

# Evenimente JavaScript

### Obiective:

- să înțeleagă noțiunea de eveniment JavaScript
- să poată utiliza un eveniment JavaScript acolo unde este necesar
- să înțeleagă noțiunile de eveniment și tratarea evenimentelor
- să poată crea un script ce utilizează evenimente JavaScript

## 7.1. Introducere

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.

## 7.2. 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.

## 7.3. 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.   |

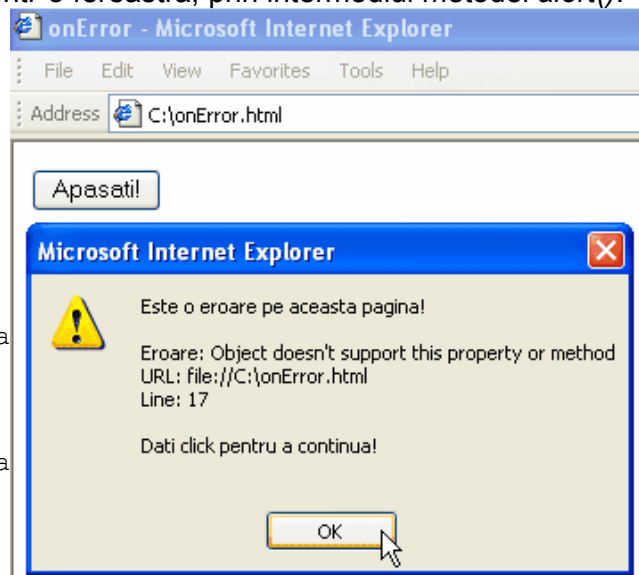
## 7.3.1. Tratarea evenimentelor de către sistem

**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 *writtnln*). 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.writteln("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.

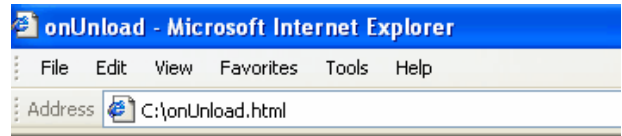
*onUnload* este event handler pentru obiectul: *Window*.

**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;
 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>

```



**Alegeti varianta corecta!**

**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



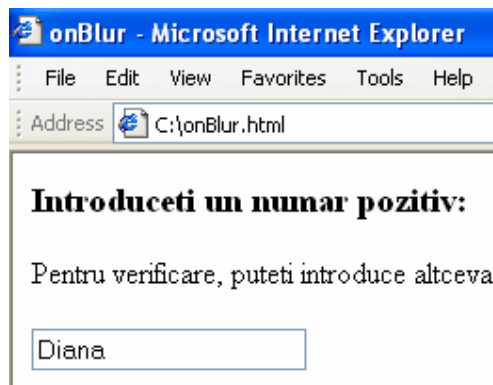
## 7.3.2. Tratarea evenimentelor de către un formular

**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("Introduceți un număr!");
    else
  if (x<0)
    {
      alert("Va rog sa introduceți un număr
        pozitiv!");
    }
  }
</script>
</head>
<body>
<H3>Introduceți un număr 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>
```

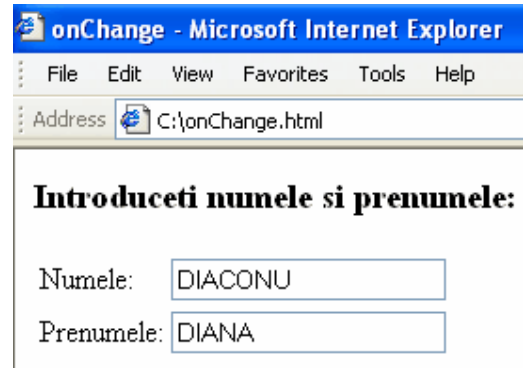


**onChange** va acționa în urma pierderii focalizării asupra unui element și a modificării conținutului unui câmp.

*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>
```

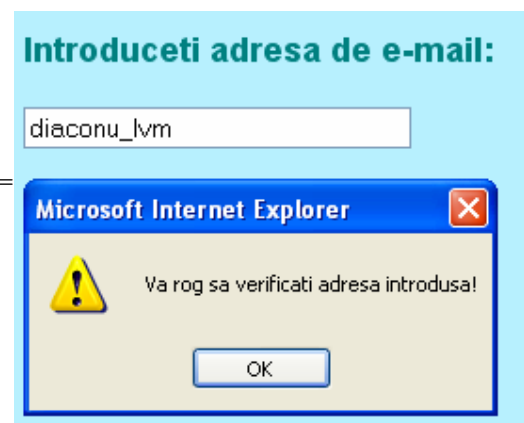


**onFocus** va acționa în urma focalizării asupra unui element cum ar fi: o fereastră, un frame, un element dintr-un formular.

*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.

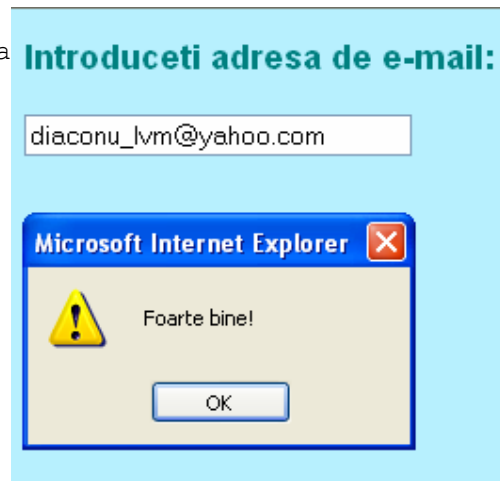
```
<html>
<head>
    <title>
        Adresa email
    </title>
<script type="text/javascript" language="
JavaScript">
this.formular.email.focus();
this.formular.email.select();
function verificare()
{
    x = new String();
    x = this.formular.email.value;
    if (x.match("@"))
```



```

        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.

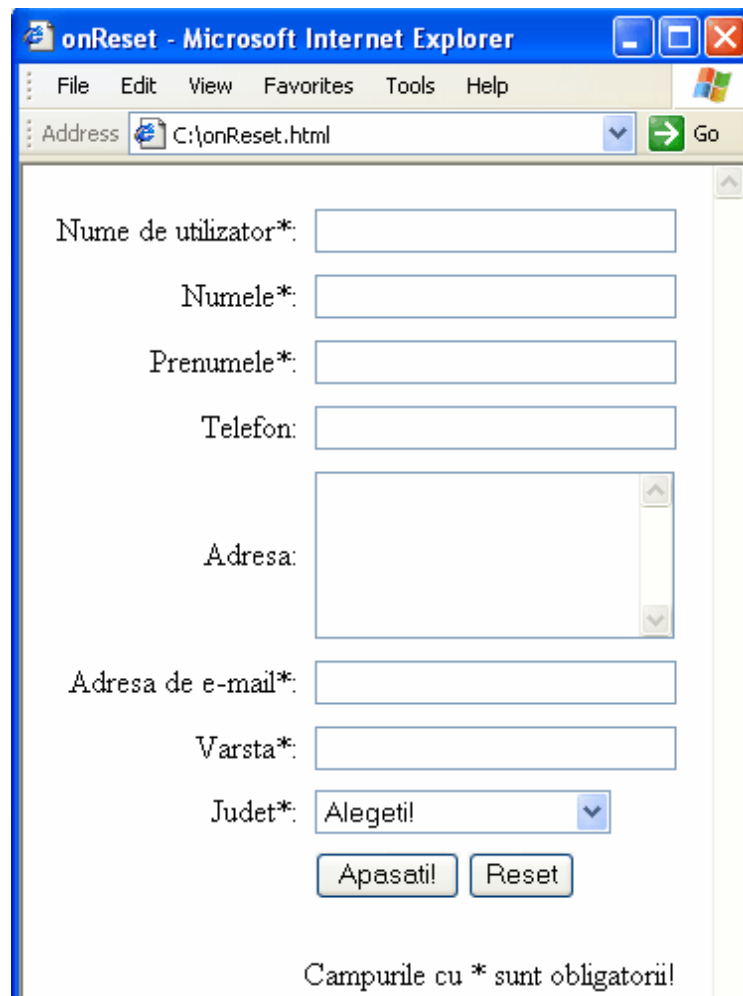
După scrierea formularului, am introdus un script de verificare, ce apelează diverse funcții poziționate în antetul paginii. De exemplu, am creat o funcție ce permite atenționarea utilizatorilor în cazul în care au lăsat anumite câmpuri necompletate. Una dintre acestea este apelată de către:

*VerificareFormular.VerifDistinct("utilizator","cerinta","Introduceti 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.

Mai jos, am afișat 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");*

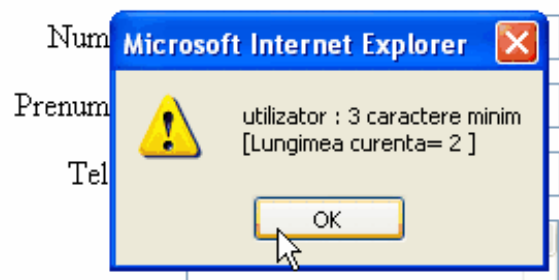


## Mesaje de eroare

VerificareFormular.

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

Nume de utilizator\*: di

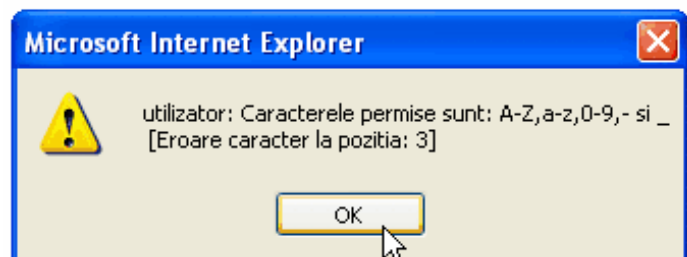


Apelul:

VerificareFormular.

`VerifDistinct("utilizator", "CazUtilizator");`

Nume de utilizator\*: di^



Cazul:

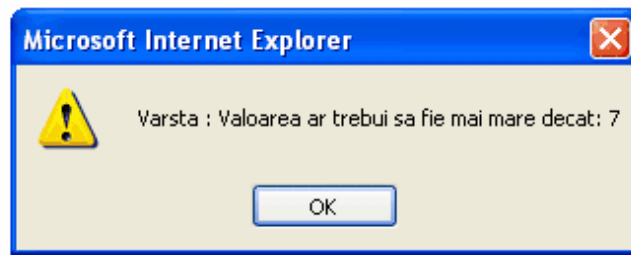
case "CazUtilizator":



Apelul:  
*VerificareFormular.*  
*VerifDistinct("Varsta","MaiMareDecat");*

Varsta\*:

Cazul:  
*case "MaiMareDecat":*



```
<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.AVF=VF;
      this.CL = Curata;
    }
function VF(NumeFunctie)
  {
    this.ValFormTrimis.VerifPlus = NumeFunctie;
  }
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, telefon...
    {
        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)
{
//funcție 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 + " : Introdueceti 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>
        <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>

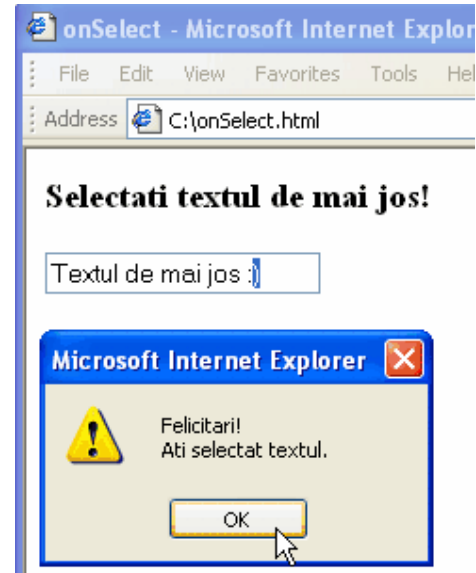
```

**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*.

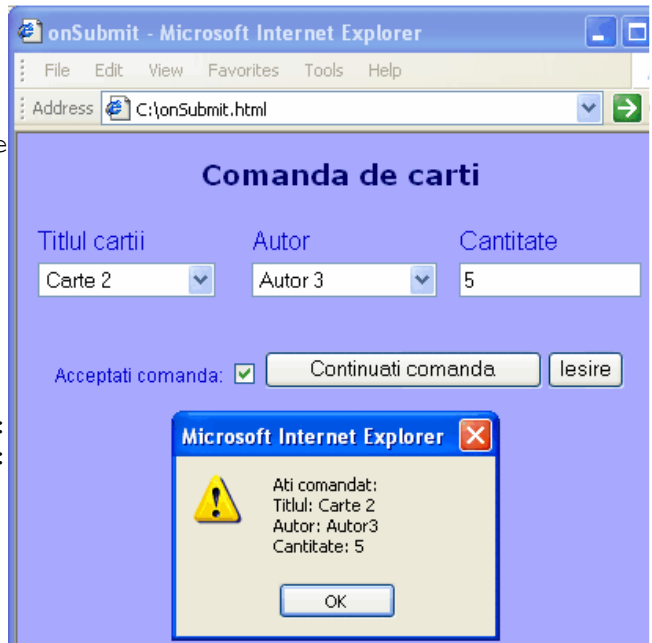
```
<html>
<head>
  <title>onSubmit</title>
<style>
body
  { font-family: Geneva, Arial;
    font-size: 12px;
    color: #0000CC;
    background-color: #A8A8FF;
    text-align: center; }
.titlul
  { font-family: Verdana, Arial;
    font-size: 18px;
    font-weight: bolder;
    color: #000066;
    text-align: 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
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 type="text"
name="cantitate" size="15" value="">
</td></tr>
</table>
<br /><br />
  Acceptati comanda:

```



```

<table border="0">
<tr>
<td><input type="checkbox" name="DeAcord" value="" />
  Acceptati comanda:

```

```



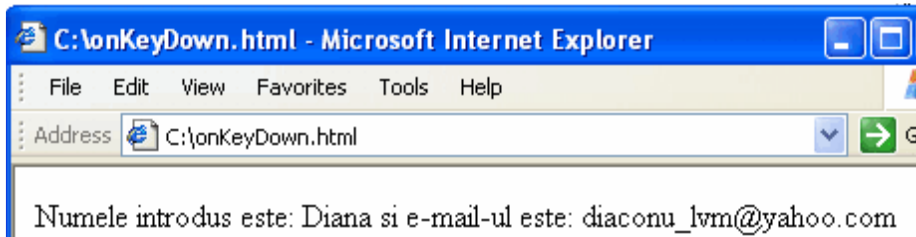
```

### 7.3.3. 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: bolder;
    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.

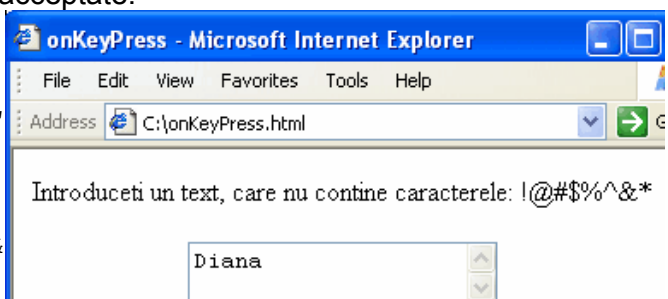
**onKeyPress** este event handler pentru obiectele: *Document*, *Image*, *Link* și *Textarea*.

