

Evenimente JavaScript

Atât evenimentele cât și tratarea evenimentelor sunt acțiuni declanșate de către utilizatori și pot fi interpretate cu ajutorul funcțiilor JavaScript. Evenimentele declanșate de către utilizatori, cum sunt: click-stânga de mouse, mișcarea mouse-ului, operația drag-and-drop sau operații ale sistemului, cum sunt: încărcarea unui document sau a unei imagini, sunt incidente comune.

Cel mai des, evenimentele sunt declanșate de către un formular, cum ar fi: click sau dublu-click de mouse pe un buton, click pe butonul Reset, selectarea unui text dintr-un câmp de tip text sau textarea. Totuși, nu numai formularele pot apela funcții cu ajutorul evenimentelor. Important de reținut este faptul că orice element poate avea asociat un eveniment cu ajutorul căruia să apeleze o funcție.

Evenimentele JavaScript nu trebuie scrise neapărat între etichetele `<script language="JavaScript">` și `</script>`, deoarece fac parte din HTML. Totuși, numai cu HTML, nu se poate scrie o funcție care să fie acționată de către un eveniment.

Un alt element important este felul în care se scrie evenimentul. În HTML, nu contează dacă se scrie cu litere mari (ONCLICK), cu litere mici (onclick) sau cu litere mari la începutul cuvântului (onClick, onDbClick). O convenție este totuși scrierea evenimentelor cu literă mare la fiecare cuvânt ce îl conține.

Funcțiile se scriu de obicei înaintea apelării lor de către evenimente, dar nu vor fi executate înaintea tratării evenimentelor respective.

Evenimentele și tratarea evenimentelor

În această carte se utilizează două noțiuni pentru a defini diferite acțiuni ale utilizatorilor și anume: evenimente (events) și tratarea evenimentelor (event handlers). Evenimentele sunt: **Click** (click de mouse), **MouseOver** (cursorul mouse-ului trece peste un anumit element), **KeyDown** (este apăsată o tastă), iar tratarea evenimentelor reprezintă acțiunea ce urmează la acționarea evenimentului, cum ar fi: onClick, onMouseOver, onKeyDown. Acționarea sau tratarea unui eveniment, aduce după sine acționarea unei funcții JavaScript ce va prelua efectiv sarcina atribuită evenimentului.

De exemplu, dacă se dă click pe un buton dintr-un formular, se poate acționa o funcție ce va realiza: încărcarea unei pagini Web, a unei imagini, va deschide sau va închide o fereastră, va face diverse calcule, va trimite un e-mail.

Tratarea evenimentelor - clasificare

Evenimentele pot fi acționate de diferite elemente, cum ar fi:

Tratarea evenimentelor de către sistem

`onError` Acțiune ce va fi declanșată în urma eșecului încărcării unei imagini sau a unui document.
`onLoad` Acțiune ce va fi declanșată la încărcarea unui document.
`onUnLoad` Acțiune ce va fi declanșată atunci când un document nu se încarcă.

Tratarea evenimentelor de către un formular

`onBlur` Acțiune ce va fi declanșată când se pierde focalizarea de pe un element.
`onChange` Acțiune ce va fi declanșată la schimbarea unui element.
`onFocus` Acțiune ce va fi declanșată de focalizarea asupra unui element.
`onReset` Acțiune ce va fi declanșată când va fi acționat butonul Reset al unui formular.
`onSelect` Acțiune ce va fi declanșată atunci când se selectează un element.
`onSubmit` Acțiune ce va fi declanșată când va fi acționat butonul Submit al unui formular.

Tratarea evenimentelor de către tastatură

`onKeyDown` Acțiune ce va fi declanșată când se apasă o tastă.

onKeyPress Acțiune ce va fi declanșată când se apasă și se eliberează o tastă.
onKeyUp Acțiune ce va fi declanșată când se eliberează o tastă.

Tratarea evenimentelor de către mouse

onClick Acțiune ce va fi declanșată când se dă click stânga pe mouse.
ondblclick Acțiune ce va fi declanșată când se dă dublu click stânga pe mouse.
onmousedown Acțiune ce va fi declanșată când un buton al mouse-ului este apăsat.
onmousemove Acțiune ce va fi declanșată când se mișcă cursorul mouse-ului.
onmouseout Acțiune ce va fi declanșată când cursorul mouse-ului se deplasează în afara elementului sau a suprafeței delimitate.
onmouseover Acțiune ce va fi declanșată când cursorul mouse-ului se deplasează peste un element sau peste suprafața delimitată.
onmouseup Acțiune ce va fi declanșată când butonul mouse-ului este eliberat.

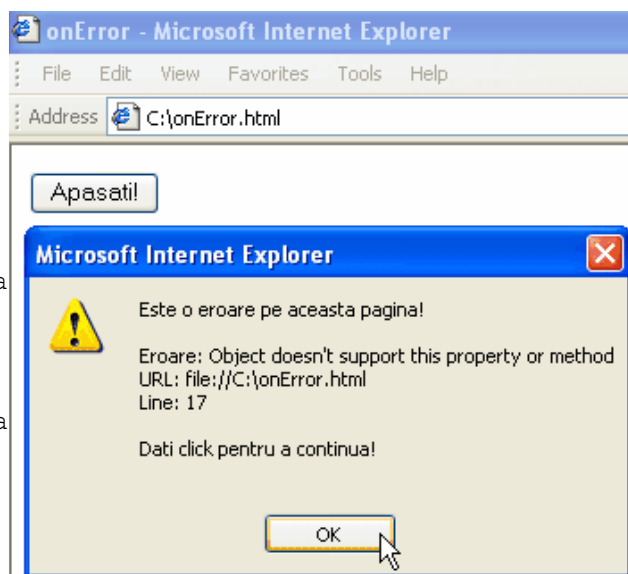
Tratarea evenimentelor de către sistem

onError va acționa în urma unei erori de încărcare a unei imagini sau a unui document. Pentru a utiliza acest eveniment, se apelează o funcție JavaScript ce va realiza o acțiune ca urmare a producerii unei erori. Funcției respective i se va transmite un mesaj, URL-ul la care s-a produs mesajul și numărul liniei de cod.

onError este event handler pentru obiectele: *Image* și *Window*.

Exemplu În acest exemplu, avem un formular ce conține butonul cu numele *Apasati!*. În urma acționării prin click de mouse asupra acestui buton, se apelează funcția *f()*, ce conține o eroare (în loc de *writeln*, am scris *writeln*). Fiind în interiorul scriptului JavaScript, va acționa *onError*, prin apelarea funcției *MesajEroare*. Funcția aceasta, primește prin cei trei parametri: mesaj, url și linie, datele de identificare ale erorii care s-a produs. Toate aceste informații se vor afișa într-o fereastră, prin intermediul metodei *alert()*.

```
<html>
<head>
  <title>   onError </title>
<script type="text/javascript">
onerror=MesajEroare
var x=""
function
MesajEroare (mesaj , url , linie)
{
  x="Este o eroare pe aceasta
    pagina!\n\nEroare:
    "+mesaj+"\n";
  x+="URL: " + url + "\n";
  x+="Line: " + linie + "\n\n";
  x+="Dati click pentru a
    continua!\n\n";
  alert(x);
  return true;
}
function f()
{
  document.writeln("Nu o sa ma
    vedeti ;)");
}
</script>
</head>
<body>
<form>
<input
                                type="button"
```



```
value="Apasati!" onclick="f()">
</form>
</body></html>
```

onLoad va acționa în urma încărcării unui document, a tuturor cadrelor dintr-o etichetă FRAMESET sau a unei imagini. Pentru a utiliza acest eveniment, apelăm o funcție JavaScript ce va realiza o acțiune ca urmare a încărcării unei imagini sau a unui document. De obicei, onLoad se utilizează pentru a verifica tipul de browser al vizitatorilor și versiunea pentru a încărca un document corect pe baza acestei informații.

onLoad este event handler pentru obiectele: *Image*, *Layer* și *Window*.

Exemplu

```
<html>
<head>
  <title>
    onLoad
  </title>
<script type="text/javascript">
function g()
{
  alert("Salut!");
}
</script>
</head>
<body onLoad="g()">
</body>
</html>
```

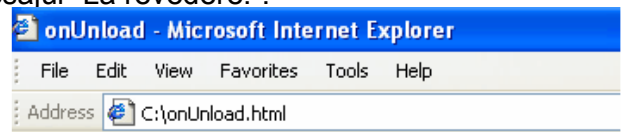


onUnload va acționa în momentul în care utilizatorul părăsește documentul afișat în fereastra curentă.

onUnload este event handler pentru obiectul: *Window*.

Exemplu În acest exemplu este un formular ce permite selectarea unei variante corecte. În urma alegerii făcute, va apărea un mesaj afișat într-o fereastră acționată de către metoda *alert*. La închiderea paginii, acționează evenimentul *onUnload*, ce apelează funcția *g()*, care afișează mesajul "La revedere!".

```
<html>
<head>
  <title>onUnload</title>
<script type="text/javascript">
function g()
{
  alert("La revedere!");
}
function f(raspuns)
{
  if (raspuns==1)
    alert("Corect!\n\nFelicitari!!");
  else
    alert("Nu prea va pricepeti :(");
}
</script>
<style type="text/css">
.anunt
{font-family: Geneva, Arial;
font-size: 18px;
font-weight: bolder;
color: #FF0000;
letter-spacing: 0.3em; }
.titlul
{font-family: Geneva, Arial;
```



Alegeți varianta corectă!

Cand ai nevoie de dragoste

cand ai nevoie de dragoste nu ti se da dragoste.
 cand trebuie sa iubesti nu esti iubit.
 cand esti singur nu poti sa scapi de singuratate.
 cand esti nefericit nu are sens sa o spui.

Mircea Cartarescu

Alexandru Andries

```

font-size: 18px;
font-weight: bold;
color: #0000CC;
border-bottom-width: thin;
border-bottom-style: ridge;
border-bottom-color: #0000CC;
background-color: #00FFFF;
border-top-style: ridge;
border-top-width: thin; }
</style>
</head>
<body onUnload="g()">
<form>
  <p class="anunt">Alegeti varianta
  corecta! </p>
  <p class="titlul">Cand ai nevoie de
  dragoste </p>
  cand ai nevoie de dragoste nu ti se
  da dragoste.<br />
  cand trebuie sa iubesti nu esti
  iubit.<br />
  cand esti singur nu poti sa scapi
  de singuratate.<br />
  cand esti nefericit nu are sens sa o
  spui.
  <p><input type="submit" name="Submit"
  value="Mircea Cartarescu"
  onClick="f(1)">
  <input type="submit" name="Submit2"
  value="Alexandru Andries"
  onClick="f(2)">
  </p>
</form>
</body>
</html>

```



Tratarea evenimentelor de către un formular

onBlur va acționa în urma pierderii focalizării asupra unui element. Se utilizează în special în formulare, când acționăm asupra unui buton, după care ne focalizăm pe alt buton sau undeva în afara lui, unde dăm click.

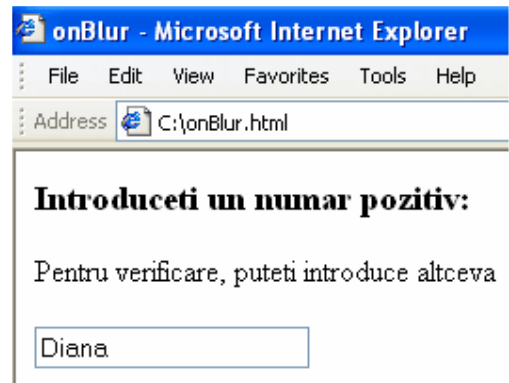
onBlur este event handler pentru obiectele: *Button*, *Checkbox*, *FileUpload*, *Layer*, *Password*, *Radio*, *Reset*, *Select*, *Submit*, *Text*, *Textarea*, *Window*.

