

CSS

CSS je jezik formatiranja pomoću kog se definiše izgled elemenata veb-stranice.

Prvobitno, [HTML](#) je služio da definiše kompletan izgled, strukturu i sadržaj veb-stranice, ali je od verzije 4.0 HTML-a uveden CSS koji bi definisao konkretan izgled, dok je HTML ostao u funkciji definisanja strukture i sadržaja.

Jezik

CSS sintaksa se sastoji od *opisa* izgleda elemenata u dokumentu. Opis može da definiše izgled više elemenata, i više opisa može da definiše jedan element. Na taj način se opisi *slažu* jedan preko drugog da bi definisali konačni izgled određenog elementa (otuda naziv *Cascading* da bi se dočaralo slaganje jednog stila preko drugog u definisanju konačnog izgleda elementa)

Svaki opis se sastoji od tri elementa:

- definicija ciljnih elemenata
- svojstva
- vrednosti

Nakon što definišemo ciljne elemente, tj. elemente na koje će se trenutni opis odnositi, nizom parova svojstvo-vrednost definišemo izgled svakog ciljnog elementa. Sintaksa koja se pri tome definiše je sledećeg oblika:

```
ciljni elementi {      : vrednost;      svojstvo: vrednos  
t;      ... }
```

CSS podržava i komentare, koji izgledaju kao u programskom jeziku [C](#), dakle navodeći se između znakova */** i **/*.

Ciljni elementi

Ciljni elementi se definišu na tri načina:

- navodeći HTML tag ciljnih elemenata
- navodeći *klasu* elemenata
- navodeći direktnu identifikacionu vrednost (**ID**) elementa

Kada definišemo preko HTML taga, to znači da će opis uticati na sve elemente u dokumentu koji imaju ovaj tag. Definicija ciljnih

elemenata se tada vrši preko direktnog upisa odgovarajućeg HTML taga:

```
body {      svojstvo: vrednost;      ... }
```

Klasa HTML elementa je reč koju stavimo kao vrednost argumenta **CLASS** pri definiciji tog elementa. Definicija ciljnih elemenata (svih koji imaju određenu istu klasu) se vrši tako što upišemo znak tačke (.) a zatim naziv klase:

```
.imeKlase {      svojstvo: vrednost;      ... }
```

Ovaj opis će, dakle, imati uticaja na sve elemente u dokumentu koji su definisani na sledeći način:

```
<p class="imeKlase">...</p> <div class="imeKlase">...</div> ...
```

Identifikaciona vrednost elementa je vrednost argumenta **ID** pri definiciji tog elementa. U jednom dokumentu identifikacione vrijednosti moraju biti jedinstvene, tj. može postojati samo jedan element sa određenom identifikacionom vrednošću, što znači da ovakvi opisi mogu uticati samo na po jedan element u dokumentu. Definišu se pomoću znaka tarabe (#), a zatim identifikacionu vrednost:

```
#identifikacionaVrednost {      svojstvo: vrednost;      ... }
```

Ovaj će, dakle, opis uticati na sve elemente koji u svojoj HTML definiciji imaju **id="identifikacionaVrednost"**.

Svojstva i vrednosti

Svojstva na koja želimo da utičemo u datom opisu se definišu preko niza ključnih reči definisanih u [W3C](#) standardu, a koje se kategorišu u sledeće grupe:

- definicija pozadine elementa
- ivica
- okvir
- prikaz
- dimenzije

- font
- generisani sadržaj
- margine
- unutrašnji prazan prostor
- pozicija
- izgled pripadajućeg teksta

Vrednosti pojedinih svojstava se definišu na dva načina:

- predefinisanim ključnim rečima
- brojevnim vrednostima

Predefinisane ključne reči za vrednosti svojstava se koriste u situacijama kada dato svojstvo ima ograničen broj mogućnosti. Tako na primer svojstvo za definiciju pozadine elementa može biti samo **scroll** (da se kreće uvek zajedno sa sadržajem elementa) i **fixed** (da stoji uvek na istom mjestu bez obzira na sadržaj).

