span {
  color: #FF0000;
  font-size: 24px;
}
```

U primeru iz listinga 2-3, pravilo utiče na sve elemente **`<p>`** i **`<span>`** u dokumentu.

Možemo referencirati i samo one elemente koji se nalaze unutar određenog elementa, nabrajajući ih i razdvajajući razmacima. Na primer, elementi **`<span>`** koji predstavljaju decu elementa **`<p>`**:

Listing 2-4: *Deklarisanje pravila CSS-a.*

```
p span {
  color: #FF0000;
  font-size: 24px;
}
```

Postoje različite metode za referenciranje HTML elemenata iz CSS-a. Ona koju smo primenili u prethodnom primeru koristi samo ime tipa elementa na koji se pravilo primenjuje. To je opšta referenca, ali postoje i specifične reference o kojima ćemo uskoro govoriti.

Svojstva

Svojstva čine jezgro CSS-a. Ima ih mnogo, a u svaku verziju jezika dodavana su nova. Neka od njih primenićemo kroz praktične primere u nastavku knjige. Sledi spisak najčešćih svojstava (uključujući i ona koja se koriste od prve verzije CSS-a):

`font` – Omogućava da deklarišemo stilove za tekst, kao što su intenzitet, veličina, porodica fontova itd. Vrednosti moraju biti razdvojene razmacima i deklarisane određenim redosledom (npr., **`font: bold 24px arial,sans-serif`**). Svaki stil možemo deklarisati i nezavisno, pomoću pojedinačnih svojstava **`font-style`**, **`font-variant`**, **`font-weight`**, **`font-size`**, **`line-height`** i **`font-family`** (npr., **`font-size: 24px`**).

`color` – Deklariše boju elementa. Vrednost se može izraziti heksadecimalnim brojevima (npr., **`color: #FF0000;`**) ili decimalnim brojevima (npr., **`color: rgb(255, 0, 0);`**).

`background` – Omogućava da na pozadinu elementa primenimo stilove, kao što su boja, slika, ponavljanje itd. Zadate vrednosti se moraju razdvojiti razmacima (npr., **`background: #0000FF url('bricks.jpg') no-repeat;`**). Svaki stil se može deklarisati i nezavisno, pomoću pojedinačnih svojstava. Ova i sva pridružena svojstva detaljnije ćemo proučavati u poglavlju 3.

`width` – Deklariše širinu elementa (npr., **`width: 200px`**).

`height` – Deklariše visinu elementa (npr., **`height: 200px`**).

`margin` – Deklariše spoljnu marginu elementa koja predstavlja prostor oko elementa. Ona može imati četiri vrednosti: gore, desno, dole i levo, koje se navode tim redosledom i razdvojene razmacima (npr., **`margin: 10px 30px 10px 30px;`**). Međutim, ako se deklariše samo jedan, dva ili tri parametra, ostalima se dodeljuju iste vrednosti (npr., **`margin: 10px 30px`**). I ove vrednosti se mogu deklarisati nezavisno, pomoću pojedinačnih svojstava **`margin-top`**, **`margin-right`**, **`margin-bottom`** i **`margin-left`** (npr., **`margin-left: 10px;`**).

