

Tema: Transparența elementelor HTML, filtre, transformări, gradiente

După această lecție:



- vei afla ce reprezintă transparența elementelor și cum pot fi create elemente HTML transparente;
- vei cunoaște cum poate fi controlat nivelul de opacitate al elementelor HTML;
- vei învăța cum să rotunjești colțurile elementelor HTML de formă dreptunghiulară și cum poți să le adaugi umbre, pentru a adăuga efecte vizuale 3D;
- vei afla ce reprezintă și vei învăța să definești transformări 2D asupra elementelor HTML;
- vei cunoaște ce sunt gradientele de culori și cum pot fi ele definite în CSS.

Am cunoscut deja suficiente proprietăți CSS, pe care le putem aplica pentru a prezenta frumos conținutul paginilor unui site. Azi vom mai învăța câteva, ce au apărut în CSS3. CSS3 este cel mai recent standard pentru stiluri și este complet compatibil cu versiunile anterioare ale CSS.

Crearea efectului de transparență

Efectul de transparență, sper, că deja l-ai observat, atunci când utilizăm culoare de fundal, specificată prin metoda RGBA, cu filtru alfa. Voi mai realiza un exemplu, ca să-ți amintești, dar voi suprapune culoarea transparentă a unui paragraf **P**, peste un bloc definit cu **DIV**, ce are o imagine pe fundal.

Deci, codul HTML este:



```
<div class="text">  
  <p>Deoarece cafeaua are un ... etc.</p>  
</div>
```

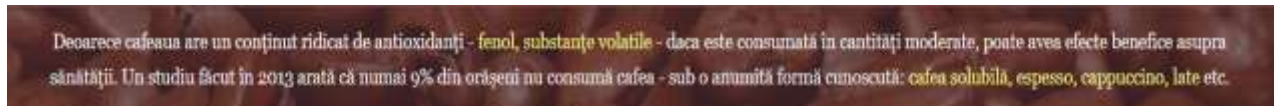
Stilurile pentru aceste două elemente:



```
.text{  
  width: 100%;  
  margin: 0 auto;  
  background-image: url(../images/background.jpg);  
  background-position: 50% 50%;  
  background-size: cover;  
}
```

```
.text p{
  padding: 15px;
  width:100%;
  font-size: 1.1em;
  color: whitesmoke;
  background-color: rgba(77, 46, 51, 0.7);
  margin:0;
}
```

Rezultatul:



Observă că elementul **P** are culoare de fundal transparentă și se suprapune peste blocul, ce are imagine de fundal. Boabele de cafea se întrevăd pe fundal.

Transparența imaginilor

Azi vom discuta despre transparența imaginilor, dar principiul aplicat pentru asigurarea transparenței imaginilor este asemănător cu asigurarea transparenței blocurilor de elemente, create cu **DIV**.

Pentru a face o imagine transparentă, se utilizează proprietatea **opacity**, care asigură transparența imaginii în browserele InternetExplorer9 (IE9), Firefox, Chrome, Opera și Safari. Această proprietate ia valori de la 0 la 1. O valoare mai mică, atribuită proprietății **opacity** face elementul mai transparent. IE8 și versiunile anterioare, folosește proprietatea **filter:alpha(opacity=x)**, unde **x** poate lua valori între 0 – 100. O valoare mai mică atribuită acestei proprietăți face elementul mai transparent. Noi vom specifica ambele versiuni ale proprietății, fiindcă nu cunoaștem ce browser va interpreta aceste coduri.

Să realizăm un exemplu: pentru un set de imagini din pagină, le voi seta opacitatea 0.7, iar atunci când voi trece cu cursorul deasupra lor – lei voi seta opacitatea 1.



```
.content img {
  max-width: 20%;
  height: auto;
  border: 3px dotted rgb(247, 238, 239);
  padding: 3px;
  margin: 5px;
  opacity: 0.8;
  filter: alpha(opacity=80);
}
.content img:hover{
```

```

    opacity: 1.0;
    filter: alpha(opacity=100);
}

```

Rezultat încărcare pagină în browser:



Și atunci când plasez cursorul pe una din poze – vezi cum se schimbă opacitatea celei de-a treia poză:



Rotunjirea colțurilor elementelor și adăugarea umbrelor

Aplicând noile proprietăți din CSS3, este posibilă rotunjirea colțurilor elementelor sau inserarea umbrelor pentru chenare. Aceste noi proprietăți sunt: **border-radius** și **box-shadow**.

