

Osnove CSS-a

skripta

1. CSS sintaksa

Kako je HTML prezentacijski jezik i njegova namjena nije bila ciljana na izgled web stranice, pojavio se **CSS**. CSS je kratica od (eng. Cascading Style Sheets). Radi se o stilskom jeziku, kojim upravljamo izgledom dokumenta napisanog pomoću HTML jezika. Izvršava se u web pregledniku.

CSS sintaksa-lista sastoji se od

- **imena** (selektor),
- skupa određenih **svojstava**
- koja imaju neku **vrijednost**.

Nakon imena-selektora slijede vitičaste zagrade `{ }` unutar kojih se navode svojstva i njihove vrijednosti. Između svakog svojstva i vrijednosti dolazi točka-zarez ;

ime (selector) → **h1** {
svojstvo → **font-size: 24px;** ← **vrijednost**
color: #23a8b4
}

2. Ubacivanje CSS-a u HTML stranicu

Stilovi se mogu ubaciti na tri načina:

1. direktno u **kodu** unutar pojedinog HTML taga
2. **CSS lista u HTML dokumentu** unutar taga `<head> ...</head>`
3. **CSS lista u posebnoj datoteci** s nastavkom `.css` (npr. `stil.css`)

2.1. U kodu

U kodu možemo definirati svojstva i vrijednosti CSS stila za pojedine HTML tagove. Ova svojstva i vrijednosti CSS stila odnose se samo na **dotičnu** HTML oznaku.

```
<body>
  <h1 style="font-size:18px; color:#000099; font-family:Arial, Helvetica, sans-serif">Naslov</h1>
  <h1>Naslov1 </h1>
</body>
```

2.2. CSS lista u HTML dokumentu

U HTML dokumentu i to unutar taga <head> ... </head> . Svojstva i vrijednosti CSS lista se primjenjuju samo na **dotičnu HTML datoteku** i ne utječu na ostale HTML datoteke.

Unutrašnje liste započinju s tagom <style>

<style type="text/css"> (atribut type="text/css" nije potrebno navoditi HTML5 jeziku)

.....

</style>.

Unutar taga **<style>** nalazi se liste tj. selektori sa svojim svojstvima i vrijednostima

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>CSS</title>
  <style type="text/css"> (atribut type="text/css" nije potrebno navoditi HTML5 jeziku)
    h1 {
      font-size: 24px;
      color: aquamarine;
    }
    p {
      font-size: 14px;
      color: #52776b;
    }
  </style>
</head>
```

2.3. Vanjska lista

Kreiramo vanjsku datoteku sa proizvoljnim imenom i ekstenzijom .css (npr. tomo.css).



Svojstva i vrijednosti stilova iz vanjske css datoteke mogu se primjenjivati na **sve HTML datoteke** u koje je pozvana vanjska css datoteka.

Da bismo **pozvali vanjsku CSS datoteku** potrebno je u HTML datoteku unutar oznaka **<head> ... </head>** napisati

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>CSS</title>
  <link href="tomo.css" (putanja do css datoteke) rel="stylesheet" type="text/css">

</head>
```

3.CSS selektori – načini primjene css stilova

3.1. Oznake ili tag

Svaka **HTML oznaka** može se koristiti kao selektor-ime stila. Takav stil **automatski** se **primjenjuje** na **sve** one elemente web stranice koji su pod tom HTML oznakom.

Primjer:

```
p {  
  color:#ffffff;  
  background: #000000;  
}
```

Ovaj css stil primjenjuje se na sve tekstove koji su u HTML kodu obuhvaćeni tagom **<p>....</p>**

```
img {  
  float: left;  
  margin-right: 20px;  
}
```

Ovaj css stil primjenjuje se na sve slike koji su u HTML kodu definirani tagom **.....**

Takoder možemo grupirati više HTML tagova koje **imaju ista** svojstva i vrijednosti, te time izbjegavamo ponavljanje

Primjer:

```
p, h2, ul {  
  color:ffffff;  
  background: #000000;  
}
```

3.2. Klase

Ako ne želimo promijeniti npr. veličinu teksta **svakom** odlomku na web stranici, već **samo jednom ili nekima** tada su nam oznake ili tagovi kao CSS selektori ograničavajući jer se automatski primjenjuju na **sve** elemente pod tom oznakom.

U ovakvim slučajevima možemo koristiti klase koje možemo **proizvoljno primijeniti** na **pojedine** elemente na web stranici. Svaka klasa započinje točkom i koristimo proizvoljno asocijativno ime.