Exemplu În exemplul de mai jos, am utilizat o funcție care preia o valoare introdusă în formular, o verifică și afișează un mesaj corespunzător. Dacă nu a fost introdus un număr, funcția `isNaN()` va returna valoarea "true", iar pe monitor va apărea mesajul "Introduceți un număr!". Mesajele afișate vor apărea pe ecran numai dacă introducem altceva decât un număr pozitiv. Evenimentul *onBlur* își face efectul, adică transmite valoarea introdusă de către utilizator, funcției **verificare**, dacă dăm click în afara zonei de text a formularului.

```

<html>
<head>
  <title>
    onBlur
  </title>
<script type="text/javascript">
function verificare(form)
  {
    var x=0;
    x=document.formular.date.value;
    y=isNaN(x);
    if (y==true)
      alert("Introduceti un numar!");
    else
      if (x<0)
        {
          alert("Va rog sa introduceti un numar
            pozitiv!");
        }
      }
  }
</script>
</head>
<body>
<H3>Introduceti un numar pozitiv:</H3>
Pentru verificare, puteti introduce altceva
<br />
<form name="formular">
<input type="text" name="date" value=""
size="20" onBlur='verificare(this.form) ' >
</form>
</body>
</html>

```



onChange va acționa în urma pierderii focalizării asupra unui element și a modificării conținutului unui câmp.

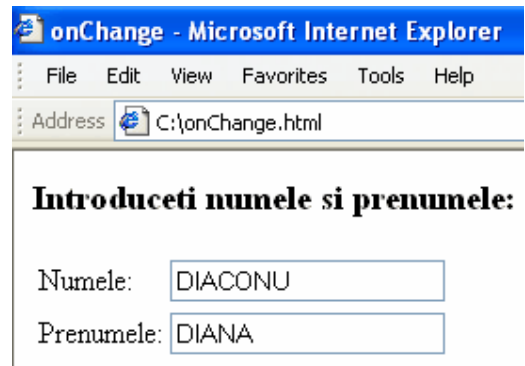
onChange este event handler pentru obiectele: *FileUpload*, *Select*, *Text*, *Textarea*.

Exemplu În exemplul de mai jos, am utilizat o funcție care preia șirul de caractere introdus în formular și îl transformă în șir scris cu majuscule.

```

<html>
<head><title>onChange</title>
<script type="text/javascript">
function majuscule(a)
  {
    a.value = a.value.toUpperCase();
  }
</script>
</head>
<body>
<H3>Introduceti numele si prenumele:</H3>
<form name="formular">
<table>
<tr>
  <td>Numele:</td>
  <td><input type="text" name="x" value=""
    size="20"
    onChange="majuscule(this) "></td>
</tr>

```



```

<tr>
  <td>Prenumele:</td>
  <td><input type="text" name="y" value=""
    size="20"
onChange="majuscule(this)"></td>
</tr>
</table>
</form>
</body>
</html>

```

onFocus va acționa în urma focalizării asupra unui element cum ar fi: o fereastră, un frame, un element dintr-un formular.

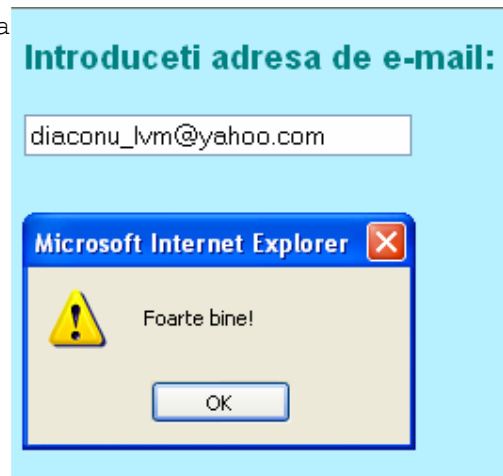
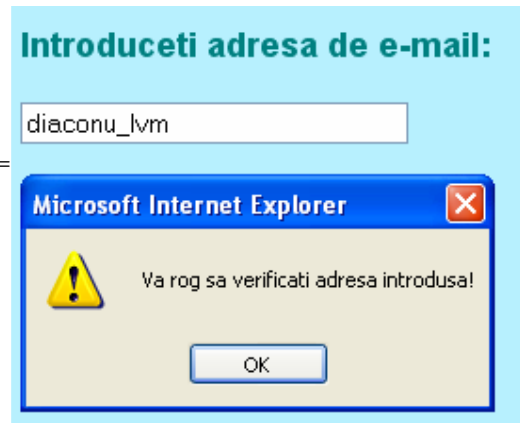
onFocus este event handler pentru obiectele: *Button, Checkbox, FileUpload, Layer, Password, Radio, Reset, Select, Submit, Text, Textarea* și *Window*.

Exemplu În exemplul de mai jos, am utilizat o funcție care verifică dacă în șirul de caractere introdus de la tastatură se află caracterul @, ce intră în componența unei adrese de e-mail validă.

```

<html>
<head>
  <title>
    Adresa email
  </title>
<script type="text/javascript" language=
"JavaScript">
this.formular.email.focus();
this.formular.email.select();
function verificare()
  {
    x = new String();
    x = this.formular.email.value;
    if (x.match("@"))
      alert("Foarte bine!");
    else
      alert("Va rog sa verificati adresa
introdusa!");
  }
</script>
<style type="text/css">
body
  {
    font-family: Geneva, Arial, Helvetica;
    font-size: 14px;
    font-weight: bolder;
    color: #008080;
    background-color: #B7F0FF;
    cursor: text;
  }
</style>
</head>
<body>
<H3>Introduceti adresa de e-mail:</H3>
<form name="formular">
<INPUT TYPE="text" VALUE="" NAME="email"
size="30"
onFocus="verificare(this.value)">
</form>
</body>
</html>

```



onReset va acționa în urma executării unui click de mouse pe un buton de tip *Reset*, al unui formular.

onReset este event handler pentru obiectul: *Form*.

Exemplu În exemplul de mai jos, am creat un formular mai complex, care verifică date introduse în toate câmpurile.

De cele mai multe ori în cadrul unui formular avem nevoie ca utilizatorul să introducă mai multe date, precum: numele de utilizator, numele lui, prenumele, adresa de e-mail și altele date de felul acesta. Odată introduse datele în formular, obligația noastră este să verificăm dacă datele introduse corespund solicitărilor din formular înainte ca formularul să fie transmis.

Funcțiile JavaScript pot realiza acest lucru, după cum vom observa în exemplul următor.

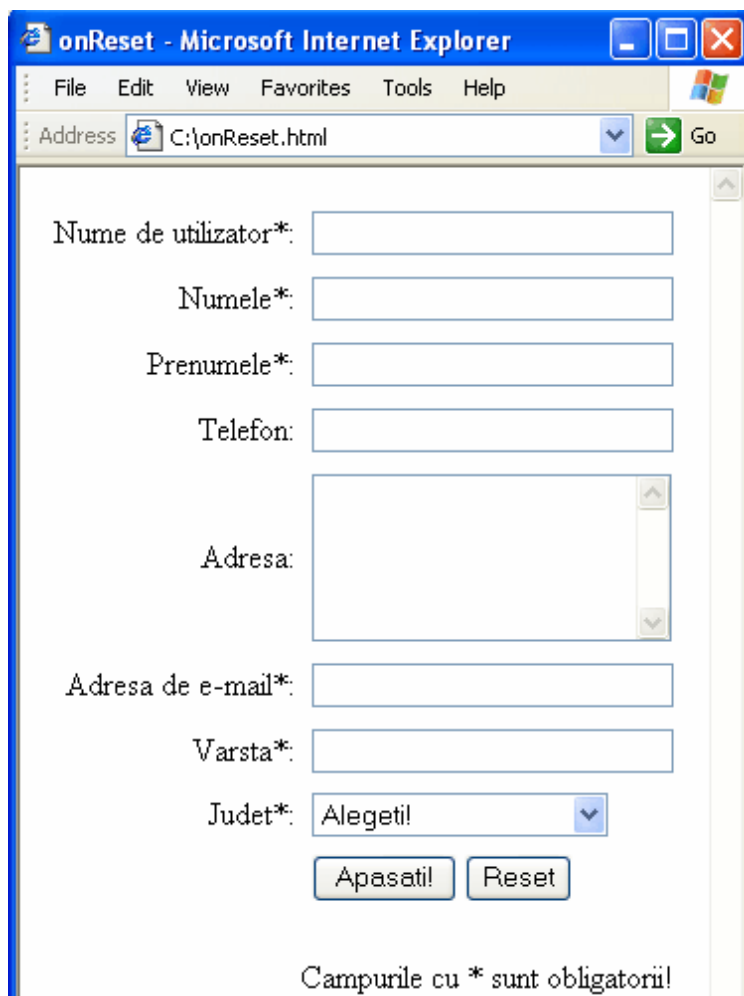
În cadrul unui document HTML am introdus un formular care la apăsarea butonului « Apasati ! » va verifica fiecare câmp din formnular prin apelarea funcțiilor JavaScript (verificare, VerificareEmail, VerificareaDatelor), poziționate în antetul paginii.

De exemplu, am creat o funcție ce permite atenționarea utilizatorilor în cazul în care au lăsat anumite câmpuri necompletate. Una dintre acestea este apelată de către:

```
VerificareFormular.VerifDistinct("utilizator","cerinta","Introduceți numele de utilizator!");
```

În codul sursă al formularului, am introdus nume sugestive atât pentru variabile, cât și pentru funcții și mai multe rânduri cu scurte comentarii, utile pentru înțelegerea corectă a scriptului.

Butonul **Reset** are ca scop golirea câmpurilor completate greșit de către utilizator.



The screenshot shows a Microsoft Internet Explorer browser window with the title "onReset - Microsoft Internet Explorer". The address bar displays "C:\onReset.html". The main content area contains a form with the following fields and controls:

- Nume de utilizator*:
- Numele*:
- Prenumele*:
- Telefon:
- Adresa:
- Adresa de e-mail*:
- Varsta*:
- Judet*:

At the bottom of the form, there are two buttons: "Apasati!" and "Reset". Below the buttons, a note reads: "Campurile cu * sunt obligatorii!"

```

<html>
<head>
  <title>onReset</title>
<script language="JavaScript" type="text/javascript">
function verificare (NumeFormular)
  {
    this.ValFormTrimis=document.forms[NumeFormular];
    if(!this.ValFormTrimis)
//daca nu ati introdus numele formularului, va aparea un mesaj...
    {
      alert("BUG: Nu ati introdus numele obiectului Form: "+NumeFormular);
      return;
    }
    if(this.ValFormTrimis.onsubmit)
    {
      this.ValFormTrimis.onsubmit=null;
    }
    this.ValFormTrimis.onsubmit=TrimisFormular;
    this.VerifDistinct = VerifVal;
    this.ValFormTrimis.VerifPlus = "";
    this.CL = Curata;
  }
function Curata ()
//functia care curata campurile formularului
  {
    for(var i=0;i < this.ValFormTrimis.elements.length;i++)
    {
      this.ValFormTrimis.elements[i].SetValidare = null;
    }
  }
function TrimisFormular ()
  {
    for(var i=0;i < this.elements.length;i++)
    {
      if(this.elements[i].SetValidare && !this.elements[i]. SetValidare.
        valid())
      {
        return false;
      }
    }
    if(this.VerifPlus)
    {
      xS =" var ret = "+this.VerifPlus+"()";
      eval(xS);
      if(!ret) return ret; }
    return true;
  }
function VerifVal (xN,xD,MesEroare)
//functia care verifica valorile introduse in formular
//valorile introduse sunt transmise prin variabilele xN,xD,MesEroare
  {
    if(!this.ValFormTrimis)
    {
      alert("BUG: Obiectul Form nu este setat corect!");
      return;
    }
    var xO = this.ValFormTrimis[xN];
    if(!xO)

