

**Universitatea Tehnică Cluj-Napoca**  
**Școala Academică Postuniversitară de Informatică Aplicată și Programare**

**DHTML**

**(Dynamic Hypertext Markup Language).**

**Aplicații grafice Java Script**

**Coordonator,**

**Șef lucr. dr. ing. Lorentz Jäntschi**

**Absolventă,**

**Ing. Luminița Cristina Pirău**

**Cluj-Napoca**

**2002**

## Cuprins

Prefață .....	3
Preface.....	5
<b>1. DHTML (Dynamic Hypertext Markup Language).....</b>	<b>7</b>
1.1. Introducere .....	7
1.2. Ghid introductiv .....	8
<b>2. Aplicații grafice Java Script.....</b>	<b>16</b>
2.1. Aplicația 1: Schimbarea dinamică a paginii web .....	16
2.2. Aplicația 2: Amplasarea unui element HTML în document .....	17
2.3. Aplicația 3: Amplasarea unui imagini în document .....	19
2.4. Aplicația 4: Descărcarea de fonturi .....	20
2.5. Aplicația 5: Legarea datelor.....	21
2.6. Aplicația 6: Transferul de date.....	24
2.7. Aplicația 7: Informații despre client .....	40
2.8. Aplicația 8: Instrumente grafice .....	41
2.9. Aplicația 9: Mișcarea planetelor .....	44
2.10. Aplicația 10: IFRAME și schimbarea conținutului documentului ..	51
<b>3. Concluzii .....</b>	<b>53</b>
<b>4. Lista aplicațiilor .....</b>	<b>54</b>
<b>5. Bibliografie .....</b>	<b>55</b>

## Prefață

Lucrarea de față **“DHTML (Dynamic Hypertext Markup Language); Aplicații grafice Java Script”** a fost concepută pentru a extinde tradiționalul document HTML static și a realiza unele pagini Web mai interactive și mai angajante.

Lucrarea se dorește a fi și un material didactic auxiliar, care să permită autorilor să manipuleze orice element de pagină în orice moment și să facă paginile interactive.

În DHTML s-a introdus Modelul obiectual al documentului (DOM) care programează direct accesul la componentele individuale ale paginilor web, de la elemente individuale la containere. Acest acces este combinat cu modelul pe evenimente (EM) și permite browser-ului să reacționeze la acțiunile utilizatorului, să execute script-uri în acesta, să afișeze noul conținut fără a fi nevoie de a descărca de pe server documente suplimentare.

Lucrarea se imparte în trei părți. În realizarea ei s-au utilizat resurse folosite de DHTML (browsere, meniuri, fișiere HTML, casete de dialog, butoane, casete de editare, etichete statice, funcții și variabile care tratează anumite evenimente și fac legatura între surse), în realizarea paginilor WEB se folosește limbajul DHTML.

Prima parte a lucrării este o prezentare succintă a paginilor DHTML cu câteva exemple, în a doua parte sunt prezentate aplicații referitoare la modul de utilizare a paginilor DHTML, inclusiv IFRAME și schimbarea conținutului documentului ca rezultat al execuției aplicației. Capitolul trei conține concluziile și posibilitățile de dezvoltare ale aplicațiilor.

Se prezintă și lista aplicațiilor, precum și a imaginilor dintr-o aplicație mai spectaculoasă.

Lucrarea se încheie cu prezentarea bibliografiei utilizate.

Datorită importanței sale, aplicația poate fi folosită cu succes de către toți utilizatorii care doresc să realizeze pagini WEB mai plăcute și mai interesante, verificând în acest mod și cunoștințele asimilate în cadrul programului de instruire-educație.

## Preface

The present paper "**DHTML (Dynamic Hypertext Markup Language) Java Script Graphical Applications**" was designed in order to extend the traditional static HTML document and to accomplish some more interactive and engaging web pages.

The paper sets out to be an auxiliary teaching material, which allows the authors to manipulate each page element at any time and to make the pages more interactive.

In DHTML the Document Object Model (DOM) was introduced which directly programs the access at individual components of web pages, from individual elements to containers. This access is combined with the Event Model (EM) and allows the browser to react on users actions, to run scripts within the browser, to display the new content the need of downloading additional documents from the server.

The paper is divided in three parts. In its making DHTML resources were used (browsers, HTML files, dialog boxes, buttons, editing boxes, static labels, functions and variables that handle certain events and make the link between sources) for web page editing using DHTML language.

The first part of the paper is a short presentation of DHTML pages with a few examples, in the second part applications associated with the way DHTML pages are used are presented, including IFRAME and modification of document content as a result of running the application. The third chapter contains the conclusions and development possibilities of the applications.

The list of applications, as well as images form a more spectacular application is presented.

The paper ends by presenting the bibliography used.

Due to its importance, the application can be used with great success by all users which want to make more pleasant and more interesting web pages, and also in this way by verifying the knowledge assimilated during the training education program.

## **1. DHTML (Dynamic Hypertext Markup Language)**

### **1.1. Introducere**

DHTML este un model orientat obiect care extinde tradiționalul document HTML static și permite constructorilor de site-uri web să creeze pagini Web mai interactive și mai angajante.

DHTML pune la dispoziție un instrument puternic de gestiune a controalelor care permite autorilor să manipuleze orice element de pagină în orice moment.

DHTML este de asemenea cea mai simplă cale de a face paginile interactive, utilizând funcții cu deschidere către tehnologiile bazate pe standardul HTML.

Numeroase îmbunătățiri și funcții noi au fost adăugate limbajului HTML standard (fie HTML 4.0 ca referință) de către firma Microsoft, prin implementarea librăriilor dinamice în pachetele integrate Microsoft Internet Explorer 4.0, 4.5, 5.0, 5.5 și 6.0. Astfel, anumite facilități din cele ce vor fi discutate pe parcursul acestei lucrări sunt accesabile numai din versiunile de *nuovo* ale lui Internet Explorer.

Lucrarea este organizată pe structura unei prezentări a limbajului HTML, insistându-se numai asupra componentelor care fac ca pagina sau paginile să primească atributul de active. Exemplele prezentate se încadrează în aceeași categorie a aplicațiilor active (interactive).

## 1.2. Ghid introductiv

Modelul obiectual al documentului (DOM) introdus de DHTML permite autorilor de a programa direct accesul la componentele individuale ale paginilor web, de la elemente individuale la containere. Acest acces, combinat cu modelul pe evenimente (EM) permite browser-ului să reacționeze la acțiunile utilizatorului, să execute script-uri în acesta, să afișeze noul conținut fără a fi nevoie de a descărca de pe server documente suplimentare.

DOM execută o interactivitate avansată într-un mod simplu de realizat de către un autor cu pregătire medie în domeniul HTML.

Ceea ce diferă în modelul obiectual este că acum fiecare element HTML este programabil. Aceasta înseamnă că fiecare element HTML de pe pagină poate avea un script în spatele său care poate fi folosit să interacționeze cu acțiunile utilizatorului și să schimbe conținutul paginii dinamic. Modelul pe evenimente permite ca documentul să reacționeze când utilizatorul execută o acțiune asupra paginii, cum ar fi deplasarea mouse-ului pe un element anume, apăsarea unei taste sau completarea unui câmp dintr-un formular. Fiecare eveniment poate fi legat de un script care să transmită browser-ului o acțiune de modificare a conținutului din acesta, fără a interacționa cu serverul pentru a descărca un nou fișier.

Avantajul care îl prezintă această tehnică de lucru este pus la dispoziția autorilor un instrument care este capabil să creeze site-uri Web cu mai puține



pagini astfel încât utilizatorul să nu aștepte ca browser-ul să descarce noile pagini de pe serverul Web, mărinđ astfel viteza de parcurgere și performanța în Internet, ca o consecință a acestui fapt.

Fiecare element HTML este un obiect scriptabil în modelul obiectual, cu propriul său set de proprietăți, metode și evenimente. Oricum, pentru a realiza un script pentru fiecare obiect, autorul trebuie să știe cum să acceseze acel element.

Modelul obiect este orientat către colecțiile de elemente, o ierarhie care grupează elementele unele în altele. Cele mai importante colecții sunt aici colecția *all* și colecția *children*. Un document DHTML este o structură ordonată ierarhic de elemente.

(1). Fie următorul [exemplu](#) clasic de document HTML:

```
<HTML>
  <BODY>
    <DIV>
      <P>Some text in a
        <B>paragraph</B>
      </P>
      <IMG id=image1 src="mygif.gif">
    </DIV>
    <IMG id=image2 src="mygif.gif">
  </BODY>
</HTML>
```

În acest exemplu obiectul <DIV> conține (și este părintele) obiectelor <P> și <IMG id=image1>. Invers, <IMG id=image1> și <P> sunt fii lui <DIV>. Obiectul <IMG id=image2> este fiul obiectului <BODY>.

Fiecare obiect are o colecție *all* care conține toate elementele care îi urmează în ierarhie și o colecție *children* care conține numai elementele care sunt descendenți direcți ai acestui element. În exemplul anterior, <B> apare în colecția *all* a lui <DIV> dar nu apare în colecția *children* a acestuia. Similar, <DIV> este un membru al colecției *children* a lui <BODY>, în timp ce <P> nu este. În adădire la aceste colecții pentru fiecare element documentul însuși (reprezentat de obiectul document) are un număr de colecții de elemente și nonelemente. Cea mai importantă colecție a acestuia este colecția *all* a acestuia (*document.all*) care conține toate elementele paginii web. Această colecție este calea primară de acces la elemente prin intermediul script-ului.

Apăsând un buton, mutând pointerul mouse-ului peste o parte a unei pagini web, selectând text de pe pagină, aceste acțiuni toate produc evenimente și autorul DHTML poate scrie cod care să răspundă la aceste evenimente. *Event handler* este acea secvență de cod care tratează evenimentul.

Tratarea evenimentelor nu este un atribut al Microsoft Internet Explorer 4.0 sau superior. Browsere ca Microsoft Internet Explorer 3.0, Netscape Navigator 3.x sau Communicator au de asemenea librării dinamice incluse de tratare a evenimentelor.

Oricum, începând cu Microsoft Internet Explorer 4.0 elementele HTML ale paginii Web care sunt surse de evenimente și tipurile de evenimente care sunt generate sunt mult mai numeroase.

Anterior, doar un set mic de elemente HTML puteau genera evenimente cum ar fi *ANCHOR*, *IMAGE MAP*, *FORM ELEMENTS* (*BUTTON*, *INPUT*, *TEXTAREA*, ...) *APPLICATION* și *OBJECT*.

Cu Microsoft Internet Explorer 4.0 și modelul său de evenimente orice element HTML din pagină poate fi sursă pentru toate evenimentele create din mouse sau din tastatură.