**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) ||

```



```

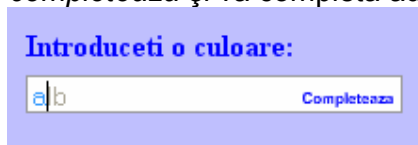
        (event.keyCode > 90 &&
        event.keyCode < 97))
        event.returnValue = false; }
</script>
</head>
<body>
<center>
<form onSubmit="return false;"
Introduceti un text, care nu contine
caracterele: !@#$%^&* <br>
<textarea rows=2 cols=20
name=comments onKeyDown="f()" >
</textarea>
</form>
</center>
</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="s1b" style="position: absolute; top: 5px; right: 5px; z-index:
    3; display: none;">
    <a href="#" onClick="SetareCuloare()">
      
    </a>
  </div>
</div>
</form>
</div>
</body>
</html>

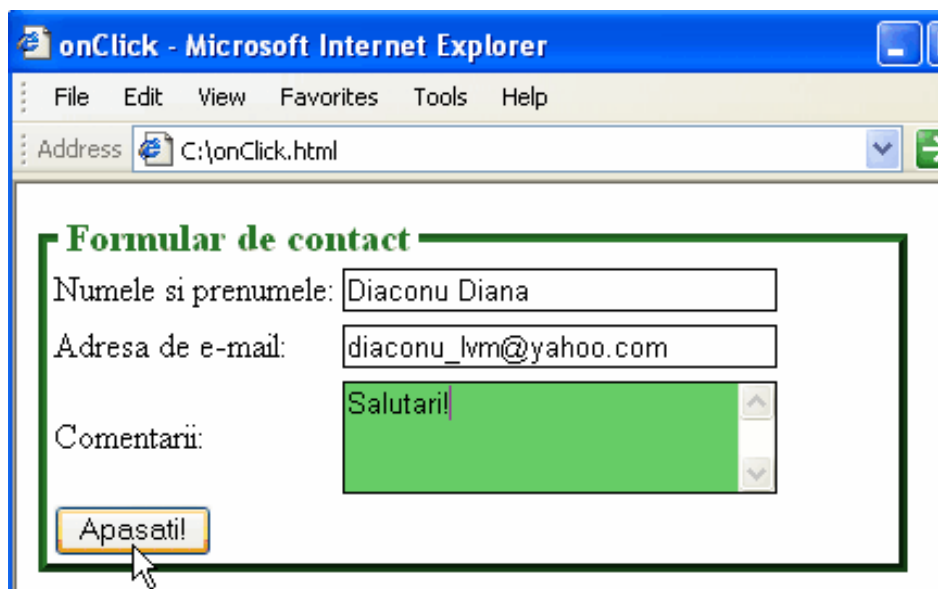
```

### 7.3.4. Tratarea evenimentelor de către mouse

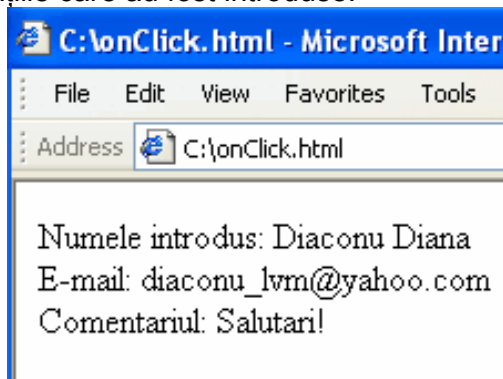
**onClick** va acționa în urma unui click stânga de mouse dat de către utilizator pe un obiect specificat dintr-un formular.

*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; }
</style>
</head>
<body>
  <form>
    Numele si prenumele: <input type="text" value="Diaconu Diana"/>
    Adresa de e-mail: <input type="text" value="diaconu_lvm@yahoo.com"/>
    Comentarii: <input type="text" value="Salutari!"/>
    <input type="button" value="Apasati!"/>
  </form>
</body>
</html>
```

```

.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].tagName.
                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>

```

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

*onDbIClick* 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>onDbIClick</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" onDblclick="f(1)">
    <input type="button" value="Stil 2" onDblclick="f(2)">
    <input type="button" value="Stil 3" onDblclick="f(3)">
    <input type="button" value="Stil 4" onDblclick="f(4)">
</form>
</div>
</body>
</html>

```

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

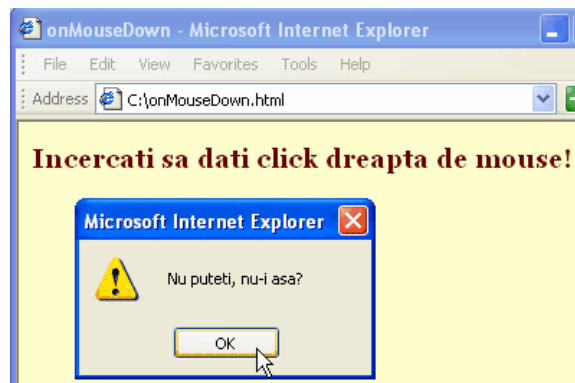
*onMouseDown* este event handler pentru obiectele: *Button*, *Document* și *Link*.

**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>

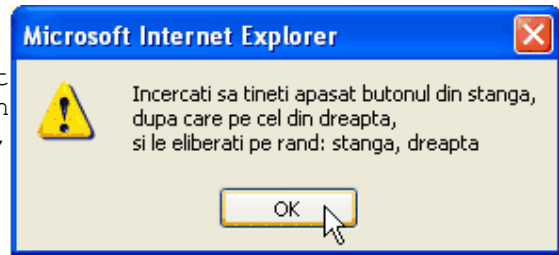
```



```

<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>

```



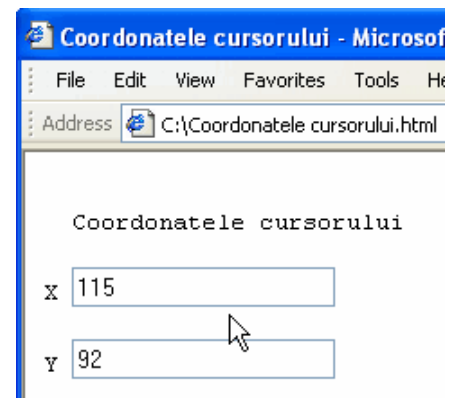
**onMouseMove** este o acțiune ce va fi declanșată de mișcarea cursorului mouse-ului.

**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  onMouseOver="bgColor='#AAAAFF'"  onClick="window.location='index.html'"
      style="cursor:hand" onMouseOut= "bgColor= '#DDDDFF'" bgcolor="#FFFFFF"
      width="100" 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
      onMouseOver="bgColor='#AAAAFF'"
      onClick="window.location='pagina2.html'"
      onMouseOut="bgColor='#DDDDFF'"          bgcolor="#FFFFFF"          width="100"
      align="center" valign="center"><b><font color="#D5D5D5">
      <a href="pagina2.html" class="legatura">Pagina 2</a></font></b></td>

```

```

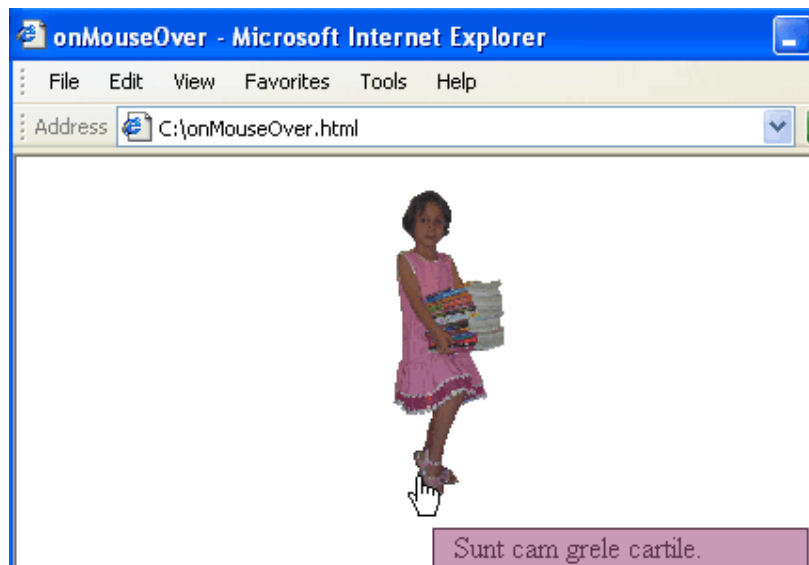
<td width="1" align="center">|</td>
<td
    onMouseOver="bgColor='#AAAAFF'"
    onClick="window.location='pagina3.html'" style="cursor:hand"
    onMouseOut="bgColor='#DDDDFF'" bgcolor="#FFFFFF" width="100"
    align="center" valign="center"><b><font color="#D5D5D5">
    <a href="pagina3.html" class="legatura">Pagina 3</a></font></b></td>
</tr>
</table>
<p class="citata">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">'+textul+'</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>
</head>
<body>
<div id="x" class="unu"></div>
<div align="CENTER" onMouseover="f(this,event,'Sunt cam grele cartile.');"
onMouseout="g();" class="doi">
<IMG SRC="printesa.JPG" WIDTH="121.4" HEIGHT="171.2" BORDER="0">
</div>
</body>
</html>

```

**onMouseUp** este o acțiune ce va fi declanșată în momentul în care cursorul mouse-ului este eliberat.

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

**Exemplu** În exemplul de mai jos, se afișează un mesaj în momentul în care eliberăm cursorul mouse-ului, după ce am dat click pe o suprafață din interiorul paginii. Captura de mai jos este de fapt un tabel cu trei coloane și cu o imagine ce conține trandafiri albi ca fundal, iar pentru titlu și autor am utilizat un stil special creat, pentru a se potrivi cu fundalul.

```

<html>
<head>
  <title>onMouseUp</title>
<script type="text/javascript" language="javascript">
function Eveniment (e)
  {
    var x1
    if (!e)
      var e = window.event
    if (e.target)
      x1 = e.target
    else
      if (e.srcElement)
        x1 = e.srcElement
    if (x1.nodeType == 3)
      x1 = x1.parentNode
    var y
    y=x1.tagName
    alert("Ati dat click pe elementul " + y )
  }
</script>
<style type="text/css">
.titlul
  { font-family: "Aldine721 Lt BT", "Adolescence", "Alexei Copperplate";
    font-size: 16px;
    font-weight: bolder;

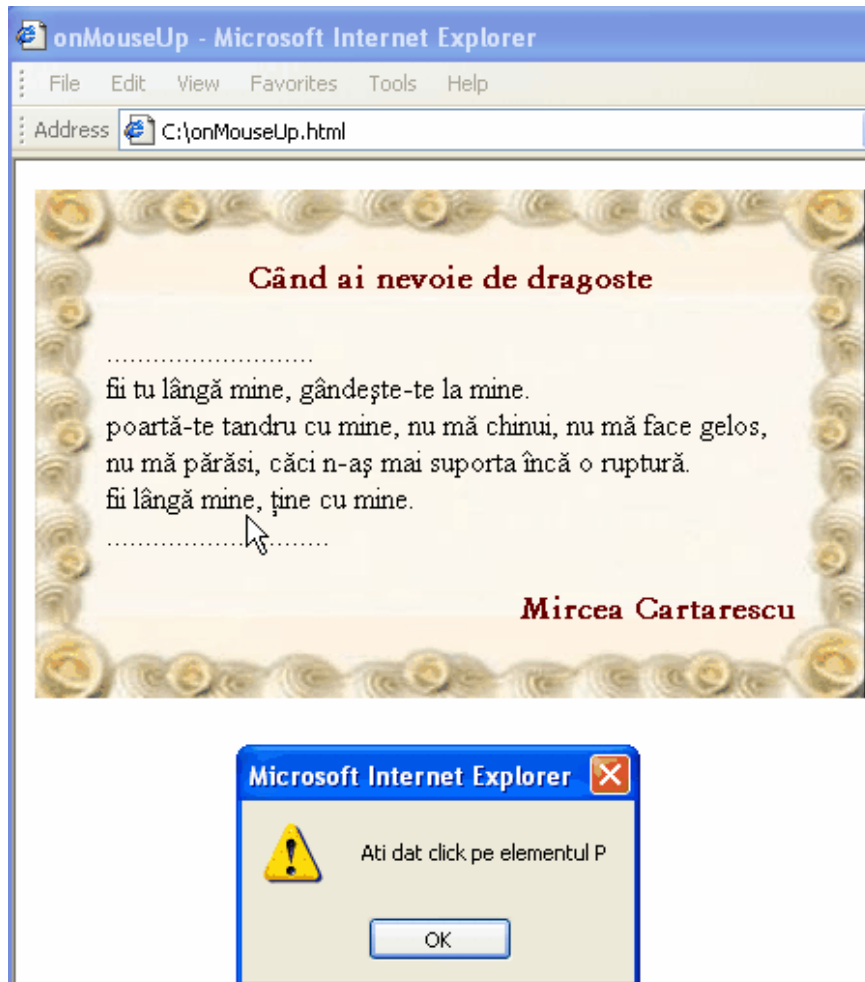
```

```

        color: #660000; }
</style>
</head>

<body onMouseUp="Eveniment(event)">
<table border="0" background="tr_albi.jpg" valign="middle" height="263"
width="430" id="tr">
  <tr>
    <td width="30">&nbsp;</td>
    <td >
      <div class="titlul" align="center">C&acirc;nd ai nevoie de dragoste
      </div>
      <p>.....<br />
      fii tu l&acirc;ng&#259; mine, g&acirc;nde&#351;te-te la mine.<br />
      poart&#259;-te tandru cu mine, nu m&#259; chinui, nu m&#259; face
      gelos,<br />
      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>

```



## 7.4. 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;
  }
</script>
```

```

    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 tdO = 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, tdO);
                    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 CELLSPACING=1 CELLPADDING=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>

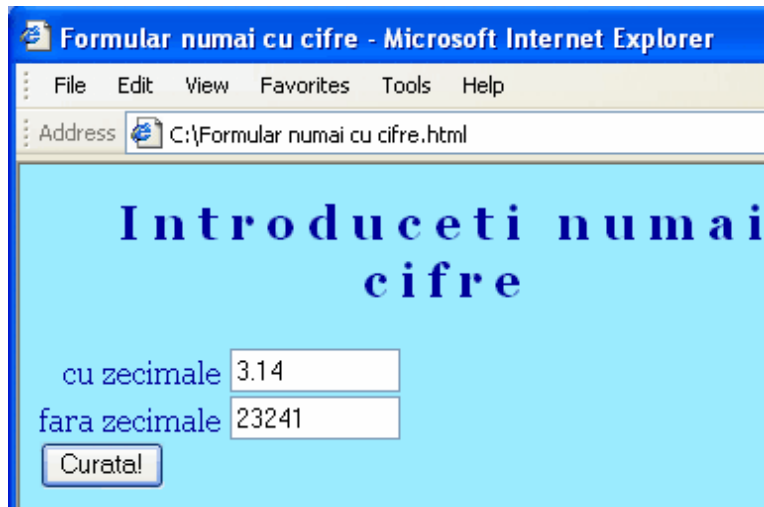
```



```

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.

```

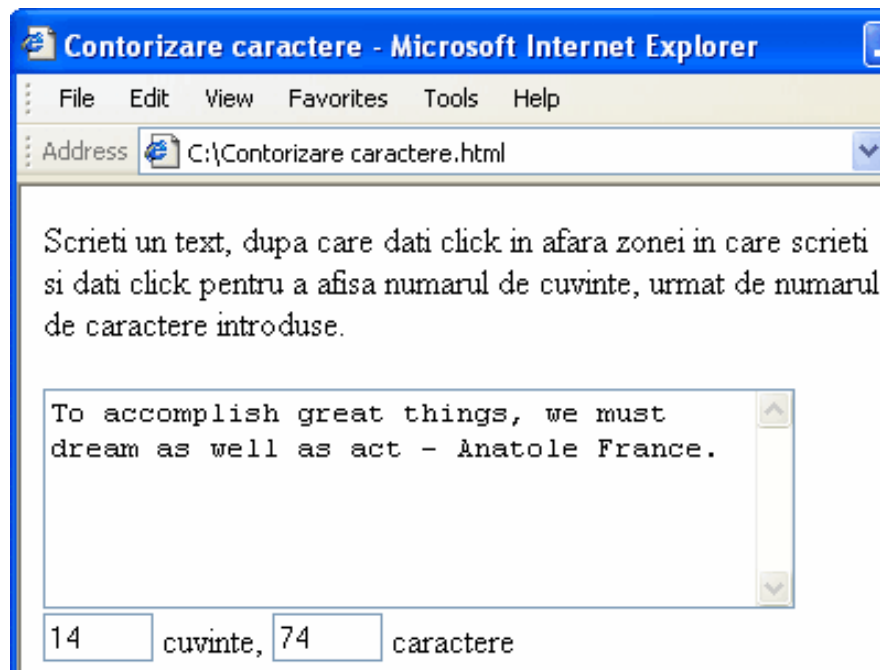
<html>
<head>
<title>
Contorizare caractere
</title>
<script language="javascript">
function f(x)
{
var i=0;
var NrCuv=1;
while(i<=x.length)
{
if (x.substring(i,i+1) == " ")
{

```

```

        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.character.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>

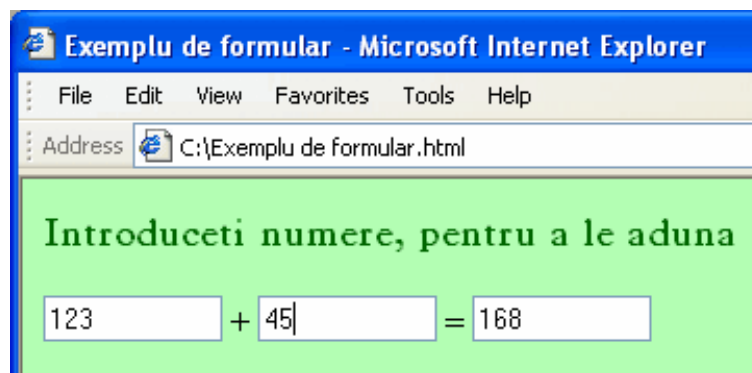
```



## 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>
<style>
body
{
  background-color: #B3FFB3;
}
.titlul
{ font-family: "Aldine721 Lt BT", Adolescence, "Alexei Copperplate";
  font-size: 18px;
  font-weight: bolder;
  color: #006600; }
</style>
<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>
```



## Evaluare

1. Enumerați și descrieți evenimentele JavaScript.
2. Care este diferența dintre evenimente și tratarea evenimentelor?
3. Realizați o pagină ce conține trei imagini care dacă dăm click pe ele se schimbă culoarea fundalului.
4. Care sunt evenimentele declanșate de către mouse?

## Capitolul 8.

# Obiecte JavaScript

**Obiective:**

- să înțeleagă noțiunea de obiect JavaScript.
- să poată utiliza un obiect JavaScript acolo unde este necesar.
- să poată crea un script ce utilizează obiecte JavaScript.

### 8.1. Introducere

Pentru a nu crea confuzii, trebuie să plecăm de la ideea că JavaScript nu este Java. Java este un limbaj de programare orientat-obiect sau OOP (Object Oriented Programming), iar JavaScript este **bazat pe obiecte**.

#### Definiție

Un **obiect** este o colecție de proprietăți și metode reținute sub un singur nume.

De exemplu, am putea avea obiectul *televizor*. Un televizor are câteva proprietăți: data fabricației, marca, culoarea, modelul și metode: *pornește()*, *seOprește()*. Toate aceste proprietăți și metode definesc obiectul *televizor*. Ele pot face parte și din caracteristicile altui obiect, dar luate împreună definesc obiectul *televizor*. JavaScript permite crearea unor obiecte cu caracteristici proprii, dar are și obiecte predefinite, cum ar fi: *Anchor*, *Area*, *Applet*, *Form*, *Image*, pe care o să le studiem în acest capitol.

**Proprietățile** sau atributele unui obiect descriu caracteristicile speciale și identitatea.

**Metodele** sau funcțiile unui obiect sunt vizibile din exterior. Cu ajutorul lor putem crea obiecte sau le putem modifica.

Pentru a realiza mai ușor și mai eficient scripturi JavaScript, este bine să utilizăm obiectele deja existente și să creăm altele noi doar dacă este strict necesar.