```



```

//daca nu ati scris numele unui element din formular, cum ar fi: numele, prenumele ...
    {
        alert("BUG: Nu ati introdus numele obiectului denumit: "+xN);
        return;
    }
    if(!xO.SetValidare)
        {
            xO.SetValidare = new SetValidare(xO);
        }
    xO.SetValidare.add(xD,MesEroare);
}
function VerifDes(xi,desc,xEroare)
{
    this.desc=desc;
    this.xEroare=xEroare;
    this.xO = xi;
    this.valid=vdesc_valid;
}
function vdesc_valid()
{
    if(!VerificareaDatelor(this.desc,this.xO,this.xEroare))
        { this.xO.focus();
          return false;
        }
    return true;
}
function SetValidare(xi)
{
    this.VectorS=new Array();
    this.add= ADV;
    this.valid= VerifVectorS;
    this.xO = xi;
}
function ADV(desc,xEroare)
{
    this.VectorS[this.VectorS.length]= new VerifDes(this.xO,desc,xEroare);
}
function VerifVectorS()
{
    for(var i=0;i<this.VectorS.length;i++)
        {
            if(!this.VectorS[i].valid())
                { return false; }
        }
    return true;
}
function VerificareEmail(email)
{
//functie de verificare a câmpului pentru e-mail
    if(email.length <= 0)
        {
            return true;
        }
    var Impartit = email.match("^(.+)@(.+)$");
    if(Impartit == null)
        return false;
    if(Impartit[1] != null )
        { var xU=/^\\"?[\w-\.\.]*\\"?$/;

```

```

//variabila pentru verificarea numelui de utilizator din adresa de e-mail
    if(Impartit[1].match(xU) == null)
        return false;
    }
    if(Impartit[2] != null)
    { var xDomeniu=/^[\\w-\\.]*\\.[A-Za-z]{2,4}$/;
//variabila pentru verificarea numelui de domeniu din adresa de e-mail
    if(Impartit[2].match(xDomeniu) == null)
    {
        var xIP =/^[\\d{1,3}\\.[\\d{1,3}\\.[\\d{1,3}\\.[\\d{1,3}\\]}$/;
        if(Impartit[2].match(xIP) == null)
            return false;
        }
        return true;
    }
return false;
}
function VerificareaDatelor (VValid,ValoareOb,SirEroare)
//functie pentru verificarea datelor
{
    var vp = VValid.search("=");
    var comanda = "";
    var ValoareComanda = "";
    if(vp >= 0)
    { comanda = VValid.substring(0,vp);
      ValoareComanda = VValid.substr(vp+1); }
    else
    { comanda = VValid; }
    switch(comanda)
    {
        case "cerinta":
//verifica daca a fost introdusa informatia ceruta
        {
            if(eval(ValoareOb.value.length) == 0)
            { if(!SirEroare || SirEroare.length ==0)
                { SirEroare = ValoareOb.name + " : Introduceti informatia
                  ceruta!"; }
                alert(SirEroare);
                return false; }
            break;
        }
        case "LungimeMax":
//verifica daca se respecta lungimea maxima impusa pentru sirul de caractere
        { if(eval(ValoareOb.value.length) > eval(ValoareComanda))
            { if(!SirEroare || SirEroare.length ==0)
                { SirEroare = ValoareOb.name + " : "+ValoareComanda+"
                  characters maximum "; }
                alert(SirEroare + "\n[Lungimea curenta= " +
                  ValoareOb.value.length + " ]");
                return false; }
            break; }
        case "LungimeMin":
//verifica daca se respecta lungimea minima impusa pentru sirul de caractere
        {
            if(eval(ValoareOb.value.length) < eval(ValoareComanda))
            {if(!SirEroare || SirEroare.length ==0)
                {SirEroare = ValoareOb.name + " : " + ValoareComanda + " caractere
                  minim "; }
            }
        }
    }
}

```

```

        alert(SirEroare + "\n[Lungimea curenta= " +
            ValoareOb.value.length + " ]");
        return false; }
        break;
    }
    case "numeric":
//verifica daca au fost introduse cifre
    {
        var PozitieCaracter = ValoareOb.value.search("[^0-9]");
        if(ValoareOb.value.length > 0 && PozitieCaracter >= 0)
            { if(!SirEroare || SirEroare.length ==0)
                {SirEroare=ValoareOb.name+": Sunt permise numai cifre!"; }
                alert(SirEroare + "\n [Eroare caracter la pozitia: " +
                    eval(PozitieCaracter+1)+"]");
                return false; }
            break;
        }
    case "Alfabet":
//verifica daca au fost introduse caractere din alfabet
    {
        var PozitieCaracter = ValoareOb.value.search("[^A-Za-z]");
        if(ValoareOb.value.length > 0 && PozitieCaracter >= 0)
            { if(!SirEroare || SirEroare.length ==0)
                { SirEroare = ValoareOb.name+": Sunt permise numai caractere
                    din alfabet "; }
                alert(SirEroare + "\n [Eroare caracter la pozitia: " +
                    eval(PozitieCaracter+1)+"]");
                return false; }
            break;
        }
    case "CazUtilizator":
//verifica daca a fost introdus corect numele de utilizator
    {
        var PozitieCaracter = ValoareOb.value.search("[^A-Za-z0-9\-\_]");
        if(ValoareOb.value.length > 0 && PozitieCaracter >= 0)
            { if(!SirEroare || SirEroare.length ==0)
                { SirEroare = ValoareOb.name+": Caracterele permise sunt: A-
                    Z,a-z,0-9,- si _"; }
                alert(SirEroare + "\n [Eroare caracter la pozitia: " +
                    eval(PozitieCaracter+1)+"]");
                return false; }
            break;
        }
    case "email":
//verifica daca a fost completat corect campul pentru e-mail
    {
        if(!VerificareEmail(ValoareOb.value))
            { if(!SirEroare || SirEroare.length ==0)
                {SirEroare = ValoareOb.name+": Introduceti o adresa de email
                    corecta!"; }
                alert(SirEroare);
                return false; }
            break;
        }
    case "MaiMareDecat":
//verifica daca a fost introdusa varsta corect
    {
        var ValoareComanda=7;

```

```

        if(isNaN(ValoareOb.value))
        { alert(ValoareOb.name+": Ar trebui sa fie un numar!");
          return false; }
        if(eval(ValoareOb.value) <= eval(ValoareComanda))
        { if(!SirEroare || SirEroare.length ==0)
          { SirEroare = ValoareOb.name + " : Valoarea ar trebui sa fie
            mai mare decat: "+ ValoareComanda; }
          alert(SirEroare);
          return false; }
          break;
        }
    case "SelectatNu":
//verifica daca a fost selectata o optiune pentru judet
    {
        if(ValoareOb.selectedIndex == null)
        {
            alert("BUG: Nu a fost selectat!");
            return false;
        }
        if(ValoareOb.selectedIndex == eval(ValoareComanda))
        { if(!SirEroare || SirEroare.length ==0)
          {
              SirEroare = ValoareOb.name+": Selectati o optiune!";
          }
          alert(SirEroare);
          return false; }
          break; }
    }
    return true;
}
</script>
</head>
<body>
<form action="" name="formular" >
<table cellspacing="3" cellpadding="3" border="0" align="center">
<tr>
    <td align="right">Nume de utilizator*:</td>
    <td><input type="text" name="utilizator" size="26"></td></tr>
<tr>
    <td align="right">Numele*:</td>
    <td><input type="text" name="numele" size="26"></td>
</tr>
<tr>
    <td align="right">Prenumele*:</td>
    <td><input type="text" name="prenumele" size="26"></td>
</tr>
<tr>
    <td align="right">Telefon:</td>
    <td><input type="text" name="telefon" size="26"></td>
</tr>
<tr>
    <td align="right">Adresa:</td>
    <td><textarea cols="20" rows="5" name="adresa"></textarea></td>
</tr>
<tr>
    <td align="right">Adresa de e-mail*:</td>
    <td><input type="text" name="Email" size="26"></td>
</tr>
<tr>

```

```

        <td align="right">Varsta*:</td>
        <td><input type="text" name="Varsta" size="26"></td>
</tr>
<tr>
    <td align="right" size="26">Judet*:</td>
    <td>
        <select id="Judet" name="Judet">
            <option value="" selected>Alegeti!
                <option VALUE="AB">ALBA</option>
                <option VALUE="AR">ARAD</option>
                <option VALUE="AG">ARGES</option>
                <option VALUE="BC">BACAU</option>
                <option VALUE="BH">BIHOR</option>
                <option VALUE="BN">BISTRITA-NASAUD</option>
                <option VALUE="BT">BOTOSANI</option>
                <option VALUE="BR">BRAILA</option>
                <option VALUE="BV">BRASOV</option>
                <option VALUE="B">BUCURESTI</option>
                <option VALUE="BZ">BUZAU</option>
                <option VALUE="CL">CALARASI</option>
                <option VALUE="CS">CARAS-SEVERIN</option>
                <option VALUE="CJ">CLUJ</option>
                <option VALUE="CT">CONSTANTA</option>
                <option VALUE="CV">COVASNA</option>
                <option VALUE="DB">DAMBOVITA</option>
                <option VALUE="DJ">DOLJ</option>
                <option VALUE="GL">GALATI</option>
                <option VALUE="GR">GIURGIU</option>
                <option VALUE="GJ">GORJ</option>
                <option VALUE="HR">HARGHITA</option>
                <option VALUE="HD">HUNEDOARA</option>
                <option VALUE="IL">IALOMITA</option>
                <option VALUE="IS">IASI</option>
                <option VALUE="IF">ILFOV</option>
                <option VALUE="MM">MARAMURES</option>
                <option VALUE="MH">MEHEDINTI</option>
                <option VALUE="MS">MURES</option>
                <option VALUE="NT">NEAMT</option>
                <option VALUE="OT">OLT</option>
                <option VALUE="PH">PRAHOVA</option>
                <option VALUE="SJ">SALAJ</option>
                <option VALUE="SM">SATU-MARE</option>
                <option VALUE="SB">SIBIU</option>
                <option VALUE="SV">SUCEAVA</option>
                <option VALUE="TR">TELEORMAN</option>
                <option VALUE="TM">TIMIS</option>
                <option VALUE="TL">TULCEA</option>
                <option VALUE="VL">VALCEA</option>
                <option VALUE="VS">VASLUI</option>
                <option VALUE="VN">VRANCEA</option>
            </select>
        </td>
</tr>
<tr>
    <td align="right">
        </td>
    <td><input type="submit" value="Apasati!">
        <input name="reset" type="reset" value="Reset">
        </td>
</tr>