Următorul set este setul comun de evenimente care fiecare element HTML îl poate genera:

- Evenimente **generate de mouse** când utilizatorul:
  - *OnMouseOver* – mută pointerul mouse-ului peste (aceasta este intrarea) un element;
  - *OnMouseOut* – mută pointerul mouse-ului în afara (aceasta este ieșirea) unui element;
  - *OnMouseDown* – apasă oricare dintre butoanele mouse-ului pe element;
  - *OnMouseUp* – eliberează oricare dintre butoanele mouse-ului pe element;

- *OnMouseMove* – mută pointerul mouse-ului în interiorul unui element;
  - *OnClick* – apasă butonul stânga al mouse-ului pe un element;
  - *OnDbClick* – apasă dublu click stânga pe un element;
- Evenimente **generate de tastatură**:
    - *OnKeyPress* – apasă și eliberează o tastă (ciclu complet de apăsare); dacă tasta este ținută apăsată se generează un multiplu de evenimente *OnKeyPressed*;
    - *OnKeyDown* – apasă o tastă; dacă e ținută tasta în continuare, un singur eveniment este generat;
    - *OnKeyUp* – eliberează o tastă;

Pentru a ajuta autorii să creeze un cod compact, simplu, și ușor de întreținut, Microsoft Internet Explorer 4.0 introduce conceptul de *event bubbling* ca o nouă modalitate de tratare a evenimentelor. Motivul este că Microsoft Windows, OS/2, OSF Motif și aproape orice alt sistem bazat pe interfața grafică cu utilizatorul (GUI – graphical user interface) utilizează această tehnică pentru a procesa evenimentele în propriile lor interfețe cu utilizatorul.

Oricum, *event bubbling* este nou pentru HTML și pune la dispoziție un model eficient pentru încorporarea tratării pe evenimente în documentele Web. În browserele mai vechi, dacă un element HTML genera un eveniment dar nici o

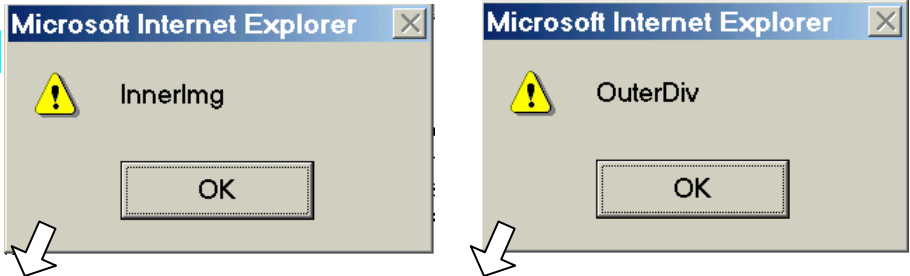
rutină de tratare a evenimentului nu a fost scrisă pentru acesta, evenimentul dispărea, fără a mai putea fi văzut altă dată. Event *bubbling* transferă însă aceste evenimente netratate către elementul părinte pentru a fi tratate de acesta. Evenimentul continuă să urce în ierarhie („*bubbles up*”) până când găsește rutina de tratare sau atinge cel mai superior obiect în ierarhie, obiectul document.


*Event bubbling* este util deoarece:

- Permite ca acțiunile comune să fie tratate centralizat (de o singură rutină de tratare a evenimentului);
- Reduce cantitatea de cod din întreaga pagină Web;
- Reduce numărul de schimbări în cod care trebuiesc efectuate la modificarea (actualizarea) documentului;

(2). Următorul [exemplu](#) este unul de acest tip, în care tratarea se face în ierarhie:

```
<HTML>
<BODY>
<DIV id=OuterDiv style="background-color: red"
onmouseover="alert(window.event.srcElement.id);">
This is some text
<IMG id=InnerImg
src ="...">
</DIV>
</BODY>
</HTML>
```



This is some text 

(3). Un alt [exemplu](#) simplu este atașarea unui eveniment *RollOver* cu instrumentele specifice MS\_IE\_4:

```

<HTML>
<HEAD>
<STYLE>
.Item {
  cursor: hand;
  font-family: verdana;
  font-size: 20;
  font-style: normal;
  background-color: blue;
  color: white
}
.Highlight {
  cursor: hand;
  font-family: verdana;
  font-size: 20;
  font-style: italic;
  background-color: white;
  color: blue
}
</STYLE>
</HEAD>
<BODY>
<SPAN class=Item>Milk</SPAN>
<SPAN class=Item>Cookies</SPAN>
<SPAN class=Item>Eggs</SPAN>
<SPAN class=Item>Ham</SPAN>
<SPAN class=Item>Cheese</SPAN>
<SPAN class=Item>Pasta</SPAN>
<SPAN class=Item>Chicken</SPAN>
<SCRIPT>
function rollon() {
  if (window.event.srcElement.className == "Item") {
    window.event.srcElement.className = "Highlight";
  }
}
document.onmouseover = rollon;

```



Milk Cookies Eggs Ham Cheese Pasta Chicken



*Milk* Cookies Eggs Ham Cheese Pasta Chicken



Milk *Cookies* Eggs Ham Cheese Pasta Chicken



Milk Cookies *Eggs* Ham Cheese Pasta Chicken



Milk Cookies Eggs *Ham* Cheese Pasta Chicken



Milk Cookies Eggs Ham *Cheese* Pasta Chicken



Milk Cookies Eggs Ham Cheese *Pasta* Chicken



Milk Cookies Eggs Ham Cheese Pasta *Chicken*

```
function rolloff() {  
  if (window.event.srcElement.className == "Highlight") {  
    window.event.srcElement.className = "Item";  
  }  
}  
document.onmouseout = rolloff;  
</SCRIPT>  
</BODY>  
</HTML>
```

## 2. Aplicații

2.1. [Aplicatia 1](#). Schimbarea dinamică a corpului paginii web cu ajutorul funcției *insertAdjacentHTML*.

```
<HTML>
<HEAD><TITLE>Salut!</TITLE>
<SCRIPT LANGUAGE="JScript">
function changeMe() {
    MyHeading.outerHTML = "<H1 ID=MyHeading>Dynamic HTML!</H1>";
    MyHeading.style.color = "green";
    MyText.innerText = "Se pot face lucruri interesante cu efort mic";
    MyText.align = "center";
    document.body.insertAdjacentHTML(
        "BeforeEnd", "<P ALIGN=\"center\">Doar incearca!</P>"
    );
}
</SCRIPT>
<BODY onclick="changeMe()">
<H3 ID=MyHeading>Bine ai venit in lumea lui <I>Dynamic HTML!</I></H3>
<P ID=MyText>Apasa click oriunde pe acesta pagina.</P>
</BODY>
</HTML>
```

Rezultatul încărcării paginii în Microsoft Internet Explorer este:



**Bine ai venit in lumea lui *Dynamic HTML!***

Apasa click oriunde pe acesta pagina.

iar la apăsarea unui click:



## Dynamic HTML!

Se pot face lucruri interesante cu efort mic

Doar incerca!

Apăsarea repetată a click-ului duce la:

## Dynamic HTML!

Se pot face lucruri interesante cu efort mic

Doar incerca!

Doar incerca!

Doar incerca!

Doar incerca!

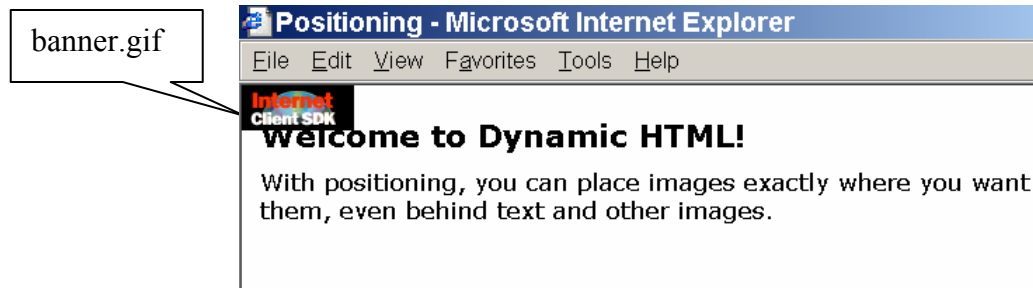
### 2.2. [Aplicatia 2](#). Amplasarea unui element HTML în document

Poziționarea este abilitatea de a plasa un element HTML la o coordonată anume în document prin asignarea coordonatelor x și y și a planului z al acelui element. Aceasta înseamnă că se pot plasa elemente (imagini, controale și text) exact acolo unde dorim să fie și în plus li se pot atribui efecte de suprapunere prin definirea ordinii de așezare unul peste altul a elementelor de la aceeași coordonată. Poziționarea este o extensie a CSS (*cascading style sheets*). Aceasta înseamnă că se

poate defini ordinea unui element prin setarea atributului său CSS corespunzător.

Următorul exemplu arată cum se setează poziția absolută a unei imagini.

```
<HTML>
<HEAD><TITLE>Positioning</TITLE>
<BODY>
<H3>Welcome to Dynamic HTML!</H3>
<P>With positioning, you can place images exactly where you want them, even
behind text and other images.
<IMG STYLE="position:absolute;top:0; left:0; z-index:-1" SRC="banner.gif">
</BODY>
</HTML>
```

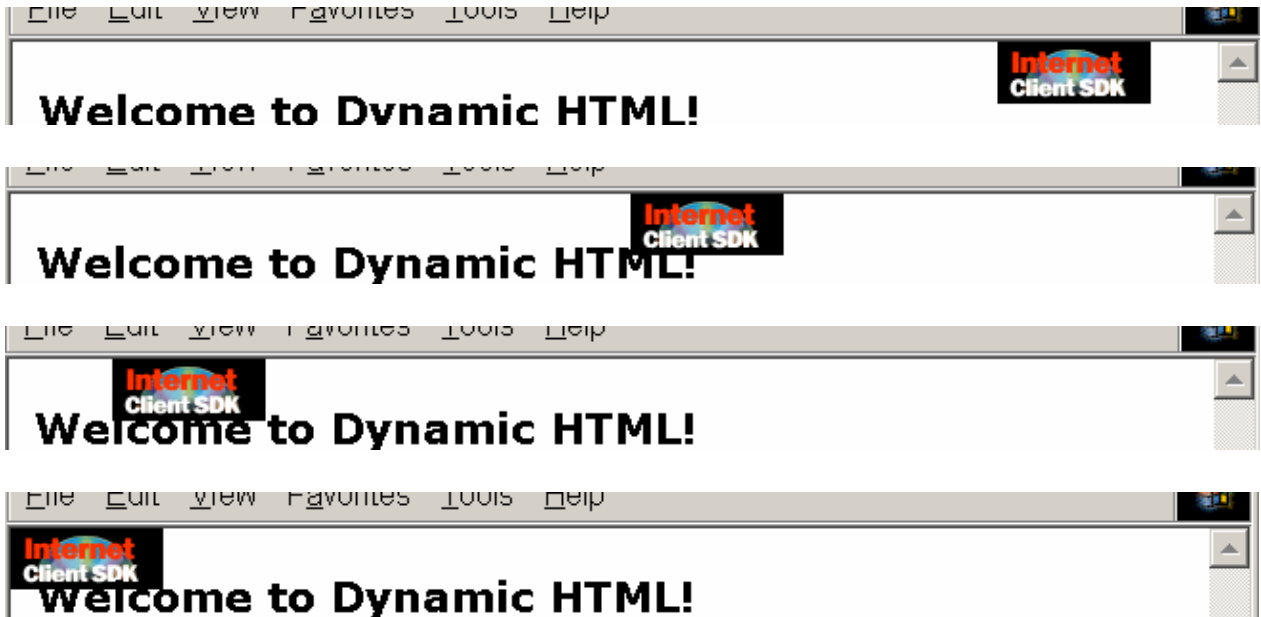


În exemplul precedent imaginea este plasată în colțul dreapta sus al documentului. Setând atributul *z-index* la -1 face ca imaginea să fie plasată în spatele textului de pe pagină.