### Anchor

Obiectul Anchor este un loc dintr-o pagină (un șir de caractere sau o imagine) care reprezintă ținta unei legături de tip hipertext.

Proprietăți: *accessKey*, *charset*, *coords*, *href*, *id*, *innerHTML*, *name*, *rel*, *rev*, *shape*, *target*, *type*.

Metode: *blur()*, *focus()*.

Evenimente: *onBlur*, *onFocus*.

| Proprietate      | Descriere  |
|------------------|--|
| <i>accessKey</i> | Setează o tastă pentru a realiza accesul către o legătură de tip hipertext.                                |
| <i>charset</i>   | Setează un set de caractere pentru a realiza accesul către diverse resurse prin legături de tip hipertext. |
| <i>coords</i>    | Setează o listă de coordonate separate prin virgulă, pentru a defini o anumită regiune activă.             |
| <i>href</i>      | Setează un URL pentru a realiza accesul către diverse resurse prin legături de tip hipertext.              |
| <i>id</i>        | Setează un <i>id</i> pentru o legătură de tip hipertext.   |
| <i>innerHTML</i> | Setează un <i>text</i> pentru o legătură de tip hipertext.   |
| <i>name</i>      | Returnează numele unei legături de tip hipertext.  |
| <i>rel</i>       | Setează relația dintre documentul curent și ținta URL.   |
| <i>rev</i>       | Setează relația dintre documentul curent și ținta URL.   |



|        |   |
|--------|---|
| shape  | Setează o suprafață activă. Coordonatele sunt definite cu ajutorul proprietății <i>coords</i> . |
| target | Setează unde anume se va deschide ținta URL.  |
| type   | Setează tipul MIME al unei legături de tip hipertext.   |

| Metodă  | Descriere   |
|---------|---|
| blur()  | Înlătură focalizarea de pe o legătură de tip hipertext. |
| focus() | Oferă focalizare asupra unei legături de tip hipertext. |

| Eveniment | Descriere   |
|-----------|---|
| onblur    | Acțiune ce va fi declanșată când se pierde focalizarea de pe o legătură de tip hipertext. |
| onfocus   | Acțiune ce va fi declanșată de focalizarea asupra unei legături de tip hipertext.         |

**Exemplu** În acest exemplu, am utilizat obiectul Anchor, cu proprietatea *accessKey* pentru a obține accesul la o legătură de tip hipertext cu ajutorul tastelor. În funcția *f()* am definit tastele a și b pentru accesul cu ajutorul lor la site-urile: [www.didactic.ro](http://www.didactic.ro) și [www.olimpiada.info](http://www.olimpiada.info). În exemplul de mai jos am utilizat Mozilla Firefox

```
<html>
<head>
  <title>Array cu accessKey</title>
<script type="text/javascript">
function f()
  {
    document.getElementById('ancora1').accessKey="a";
    document.getElementById('ancora2').accessKey="b";
  }
</script>
<style type="text/css">
.titlul
{
  font-family: Georgia, "Times New Roman", Times;
  font-size: 18px;
  font-weight: bolder;
  text-transform: uppercase;
  color: #006633;
  border-bottom-width: thin;
  border-bottom-style: outset;
  border-bottom-color: #006633;
}
</style>
</head>
<body onLoad="f()">
<span class="titlul">Selectati tastele: </span>
  <br /><br />
Alt+a pentru a vizita site-ul <a id="ancora1"
  href="http://www.didactic.ro">
didactic.ro</a><br />sau<br />
Alt+b pentru a vizita site-ul <a id="ancora2"
  href="http://www.olimpiada.info">
olimpiada.info</a><br />
</body>
</html>
```

### SELECTATI TASTELE:

Alt+a pentru a vizita site-ul [didactic.ro](http://www.didactic.ro)

sau

Alt+b pentru a vizita site-ul [olimpiada.info](http://www.olimpiada.info)

**Exemplu** În acest exemplu, am utilizat obiectul Anchor, cu proprietatea *innerHTML* pentru a obține accesul la două legături diferite de tip hipertext, prin apăsarea butonului

**Schimbați legatura.** După cum se poate vedea din exemplu, dacă dăm click pe butonul din pagină, se schimbă legătura de la [www.didactic.ro](http://www.didactic.ro), la [www.olimpiada.info](http://www.olimpiada.info). În exemplul de mai jos am utilizat Mozilla Firefox.

```
<html>
<head>
<title>Anchor cu innerHTML</title>
<script
                                type="text/javascript"
language="javascript">
function f()
{

    document.getElementById('legatura').innerHTML="
    Olimpiada de informatica"

    document.getElementById('legatura').href="http:
    //www.olimpiada.info"
}
</script>
</head>
<body>
<a id="legatura" href="http://www.didactic.ro">
Portalul profesorului modern</a>
<br /><br />
<input type="button" onclick="f()" value=
"Schimbati legatura">
</body>
</html>
```

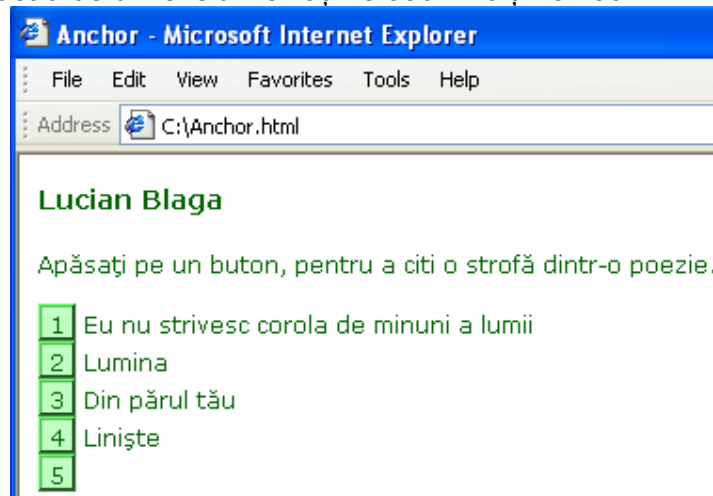
[Portalul profesorului modern](#)

Schimbati legatura

[Olimpiada de informatica](#)

Schimbati legatura

**Exemplu** În acest exemplu, am utilizat obiectul Anchor, pentru a obține accesul la o pagină în care am scris mai multe strofe din poeziile poetului Lucian Blaga. Strofele vor apărea într-o fereastră de dimensiunile: lățime 350 / înălțime 150.



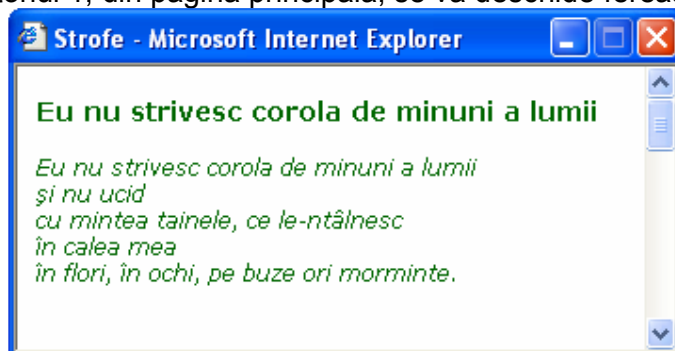
```
<html>
<head>
<title>Anchor</title>
<script type="text/javascript" language="javascript">
function f(n)
{
    fereastr=open("strofe.html","", "scrollbars=yes,width=350, height=150");
    if (fereastr.document.anchors.length >= n)
        fereastr.location.hash=n;
    else
        alert("Aceasta ancora nu exista!")
}
</script>
</head>
<body>
<h2 style="color: green;">Lucian Blaga</h2>
<p style="color: green;">Apăsați pe un buton, pentru a citi o strofă dintr-o poezie.</p>
<ol style="list-style-type: none; color: green;">
<li>1 Eu nu strivesc corola de minuni a lumii</li>
<li>2 Lumina</li>
<li>3 Din părul tău</li>
<li>4 Liniște</li>
<li>5 </li>
</ol>
</body>
</html>
```

```

}
</script>
<link href="StilBlaga.css" rel="stylesheet" type="text/css">
</head>
<body>
<span class="titlul">Lucian Blaga</span>
<form>
Ap&#259;sa&#355;i pe un buton, pentru a citi o strof&#259; dintr-o poezie.
<br /><br />
<input id="legatura" type="button" value="1" name="buton"
onClick="f(this.value)" class="b"> Eu nu strivesc corola de minuni a lumii
<br />
<input type="button" value="2" name="buton" onClick="f(this.value)"
class="b"> Lumina<br />
<input type="button" value="3" name="buton" onClick="f(this.value)"
class="b"> Din p&#259;rul t&#259;u<br />
<input type="button" value="4" name="buton" onClick="f(this.value)"
class="b"> Lini&#351;te<br />
<input type="button" value="5" name="buton" onClick="f(this.value)"
class="b">
</form>
</body>
</html>

```

Pagina strofe.html conține ținta care va fi activată în momentul în care se dă click pe un buton. Dacă dăm click pe butonul 1, din pagina principală, se va deschide fereastra de mai jos.



Dacă dăm click pe butonul 5, se va deschide o fereastră în care se specifică faptul că nu există ancoră pentru acest buton, ca în imaginea de mai jos.



```

<html>
<head><title>Strofe</title>
<link href="StilBlaga.css" rel="stylesheet" type="text/css">
</head>
<body>
<A name="1" class="titlul">Eu nu strivesc corola de minuni a lumii</A>
<br /><br />
<span class="vers">Eu nu strivesc corola de minuni a lumii<br />
&#351;i nu ucid<br />
cu mintea tainele, ce le-nt&acirc;lnesc<br />
&icirc;n calea mea<br />
&icirc;n flori, &icirc;n ochi, pe buze ori morminte.<br />
</span>

```

```

<br /><br /><br />
<A name="2" class="titlul">Lumina</A>
<br />
<br />
<span class="vers">Lumina ce-o simt <br />
n&#259;v&#259;lindu-mi &icirc;n piept c&acirc;nd te vad, <br />
oare nu e un strop din lumina <br />
creat&#259; &icirc;n ziua dint&acirc;i,<br />
din lumina aceea-nsetat&#259; ad&acirc;nc de via&#355;&#259;?</span><br />
<br /><br /><br /><br />
<A name="3" class="titlul">Din p&#259;rul t&#259;u</A>
<br />
<br />
<span class="vers">&Icirc;n&#355;elepciunea unui mag mi-a povestit odat&#259;
<br />
de-un val prin care nu putem str&#259;bate cu privirea,<br />
p&#259;ienjeni&#351; ce-ascunde pretutindeni firea,<br />
de nu vedem nimic din ce-i aieva.<br />
</span><br /><br /><br /><br /><br />
<A name="4" class="titlul">Lini&#351;te</A>
<br />
<br />
<span class="vers">At&acirc;ta lini&#351;te-i &icirc;n jur de-mi pare c&#259;
aud<br />
cum se izbesc de geamuri razele de lun&#259;.<br />
&Icirc;n piept<br />
mi s-a trezit un glas str&#259;in<br />
&#351;i-un c&acirc;ntec c&acirc;nta-n mine-un dor<br />
ce nu-i al meu.</span><br /><br />
</body>
</html>

```

**Fișierul StilBlaga.css, ce conține stilurile necesare afișării plăcute a versurilor.**

```

body
{
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #006600;
    font-size: 12px;
}

.b
{
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #006600;
    background-color: #C1FFC1;
    height: 20px;
    width: 20px;
    border-top-color: #6FFF6F;
    border-right-color: #6FFF6F;
    border-bottom-color: #6FFF6F;
    border-left-color: #6FFF6F;
}

.titlul
{
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #006600;
    font-size: 14px;
    font-weight: bolder;
}

.vers
{
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #006600;
    font-size: 12px;
    font-style: italic;
}

```

## Applet

Obiectul Applet permite includerea applet-urilor Java într-o pagină web.  
 Proprietăți: *toate proprietățile publice ale unui applet Java.*  
 Metode: *toate metodele publice ale unui applet Java.*

## Area

Obiectul Area definește o suprafață dintr-o pagină Web. Suprafața poate fi aria unei imagini sau a unei hărți de imagini. Atunci când utilizatorul dă click pe acea suprafață delimitată, este încărcată o nouă fereastră sau pagină Web.  
 Proprietăți: *length.*

| Proprietate | Descriere                                     |
|-------------|---|
| length      | Reprezintă numărul de argumente ale funcției. |

**Exemplu** În acest exemplu, am utilizat obiectul Area, pentru a defini trei suprafețe: zona capului, zona cărților și zona fetiței. Zona fetiței cuprinde celelalte două zone. În momentul în care utilizatorul trece cu mouse-ul pe deasupra zonelor definite sau părăsește aceste zone, vor apărea diverse mesaje scrise într-o zonă de tip text și în bara de stare.

Am utilizat pentru delimitare cele trei tipuri de suprafețe:

- shape="circle" COORDS="44,16,15" -> suprafața unui cerc
- shape="rect" coords="40,40,80,75" -> suprafață dreptunghiulară
- shape="poly" coords="25,1,25,135,75,135,80,1" -> suprafață poligonală



file:///D:/3ian/Carte%20HTML+JavaScript/Carti

```
<html>
<head>
  <title>Obiectul Area</title>
<script type="text/javascript" language="javascript">
function f(x)
{
  switch (x)
  {
    case 1:
    {
      self.status='Sunteti in zona capului';
      formular.text.value='Sunteti in zona capului';
      return true;
    }; break;
    case 2:
    {
      self.status='Sunteti in zona cartilor';
      formular.text.value='Sunteti in zona cartilor';
      return true;
    }; break;
    case 3:
    {
```

```

        self.status='Sunteti in zona fetitei';
        formular.text.value='Sunteti in zona fetitei';
        return true;
    }; break;
default:
    {
        self.status='Nu sunteti in zona imaginii';
        formular.text.value='Nu sunteti in zona imaginii';
        return true;
    };
}
}
function g(y)
{
    switch (y)
    {
        case 1:
            { self.status='Ati parasit zona capului';
              formular.text.value='Ati parasit zona capului';
              return true;
            }; break;
        case 2:
            { self.status='Ati parasit zona cartilor';
              formular.text.value='Ati parasit zona cartilor';
              return true;
            }; break;
        case 3:
            { self.status='Ati parasit zona fetitei';
              formular.text.value='Ati parasit zona fetitei';
              return true;
            }; break;
        default:
            {
                self.status='Nu sunteti in zona imaginii';
                formular.text.value='Nu sunteti in zona imaginii';
                return true;
            };
    }
}
</script>
</head>
<body>
<form name="formular">
<MAP name="O_fetita_cu_carti">
<AREA name="cap" shape="circle" COORDS="44,16,15" HREF="Cap" alt="cap"
    onMouseOver="f(1)" onMouseOut="g(1)">
<AREA name="carti" coords="40,40,80,75" href="Carti" alt="carti"
    onMouseOver="f(2)" onMouseOut="g(2)">
<AREA name="fetita" shape="poly" coords="25,1,25,135,75,135,80,1" href="Fetita"
    alt="o fetita cu carti" onMouseOver="f(3)" onMouseOut="g(3)">
</MAP>
<IMG SRC="Printesa.jpg" align="top" height="140" width="100"
USEMAP="#O_fetita_cu_carti" > <br />
<input type="text" name="text" value="" size="25">
</form>
</body>
</html>

```

## Array

Obiectul Array seamănă cu noțiunea Array (tablou), întâlnită mai des în programare. Un tablou poate fi unidimensional, atunci când unei singure variabile i se asociază un set de valori de același tip, sau cu mai multe dimensiuni. În JavaScript discutăm despre array, ca despre un tablou unidimensional, care are mai multe proprietăți și metode.

Proprietăți: *constructor*, *index*, *input*, *length*, *prototype*.

Metode: *concat*, *join()*, *pop()*, *push()*, *reverse()*, *shift()*, *slice()*, *splice()*, *sort()*, *toSource()*, *toString()*, *unshift()*, *valueOf()*.