Valorile proprietății **border-radius** vor fi numerice, specificate în **px** sau **%**, iar pentru proprietatea **box-shadow** se va specifica lungimea umbrei pentru coordonatele **x**, **y**, **z** și culoarea umbrei.

De exemplu, voi adăuga stilului pseudo-clasei **:hover** a imaginilor, scris anterior, două declarații suplimentare:



```

.content img:hover{
    opacity: 1.0;
    filter: alpha(opacity=100);
    border-radius: 70%;
    box-shadow: 5px 3px 3px rgba(182, 172, 172, 0.7);
}

```

Rezultatul – când voi plasa cursorul de-asupra unei imagini – aceasta își va schimba forma, fiind rotunjite colțurile și i se va adăuga umbră. Observă acest lucru:



Transformări 2D

Transformările în CSS permit să se efectueze: translări, rotiri, scalări, înclinări – asupra unui element HTML. Pentru crearea transformărilor 2D în CSS3 se folosește proprietatea **transform**, suplimentar fiind utilizate următoarele metode de transformare: **translate()**, **rotate()**, **scale()**, **skewX()**, **skewY()**, **matrix()**. Vom analiza unele din ele.

Metoda "translate()"

Translează elementul din poziția inițială, într-o nouă poziție, deplasată pe **x** și **y**. De exemplu, voi specifica ca atunci când voi trece cu cursorul de-asupra imaginii, aceasta să se deplaseze cu **10px** pe orizontală și cu **15px** pe verticală:



```
.content img:hover{  
  opacity: 1.0;  
  filter: alpha(opacity=100);  
  box-shadow: 5px 3px 3px rgba(182, 172, 172, 0.7);  
  transform: translate(10px,15px);  
}
```

Observă deplasarea din imagine. Poți încerca să realizezi și alte translări:



Voi anexa și filmulețul **translate.mp4**, ca să vezi efectul.

Metoda "rotate()"

Utilizarea acestei metode va roti elementul cu unghiul specificat. De exemplu, voi seta rotirea imaginii cu 15 grade, atunci când voi trece cu cursorul de-asupra ei:



```
.content img:hover{
  opacity: 1.0;
  filter: alpha(opacity=100);
  box-shadow: 5px 3px 3px rgba(182, 172, 172, 0.7);
  transform: rotate(15deg);
}
```

Rezultatul – este unul semănător unui joc:



Metoda "scale()"

Va crește sau va descrește dimensiunea (lățimea și înălțimea) elementului HTML specificat. De exemplu, voi seta ca imaginea să descrească ca dimensiuni, atunci când plasez cursorul de-asupra ei:



```
.content img:hover{
  opacity: 1.0;
  filter: alpha(opacity=100);
  box-shadow: 5px 3px 3px rgba(182, 172, 172, 0.7);
  transform: scale(0.7,0.7);
}
```



Dar încercă acum să specifici: **transform: scale(1.1,1.1);** ...și observă rezultatul – e ca și cum vei seta să crească imaginea cu 10%.

Metoda "skewx()"

Deformează un element pe orizontală (pe axa x). De exemplu, transformarea următoare va deforma cu 10 grade imaginea:



```
.content img:hover{  
  opacity: 1.0;  
  filter: alpha(opacity=100);  
  box-shadow: 5px 3px 3px rgba(182, 172, 172, 0.7);  
  transform: skewX(10deg);  
}
```

Rezultatul va fi:



În mod analogic se procedează și cu celelalte 2 metode de deformare: **SKEWY()** și **SKEW()** - metoda **skewY()** deformează obiectul pe verticală, iar metoda **skew()** deformează elementul și pe X și pe Y.