IE (Internet Explorer) 4.0 suportă poziționarea și animarea elementelor chiar și după ce documentul a fost încărcat. Deoarece DOM (*document object model*) furnizează accesul la stiluri, se poate seta și schimba poziția unui element tot atât de simplu cum i se poate schimba culoarea. Aceasta face ca să fie simplu să se schimbe poziția elementelor de exemplu în funcție de modul în care utilizatorul vizualizează documentul. Pentru animație, tot ce trebuie făcut este schimbarea lentă a coordonatelor unui element la un anumit interval de timp.

### 2.3. [Aplicatia 3](#). Amplasarea unei imagini în document

Următorul exemplu arată cum o imagine defilează la marginea superioară a documentului.



Codul sursă al acestei aplicații este prezentat în continuare:

```
<HTML>
<HEAD><TITLE>Dynamic Positioning</TITLE>
<SCRIPT LANGUAGE="JScript">
var id;
function StartGlide()
{
    Banner.style.pixelLeft =
    document.body.offsetWidth;
    Banner.style.visibility = "visible";
    id = window.setInterval("Glide()",50);
}
function Glide()
{
    Banner.style.pixelLeft -= 10;
    if (Banner.style.pixelLeft<=0) {
        Banner.style.pixelLeft=0;
```

```
        window.clearInterval(id);
    }
}
</SCRIPT>
<BODY onload="StartGlide()">
<H3>Welcome to Dynamic HTML!</H3>
<P>With dynamic positioning, you can move images anywhere in the document
even while the user views the document.
<IMG ID="Banner"
STYLE="visibility:hidden;position:absolute;top:0; left:0; z-index:-1"
SRC="banner.gif">
</BODY>
</HTML>
```

și reprezintă o modificare a aplicației 2, la care s-au adăugat și funcțiile de mișcare în document *StartGlide()* și *Glide()*.

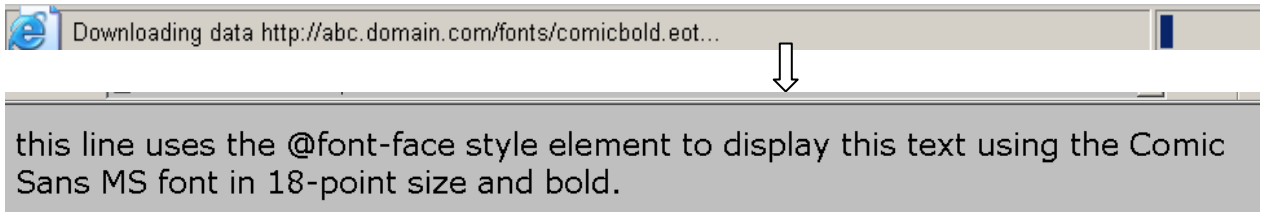
În exemplul precedent funcția *StartGlide()* este apelată când documentul este încărcat. Funcția setează poziția absolută a banerului în colțul dreapta sus al corpului documentului, afișează imaginea și inițializează intervalul de timp. Funcția *Glide()* este apelată repetat după un interval de timp de 50 de milisecunde. Funcția mută imaginea către stânga cu 10 pixeli și când în final imaginea este în colțul stâng, anulează intervalul de întrerupere.

Poziționarea dinamică are mulți beneficiari în aplicații comerciale și de afaceri. Prin combinarea stilurilor dinamice, poziționării, transparenței la imagini și la controale *ActiveX* se pot prezenta un bogat set de efecte de animație în documente.

#### 2.4. [Aplicatia 4](#). Descărcarea de fonturi

IE 4.0 suportă utilizarea fonturilor descărcate dinamic. Utilizând atributul de stil *@font-face*, un document poate referi un font care este automat descărcat, este utilizat pentru pagină și este dealocat imediat de pagina nu mai este afișată. Următorul exemplu arată cum se folosește mecanismul de descărcare a fonturilor.

```
<HTML><HEAD>
<STYLE>@font-face
{font-family:comic; src:url(http://abc.domain.com/fonts/comicbold.eot);}
</STYLE>
</HEAD>
<BODY>
<P STYLE="font-family:comic;font-size:18pt">this line uses the @font-face
style element to display this text using the Comic Sans MS font in 18-point
size and bold.
<P>
</BODY></HTML>
```



## 2.5. [Aplicatia 5](#). Legarea datelor

Legarea datelor este o facilitate DHTML care permite legarea elementelor individuale în document de datele dintr-o altă sursă, de exemplu o bază de date sau un fișier text delimitat prin virgule. Când documentul este încărcat, datele sunt automat preluate de la sursă și formate pentru a fi afișate în interiorul elementului. O cale practică de a folosi legarea datelor este generarea automată și

dinamică a tabelelor în document. Se poate face acest lucru prin legarea unui tabel de o sursă de date. Când documentul este afișat, o nouă linie este creată în tabel pentru fiecare înregistrare primită de la sursă iar celulele fiecărei linii sunt afișate cu conținutul fiecărui câmp al înregistrării primit de la sursă. Deoarece generarea se face dinamic, utilizatorul poate vizualiza documentul și în timpul cât liniile sunt create în tabel. Mai mult, după ce toate datele din tabel sunt descărcate, se poate manipula (sorta sau filtra) datele fără a fi necesară interogarea ulterioară a serverului. Tabelul este generat utilizând datele primite anterior.

O altă utilizare practică a legării datelor este de a lega unul sau mai multe elemente din document la anumite câmpuri ale unei înregistrări date. Când documentul este descărcat, elementele sunt completate cu text și date din câmpurile înregistrării, uneori numită *înregistrarea curentă*. Un exemplu simplu este atunci când un formular de scrisoare este completat cu numele, adresa de email și alte detalii despre o persoană dintr-o bază de date. Pentru a adapta scrisoarea pentru o persoană dată, este suficient să specificăm apoi care înregistrare ar trebui să fie înregistrarea curentă. Nici o altă schimbare nu mai este necesară.

Altă aplicație practică este legarea câmpurilor dintr-un formular de câmpurile dintr-o înregistrare. Nu numai că utilizatorul poate vizualiza conținutul înregistrării, dar poate și schimba conținutul acesteia prin schimbarea conținutului din formular. Utilizatorul poate apoi transmite (*submit*) aceste schimbări așa încât noile date sunt

încărcate în sursă (de exemplu către serverul HTTP sau baza de date). Pentru a putea avea date legate dinamic în document, trebuie să încărcăm mecanismul DSO (*Data Source Object*) în document. Acesta este invizibil și este un simplu control ActiveX sau applet Java care știe să comunice cu sursa de date. Microsoft pune la dispoziție două astfel de mecanisme de manipulare a datelor în IE 4.0: una pentru a accesa datele separate prin virgulă în fișiere text și alta de a accesa datele structurate pe sintaxa SQL (*Structured Query Language*) pe un server SQL sau pe o altă sursă ODBC (*Open Database Connectivity*). Și alte DSO-uri sunt disponibile pe site-ul Microsoft sau al altor *provide-ri*.

Următorul exemplu arată cum se pot lega datele de o sursă de date. Când este afișat, acest exemplu afișează primele 3 câmpuri dintr-un fișier text delimitat prin virgulă numit "*sampdata.csv*" într-un tabel.

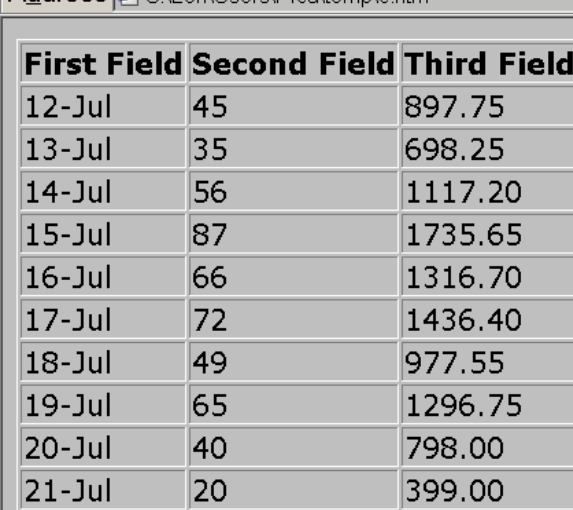
```
<HTML>
<BODY>
<OBJECT
CLASSID="clsid:333C7BC4-460F-11D0-BC04-0080C7055A83" ID=sampdata>
  <PARAM NAME="DataURL" VALUE="sampdata.csv">
  <PARAM NAME="UseHeader" VALUE="True">
</OBJECT>
<TABLE BORDER=1 DATASRC="#sampdata">
<THEAD>
<TR><TH>First Field<TH>Second Field<TH>Third Field
<TBODY>
<TR><TD>
<SPAN DATAFLD=A></SPAN>
<TD><SPAN DATAFLD=B></SPAN>
<TD><SPAN DATAFLD=C></SPAN>
</TABLE>
```

```
</BODY>  
</HTML>
```

Conținutul fișierului *sampdata.csv*:

```
A:STRING,B:STRING,C:STRING  
12-Jul,45,897.75  
13-Jul,35,698.25  
14-Jul,56,1117.20  
15-Jul,87,1735.65  
16-Jul,66,1316.70  
17-Jul,72,1436.40  
18-Jul,49,977.55  
19-Jul,65,1296.75  
20-Jul,40,798.00  
21-Jul,20,399.00
```

Efectul în browser al încărcării documentului:



First Field	Second Field	Third Field
12-Jul	45	897.75
13-Jul	35	698.25
14-Jul	56	1117.20
15-Jul	87	1735.65
16-Jul	66	1316.70
17-Jul	72	1436.40
18-Jul	49	977.55
19-Jul	65	1296.75
20-Jul	40	798.00
21-Jul	20	399.00

## 2.6. [Aplicatia 6](#). Transferul de date



```
<HEAD>
<SCRIPT>
/* -----
   fnSetInfo sets the data format in the first parameter and provides the text to
   drop in second. The second line copies text.
   -----*/
function fnSetInfo() {
  event.dataTransfer.setData("Text", "Microsoft Golf 1998");
  event.dataTransfer.effectAllowed = "copy";
}
/* -----
   fnGetInfo is called by the target object in the ondrop event.
   It cancels the default action and sets the cursor to the system copy icon.
   Then it specifies the data format to retrieve. Last, it sets the value property of
   oTarget object to the information from getData.
   -----*/
function fnGetInfo() {
  event.returnValue = false;
  event.dataTransfer.dropEffect = "copy";
  oTarget.value = event.dataTransfer.getData("Text");
}
/* -----
   fnCancelDefault Cancels the default action in ondragenter and ondragover
   so that the copy cursor is displayed until the selection is dropped.
   -----*/
function fnCancelDefault() {
  event.returnValue = false;
  event.dataTransfer.dropEffect = "copy";
}
</SCRIPT>
</HEAD>
<BODY>
<IMG ID=oSource SRC="Golf98.gif" ondragstart="fnSetInfo()">
<P>Drag the image and drop it onto the text box below. Text data will be pasted
into the text box.</P>
<INPUT ID=oTarget VALUE="[drop text here]"
  ondragenter="fnCancelDefault()"
  ondrop="fnGetInfo()"
  ondragover="fnCancelDefault()">
</BODY>
```

Codul din acest exemplu stabilește proprietatea *effectAllowed* (efectPermis) de a copia și proprietatea *dropEffect* (aplicaEfect) pentru a afișa cursorul de copiere. Acțiunea implicită trebuie anulată în toate cazurile care sunt abordate – în acest exemplu, *ondragstart* (la tragere început), *ondragover* (la tragere peste), *ondragenter* (la tragere intrare) și *ondrop* (la lasare).