Primjer:

```
.tekst {
color:ffffff;
background: #000000;
}
ili
p.tekst {
color:ffffff;
background: #000000;
}
```

Klasa se **ne primjenjuje automatski** već je moramo pozvati na svaku HTML oznaku na koju želimo da se primjeni.

```
<p class="tekst">Ovo je tekst</p>
```

Na jednu HTML oznaku možemo pozvati više različitih klasa. Nalaze se unutar istih navodnika, a između njih je razmak(space).

```
<p class="tekst paragraf1 znak">Ovo je tekst</p>
```

3.3. ID oznake

Svaka ID oznaka započinje # i koristimo proizvoljno asocijativno ime. Veoma su slični klasama i glavna razlika je što istu ID oznaku možemo primijeniti samo **jedanput** unutar jedne HTML stranice, a klasu **možemo primijeniti više puta** unutar jednog HTML dokumenta.

Primjer:

```
#tekst {
color:ffffff;
background: #000000;
}
ili
p#tekst {
color:ffffff;
background: #000000;
}
```

ID oznaka se **ne primjenjuje automatski** već je moramo pozvati za svaki elemente na koji želimo da se primjeni.

```
<p id="tekst">Ovo je tekst</p>
```

3.4. Kontekstualni selektori

Primjena kontekstualnih selektora jest da oblikuje određene „segmente“ HTML stranice. Ovi selektori imaju razmake unutar svog imena.

Primjer:

```
p strong {  
  color:white;  
  background: #000000;  
}
```

Ovaj kontekstualni selektor **djeluje samo** na tagove koji se nalazi unutar paragrafa <p>.

```
ol ol strong {  
  color:white;  
  background: #000000;  
}
```

Ovaj kontekstualni selektor **djeluje samo** na tagove koji se nalazi unutar liste koja se nalazi unutar još jedne liste

Kontekstualni tagovi mogu imati klase ili iD oznake kada ne želimo primjeniti na sve, već samo na nekim oznakama

Primjer:

```
ul.lista strong {  
  color:ffffff;           ili  
  background: #000000;  
}                          ul#lista strong {  
  color:ffffff;  
  background: #000000;  
}
```

Primjenjuje se samo na tagove koji se nalaze unutar taga na koju je primijenjena klasa lista ili id lista.

3.5. Pseudo css selektori

Pseudo selektori određuju **posebna stanja** odabranih html oznaka.

Razlikujemo

- pseudo elementi i
- pseudo klase

1. Pseudo elementi

Pseudo-elementi se **odvajaju dva puta s dvotočkom**. Pseudo-elementu možemo pridodati HTML oznaku, klasu ili ID oznaku

Standardni pseudo-elementi

```
::after (:after)           ::first-letter (:first-letter)  ::placeholder 🚩
::backdrop 🚩              ::first-line (:first-line)     ::selection
::before (:before)        ::grammar-error 🚩             ::slotted()
::cue                     ::marker 🚩                    ::spelling-error 🚩
::cue-region              ::part() 🚩
```

Primjer:

```
p::first-letter{
  color:white;
  font-size: 40px
}
```

Ova tip html selektora uz pseudo element nam kaže da će prvo slovo u svakog odlomka biti u drugoj boji i veličini

Primjer:

```
p.tekst:first-letter {
  color:#ff0000
}
```

Ova tip klase kao selektora uz pseudo element nam kaže da će prvo slovo odlomka koji ima klasu tekst biti u drugoj boji i veličini

2. Pseudo klase

Pseudoklase se **odvajaju dvotočkom**. Pseudo klasu možemo pridodati HTML oznaku, klasu ili ID oznaku.

Standardne pseudoklase

```
:active
:any-link 🔗
:blank 🔗
:checked
:current 🔗
:default
:defined
:dir() 🔗
:disabled
:drop 🔗
:empty
:enabled
:first
:first-child
:first-of-type
:fullscreen 🔗
:future 🔗
:focus
:focus-visible 🔗
:focus-within
:has() 🔗
:host
:host()
:host-context() 🔗
:hover
:indeterminate
:in-range
:invalid
:is() 🔗
:lang()
:last-child
:last-of-type
:left
:link
:local-link 🔗
:not()
:nth-child()
:nth-col() 🔗
:nth-last-child()
:nth-last-col() 🔗
:nth-last-of-type()
:nth-of-type()
:only-child
:only-of-type
:optional
:out-of-range
:past 🔗
:placeholder-shown 🔗
:read-only
:read-write
:required
:right
:root
:scope
:state() 🔗
:target
:target-within 🔗
:user-invalid 🔗
:valid
:visited
:where() 🔗
```

Primjer:

```
a: hover {
  color: #ff00ff;
}
```

Ova tip html kao selektora uz pseudo klasu nam kaže da kada mišom pređemo preko svih linkova (imaju html oznaku a) doći će do promjene boje linka.