Brojeвне vrednosti se mogu zadavati na nekoliko načina:

- zadajući samo brojeвnu vrednost
- navodeći i jedinicu veličine skupa sa opisom (**px**, **em**, **pt**, ...)

Ako zadajemo samo brojeвnu vrednost, imamo mogućnost da je pišemo na dva načina:

- u dekadnom sistemu (709)
- u heksadekadnom sistemu (#a4fde3); ovaj sistem se najviše koristi u definiciji boje

Primeri

- Primer za definiciju pozadine cijele stranice:

```
body {      background-color: aqua; /* pozadina boje aqua */ }
```

- Primer za definiciju izgleda svih elemenata klase "citati":

```
.citati {   font-family: serif; /* font sa kukicama, kao Times New Roman */      font-style: italic; /* iskošen tekst */      font-size: 9pt;      /* veličina slova 9 tačaka */ }
```

Organizacija CSS koda

CSS kod se može zadavati na tri standardna mesta:

- direktno u tagu, koristeći argument **style**
- u zaglavlju dokumenta unutar taga **style**
- u eksternoj datoteci, koja se u dokument uključuje tagom **link**

Za jako male dokumente, argument **style** taga može biti prihvatljiv, ali za veće dokumente ne može jer se na ovaj stil definiše za svaki element ponaosob. Definicija stilova se najčešće radi grupišući elemente u klase, pomoću argumenta **class** taga, a zatim definišući izgled tih klasa bilo u okviru elementa **style** u zaglavlju ili u eksternoj datoteci. Argument **style** taga se najčešće koristi ako neki element ne pripada nijednoj grupi i ima posebne zahteve u odnosu na sve ostale elemente.

Stilovi u eksternoj datoteci imaju jednu posebnu prednost u odnosu na preostala dva načina, a to je da na taj način možemo da utičemo na elemente više dokumenata, praktično na čitav jedan sajt. Dovoljno je da pisac bude dosledan u definisanju klasa elemenata, da bi jedan fajl sa stilovima određivao izgled svih stranica. Na ovaj način jedna promjena u fajlu sa stilovima utiče na sve stranice te su stilovi centralizovani.

Ograničenja

Poznata ograničenja „čistog“ CSS-a uključuju u sledeća:

- Nedoslednost pretraživača:
Različiti pretraživači će interpretirati CSS kôd na različite načine zbog svojih bagova ili zbog nepodržavanja određenih mogućnosti CSS-a. Na primer, Internet Explorer, čija ranija izdanja nisu podržavala mnoge mogućnosti CSS-a 2.1, su pogrešno interpretirala veliki broj bitnih opisa objekata, uključujući širinu, visinu i plutanje. Da bi se postigla doslednost prikaza u brojnim popularnim brauzerima, moraju se koristiti raznorazne „veštine“ i „štimanja“. I pored toga, ponekad je

nemoguće postići identičan prikaz dokumenta u različitim pretraživačima.

- Složenost

Veliki sajtovi imaju problem veličine CSS dokumenata, koji vremenom postaju toliko složeni i dugački da održavanje i dodavanje novih svojstava postaje izuzetno težak posao.

- Opisi elemenata se ne mogu bazirati na drugim objektima

CSS ne podržava „nasleđivanje“ - ne postoji način da se izabere opis nekog drugog elementa na kojem bi se bazirao trenutni element. Nasleđivanje postoji, ali se bazira na unapred utvrđenom pravilu da se opisi nasleđuju od elementa u kojem se dotični element nalazi, tj. od roditeljskog elementa. Radna grupa za CSS objašnjava da ne želi da uvede proizvoljno nasleđivanje jer bi to otežalo i usporilo rad brauzera.