În ceea ce privește Microsoft® Internet Explorer 5, este permis accesul la o implementare extensivă a transferului de date. Pentru Internet Explorer, transferul de date implică scrierea informației în memoria clipboard și citirea ei printr-un obiect de transfer de date. Informația transferată include atât formatul datelor cât și datele în sine.

Transferul de date implică două procese:

Primul proces este editarea folosind *cut/copy/paste* (tăiere/copiere/lipire) din meniul *Edit* al browserului și meniurile de shortcut. Această capacitate este importantă pentru facilitarea sarcinilor repetate de editare, unde informația păstrată în memoria sistemului asigură ușurința în folosire. Un exemplu al acestei capacități este posibilitatea de a copia aceleași date de mai multe ori în locații diferite.

Al doilea proces este operația de *drag and drop* (trage și lasă). *Drag and drop* este un proces imediat și intuitiv deoarece utilizatorul pur și simplu selectează un element HTML și îl mută într-o nouă locație. Această capacitate este potrivită cel mai bine operațiilor singulare, deoarece odată ce operația de *drag* (tragere) se termină, orice urmă a datelor care sunt transferate dispare. Luate împreună, aceste capacități de editare permite să se implementeze funcții tipice utilizator - interfață pe care utilizatorii le așteaptă de la un software bun.

Implementarea transferului de date în Internet Explorer 5 asigură un control programatic asupra formatelor de date care sunt transferate de la sursă la țintă. Acest control se extinde la transferurile de date din cadrul ferestrei browserului, de la o fereastră a browserului la alta, de la *browser* la *desktop* și de la browser la un alt *Component Object Model* (COM) (model al obiectul component) sau aplicație *drag-and-drop*, cum ar fi Microsoft Word. În prezent, din motive de securitate a datelor, Internet Explorer nu permite nici o țintă a aplicației *drop* în browser din accesarea datelor care provin din alt domeniu de securitate sau din altă aplicație de desktop.

Sursa de date stipulează tipul transferului de date: *move* (mutare), *copy* (copiere) sau *link* (legare). De asemenea, ea stipulează și tipul informației – o cale sursă de imagine, un șir de text sau un URL (*Universal Resource Locator*) pentru o legătură – care va fi transferată. De exemplu, aplicațiile comerciale tip *shopping-*

*cart* folosesc această capabilitate. Acest tip de aplicații afișează un catalog și un formular de comandă. Ele permit utilizatorilor să “tragă” (*drag*) imaginea articolului pe care doresc să-l cumpere și să o “lase” (*drop*) în formularul de comandă. În acest moment, textul asociat cu imaginea, mai degrabă decât imaginea în sine, este copiat în formular. Printr-o singură operație sunt afișate prețul, produsul, marca și numărul de serie. Bineînțeles, imaginile nu sunt singurele surse de date – ancorele, textele și controalele de text funcționează de asemenea ca surse de date.

Obiectele țintă pot arăta exact ca obiectele sursă. Totuși, rolul lor este foarte diferit față de cel al sursei de date. Obiectele țintă furnizează feedback-ul privind operația de transfer a datelor. Feedback-ul vine sub două forme: informații pentru utilizator și notificările programatice. Se poate seta sistemul astfel încât cursorul să afișeze icoanele de sistem pentru *copy*, *move* sau *link*. Aceste icoane furnizează informații către utilizator în timpul unei operații de *drag-and-drop*. Evenimentele de notificare, cum ar fi *onbeforepaste* și *ondrop*, asigură informații care se pot folosi pentru a înscena o comportare personalizată. În cazul aplicației *shopping-cart*, această comportare personalizată în primul rând recuperează informațiile despre produsul asociat imaginii și apoi le introduce în formularul de comandă.

Internet Explorer 4.0 are o funcționalitate limitată de transfer a datelor. El suportă comportarea implicită *drag-and-drop* pentru imagini, ancore și controale de

text. Controlul asupra funcționalității de transfer a date în Internet Explorer 4.0 este limitat la activarea sau dezactivarea acesteia. Internet Explorer 5 a îmbunătățit și a sporit această funcționalitate.

Se poate folosi funcționalitatea transferului de date subliniată în această secțiune pentru a coda comportări personalizate de *drag-and-drop* și editare.

Internet Explorer 5 suportă două obiecte pentru transferarea datelor, obiectele *dataTransfer* (`dataTransfer`) și *clipboardData* (`clipboardData`). Disponibile prin obiectul *event* (eveniment) și respectiv *window* (fereastră), fiecare conține un *portal* (gateway) prin care informația este așezată pe obiectele sursă și recuperată pe obiectele țintă.

Obiectul *dataTransfer* oferă control programatic asupra operațiilor de *drag-and-drop*. El transferă date în propria memorie clipboard, unde aceasta rămâne doar până la apariția evenimentului *ondragend*. Odată cu declanșarea acestui eveniment, datele sunt șterse din memorie, indiferent dacă transferul de date a reușit sau nu.

Obiectul *dataTransfer* suportă două proprietăți specifice *drag-and-drop* precum și unele metode de transfer a datelor. Proprietățile *effectAllowed* și *dropEffect* furnizează feedback vizual în timpul operației drag și sunt vitale pentru atingerea consistenței cu interfața Microsoft Windows. Prin metodele *setData* (`setData`), *getData* (`getData`) și *clearData* (`clearData`), puteți folosi obiectul

*dataTransfer* pentru a accesa formate predefinite de date. Aceste formate suportă manipularea de către interfața a oricărui element de la text la imagini și ancore.

Obiectul *clipboardData* este rezervat pentru acțiunile de editare executate prin meniul Edit, meniurile de *shortcut* și tastele de shortcut. Spre deosebire de obiectul *dataTransfer*, el transferă informația folosind memoria clipboard a sistemului și o reține până când ea este acoperită de datele necesare următoarei operații de editare. Așa cum am menționat anterior, această formă de transfer a datelor este potrivită în special pentru copierea multiplă a acelorași date. Ca și obiectul *dataTransfer*, obiectul *clipboardData* suportă metodele *setData*, *getData* și *clearData*.

Se poate folosi codul de editare personalizată pentru a executa oricare din următorii manipulatori de eveniment legați de editare: *onbeforecut*, *onbeforecopy*, *onbeforepaste*, *oncut*, *oncopy* și *onpaste*. Operațiile *cut* (tăiere) și *copy* (copiere) ale obiectului *clipboardData* folosesc metoda *setData* pentru a transfera datele de la obiectul sursă; operațiile sursă folosesc metoda *getData* pentru a recupera datele pentru obiectul țintă.

Spre deosebire de Windows, implementarea transferului de date în Internet Explorer 5 nu suportă modificările folosind tastele *CTRL* și *SHIFT* pentru operațiile de *drag-and-drop*. Cu alte cuvinte, utilizatorul nu poate alege dacă dorește să copieze, să mute sau să lege obiectul folosind aceste modificări. În schimb, trebuie

să folosiți proprietatea *effectAllowed* pentru a stipula funcționalitatea de *drag* și proprietatea *dropEffect* pentru a notifica utilizatorul.

Proprietatea *effectAllowed* specifică care operații de transfer a datelor sunt permise pentru obiect. Această proprietate este în general stabilită în evenimentul *ondragstart* pentru obiectul sursă, deoarece acesta este ultimul eveniment declanșat înainte de începerea operației de *drag-and-drop*.

Proprietatea *effectAllowed* suportă un set flexibile de valori, inclusiv *copy* (copiere), *move* (mutare), *link* (legare), *copyMove* (copiereMutare), *copyLink* (copiereLegare), *linkMove* (legareMutare), *all* (toate) și *none* (nici una). Pentru valori compuse cum ar fi *copyMove*, care denumesc două operații diferite de transfer a datelor, obiectul țintă implicit determină care din aceste valori este implementată.

Proprietatea *effectAllowed* poate de asemenea modifica comportarea implicită a altor aplicații. De exemplu, script-ul browserului poate seta proprietatea *copy* (copiere) și astfel să schimbe proprietatea implicită a Word-ului care este de *move* (mutare). În cadrul browserului, copierea este comportarea implicită a *effectAllowed*, exceptând ancorele care sunt setate implicit pe *link* (legare).

Atât *effectAllowed*, cât și *dropEffect* sunt disponibile doar prin obiectul *dataTransfer* și trebuie folosite împreună pentru ca *dropEffect* să funcționeze.

Proprietatea *dropEffect* stipulează care cursor standard de sistem va fi afișat în timpul unei operații de *drag-and-drop*. Valoarea acestei proprietăți este stabilită pentru obiectul țintă în evenimentul *ondragenter* sau *ondragover*. Excepția apare când o imagine, o ancoră dintr-un cadru, un obiect *textArea* sau un obiect pentru introducerea textului conține elementul sursă. Aceste obiecte suportă implicit *drag-and-drop* încă de la Internet Explorer 4.0. În consecință, obiectul țintă nu poate schimba setarea *dropEffect*.

**Notă:** Pentru afișarea cursorului dorit până la drop-ul final, trebuie să se anuleze acțiunea implicită a evenimentelor *ondragenter*, *ondragover* și *ondrop* și trebuie să se seteze proprietatea *dropEffect*. În caz contrar, cursorul de copiere, mutare sau legare stabilit de această proprietate este afișat doar până când mouse-ul este mutat deasupra primei ținte valabile pentru *drop*, moment în care el este înlocuit cu cursorul *drop/no-drop* pe durata operației de *drag*.

Toți parametrii pentru metodele *setData*, *getData* și *clearData* sunt insensibile la scrierea cu majuscule sau fără majuscule (*case-insensitive*) și primul parametru este întotdeauna o valoare șir. Atât obiectul *dataTransfer*, cât și *clipboardData* suportă aceste metode.

