

Tema: Tranziții și animații în CSS

După această lecție:



- vei afla ce reprezintă tranzițiile și cum pot ele fi definite în CSS;
- vei cunoaște ce sunt animațiile și cum pot ele fi definite în CSS.

La lecția trecută am învățat să implementăm efecte vizuale mai deosebite cu CSS3. Vom continua cu studierea altor posibilități din CSS3 de stilizare a elementelor HTML...

Crearea efectelor de tranziție

Acest efect este foarte des utilizat, în ultimul timp. Ai observat data trecută când aplicam metodele de transformare cum deplasarea sau mărirea elementelor se realiza brusc, atunci când plasam cursorul deasupra elementului... Proprietatea **transition** poate fi utilizată pentru a atenua acest neajuns, realizând implementarea unui stil mai lent, în decursul a câteva secunde. Generarea efectului de tranziție necesită specificarea altor proprietăți CSS.

Sintaxa pentru forma prescurtată, a proprietății de tranziție este:

transition: *property duration timing-function delay*

Această forma prescurtată include valorile pentru următoarele, alte proprietăți CSS:

- *transition-property* – specifică proprietatea CSS pentru care se definește tranziția. Poate fi: **color, border, background, transform** etc.;
- *transition-duration* – se specifică durata tranziției în secunde. Dacă nu este specificată durata, efectul nu se va produce, fiindcă valoarea implicită pentru durată este de 0 secunde;
- *transition-timing-function* – se specifică forma curbei pentru efectul de tranziție. Valori posibile: *linear, ease, ease-in, ease-out, ease-in-out, step-start, step-end, steps(int,start|end), cubic-bezier(n,n,n,n)*.
- *transition-delay* – specifica durata pauzei de până la începerea tranziției.

Valorile obligatorii, ce trebuie specificate, sunt pentru *proprietate* și *durata tranziției*. Proprietatea **transition**, ca și cele descrise la lecția trecută, necesită specificarea diferită pentru diferite browsere, utilizând prefixele corespunzătoare.

Să realizăm un exemplu – pentru imaginile din blocul cu clasa **.content**, din documentul HTML, creat la lecțiile anterioare, voi adăuga proprietățile **transition**:



```
.content img {
  max-width: 20%;
  height: auto;
  border: 3px dotted rgb(247, 238, 239);
  padding: 3px;
  margin: 5px;
  opacity: 0.7;
  filter: alpha(opacity=70);
  -webkit-transition: transform 2s; /* Safari */
  transition: transform 2s;
}
.content img:hover{
  opacity: 1.0;
  filter: alpha(opacity=100);
  box-shadow: 5px 3px 3px rgba(182, 172, 172, 0.7);
  transform: rotate(20deg);
  -webkit-transition: transform 2s; /* Safari */
  transition: transform 2s;
}
```

Rezultatul îl poți vedea în filmulețul **transition.mp4**. Dacă vei fi atent, atunci vei observa cum se rotesc imaginile cu proprietatea **transition** și cum se rotesc ele – fără această proprietate. Ai observat că am adăugat această proprietate și imaginilor, adică selectorului **img**? Cum crezi de ce? Deoarece, dacă nu o adăugam, în momentul în care feream mouse-ul de pe imagine, ea revenea foarte rapid în poziția ei inițială. Eu însă am vrut ca și revenirea să se facă mai lent, în timp de 2 secunde.

Bine... Acum aș vrea ca și restul efectelor vizuale să se aplice lent, de exemplu lent să apară umbra și să se schimbe filtrele. Cum voi proceda? Voi specifica aplicarea tranziției pentru toate proprietățile, specificând la "denumirea proprietății", asupra căreia trebuie să se producă efectul de tranziție – valoarea **"all"**, ceea ce înseamnă că durata tranziției se va aplica tuturor proprietăților de stil specificate în selector:



```
-webkit-transition: all 2s; /* Safari */
transition: all 2s;
```

Rezultatul îl poți urmări în filmulețul **all_transition.mp4**. Observă cum se rotesc încet imaginile și cum apare și dispare lent umbra.

Încearcă și tu să realizezi diferite efecte cu proprietatea **transition**. Urmărește rezultatele.

Aș vrea să-ți mai arăt un efect de tranziție, aplicat asupra meniului. Pentru referințele din meniu, pe lângă proprietățile existente, voi mai adăuga proprietățile de tranziție a culorii, atunci când trec cu cursorul deasupra textului referințelor:



```
a:hover {
    color: rgba(58, 240, 231, 0.7);
    padding: 10px;
    margin: 5px;
    -webkit-transition: color 1s; /* Safari */
    transition: color 1s;
}
```

Rezultatul obținut – utilizatorul când va plasa cursorul deasupra opțiunii meniului – textul își va schimba lent culoarea:



Încearcă și tu să realizezi această tranziție!

Animații cu CSS

Animațiile elaborate cu CSS3 permit crearea acelor animații pentru elementele HTML, care până nu demult erau realizate doar cu utilizarea JavaScript sau Flash. **Animația** presupune schimbarea treptată, de la un stil la altul, a unuia sau a mai multe elemente HTML, prin realizarea cadrelor (sau slide-urilor). Pot fi schimbate și mai multe proprietăți CSS și nu doar o singură dată, dar chiar și de câteva ori.

Pentru a defini animația cu CSS3, trebuie să se specifice mai întâi câteva cadre cheie pentru plasarea animației. Cadre trebuie să fie minimum două. Regulile de trecere, de la un cadru la altul, pentru animație, se specifică în **@keyframes nume**.