`padding` – Deklariše unutrašnju marginu elementa, koja predstavlja prostor oko sadržaja unutar ivica elementa – na primer, prostor između naslova i ivice virtuelnog okvira, formiranog pomoću elementa **`<h1>`** koji sadrži taj naslov. Vrednosti se deklarišu na isti način kao za svojstvo **`margin`**. I ove vrednosti se mogu deklarisati nezavisno, upotrebom svojstava **`padding-top`**, **`padding-right`**, **`padding-bottom`** i **`padding-left`** (npr., **`padding-top: 10px;`**).

`border` – Omogućava da deklarišemo širinu, stil i boju ivice elementa (npr., **`border: 1px solid #990000;`**). Moguće vrednosti za stil su **`none`**, **`hidden`**, **`dotted`**, **`dashed`**, **`solid`**, **`double`**, **`groove`**, **`ridge`**, **`inset`**, **`outset`** i **`inherit`**. I ovo svojstvo ima pojedinačna svojstva za svaku stranu elementa: **`border-top`**, **`border-bottom`**, **`border-left`** i **`border-right`**. Svakoj strani nezavisno možemo dodeliti i vrednosti za širinu, stil i boju, pri čemu se ime dodaje na kraju svojstva. Ivicu uvek

moramo deklarisati pre nego što je na bilo koji način izmenimo (npr., **`border-style: solid; border-top-width: 10px;`**).

`text-align` – Poravnava element unutar njegovog roditelja. Moguće vrednosti su **`left`**, **`right`**, **`center`**, **`justify`** i **`inherit`**.

VAŽNO Ovo nisu jedina dostupna svojstva CSS-a. U ovoj knjizi primenićemo samo nekoliko postojećih svojstava i proučavati nova koja su ugrađena u CSS3. Da biste naučili više, posetite našu veb lokaciju i pratite veze za ovo poglavlje.

Umetnuti stilovi

Primena stilova na HTML elemente menja način njihovog prikazivanja na ekranu. Pomenuli smo da čitači veba podrazumevano definišu skup stilova koji, u većini slučajeva, ne zadovoljavaju potrebe dizajnera. Pomoću različitih metoda, te stilove možemo zanemariti i zameniti sopstvenim stilovima. Jedna od tih metoda dodeljuje stilove unutar elementa u obliku atributa. Listing 2-5 prikazuje jednostavan HTML kôd u kome je na element **`<p>`** primenjen atribut **`style`** sa dodeljenom vrednošću **`font-size: 20px`**. Ovo svojstvo redefiniše podrazumevanu veličinu teksta unutar elementa **`<p>`** tako da je nova veličina 20 piksela.

Listing 2-5: *CSS stilovi unutar HTML oznaka.*

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>This text is the title of the document</title>
</head>
<body>
  <p style="font-size: 20px">My text</p>
</body>
</html>
```