Pentru a înțelege mai bine proprietățile și metodele, am să le explic în câteva cuvinte, în tabelele următoare:

| Proprietate | Descriere  |
|-------------|--|
| constructor | Specifică funcția care crează prototype unui obiect.         |
| index       | Pentru un Array dintr-o expresie, șirul începe de la zero.   |
| input       | Specifică șirul original.                                    |
| length      | Numărul elementelor dintr-un Array.                          |
| prototype   | Permite adăugarea unor proprietăți obiectului <i>Array</i> . |

| Metodă     | Descriere  |
|------------|--|
| concat()   | Lipește două <i>Array</i> într-unul singur.  |
| join()     | Alătură toate elementele dintr-un <i>Array</i> , într-un string.   |
| pop()      | Înlătură ultimul element dintr-un <i>Array</i> și returnează acel element.   |
| push()     | Adaugă unul sau mai multe elemente la sfârșitul unui <i>Array</i> și returnează o nouă lungime.                          |
| reverse()  | Inversează elementele unui <i>Array</i> , în așa fel încât primul va deveni ultimul și ultimul va deveni primul.         |
| shift()    | Înlătură primul element dintr-un <i>Array</i> și returnează acel element.  |
| slice()    | Extrage o porțiune dintr-un <i>Array</i> și returnează un nou <i>Array</i> .   |
| splice()   | Adaugă sau înlătură elemente dintr-un <i>Array</i> .   |
| sort()     | Sortează elementele dintr-un <i>Array</i> .  |
| toSource() | Returnează un șir ce reprezintă codul sursă al unui <i>Array</i> .   |
| toString() | Returnează un șir ce reprezintă un <i>Array</i> și elementele sale.  |
| unshift()  | Adaugă unul sau mai multe elemente în prima parte a unui <i>Array</i> și returnează noua lungime a acelui <i>Array</i> . |
| valueOf()  | Returnează prima valoare a unui <i>Array</i> .   |

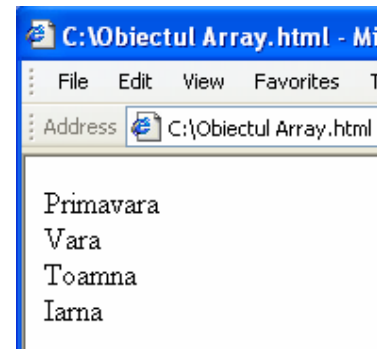
**Exemplu** În acest exemplu, am utilizat obiectul *Array()*, pentru a reține cele 4 anotimpuri. Variabila *anotimpuri*, datorită faptului că am declarat-o *Array*, poate reține mai multe valori, cum ar fi: « Primavara », « Vara », « Toamna » și « Iarna ». Prima valoare pe care o poate reține, se va duce pe prima poziție, adică *anotimpuri[0]*. (Pentru cei care cunosc limbajele C sau Java, este mult mai ușor). Dacă nu cunoaștem numărul de valori reținute în *anotimpuri*, putem utiliza *anotimpuri.length* care ne va returna numărul de elemente reținute în *anotimpuri*.

```
<html>
<head>
  <title>Obiectul Array</title>
<script type="text/javascript" language=
"javascript">
function f()
{
  var anotimpuri = new Array()
  anotimpuri[0] = "Primavara"
```

```

    anotimpuri[1] = "Vara"
    anotimpuri[2] = "Toamna"
    anotimpuri[3] = "Iarna"
    for (i=0;i<anotimpuri.length;i++)
    {
        document.write(anotimpuri[i] + "<br />")
    }
}
</script>
</head>
<body onLoad="f()" >
</body>
</html>

```



**Exemplu** Același exemplu, ca cel de mai sus, dar puțin mai complicat, în sensul că am utilizat funcția *g()* pentru a pune în valoare metoda *sort()* și funcția *h()* pentru a pune în valoare metoda *concat()*. Fiecare dintre metodele de mai sus, pot fi utilizate cu obiectul *Array()*.

```

<html>
<head>
  <title>
    Obiectul Array
  </title>
<script type="text/javascript" language="javascript">
var anotimpuri = new Array()
anotimpuri[0] = "Primavara"
anotimpuri[1] = "Vara"
anotimpuri[2] = "Toamna"
anotimpuri[3] = "Iarna"
var culori=new Array("Alb", "Negru",
"Rosu");
function f()
{
  for (i=0;i<anotimpuri.length;i++)
  {
    document.write(anotimpuri[i]);
    if (i<anotimpuri.length-1)
      document.write(", ");
  }
}
function g()
{
  document.write(anotimpuri.sort());
}
function h()
{
  document.write(anotimpuri.concat(culori));
}
</script>
</head>
<body>
<form name="formular" method="post"
action="" >
Anotimpurile.<br />
  <input type="submit" name="x1"
value="Apasati!" onclick="f()"><br />
Anotimpurile in ordine alfabetica.<br />
  <input type="submit" name="x2"
value="Apasati!" onclick="g()"><br />
Denumirile anotimpurilor, lipite de denumirile culorilor.
  <input type="submit" name="x3"
value="Apasati!" onclick="h()">

```

Anotimpurile.

Apasati!

Anotimpurile in ordine alfabetica.

Apasati!

Denumirile anotimpurilor, lipite de denumirile culorilor.

Apasati!

Dacă apăsați primul buton, se va afișa:

Primavara, Vara, Toamna, Iarna

Dacă apăsați al doilea buton, se va afișa:

Iarna,Primavara,Toamna,Vara

Dacă apăsați al treilea buton, se va afișa:

Primavara,Vara,Toamna,Iarna,Alb,Negru,Rosu



```



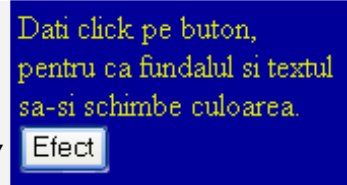
```

**Exemplu** În acest exemplu, am utilizat obiectul *Array()*, în care am introdus perechi de numere ce reprezintă porțiuni din codul hexazecimal, ce reprezintă o culoare.

```

<html>
<head>
<title>Exemplu_Array</title>
<script type="text/javascript" language="javascript">
var culoare= new Array("00","11","22","33","44","55",
"66","77","88","99","AA","BB","CC","DD","EE","FF");
function efect_fundal(x)
{
if (x >= 1)
{
document.bgColor="#"+"00"+culoare[x];
document.fgColor="#"+"culoare[x+8]+"0000";
x -= 1;
setTimeout("efect_fundal("+x+")", 50);
}
else
{
setTimeout('efect_invers(1)', 50);
}
}
function efect_invers(x)
{
if (x <=16)
{
document.bgColor="#"+"0000"+culoare[x];
document.fgColor="#"+"culoare[x+8]+"0000";
x += 1;
setTimeout("efect_invers("+x+")", 50)
}
else
{
setTimeout("efect_fundal(16)", 50);
}
}
</script>
</head>
<body >
<form>
Dati click pe buton, <br />
pentru ca fundalul si textul <br />
sa-si schimbe culoarea.<br />
<input name="" type="button" onClick="efect_fundal(16)"
value="Efect">
</form>
</body>
</html>

```



## Boolean

Obiectul Boolean este utilizat pentru a transforma o variabilă care nu aparține acestui tip într-o valoare de tip boolean. O variabilă de tip boolean poate avea două valori: true sau false.

Proprietăți: *constructor*, *prototype*.

Metode: *toSource()*, *toString()*, *valueOf()*.

| Proprietate | Descriere  |
|-------------|--|
| constructor | Specifică funcția care creează prototype unui obiect.          |
| prototype   | Permite adăugarea unor proprietăți obiectului <i>Boolean</i> . |

| Metodă            | Descriere  |
|-------------------|--|
| <i>toSource()</i> | Returnează un șir care reprezintă codul sursă al unui <i>Array</i> . |
| <i>toString()</i> | Returnează un șir ce reprezintă un <i>Array</i> și elementele sale.  |
| <i>valueOf()</i>  | Returnează prima valoare a obiectului <i>Boolean</i> .               |

**Exemplu** În acest exemplu, am utilizat obiectul *Boolean()*, pentru a evidenția elementele pentru care se obțin valorile de adevăr true și false.

```
<html>
<head>
  <title>
    Obiectul Boolean
  </title>
<script      type="text/javascript"      language=
"JavaScript">
x1 = new Boolean();
x2 = new Boolean(0);
x3 = new Boolean(null);
x4 = new Boolean("");
x5 = new Boolean(false);
document.writeln("new Boolean()= "+x1+"<br />");
document.writeln("new Boolean(0)= "+x2+"<br />");
document.writeln("new Boolean(null)= "+x3+" <br
/>");
document.writeln('new Boolean("")= '+x4+"<br />");
document.writeln("new Boolean(false)= "+x5+" <br
/>"+<br />");
y1 = new Boolean(true);
y2 = new Boolean(1);
y3 = new Boolean("true");
y4 = new Boolean("false");
y5 = new Boolean("Diana");
document.writeln("new Boolean(true)= "+y1+"
<br />");
document.writeln('new Boolean(1)= '+y2+"<br />");
document.writeln('new Boolean("true")= '+y3+"
<br />");
document.writeln('new Boolean("false")= '+y4+"
<br />");
document.writeln('new Boolean("Diana")= '+y5+"
<br />");
</script>
</head>
<body>
</body>
</html>
```

```
new Boolean()= false
new Boolean(0)= false
new Boolean(null)= false
new Boolean("")= false
new Boolean(false)= false
new Boolean(true)= true
new Boolean(1)= true
new Boolean("true")= true
new Boolean("false")= true
new Boolean("Diana")= true
```

## Button

Obiectul Button este un buton care poate fi acționat de către evenimentele asociate din formularul HTML și care execută un anumit script specificat.

Proprietăți: *form, name, type, value*.

Metode: *blur(), click(), focus()*.

Evenimente: *onBlur, onClick, onFocus*.

| Proprietate | Descriere  |
|-------------|--|
| form        | Obiectul form care include butonul.                        |
| name        | Numele butonului.  |
| type        | Tipul obiectului, care poate fi: Button, Submit sau Reset. |
| value       | Valoarea butonului, care apare afișată pe buton.           |

| Metodă  | Descriere   |
|---------|---|
| blur()  | Înlătură focalizarea de pe obiectul specificat.   |
| click() | Această funcție acționează dacă utilizatorul dă click pe obiectul specificat dintr-un formular. |
| focus() | Dă focalizare asupra unui obiect specificat.  |

| Eveniment | Descriere  |
|-----------|--|
| onBlur    | Execută codul JavaScript în momentul în care se înlătură focalizarea de pe obiectul specificat.  |
| onClick   | Execută codul JavaScript în momentul în care efectuăm click de mouse pe obiectul specificat dintr-un formular.                                     |
| onFocus   | Execută codul JavaScript în momentul în care are loc evenimentul focus, adică atunci când o fereastră, un frame sau un frameset primește un focus. |

**Exemplu** În acest exemplu, am utilizat obiectul Button, pentru a realiza un test cu mai multe variante de răspuns corecte.



```
<html>
<head>
  <title>Obiectul Button</title>
<style type="text/css">
body
{
  font-family: Geneva, Arial, Helvetica, sans-serif;
  font-size: 18px;
  color: #0000FF;
  background-color: #DDDDFF;
  font-weight: bolder;
}
.buton
{
  color: #0000FF;
  background-color: #DDDDFF;
  font-family: Geneva, Arial, Helvetica, sans-serif;
  font-size: 12px
}
```

```

</style>
<script type="text/javascript" language="JavaScript">
function Opreste ()
{
    document.muzica.stop()
}
function Porneste ()
{
    document.muzica.play()
}
function schimba ()
{
    if (document.formular.B1.value=='Opreste muzica')
    {
        document.formular.B1.value='Porneste muzica';
        Opreste()
    }
    else
    {
        document.formular.B1.value='Opreste muzica';
        Porneste()
    }
}
</script>
</head>
<body>
<embed name="muzica" src="RichardClayderman.mp3" width="128" height="128"
    hidden> <br />
<form name="formular">
    Richard Claiderman<br /><br />
<input type="button" value="Opreste muzica" name="B1" onClick="schimba()"
    class="buton">
</form>
</body>
</html>

```

## Checkbox

Obiectul Checkbox este o casetă de validare ce permite utilizatorului să o selecteze prin executarea unui click de mouse în interiorul ei. Caseta va avea valoarea true sau false în funcție de opțiunea utilizatorului.

Proprietăți: *checked*, *defaultChecked*, *form*, *name*, *type*, *value*.

Metode: *blur()*, *click()*, *focus()*.

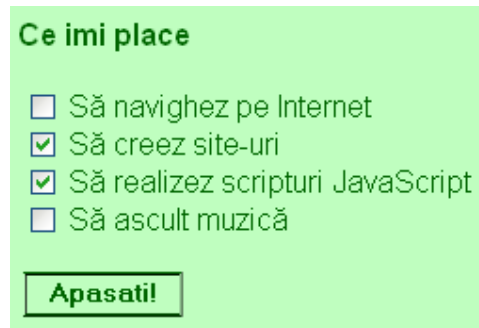
Evenimente: *onBlur*, *onClick*, *onFocus*.

| Proprietate    | Descriere   |
|----------------|---|
| checked        | Returnează valoarea logică <i>true</i> dacă este selectat un obiect checkbox, altfel returnează <i>false</i> .                      |
| defaultChecked | Returnează valoarea logică <i>true</i> dacă starea inițială a unui obiect checkbox este selectată, altfel returnează <i>false</i> . |
| form           | Specifică referința unui obiect dintr-un formular ce conține un buton.  |
| name           | Reprezintă un șir ce specifică valoarea atributului <i>name</i> .   |
| type           | Specifică tipul elementelor unui formular.  |
| value          | Valoarea butonului, care apare afișată.   |

| Metodă | Descriere |
|--------|-----------|
|--------|-----------|

|                  |  |
|------------------|--|
| blur()           | Înlătură focalizarea de pe checkbox dintr-un formular.   |
| click()          | Această funcție acționează dacă utilizatorul dă click pe butonul checkbox dintr-un formular.   |
| focus()          | Dă focus asupra unui checkbox dintr-un formular.   |
| <b>Eveniment</b> | <b>Descriere</b>   |
| onBlur           | Execută codul JavaScript în momentul în care se înlătură focalizarea de pe obiectul specificat.  |
| onClick          | Execută codul JavaScript în momentul în care efectuăm click de mouse pe obiectul specificat dintr-un formular.                                     |
| onFocus          | Execută codul JavaScript în momentul în care are loc evenimentul focus, adică atunci când o fereastră, un frame sau un frameset primește un focus. |

**Exemplu** În acest exemplu, am realizat un formular care selectează toate butoanele de validare checkbox.



```

<html>
<head>
  <title>
    Obiectul Checkbox
  </title>
<style type="text/css">
body
{
  font-family: Geneva, Arial, Helvetica, sans-serif;
  font-size: 12px;
  color: #006600;
  background-color: #BFFFBF;
}
.titlul
{
  font-size: 16px;
  font-weight: bolder;
  text-align: center;
}
.buton
{
  font-size: 14px;
  color: #006600;
  font-weight: bolder;
  background-color: #BFFFBF;
  border-top-width: thin;
  border-right-width: thin;
  border-bottom-width: thin;
  border-left-width: thin;
  border-top-style: ridge;
  border-right-style: ridge;
  border-bottom-style: ridge;

```

```

border-left-style: ridge;
border-top-color: #006600;
border-right-color: #BFFFBF;
border-bottom-color: #006600;
border-left-color: #BFFFBF;
}
</style>
<script type="text/javascript" language="JavaScript">

var verificare = "false";
function f(x)
{
    if (verificare == "false")
    {
        for (i = 0; i < x.length; i++)
        {
            x[i].checked = true;
        }
        verificare = "true";
        return "Deselectate";
    }
    else
    {
        for (i = 0; i < x.length; i++)
        {
            x[i].checked = false;
        }
        verificare = "false";
        return "Selectate";
    }
}
</script>
</head>
<body>
<span class="titlul">Ce imi place
</span><br /><br />
<table>
<tr>
<td>
<form name="formular" >
    <input type="checkbox" name="valoare" value="1"> S&#259; navighez pe
        Internet<br />
    <input type="checkbox" name="valoare" value="2">
        S&#259; creez site-uri<br />
    <input type="checkbox" name="valoare" value="3">
        S&#259; realizez scripturi JavaScript<br />
    <input type="checkbox" name="valoare" value="4"> S&#259; ascult
        muzic&#259;<br /><br />
    <input type="button" value="Apasati!" onClick="this.value=
        f(this.form.valoare)" class="buton">
</form>
</td>
</tr>
</table>
</body>
</html>

```

## Date

Obiectul Date este utilizat în lucrul cu ore și date calendaristice. Atunci când lucrăm cu acest obiect, data și ora computerului client sunt cele la care se raportează scriptul JavaScript când este executat. Data este măsurată în milisecunde începând cu 1 ianuarie 1970. O zi are 86.400.000 milisecunde. Pentru a realiza calcule corecte, trebuie să specificăm anul în întregime, nu prescurtat, adică 1989 nu 89.

Proprietăți: *constructor*, *prototype*.

