

CSS3

U ovoj jedinici ćemo se upoznati sa CSS verzijom 3.

CSS3 i CSS2 odnos

Kao što već znamo, CSS (Cascading Style Sheets) je jezik koji koristimo za stilizaciju i izmenu prikaza HTML elemenata. CSS3 je trenutno najnovija verzija CSS specifikacije (jezika). Često se spominje zajedno sa HTML5, ali su to nezavisni standardi, mada odlično funkcionišu zajedno.

CSS3 sadrži praktično sve što se može naći u CSS2.1 specifikaciji, ali dodaje i nove funkcionalnosti, kako bi za određene detalje izbegli nesemantičke elemente, složene solucije „pomoću štapa i kanapa“ i tako dalje. Praktično je dosta funkcionalnosti proisteklo iz prakse i potreba autora. Na primer, ranije, da bi postavili gradijent ili zaobljene uglove za određeni element morali smo kreirati nove slike u Photoshopu ili sličnom alatu, dodavati HTML tagove i više CSS opisa. Sve to nepotrebno komplikuje razvoj. Sada možemo jednostavno postaviti direktno jedan CSS3 opis, tačnije svojstvo i stilizovati gradijent, zaobljene uglove i slično. Nije potrebno posebno naglašavati da time dobijamo čistiji kod, lakše kasnije održavanje i eventualne izmene, manje slika i drugih elemenata, brže učitavanje i tako dalje.

Nove funkcionalnosti u CSS3 uključuju proširene i kompleksnije selektore, senke nad elementima, zaobljene uglove, višestruke pozadine, nove načine pisanja i postavljanja boja, animacije, providnost i tako dalje.

CSS3 moduli

Za razliku od CSS2, koji je kompaktna celina i jedna ogromna specifikacija, CSS3 je podeljen u takozvane **module**. Svaki modul se razvija skoro nezavisno i zadržava kompatibilnost sa ostalim modulima, kao i starijim verzijama jezika.

Neki od najvažnijih modula su:

- Selectors (selektori)
- Box Model
- Backgrounds and Borders (pozadine i borderi)
- Text Effects (tekst efekti)
- 2D Transformations (2D transformacije)
- 3D Transformations (3D transformacije)
- Animations (animacije)
- Multiple Column Layout (prelom teksta – kolone)
- I tako dalje...¹

¹ Redovno ažuriran spisak modula možete naći na: <http://www.css3.info/modules/>

Sve ove module ne treba pamtit napamet. Mi autori koristimo funkcionalnosti iz svih njih po potrebi. Bitno je znati funkcije, svojstva i načine primene, a kome modulu pripadaju je već stvar razvoja CSS jezika.

Ne zaboravite da je CSS3 još uvek u razvoju i da može doći do promena u nekim funkcijama.

Prefiksi i sufiksi

Pomoću CSS prefiksa, tvorci browsera omogućavaju testiranje i ograničenu upotrebu novih CSS mogućnosti. Prefiksi za određene browsere su:

- Chrome: `-webkit-`
- Firefox: `-moz-`
- Internet Explorer: `-ms-`
- Opera: `-o-`
- Safari: `-webkit-`
- Android: `-webkit-`
- iOS: `-webkit-`

Primetićete da Chrome, Safari, ali i Android, iOS i još neki browseri imaju isti prefiks – *webkit*, jer se baziraju na istoj osnovi, odnosno na istom jezgru (engine) pod nazivom webkit.

Uglavnom, napredniji, noviji CSS se kreira tako što standardnom CSS svojstvu dodamo željeni prefiks. Na primer, ako želimo dodati CSS3 tranziciju u nekom opisu, možemo koristiti transition svojstvo, ali ćemo dodati prefikse:

```
-webkit-transition: all 4s ease;  
-moz-transition: all 4s ease;  
-ms-transition: all 4s ease;  
-o-transition: all 4s ease;  
transition: all 4s ease;
```

Naravno, ne možemo uvek dodati prefiks, jer sam browser mora prepoznati svojstvo i znati šta da čini sa njim. Pored toga, ne upisujemo uvek iste vrednosti za različita, prefiksom određena svojstva. Videćemo u nastavku takav primer kod gradijenata.

Najčešća nova svojstva

Border-radius

Možda najčešće pominjano CSS3 svojstvo jeste border-radius. Njime za postavljen border možemo definisati ugao zakrivljenja. Pogledajmo u kodu:

```
.box {  
    border:1px solid #333;  
    border-radius:10px;  
}
```

Na ovaj način smo elementu sa klasom *box* postavili jednostavnu sivu ivicu (border) koja je debljine 1 piksel i pri tom su uglovi zaobljeni pod radijusom od 10 piksela.

Na ovom primeru lepo se vidi kako starija CSS specifikacija (border) i novija (border-radius) sarađuju.