Primjer:

```
ul:nth-child(2n) {  
  color: green;  
}
```

Ova tip html kao selektora uz pseudo klasu nam kaže da će **sve parne stavke** u svim listama unutar html dokumenta imati zelenu boju,

Primjer:

```
ul ul:nth-child(2n) {  
  color: green;  
}
```

Ova tip kontekstualnog selektora kao selektora uz pseudo klasu nam kaže da će **sve parne stavke** dobiti zelenu boju koje se nalaze unutar liste `` koja se nalazi unutar još jedne liste ``

Primjer:

```
ul.skola li:first-child {  
  background: yellow;  
}
```

Ova tip klase kao selektora uz pseudo klasu nam kaže da će **prva stavke** dobiti pozadinsku boju unutar liste `` kojoj je pridodana klasa `skola`.

4. Svojstva i vrijednosti CSS stila za tekst

4.1. Vrsta pisma – font

U CSS definiramo **vrstu pisma** pomoću svojstva

```
{  
  font-family: Verdana, Arial, Helvetica, sans-serif;  
}
```

Na webu koristimo 5 familija fontova:

- serif
- sans-serif
- cursive
- fantasy
- monospace

Na većini web stranica koriste se najčešće dvije familije fontova:

1. Serifni fontovi – kao što su Times, Garamond, Georgia,...
2. Bez serifa fontovi (sans serif)- kao što su Arial, Helvetika, Verdana

Problem kod definiranja fontova je taj što font kojim smo izabrali za web stranicu, se neće prikazati ako korisnik na svom računalu nema instaliran taj font. Zbog toga se definira veći broj fontova i na kraju se navede sans-serif ili serif te se kaže računalu ako korisnik nema niti jedan od prije navedenih fontova, prikaži bilo koji bez serifni ili serifni font koji postoji na korisnikovom računalu.

font-family: Verdana, Arial, Helvetica, sans serif; govori pregledniku neka nam prikaži Verdana font, a ako ne postoji u računalu prikaži Arial font, ako Arial ne postoji prikaži Helvetiku font, a ako ne postoji Helevetica prikaži bilo koji bez serifni font koji postoji na računalu.

Danas je taj problem riješen preko **google fonts** ili korištenjem **css3 @font-face**

4.2. Veličina pisma

U CSS definiramo **veličinu pisma** pomoću svojstva

```
{  
  font-size: 10px;  
}
```

Vrijednosti se izražavaju brojkama, a mogu se zadavati u različitim mjernim jedinicama (pikselima (px), tipografskim točkama (pt), inčima (in), centimetrima (cm), milimetrima (mm), postocima(%), emovima (em), exovima (ex)).

4.3. Stil pisma

U CSS definiramo **stil pisma** pomoću svojstva

```
{  
  font-style: normal;  
}
```

Vrijednosti mogu biti

normal,

italic (kurziv verzija fonta koji postoji) i

oblique (normalni font koje je računalo samo udesno ukosilo-deformacija).

4.4. Razmak između redova

U CSS definiramo **razmak između redova** pomoću svojstva

```
{  
  line-height: 20px;  
}
```

Vrijednosti se izražavaju brojkama, a mogu se zadavati u različitim mjernim jedinicama (pikselima (px), tipografskim točkama (pt), inčima (in), centimetrima (cm), milimetrima (mm), postocima(%), emovima (em), exovima (ex)).

4.5. Položaj crte u odnosu na tekst

U CSS definiramo **različit položaj crte** u odnosu na tekst pomoću svojstva

```
{  
  text-decoration: underline;  
}
```

Vrijednosti mogu biti

underline (podcrtan tekst),
overline (crta iznad teksta),
line-through (precrtan tekst),
blink (blinkanje teksta),
none (nema crte)

4.6. Debljina slova

U CSS definiramo **debljinu slova** pomoću svojstva

```
{  
  font-weight: bold;  
}
```

Vrijednosti mogu biti **bold** (podebljan tekst), a također vrijednosti se mogu zadavati 100, 200, 300, 400, 500, 600, 700, 800, 900, gdje je 900 najdeblje slovo. Bitno je imati rezove pisama u različitim debljinama.