Proprietățile specifice definirii animațiilor în CSS, le poți vedea în tabelul de mai jos:

Proprietatea	Descrierea
@keyframes	Specifică definirea cadrelor animației
animation	Forma prescurtată pentru definirea animației, a tuturor proprietăților, înafara de: <i>animation-play-state</i> , <i>animation-fill-mode</i>

animation-delay	Specifică o pauză până la începerea animației
animation-direction	Specifică dacă o animație ar trebui să se deplaseze în sens invers sau definește cicluri alternative
animation-duration	Specifică durata animației: câte secunde sau milisecunde durează animația
animation-fill-mode	Specifică stilul elementului atunci când animația s-a terminat sau când e în pauză
animation-iteration-count	Specifică de câte ori trebuie să se producă (să se repete) animația
animation-name	Specifică denumirea animației @keyframes
animation-play-state	Specifică dacă animația e activă sau întreruptă
animation-timing-function	Specifică viteza curburii animației

Exact la fel, ca în cazul altor proprietăți din CSS3, se recomandă folosirea a diferite prefixe ale proprietății pentru diferite browsere.

Sintaxa proprietății pentru definirea animației:

```
@keyframes denumireAnimatie {  

    keyframes-selector {css-styles;}  

}
```

unde

- *denumireAnimatie* – este obligatorie, fiindcă definește numele animației;
- *keyframes-selector* – este obligatoriu. Valori posibile: o valoare numerică, 0-100%, exprimată în procente – exprimă procentul animației; Sau pot fi utilizate următoarele cuvinte-rezervate: **from** (corespunde - 0%), **to** (corespunde - 100%). Pot fi definite două sau mai multe cadre pentru o animație (am menționat deja).
- *css-styles* - obligatorii. Definește una sau mai multe proprietăți de stil pentru un cadru al animației.

În continuare voi defini o animație simplă, pentru logo-ul, din meniul paginii, care va dura timp de 4 secunde și va schimba culoarea textului, iar pe final va adăuga umbră literelor.

Animația se va repeta continuu, atâta timp cât utilizatorul stă pe pagină. Analizează atent definirea animației:



```
@keyframes anima{
  0% {
    opacity: 0.5;
  }
  60% {
    color: rgb(252, 160, 56);
    opacity: 0.8;
  }
  100% {
    text-shadow: 1px 2px 2px rgb(139, 50, 223);
  }
}
.menu .logo{
  animation-duration: 4s;
  animation-iteration-count: infinite;
  animation-name: anima;
}
```

Rezultatul:



Voi mai realiza o animație cu imaginile din pagină – ele se vor deplasa continuu la stânga și la dreapta, cu micșorarea dimensiunilor imaginilor:



```
.content img {
  max-width: 20%;
  height: auto;
  border: 3px dotted rgb(247, 238, 239);
  padding: 3px;
  margin: 5px;
  opacity: 0.7;
  filter: alpha(opacity=70);
  -webkit-animation: run 8s infinite ease-in-out;
  animation: run 8s infinite ease-in-out;
}
@-webkit-keyframes run {
  0% { transform: translate(0%); }
  25% { transform: translate(-50%) scale(0.7, 0.7); }
  50% { transform: translate(0%) scale(1, 1); }
  75% { transform: translate(50%) scale(0.7, 0.7); }
}
@keyframes run {
  0% { transform: translate(0%); }
```

```
25% { transform: translate(-50%) scale(0.7, 0.7); }  
50% { transform: translate(0%) scale(1, 1); }  
75% { transform: translate(50%) scale(0.7, 0.7); }  
}
```

Rezultatul deplasării și schimbării dimensiunilor imaginilor, îl prezint mai jos:



Si...



Si...



Rezultatul aplicării acestei animații o poți vedea și în filmulețul **animation.mp4**. De asemenea, în acest filmuleț și animația cu logo-ul, tot este vizibilă.

O recomandare ar fi – să nu exagerați cu animațiile. Una pe pagină, cred ca este suficientă.

Alte exemple de creare a animațiilor, vezi la acest link:

http://www.w3schools.com/css/css3_animations.asp

De asemenea, ar fi bine să știi, că în Internet găsești mai multe librării, cu coduri gata scrise pentru animații. Una din librăriile utilizate pentru adăugarea de efecte de animație este librăria **animate.css**, bazată pe CSS3, care poate fi descărcată de pe site-ul - <https://daneden.github.io/animate.css/>. După ce descarci librăria, tastând butonul **Download Animate.css**, copii conținutul într-un fișier, pe care îl numești **animate.css**, pe care, apoi, îl salvezi în folder-ul **css**, din proiect. În continuare îl poți utiliza ca pe un simplu fișier **.css**. Tot pe acest site poți încerca un anumit efect de animație, ca să știi ce clasă poți adăuga elementului HTML, pentru a-l anima corespunzător doleanțelor.

Să repetăm cele învățate în cadrul acestei lecții:



- în CSS pot fi definite tranzițiile, care reprezintă niște treceri, de obicei mai lente, ale unor elemente HTML, dintr-o stare în alta. Acest efect de tranziție se aplică foarte des opțiunilor de meniu, imaginilor-referințe, referințelor textuale, butoanelor, atunci când utilizatorul trece cu cursorul deasupra lor (*hover*), pentru a-i atrage cumva atenția și de a-l provoca să facă click pe ele;
- animația reprezintă derularea secvențială ale unor slide-uri sau cadre. Animația poate avea o durată limitată de timp sau poate dura la infinit.

Închei aici lecția. Data viitoare vom învăța cum putem utiliza CSS pentru a crea conținuturi "sensibile" la ecranul ce prezintă aceste conținuturi...