Pored podjednagog zaobljenja, možemo definisati i posebne vrednosti za sva četiri ugla. Upisaćemo četiri vrednosti koje su za uglove: gornji-levi, gornji-desni, donji-desni, donji-levi:

```
border-radius:10px 0 20px 0;
```

ovako unet kod bi zaoblio gornji levi i donji desni ugao.



Primeri u browseru za neke border-radius primene

Box-shadow

Ovo svojstvo definiše senku koju određeni element baca „ispod” sebe. Zahteva nešto više parametara. U kodu postavljamo na primer:

```
.box {  
  box-shadow: 10px 20px 5px 0px #000000;  
}
```

Ovaj primer bi postavio senku nad elementom i to senku koja je pomerena za 10px udesno od elementa, pomerena za 20px nadole od elementa, 5px je vrednost zamućenja, 0px je

spread vrednost (koja proširuje senku) i na kraju dolazi boja same senke. U ovom primeru je korišćen hex, dok može biti i rgba, na primer.

Dakle, svojstvo box-shadow traži sledeće vrednosti:

```
box-shadow: hor-pozicija vert-pozicija zamućenje(blur) spread boja;
```

Ukoliko želimo senku koja pada unutar elementa, možemo upisati i ključnu reč *inset*.



Primer u browseru za drop shadow

Background size

Relativno čest problem ranije je predstavljala nemogućnost skaliranja slika u pozadini. Odnosno, možemo pomerati background, ali ne možemo promeniti veličinu. U CSS3 novo svojstvo je background-size koje u tome pomaže.

```
.box {  
    background:url(slika.png);  
    background-size:20px 100px;  
    background-repeat:no-repeat;  
}
```

Slika će u pozadini biti u ovom primeru 20x100 piksela, iako je možda originalna slika potpuno drugačijih dimenzija.

Kao i kod slika drugde u HTML/CSS-u, budite oprezni sa promenom veličine i činite to samo kad je neophodno, jer opet se učitava potpuna, velika slika i može doći do degradacije kvaliteta.

Višestruke pozadine

Pošto smo u domenu pozadinskih slika, zgodno je pomenuti da CSS3 omogućava više pozadinskih slika nad istim elementom. Tako da možemo pisati npr:

```
body {
```

```
background: url(prva.png) center bottom no-repeat, url(druga.png)
left top no-repeat;
}
```

Prvo se prikazuje poslednja unesena, a preko nje redom unazad u kodu.

Ukoliko želimo, možemo pisati na duži način (posebno background-image, background-position, background-repeat i td.). U tom slučaju, opet odvajamo zarezom i pišemo za svaku sliku. Konkretno na našem primeru:

```
body {
background-image: url(prva.png), url(druga.png);
background-position: center bottom, left top;
background-repeat: no-repeat, no-repeat;
}
```

Text-shadow

Kao što i samo ime kaže, svojstvo text-shadow postavlja senku koju tekst baca na element ispod njega. U kodu postavljamo, na primer:

```
text-shadow: 2px 1px 3px #000000;
```

Slično poput senke elementa, potrebno je više vrednosti. Prvo upisujemo horizontalnu poziciju (2px u primeru), zatim vertikalnu (1px), zamućenje (3px) i na kraju boju.

Gradijenti

Kreiranje gradijenata (prelaza boja) je nešto komplikovanije zbog različite podrške browsera i različitog definisanja. Zbog toga koristimo prefikse za browsere koje smo pomenuli ranije. Pogledajmo jedan primer koji kreira pozadinu u gradijentu, od svetlije ka tamnijoj crvenoj:

```
background: #d61111; /* Old browsers */

background: -moz-linear-gradient(top, #d61111 0%, #7c0000 100%);
/* FF3.6+ */

background: -webkit-gradient(linear, left top, left bottom, color-stop(0%,#d61111), color-stop(100%,#7c0000));
/* Chrome,Safari4+ */

background: -webkit-linear-gradient(top, #d61111 0%,#7c0000 100%);
/* Chrome10+,Safari5.1+ */

background: -o-linear-gradient(top, #d61111 0%,#7c0000 100%);
/* Opera 11.10+ */

background: -ms-linear-gradient(top, #d61111 0%,#7c0000 100%);
/* IE10+ */

background: linear-gradient(to bottom, #d61111 0%,#7c0000 100%);
```

```

/* W3C */

filter: progid:DXImageTransform.Microsoft.gradient(
startColorstr='#d61111', endColorstr='#7c0000',GradientType=0 );
/* IE6-9 */

```

Vidimo da je manje-više za svaki browser drugačije definisana vrednost. Pored toga, definisana je i „obična” pozadina za stare browsere na početku.

Ovaj konkretan kod preuzet je sa jednog odličnog generatora gradijenata na adresi: <http://www.colorzilla.com/gradient-editor>. Zbog komplikovanog načina pisanja, preporučujem ovaj ili slične alate kako bismo izbegli moguće greške.