- Problemi pri vertikalnoj orijentaciji elemenata

Iako je prilično lako postići horizontalni tok elemenata, vertikalni tok je prilično teško, i ponekad nemoguće, postići na željeni način. Jednostavni prohtevi, poput centriranja objekta po vertikali, ili prikaz podnožja dokumenta na samom dnu vidljivog dijela prozora, zahtijevaju prilično komplikovane opise, ili opise koje podržavaju samo određeni brauzeri. Ponekad je najlakše uposliti skriptni jezik poput [Javaskripta](#) da na osnovu tekućeg stanja klijenta prikaže element na odgovarajućoj poziciji, čime se opet gubi na kompatibilnosti jer određeni korisnici ne žele da koriste Javaskript.

- Nedostatak aritmetičkih operacija

Dosadašnje verzije CSS-a ne podržavaju ni najjednostavnije aritmetičke operacije (npr. `margin-left: 10%-10px;`). Neki

pomaci su napravljeni, raspravom Radne grupe za CSS o uvođenju izraza `calc()`, a Internet Explorer od verzije 5.0 podržava izraz `expression`, sa sličnim mogućnostima.

- Neslaganje pojedinačnih opisa
Određena svojstva imaju sličnu svrhu, i često se ne slažu u potpunosti. Na primer, svojstva `position`, `float` i `display` određuju lokaciju prikaza elementa i njihovo kombinovanje često prouzrokuje neočekivane rezultate a određene kombinacije su po pravilu zabranjene. Na primer, ako određenom elementu dodijelimo svojstvo `display: table-cell`, onda mu ne možemo dodijeliti svojstvo `float` niti `position: relative`.
- Određeni elementi ne mogu sadržati plutajuće pod-elemente
Ne postoji svojstvo koje takve elemente forsira da podržavaju plutajuće pod-elemente. Ponekad dodeljivanje svojstva `position: relative` rešava ovaj problem, ili definisanje i samog nad-elementa da ima plutajuće svojstvo, ali u situacijama kada to nije prihvatljivo za organizaciju dokumenta, to nije rešenje.
- Nemogućnost postavljanja više pozadinskih tema istom elementu
Složena grafička rešenja zahtevaju više od jedne pozadinske slike po elementu, ali CSS podržava samo jednu. Zbog toga su dizajneri primorani da elemente slažu jedan preko drugog da bi dobili ono što su tražili, ili da odustanu od prvobitne ideje. Ovaj problem se planira riješiti u trećoj verziji CSS-a, a određena rešenja već postoje u pretraživačima [Safari](#) i [Konkveror](#).
- Kontrola oblika elemenata

CSS trenutno dopušta samo pravougaone elemente. Elementi zaobljenih uglova ili drugih nepravougaonih oblika zahtevaju izlaženje iz opsega CSS-a ili korišćenje sličica.

- Ne postoje striktna pravila o redosledu opisa i pripadajućih svojstava

Trenutno se opisi i svojstva mogu zadavati u bilo kom redu, ali ukoliko su u konfliktu, prednost ima poslednji zadati izraz.

- Nepostojanje promenljivih

CSS ne podržava promenljive ni u kakvom obliku. Promenljive bi poslužile za centralizovani opis određene boje, veličine ili čitavog skupa svojstava, što bi obezbedilo lakše održavanje i menjanje, ali i manje CSS dokumente. Sa trenutnim verzijama, pristupa se grupnom opisivanju nekoliko različitih klasa razdvojenim zarezom, da bi se umanjio problem nepostojanja promenljivih.

- Nedostatak kolumna (stubaca)

Kolumne teksta, poput novinarskih, se najčešće rešavaju plutajućim elementima ili tabelama, ali različiti brauzeri, na monitorima različitih rezolucija, veličina i odnosa strana će prikazati različite rezultate. Dodavanje jednostavnih deklaracija za definisanje kolumna bi rešilo ove probleme.