```

```

</tr>
<tr>
  <td colspan="2">
    <br />
    <p align="right"> Campurile cu * sunt obligatorii!</p>
  </td>
</tr>
</table>
</form>
<script language="JavaScript" type="text/javascript">
  var VerificareFormular = new verificare("formular");
  VerificareFormular.VerifDistinct("utilizator","cerinta","Introduceti
    numele de utilizator!");
  VerificareFormular.VerifDistinct("utilizator","LungimeMax=10", "Lungimea
    maxima pentru numele de utilizator este 10");
  VerificareFormular.VerifDistinct("utilizator","LungimeMin=3");
  VerificareFormular.VerifDistinct("utilizator","CazUtilizator");
  VerificareFormular.VerifDistinct("numele","cerinta","Introduceti
    numele!");
  VerificareFormular.VerifDistinct("numele","LungimeMax=20", "Lungimea
    maxima pentru nume este 20");
  VerificareFormular.VerifDistinct("numele","LungimeMin=3");
  VerificareFormular.VerifDistinct("numele","Alfabet");
  VerificareFormular.VerifDistinct("prenumele","cerinta", "Introduceti
    prenumele!");
  VerificareFormular.VerifDistinct("prenumele","LungimeMax=30", "Lungimea
    maxima pentru prenume este 30");
  VerificareFormular.VerifDistinct("prenumele","LungimeMin=3");
  VerificareFormular.VerifDistinct("prenumele","Alfabet");
  VerificareFormular.VerifDistinct("telefon","LungimeMax=50", "Introduceti
    telefonul!");
  VerificareFormular.VerifDistinct("telefon","numeric");
  VerificareFormular.VerifDistinct("adresa","LungimeMax=50", "Introduceti
    adresa!");
  VerificareFormular.VerifDistinct("Email","LungimeMax=50");
  VerificareFormular.VerifDistinct("Email","cerinta");
  VerificareFormular.VerifDistinct("Email","email");
  VerificareFormular.VerifDistinct("Varsta","cerinta","Introduceti
    varsta!");
  VerificareFormular.VerifDistinct("Varsta","MaiMareDecat");
  VerificareFormular.VerifDistinct("Varsta","LungimeMax=3", "Lungimea maxima
    pentru varsta este 3");
  VerificareFormular.VerifDistinct("Varsta","LungimeMin=1");
  VerificareFormular.VerifDistinct("Judet","SelectatNu=0");
</script>
</body>
</html>

```

Mai jos, am descris pe scurt funcțiile care apar în scripturile din cadrul exemplului de mai sus și câteva dintre mesajele de eroare care apar, dacă nu sunt completate corect câmpurile formularului. De exemplu, dacă pentru numele de utilizator introducem mai puțin de 3 caractere, intrăm în contradicție cu condiția pusă în apelul de mai jos:

```
VerificareFormular.VerifDistinct("utilizator","LungimeMin=3");
```

Funcțiile din formular

În cadrul codului avem trei zone de cod:

1. zona de declarare a funcțiilor, în antetul paginii web (head)
2. zona în care este scris formularul cu HTML, în corpul paginii web (body)
3. zona de apel a funcțiilor, în cadrul unui script JavaScript, în corpul paginii web (body)

Ordinea de rulare a funcțiilor este dată de ordinea de apel a funcțiilor din zona 3. Prima funcție apelată este verificare prin `new verificare("formular")`.

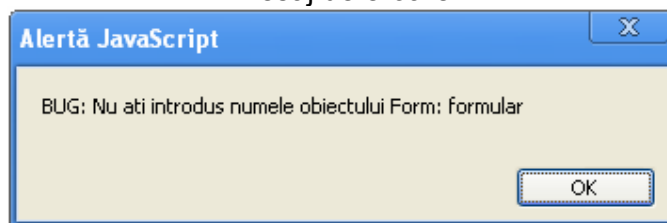
Funcția **verificare** are rolul de a verifica dacă formularul HTML are nume, în caz contrar afișând un mesaj de eroare (1). Dacă formularul are nume, se apelează funcția `TrimisFormular`. După verificări, se depune în `this.CL` rezultatul apelului funcției `Curata`

Nr. Descriere

1 Apelul:
`var VerificareFormular = new
verificare("formular");`

Cazul:
`!this.ValFormTrimis`

Mesaj de eroare



Funcția **Curata** are rolul de a curăța câmpurile formularului cu ajutorul unei structuri repetitive. Curățarea se realizează prin introducerea valorii `null` în formular ce are ca scop pregătirea formularului pentru noi date.

Funcția **TrimisFormular** are rolul de a transmite formularul completat după ce este validat de funcția `SetValidare` apelată din interiorul instrucțiunii de decizie `if`.

Funcția **VerifVal** verifică valorile introduse în formular. Valorile introduse sunt transmise prin variabilele `xN`, `xD`, `MesEroare`. Dacă nu ați introdus numele unui element din formular, care trebuie să fie introdus obligatoriu (are `*`) cum ar fi numele, prenumele, adresa de e-mail, vârsta și județul, va apare mesajul de eroare:

"BUG: Nu ati introdus numele obiectului denumit:"

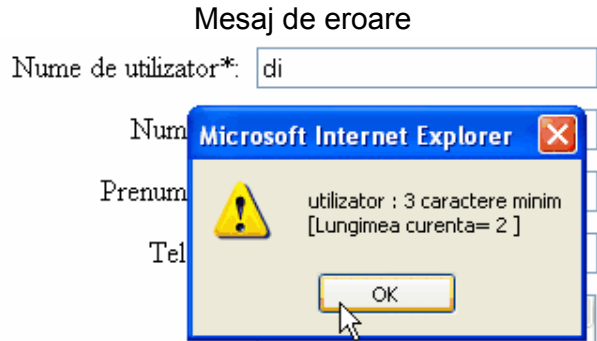
VerificareEmail este o funcție de verificare a câmpului pentru e-mail. Utilizatorii trebuie să introducă pentru adresa de e-mail numele de utilizator, la care se adaugă `@` și numele de domeniu (de exemplu `yahoo.com`).

Numele de utilizator poate fi orice combinație de litere (a-z, A-Z), cifre, minus(-) și underscore (_) care trebuie să înceapă cu o literă.

VerificareaDatelor este o funcție mai amplă pentru verificarea datelor introduse în formular ce conține o instrucțiune de decizie multiplă `switch` ce conține cazurile:

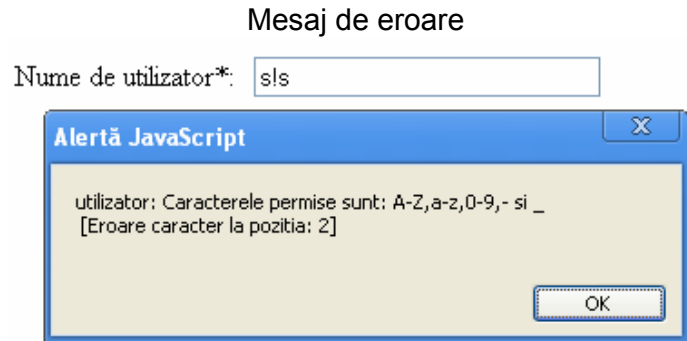
- **cerință** – verifică dacă a fost introdusă informația cerută
- **LungimeMax** – verifică dacă se respectă lungimea maximă impusă pentru șirul de caractere, adică 10 pentru numele de utilizator, 20 pentru nume, 30 pentru prenume, specificate în scriptul final.
- **LungimeMin** – verifică dacă se respectă lungimea minimă impusă pentru șirul de caractere, adică 3 pentru numele de utilizator, 3 pentru nume, 1 pentru vârstă, specificate în scriptul final, așa cum puteți observa în mesajul de eroare 2.

Nr. Descriere
2 VerificareFormular.
VerifDistinct("utilizator",
"LungimeMin=3");



- **numeric** – verifică dacă au fost introduse cifre în câmpul telefon, altfel afișează un mesaj de eroare
- **Alfabet** – verifică dacă au fost introduse caractere din alfabet pentru câmpurile nume și prenume
- **CazUtilizator** – verifică dacă a fost introdus corect numele de utilizator. Mesajul de eroare ne specifică poziția caracterului unde am introdus primul caracter greșit și caracterele care sunt permise pentru numele utilizatorului, ca în mesajul de eroare 3.

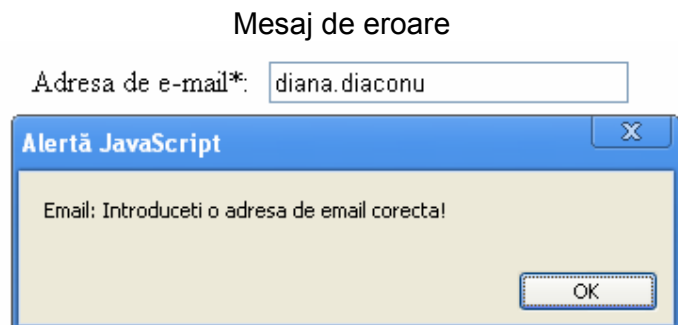
Nr. Descriere
3 alert(SirEroare + "\n
[Eroare caracter la pozitia:
" + eval(PozitieCaracter
+1)+"]"); s



- **email** – verifică dacă a fost completat corect câmpul pentru e-mail, adică dacă lungimea adresei de e-mail este egală cu zero sau a fost introdusă o adresă greșită. În cazul acesta afișează mesajul de eroare 4. Mesajul de eroare este reținut în variabila SirEroare și utilizat de metoda alert().

Nr. Descriere
4 SirEroare =
ValoareOb.name+":
Introduceti o adresa de
email corecta!";

alert(SirEroare);



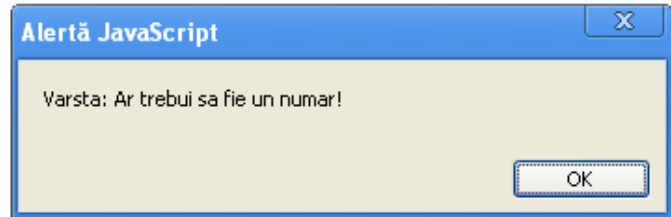
- **MaiMareDecat** – verifică dacă a fost introdusă vârsta corect cu ajutorul funcției isNaN() cunoscută sub numele "is Not-a-Number". În cazul în care utilizatorii introduc altceva decât un număr, se va afișa mesajul de eroare, 5, iar dacă introduc un număr mai mic decât 7, se va afișa mesajul de eroare 6.

Nr. Descriere

```
5 alert(ValoareOb.name+": Ar  
   trebui sa fie un numar!");
```

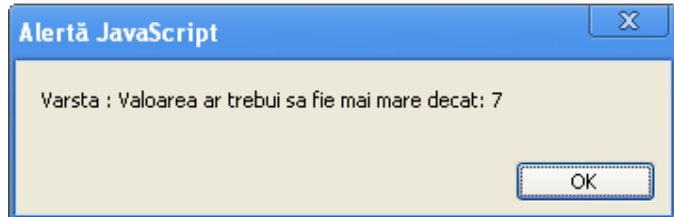
Mesaj de eroare

Varsta*:



```
6 SirEroare = ValoareOb.name +  
   " : Valoarea ar trebui sa  
   fie mai mare decat: "+  
   ValoareComanda;  
  
   alert(SirEroare);
```

Varsta*:



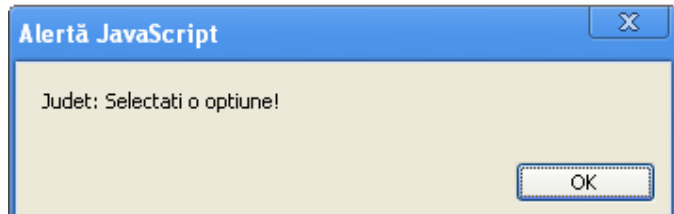
- **SelectatNu** – verifică dacă a fost selectată o opțiune pentru județ. Dacă nu a fost selectată nici o opțiune se va afișa mesajul de eroare 7.

Nr. Descriere

```
7 if(!SirEroare ||  
   SirEroare.length ==0)  
   {SirEroare =  
     ValoareOb.name+":  
     Selectati o optiune!";  
   }  
   alert(SirEroare);
```

Mesaj de eroare

Judet*:



onSelect va acționa în urma selectării uneia dintre opțiunile dintr-un câmp de tip *text* sau *textarea* dintr-un formular.

onSelect este event handler pentru obiectele: *Text*, *Textarea*.