Proprietatea "filter"

Această proprietate aduce efecte vizuale plăcute ochiului, aplicată elementelor HTML – ca de exemplu imaginilor. Setarea acestei proprietăți unui element va conduce la aplicarea unui filtru, cu denumirea specificată. Valori posibile pentru această proprietate, **filter: none, blur(), brightness(), contrast(), drop-shadow(), grayscale(), hue-rotate(), invert(), opacity(), saturate(), sepia()** etc.

Trebuie să știi că nu toate browserele recunosc această proprietate și din acest motiv se recomandă pe lângă setarea filtrului cu proprietatea **filter**, aceasta să mai fie setată și cu adăugarea prefixului "**-webkit-**". Pentru Safari, de exemplu, vom scrie **-webkit-filter**.

Unei imagini îi pot fi aplicate și câteva efecte vizuale – acestea fiind separate prin spațiu.

Voi prelua o imagine mai deosebită de pe un site cu imagini. Imagini "frumoase" poți prelua gratuit de pe resursa:

<https://unsplash.com/?fbclid=IwAR0l4SozmalywNkt2nem5ndE-nz0-AvQeFVtX2eEkSf2wjaCJXoehzWVpKw>

Deci, am preluat o imagine, căreia-i voi aplica mai multe filtre:



```




```

Stilurile scrise:



```
.original {
  box-shadow: 0 10px 15px 0 rgba(0, 0, 0, 0.2), 0 12px 20px 5px rgba(0, 0, 0, 0.22);
}
img:hover {
  -webkit-filter: inherit;
  filter: inherit;
}
.grayscale {
  box-shadow: 0 10px 15px 0 rgba(0, 0, 0, 0.2), 0 12px 20px 5px rgba(0, 0, 0, 0.22);
  -webkit-filter: grayscale(100%);
  filter: grayscale(100%);
}
.blur {
  box-shadow: 0 10px 15px 0 rgba(0, 0, 0, 0.2), 0 12px 20px 5px rgba(0, 0, 0, 0.22);
  -webkit-filter: blur(3px); /* Safari */
  filter: blur(3px);
}
.sepia {
  box-shadow: 0 10px 15px 0 rgba(0, 0, 0, 0.2), 0 12px 20px 5px rgba(0, 0, 0, 0.22);
  -webkit-filter: sepia(70%);
  filter: sepia(70%);
}
```

Rezultatul aplicării acestor stiluri va fi:



Și adaug un filmuleț, pentru a vedea efectul când trec cu cursorul de-asupra acestor imagini – cum se schimbă filtrul – filmulețul **filter.mp4**.

Alte exemple de aplicare a filtrelor poți vedea accesând acest link:

https://www.w3schools.com/cssref/css3_pr_filter.asp

Gradiente

Gradientul – reprezintă trecerea unei culori într-o altă culoare sau mai multe. Folosind gradientul, pentru definirea culorii de fundal a unui element, se reduce timpul de încărcare a paginii în browser, comparativ cu cazul când sunt utilizate imagini. În CSS3 pot fi definite două tipuri de gradient: Linear Gradients – **gradient linear** (cu schimbarea culorilor: în jos, în sus, la stânga, la dreapta sau pe diagonală, și gradient circular - **Radial Gradients** (gradientul pornește din centru). Nu toate versiunile browsere-lor (mai ales cele vechi) suportă definirea gradientului. Din acest motiv se recomandă și definirea în mod tradițional a culorii de fundal.

Gradientul linear

Pentru a crea un gradient liniar, trebuie să fie definite cel puțin două culori: culoarea inițială și cea finală, la care se face trecerea.

Sintaxa:

background: linear-gradient(direcția, c1, c2, ...);

Valoarea implicită pentru direcție, la gradientul linear va fi „**Top to Bottom**”. Observă că direcția se specifică utilizând cuvintele englezești.