Primer u kome je DIV-u za pozadinu postavljen gradijent

Providnost

Kada smo govorili o bojama, razjasnili smo i detalje vezano za **Alpha** komponentu boja, kojom možemo definisati providnost. Radi podsećanja, možemo pisati:

```

div {
    color: rgba(255,0,255,0.5);
}

div {
    color: hsla(80,50%,60%,0.8);
}

```

I kod klasičnog RGB i HSL prikaza smo dodali četvrtu A (alpha) komponentu. Vrednost može biti u rasponu 0 do 1.

CSS 2D i 3D transformacije i tranzicije

CSS3 je doneo i mogućnosti transformisanja elemenata prilikom prikaza. Na primer, možemo promeniti veličinu elemenu, rotirati ga, pomeriti, deformisati i tako dalje. Takođe,

možemo animirati pokret i slično. Ipak, sve ove funkcionalnosti su manje-više u test fazi i daleko su od neke realne primene na sajtovima.

@font-face

Da bi primenili neki nestandardni font za naše tekstove u HTML dokumentima, možemo iskoristiti @font-face opciju. Kod nje je specifično što moramo prvo imenovati font na stranici, zatim pozvati fajl (fajlove) sa samim fontom i definisati format, a tek onda kao i ranije dodeliti određeni font našim elementima.

U CSS kodu to bi izgledalo, na primer, ovako:

```
@font-face {
    font-family: 'DejaVuSansBook';

    src: url('DejaVuSans-webfont.eot');
    src: url('DejaVuSans-webfont.eot?#iefix') format('embedded-opentype'),
        url('DejaVuSans-webfont.woff') format('woff'),
        url('DejaVuSans-webfont.ttf') format('truetype'),
        url('DejaVuSans-webfont.svg#DejaVuSansBook') format('svg');
}

body {
    font-family: 'DejaVuSansBook', Arial, Helvetica, sans-serif;
}
```

Ono što je specifično kod @font-face jeste da različiti browseri podržavaju različite formate fajlova. Na primer, Internet Explorer podržava samo EOT standard. Da bi sve funkcionisalo kako treba na različitim browserima, moramo pripremiti i različite varijane fajla za isti font i primeniti kao na primeru iznad.

Podsetiću vas, ove elemente smo već pominjali u jedinici o tipografiji.

CSS3 selektori

Sami selektori u CSS-u napredovali su pojavom verzije 3. Ipak, i dalje ostaje problem Internet Explorera i njegove podrške, odnosno odsustva iste, za novije selektore.

Za više informacija o novim selektorima, preporučujem jedan, relativno star, ali aktuelan tekst: http://www.456bereastreet.com/archive/200601/css_3_selectors_explained/

CSS3 Media Queries

Još jedan značajan aspekt u CSS3 jeste pojava Media Queries funkcije. Pomoću pomenute funkcije možemo izvršiti upit vezano za uređaj koji naš sadržaj prikazuje i zatim prilagoditi prikaz.

CSS2 je doneo podršku za *media="screen"* tako da smo mogli distancirati CSS za ekrane, printere i sl, mada nije naišlo n značajniju primenu. Sada, sa pojavom CSS3 to je prošireno na rezolucije, među ostalim detaljima, što svakako postaje značajno.

U suštini, možemo testirati koja je veličina prikaza (browsera) i shodno tome primeniti drugačiji CSS.

Na primer, pogledajmo sledeći kod:

```
@media screen and (min-width: 960px) {  
    .element {  
        background:red;  
    }  
}  
  
@media screen and (min-width: 480px) and (max-width: 959px) {  
    .element {  
        background:blue;  
    }  
}  
  
@media screen and (max-width: 479px) {  
    .element {  
        background:green;  
    }  
}
```

Vidimo sada tri opisa za *.element* koji su veoma slični, ali daju drugačiju vrednost za background, tačnije drugačiju boju pozadine. Ono što je bitno u ovom primeru jeste da su okruženi određenim upitima koji proveravaju rezoluciju i time dobijamo različit prikaz. U ovom primeru je postavljen samo po jedan opis u svakom upitu, ali ih može biti mnogo više. Media Queries je veoma korisna funkcija kojom možemo razgraničiti stilizaciju za određene uređaje i možemo kreirati „poseban” prikaz za mobilne telefone, tablet računare i tako dalje. Kao dobre primere, pogledajte sajtove <http://www.smashingmagazine.com/> ili <http://bostonglobe.com/>, na primer. Ako budete menjali veličinu prozora browsera na računaru, videćete promene. Naravno, ukoliko pomenute sajtove otvorite pomoću mobilnog telefona (Android, iOS) videćete da se stilizacija razlikuje od desktop varijante.

Pitanje u jedinici

Ukoliko želimo da ugradimo nestandardni font u dokument, koristićemo:

- **@font-face**
- @font-choice
- @font-family
- @font-src

Najvažnije iz lekcije

- CSS3 je, slično poput HTML5, sledeći korak razvoja i nadolazeći standard.
- Već danas CSS3 možemo koristiti uporedo i zajedno sa CSS2 na našim sajtovima, ali ne prepoznaju svi browseri sve nove elemente CSS-a.