Exemplu În următorul exemplu, am utilizat o funcție care afișează un mesaj, în cazul în care selectăm caracterele scrise într-un câmp de tip text.

```
<html>  
<head>  
  <title>onSelect</title>  
</head>  
<script type="text/javascript" language=  
"JavaScript">  
function f()  
{  
  alert("Felicitari! \nAti selectat  
  textul.");  
}  
</script>  
<body>  
<H3>Selectati textul de mai jos!</H3>  
<form name="formular">  
<input type="text" name="data"  
  value="Textul de mai jos :)"  
  size=20 onSelect='f()'>
```



```

</form>
</body>
</html>

```

onSubmit este utilizat pentru a executa un cod JavaScript, atunci când utilizatorul dă click pe butonul Submit al unui formular. Se poate anula trimiterea informațiilor din formular către server, dacă se pune *return false* ca ultimă instrucțiune în secvența de tratare a evenimentului.

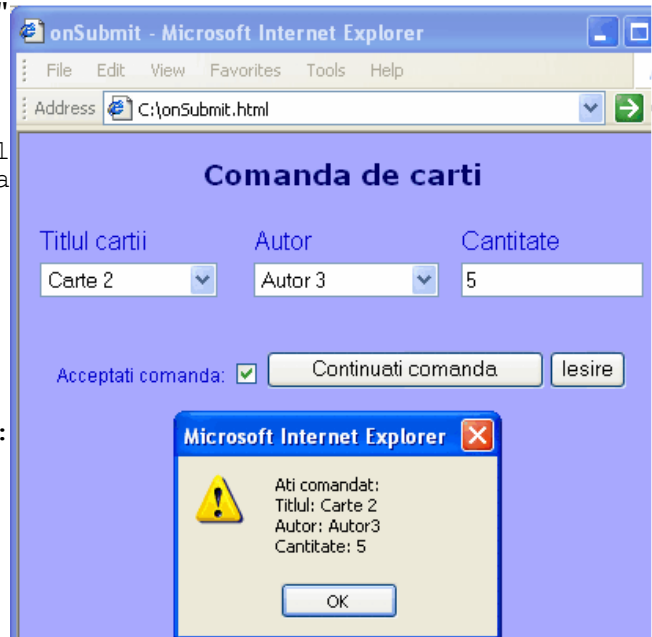
onSubmit este event handler pentru obiectul: *Form*.

Exemplu În exemplul de mai jos, am utilizat o funcție care verifică dacă este selectat un buton de tip checkbox. În cazul în care este selectat, formularul permite continuarea comenzii, altfel apare un mesaj de atenționare. Butonul *Continuati comanda* este de tip submit, iar în momentul în care este acționat, formularul declanșează *onSubmit="return verifica(this)"*. În final, *onSubmit* apelează funcția *verifica*.

```

<html>
<head>
  <title>onSubmit</title>
<style>
body
  { font-family: Geneva, Arial;
    font-size: 12px;
    color: #0000CC;
    background-color: #A8A8FF;
    text-align: center; }
.titlul
  { font-family: Verdana, Arial;
    font-size: 18px;
    font-weight: bold;
    color: #000066;
    text-align: center; }
</style>
<script type="text/javascript"
language="JavaScript">
function verifica(f)
  {
    if(f.DeAcord.checked == false )
      {alert('Selectati butonul
        inainte de a
        continua!');
        return false;
      }
    else
      {
        var tit=f.carte.value;
        var aut=f.autor.value;
        var can=f.cantitate.value;
        alert("Ati comandat:\nTitlul:
          "+tit+"\nAutor:
          "+aut+"\nCantitate: "+can);
        return true;
      }
  }
</script>
</head>
<body>
<p align="center" class="titlul">
Comanda de carti</p>
<form action="/pagina.html"
method="POST" onSubmit="return
Prof. Diaconu Diana

```



```

verifica(this)">
<table width="400" border="0">
  <tr>
    <td>Titlul cartii</td>
    <td>Autor</td>
    <td>Cantitate</td></tr>
  <tr>
    <td><select name="carte">
      <option selected="">Alegeti
        cartea!</option>
      <option value="Carte 1">Carte
        1</option>
      <option value="Carte 2">Carte
        2</option>
      <option value="Carte 3">Carte
        3</option>
    </select> &nbsp;</td>
    <td><select name="autor">
      <option selected="">Alegetii
        autorul!</option>
      <option value="Autor1">Autor
        1</option>
      <option value="Autor2">Autor
        2</option>
      <option value="Autor3">Autor
        3</option>
    </select></td>
    <td><input
      name="cantitate" size="15" value=""
      type="text">
    </td></tr>
</table>
><br /><br />
  Acceptati comanda:
  <input type="checkbox" value="0"
name=" DeAcord ">
  <input type="submit"
value="Continuati comanda">
  <input type="button" value="Iesire"
onClick="document.location.href='/in
dex.html';">
</form>
</body>
</html>

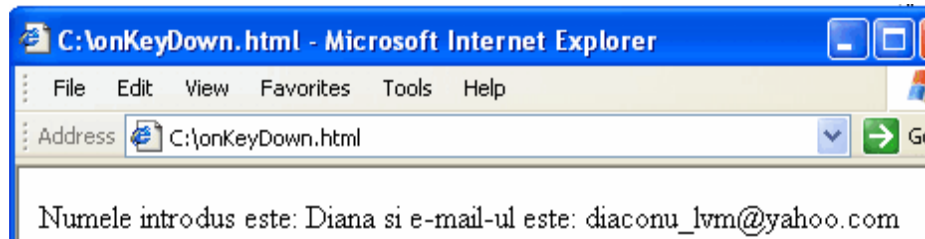
```

Tratarea evenimentelor de către tastatură

onKeyDown va acționa în urma apăsării unei taste de către utilizator. Evenimentul **KeyDown** îl precede pe **KeyPress**.

onKeyDown este event handler pentru obiectele: *Document*, *Image*, *Link* și *Textarea*.

Exemplu În acest exemplu, avem un formular în care introducem numele și adresa de e-mail, iar la apăsarea unei taste de către utilizator, se afișează un mesaj cu datele introduse. În cazul în care nu s-a introdus nimic în câmpurile formularului, ne atenționează printr-un mesaj în legătură cu această lipsă.



```

<html>
<head>
  <title>onKeyDown</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
body,td,th
  { color: #006600;
    font-size: 14px; }
body
  { background-color: #AEFFAE;
    font-size: 18px;
    font-weight: bold;
    text-align: center; }
.interior
  { color: #006633;
    background-color: #E1E1E1; }
</style>
</head>
<body>
<script language="JavaScript">
function f()
  {
    a=document.form1.t1.value;
    b=document.form1.t2.value;
    if ((a==null)|| (a.length==0)) && ((b==null)|| (b.length==0))
      document.writeln("Nu ati introdus numele, si e-mail-ul!");
    else
      document.writeln("Numele introdus este: ", a, " si e-mail-ul este: ",b);
  }
</script>
<form name="form1">
Formular<br /><br />
<table width="417" border="1">
  <tr>

```

```

        <td width="150">Numele si prenumele:</td>
        <td width="251"><input name="t1" type="text" size=40 maxlength=40
            class="interior"></td> </tr>
    <tr>
        <td>E-mail: </td>
        <td><input type="text" name="t2" size=40 maxlength=40
            class="interior"></td> </tr>
    <tr>
        <td colspan="2"><div align="center">
            <input type="Button" name="buton" value="Apasati!"
                onKeyDown="f(this.form)" >
        </div></td></tr>
</table>
</form>
</body>
</html>

```

onKeyPress va acționa în urma apăsării scurte sau apăsării îndelungate a unei taste de către utilizator.

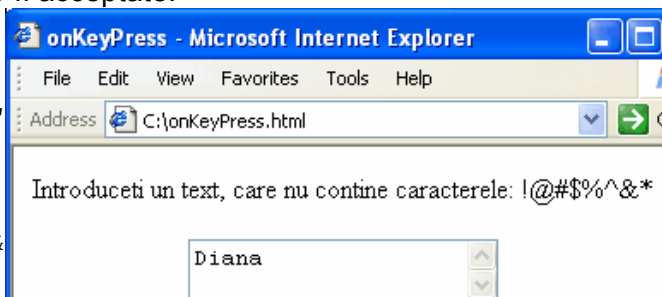
onKeyPress este event handler pentru obiectele: *Document*, *Image*, *Link* și *Textarea*.

Exemplu În acest exemplu, avem un formular ce permite introducerea unor caractere diferite de caracterele speciale, cum ar fi !@#\$%^&*. Dacă încercăm să introducem aceste caractere, ele pur și simplu nu vor fi acceptate.

```

<html>
<head>
<title>onKeyPress</title>
<script type="text/javascript"
language="JavaScript">
function f()
{
    if ((event.keyCode > 32 &&
        event.keyCode < 48) ||
        (event.keyCode > 57 &&
        event.keyCode < 65) ||
        (event.keyCode > 90 &&
        event.keyCode < 97))
        event.returnValue = false; }
</script>
</head>
<body>
<center>
<form onSubmit="return false;">
Introduceti un text, care nu
    contine caracterele: !@#$%^&*
    <br>
<textarea rows=2 cols=20
    name=comments onKeyPress="f()" >
    </textarea>
</form>
</center>
</body>
</html>

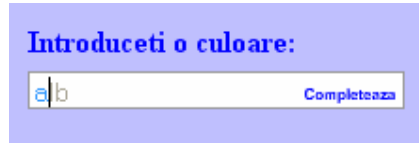
```



onKeyUp va acționa în urma eliberării unei taste de către utilizator, după ce aceasta a fost apăsată.

onKeyUp este event handler pentru obiectele: *Document*, *Image*, *Link* și *Textarea*.

Exemplu În acest exemplu, avem un formular ce permite introducerea unor litere din componența unor culori și sugerează completarea cuvântului, pe baza unui vector ce conține culorile respective. În partea dreaptă, avem o legătură realizată pe cuvântul *Completeaza*, ce permite completarea numelui culorii. De exemplu, dacă se tastează litera a, formularul ne sugerează cuvântul *alb*. Pentru a nu mai scrie toate literele, dăm click pe cuvântul *completează* și va completa automat formularul cu cuvântul *alb*.



```

<html>
<head>
  <title>onKeyUp</title>
  <script type="text/javascript" >
culori = new Array("alb", "albastru", "argintiu", "caramiziu", "crem",
  "galben", "gri", "maron", "mov", "negru", "pastel",
  "portocaliu", "rosu", "roz", "verde", "violet", "visiniu" );
var s1 = "";
var sd = "";
function OCuloare()
{
  var intrare = document.forms['formular'].culoare.value;
  var lungime = intrare.length;
  sd = ""; s1 = "";
  if (intrare.length)
  {
    for (xc in culori)
    {
      if (culori[xc].substr(0, lungime).toLowerCase() ==
        intrare.toLowerCase())
      {
        sd = intrare + culori[xc].substr(lungime);
        s1 = culori[xc];
        break;
      }
    }
  }
  document.forms['formular'].cul.value = sd;
  if (!s1.length || intrare == sd)
    document.getElementById('s1b').style.display = "none";
  else
    document.getElementById('s1b').style.display = "block";
}
function SetareCuloare()
{
  document.forms['formular'].culoare.value = s1;
  ascund();
}
function ascund()
{
  document.forms['formular'].cul.value = "";
  document.getElementById('s1b').style.display = "none";
}
</script>
<style type="text/css">