Uradite sami U praznu tekstualnu datoteku kopirajte HTML kôd i otvorite datoteku u svom čitaču veba da biste proverili kako radi. Ime datoteke mora imati nastavak **`.html`** da bi se pravilno otvorila (npr., **`myfile.html`**).

Navedena metoda korišćenja *umetnutih* ili *rednih stilova* (engl. *inline styles*) dobar je način za testiranje stilova i proveru rezultata, ali se ne preporučuje u većim projektima. Razlog je jednostavan: ona zahteva da se svaki stil upisuje i ponavlja za svaki element, što znatno povećava dokument i može onemogućiti njegovo ažuriranje i održavanje. Na primer, ako biste odlučili da veličina teksta u svakom elementu **`<p>`** umesto 20 piksela bude 24 piksela, morao bi da se promeni svaki stil u svakoj oznaci **`<p>`** u celom dokumentu, kao i u svakom dokumentu vaše veb lokacije.

Ugrađeni stilovi

Bolje rešenje od prethodnog jeste umetanje stilova u glavu dokumenta, nakon čega se na odgovarajuće HTML elemente može uticati pomoću referenci.

Listing 2-6: *Stilovi u glavi dokumenta.*

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>This text is the title of the document</title>
  <style>
    p { font-size: 20px }
  </style>
</head>
<body>
  <p>My text</p>
</body>
</html>
```

Element **<style>** omogućava programerima da CSS stilove umeću u dokumente. U ranijim verzijama HTML-a morala se definisati vrsta stilova koja se umeće; u jeziku HTML5, podrazumevani stil je CSS, pa stoga početnoj oznaci **<style>** ne moraju da se dodaju atributi.

Podebljan kôd listinga 2-6 ima istu namenu kao i red koda u listingu 2-5, ali u ovom primeru stil ne mora da se upisuje unutar svake oznake **<p>** u dokumentu, jer su svi elementi **<p>** već stilizovani. Ovom metodom skraćuje se kôd i dodeljuje odgovarajući stil odgovarajućim elementima pomoću referenci.

Spoljne datoteke

Deklarisanjem stilova u glavi dokumenta štedi se prostor, pa je kôd konzistentniji i lakše se održava, ali zahteva kopiranje stilova u svaki dokument veb lokacije. Problem ponavljanja može se rešiti tako što se svi stilovi premeste u spoljnu datoteku, a ta datoteka zatim, pomoću elementa **<link>**, umetne u svaki dokument koji treba stilizovati. Ova metoda omogućava i to da se ceo skup stilova izmeni upotrebom druge datoteke. Tako se naši dokumenti mogu izmeniti ili prilagoditi svakoj okolnosti i svakom uređaju, o čemu ćemo govoriti pri kraju ove knjige.

U poglavlju 1 razmotrili smo oznaku **<link>** i naučili da umećemo CSS datoteke u dokumente. Pomoću reda **<link rel="stylesheet" href="mystyles.css">** naložili smo čitaču veba da učita datoteku **mystyles.css**, koja sadrži sve stilove potrebne za prikazivanje stranice na ekranu. To je praksa velikog broja dizajnera koji već koriste HTML5. Oznaka **<link>** koja referencira CSS datoteku biće umetnuta u svaki dokument kome trebaju dati stilovi.

Listing 2-7: *Primena CSS stilova iz spoljne datoteke.*

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>This text is the title of the document</title>
  <link rel="stylesheet" href="mystyles.css">
</head>
<body>
  <p>My text</p>
</body>
</html>
```

> **Uradite sami** CSS stilove dodavaćemo datoteci **`mystyles.css`**. Ovu datoteku morate napraviti u istom direktorijumu u kome je i HTML datoteka, i u nju kopirati CSS stilove da biste videli kako rade.
>
> **Podsetite se osnova** CSS datoteke su obične tekstualne datoteke. I njih, kao i HTML datoteke, možete praviti pomoću bilo kojeg editora teksta, kao što je Windowsov Notepad.

Reference

Sve stilove koje koristite možete grupisati u jednu spoljnu datoteku, koju ćete povezati sa svojim dokumentima. Ovo rešenje je praktično, ali nam trebaju mehanizmi za uspostavljanje posebnih odnosa između tih stilova i elemenata unutar dokumenata na koje će se oni primenjivati.

U prethodnim primerima primenili smo jednu moguću metodu za dodeljivanje stilova elementu. Na primer, stil koji menja veličinu fonta u listingu 2-6 referencirao je svaki element **`<p>`** pomoću rezervisane reči **`p`**. Kao rezultat, stil umetnut između oznaka **`<style>`** dodelio je taj CSS stil svakom elementu **`<p>`** u dokumentu.

Postoje tri osnovne metode za biranje HTML elementa na koji će se primeniti CSS pravilo:

- pomoću rezervisane reči elementa
- pomoću atributa **`id`**
- pomoću atributa **`class`**

Međutim, kasnije ćemo videti da je CSS3 prilagodljiv, i da obuhvata nove i specifičnije načine primene CSS pravila.

Referenciranje pomoću rezervisane reči

Deklarisanje CSS pravila pomoću rezervisane reči elementa uticaće na svaki element tog tipa u dokumentu. Na primer, naredno pravilo menja stilove elemenata **`<p>`**:

Listing 2-8: *Referenciranje pomoću rezervisane reči.*

```
p { font-size: 20px }
```

Kada se rezervisana reč **p** nalazi ispred pravila, nalažemo čitaču veba da se ono mora primeniti na svaki element **<p>** u HTML dokumentu. Kao rezultat, ceo tekst unutar oznaka **<p>** biće veličine 20 piksela.