Metode: *getDate()*, *getDay()*, *getFullYear()*, *getHours()*, *getMilliseconds()*, *getMinutes()*, *getMonth()*, *getSeconds()*, *getTime()*, *getTimezoneOffset()*, *getUTCDate()*, *getUTCDay()*, *getUTCFullYear()*, *getUTCHours()*, *getUTCMilliseconds()*, *getUTCMinutes()*, *getUTCMonth()*, *getUTCSeconds()*, *getYear()*, *parse()*, *setDate()*, *setFullYear()*, *setHours()*, *setMilliseconds()*, *setMinutes()*, *setMonth()*, *setSeconds()*, *setTime()*, *setUTCDate()*, *setUTCFullYear()*, *setUTCHours()*, *setUTCMilliseconds()*, *setUTCMinutes()*, *setUTCMonth()*, *setUTCSeconds()*, *setYear()*, *toGMTString()*, *toLocaleString()*, *toSource()*, *toString()*, *toUTCString()*, *UTC()*, *valueOf()*.

| Proprietate | Descriere   |
|-------------|---|
| constructor | Specifică funcția care creează prototype unui obiect. |
| prototype   | Permite adăugarea unor proprietăți obiectului Date.   |

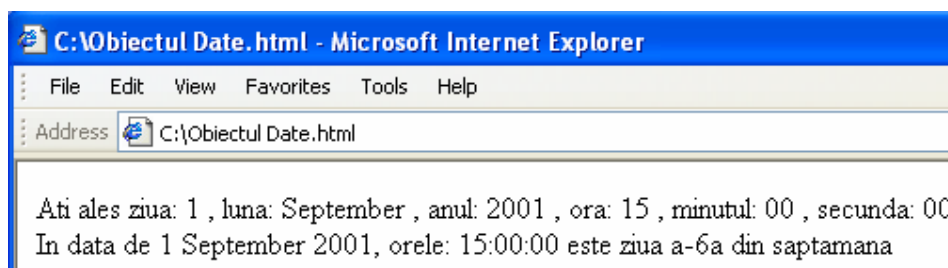
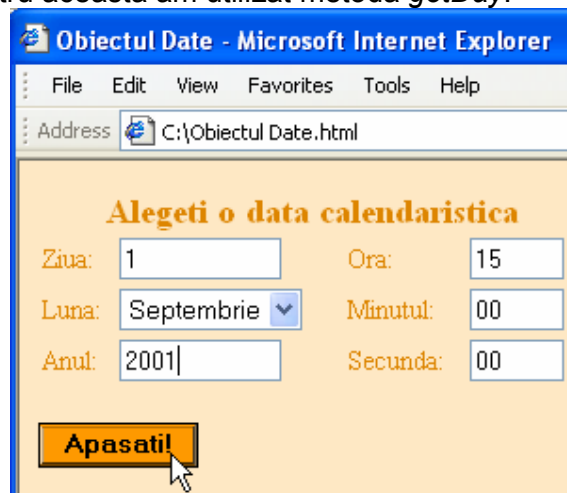
| Metodă                     | Descriere  |
|----------------------------|--|
| <i>getDate()</i>           | Returnează o valoare numerică din intervalul 1-31, ce reprezintă ziua dintr-o dată specificată, conform timpului local.                            |
| <i>getDay()</i>            | Returnează o valoare numerică din intervalul 0-6, ce corespunde unei zile din săptămână, dintr-o dată specificată, conform timpului local.         |
| <i>getFullYear()</i>       | Returnează anul întreg dintr-o dată specificată, conform timpului local.   |
| <i>getHours()</i>          | Returnează o valoare numerică din intervalul 0-23, ce corespunde unei ore din zi, dintr-o dată specificată, conform timpului local.                |
| <i>getMilliseconds()</i>   | Returnează o valoare numerică din intervalul 0-999, ce corespunde unei milisecunde, dintr-o dată specificată, conform timpului local.              |
| <i>getMinutes()</i>        | Returnează o valoare numerică din intervalul 0-59, ce corespunde unui minut dintr-o oră, dintr-o dată specificată, conform timpului local.         |
| <i>getMonth()</i>          | Returnează o valoare numerică din intervalul 0-11, ce corespunde unei luni din an, dintr-o dată specificată, conform timpului local.               |
| <i>getSeconds()</i>        | Returnează o valoare numerică din intervalul 0-59, ce corespunde unei secunde dintr-un minut, dintr-o dată specificată, conform timpului local.    |
| <i>getTime()</i>           | Returnează o valoare numerică în milisecunde începând din 1 ianuarie 1970 00:00:00, conform timpului local.  |
| <i>getTimezoneOffset()</i> | Returnează diferența între GMT și ora locală.  |
| <i>getUTCDate()</i>        | Returnează o valoare numerică din intervalul 1-31, ce corespunde unei zile din lună, dintr-o dată specificată, conform timpului universal.         |
| <i>getUTCDay()</i>         | Returnează o valoare numerică din intervalul 0-6, ce corespunde unei zile dintr-o săptămână, dintr-o dată specificată, conform timpului universal. |

|                      |   |
|----------------------|---|
| getUTCFullYear()     | Returnează un număr absolut corespunzător unui an, dintr-o dată specificată, conform timpului universal.  |
| getUTCHours()        | Returnează o valoare numerică din intervalul 0-23, ce corespunde unei ore dintr-o zi, dintr-o dată specificată, conform timpului universal.   |
| getUTCMilliseconds() | Returnează o valoare numerică din intervalul 0-999, ce corespunde unei milisecunde, dintr-o dată specificată, conform timpului universal.   |
| getUTCMinutes()      | Returnează o valoare numerică din intervalul 0-59, ce corespunde unui minut, dintr-o oră, dintr-o dată specificată, conform timpului universal.                                       |
| getUTCMonth()        | Returnează o valoare numerică din intervalul 0-11, ce corespunde unei luni, dintr-un an, dintr-o dată specificată, conform timpului universal.  |
| getUTCSeconds()      | Returnează o valoare numerică din intervalul 0-59, ce corespunde unei secunde, dintr-un minut, dintr-o dată specificată, conform timpului universal.                                  |
| getYear()            | Returnează anul dintr-o dată specificată, conform timpului local.   |
| parse()              | Returnează un număr de milisecunde dintr-o dată, ca diferență dintre <i>1 ianuarie 1970 00:00:00</i> și data respectivă.  |
| setDate()            | Returnează ziua din lună pentru o dată specificată, conform timpului local.   |
| setFullYear()        | Setează un an întreg pentru o dată specificată, conform timpului local.   |
| setHours()           | Setează ora pentru o dată specificată, conform timpului local.  |
| setMilliseconds()    | Setează milisecundele pentru o dată specificată, conform timpului local.  |
| setMinutes()         | Setează minutele pentru o dată specificată, conform timpului local.   |
| setMonth()           | Setează luna pentru o dată specificată, conform timpului local.   |
| setSeconds()         | Setează secundele pentru o dată specificată, conform timpului local.  |
| setTime()            | Setează valoarea obiectului <i>Date</i> , conform timpului local. Se utilizează metoda <i>setTime</i> pentru a ajuta la atribuirea datei și a timpului către alt obiect <i>Date</i> . |
| setUTCDate()         | Setează ziua dintr-o lună pentru o dată specificată, conform timpului universal.  |
| setUTCFullYear()     | Setează un an întreg pentru o dată specificată, conform timpului universal.   |
| setUTCHours()        | Setează ora pentru o dată specificată, conform timpului universal.  |
| setUTCMilliseconds() | Setează milisecundele dintr-o dată specificată, conform timpului universal.   |
| setUTCMinutes()      | Setează minutele dintr-o dată specificată, conform timpului universal.  |
| setUTCMonth()        | Setează luna dintr-o dată specificată, conform timpului universal.  |
| setUTCSeconds()      | Setează secundele dintr-o dată specificată, conform timpului universal.   |
| setYear()            | Setează anul pentru o dată specificată, conform timpului local.   |
| toGMTString()        | Transformă o dată într-un șir de caractere, folosind convențiile GMT.   |



|                      |   |
|----------------------|---|
| toLocaleString()     | Transformă o dată într-un șir de caractere, folosind convențiile locale curente.  |
| toLocaleDateString() | Returnează o porțiune <i>date</i> dintr-un șir de caractere al unui obiect <i>Date</i> specificat, folosind convențiile locale curente. |
| toLocaleTimeString() | Returnează o porțiune <i>time</i> dintr-un șir de caractere al unui obiect <i>Date</i> specificat, folosind convențiile locale curente. |
| toSource()           | Returnează un obiect literal reprezentând un obiect <i>Date</i> specificat.   |
| toString()           | Returnează un șir de caractere al unui obiect <i>Date</i> specificat.   |
| toUTCString()        | Returnează un șir de caractere al unui obiect <i>Date</i> specificat, folosind convenția UTC pentru timpul universal.                   |
| UTC()                | Returnează un număr în milisecunde dintr-o dată folosind timpul GMT.  |
| valueOf()            | Returnează valoarea primitivă a unui obiect <i>Date</i> .   |

**Exemplu** În acest exemplu, am realizat un formular care permite introducerea unor numere ce reprezintă o dată aleasă de către utilizator și returnează ziua din săptămâna respectivă. Pentru aceasta am utilizat metoda *getDay*.



```
<html>
<head>
  <title>Obiectul Date</title>
<style type="text/css">
body
{
  background-color: #FFE8C4;
}
.titlul
{
  font-size: 18px;
  font-weight: bold;
}
```

```

    color: #DD8500;
    text-align: center;
}
.text
{
    font-size: 14px;
    color: #DD8500;
}
.buton
{
    font-size: 14px;
    font-weight: bolder;
    background-color: #FF9900;
    border-top-width: thin;
    border-right-width: thin;
    border-bottom-width: thin;
    border-left-width: thin;
    border-top-style: outset;
    border-right-style: outset;
    border-bottom-style: outset;
    border-left-style: outset;
    border-top-color: #FF9900;
    border-right-color: #FFDBA6;
    border-bottom-color: #FF9900;
    border-left-color: #FFDBA6;
}
</style>
<script type="text/javascript" language="JavaScript">
function f()
{
var x1=formular.zi.value;
var x2=formular.luna.value;
var x3=formular.an.value;
var x4=formular.ora.value;
var x5=formular.minut.value;
var x6=formular.secunda.value;
//ziua
if(isNaN(x1))
    {
        alert(x1+": Ar trebui sa introduceti un numar!");
    }
else
    if ((x1>=1)&&(x1<=31))
        document.writeln("Ati ales ziua: "+x1);
    else
        document.writeln("Introduceti un numar intre 1 si 31.");
//luna
document.writeln(", luna: "+x2);
//anul
if(isNaN(x3))
    {
        alert(x3+": Ar trebui sa introduceti un numar!");
    }
else
if (x3>=1970)
    document.writeln(", anul: "+x3);
else
    document.writeln("Introduceti un numar pozitiv.");
//ora

```

```

if(isNaN(x4))
    {
        alert(x4+": Ar trebui sa introduceti un numar!");
    }
else
if ((x4>=0)&&(x4<=23))
    document.writeln(", ora: "+x4);
else
    document.writeln("Introduceti un numar intre 0 si 23.");
//minut
if(isNaN(x5))
    {
        alert(x5+": Ar trebui sa introduceti un numar!");
    }
else
if ((x5>=0)&&(x5<=60))
    document.writeln(", minutul: "+x5);
else
    document.writeln("Introduceti un numar intre 0 si 60.");
//secunda
if(isNaN(x6))
    {
        alert(x6+": Ar trebui sa introduceti un numar!");
    }
else
if ((x6>=0)&&(x6<=60))
    document.writeln(", secunda: "+x6);
else
    document.writeln("Introduceti un numar intre 0 si 60.");
//afişare
var ziua=new Date(x2+" "+x1+", "+x3+" "+x4+": "+x5+": "+x6);
document.writeln("<br />In data de "+x1+" "+x2+" "+x3+", orele:
    "+x4+": "+x5+": "+x6);
document.writeln(" este ziua a-"+ziua.getDay()+"a din saptamana");
}
</script>
</head>
<body>
<form name="formular" method="post" action="">
    <table width="30%" border="0" cellspacing="1" cellpadding="1" >
        <tr>
            <td colspan="4" class="titlul">Alegeti o data calendaristica</td>
        </tr>
        <tr >
            <td class="text">Ziua:</td>
            <td><input type="text" name="zi" size="10" value=""></td>
            <td class="text">Ora:</td>
            <td><input type="text" name="ora" size="4" ></td>
        </tr>
        <tr>
            <td class="text">Luna:</td>
            <td><select name="luna">
                <option selected value="January">Ianuarie</option>
                <option value="February">Februarie</option>
                <option value="March">Martie</option>
                <option value="April">Aprilie</option>
                <option value="May">Mai</option>
                <option value="June">Iunie</option>
                <option value="July">Iulie</option>
            </td>
        </tr>
    </table>

```

```

        <option value="August">August</option>
        <option value="September">Septembrie</option>
        <option value="October">Octombrie</option>
        <option value="November">Noiembrie</option>
        <option value="December">Decembrie</option>
    </select></td>
    <td class="text">Minutul:</td>
    <td><input type="text" name="minut" size="4"></td>
</tr>
<tr>
    <td class="text">Anul:</td>
    <td><input type="text" name="an" size="10"></td>
    <td class="text">Secunda:</td>
    <td><input type="text" name="secunda" size="4"></td>
</tr>
</table>
<br />
<input name="Submit" type="submit" class="buton" onClick="f()"
    value="Apasati!">
</form>
</body>
</html>

```

## Document

Obiectul Document conține informații despre documentul curent.

Proprietăți: *alinkColor*, *anchors*, *bgColor*, *cookie*, *domain*, *embeds*, *fgColor*, *formName*, *forms*, *height*, *ids*, *images*, *lastModified*, *layers*, *linkColor*, *links*, *plugins*, *referrer*, *tags*, *title*, *URL*, *vlinkColor*, *width*.

Metode: *close()*, *getElementById()*, *open()*, *write()*, *writeln()*.

Evenimente: *onClick*, *ondblclick*, *onkeydown*, *onkeypress*, *onkeyup*, *onmousedown*, *onmouseup*.

| Proprietate         | Descriere   |
|---------------------|---|
| <i>alinkColor</i>   | Reprezintă un șir de caractere care specifică ce culoare are link-ul activ.               |
| <i>anchors</i>      | Reprezintă un masiv de obiecte ce corespunde numelui ancorei.                             |
| <i>bgColor</i>      | Reprezintă un șir de caractere ce setează culoarea fundalului documentului.               |
| <i>cookie</i>       | Este utilizat pentru a identifica valoarea unui cookie.                                   |
| <i>domain</i>       | Reprezintă numele de domeniu al unui document de pe un server.                            |
| <i>embeds</i>       | Un array ce conține toate plugin-urile dintr-un document.                                 |
| <i>fgColor</i>      | Reprezintă un șir de caractere care specifică ce culoare are textul dintr-un document.    |
| <i>formName</i>     | Reprezintă numele unui formular.  |
| <i>forms</i>        | Reprezintă un masiv ce conține o intrare pentru fiecare formular din document.            |
| <i>height</i>       | Reprezintă un șir ce specifică înălțimea unei imagini, în pixeli.                         |
| <i>lastModified</i> | Reprezintă un șir ce reprezintă data când a fost modificat documentul pentru ultima oară. |
| <i>linkColor</i>    | Reprezintă culoarea legăturilor de tip hipertext nevizitate dintr-un document.            |
| <i>links[]</i>      | Un array ce conține toate legăturile de tip hipertext dintr-un document.                  |

|            |   |
|------------|---|
| plugins    | Un array ce conține toate plugins dintr-un document.  |
| referrer   | Un string ce specifică URL-ul în care utilizatorul ajunge printr-o legătură de tip hipertext. |
| title      | Specifică titlul unui document  |
| URL        | Un string ce specifică un URL complet dintr-un document.                                      |
| vlinkColor | Specifică culoarea legăturilor de tip hipertext vizitate dintr-un document.                   |

| Metodă               | Descriere   |
|----------------------|---|
| close()              | Închide documentul deschis cu document.open().  |
| getElementById("ID") | O metodă de accesare a unui element dintr-o pagină prin atributul "ID".                           |
| open()               | Deschide o fereastră nouă.  |
| write()              | Realizează scrierea într-un document HTML a unui șir de caractere.                                |
| writeln()            | Realizează scrierea într-un document HTML a unui șir de caractere, după care trece la linie nouă. |

| Eveniment   | Descriere   |
|-------------|---|
| onClick     | Execută codul JavaScript în momentul în care efectuăm click de mouse pe un obiect specificat. |
| onDbClick   | Ațiune ce va fi declanșată când se dă dublu click stânga de mouse pe un obiect specificat.    |
| onKeyDown   | Ațiune ce va fi declanșată când se apasă o tastă.   |
| onKeyPress  | Ațiune ce va fi declanșată când se apasă și se eliberează o tastă.                            |
| onKeyUp     | Ațiune ce va fi declanșată când este eliberată o tastă.                                       |
| onMouseDown | Ațiune ce va fi declanșată când un buton al mouse-ului este apăsat.                           |
| onMouseUp   | Ațiune ce va fi declanșată când butonul mouse-ului este eliberat.                             |

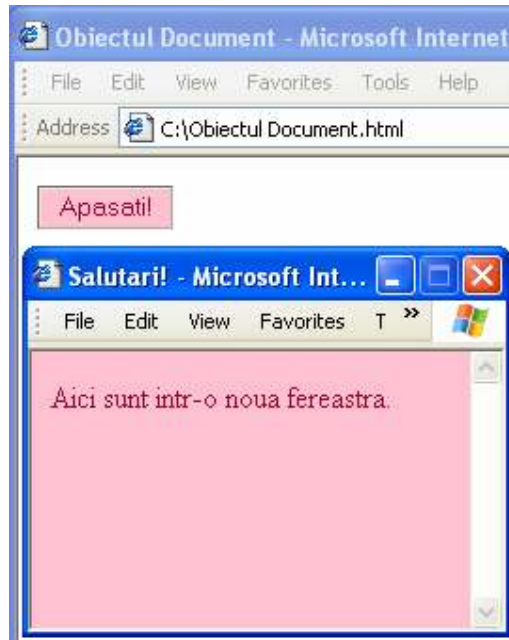
**Exemplu** În acest exemplu, am utilizat obiectul Document pentru a deschide o nouă fereastră în care am scris un mesaj. Fereastra am deschis-o cu *window.open* și am scris în ea cu *document.write*.

```
<html>
<head>
<title>Obiectul Document</title>
<style type="text/css">
.mesaj
{
font-size: 14px;
color: #A60029;
background-color: #FFC1D1;
border: thin groove;
}
</style>
</head>
<body>
<script type="text/javascript" language="JavaScript">
function f()
{ msgWindow=window.open ("", "displayWindow", "toolbar=no, width=250,
height=15,directories=no,status=no,scrollbars=yes,resize=yes,
```

```

    menubar=yes")
    msgWindow.document.write ("<head><title>Salutari! </title></head>")
    msgWindow.document.write ("<body bgcolor=#FFC1D1 text=#A60029>Aici sunt
    intr-o noua fereastră. ")
}
</script>
<form>
<INPUT type="button" value="Apasati!"
    onclick=f(this.form) name="buton" alt="mesaj" class="mesaj">
</form>
</body>
</html>

```



## Event

Obiectul Event este creat automat de JavaScript la întâlnirea unui eveniment.

Proprietăți: *altKey*, *ctrlKey*, *shiftKey*, *button*, *clientX*, *clientY*, *fromElement*, *toElement*, *keyCode*, *offsetX*, *offsetY*, *returnValue*, *srcElement*, *type*.