```

```

body
{
  background-color: #BFBFFF;
}
body,td,th
{
  color: #0000FF;
  font-weight: bold;
}
</style>
</head>
<body>
<div style="width: 202px; margin: 100px auto 0 auto;">
<form name="formular">
  <div>Introduceti o culoare:</div>
  <div style="position: relative; margin: 5px 0 5px 0; height: 30px;">
  <div style="position: absolute; top: 0; left: 0; width: 200px; z-index:
    1;">
    <input type="text" name="cul" style="background-color: #ffffff; border:
      1px solid #aaaaaa; width: 200px; padding: 2px" disabled />
  </div>
  <div style="position: absolute; top: 0; left: 0; width: 200px; z-index:
    2;">
    <input autocomplete="off" type="text" name="culoare"
      style="background: none; color:#39f; border: 1px solid #999; width:
        200px; padding: 2px" onFocus="OCuloare()" onKeyUp="OCuloare()" />
  </div>
  <div id="slb" style="position: absolute; top: 5px; right: 5px; z-
    index: 3; display: none;">
    <a href="#" onClick="SetareCuloare()">
      
    </a>
  </div>
</div>
</form>
</div>
</body>
</html>

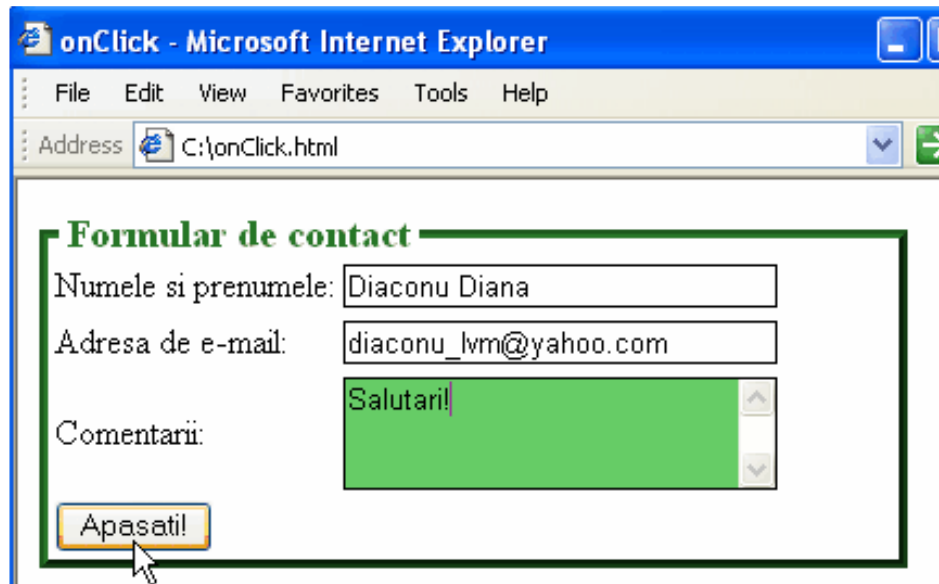
```

Tratarea evenimentelor de către mouse

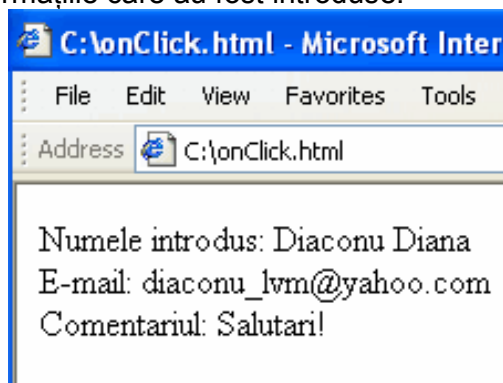
onClick va acționa în urma unui click stânga de mouse dat de către utilizator pe un obiect specificat dintr-un formular.

onClick este event handler pentru obiectele: *Button*, *Checkbox*, *Document*, *Link*, *Radio*, *Reset* și *Submit*.

Exemplu În acest exemplu, avem un formular de contact în care se pot introduce numele, adresa de e-mail și comentarii. În momentul în care vrem să introducem ceva în formular, se va evidenția zona respectivă. Pentru evidențiere am utilizat *onBlur* și *onFocus*, deoarece atunci când introducem ceva într-un câmp, acționează *onFocus*, iar când ne îndepărtăm, acționează *onBlur*.



După ce se introduc date în formularul de contact, se dă click pe butonul *Apasati!* și se vor afișa pe ecran informațiile care au fost introduse.



```

<html>
<head>
  <title>onClick</title>
  <style type="text/css">
    fieldset
    {
      width: 400px;
      border: medium outset #277627; }
    legend
    {
      color: #277627;
      font-weight: bolder;
      font-size: 18px; }
    .textInput,textarea
    {
      width: 200px;
      font-family: arial;
      background-color: #FFFFFF;
      border: 1px solid #000; }
    .NEvid
    {
      width: 200px;
      background-color: #66CC66;
      color: #000;
      border: 1px solid #000; }
  </style>
  <script type="text/javascript" language="JavaScript">
    var IntrareCurenta = false;
    var NumeStilIntrCurenta = false;
    function f()

```



```

{
    var n=document.formular.nume.value;
    var e=document.formular.email.value;
    var c=document.formular.comentariu.value;
    document.writeln("Numele      introdus:      ",      n,"<br>E-mail:      ",
        e,"<br>Comentariul: ", c);
}
function IntrEvidentiata()
{
    if(IntrareCurenta)
    {
        IntrareCurenta.className = NumeStilIntrCurenta;
    }
    NumeStilIntrCurenta = this.className;
    this.className = 'NEvid';
    IntrareCurenta = this;
}
function IntrBlur()
{
    this.className = NumeStilIntrCurenta;
}
function Evidentiat()
{
    var etichete = ['INPUT','TEXTAREA'];
    for(i=0;i<etichete.length;i++){
        var intrari = document.getElementsByTagName(etichete[i]);
        for(var no=0;no<intrari.length;no++)
        {
            if(intrari[no].className && intrari[no].className=='x')continue;

            if(intrari[no].tagName.toLowerCase()=='textarea' || (intrari[no].t
agName.      toLowerCase()=='input'      &&      intrari[no].      type.
toLowerCase() == 'text'))
            {
                intrari[no].onfocus = IntrEvidentiata;
                intrari[no].onblur = IntrBlur;
            }
        }
    }
}
</script>
</head>
<body>
<form method="post" action="#" name="formular">
<fieldset>
<legend>Formular de contact </legend>
<table>
<tr>
<td><label for="nume">Numele si prenumele: </label></td>
<td><input class="textInput" type="text" name="nume" id="nume"></td>
</tr>
<tr>
<td><label for="email">Adresa de e-mail: </label></td>
<td><input class="textInput" type="text" name="email" id="email"></td>
</tr>
<tr>
<td><label for="comentariu">Comentarii:</label></td>
<td><textarea      id="comentariu"      name="comentariu"
rows="3"></textarea></td>
</tr>
<tr>

```

```

<td colspan="2">
  <input type="submit" onClick="f()" value="Apasati!">
</td>
</tr>
</table>
</fieldset>
</form>
<script type="text/javascript"> Evidentiat();
</script>
</body>
</html>

```

onDbIcClick este o acțiune ce va fi declanșată dacă utilizatorul efectuează dublu click stânga de mouse pe un obiect specificat dintr-un formular.

onDbIcClick este event handler pentru obiectele: *Document* și *Link*.

Exemplu În exemplul de mai jos, am utilizat o funcție care permite alegerea unui stil propriu de afișare a paginii. De exemplu, dacă dăm dublu click pe butonul *Stil 3*, conținutul paginii se va afișa cu textul alb pe fundal albastru, iar titlul paginii va fi: *Fundal albastru*.



```

<html>
<head>
  <title>onDbIcClick</title>
<script type="text/javascript" language="javascript">
function f(stil)
  {
    switch(stil)
    {
      case 1:
      {
        document.bgColor='yellow';
        document.title='Fundal galben';
        document.fgColor='#663300';}
      break;
      case 2:
      {
        document.bgColor='red';
        document.title='Fundal rosu';
        document.fgColor='blue';
      }
      break;
      case 3:
      {
        document.bgColor='blue';
        document.title='Fundal albastru';
        document.fgColor='white';
      }
      break;
      case 4:

```

```

    {
      document.bgColor='green';
      document.title='Fundal verde';
      document.fgColor='yellow';
    }
    break;
default: document.bgColor='white';
}}
</script>
</head>
<body>
<div align="center">
Dati dublu click pentru a alege un stil pentru pagina
<form>
  <input type="button" value="Stil 1" onDbclick="f(1)">
  <input type="button" value="Stil 2" onDbclick="f(2)">
  <input type="button" value="Stil 3" onDbclick="f(3)">
  <input type="button" value="Stil 4" onDbclick="f(4)">
</form>
</div>
</body>
</html>

```

onMouseDown este o acțiune ce va fi declanșată dacă utilizatorul apasă unul dintre butoanele mouse-ului.

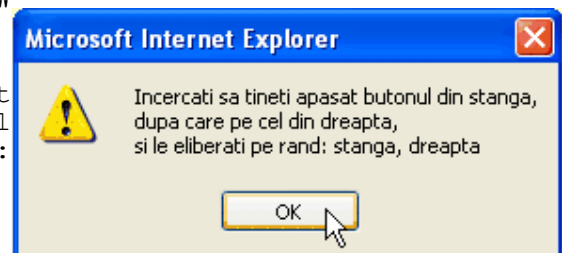
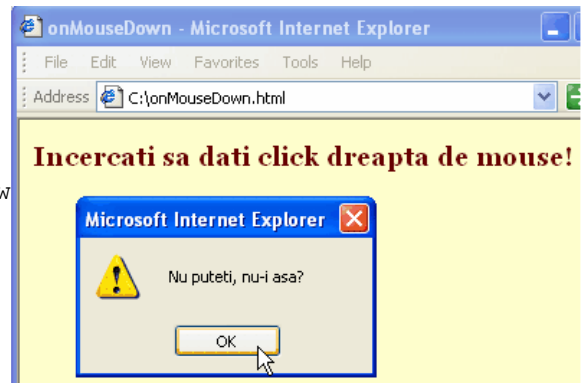
onMouseDown este event handler pentru obiectele: *Button*, *Document* și *Link*.

Exemplu În exemplul de mai jos, se utilizează o funcție care verifică dacă a fost apăsat butonul dreapta al mouse-ului. Dacă a fost apăsat, se vor afișa pe rând, două mesaje de atenționare.

```

<html>
<head>
  <title>onMouseDown</title>
<style type="text/css">
body
{
  font-family: Georgia, "Times New
Roman";
  font-size: 18px;
  font-weight: bold;
  color: #660000;
  background-color: #FFFFCC;
}
</style>
<script type="text/javascript"
language="javascript">
var mesaj1 = "Nu puteti, nu-i asa?";
var mesaj2="Incercati sa tineti apasat
butonul din stanga, \ndupa care pe cel
din dreapta, \nsi le eliberati pe rand:
stanga, dreapta";
function f()
{
  if (document.all)
  {
    if (event.button==2)
    {
      alert(mesaj1);
      alert(mesaj2);
      return false;
    }
  }
}
</script>

```



```

</head>
<body>
Incercati sa dati click dreapta de
mouse!
<BR>
<script type= "text/javascript"
language="javascript">
document.onmousedown=f;
</script>
</body>
</html>

```

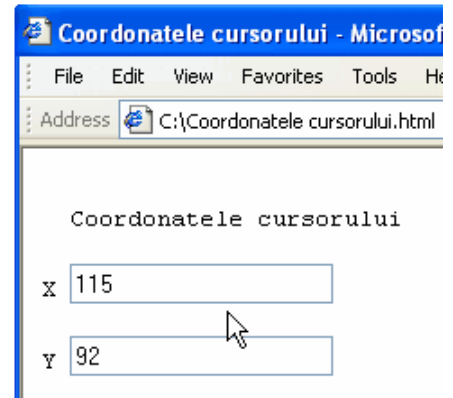
onMouseMove este o acțiune ce va fi declanșată de mișcarea cursorului mouse-ului.