4.7. Zamjena slova velikim umanjenim

U CSS definiramo **zamjenu slova velikim, ali umanjenim** pomoću svojstva

```
{  
  font-variant: small-caps;  
}
```

Vrijednosti mogu biti

small-caps (velika umanjena slova),
normal

4.8. Zamjena slova velikim ili malima

U CSS definiramo **zamjenu slova** pomoću svojstva

```
{  
  text-transform: lowercase;  
}
```

```
}
```

Vrijednosti mogu biti

capitalize (prvo slovo u riječi veliko),

uppercase (sva slova velika)

lowercase (sva slova mala)

4.9. Boja

U CSS definiramo **boju** teksta pomoću svojstva

```
{  
  color: #006666;  
}
```

Boje na ekranu se definiraju s tri boje: **crvenom, zelenom i plavom**. Tako definirane boje nazivaju se RGB od red, green i blue. Naziva se i aditivni sistem miješanja boja, jer se boje zbrajaju i miješanjem svih tri boja u punom intenzitetu dobivamo bijelu boju.

Svaka od te tri boje može imati 256 različitih nijansi tj. tonova. Ukupno boja koje RGB model može reproducirati je $256 \times 256 \times 256 = 16\,777\,216$ nijansi.

U **heksadecimalnom sustavu** u css-u koristimo šest znamenki s prefiksom „#“(može i tri). Prve dvije znamenke su heksadecimalna vrijednost crvene boje, druge dvije su za zelenu boju i zadnje dvije za plavu boju. Svaka boja može imati raspon od (00-ff)

Primjer: color: #00ffaa ili color: #0fa
color: #abc1ea

Za dodavanje prozirnosti mpžemo dodati dvije dodatne znamenke između 00 i FF.

Primjer: color: #00ffaa00 - prozirnost potpuna
color: #abc1eaff – nema prozirnosti

U **dekadskom sustavu** u css-u koristimo raspon boja od (0 – 255). Prva brojčana vrijednost je za crvenu boje, druga brojčana vrijednost za zelenu boju i treća brojčana vrijednost za plavu boju.

Primjer: color: rgb(0, 115, 172)

RGBA vrijednosti boja proširenje su RGB vrijednosti boja s alfa kanalom - koji određuje neprozirnost objekta između 0 i 1.

Primjer: `color: rgba(255,34,123,0)` - prozirnost potpuna

Primjer: `color: rgba(255,34,123,1)` - nema prozirnosti

HSL je kratica za nijansu, zasićenost i svjetlinu - i predstavlja cilindrično-koordinatni prikaz boja.

Vrijednost HSL boje određena je funkcijom `hsl()`: **`hsl(nijansa, zasićenost, svjetlina)`**

Nijansa je stupanj na kotaču boja (od 0 do 360) - 0 je crvena, 120 je zelena, 240 je plava.

Zasićenost je postotna vrijednost; 0% znači nijansu sive, a 100% je puna boja.

Svjetlina je također postotak; 0% je crno, 100% je bijelo.

Primjer: `color: hsl(120,100%,50%)`

Vrijednost css svojstva može biti **ime neke** boje. Međutim, u tom slučaju imamo na raspolaganju manji opseg boja, jer sve nijanse boja nemaju svoje ime.

Primjer: `color: lightblue`

4.10 Razmak između riječi

U CSS definiramo **razmak između riječi** pomoću svojstva

```
{  
  word-spacing: 20px;  
}
```

Vrijednosti se izražavaju brojkama, a mogu se zadavati u različitim mjernim jedinicama (pikselima (px), tipografskim točkama (pt), inčima (in), centimetrima (cm), milimetrima (mm), postocima(%), emovima (em), exovima (ex)).

4.11 Razmak između slova

U CSS definiramo **razmak između slova** pomoću svojstva

```
{  
  letter-spacing: 20px;  
}
```

Vrijednosti se izražavaju brojkama, a mogu se zadavati u različitim mjernim jedinicama (pikselima (px), tipografskim točkama (pt), inčima (in), centimetrima (cm), milimetrima (mm), postocima(%), emovima (em), exovima (ex)).

4.12 Poravnanje teksta

U CSS definiramo **poravnanje teksta** pomoću svojstva

```
{  
  text-align: left;  
}
```

Vrijednosti mogu biti

left (lijevo poravnanje),

center (centralno),

right (desno) i

justify (obostrano)

4.13 Uvlaka

U CSS definiramo **uvlaku** između slova pomoću svojstva

```
{  
  text-indent: 20px;  
}
```

Vrijednosti se izražavaju brojkama, a mogu se zadavati u različitim mjernim jedinicama (pikselima (px), tipografskim točkama (pt), inčima (in), centimetrima (cm), milimetrima (mm), postocima(%), emovima (em), exovima (ex)).