Metode: *preventDefault()*, *stopPropagation()*.

| Proprietate                                      | Descriere   |
|--|---|
| <i>altKey</i> , <i>ctrlKey</i> , <i>shiftKey</i> | Proprietăți ce indică momentul când sunt apăstate tastele Alt, Ctrl și Shift în timpul declanșării unui eveniment.  |
| <i>button</i>                                    | Un număr întreg ce arată când este apăsat sau eliberat un buton al mouse-ului. Numerele sunt 1 – pentru butonul din stânga, 2 – dreapta, 4 – mijloc. Dacă sunt apăstate mai multe butoane, valoarea rezultată este suma acelor butoane. De exemplu, 3 înseamnă că au fost apăstate butoanele 1 și 2, adică stânga și dreapta. |
| <i>clientX</i> , <i>clientY</i>                  | Returnează coordonatele mouse-ului.   |
| <i>fromElement</i> , <i>toElement</i>            | Pentru evenimentele <i>mouseover</i> și <i>mouseout</i> , aceste proprietăți indică momentul când elementele mouse-ului părăsesc sau se deplasează pe deasupra obiectului respectiv.  |
| <i>keyCode</i>                                   | Indică codul <i>Unicode</i> pentru tasta apăsată.   |
| <i>offsetX</i> , <i>offsetY</i>                  | Returnează coordonatele mouse-ului relativ la elementul de  |

|             |  |
|-------------|--|
|             | origine.   |
| returnValue | Setează <i>false</i> pentru a anula orice acțiune inițială a evenimentului.              |
| srcElement  | Elementul asupra căruia acționează evenimentul curent.                                   |
| type        | Un șir de caractere ce indică tipul evenimentului, cum ar fi : « click », « mouseover ». |

| Metodă            | Descriere  |
|-------------------|--|
| preventDefault()  | Setează <i>true</i> pentru a anula orice acțiune inițială a evenimentului. |
| stopPropagation() | Setează <i>true</i> pentru a preveni propagarea evenimentului.             |

**Exemplu** În acest exemplu, am realizat o bară verticală pe care glisează un dreptunghi. Pe măsură ce dreptunghiul își schimbă poziția, va fi afișată noua valoare.



```
<html>
<head>
  <title>Exemplu cu Event</title>
<style type="text/css">
*.Horizontal
{ color: #333;
  width: 120px;
  float: left;
  margin: 0;
  line-height: 0px;
  font-size: 0px;
  text-align: left;
  padding: 4px;
  border-top-width: 1px;
  border-right-width: 1px;
  border-bottom-width: 1px;
  border-left-width: 1px;
  border-top-style: solid;
  border-right-style: solid;
  border-bottom-style: solid;
  border-left-style: solid;
}
*.IntVertical
{ color: #333;
  padding: 3px 6px 15px 6px;
  width: 24px;
  height: 100px;
  border-top-width: 1px;
  border-right-width: 1px;
  border-bottom-width: 1px;
  border-left-width: 1px;
  border-top-style: solid;
  border-right-style: solid;
  border-bottom-style: solid;
  border-left-style: solid;
}

```

```

*.Horizontal *.Intr
{ background-color: #333;
  color: #ccc;
  width: 110px;
  height: 3px;
  line-height: 0px;
  position: absolute;
  z-index: 1;
  margin-top: 4px;
  margin-right: 4px;
  margin-bottom: 2px;
  margin-left: 4px;
  border-top-width: 1px;
  border-right-width: 1px;
  border-bottom-width: 1px;
  border-left-width: 1px;
  border-top-style: solid;
  border-right-style: solid;
  border-bottom-style: solid;
  border-left-style: solid;
  border-top-color: #aaa;
  border-right-color: #eee;
  border-bottom-color: #eee;
  border-left-color: #aaa;
}
*.IntVertical *.Intr
{ background-color: #000;
  color: #333;
  width: 2px;
  height: 100px;
  position: absolute;
  margin: 4px 10px 4px 10px;
  padding: 4px 0 1px 0;
  line-height: 0px;
  font-size: 0;
}
*.Horizontal *.bara
{ width: 16px;
  color: #333;
  position: relative;
  margin: 0;
  height: 9px;
  z-index: 1;
  text-align: left;
}
*.IntVertical *.bara
{ width: 20px;
  background-color: #666;
  color: #333;
  position: relative;
  margin: 0;
  height: 9px;
  z-index: 1;
  line-height: 0px;
  font-size: 0px;
  text-align: left;
  border-top-width: 2px;
  border-right-width: 2px;
  border-bottom-width: 2px;
}

```



```

border-left-width: 2px;
border-top-style: solid;
border-right-style: solid;
border-bottom-style: solid;
border-left-style: solid;
border-top-color: #aaa;
border-right-color: #444;
border-bottom-color: #444;
border-left-color: #aaa;
}
*.DrAfis
{ color: #333;
width: 35px;
margin: 0;
float: left;
height: 20px;
text-align: right;
padding-top: 0;
padding-right: 2px;
padding-bottom: 0;
padding-left: 0;
}
.AfisBara
{ color: #006600;
padding: 3px 1px 0 0;
width: 30px;
text-align: right;
font-size: 11px;
line-height: 10px;
font-family: verdana, arial, helvetica, sans-serif;
font-weight: bold;
border: 0;
cursor: default;
}
</style>
<script type="text/javascript" language="javascript">
var LungineB = 100
var Orientare = 'horizontal'
var NumeBara = 'bara'
var AfisBr = 'AfisBara'
function NumeStil(className)
{
var ElemStil = new Array();
var Xe = document.getElementsByTagName("*");
var LumgimeXe = Xe.length;
var Xa = new RegExp("\\b" + className + "\\b")
for (i = 0, j = 0; i < LumgimeXe; i++)
{
if ( Xa.test(Xe[i].className) )
{
ElemStil[j] = Xe[i];
j=j+1;
}
}
return ElemStil;
}
function Stanga(Xel, Xp)
{
if (!(Xel = document.getElementById(Xel)))

```

```

    return 0;
if (Xel.style && (typeof(Xel.style.left) == 'string'))
{
    if (typeof(Xp) == 'number')
        Xel.style.left = Xp + 'px';
    else
    {
        Xp = parseInt(Xel.style.left);
        if (isNaN(Xp))
            Xp = 0;
    }
}
else
if (Xel.style && Xel.style.pixelLeft)
{
    if (typeof(Xp) == 'number')
        Xel.style.pixelLeft = Xp;
    else
        Xp = Xel.style.pixelLeft;
}
return Xp;
}
function Sus(Xel, Xp)
{
    if (!(Xel = document.getElementById(Xel)))
        return 0;
    if (Xel.style && (typeof(Xel.style.top) == 'string'))
    {
        if (typeof(Xp) == 'number') Xel.style.top = Xp + 'px';
        else
        {
            Xp = parseInt(Xel.style.top);
            if (isNaN(Xp)) Xp = 0;
        }
    }
    else
    if (Xel.style && Xel.style.pixelTop)
    {
        if (typeof(Xp) == 'number')
            Xel.style.pixelTop = Xp;
        else
            Xp = Xel.style.pixelTop;
    }
    return Xp;
}
function Miscare(Eventiment)
{
    if (!Eventiment)
        Eventiment = window.event;
    else
        Eventiment = Eventiment;
    if (mouseover)
    {
        x = Xms.startOffsetX + Eventiment.screenX
        y = Xms.startOffsetY + Eventiment.screenY
        if (x > Xms.xMax)
            x = Xms.xMax
        if (x < 0)
            x = 0
    }
}

```

```

    if (y > Xms.yMax)
        y = Xms.yMax
    if (y < 0)
        y = 0
    Stanga(Xms.id, x)
    Sus(Xms.id, y)
    ValoareXms = x + y
    PozitieXms = (Xms.distance / display.valuecount) *
        Math.round(display.valuecount * ValoareXms / Xms.distance)
    v = Math.round((PozitieXms * Xms.scale + Xms.from) *
        Math.pow(10, display.decimals)) / Math.pow(10, display.decimals)
    display.value = v
    return false
}
return
}
function Ms(Eventiment)
{
    if (!Eventiment)
        Eventiment = window.event;
    if (Eventiment.target)
        Xms = Eventiment.target
    else
        Xms = Eventiment.srcElement;
    dist = parseInt(Xms.getAttribute('distance'))
    if (dist)
        Xms.distance = dist;
    else
        Xms.distance = LungineB
    ori = Xms.getAttribute('orientation')
    if (ori == 'vertical')
        orientation = ori;
    else
        orientation = Orientare;
    displayId = Xms.getAttribute('display')
    display = document.getElementById(displayId)
    display.sliderId = Xms.id
    dec = parseInt(display.getAttribute('decimals'))
    if (dec)
        display.decimals = dec;
    else
        display.decimals = 0;
    val = parseInt(display.getAttribute('valuecount'))
    display.valuecount = val ? val : Xms.distance + 1
    from = parseFloat(display.getAttribute('from'))
    from = from ? from : 0
    to = parseFloat(display.getAttribute('to'))
    to = to ? to : Xms.distance
    Xms.scale = (to - from) / Xms.distance
    if (orientation == 'vertical')
    {
        Xms.from = to
        Xms.xMax = 0
        Xms.yMax = Xms.distance
        Xms.scale = -Xms.scale
    }
    Xms.startOffsetX = Stanga(Xms.id) - Eventiment.screenX
    Xms.startOffsetY = Sus(Xms.id) - Eventiment.screenY
    mouseover = true
}

```

```

document.onmousemove = Miscare
document.onmouseup = sliderMouseUp
return false
}
function sliderMouseUp()
{
  if (mouseover)
  {
    v = (display.value) ? display.value : 0
    Xp = (v - Xms.from)/(Xms.scale)
    if (Xms.yMax == 0)
    {
      Xp = (Xp > Xms.xMax) ? Xms.xMax : Xp
      Xp = (Xp < 0) ? 0 : Xp
      Stanga(Xms.id, Xp)
    }
    if (Xms.xMax == 0)
    {
      Xp = (Xp > Xms.yMax) ? Xms.yMax : Xp
      Xp = (Xp < 0) ? 0 : Xp
      Sus(Xms.id, Xp)
    }
  }
  mouseover = false
}
function Afisare(Eventiment)
{
  if (!Eventiment)
  Eventiment = window.event;
  if (Eventiment.target)
    display = Eventiment.target;
  else
    display = Eventiment.srcElement;
  incuiat = display.getAttribute('typelock')
  display.blur()
  return
}
window.onload = function()
{
  Xs = NumeStil(NumeBara)
  for (i = 0; i < Xs.length; i++)
  {
    Xs[i].onmousedown = Ms
  }
  Afisari = NumeStil(AfisBr)
  for (i = 0; i < Afisari.length; i++)
  {
    Afisari[i].onfocus = Afisare
  }
}
</script>
</head>
<body>
<br /><br />
  <div class="IntVertical" style="height: 50px; background-color: #fff;
    border-color: #fff;">
    <div class="Intr" style="height: 50px; "> </div>
    <div class="bara" id="linie" orientation="vertical" distance="50"
      display="Afis"

```

```

        style="top: 25px; background-color: #7a7; border-color: #ada #474
            #474 #ada;"> </div>
</div>
<div class="DrAfis" style="background-color: #fff; border-color: #fff;">
    <input    class="AfisBara"    id="Afis"    style="background:    #fff;"
        type="text"    from="-500"    to="500"    valuecount="50"    value="0"
        typelock="on"    decimals="2"/>
</div>
</div>
</div>
</body>
</html>

```

## FileUpload

Obiectul FileUpload este o proprietate a obiectului *Form*. El permite utilizatorilor să *Upload*-eze fişiere de pe computerul client.

Proprietăți: *form, name, type, value*..

Metode: *blur(), focus()*.

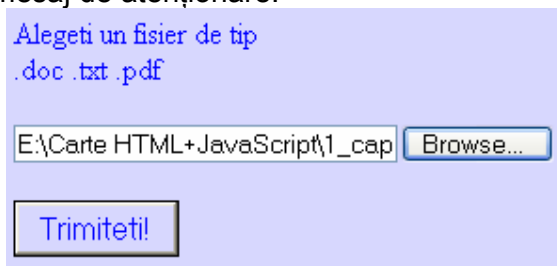
Evenimente: *onBlur, onChange, onFocus*.

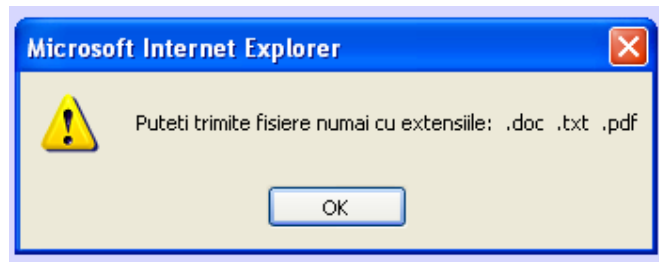
| Proprietate | Descriere   |
|-------------|---|
| form        | Obiectul Form, care include obiectul <i>fileUpload</i> .  |
| name        | Numele elementului <i>fileUpload</i> , care nu este același cu numele fişierului care este Upload-at. |
| type        | Tipul elementului, care este: <i>file</i> .   |
| value       | Reprezintă valoarea inițială a numelui fişierului care este Upload-at.                                |

| Metodă  | Descriere                            |
|---------|--------------------------------------|
| blur()  | Înlătură focalizarea de pe formular. |
| focus() | Se focalizează asupra unui formular. |

| Eveniment | Descriere  |
|-----------|--|
| onblur    | Acțiune ce va fi declanșată când se pierde focalizarea de pe elementele unui formular. |
| onChange  | Acțiune ce va fi declanșată la schimbarea unui element dintr-un formular.              |
| onfocus   | Acțiune ce va fi declanșată de focalizarea asupra unui element din formular.           |

**Exemplu** În acest exemplu, am utilizat obiectul *FileUpload*, pentru a face upload la fişiere cu anumite extensii. Dacă încercăm să trimitem fişiere cu alte extensii, va apare o fereastră cu un mesaj de atenționare.





```
<html>
<head>
  <title>Obiectul fileUpload</title>
<style type="text/css">
.scris
{ font-size: 14px;
  color: #0000FF;
  background-color: #D7D7FF;
}
</style>
<script type="text/javascript" language="javascript">
ExtensiiFisiere = new Array(".doc", ".txt", ".pdf");
function f(form, file)
{
  Accept = false;
  if (!file)
    return;
  while (file.indexOf("\\") != -1)
    file = file.slice(file.indexOf("\\") + 1);
  Extensie = file.slice(file.indexOf(".")).toLowerCase();
  for (var i = 0; i < ExtensiiFisiere.length; i++)
  {
    if (ExtensiiFisiere[i] == Extensie)
    {
      Accept = true;
    }
  }
}
if (Accept)
  form.submit();
else
  alert("Puteti trimite fisiere numai cu extensiile: "
+ (ExtensiiFisiere.join(" ")) );
}
</script>
</head>
<body class="scris">
  Alegeti un fisier de tip<br />
<script>
document.write(ExtensiiFisiere.join(" "));
</script>
<FORM method="post" NAME="formular" action="/cgi-bin/un script.cgi"
enctype="multipart/form-data">
  <input type="file" NAME="fisier" size="30"><br /><br />
  <input type="submit" name="Submit" value="Trimiteti!"
onClick="f(this.form, this.form.fisier.value)" class="scris">
</FORM>
</body>
</html>
```

## Form

Obiectul Form permite utilizatorilor să introducă elemente cum ar fi listele de selecție, butoanele radio, checkboxes. Cu JavaScript se pot trimite date către un server.

Proprietăți: *action, elements, encoding, length, method, name, target*.

Metode: *handleEvent(), reset(), submit()*.

