

Meniuri flotante în JavaScript

Se poate întâmpla, în crearea unei interfețe web, să doriți ca o parte a acesteia să fie permanent în atenția utilizatorilor / vizitatorilor. Fie că este vorba despre logo-ul companiei sau despre un meniu de navigare, este util ca acest element să rămână în câmpul vizual al unei persoane, chiar dacă aceasta derulează fereastra (browser-ului în acest caz). Soluția pe care am ales-o eu face uz, evident, de JavaScript.

În interesul clarității expunerii și pentru a nu mă pierde în generalități, voi prezenta o implementare concretă, pe care am și folosit-o în crearea unui site web. Este vorba despre un meniu de navigare, care "urmărește" vizitatorul când acesta derulează pagina.. Și, după cum am amintit, nu este vorba despre folosirea frame-urilor ci a JavaScript. Căci, veți vedea, poziția nu este statică, ci este calculată pe măsura derulării paginii.

Înainte de toate, se ridică firească întrebare "ce voi construi eu aici?". Reformulez: "cum se numește dracovenia asta?". Cel mai des un asemenea element este numit sliding toolbar / menu.. Traducerea (și adaptarea) mea ar fi meniu flotant. Și cred că sintagma acoperă scopul și comportarea acestui element.

Nu mă pot abține să nu exprim și unul dintre cele mai mari regrete ale mele. Veți putea constata și dumneavoastră: realizarea efectivă a unui meniu flotant nu este extrem de complicată. Din păcate, din cauza incompatibilităților dintre browsere, asigurarea cadrului de lucru necesită cel puțin la fel de multă muncă... și mult mai multă testare. În fine, asta e. Ne adaptăm condițiilor de lucru.

Proiectarea

În urma proiectării am ajuns la următoarea configurație a ferestrei:

După cum se poate observa, meniul va fi afișat în partea stângă, iar conținutul în partea dreaptă. Prezența banner-ului este opțională, la fel cum opțională este și adăugarea unui banner la baza inferioară a paginii. Dacă va întrebați cumva de ce am prevăzut în mod expres prezența unui banner, răspunsul e simplu: el poate servi atât pentru identificarea mai clară a paginii curente (poate fi inclus numele paginii, secțiunii), cât și pentru adăugarea banner-elor publicitare (cerința devenită aproape standard în zilele noastre de Web-economizare).

Pentru ca pagina să fie afișată coerent pe diferite rezoluții, toate elementele vor fi centrate în cadrul ferestrei browserului. Dimensiunile vor fi astfel calculate încât pagina să nu necesite defilare orizontală nici măcar la rezoluția 640x480. Acestea fiind stabilite, să-i dam drumul.

Realizarea

Identificarea browserelor se face simplu, prin două linii de cod:

```
isIE = (document.all) ? 1 : 0;  
isNS = (document.layers) ? 1 : 0;
```

Este suficientă această identificare, căci nu se folosesc elemente ultra-specifice, care să se știe că ridică probleme în diferite versiuni ale browserelor. Asadar, versiunea efectivă nu ne interesează (cel puțin atât timp cât este peste 3, pentru Netscape și peste 4 pentru Internet Explorer).

Secvența următoare de cod este una extrem de interesantă și utilă (nu ar trebui să laud chiar eu codul, însă chiar cred că este o găselniță extraordinară). La ce folosește? Ei bine, la determinarea dimensiunilor ferestrei - atenție, utile (în care pot fi afișate elemente) a browser-ului. Secvența de cod pentru Netscape este:

```
awidth = window.innerWidth;  
aheight = window.innerHeight;
```

iar pentru Internet Explorer:

```
awidth = document.body.clientWidth;
```

```
aheight = document.body.clientHeight;
```

Determinarea trebuie facuta dupa ce documentul a fost încarcat, însa înainte de a afisa elemente pe ecran. Poate va întrebati cum poate fi realizat acest lucru - încarcarea documentului, dar neafisarea elementelor paginii. Facem în urmatorul fel: pozitionam elementele paginii, numite în continuare Titlu, Meniu si Continut în "containere" si le declaram drept hidden, folosind CSS (Cascading Style Sheets).

Observatie: Veti vedea în codul paginii valoarea 600 (pixeli) ce reprezinta latimea toatala a elementelor paginii. Mai elegant este sa declarati o constanta cu aceasta valoare, sa zicem mwidth, pe care sa o referiti ulterior (daca nu va mai aduceti aminte, prin `&{mwidth}`); - în codul HTML si pur si simplu mwidth în cadrul script-urilor).

Dupa crearea layer-elor (a nu se confunda cu elemntele specifice Netscape - se mai numesc layer-e si elementele `<div>`) ele trebuie afisate si pozitionate corect în pagina. Pentru aceasta am definit o functie ShowPage(), apelata în urma aceluiasi eveniment, onload. Principalul lucru pe care îl realizeaza functia este modificarea unor proprietati ale layer-elor, proprietati ce tin tot de CSS.

În acelasi context trebuie apelata functia care determina derularea ferestrei. Functia este numita ScrollHandler(). Ea are doua componente principale: modifica elementul top (distanța elementului fata de marginea de sus a ferestrei) si asigura reapelarea sa automata, folosind setTimeout (pentru mai multe detalii a se vedea e-mail REPORT nr.16, "JavaScript - trusa de instrumente"). Derularea paginii poate fi obtinuta prin proprietatea (obiectului window - vezi e-mail REPORT nr.17 - "JavaScript - obiectul muncii") pageYOffset, pentru Netscape, iar pentru IE prin proprietatea (obiectului document), scrollTop. Referitor la intervalul setat pentru reactivarea functiei, depinde de efectul pe care îl doriti - "urmarire" brusca sau una mai lina (valori mai mici pentru timeout). O valoare moderata este 150 (milisecunde). Nota: ds este o variabila globala ce mentine ultimul offset (deplasare) a paginii.

Cam aceasta este realizarea meniului. Un exemplu de pagina ce contine codul JavaScript si principalele elemente HTML amintite puteti vedea urmînd link-ul de mai jos. O singura remarca mai am: folosirea tag-ului `<spacer>` (specific Netscape) este necesara caci altfel Netscape "taie" ultimele linii ale layer-ului "Continut". Motivul exact nu îl stiu. Probabil ca nici programatorii de la Netscape nu îl cunosc prea bine. Oricum tag-ul este ignorat (nu produce nici un efect) de catre Internet Explorer.

Testarea si dezvoltari ulterioare

Testarea este extrem de importanta. Banuiesc ca stiti acest lucru. Incercati codul pe cît mai multe versiuni ale ambelor browsere. În caz contrar s-ar putea sa aveti surprize - vezi necesitatea folosirii `<spacer>`-ului.

Codul prezentat va asigura doar punctul de plecare. Desi poate fi folosit cu succes si în forma actuala, el este menit mai ales dezvoltarii unor aplicatii mai complexe. Sau poate fi adaptat pentru un logo derulant, cum se întîlneste des pe Web. Oricare ar fi utilitatea sa viitoare, sper ca îl veti fi gasit instructiv.