Vrijednost uvlake iznosi koliko i veličina teksta.

5. Svojstva i vrijednosti CSS stila za pozadinu

Preko CSS-a možemo ubaciti pozadinu svakom html elementu. To znači da naslov, tekst, tablica ili neki drugi html element može imati svoj stil za pozadinu. Ako želimo pozadinu primijeniti na cijelu stranicu, tada je zadajemo uz oznaku body.

5.1. Boja pozadine

U CSS definiramo **boju pozadine** pomoću svojstva

```
{  
  background-color: #0066CC;  
}
```

Vrijednost može biti definirana tekstualno, heksadecimalno ili u dekadskom sustavu

5.2. Slika u pozadini

U CSS definiramo **sliku u pozadini** pomoću svojstva

```
{  
  background-image: url(grupa_a/slike/bgON.gif);  
}
```

Prvi dio url je **putanja** (grupa_a/slike), a drugi dio je **naziv slike i format slike** (bgON.gif)

5.3. Ponavljanje pojavljivanja slike

U CSS definiramo **ponavljanje slike** pomoću svojstva

```
{  
  background-repeat: repeat-x;  
}
```

Slika se ponavlja kao zidne pločice, ako je manja od dimenzija pozadine koju popunjava. Vrijednosti mogu biti

repeat (po horizontali i vertikali),

repeat-x (samo po horizontali) i

repeat-y (samo po vertikali) i

no repeat (bez ponavljanja)

5.4. Statičnost slike

U CSS definiramo **statičnost slike** pomoću svojstva. Ova opcija određuje da li je slika u pozadini statična, tako da stranica prelazi preko nje ili se pozadina kreće zajedno sa stranom.

```
{  
  background-attachment: scroll;  
}
```

Vrijednosti mogu biti

scroll (pozadina se zajedno kreće sa stranicom kada se dokument skrola),

fixed (pozadina se ne kreće sa stranicom kada se dokument skrola).

5.5. Horizontalna pozicija slike

U CSS definiramo **horizontalna poziciju slike** kada je veličina pozadinske slike manja od veličine pozadine koju popunjava, i to pomoću svojstva

```
{  
  background-position: center;  
}
```

Postoji mogućnost relativnog postavljanja u odnosu na element na koji se postavlja. Vrijednosti mogu biti

left(lijevo),

right(desno) i

center (centralno),

Također kao vrijednosti mogu se zadavati brojčane vrijednost u odnosu na gornji lijevi vrh elementa na koji se postavlja. Mjere mogu biti isto kao kod veličine slova (px, pt, in, cm, mm, %, em, ex).

5.6. Vertikalna pozicija slike

U CSS definiramo vertikalnu poziciju slike kada je veličina pozadinske slike manja od veličine pozadine koju popunjava, i to pomoću svojstva

```
{  
  background-position: top;  
}
```

Postoji mogućnost relativnog postavljanja u odnosu na element na koji se postavlja. Vrijednosti mogu biti

top(vrh),

bottom(dno) i

center (centralno),

Također kao vrijednosti može se zadavati numerička brojka koja predstavlja pomak od gornjeg lijevog vrh pozadinskog elementa na koji se postavlja pozadinska slika. Mjere mogu biti isto kao kod veličine slova (px, pt, in, cm, mm, %, em, ex).

5.7. Horizontalna i vertikalna pozicija slike

U **većini slučajeva** zadaju se istovremeno obje vrijednosti i za horizontalno i za vertikalno pozicioniranje. Prva vrijednost je horizontalno pozicioniranje, a druga vertikalno pozicioniranje.

```
{
    background-position: left top;
}
ili apsolutno
{
    background-position: 20px 50px;
}
```

5.8. Dimenzija pozadinske slike

U CSS možemo promijeniti dimenzije pozadinske slike kada je njena veličina manja od veličine pozadine koju popunjava, i to pomoću svojstva

```
{
    background-size: cover;
}
```

Vrijednosti mogu biti

cover (proporcionalna promijena veličine pozadinske slike kako bi se pokrila cijeli pozadina, a pri tome ne mora biti vidljiva cijela slika pozadinska slika)

contain (proporcionalna promijena veličine pozadinske slike koja pokriva pozadinu ali je uvijek vidljiva cijela pozadinska slika).

```
{  
background-size: 40% 70%;  
}
```

Također kao vrijednosti mogu se zadavati brojke. Mjere mogu biti isto kao kod veličine slova (px, pt, in, cm, mm, %, em, ex). Prva numerička brojka je širina, a druga visina.