Metoda *setData* specifică ce format de date și ce informații vor fi transferate prin *dataTransfer* sau *clipboardData*. Metoda *setData* este de obicei invocată în evenimentul *ondragstart* pentru sursa de *drag* sau de evenimentul *onbeforecut* sau



*onbeforecopy* pentru editare. Primul parametru acceptă o valoare șir de text sau *URL*, textul fiind implicit. Al doilea parametru, o variabilă, asociază informația cu obiectul sursă. Această informație poate fi text descriptiv, o cale sursă către o imagine sau un *URL* pentru o ancoră. Al doilea parametru este necesar când sunt dorite informații privind locația obiectului care este transferat. Astfel de situații include furnizarea unor *URL*-uri pentru ancore sau shortcut-uri de desktop, precum și căi sursă pentru imagini.

Metoda *getData* specifică ce format de date vor fi recuperate din clipboard folosind obiectul *dataTransfer*. Această metodă este invocată de obiectul țintă și are un singur parametru necesar. Parametrul este un șir reprezentând formatul de date pe care ținta de drop trebuie să-l citească. Acest parametru trebuie să se potrivească cu formatul datelor care sunt transferate prin metoda *setData*, și anume text sau URL. Valoarea parametrului determină modul în care metoda *getData* interpretează informația transferată prin al doilea parametru a metodei *setData*. Când formatul datelor transferate prin *setData* și *getData* nu se potrivesc, operația de *drag* eșuează.

Metoda *clearData* a obiectului *dataTransfer* șterge unul sau mai multe formate de date din obiectul *dataTransfer*. Această metodă recunoște mai multe formate decât celelalte două, inclusiv, text, URL, fișier, HTML și imagine. Toate formatele sunt șterse dacă nu este indicat nici un parametru. În general, această

metodă este cea mai utilă în evenimente sursă, cum ar fi evenimentul *ondragstart*. Totuși, *clearData* poate modifica comportarea implicită a obiectului țintă când este invocată în evenimentul *ondrop*.

Pentru a completa noile elemente subliniate mai sus, Internet Explorer are un set bogat de evenimente de notificare privind obiectele sursă de date, cât și obiectele țintă. Toate evenimentele de transfer a datelor suportă modificarea și anularea acțiunii implicite. În consecință, se poate face codul de transfer a datelor modular prin abordarea acestor evenimente la un nivel reținut. La codarea unei comportări personalizate, trebuie să se anuleze acțiunile implicite ale tuturor evenimentelor de transfer a datelor care sunt abordate, astfel încât comportările implicite ale browserului să nu poată modifica codul personalizat.

Următoarele evenimente *drag-and-drop* se declanșează asupra obiectului sursă. Se observă că obiectul sursă primește notificare când operația de *drag* se termină – sub forma evenimentului *ondragend* – dar și dacă selecția a fost depusă la o țintă valabilă. Această informație este furnizată de către evenimentul *ondrop* a obiectului țintă.

*ondrag (la tragere)*

*ondragend (la tragere sfârșit)*

*ondragenter (la tragere intrare)*

*ondragstart (la tragere pornire)*

Următoarele evenimente se declanșează asupra obiectului țintă.

*ondragenter (la tragere intrare)*

*ondragleave (la tragere eliberare)*

*ondragover (la tragere peste)*

*ondrop (la lăsare)*

Următoarele evenimente acționează asupra obiectului sursă pentru evenimentele de editare. Se folosește metoda *getData* a obiectului *clipboardData* pentru a transfera informația:

*onbeforecopy (la înainte copiere)*

*onbeforecut (la înainte tăiere)*

*oncopy (la copiere)*

*oncut (la tăiere)*

Următoarele evenimente acționează asupra obiectului țintă pentru evenimentele de editare. Se folosește metoda *getData* a obiectului *clipboardData* pentru a transfera informația:

*onbeforepaste (la înainte aplicare)*

*onpaste (la aplicare)*

Se folosește oricare din evenimentele amintite pentru a modifica comportarea implicită a browserului în favoarea abordării personalizate a evenimentelor într-o aplicație web. Tot ce trebuie să se facă este să se anuleze acțiunea implicită a

evenimentului prin introducerea valorii de *return event.returnValue=false* în manipulatorul adecvat al evenimentului.

Urmează o descriere a evenimentelor *drag-and-drop*, organizată în ordinea în care acestea se declanșează.

Evenimentul *ondragstart* inițiază operația de *drag*. Introducerea valorii false în acest eveniment dezactivează toată funcționalitatea de *drag*. Este de asemenea evenimentul cheie pentru folosirea efectivă a *setData* și *effectAllowed*.

Evenimentul *ondrag* acționează continuu, indiferent dacă pointer-ul mouse-ului este deasupra unui obiect sursă, țintă sau invalid.

Evenimentul *ondragenter* acționează înaintea evenimentului *ondragover* asupra unei ținte. Evenimentul *ondragenter* este un loc bun pentru folosirea *getData*. Pentru menținerea cursorului *drag/drop*, se setează *dropEffect* și se anulează comportarea implicită a browserului în acest eveniment.

Evenimentul *ondragover* acționează continuu în timp ce selecția operației *drag* este deasupra unei ținte valabile pentru operație de *drop*. Pentru menținerea cursorului *drag/drop*, setează *dropEffect* și se anulează comportarea implicită a browserului în acest eveniment.

Evenimentul *ondragleave* se declanșează dacă pointerul mouse-ului părăsește ținta de drop.

Evenimentul *ondrop* se declanșează dacă utilizatorul lasă selecția deasupra unei ținte de *drop* valabile. Este cel mai eficient eveniment pentru invocarea *getData*. Pentru menținerea cursorului *drag/drop*, se setează *dropEffect* și se anulează comportarea implicită a browserului în acest eveniment.

Evenimentul *ondragend* se declanșează când operația de *drag* se termină, printr-un *drop* reușit sau nu.

În Internet Explorer 4.0, controlul programatic asupra operațiilor de transfer a datelor este limitat. Nu există obiecte, metode sau proprietăți ale transferului de date. Mai mult, orice capacitate existentă de *drag-and-drop* este restrânsă doar la browser. Ancorele și imaginile crează o legătură când utilizatorul le “trage” pe desktop. Dacă utilizatorul “trage” o ancoră spre o altă fereastră a browserului, cel de al doilea browser navighează la URL-ul ancorei. O imagine “trasă” într-o altă fereastră a browserului este afișată singură pe o pagină. Se pot folosi controalele de text pentru a muta conținutul dintre acestea pe aceeași pagină, precum și de la o pagină la alta. Se poate anula oricare din aceste comportări implicite sau se poate trimite notificări că o operație de *drag* a fost inițiată în evenimentul *ondragstart*. Nu este disponibil nici un control programatic asupra editării prin meniul *Edit*, meniul de *shortcut* sau acceleratori de tastatură în Internet Explorer 4.0. Această funcționalitate a fost adăugată în Internet Explorer 5 și după sub forma noilor evenimente *cut*, *copy* și *paste*.

Internet Explorer 5 onorează implementarea în Internet Explorer 4 funcționalității implicite de *drag-and-drop*. Pentru a introduce coduri personalizate pentru acțiunile implicite în versiune 4.0 ale acestor obiecte, pur și simplu se anulează valoarea de *return* în *ondragstart*. În acest moment, imaginile, ancorele și controalele de text necesită același volum de codare ca și de exemplu obiectele “*div*” sau “*span*”. Pe scurt, dacă se mută text într-un obiect *textArea*, trebuie să se ia textul din sursa *textArea* și apoi să se updateze sursa și ținta *textArea*. Proprietatea “*innerText*” este adecvată acestui scop.

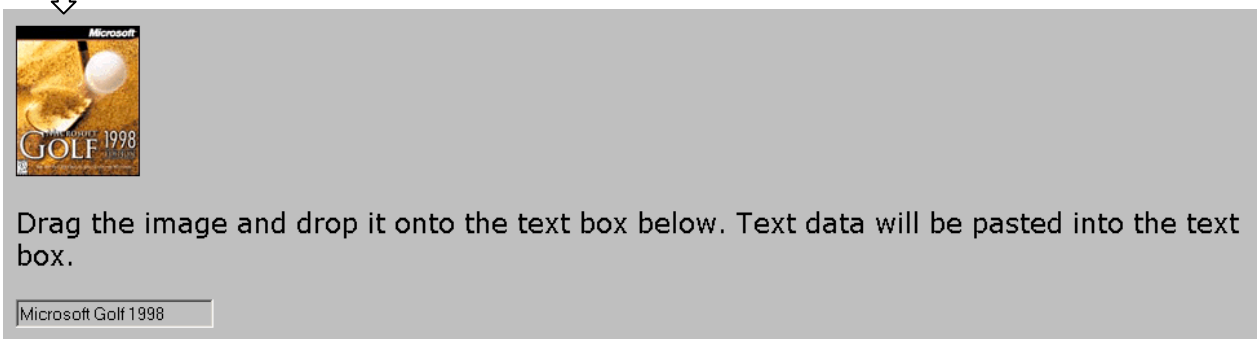
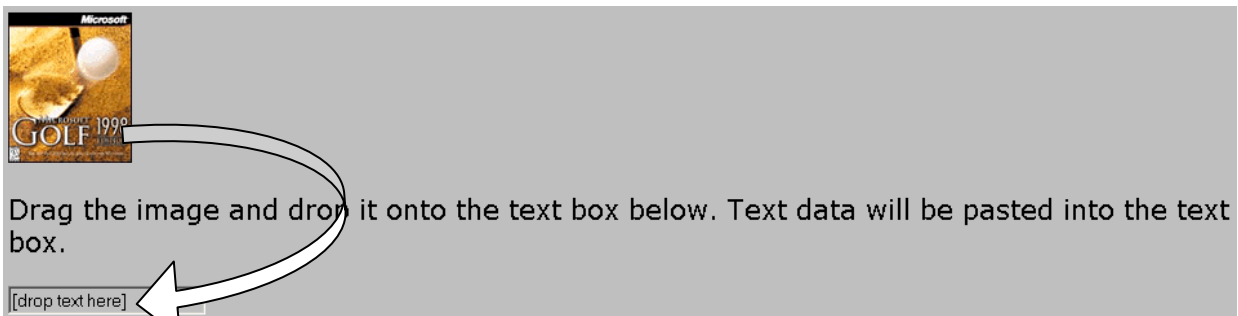
Dacă se dorește scrierea unor scripturi pentru browsere lansate înainte de Internet Explorer 5, se poate realiza o verificare simplă pentru a preveni erorile de script care pot apărea în paginile web care conțin funcționalitate de transfer a datelor. Tot ceea ce trebuie să se facă este introducerea unei propoziții condiționale prin care executarea codului se va baza pe existența obiectelor *dataTransfer* și *clipboardData*, așa după cum se arată în exemplul următor.

Se poate folosi setările de securitate ale Internet Explorer pentru a controla operațiile de *script*, de *copy* și *paste* care folosesc obiectele *dataTransfer* și *clipboardData*. Implicit, nu există nici o restricție asupra capacității de scriptare pentru a modifica memoria *clipboard* a sistemului și toate formatele de date recunoscute pot fi transferate liber între Internet Explorer și alte aplicații care rulează în sistem.