Ovo može da se primeni i na sve druge HTML elemente u dokumentu. Na primer, ako smo umesto **p** definisali rezervisanu reč **span**, svaki tekst između oznaka **** biće 20 piksela.

Listing 2-9: *Referenciranje pomoću druge rezervisane reči.*

```
span { font-size: 20px }
```

Ali šta ako hoćemo da referenciramo samo jednu oznaku? Treba li da koristimo atribut **style** unutar te oznake? Odgovor je odričan. Pomenuli smo da se metoda umetnutih stilova (koja koristi atribut **style** unutar HTML oznaka) smatra prevaziđenom i da je treba koristiti samo za testiranje. Da biste izabrali određeni HTML element iz pravila u CSS datoteci, možete upotrebiti dva različita atributa: **id** i **class**.

Referenciranje pomoću atributa id

Atribut **id** više liči na ime i predstavlja identifikaciju elementa; to znači da vrednost ovog atributa mora biti jedinstvena u celom dokumentu.

Da biste određeni element iz naše CSS datoteke referencirali pomoću atributa **id**, pravilo morate deklarisati tako da ispred vrednosti za identifikaciju stavite znak tarabe (**#**).

Listing 2-10: *Referenciranje elementa pomoću vrednosti atributa* `id`.

```
#text1 { font-size: 20px }
```

Pravilo iz listinga 2-10 primeniće se na HTML element identifikovan pomoću atributa **id="text1"**. HTML kôd sada izgleda ovako:

Listing 2-11: *Identifikovanje elementa* <p> *pomoću njegovog atributa* `id`.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>This text is the title of the document</title>
  <link rel="stylesheet" href="mystyles.css">
```

```
</head>
<body>
  <p id="text1">My text</p>
</body>
</html>
```

Kad god u CSS datoteci referenciramo neki element pomoću identifikatora **text1**, element s tom id vrednošću se menja, ali ostali elementi **<p>**, kao i drugi elementi u dokumentu, ostaju nepromenjeni.

Ovaj način referenciranja je krajnje specifičan i koristi se uglavnom za opštije elemente, kao što su strukturne oznake. Atribut **id** i njegova specifičnost više odgovaraju referenciranju elemenata iz JavaScripta, o čemu će biti reči u narednim poglavljima.

Referenciranje pomoću atributa class

Često je bolje da, umesto atributa **id**, za stilizovanje koristite atribut **class**. On je prilagodljiviji i može se dodeliti svim HTML elementima u dokumentu koji imaju zajedničke stilove.

Listing 2-12: *Referenciranje pomoću vrednosti atributa* `class`.

```
.text1 { font-size: 20px }
```

Da bismo koristili atribut **class,** moramo da deklarišemo pravilo tako da stavimo tačku ispred njegovog imena. Prednost ove metode je to što je umetanje atributa **class**, koji ima vrednost **text1**, dovoljno da se to isto pravilo dodeli bilo kom izabranom elementu.

Listing 2-13: *Dodeljivanje stilova pomoću atributa* `class`.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>This text is the title of the document</title>
  <link rel="stylesheet" href="mystyles.css">
</head>
<body>
  <p class="text1">My text</p>
  <p class="text1">My text</p>
  <p>My text</p>
</body>
</html>
```

Elementi **<p>** u prva dva reda listinga 2-13 imaju atribut **class** s vrednošću **text1**. Pomenuli smo da se isto pravilo može primeniti na različite elemente u istom dokumentu. Stoga prva dva elementa imaju iste stilove i na oba se primenjuje pravilo iz listinga 2-12. To ne utiče na poslednji element **<p>**, koji se prikazuje s podrazumevanim stilovima.

Pravila dodeljena pomoću atributa `class` nazivaju se **klasama**. Istom elementu može se dodeliti više klasa, samo treba deklarisati imena klasa razdvojena razmacima (npr., **class="text1 text2"**). Klasa se može deklarisati isključivo za određene tipove elemenata, i to tako što se ispred tačke doda selektor.