Evenimente: *onReset, onSubmit*.

| Proprietate | Descriere   |
|-------------|---|
| action      | Specifică adresa URL către care vor fi trimise datele introduse în formular.  |
| elements    | Specifică elementele unui formular în ordinea în care sunt în sursă, similar unui masiv. Ne putem referi la elementele dintr-un formular utilizând elementele unui masiv. |
| encoding    | Returnează un șir de caractere care specifică un cod MIME dintr-un formular.  |
| length      | Returnează numărul de elemente dintr-un formular.   |
| method      | Specifică modalitatea în care datele introduse în formular vor fi trimise către server. Cele două posibilități sunt: <i>get</i> sau <i>post</i> .                         |
| name        | Returnează numele unui formular.  |
| target      | Returnează fereastra <i>target</i> către care este trimis un răspuns de către elementul <i>submit</i> al unui formular.   |

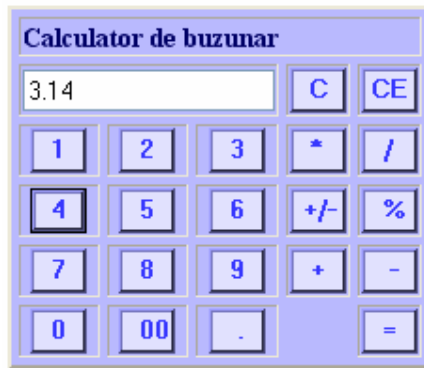
| Metodă        | Descriere   |
|---------------|---|
| handleEvent() | Specifică handlerul de eveniment pentru un anumit eveniment specificat.   |
| reset()       | Resetează valorile inițiale dintr-un formular. Acest buton este utilizat cu butonul <i>Reset</i> dintr-un formular. |
| submit()      | Trimite valorile dintr-un formular. Acest buton este utilizat cu butonul <i>Submit</i> dintr-un formular.           |

| Eveniment | Descriere  |
|-----------|--|
| onReset   | Este utilizat pentru a executa un anumit cod JavaScript după ce s-a dat click pe butonul <i>Reset</i> din formular.  |
| onSubmit  | Este utilizat pentru a executa un anumit cod JavaScript după ce s-a dat click pe butonul <i>Submit</i> din formular. |

**Exemplu** În acest exemplu, s-a utilizat obiectul Form, pentru ca utilizatorul să poată realiza anumite calcule matematice simple, dând click pe diverse butoane. Am utilizat funcții diferite ce realizează diverse operații: funcția *Operatia* pentru înmulțire, împărțire, adunare, scădere, funcția *PozNeg* pentru a realiza calcule cu semn, funcția *Procent* pentru a calcula procentul dintr-un număr introdus de către utilizator, funcția *Cifra* pentru a introduce cifrele necesare calculelor.

```
<html>
<head>
  <title>Exemplu_Calculator</title>
<style type="text/css">
.calc
{
  font-size: 14px;
  font-weight: bolder;
  color: #000066;
  background-color: #B9B9FF;
}

```



```

.buton
{
  color: #3737FF;
  background-color: #E1E1FF;
  width: 30px;
  border: thin outset #9F9FFF;
  font-weight: bolder;
  text-align: center;
}
</style>
</head>
<body>
<center>
<FORM name="formular" >
<TABLE border="2" width="50" height="60" cellpadding="1" cellspacing="4"
  class="calc">
<tr>
  <td colspan="6">Calculato de buzunar</td>
</tr>
<tr>
  <td align="center" colspan="3">
    <input name="Citire" type="Text" size=18 value="0" width="100%">
  </td>
  <td>
    <input name="Curata_" type="Button" value="C" onClick="Curata()"
      class="buton">
  </td>
  <td>
    <input name="CurataIntr_" type="Button" value="CE"
      onClick="CurataIntr()" class="buton">
  </td>
</tr>
<tr >
  <td align="center" colspan="1" >
    <input name="unu" type="Button" value=" 1 " onClick="Cifra(1)"
      class="buton">
  </td>
  <td align="center" colspan="1" >
    <input name="doi" type="Button" value=" 2 " onClick="Cifra(2)"
      class="buton">
  </td>
  <td align="center" colspan="1">
    <input name="trei" type="Button" value=" 3 " onClick="Cifra(3)"
      class="buton">
  </td>
  <td align="center" colspan="1">
    <input name="Inmultit" type="Button" value="*" onClick="Operatia('*)"

```



```

        class="buton">
    </td>
    <td align="center" colspan="1">
        <input name="Impartit" type="Button" value="/" onClick="Operatia('/') "
            class="buton">
    </td>
</tr>
<tr>
    <td align="center" colspan="1">
        <input name="patru" type="Button" value=" 4 " onClick="Cifra(4) "
            class="buton"></td>
    <td align="center" colspan="1">
        <input name="cinci" type="Button" value=" 5 " onClick="Cifra(5) "
            class="buton"></td>
    <td align="center" colspan="1">
        <input name="sase" type="Button" value=" 6 " onClick="Cifra(6) "
            class="buton"></td>
    <td align="center" colspan="1">
        <input name="Negativ" type="Button" value=" +/- " onClick="PozNeg() "
            class="buton"></td>
    <td align="middle">
        <input name="Procent" type="Button" value=" % " onClick="Procent() "
            class="buton"></td>
</tr>
<tr>
    <td align="center" colspan="1">
        <input name="sapte" type="Button" value=" 7 " onClick="Cifra(7) "
            class="buton" ></td>
    <td align="center" colspan="1">
        <input name="opt" type="Button" value=" 8 " onClick="Cifra(8) "
            class="buton"></td>
    <td align="center" colspan="1">
        <input name="noua" type="Button" value=" 9 " onClick="Cifra(9) "
            class="buton"></td>
    <td align="center" colspan="1">
        <input name="Plus" type="Button" value=" + "
            onClick="Operatia('+)' " class="buton"> </td>
    <td align="center" colspan="1">
        <input name="Minus" type="Button" value=" - " onClick="Operatia('-
            ')" class="buton"></td>
</tr>
<tr>
    <td align="center" colspan="1">
        <input name="zero" type="Button" value=" 0 " onClick="Cifra(0) "
            class="buton"></td>
    <td align="center" colspan="1">
        <input name="zero_zero" type="Button" value=" 00 "
            onClick="Cifra00(0) " class="buton"> </td>
    <td align="center" colspan="1">
        <input name="PunctZecimal" type="Button" value=" ."
            onClick="Zecimal() " class="buton"> </td>
    <td colspan="1"></td>
    <td>
        <input name="Egal" type="Button" value=" = " onClick="Operatia('=)' "
            class="buton"> </td>
</tr>
</TABLE>
</TABLE>
</FORM>

```

```

</center>
<script type="text/javascript" language="javascript">
var Valformular = document.formular;
var VarOp = 0;
var Nr = false;
var Liber = "";
function Cifra (Numar)
{
  if (Nr)
  {
    Valformular.Citire.value = Numar;
    Nr = false;
  }
  else
  {
    if (Valformular.Citire.value == "0")
      Valformular.Citire.value = Numar;
    else
      Valformular.Citire.value += Numar;
  }
}
function Cifra00 (Numar)
{
  if (Nr)
  {
    Valformular.Citire.value = Numar;
    Nr = false;
  }
  else
  {
    if (Valformular.Citire.value == "00")
      Valformular.Citire.value = Numar;
    else
      Valformular.Citire.value += Numar;
      Valformular.Citire.value += Numar;
  }
}
function Operatia (XOpr)
{
  var Citire = Valformular.Citire.value;
  if (Nr && Liber != "=");
  else
  {
    Nr = true;
    if ( '+' == Liber )
      VarOp += parseFloat(Citire);
    else
    if ( '-' == Liber )
      VarOp -= parseFloat(Citire);
    else
    if ( '/' == Liber )
      VarOp /= parseFloat(Citire);
    else
    if ( '*' == Liber )
      VarOp *= parseFloat(Citire);
    else
      VarOp = parseFloat(Citire);
  }
  Valformular.Citire.value = VarOp;
  Liber = XOpr;
}

```

```

    }
}
function Zecimal ()
{
    var VarZec = Valformular.Citire.value;
    if (Nr)
    {
        VarZec = "0.";
        Nr = false;
    }
    else
    {
        if (VarZec.indexOf(".") == -1)
            VarZec += ".";
    }
    Valformular.Citire.value = VarZec;
}
function CurataIntr ()
{
    Valformular.Citire.value = "0";
    Nr = true;
}
function Curata ()
{
    VarOp = 0;
    Liber = "";
    CurataIntr();
}
function PozNeg ()
{
    Valformular.Citire.value = parseFloat(Valformular.Citire.value) * -1;
}
function Procent ()
{
    Valformular.Citire.value = (parseFloat(Valformular.Citire.value) / 100) *
parseFloat(VarOp);
}
</SCRIPT>
</font>
</body>
</html>

```

## Frame

Obiectul Frame permite utilizarea mai multor ferestre în același timp. Fiecare dintre aceste ferestre poate avea propriul URL.

**Exemplu** În acest exemplu, am utilizat obiectul Frame, pentru ca utilizatorul să poată vedea în partea din dreapta a paginii, adică în cadrul din dreapta, o poezie pe care o alege.

Pagina principală :

```

<html>
<head>
    <title>Obiectul Frame</title>
</head>
<frameset cols="20%,80%" framespacing="1" frameborder= "yes" border="1">
    <frame src="stanga.htm" name="stanga" scrolling="no" noresize >
    <frame src="dreapta.htm" name="dreapta" scrolling="yes">

```

```

</frameset>
<body>
</body>
</html>

```

| Lucian Avramescu   | Despre ambitia cuiva de a face alt om din mine   |
|--|--|
| <a href="#">Despre ambitia cuiva de a face alt om din mine</a> | da-ma la remaiat,<br>du-ma la intors, tese-ma din nou,<br>zugraveste-ma, pune-mi alt guler,<br>alta manseta, alt zbenghi,<br>da-mi alt numar la pantofi,<br>toaca-ma marunt<br>si umple-ma cu condimentele moralei tale,<br>impunge-ma cu o mie de sfaturi,<br>imprastie-ma in patru vanturi<br>si vei observa ca acolo unde cad<br>se umple locul de mine |
| <a href="#">Singurul lucru care conteaza</a>                   |  |
| <a href="#">Niciodata, iubito</a>                              |  |
| <a href="#">Ia aminte</a>                                      |  |

#### Pagina din stânga:

```

<html>
<head>
  <title>Lucian Avramescu</title>
  <link href="StilPoezie.css" rel="stylesheet" type="text/css">
</head>
<body>
<span class="titlul">Lucian Avramescu
</span><br /><br />
<a href="poezia1.html" target="dreapta" >Despre ambitia cuiva de a face
alt om din mine </a><br /><br />
<a href="poezia2.html" target="dreapta">Singurul lucru care conteaza
</a><br /><br />
<a href="poezia3.html" target="dreapta">Niciodata, iubito</a> <br /><br />
<a href="poezia4.html" target="dreapta">Ia aminte </a>
</body>
</html>

```

#### Pagina din dreapta:

```

<html>
<head>
<title>Obiectul Frame</title>
<link href="StilPoezie.css" rel="stylesheet" type="text/css">
</head>
<body>
<span class="titlul">Despre ambitia cuiva de a face alt om din mine
</span> <br /><br />
da-ma la remaiat,<br />
  du-ma la intors, tese-ma din nou,<br />
  zugraveste-ma, pune-mi alt guler,<br />
  alta manseta, alt zbenghi,<br />
  da-mi alt numar la pantofi,<br />
  toaca-ma marunt<br />
  si umple-ma cu condimentele moralei tale,<br />
  impunge-ma cu o mie de sfaturi,<br />
  imprastie-ma in patru vanturi<br />
  si vei observa ca acolo unde cad<br />
  se umple locul de mine<br />
</body>
</html>

```

## Stilul extern : StilPoezie.css

```
body
{
  color: #FF0000;
  background-color: #FFFFAE;
  text-decoration: none;
}
.titlul
{
  font-size: 16px;
  font-weight: bolder;
  color: #FF0000;
  text-align: center;
  font-style: normal;
  text-decoration: none;
}
```

## Function

Obiectul *Function* specifică un șir de cod JavaScript ce poate fi compilat ca o funcție. Proprietăți: *arguments*, *arguments.callee*, *arguments.caller*, *arguments.length*, *arity*, *constructor*, *length*, *prototype*.

Metode: *apply()*, *call()*, *toSource()*, *toString()*, *valueOf()*.

| Proprietate             | Descriere  |
|-------------------------|--|
| <i>arguments</i>        | Reprezintă un masiv ce corespunde argumentelor unei funcții.                 |
| <i>arguments.callee</i> | Returnează corpul funcției care se execută.                                  |
| <i>arguments.callee</i> | Specifică numele funcției care este invocată la executarea funcției curente. |
| <i>arguments.length</i> | Specifică numărul de argumente pe care le are funcția.                       |
| <i>arity</i>            | Specifică numărul de argumente așteptate de către funcție.                   |
| <i>constructor</i>      | Specifică funcția care creează <i>prototype</i> unui obiect.                 |
| <i>length</i>           | Reprezintă un întreg ce specifică numărul de elemente ale unui masiv.        |
| <i>prototype</i>        | Permite adăugarea de proprietăți tuturor obiectelor.                         |

| Metodă            | Descriere   |
|-------------------|---|
| <i>apply()</i>    | Permite aplicarea metodei unui obiect diferit, în alt context.              |
| <i>call()</i>     | Permite executarea metodei de către un alt obiect, într-un context diferit. |
| <i>toSource()</i> | Returnează un șir de caractere ce reprezintă codul sursă al funcției.       |
| <i>toString()</i> | Returnează un șir de caractere ce reprezintă codul sursă al funcției.       |
| <i>valueOf()</i>  | Returnează un șir de caractere ce reprezintă codul sursă al funcției.       |

**Exemplu** În acest exemplu, am utilizat obiectul *Function* pentru a putea adăuga pagina preferată la favorite.

```
<html>
<head>
  <title>Obiectul Function</title>
<script
  type="text/javascript"
  language="JavaScript">
function f()
{
  if (document.all)
    window.external.AddFavorite("http://pagina.html/"
    ,"Exemplu")
}
```

Adaugati pagina la "Pagini Favorite"

Apasati!

```

}
</script>
</head>
<body>
<form name="formular" onClick="f()" >
Aduagati pagina la "Pagini Favorite"<br /><br />
<input name="" type="button" value="Apasati!">
</form>
</body>
</html>

```



## Hidden

Obiectul Hidden este o proprietate a obiectului *Form* și specifică un șir de cod JavaScript ce poate fi compilat ca o funcție. Ceea ce se scrie în formularul ce conține un obiect *Hidden* este invizibil pentru utilizatori, dar informațiile pe care le conțin proprietățile *name* și *value* sunt transmise odată cu formularul din care face parte. Proprietăți: *form*, *name*, *type*, *value*.

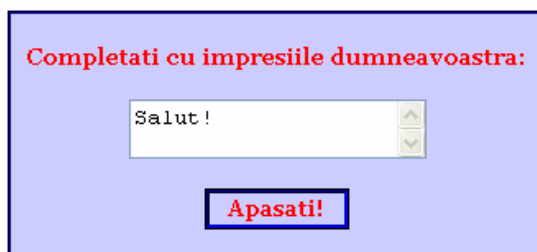
| Proprietate | Descriere  |
|-------------|--|
| form        | Reprezintă formularul care conține câmpul Hidden.                          |
| name        | Reprezintă numele câmpului Hidden.   |
| type        | Reprezintă tipul elementului Hidden.                                       |
| value       | Reprezintă un șir de caractere care specifică o valoare a câmpului Hidden. |

**Exemplu** În acest exemplu, am utilizat obiectul *Hidden* pentru a reține un anumit șir de caractere ce va fi transmis unui script JavaScript. Scriptul utilizează șirul primit pentru a face o afișare. Exemplul s-ar putea schimba, în așa fel încât informațiile deținute de elementul *value* al obiectului Hidden, să fie transmise către server fără ca utilizatorul să fie informat în legătură cu acest lucru.

```

<html>
<head>
<title>Obiectul Hidden</title>
<style type="text/css">
.a
{font-family: Georgia, Times, serif;
font-size: 14px;
font-weight: bold;
color: #FF0000;
background-color: #CCCCFF;
text-align: center;
border: thin inset #0000FF; }
</style>
<script type="text/javascript"
language="JavaScript">
function f(x)
{
alert('Ati completat cu: '+x);
}
</script>
</head>
<body>
<form name="formular" class="a">
<input type="hidden" name="x"

```



```

        value="None"><br />
Completati      cu      impresiile
dumneavoastra:<br /><br />
<textarea      name="text"      value=""
onMouseOut="document.formular.x.val
lue=this.value">
</textarea><br /><br />
<input      type="submit"      name="Submit"
value="Apasati!"
onClick="f(document.formular.x.val
ue)" class="a"><br />
<br />
</form>
</body>
</html>

```

## History

Obiectul *History* este o proprietate a obiectului *Window*. Cu ajutorul lui, putem ajunge la URL-urile vizitate, într-un mod similar cu butonul *Back* al browser-ului.

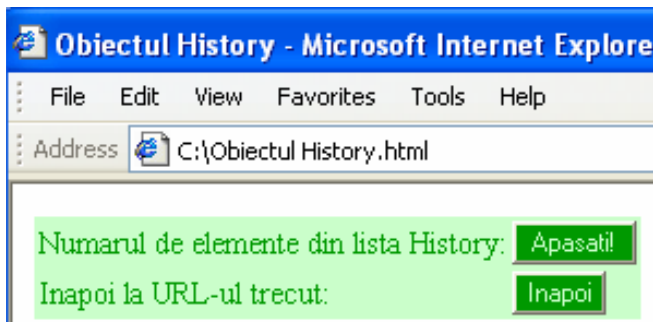
Proprietăți: *current*, *length*, *next*, *previous*.

Metode: *back()*, *forward()*, *go()*.

| Proprietate     | Descriere   |
|-----------------|---|
| <i>current</i>  | Reprezintă URL-ul documentului curent.                                |
| <i>length</i>   | Reprezintă numărul de intrări în obiectul <i>History</i> .            |
| <i>next</i>     | Reprezintă URL-ul documentului următor din obiectul <i>History</i> .  |
| <i>previous</i> | Reprezintă URL-ul documentului anterior din obiectul <i>History</i> . |

| Metodă           | Descriere   |
|------------------|---|
| <i>back()</i>    | Ne trimite la intrarea (URL) anterioară din lista <i>History</i> , similar cu butonul <i>Back</i> din browser.  |
| <i>forward()</i> | Ne trimite la intrarea (URL) următoare din lista <i>History</i> , similar cu butonul <i>Forward</i> din browser.  |
| <i>go()</i>      | Ne trimite la intrarea (URL) pe care o dorim din lista <i>History</i> , dacă menționăm a câta pagină a fost vizitată înainte sau în urmă. De exemplu: <i>go(-2)</i> sau <i>go(+3)</i> . |

**Exemplu** În acest exemplu, am utilizat obiectul *History* pentru a afișa numărul de elemente din lista *History*.





```
<html>
<head>
  <title>
    Obiectul History
  </title>
<style type="text/css">
.s
{ font-size: 14px;
  color: #009900;
  background-color: #CAFFCA;
}
.b
{ font-size: 12px;
  color: #CAFFCA;
  background-color:#009900;
}
</style>
<script type="text/javascript" language="JavaScript">
function f(x,y)
{
  alert(x+y);
}
</script>
</head>
<body>
<form name="formular">
<table border="0" cellspacing="1" cellpadding="1" class="s">
<tr>
  <td>Numarul de elemente din lista History:
  </td>
  <td><input type="hidden" name="h1" value="Numarul de elemente din lista
    History: ">
    <input          name="button"          type="button"          onClick=
    "f(document.formular.h1.value,history.length) "          value="Apasati!"
    class="b">
  </td>
</tr>
</tr>
<tr>
  <td>Inapoi la URL-ul trecut:
  </td>
  <td><input name="button2" type ="button"      onClick="history.back() "
    value="Inapoi" class="b">
  </td>
</tr>
</tr>
</table>
</form>
</body>
</html>
```



## Image

Obiectul Image este o imagine dintr-un formular.