Exemplu În exemplul de mai jos, am utilizat o funcție care permite afișarea coordonatelor cursorului unui mouse în mișcare.

```

<html>
<head>
<title>Coordonatele cursorului</title>
</head>
<body>
<pre>
<form name=formular>
Coordonatele cursorului<br /><br />
X <input type=text name=x value=""><br />
Y <input type=text name=y value="">
</form>
</pre>
<script type=text/javascript>
var InternetExplorer = document.all?true:false;
if (!InternetExplorer)
document.captureEvents(Event.mousemove);
document.onmousemove = pozitie;
function pozitie (poz)
{
var x;
var y;
if (!InternetExplorer)
{
x = poz.pageX;
y = poz.pageY;
}
if (InternetExplorer)
{
x = event.clientX +
document.body.scrollLeft;
y = event.clientY + document.body.scrollTop;
}
document.formular.x.value=x;
document.formular.y.value=y;
return true; }
</script>
</body>
</html>

```



onMouseOut este o acțiune ce va fi declanșată în momentul în care părăsim cu mouse-ul o zonă dintr-o hartă sau când ne îndepărtăm de textul ce conține legături către alte pagini. *onMouseOut* este event handler pentru obiectele: *Layer* și *Link*.

Exemplu În exemplul de mai jos, am realizat un tabel în care se schimbă culoarea celulelor la trecerea mouse-ului pe deasupra lor.



```

<html>
<head>
  <title>onMouseOut</title>
<style>
.citat
{ font-size: 14px;          font-style: italic;
  font-weight: normal;      color: #000066; }
.autor
{ color: #000066;          font-size: 16px;
  font-weight: bolder; }
.legatura
{ font-size: 12px;          font-family: Georgia, "Times New Roman",
Times;
  color: #000066;          font-weight: bolder;
  text-decoration: none; }
</style>
</head>
<body>
  <center>
<table border=0 cellpadding=1 cellspacing=1>
<tr>
<td
      onClick="window.location='index.html'"
      onMouseOver="bgColor='#AAAAFF'"
      onMouseOut="bgColor='#DDDDFF'"
      align="center" valign="center"><b><font color="#D5D5D5">
      <a href="index.html" class="legatura">Acasa</a></font></b></td>
<td width="1" align="center">|</td>
<td
      onClick="window.location='pagina2.html'"
      onMouseOver="bgColor='#AAAAFF'"
      onMouseOut="bgColor='#DDDDFF'"
      align="center" valign="center"><b><font color="#D5D5D5">
      <a href="pagina2.html" class="legatura">Pagina 2</a></font></b></td>
<td width="1" align="center">|</td>
<td
      onClick="window.location='pagina3.html'"
      onMouseOver="bgColor='#AAAAFF'"
      onMouseOut="bgColor='#DDDDFF'"
      align="center" valign="center"><b><font color="#D5D5D5">
      <a href="pagina3.html" class="legatura">Pagina 3</a></font></b></td>
</tr>
</table>
<p class="citat">Nici un lucru mare in lume nu a fost realizat fara
  pasiune.</p>
<p class="autor">Georg Wilhelm </p>
</center>

```

```
</body>
</html>
```

onMouseOver este o acțiune ce va fi declanșată în momentul în care cursorul mouse-ului pătrunde peste o zonă delimitată sau peste un obiect definit, venind din afara lor. *onMouseOver* este event handler pentru obiectele: *Layer* și *Link*.

Exemplu În exemplul de mai jos, se afișează un mesaj în momentul în care trecem cu mouse-ul pe deasupra imaginii de mai jos. Când ne îndepărtăm cu mouse-ul de imagine, mesajul dispare (va fi ascuns).



```
<html>
<head>
  <title>onMouseOver</title>
  <style type="text/css">
    .unu
    { position:absolute;          visibility:hidden;
      clip:rect(0 200 50 0);     width:200px;
      background-color:#C99AB5;  z-index:10;
      color: #5B3149; }
    .doi
    { cursor: hand; }
    .text
    { border:1px solid #5b3149; }
  </style>
  <script type="text/javascript" language="javascript">
function f(a1,event,txtul)
  {
    if (document.all&&document.readyState=="complete")
    {
      document.all.x.innerHTML='<marquee class="text">'+txtul+'</marquee>'

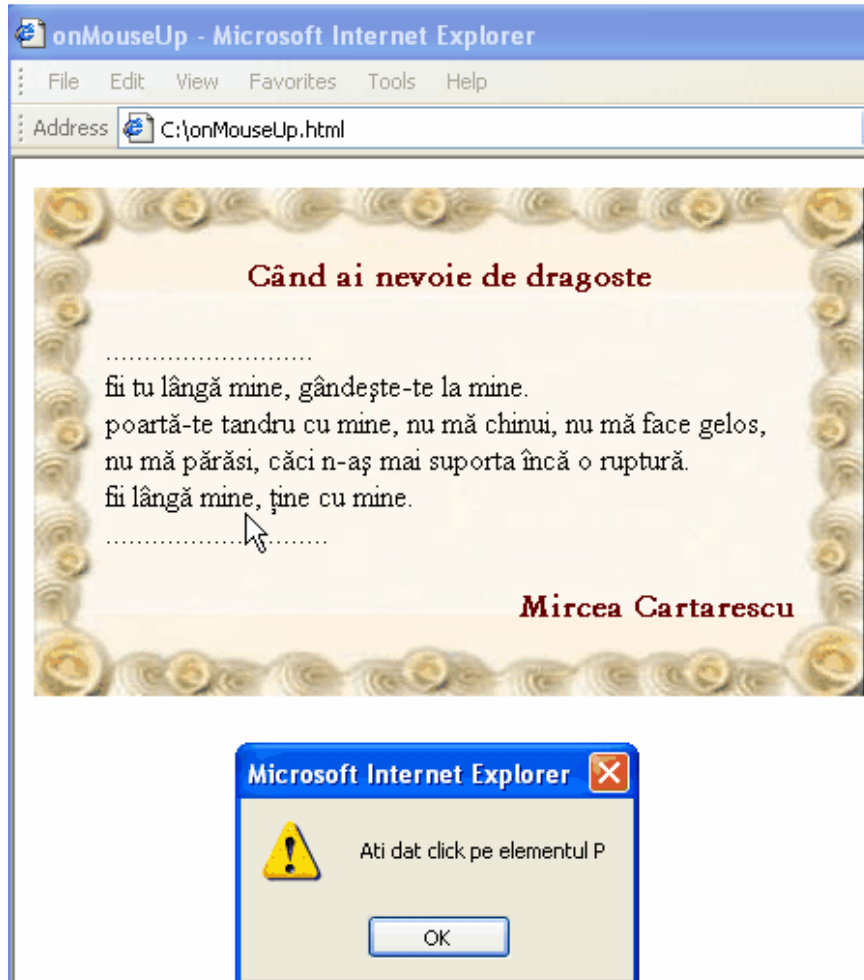
      document.all.x.style.pixelLeft=event.clientX+document.body.scrollLeft+10
      document.all.x.style.pixelTop=event.clientY+document.body.scrollTop+10
      document.all.x.style.visibility="visible"
    }
  }
function g()
  {
    if (document.all)
      document.all.x.style.visibility="hidden"
  }
</script>
```



```

nu m&#259; p&#259;r&#259;si, c&#259;ci n-a&#351; mai suporta
&icirc;nc&#259; o ruptur&#259;.<br />
fii l&acirc;ng&#259; mine, &#355;ine cu mine.<br />
.....</p>
<div class="titlul" align="right">Mircea Cartarescu </div>
</td>
<td width="30">&nbsp;</td>
</tr>
</table>
</body>
</html>

```



Aplicații cu evenimente

Sortare tabel

Aici avem o aplicație ceva mai complexă, deoarece am utilizat stiluri pentru elementele HTML, dar și pentru diverse elemente din pagină, am apelat funcții ce permit ordonarea elementelor din tabel și am tratat multe evenimente necesare utilizării aplicației.

Metoda de ordonare utilizată se numește Bubble Sort.

```
<html>
<head>
  <title>Sortare tabel</title>
<style>
body
  { color: white;
    font-size: 10pt;
    font-family: Georgia, "Times New Roman", Times, serif; }
table
  { color :#000066;
    font-size:10pt;
    font-style:normal;
    font-weight:normal;
    display:block;
    font-family: Georgia, "Times New Roman", Times, serif; }
.titlul
  { color :#000066;
    font-family: Georgia, "Times New Roman", Times, serif;
    font-size: large;
    font-weight: bolder;
    text-align: center; }
.capTabel
  { background-color:#5197FF;
    layer-background-color:#7073FE;
    border-style:outset;
    border-width:1px;
    border-left-color:white;
    border-top-color:white;
    border-bottom-color:black;
    border-right-color:black;
    color: #000066; }
.fundalTabel
  { background-color:#9BEBFF;
    layer-background-color: #FCFE52; }
</style>
<script language="JavaScript">
var d = document;
var a1=null;
var a2=null;
var mod=true;
var b=false;
function masiv(cell)
  {
    var r1=cell.parentElement.parentElement.rows;
    var t1=r1.length;
    var t2=new Array(t1-1);
    a1=new Array(t1-1);
    if
(!isNaN(Date.parse(r1.item(1).cells.item(cell.cellIndex).innerText)))
```

```

        b=true;
    for(var i=1;i<t1;i++)
        a1[i-1]=i;
    for(var i=1;i<t1;i++)
        { t2[i-1]=r1.item(i).cells.item(cell.cellIndex).innerText; }
    sortareTabel (t2);
}
function sortareTabel(array)
{
    var vector=array;
    var s1=a1;
    rest=array.length;
    for(var i=rest-1;i>=0;i--)
        { for(var j=0;j<=i;j++)
            { if (b)
                { if(Date.parse(vector[j+1])<Date.parse(vector[j]))
                    {var temp=vector[j];
                    vector[j]=vector[j+1];
                    vector[j+1]=temp;
                    var temp=s1[j];
                    s1[j]=s1[j+1];
                    s1[j+1]=temp; }
                }
            else
                { if(vector[j+1]<vector[j])
                    { var temp=vector[j];
                    vector[j]=vector[j+1];
                    vector[j+1]=temp;
                    var temp=s1[j];
                    s1[j]=s1[j+1];
                    s1[j+1]=temp; }
                }
            }
        }
}
function randuri(form)
{
    var rows = form.parentElement.parentElement.rows;
    var corpTab=form.parentElement.parentElement;
    var c1=rows.item(0).cells.length;
    var td0 = null;
    var trA=new Array(a1.length);
    var trO=null;
    if(corpTab.rows.length>0)
        { for(var i=0;i<a1.length;i++)
            { tr = d.createElement("TR");
              for(var y=(c1-1);y>=0;y--)
                  { td = d.createElement("TD");
                    tr.insertBefore(td, td0);
                    td.className = 'fundalTabel';
                    td.className =
                    rows.item(a1[i]).cells.item(y).className;
                    td.innerText =
                    rows.item(a1[i]).cells.item(y).innerText;
                    tdO=td; }
                  tdO=null;
                  trA[i]=tr; }
            stergeRand(corpTab);
            if(mod)

```

```

        {
            for(var i=trA.length-1;i>=0;i--)
                { corpTab.insertBefore(trA[i],trO);
                  trO=trA[i]; }
        }
    else
    {
        for(var i=0;i<trA.length;i++)
            { corpTab.insertBefore(trA[i],trO);
              trO=trA[i]; }
    }
}
a1=null;
b=false;
}
function stergeRand(corpTab)
{
    for(var i=corpTab.rows.length-1;i>0;i--)
        corpTab.deleteRow(i);
}
function sortare(form)
{
    if (a2==form)
        mod=!(mod);
    masiv(form);
    randuri(form);
    a2=form;
    return mod;
}
function efect (form, ef)
{
    if (ef == 1)
        { form.style.color = '#9BEBFF'; }
    else
        if (ef == 2)
            {
                form.style.color = '#000066';
            }
}
function stil(form, name, value)
{
    form.style[name] = value;
}
function antet(form, type)
{
    type = parseInt(type);
    if (isFinite(type))
        { if (type == 1)
            { stil(form, 'border', 'outset');
              stil(form, 'borderWidth', '1px');
              stil(form, 'borderRightColor', 'black');
              stil(form, 'borderBottomColor', 'black'); }
          else
            { stil(form, 'border', 'inset');
              stil(form, 'borderWidth', '1px');
              stil(form, 'borderLeftColor', 'black');
              stil(form, 'borderTopColor', 'black');
              stil(form, 'borderRightColor', 'white');
              stil(form, 'borderBottomColor', 'white');
            }
        }
}