Utilizatorul are posibilitatea de a dezactiva această componentă prin modificarea grupului de butoane radio *“Allow paste operations via script”* (Permiteți operații de copiere prin intermediul scripturilor) din căsuța de dialog *Security Settings*, disponibilă prin *appletul Internet Options din Control Panel*. Utilizatorul poate personaliza setările pentru fiecare zonă de securitate.

Exemplul arată codul primar pentru programarea operației de *drag-and-drop*. Acest cod reprezintă temelia unei aplicații *shopping-cart* în care utilizatorul “trage” icoana în formularul de comandă și primește *feedback* textual privind articolul ales.

Imaginea folosită în exemplu:



## 2.7. [Aplicația 7](#). Informații despre client

Odată cu IE 5.0, o sumă de informații despre capacitatea clientului sunt disponibile ca unul dintre comportamentele implicite ale browserului. Următorul exemplu extrage aceste capacități și le afișează:

```
<HTML xmlns:IE>
<HEAD>
<TITLE>clientCaps Behavior Sample</TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; CHARSET=iso-
8859-1">
<STYLE>
@media all {IE\:\CLIENTCAPS {behavior:url(#default#clientCaps)}}
</STYLE>
<SCRIPT><!--
function window.onload() {
sTempStr =
"availHeight  = " + oClientCaps.availHeight  + "\n" +
"availWidth   = " + oClientCaps.availWidth   + "\n" +
"bufferDepth  = " + oClientCaps.bufferDepth  + "\n" +
"colorDepth   = " + oClientCaps.colorDepth   + "\n" +
"connectionType = " + oClientCaps.connectionType + "\n" +
"cookieEnabled = " + oClientCaps.cookieEnabled + "\n" +
"cpuClass     = " + oClientCaps.cpuClass     + "\n" +
"height       = " + oClientCaps.height       + "\n" +
"javaEnabled  = " + oClientCaps.javaEnabled  + "\n" +
"platform     = " + oClientCaps.platform     + "\n" +
"systemLanguage = " + oClientCaps.systemLanguage + "\n" +
"userLanguage = " + oClientCaps.userLanguage + "\n" +
"width        = " + oClientCaps.width        + "\n" ;
oPre.innerText = sTempStr;}
--></SCRIPT>
</HEAD>
<BODY>
<BLOCKQUOTE CLASS="body">
<IE:CLIENTCAPS ID="oClientCaps" />
<PRE id="oPre"></PRE>
</BLOCKQUOTE>
```



```
</BODY>  
</HTML>
```

Rezultatul se prezintă în felul următor:

```
availHeight = 768  
availWidth = 1024  
bufferDepth = 0  
colorDepth = 32  
connectionType = lan  
cookieEnabled = true  
cpuClass = x86  
height = 768  
javaEnabled = true  
platform = Win32  
systemLanguage = en-us  
userLanguage = en-us  
width = 1024
```

Capabilitățile browserului pot fi exploatate așa cum arată exemplul următor:

```
<!--[if gte IE 5]>  
<SCRIPT LANGUAGE="Javascript">  
alert("Congratulations! You are running Internet Explorer 5 or greater.");  
</SCRIPT>  
<P>Thank you for closing the message box.</P>  
<![endif]-->
```

## 2.8. Aplicația 8. Instrumente grafice

În unele cazuri, este necesar un volum mare de scripturi suplimentare pentru a executa calcule noi când un obiect este mutat la o nouă locație. Acest lucru devine evident când se observă obiecte poziționate în relație cu alte obiecte

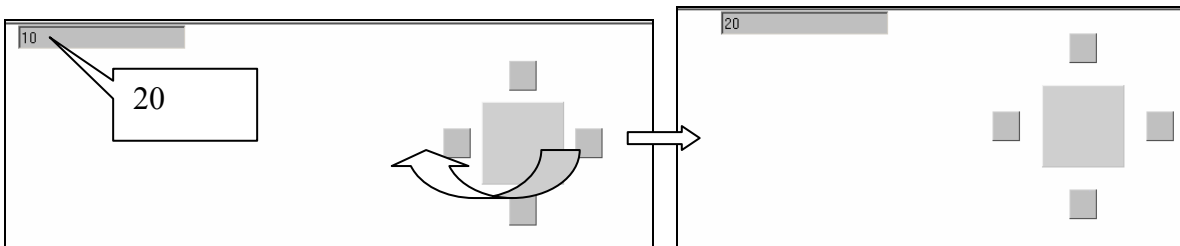
```
<HTML>  
<HEAD>  
<TITLE>Drawings...</TITLE>  
<STYLE>
```

```
.block {position: absolute; top: 100; left: 100; height: 75; width: 75; background-color: #CFCFCF; border: "1 solid"; border-left-color: #EFEFEF; border-top-color: #EFEFEF; border-bottom-color: #C0C0C0; border-right-color: #C0C0C0;}
.block2 {position: absolute; top: 0; left: 0; height: 25; width: 25; background-color: #C0C0C0; border: "1 solid"; border-left-color: #CFCFCF; border-top-color: #CFCFCF; border-bottom-color: #505050; border-right-color: #505050;}
</STYLE>
<SCRIPT>
window.onload=fnInit;
var iOffset=10;
function fnInit(){
  oDiv.setAttribute("moving",false);
  oDiv.style.left=document.body.clientWidth/2 - oDiv.offsetWidth/2;
  oDiv.style.top=oOffset.offsetTop + oOffset.offsetHeight + iOffset +
oBlock4.offsetHeight + 10;
  oBlock1.style.setExpression("top","oDiv.offsetTop - iOffset -
oBlock1.offsetHeight");
  oBlock1.style.setExpression("left","oDiv.offsetLeft + (oDiv.offsetWidth/2 -
oBlock1.offsetWidth/2)");
  oBlock2.style.setExpression("top","oDiv.offsetTop + oDiv.offsetHeight +
iOffset");
  oBlock2.style.setExpression("left","oDiv.offsetLeft + (oDiv.offsetWidth/2 -
oBlock2.offsetWidth/2)");
  oBlock3.style.setExpression("top","oDiv.offsetTop + (oDiv.offsetHeight/2 -
oBlock3.offsetHeight/2)");
  oBlock3.style.setExpression("left","oDiv.offsetLeft - iOffset -
oBlock3.offsetWidth");
  oBlock4.style.setExpression("top","oDiv.offsetTop + (oDiv.offsetHeight/2 -
oBlock4.offsetHeight/2)");
  oBlock4.style.setExpression("left","oDiv.offsetLeft + oDiv.offsetWidth +
iOffset");
}
function fnStartMove(){
  oDiv.adjustX=event.clientX - oDiv.offsetLeft;
  oDiv.adjustY=event.clientY - oDiv.offsetTop;
  oDiv.moving=true;
}
function fnStopMove(){
  oDiv.moving=false;
}
```

```

function fnMove(){
  if(oDiv.moving==true){
    if(event.clientX - oDiv.adjustX>0){
      oDiv.style.left=event.clientX - oDiv.adjustX;
    }
    if(event.clientY - oDiv.adjustY>0){
      oDiv.style.top=event.clientY - oDiv.adjustY;
    }
  }
}
function fnUpdate(){
  iOffset=parseInt(oOffset.value);
}
</SCRIPT>
</HEAD>
<BODY TOPMARGIN=0 LEFTMARGIN=0 BGPROPERTIES="FIXED"
BGCOLOR="#FFFFFF" LINK="#000000" VLINK="#808080"
ALINK="#000000">
<BLOCKQUOTE CLASS="body">
<P><input TYPE=text VALUE="10" onkeyup="fnUpdate()" ID="oOffset"></P>
<DIV ID="oDiv" CLASS="block" STYLE="cursor: move;"
onmouseout="fnStopMove()" onmouseup="fnStopMove()"
onmousemove="fnMove()" onmousedown="fnStartMove()"></DIV>
<DIV ID="oBlock1" CLASS="block2"></DIV>
<DIV ID="oBlock2" CLASS="block2"></DIV>
<DIV ID="oBlock3" CLASS="block2"></DIV>
<DIV ID="oBlock4" CLASS="block2"></DIV>
<DIV STYLE="position:absolute;top:400">
</DIV>
</BLOCKQUOTE>
</BODY>
</HTML>

```



## 2.9. [Aplicația 9](#). Mișcarea planetelor

```
<HTML>
<HEAD>
<TITLE>Orbitele planetelor</TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; CHARSET=iso-
8859-1">
<style>
.star {
    position: absolute;
    font-size: 40;
    width: 40;
    height: 40;
    color: yellow;
    vertical-align: middle;
    text-align: center;
    z-index: 3;
}
.planet {
    position: absolute;
    font-size: 35;
    width: 35;
    height: 35;
    color: blue;
    vertical-align: middle;
    text-align: center;
    z-index: 2
}
.moon {
    position: absolute;
    font-size: 15;
    width: 15;
    height: 15;
    color: white;
    vertical-align: middle;
    text-align: center;
    z-index: 1;
}
.desc {
    position: absolute;
```

```
    top: 0;
    left: 0;
    background-color: white;
    color: black;
    z-index: 0;
}
.pname {display: none;}
</style>
<SCRIPT>
//<!--
var timeBase = new Date()
var time = timeBase
function update(){
    // fake a time property that advances
    time = new Date() - timeBase
    document.recalc(true)
}
// return an angle in radians as a function of time and a rate (essentially angular
velocity)
function angle(rate, t){
    return ((t / 10) / rate) % (2 * Math.PI)
}
//following function is for determining the left property of each planet
function orbitX(obj, radius, rate, t){
    return (obj.style.pixelLeft + obj.style.pixelWidth / 2) +
Math.round(Math.cos(angle(rate, t)) * radius * 200)
}
//following function is for determining the top property of each planet
function orbitY(obj, radius, rate, t){
    return (obj.style.pixelTop + obj.style.pixelHeight / 2) +
Math.round(Math.sin(angle(rate, t)) * radius * 200)
}
function init(){
    //center the sun in the document
    sun.style.setExpression("left", "document.body.clientWidth / 2 -
sun.style.pixelWidth / 2", "javascript")
    sun.style.setExpression("top", "document.body.clientHeight / 2 -
sun.style.pixelHeight / 2", "javascript")
    //determine orbit of Mercury
```