6. Svojstva i vrijednosti CSS stila za rub-liniju

Preko CSS-a možemo ubaciti rub-liniju svakom elementu. Rub možemo zadavati za :

- **top** (vrh)
- **right** (desno)
- **bottom** (dno)
- **left** (lijevo)

Kod svakog ruba-linije moramo definirati tri svojstva:

- **vrstu ruba-linije**
- **debljina okvira- linije**
- **boju okvira-linije**

6.1.Vrsta ruba-linije

U CSS definiramo vrstu ruba-linije pomoću svojstva

```
{  
  border-style: dashed;  
}
```

Vrijednosti mogu biti

solid(obična),
dotted (točkasta),
dashed (crti) i
double (dvije),
itd .

Ako želimo zadati vrstu ruba samo sa jedne strane (npr vrh) definiramo

```
{  
  border-top-style: dashed;  
}
```

6.2. Debljina ruba- linije

U CSS definiramo **debljinu ruba** pomoću svojstva

```
{  
  border- width: 2px;  
}
```

Također kao vrijednosti mogu se zadavati brojke. Mjere mogu biti isto kao kod veličine slova (px, pt, in, cm, mm, %, em, ex).

Ako želimo zadati debljinu samo sa jedne strane (npr vrh) definiramo

```
{  
  border-top-width: dashed;  
}
```

6.3. Boja ruba- linije

U CSS definiramo **boju ruba** pomoću svojstva

```
{  
  border -color: #0099CC;  
}
```

Vrijednost može biti definirana tekstualnom, heksadecimalnom ili u dekadskom sustavu

Ako želimo zadati boju ruba samo sa jedne strane pišemo:

```
{  
  border-top-color: #993300  
}
```

6.3.Sintaksa pisanja

Ako imamo **rub-liniju samo sve jedne strane (npr.vrh)** oko nekog html elementa.

```
{  
  border-top-width: 1px;  
  border-top-style: solid;  
  border-top-color:#993300  
}  
ili jednostavnije  
{  
  border-top: 1px solid #993300  
}
```

(Redosljed pisanja vrijednosti nije bitan.)

Ako imamo **isti rub-liniju sa sve 4 strane** oko nekog html elementa.

```
{  
  border-top: 1px solid #993300;  
  border-bottom: 1px solid #993300;  
  border-left: 1px solid #993300  
  border-right: 1px solid #993300  
}  
ili jednostavnije  
{  
  border: 1px solid #993300  
}
```

Ako imamo **rub-liniju sa sve 4 strane a razlikuju se sa svih strana samo po boji** oko nekog html elementa.

```
{  
  border: 1px solid;  
  border-top-color: #993300;  
  border-right-color: #22aa11;  
  border-bottom-color: #aa12ad;  
  border-left-color: #33a145;  
}  
ili jednostavnije  
{  
  border: 1px solid;  
  border-color: #993300 #22aa11 #aa12ad #33a145;  
                top      right      bottom      left  
}
```

Može vrijediti za bilo koje svojstvo ruba-linije (debljinu, vrstu ili boju)

Ako imamo **rub-liniju sa sve 4 strane a razlikuju se samo po boji, sa time da su gornji i donju rub iste boje, te lijevi i desni rub iste boje** oko nekog html elementa.

```
{  
  border: 1px solid;  
  border-top-color: #993300;  
  border-right-color: #22aa11;  
  border-bottom-color: #993300;  
  border-left-color: #22aa11;  
}  
ili jednostavnije  
{  
  border: 1px solid;  
  border-color: #993300 #22aa11;  
                top i bottom      right i left  
}
```

Može vrijediti za bilo koje svojstvo ruba-linije (debljinu, vrstu ili boju)

7.Svojstva i vrijednosti CSS stila - display

Svaki HTML element ima zadanu vrijednost prikaza, ovisno o vrsti elementa. Postoje dvije vrijednosti prikaza:

- blok i
- inline element

Element na razini bloka uvijek započinju u **novom retku** i zauzima **punu dostupnu širinu** (proteže se lijevo i desno koliko god može).

```
<address>    <article>    <aside>    <blockquote>  <canvas>    <dd>
<div>        <dl>          <dt>        <fieldset>   <figcaption> <figure>
<footer>    <form>        <h1>-<h6>  <header>    <hr>        <li>
<main>      <nav>        <noscript> <ol>        <p>          <pre>
<section>   <table>      <tfoot>    <ul>        <video>
```