```

```

    }
}
</script>
</head>
<body bgcolor="#FFFFFF">
<div class="titlul"> Tabel nominal </div>
<TABLE CELLPADDING=1 CELLSPACING=1 BORDER=1 WIDTH=80%>
<TBODY>
<tr>
<td ALIGN=CENTER CLASS="capTabel" onMouseOver="efect(this, 1)"
onMouseOut="efect(this, 2)" onMouseDown="antet(this, 0);"
onMouseUp="antet(this, 1);" onClick="sortare(this);"><B>Nume si
prenume</B></td>
<td ALIGN=CENTER CLASS="capTabel" onMouseOver="efect(this, 1)"
onMouseOut="efect(this, 2)" onMouseDown="antet(this, 0);"
onMouseUp="antet(this, 1);"
onClick="sortare(this);"><B>Oras</B></td>
<td ALIGN=CENTER CLASS="capTabel" onMouseOver="efect(this, 1)"
onMouseOut="efect(this, 2)" onMouseDown="antet(this, 0);"
onMouseUp="antet(this, 1);" onClick="sortare(this);"><B>Data
nasterii</B></td></tr>
<tr>
<td CLASS="fundalTabel">Ionescu Daniel</td>
<td CLASS="fundalTabel">Bucuresti</td>
<td CLASS="fundalTabel">1/09/1976</td>
</tr>
<tr>
<td CLASS="fundalTabel">Popescu Corina </td>
<td CLASS="fundalTabel">Ploiesti</td>
<td CLASS="fundalTabel">04/06/1975</td>
</tr>
<tr>
<td CLASS="fundalTabel">Enescu Bogdan </td>
<td CLASS="fundalTabel">Targoviste</td>
<td CLASS="fundalTabel">03/17/1985</td>
</tr>
<tr>
<td CLASS="fundalTabel">Coman Cristina </td>
<td CLASS="fundalTabel">Iasi</td>
<td CLASS="fundalTabel">03/02/1970</td>
</tr>
<tr>
<td CLASS="fundalTabel">Chivu Ingrid </td>
<td CLASS="fundalTabel">Brasov</td>
<td CLASS="fundalTabel">29/07/1975</td>
</tr>
<tr>
<td CLASS="fundalTabel">State Gabriela </td>
<td CLASS="fundalTabel">Cluj</td>
<td CLASS="fundalTabel">17/24/1976</td>
</tr>
<tr>
<td CLASS="fundalTabel">Nedelcu Mihaela </td>
<td CLASS="fundalTabel">Pascani</td>
<td CLASS="fundalTabel">11/01/1973</td>
</tr>
</TBODY>
</TABLE>

```

```
</body>
</html>
```

The screenshot shows a Microsoft Internet Explorer window with the title 'Sortare tabel - Microsoft Internet Explorer'. The address bar shows 'C:\Sortare tabel.html'. The main content is a table with the following data:

Nume si prenume	Oras	Data nasterii
Ionescu Daniel	Bucuresti	1/09/1976
Popescu Corina	Ploiesti	04/06/1975
Enescu Bogdan	Targoviste	03/17/1985
Coman Cristina	Iasi	03/02/1970
Chivu Ingrid	Brasov	29/07/1975
State Gabriela	Cluj	17/24/1976
Nedelcu Mihaela	Pascani	11/01/1973

Formular numai cu cifre

Am creat un formular ce verifică pe rând toate caracterele introduse și ne permite doar introducerea cifrelor. În primul câmp putem introduce numere cu zecimale, iar în al doilea câmp putem introduce numai numere fără zecimale.

Funcția cifre, apelată la acționarea evenimentului *onKeyPress* primește pe rând caracterele introduse de la tastatură, verifică dacă am introdus cifre și returnează numai cifre (câmpul 2) sau cifre și semnul "."(punct). Odată introdus caracterul *punct* (.), \dagger i se atribuie variabilei *punct* valoarea 1, fapt ce nu va mai permite reafișarea acestui caracter.

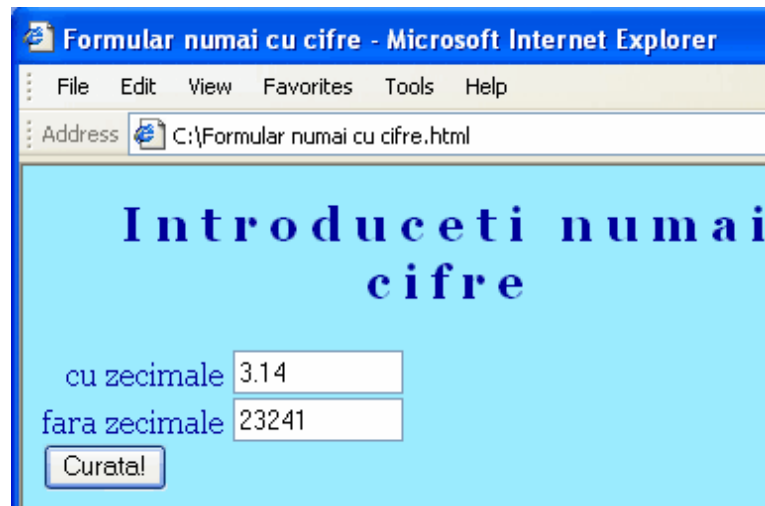
```
<html>
<head>
<title>
  Formular numai cu cifre
</title>
<style>
body
{
  color: #000099;
  font-family: Georgia, "Times New Roman", Times, serif;
  background-color: 9BEBFF;
}
.titlul
{
  font-size: 18pt;
  letter-spacing: 0.4em;
  text-align: center;
  font-weight: bold;
}
</style>
</head>
<body>
<div class="titlul">Introduceti numai cifre </div>
<br />
<form>
  &nbsp; &nbsp;  &nbsp; cu zecimale
  <INPUT name=CuZecimale TYPE=text size=10 onKeyPress="return
    cifre(event,'Zecimale') "> <br />
  fara zecimale
```

```

<INPUT name=FaraZecimala TYPE=text size=10 onKeyPress="return
cifre(event,'fara_zecimala') "> <br />
<input type="submit" name="Submit" value="Curata!" id="Submit"
onclick()="">
</form>
<script language="javascript">
var InternetExplorer = document.all?true:false;

function cifre(e,zec)
{
var tasta = (InternetExplorer) ? window.event.keyCode : e.which;
var obiect = (InternetExplorer) ? event.srcElement : e.target;
var numar = (tasta > 47 && tasta < 58) ? true:false;
var punct = (tasta==46 && zec=='Zecimala' &&
(obiect.value.indexOf(".")<0 || obiect.value.length==0)) ? true:false;
if(tasta < 32)
return true;
return (numar || punct);
}
</script>
</body>
</html>

```



Contorizare caractere

În exemplul de mai jos am creat o funcție ce contorizează cuvintele introduse de un utilizator într-un câmp de tip textarea. Distanța dintre cuvinte este dată de un spațiu. În momentul în care dăm click în afara zonei de introducere a textului, putem vedea rezultatul afișat.

```

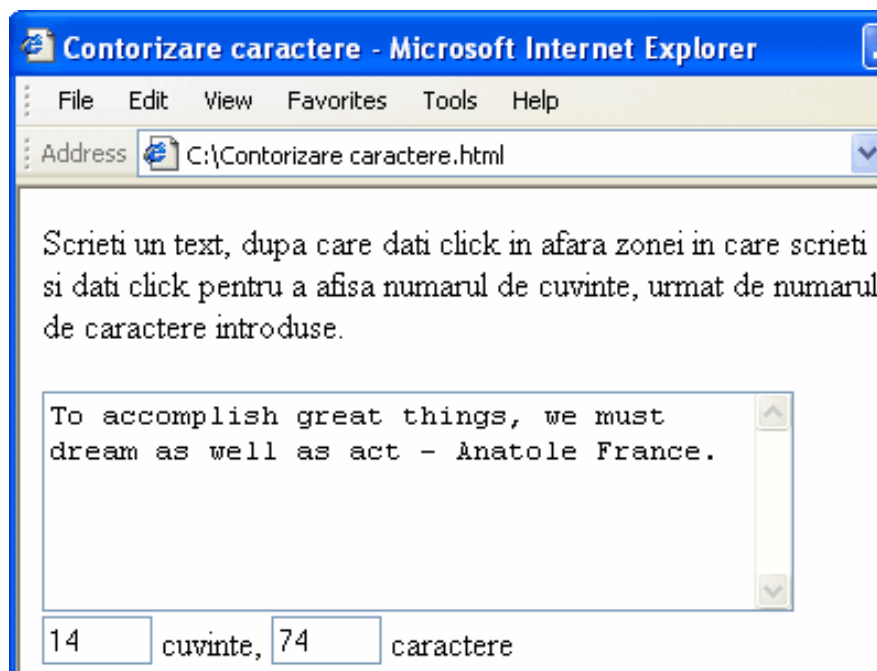
<html>
<head>
<title>
Contorizare caractere
</title>
<script language="javascript">
function f(x)
{
var i=0;
var NrCuv=1;
while(i<=x.length)
{

```

```

    if (x.substring(i,i+1) == " ")
    {
        NrCuv++;
        i++;
    }
    if (x.substring(i,i+1) == "\n")
    {
        NrCuv++;
        i++;
    }
    i++;
}
return NrCuv;
}
</script>
</head>
<body>
Scrieti un text, dupa care dati click in afara zonei in care scrieti
<br />
si dati click pentru a afisa numarul de cuvinte, urmat de numarul
<br />
de caractere introduse.
<form>
<textarea cols=40 rows=6 onChange= "this.form.caracter.value =
    this.value.length; this.form.cuvant.value=f(this.value)">
</textarea>
<br />
<input name=cuvant value=0 size=4> cuvinte,
<input name=caracter value=0 size=4> caractere
</form>
</body>
</html>

```



Formular de calcul instantaneu

În exemplul de mai jos am creat un formular ce permite calcularea unei sume, pe măsură ce sunt introduse cifrele în câmpurile formularului.

```
<html>
<head>
  <title>Exemplu de formular</title>
</head>
<body>
  {
    background-color: #B3FFB3;
  }
  .titlul
  { font-family: "Aldine721 Lt BT", Adolescence, "Alexei Copperplate";
    font-size: 18px;
    font-weight: bolder;
    color: #006600; }
</body>
<script type="text/javascript" language="JavaScript">
function PornireCalculare()
{
  interval = setInterval("Calculare()",1);
}
function Calculare()
{
  primul = document.adunare.x1.value;
  alDoilea = document.adunare.x2.value;
  document.adunare.x3.value = (primul * 1) + (alDoilea * 1);
}
function Oprire()
{ clearInterval(interval);
}
</script>
</head>
<body>
<span class="titlul">Introduceti numere, pentru a le aduna
</span><br />
<form name="adunare">
<input type="text" name="x1" value="" onFocus="PornireCalculare();" onBlur=
  "Oprire();" size="10"> +
<input type="text" name="x2" value="" onFocus="PornireCalculare();" onBlur=
  "Oprire();" size="10"> =
<input type="text" name="x3" size="10">
</form>
</body>
</html>
```