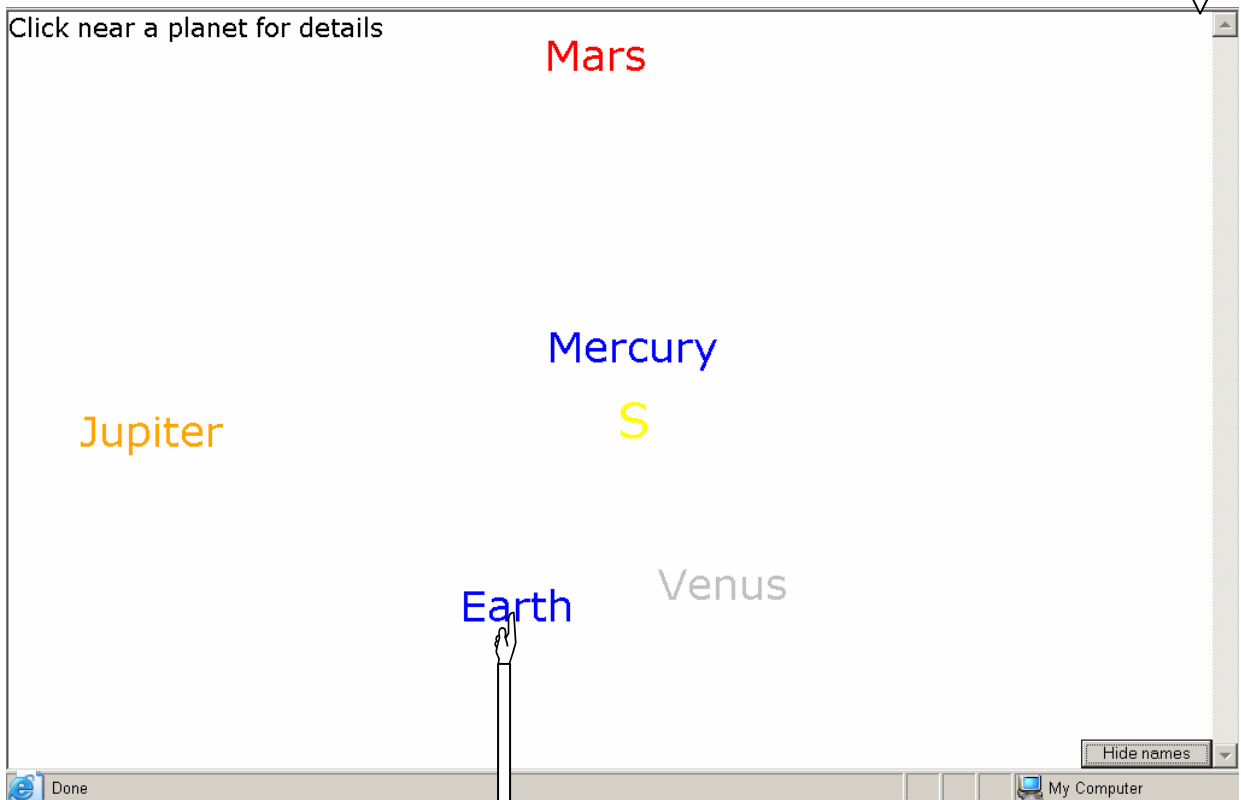
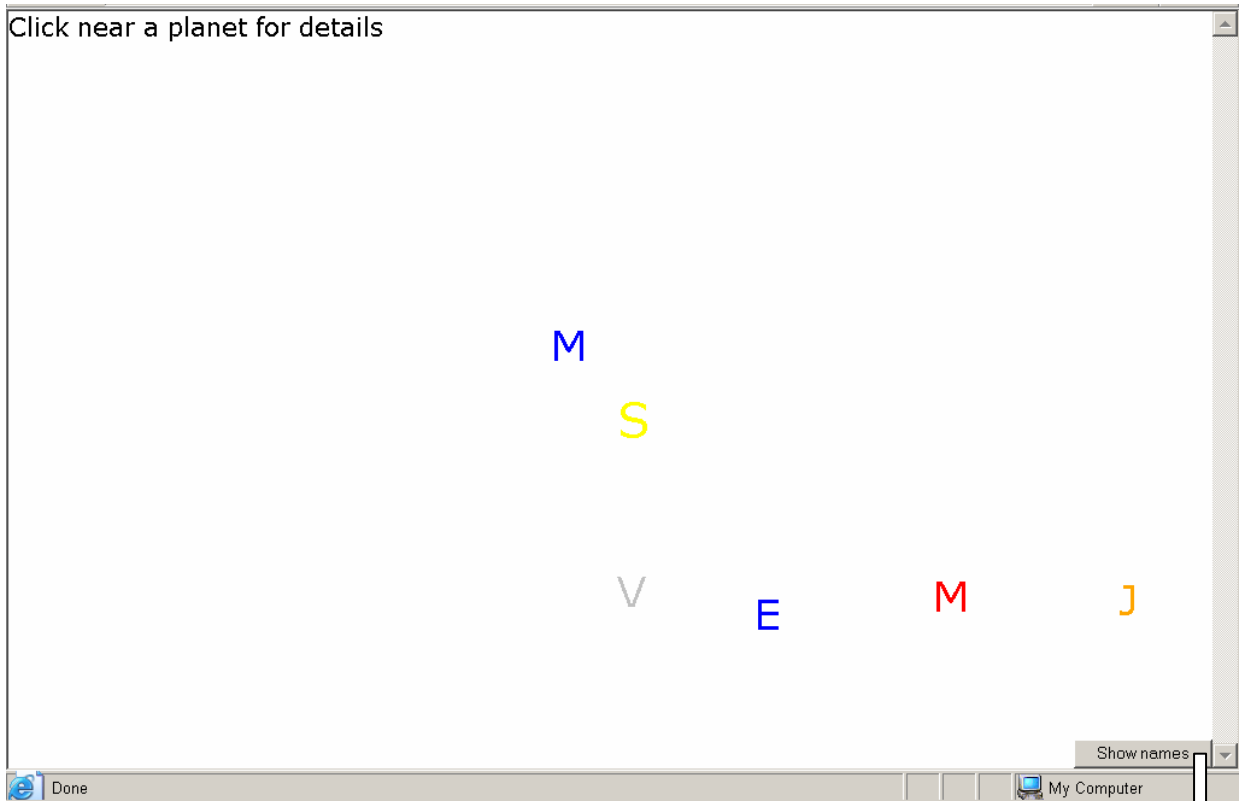
```
mercury.style.setExpression("left", "orbitX(sun, 0.387, 87.97, time) -  
mercury.style.pixelWidth / 2", "javascript")  
mercury.style.setExpression("top", "orbitY(sun, 0.387, 87.97, time) -  
mercury.style.pixelHeight / 2", "javascript")  
//determine orbit of Venus  
venus.style.setExpression("left", "orbitX(sun, 0.723, 224.7, time) -  
venus.style.pixelWidth / 2", "javascript")  
venus.style.setExpression("top", "orbitY(sun, 0.723, 224.7, time) -  
venus.style.pixelHeight / 2", "javascript")  
//determine orbit of Earth  
earth.style.setExpression("left", "orbitX(sun, 1.0, 365.26, time) -  
earth.style.pixelWidth / 2", "javascript")  
earth.style.setExpression("top", "orbitY(sun, 1.0, 365.26, time) -  
earth.style.pixelHeight / 2", "javascript")  
//determine orbit of the Moon  
luna.style.setExpression("left", "orbitX(earth, 0.1, 28, time) -  
luna.style.pixelWidth / 2", "javascript")  
luna.style.setExpression("top", "orbitY(earth, 0.1, 28, time) -  
luna.style.pixelHeight / 2", "javascript")  
//determine orbit of Mars  
mars.style.setExpression("left", "orbitX(sun, 1.524, 686.98, time) -  
mars.style.pixelWidth / 2", "javascript")  
mars.style.setExpression("top", "orbitY(sun, 1.524, 686.98, time) -  
mars.style.pixelHeight / 2", "javascript")  
//determine orbit of Deimos  
deimos.style.setExpression("left", "orbitX(mars, 0.1, 50, time) -  
deimos.style.pixelWidth / 2", "javascript")  
deimos.style.setExpression("top", "orbitY(mars, 0.1, 50, time) -  
deimos.style.pixelHeight / 2", "javascript")  
//determine orbit of Phobos  
phobos.style.setExpression("left", "orbitX(mars, 0.18, 75, time) -  
phobos.style.pixelWidth / 2", "javascript")  
phobos.style.setExpression("top", "orbitY(mars, 0.18, 75, time) -  
phobos.style.pixelHeight / 2", "javascript")  
//determine orbit of Jupiter  
jupiter.style.setExpression("left", "orbitX(sun, 2.2, 1000, time) -  
jupiter.style.pixelWidth / 2", "javascript")  
jupiter.style.setExpression("top", "orbitY(sun, 2.2, 1000, time) -  
jupiter.style.pixelHeight / 2", "javascript")  
//determine orbit of Jupiter's moons
```