Inliine element **ne započinje s novim retku** i zauzima samo onoliko **širine** koliki je **sadržaj elementa**

```
<a>          <abbr>        <acronym>   <b>          <bdo>        <big>
<br>        <button>     <cite>      <code>       <dfn>        <em>
<i>         <img>        <input>     <kbd>        <label>      <map>
<object>    <output>    <q>         <samp>       <script>     <select>
<small>    <span>      <strong>    <sub>        <sup>        <textarea>
<time>     <tt>        <var>
```

7.1.Display

CSS svojstvo prikaza određuje da li ćemo html element tretira kao blok ili inline element. Također, omogućuje i layout elemenata koji se koriste u rasporedu građenju strukture web stranice.

```
{
  display: block;
}
```

Imamo veliki broj vrijednostosti. Neke od vrijednosti mogu biti:

- inline** (prikaži element kao inline i svojstva biti širine i visine neće se primjenjivati)
- block** (prikaži element kao block i svojstva biti širine i visine će se primjenjivati).

inline-block (prikaži element kao inline i svojstva biti širine i visine će se primjenjivati).

7.2.Širina – width

U CSS definiramo širinu pomoću svojstva

```
{  
  width: 500px  
}
```

Veličina se može zadavati u **pikselima(px)**, **tipografskim točkama(pt)** gdje je 1pt = 0,376mm, (na staru HTML oznaku font size ="3" koja kad se preračuna iznosi 12pt ili 16px) **inčima (in)** gdje je 1in=2,54cm, **centimetrima(cm)**, **milimetrima (mm)**, **postocima(%)****emovima (em)**, **exovima(ex)** gdje veličina odgovara veličini malog slova x dotičnog fonta

7.3.Visina –height

U CSS definiramo veličinu pisma pomoću svojstva

```
{  
  height: 500px  
}
```

Veličina se može zadavati u **pikselima(px)**, **tipografskim točkama(pt)** gdje je 1pt = 0,376mm, (na staru HTML oznaku font size ="3" koja kad se preračuna iznosi 12pt ili 16px) **inčima (in)** gdje je 1in=2,54cm, **centimetrima(cm)**, **milimetrima (mm)**, **postocima(%)****emovima (em)**, **exovima(ex)** gdje veličina odgovara veličini malog slova x dotičnog fonta

7,3,Float

Float – koristi se za odvajanje elemenata od ostatka strane. Ostali elementi mogu teći preko ovog elementa ili da se prekidaju oko njega

```
{  
  float: left;  
}
```

Vrijednosti mogu biti **left** (lijevo) i **right**(desno).

7.4.Clear

Koristi se za definiranje područja u kojem se ne dozvoljava preklapanje elemenata

```
{  
  clear:left;  
}
```

Vrijednosti mogu biti **left** (lijevo), **right**(desno) i **both**(odostrano).

7.Svojstva i vrijednosti CSS stila – box model

U CSS-u se termin "box model" koristi kada se govori o dizajnu i izgledu. CSS box-model je u biti prostor koji obavija svaki HTML element.