Listing 2-14: *Deklarisanje klase samo za elemente* <p>.

```
p.text1 { font-size: 20px }
```

Listing 2-14 sadrži pravilo koje referencira klasu pod imenom **text1**, ali samo za elemente **<p>**. Ako bilo koji drugi element bude imao isto ime za svoj atribut **class**, na njega se ovo pravilo neće primeniti.

Referenciranje pomoću bilo kog atributa

Opisane metode za referenciranje mogu se primeniti na razne slučajeve, ali ponekad nisu dovoljne za pronalaženje elemenata koji želimo da stilizujemo. Najnovije verzije CSS-a sadrže nove načine za referenciranje HTML elemenata. Jedan od njih je *selektor atributa* (engl. *attribute selector*). Sada element može da se referencira ne samo pomoću atributa **id** i **class**, nego i pomoću bilo kog drugog atributa.

Listing 2-15: *Referenciraju se samo elementi* <p> *koji imaju atribut* `name`.

```
p[name] { font-size: 20px }
```

Pravilo iz listinga 2-15 menja samo one elemente **<p>** koji imaju atribut pod imenom **name**. Da bismo pokazali šta radimo sa atributima **id** i **class**, možemo navesti i vrednost atributa.

Listing 2-16: *Referenciraju se elementi* <p> *koji imaju atribut* `name` *s vrednošću* `mytext`.

```
p[name="mytext"] { font-size: 20px }
```

CSS3 omogućava da kombinujemo simbol „=“ sa ostalima, kako bismo napravili precizniju selekciju:

Listing 2-17: *Novi selektori u jeziku CSS3.*

```
p[name^="my"] { font-size: 20px }
p[name$="my"] { font-size: 20px }
p[name*="my"] { font-size: 20px }
```

Ako su vam već poznati regularni izrazi (engl. *regular expressions*) iz drugih jezika, poput JavaScripta ili PHP-a, prepoznaćete selektore iz listinga 2-17 – oni u jeziku CSS3 daju slične rezultate:

- Pravilo koje sadrži selektor **^**= dodeliće se svakom elementu **<p>** čija vrednost atributa **name** započinje podnizom **"my"** (npr., **mytext**, **mycar**).
- Pravilo koje sadrži selektor **$**= dodeliće se svakom elementu **<p>** čija se vrednost atributa **name** završava podnizom **"my"** (npr., **textmy**, **carmy**).
- Pravilo koje sadrži selektor *= odgovaraće svakom elementu **<p>** čija vrednost atributa **name** sadrži podniz **"my"** (u ovom slučaju, podniz se može nalaziti i u sredini – na primer, **textmycar**).

U navedenim primerima koristili smo element **<p>**, atribut **name** i slučajno izabran tekst kao što je **"my"**, ali se ista metoda može upotrebiti sa svakim atributom i vrednošću. Da biste referencirali bilo koji HTML element, upišite ime atributa i vrednost koju tražite unutar uglastih zagrada.

Referenciranje pomoću pseudoklasa

Pseudoklase omogućavaju da referenciramo HTML elemente pomoću njihovih osobina, kao što su položaj u kodu ili aktuelna stanja. CSS3 sadrži nove pseudoklase koje još preciznije referenciraju elemente. Primenićemo nekoliko njih u jednostavnom dokumentu, a ostale obraditi kasnije u drugim praktičnim primerima.

Listing 2-18: *HTML dokument za testiranje pseudklasa.*

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>This text is the title of the document</title>
  <link rel="stylesheet" href="mystyles.css">
</head>
<body>
  <div id="wrapper">
    <p class="mytext1">My text1</p>
    <p class="mytext2">My text2</p>
    <p class="mytext3">My text3</p>
    <p class="mytext4">My text4</p>
  </div>
</body>
</html>
```