Să analizăm un exemplu – voi specifica drept culoare de fundal pentru **BODY**, o culoare cu gradient. Schimbarea culorilor se va face de la stânga la dreapta, și voi scrie la direcție „to right”:



```
body{
  font: 16px/30px Georgia, serif;
  color: white;
  /* Culoarea in cazul cand browserul nu recunoaste gradientul */
  background-color: rgba(77, 46, 51, 0.8);
  /* Sintaxa standard */
  background: linear-
gradient(to right, rgba(77, 46, 51, 1), rgba(190, 150, 157, 0.3));
  /* Safari 5.1 - 6.0 */
  background: -webkit-linear-
gradient(to right, rgba(77, 46, 51, 1), rgba(190, 150, 157, 0.3));
  /* Opera 11.1 - 12.0 */
  background: -o-linear-
gradient(to right, rgba(77, 46, 51, 1), rgba(190, 150, 157, 0.3));
```



```

/* Firefox 3.6 - 15 */
background: -moz-linear-
gradient(to right, rgba(77, 46, 51, 1), rgba(190, 150, 157, 0.3));
margin: 0;
padding: 0;
}

```

Observă cum am definit culoarea de fundal – am specificat gradientul utilizând toate prefixele, specifice diferitor browsere. De asemenea, m-am asigurat și am specificat și în mod obișnuit culoarea de fundal, în cazul în care utilizatorul va avea o versiune mai veche a browserului, care nu va putea interpreta gradientul.

Mai jos vezi rezultatul interpretării stilului pentru gradient și vezi cum are loc schimbarea culorii de fundal, de la stânga la dreapta.



În cazul în care se dorește schimbarea culorii pe diagonală, se va specifica direcția "to bottom right", adică "spre dreapta și în jos":

```

background: linear-gradient(to bottom right, rgba(77, 46, 51, 1), rgba(190, 150, 157, 0.3));
...și așa se va proceda și pentru restul prefixelor.

```

Sau în cazul în care se dorește trecerea prin mai multe culori, acestea vor fi specificate, utilizând drept separator virgula.

```

background: linear-
gradient(to right, rgba(77, 46, 51, 1), rgba(190, 150, 157, 0.3), rgba(77, 46, 51, 1));

```

Rezultatul va fi:



Observă că la mijlocul paginii culoarea este mai roz, iar în părți mai întunecată, deoarece în stânga a fost specificată culoarea închisă, apoi cea mai roz, iar spre dreapta iar cea întunecată.

Gradientul circular

Schimbarea culorilor se va realiza circular, din mijlocul paginii spre margini.



```
background-color: rgba(77, 46, 51, 0.8);
/* Sintaxa standard */
background: radial-gradient(rgba(190, 150, 157, 0.3), rgba(77, 46, 51, 1));
background: -webkit-radial-
gradient(rgba(190, 150, 157, 0.3)); /* Safari 5.1 - 6.0 */
background: -o-radial-
gradient(rgba(190, 150, 157, 0.3)); /* Opera 11.1 - 12.0 */
background: -moz-radial-
gradient(rgba(190, 150, 157, 0.3)); /* Firefox 3.6 - 15 */
```

Rezultatul:



Observă că cea mai deschisă culoare, specificată pe prima poziție, pornește din centrul paginii, iar la margini e culoarea închisă specificată.

Poți încerca și tu să specifice culori de fundal la diferite blocuri, utilizând gradient. Analizează rezultatele obținute.

Să repetăm ce am învățat în cadrul acestei lecții:



- elementelor HTML le putem adăuga transparență sau umbre, pentru a crea efecte vizuale mai deosebite;
- elementelor HTML de formă dreptunghiulară, le putem rotunji colțurile, cu proprietatea apărută în CSS3, *border-radius*;

- elementele HTML pot fi deplasate în pagină, mărite, micșorate, înclinate, rotite etc., utilizând proprietăți CSS de transformare;
- elementele HTML pot avea culoare de fundal definită prin intermediul gradientelor.