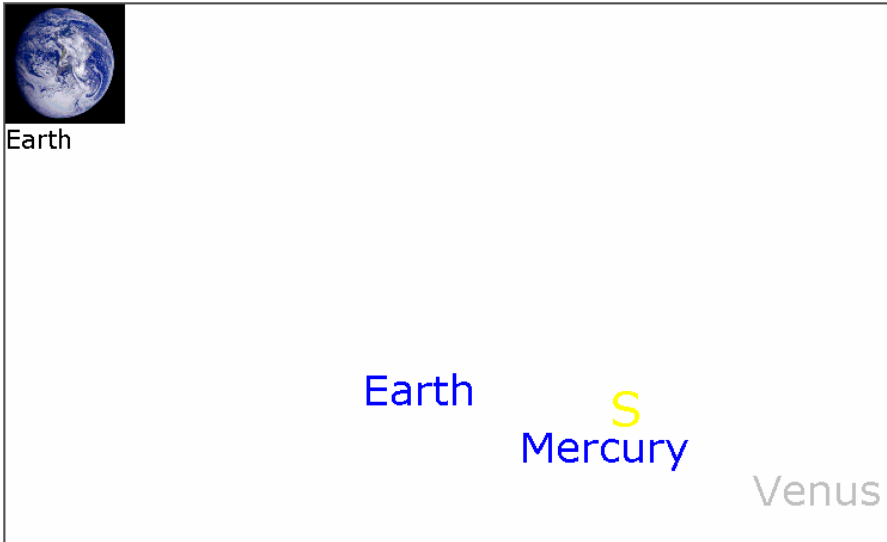
```
    j1.style.setExpression("left", "orbitX(jupiter, 0.1, 100, time) -
j1.style.pixelWidth / 2", "javascript")
    j1.style.setExpression("top", "orbitY(jupiter, 0.1, 100, time) -
j1.style.pixelHeight / 2", "javascript")
    j2.style.setExpression("left", "orbitX(jupiter, 0.12, 50, time) -
j2.style.pixelWidth / 2", "javascript")
    j2.style.setExpression("top", "orbitY(jupiter, 0.12, 50, time) -
j2.style.pixelHeight / 2", "javascript")
    j3.style.setExpression("left", "orbitX(jupiter, 0.13, 200, time) -
j3.style.pixelWidth / 2", "javascript")
    j3.style.setExpression("top", "orbitY(jupiter, 0.13, 200, time) -
j3.style.pixelHeight / 2", "javascript")
    j4.style.setExpression("left", "orbitX(jupiter, 0.15, 300, time) -
j4.style.pixelWidth / 2", "javascript")
    j4.style.setExpression("top", "orbitY(jupiter, 0.15, 300, time) -
j4.style.pixelHeight / 2", "javascript")
    j5.style.setExpression("left", "orbitX(jupiter, 0.17, 400, time) -
j5.style.pixelWidth / 2", "javascript")
    j5.style.setExpression("top", "orbitY(jupiter, 0.17, 400, time) -
j5.style.pixelHeight / 2", "javascript")
    //position the Show Names toggle button
    toggle.style.setExpression("left", "document.body.clientWidth -
toggle.offsetWidth", "javascript")
    toggle.style.setExpression("top", "document.body.clientHeight -
toggle.offsetHeight", "javascript")
    setInterval("update()", 50)
}
var showNames = "none"
function toggleNames(){
    if (showNames == "none"){
        toggle.value = "Hide names"
        showNames = "inline"
    }else{
        toggle.value = "Show names"
        showNames = "none"
    }
}
for (i = 0 ; i < document.all.tags("SPAN").length ; i++){
    elem = document.all.tags("SPAN").item(i)
    if (elem.className == "pname"){
        elem.style.display = showNames
```

```
}
}
}
//-->
</SCRIPT>
</HEAD>
<body bgcolor="#FFFFFF" onload="init()" onresize="update()"
onclick="showinfo()">
<input id=toggle type=button value="Show names" style="position: absolute"
onclick="toggleNames()">
<div id=test style="font-size: 20; color: green"></div>
<div id=sun class=star><center>S</center></div>
<div id=mercury class=planet style="color: grey">M<span
class=pname>ercury</span></div>
<div id=venus class=planet style="color: rgb(192, 192, 192)">V<span
class=pname>enus</span></div>
<div id=earth class=planet style="color: light-blue">E<span
class=pname>arth</span></div>
<div id=luna class=moon>l</div>
<div id=mars class=planet style="color: red">M<span
class=pname>ars</span></div>
<div id=deimos class=moon style="color: grey">d</div>
<div id=phobos class=moon style="color: grey">p</div>
<div id=jupiter class=planet style="color: orange">J<span
class=pname>upiter</span></div>
<div id=j1 class=moon>j1</div>
<div id=j2 class=moon>j2</div>
<div id=j3 class=moon>j3</div>
<div id=j4 class=moon>j4</div>
<div id=j5 class=moon>j5</div>
<div id=defaultDesc class=desc>Click near a planet for details</div>
<SCRIPT>
//<!--
var descCur = defaultDesc
function showinfo(){
    // find the nearest element
    for (i = 0 ; i < document.all.tags("DIV").length ; i++){
        elem = document.all.tags("DIV").item(i)
        if (elem.className == "planet"){
            dx = event.x - elem.style.pixelLeft
```

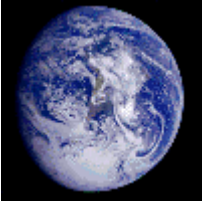
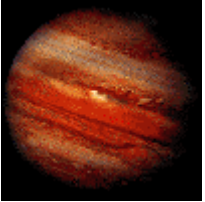
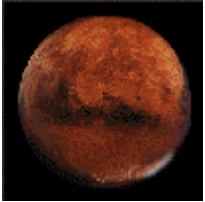
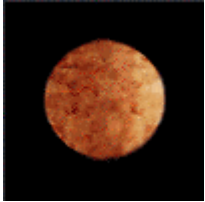



```
        dy = event.y - elem.style.pixelTop
        if (((dx * dx) + (dy * dy)) < 2500){
            descCur.style.visibility = "hidden"
            descCur = eval(elem.id + "Desc")
            descCur.style.visibility = "visible"
            return
        }
    }
}
showdefault()
}
function showdefault(){
    descCur.style.visibility = "hidden"
    descCur = defaultDesc
    descCur.style.visibility = "visible"
}
//-->
</SCRIPT>
<div id=venusDesc style="visibility: hidden" class=desc><br>Venus</div>
<div id=mercuryDesc style="visibility: hidden" class=desc><br>Mercury</div>
<div id=earthDesc style="visibility: hidden" class=desc><br>Earth</div>
<div id=marsDesc style="visibility: hidden" class=desc><br>Mars</div>
<div id=jupiterDesc style="visibility: hidden" class=desc><br>Jupiter (orbit not
accurate in demo)</div>
</BODY>
</HTML>
```





Imaginile folosite în aplicație:

earglobe.gif	jupglobe.gif	marglobe.gif	merglobe.gif	venglobe.gif
				

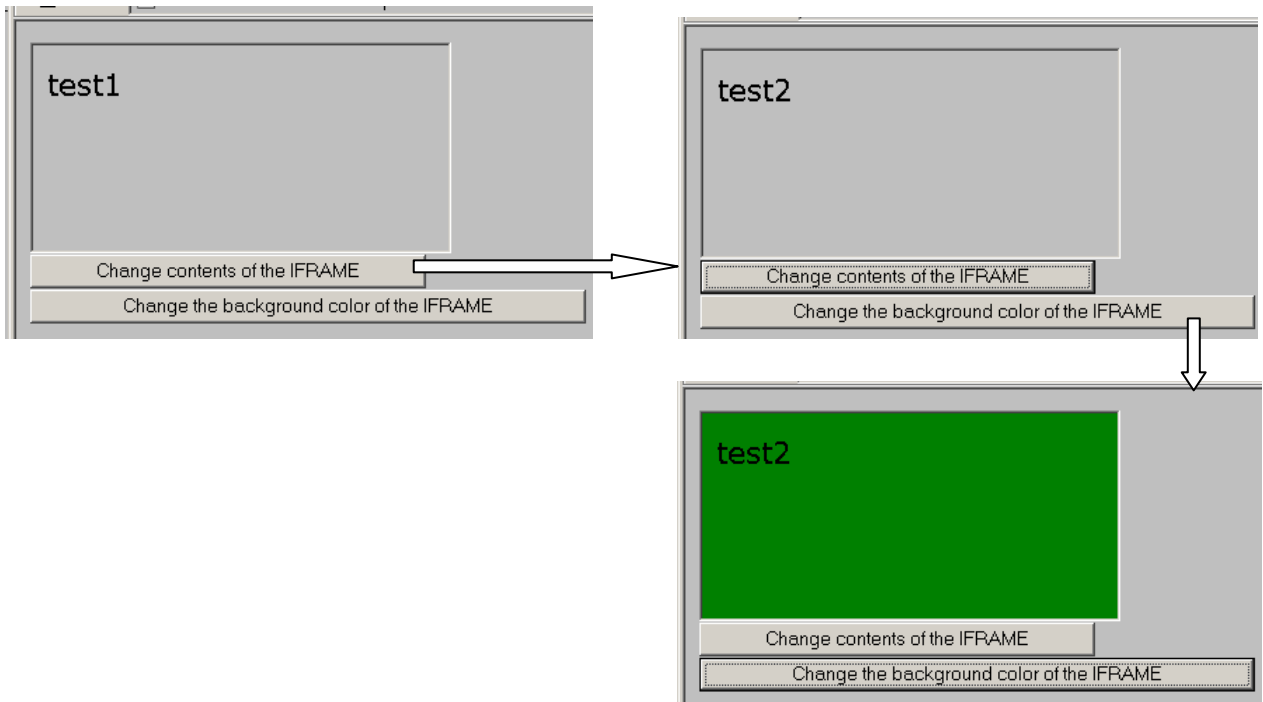
**2.10. [Aplicatia 10](#).** IFRAME și schimbarea conținutului documentului

```

<HTML>
<HEAD><TITLE>Changing Element Attributes</TITLE>
<SCRIPT LANGUAGE="JavaScript">
function change_image() {
    document.all.myiframe.src="exf";
    //use the all collection to access attributes of the IFRAME element
}
function change_background() {
    document.frames.myiframe.document.body.style.backgroundColor="green";
    // use the frames collection to access the document contained in the IFRAME.
}
</SCRIPT>
    
```

```
</HEAD>  
<BODY>  
  <IFRAME id=myiframe src="exe">  
  </IFRAME>  
  <BR>  
  <BUTTON onclick=change_image()>Change contents of the  
IFRAME</BUTTON>  
  <BR>  
  <BUTTON onclick=change_background()>Change the background color of the  
IFRAME</BUTTON>  
</BODY>  
</HTML>
```

Rezultatul execuției aplicației:



## Concluzii

Câteva concluzii se pot desprinde acum:

1. Modelul pe obiecte al documentului (DOM) introdus de firma Microsoft odată cu lansarea pe piață a produsului Microsoft Internet Explorer 4.0 face ca orice element HTML standard să poată fi referit prin intermediul identificatorului său (*id* =);
2. Introducerea evenimentelor și tratării evenimentelor (E.M.) de mouse și tastatură odată cu același I.E. 4.0 face ca la elementele individuale ale documentului să li se poată asocia evenimente așa încât documentul devine activ la evenimentele client și capătă atributul de dinamic;
3. Mecanismul *event bubbling* este nou pentru HTML și pune la dispoziție un model eficient pentru încorporarea tratării pe evenimente în documentele Web bazat pe moștenirea simplă;
4. Evenimentele și proprietățile noi aduse de I.E. 5.0, I.E. 5.5 și I.E. 6.0 vin să completeze din neajunsurile descoperite în exploatarea documentelor cu I.E. 4.0;
5. Evoluția sistemelor client-server bazate pe Internet Explorer nu se oprește aici; este suficient să menționăm și alte elemente (netratate aici) care fac obiectul dezvoltării de aplicații web: ActiveX-uri, limbajul XML (extended markup language), WSH (windows scripting host), HTA (HTML Applications).

## Lista aplicațiilor

Exemplu clasic de HTML: ex1.htm

Elementul HTML sursă de evenimente: ex2.htm

Atașarea unui eveniment: ex3.htm

Aplicația 1: ex4.htm

Aplicația 2: ex5.htm

Aplicația 3: ex6.htm

Aplicația 4: ex7.htm

Aplicația 5: ex8.htm

Aplicația 6: ex9.htm

Aplicația 7: exa.htm

Aplicația 8: exb.htm

Aplicația 9: exc.htm

Aplicația 10: exd.htm, exe.htm, exf.htm

## Bibliografie

1. Mihai Damian, “Aplicații multimedia și documente hypertext” Cluj-Napoca, 2000.
2. Tiberiu Leția, “Java și Java Script”, Cluj-Napoca, 2001.
3. Eager Bil, “Utilizarea internetului”, Editura Teora, București, 1996.
4. Raggett Dave, “HTML 3\_2 Reference Specification\_files - W3C Recommendation”, 14-Jan-1997.
5. Darcy Dinucci, “Elements of WEB Design”, Perchpit Press, Berkeley, CA, 1998.
6. Spencer Drate Tom Olejar, “WEB Art”, Madison Square Press, New York, 1999.
7. Fred T. Hofstter, “Multimedia Leteracy”, McGraw-Hill Companies, New York, 2001.
8. \* \* \*, HTML and DHTML Reference; WEB development; .net development, [http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/dhtml.a  
sp.](http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/dhtml.asp)