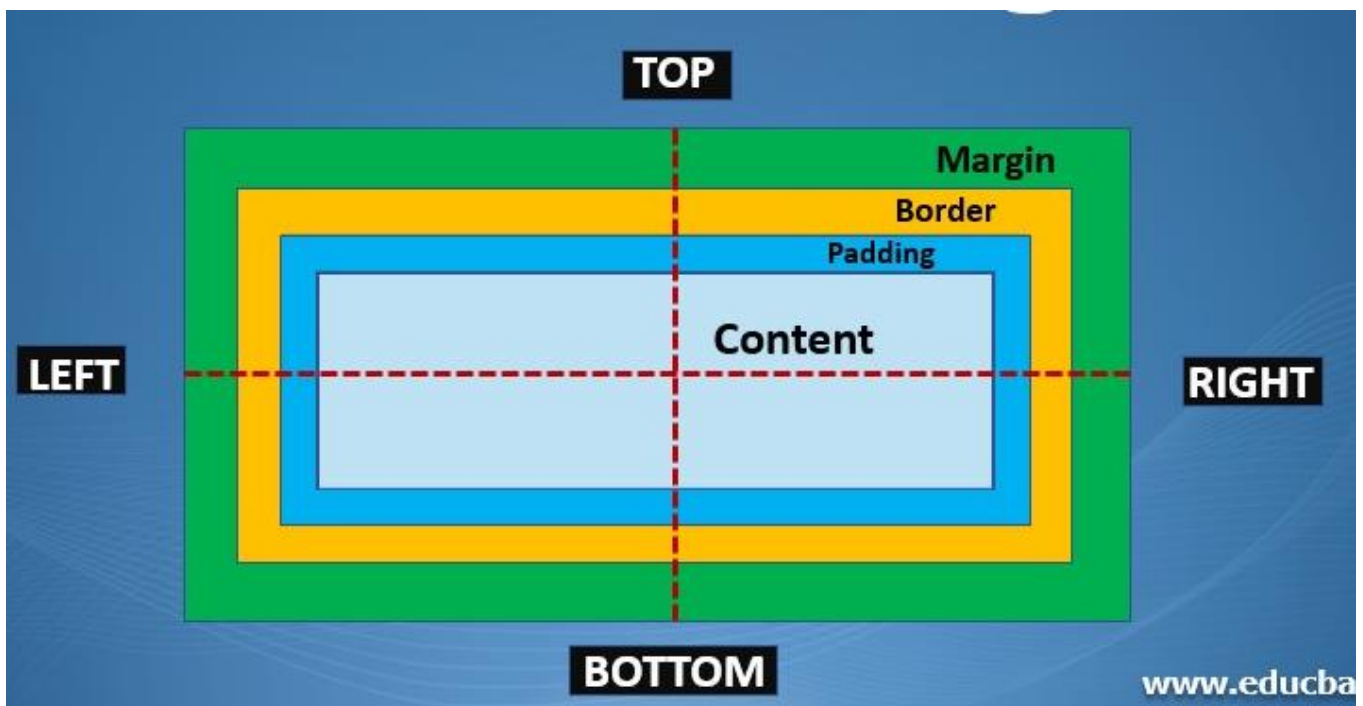
Sastoji se od:

- paddinga.
- margina,
- ruba-border i
- sadržaja.

7.1 Padding

Padding definira razmak između sadržaja elementa i njegovog ruba-bordera. Imamo 4 vrijednosti:

- vrh (**top**),
- lijevo (**left**),
- desno (**right**) i
- noge (**bottom**)



```
{  
  padding: 10px;  
}
```

U ovom slučaju vrijednosti su **iste** za vrh(**top**), lijevo(**left**), desno(**right**) i noge(**bottom**)

```
{  
padding-top: 5px;  
padding-right: 10px;  
padding-bottom: 15px;  
padding-left: 20px;  
}  
ili  
padding: 5px 10px 15px 20px;
```

Možemo zadavati padding posebno za vrh(**top**), desno(**right**), noge(**bottom**) i lijevo(**left**), Ako ostavimo samo padding, tada je **prva** vrijednosti za **vrh**, **druga** vrijednost je za **desno**, **treća** vrijednost je za **noge** i četvrta vrijednost je za **lijevo**.

```
{  
padding: 20px 10px;  
}
```

U ovom slučaju **prva** vrijednosti je za **vrh i noge**, a **druga** vrijednost za **lijevo i desno**.

```
{  
padding: 20px 10px 30px;  
}
```

U ovom slučaju **prva** vrijednosti je za **vrh**, **druga** vrijednost je za **lijevo i desno**, a **treća** vrijednost je za **noge**.

Margin

Definira rastojanje od ruba dvije različite HTML oznake. Imamo 4 vrijednosti:

- vrh (**top**),
- lijevo (**left**),
- desno (**right**) i
- noge (**bottom**)

```
{  
  margin: 10px;  
}
```

U ovom slučaju vrijednosti su **iste** za vrh(**top**), lijevo(**left**), desno(**right**) i noge(**bottom**)

```
{  
  margin-top: 5px;  
  margin-right: 10px;  
  margin-bottom: 15px;  
  margin-left: 20px;  
}  
ili  
margin: 5px 10px 15px 20px;
```

Možemo zadavati padding posebno za vrh(**top**), desno(**right**), noge(**bottom**) i lijevo(**left**),
Ako ostavimo samo padding, tada je **prva** vrijednosti za **vrh**, **druga** vrijednost je za **desno**, **treća** vrijednost je za **noge** i četvrta vrijednost je za **lijevo**.

```
{  
  margin: 20px 10px;  
}
```

U ovom slučaju **prva** vrijednosti je za **vrh i noge**, a **druga** vrijednost za **lijevo i desno**.

```
{  
  margin: 20px 10px 30px;  
}
```

U ovom slučaju **prva** vrijednosti je za **vrh**, **druga** vrijednost je za **lijevo i desno**, a **treća** vrijednost je za **noge**.