Proprietăți: *border, complete, height, hspace, lowsrc, name, prototype, src, vspace, width*.

Evenimente: *onAbort, onError, onKeyDown, onKeyPress, onKeyUp, onLoad*.

| Proprietate | Descriere  |
|-------------|--|
| border      | Reprezintă un număr întreg care specifică dimensiunea bordurii imaginii, în pixeli.  |
| complete    | Reprezintă o valoare de tip boolean, care specifică momentul în care imaginea s-a încărcat complet.                                |
| height      | Reprezintă înălțimea imaginii.   |
| hspace      | Reprezintă spațiul pe orizontală de la stânga spre dreapta imaginii.   |
| lowsrc      | Reprezintă un șir de caractere ce specifică versiunea URL de slabă rezoluție a unei imagini, pentru a fi afișată într-un document. |
| name        | Reprezintă numele imaginii.  |
| prototype   | Este utilizat pentru a adăuga proprietăți specifice unei imagini.  |
| src         | Reprezintă URL-ul unei imagii, pentru a fi afișată.  |
| vspace      | Reprezintă spațiul pe verticală în jurul imaginii.   |
| width       | Reprezintă lățimea imaginii în pixeli.   |

| Eveniment  | Descriere   |
|------------|---|
| onAbort    | Acțiune ce va fi declanșată în urma renunțării la download-area unei imagini.             |
| onError    | Acțiune ce va fi declanșată în urma eșecului încărcării unei imagini sau a unui document. |
| 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 este eliberată o tastă.                                  |
| onLoad     | Acțiune ce va fi declanșată în urma unui download complet al unei imagini.                |

**Exemplu** În acest exemplu, am utilizat obiectul *Image* pentru a afișa o imagine. Imaginea se deplasează pe suprafața disponibilă a browser-ului. Putem opri imaginea, dacă dăm click pe ea.

```
<html>
<head>
  <title>Obiectul Image</title>
<style type="text/css">
body
{ font-size: 18px;
  font-weight: bolder;
  color: #FF00FF;
  background-color: #FFCAFF;
  text-align: center;
}
</style>
</head>
<body>
Zambila<br /><br />
<div id="img" style="position:absolute;">
```



```


</div>
<script type="text/javascript" language=
"JavaScript">
var height = 0;
var ina = 0;
var lat = 0;
var y = 0;
var x = 0;
var pauza = true;
var interval;
var Xo = 880;
var Yo = 0;
var pas = 1;
var timp = 10;
function PozitiePoza()
{
latime = document.body.clientWidth;
inaltime = document.body.clientHeight;
ina = img.offsetHeight;
lat = img.offsetWidth;
img.style.left = Xo + document.body.scrollLeft;
img.style.top = Yo + document.body.scrollTop;
if (y)
{ Yo+= pas; }
else
{ Yo -= pas; }
if (Yo < 0)
{ y = 1;
Yo = 0; }
if (Yo >= (inaltime - ina))
{ y = 0;
Yo = (inaltime - ina); }
if (x)
{ Xo += pas; }
else
{ Xo -= pas; }
if (Xo < 0)
{ x = 1;
Xo = 0; }
if (Xo >= (latime - lat))
{ x = 0;
Xo = (latime - lat); }
}
function start()
{
img.visibility = "visible";
interval = setInterval('PozitiePoza()',timp);
}
function PauzaC()
{
if(pauza)
{
clearInterval(interval);
pauza = false;
}
}
else

```

```

    {
        interval = setInterval('PozitiePoza()', timp);
        pauza = true;
    }
}
start();
</script>
</body>
</html>

```

## Java

Java este un obiect de nivel înalt și este utilizat pentru a accesa orice clasă Java din *package java*.

## JavaArray

JavaArray este un masiv ce poate fi accesat din interiorul unui script JavaScript

Proprietăți: *length*;

Metode: *toString*;

| Proprietate | Descriere  |
|-------------|--|
| length      | Reprezintă numărul de elemente ale masivului Java, reprezentat de JavaArray. |

| Metodă     | Descriere  |
|------------|--|
| toString() | Returnează un șir de caractere ce reprezintă elementele JavaArray. |

## JavaClass

JavaClass reprezintă o trimitere către o clasă Java.

## JavaObject

JavaObject reprezintă o instanță a unei clase Java, creată cu ajutorul unui script JavaScript.

## JavaPackage

JavaPackage reprezintă o colecție de clase sau de Java packages.

## Layer

Obiectul Layer oferă posibilitatea de a poziționa diferite blocuri cu gradele de transparență dorite, într-o pagină HTML.

Proprietăți: *above, background, bgColor, below, clip.bottom, clip.height, clip.left, clip.right, clip.top, clip.width, document, left, name, pageX, pageY, parentLayer, siblingAbove, siblingBelow, src, top, visibility, window, x, y, zIndex*.

Metode: *captureEvents(), handleEvent(), load(), moveAbove(), moveBelow(), moveBy(), moveTo(), moveToAbsolute(), releaseEvents(), resizeBy(), resizeTo(), routeEvent()*.

Evenimente: *onBlur, onFocus, onLoad, onMouseOut, onMouseOver*.

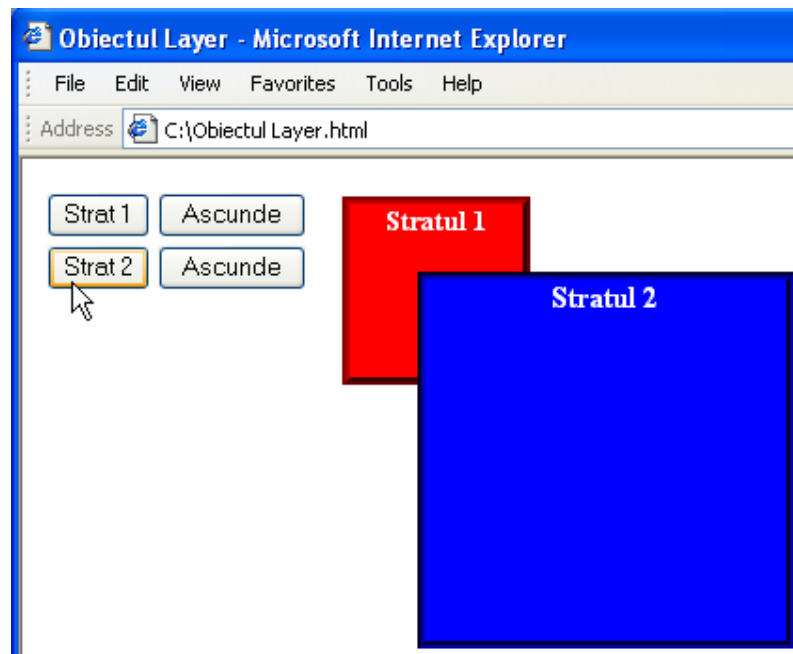
| Proprietate | Descriere  |
|-------------|--|
| above       | Obiectul Layer permite crearea mai multor straturi, poziționate unul deasupra celuilalt. Când utilizăm mai multe layer-e, trebuie să specificăm care dintre ele se află deasupra (above) și care se află în spate (below). |
| background  | Reprezintă fundalul unui layer.  |

|              |   |
|--------------|---|
| bgColor      | Reprezintă culoarea de fundal a unui layer.   |
| below        | Specifică faptul că layer-ul respectiv se află în spatele altui layer.  |
| clip.bottom  | Reprezintă partea de jos a unui dreptunghi vizibil.   |
| clip.height  | Reprezintă înălțimea dreptunghiului vizibil.  |
| clip.left    | Reprezintă partea din stânga a unui dreptunghi vizibil.   |
| clip.right   | Reprezintă partea din dreapta a unui dreptunghi vizibil.  |
| clip.top     | Reprezintă partea de sus a unui dreptunghi vizibil.   |
| clip.width   | Reprezintă lățimea dreptunghiului vizibil.  |
| document     | Reprezintă documentul asociat unui layer.   |
| left         | Reprezintă poziția pe orizontală, dinspre stânga, măsurată în pixeli.   |
| name         | Reprezintă un șir de caractere ce se asociază numelui unui layer.   |
| pageX        | Reprezintă poziția pe orizontală a unui layer, relativ la pagină, măsurată în pixeli.   |
| pageY        | Reprezintă poziția pe verticală a unui layer, relativ la pagină, măsurată în pixeli.  |
| parentLayer  | Reprezintă un layer care se consideră layer de referință pentru alte layer-e.   |
| siblingAbove | Reprezintă o referință pentru obiectul <i>Layer</i> de deasupra unui layer specificat în z-order, printre mai multe layer-e ce împart același parent layer. |
| siblingBelow | Reprezintă o referință pentru obiectul <i>Layer</i> de sub un layer specificat în z-order, printre mai multe layer-e ce împart același parent layer.        |
| src          | Un șir de caractere ce specifică URL-ul și conținutului layer-ului.   |
| top          | Poziția pe verticală, din partea de sus a unui layer, în pixeli, relativ la origina parent layer.   |
| visibility   | Reprezintă starea de vizibilitate sau invizibilitate a unui layer.  |
| window       | Obiectele <i>Window</i> sau <i>Frame</i> care conțin un layer în relație cu alte layer.   |
| x            | Poziția pe orizontală dinspre stânga a unui layer, în pixeli, relativ la parent layer.  |
| y            | Poziția pe verticală dinspre stânga a unui layer, în pixeli, relativ la parent layer.   |
| zIndex       | Returnează z-order relativ la un layer specificat în relația cu orice <i>sibling layers</i> .   |

| Metodă          | Descriere   |
|-----------------|---|
| captureEvents() | Setează fereastra sau documentul pentru a captura toate evenimentele unui tip specificat.     |
| handleEvent()   | Invocă un handler pentru un eveniment specificat.   |
| load()          | Schimbă sursa unui layer, către conținutul unui fișier specificat.                            |
| moveAbove()     | Blochează un layer deasupra unui layer specificat, fără schimbarea poziției.                  |
| moveBelow()     | Blochează un layer sub un layer specificat, fără schimbarea poziției.                         |
| moveBy()        | Schimbă poziția unui layer, prin specificarea numărului de pixeli pe orizontală și verticală. |
| moveTo()        | Schimbă coordonatele colțului din stânga-sus al unui layer.                                   |

|                  |  |
|------------------|--|
| moveToAbsolute() | Schimbă coordonatele poziției absolute a unui layer specificat (colțul din stânga-sus).    |
| releaseEvents()  | Setează fereastra sau documentul pentru a elibera toate evenimentele unui tip specificat.  |
| resizeBy()       | Redimensionează un layer prin specificarea valorilor în pixeli, pentru înălțime și lățime. |
| resizeTo()       | Redimensionează un layer să aibă o înălțime și o lățime specificată.                       |
| routeEvent()     | Trimiterea evenimentului capturat în fața evenimentului ierarhic.                          |

| Eveniment   | Descriere  |
|-------------|--|
| onBlur      | Acțiune ce va fi declanșată când se pierde focalizarea de pe un element.   |
| onFocus     | Acțiune ce va fi declanșată de focalizarea asupra unui element.  |
| onLoad      | Acțiune ce va fi declanșată la încărcarea unui document.   |
| 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.  |



**Exemplu** În acest exemplu, am utilizat obiectul *Layer* pentru a afișa și a ascunde două pătrate de dimensiuni diferite.

```
<html>
<head>
<title>Obiectul Layer</title>
<style type="text/css">
#strat1
{
background-color: red;
left: 170;
position: absolute;
top: 20;
visibility: hidden;
font-size: 14px;

```

```

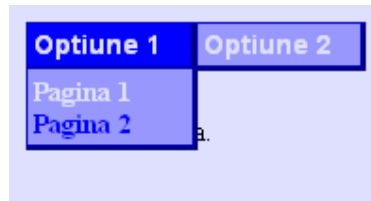
    font-weight: bold;
    color: #FFFFFF;
    border: medium outset #B90000;
    text-align: center;
    height: 100px;
    width: 100px;
}
#strat2
{ background-color: blue;
  left: 210;
  position: absolute;
  top: 60;
  visibility: hidden;
  font-weight: bolder;
  color: #FFFFFF;
  text-align: center;
  border: medium groove #0000A6;
  height: 200px;
  width: 200px;
}
</style>
<script type="text/javascript" language="JavaScript1.2">
    var layer = new String();
    var style = new String();
function VerifBrowser()
    {
        if(navigator.userAgent.indexOf("MSIE") != -1)
            {
                layer = ".all";
                style = ".style";
            }
        else
            if(navigator.userAgent.indexOf("Nav") != -1)
                {
                    layer = ".layers";
                    style = "";
                }
    }
function f(Xstrat, Xstare)
    {
        eval("document" + layer + "[" + Xstrat + "]" + style + ".visibility =
            '" + Xstare + "'");
    }
function g(Ystrat, Ystare)
    {
        eval("document"+ layer + "[" + Ystrat + "]" + style + ".visibility =
            '" + Ystare + "'");
    }
</script>
</head>
<body onload="VerifBrowser()">
    <form name="formular">
    <table>
    <tr>
    <td><div name="strat1" id="strat1"> Stratul 1 </div>
        <input type="button" name="buton1" value="Strat 1"
            onClick="f('strat1','visible')"><br />
    </td>
    <td><input type="button" name="buton2" value="Ascunde"

```

```

        onClick="f('strat1','hidden')"><br />
    </td>
</tr>
<tr>
    <td><div name="strat2" id="strat2"> Stratul 2 </div>
        <input type="button" name="buton3" value="Strat 2"
            onClick="g('strat2','visible')"><br />
    </td>
    <td>
        <input type="button" name="buton4" value="Ascunde"
            onClick="g('strat2','hidden')"><br />
    </td>
</tr>
</table>
</form>
</body>
</html>

```



**Exemplu** În acest exemplu, am utilizat obiectul *Layer* pentru a afișa și a ascunde elementele unui meniu orizontal, cu submeniuri.

```

<html>
<head>
    <title>Exemplu de meniu cu Layer</title>
<style type="text/css">
.MeniuVertical
{ border-top: 1px ridge #0000FF;
  border-bottom: 3px ridge #0000FF;
  border-left: 1px ridge #0000FF;
  border-right: 3px ridge #0000FF;
  background-color: #9797FF;
  layer-background-color: #999999;
  z-index: 60;
  visibility: hidden;
  width: 91px;
  position: absolute;
  color: #0000FF;
  padding: 3px;
}
.MeniuCuOptiuni
{ border-top: 1px ridge #0000FF;
  border-bottom: 3px ridge #0000FF;
  border-left: 1px ridge #0000FF;
  border-right: 3px ridge #0000FF;
  background-color: #9797FF;
  width: 90px;
  padding: 3px;
}
.MeniuOrizantal
{
  color: #DFDFFF;
  font-family: Geneva, Arial, Helvetica, sans-serif;
  text-decoration: none;
  outline: none;
}

```

```

    font-size: 14px;
    font-weight: bold;
}
.MeniuVerticalLink
{ color: #0000FF;
  text-decoration: none;
  outline: none;
  font-weight: bold;
}
A.MeniuVerticalLink:hover
{
  color: #DFDFFF;
  text-decoration: none;
}
#MeniuCuOptiuni1
{
  font-size: 14px;
  left: 60px;
  position: absolute;
  top: 20px;
}
#MeniuCuOptiuni2
{
  font-size: 14px;
  left: 150px;
  position: absolute;
  top: 20px;
}
#MeniuVertical1
{
  font-size: 14px;
  left: 60px;
  position: absolute;
  top: 44px;
}
#MeniuVertical2
{
  font-size: 14px;
  left: 150px;
  position: absolute;
  top: 44px;
  height: 61px;
}
#ContinutDiv
{
  font-size: 14px;
  left: 60px;
  position: absolute;
  top: 70px
}
</style>
<script type="text/javascript" language="JavaScript">
function Vizibilitate(IdMeniu, Viz)
{
  var StilMeniu = document.all(IdMeniu).style;
  if(StilMeniu)
  {
    StilMeniu.visibility = Viz;
    return true;
  }
}

```



```

    }
    else
    {
        return false;
    }
}
function FMenuVertical(nr, eveniment)
{
    AscundeMenuVertical();
    var IdMenuVertical = 'MenuVertical' + nr;
    if(Vizibilitate(IdMenuVertical, 'visible'))
    {
        var MenuCuOptiuni = document.all('MenuCuOptiuni' + nr).style;
        MenuCuOptiuni.backgroundColor = '#0000FF';
        eveniment.cancelBubble = true;
        return true;
    }
    else
    {
        return false;
    }
}
var NrMenuVertical = 2;
function AscundeMenuVertical()
{
    for(i = 1; i <= NrMenuVertical; i++)
    {
        Vizibilitate('MenuVertical' + i, 'hidden');
        var MenuCuOptiuni = document.all('MenuCuOptiuni' + i).style;
        MenuCuOptiuni.backgroundColor = '#9797FF';
    }
}
document.onclick = AscundeMenuVertical;
</script>
</head>
<body bgcolor="#DFDFFF">
<div class=MenuVertical id=MenuVertical1 onclick=event.cancelBubble
='true;''>
    <A class=MenuVerticalLink href="pagina1.html">Pagina 1</A><br />
    <A class=MenuVerticalLink href="pagina2.html">Pagina 2</A><br />
</div>
<div class=MenuVertical id=MenuVertical2 onclick="event.cancelBubble =
true;">
    <A class=MenuVerticalLink href="pagina3.html">Pagina 3</A><br />
    <A class=MenuVerticalLink href="pagina4.html">Pagina 4</A><br />
    <A class=MenuVerticalLink href="pagina5.html">Pagina 5</A>
</div>
<div class=MenuCuOptiuni id=MenuCuOptiuni1>
    <A class=MenuOrizantal onmouseover="return !FMenuVertical('1',
event);">Optiune 1</A>
</div>
<div class=MenuCuOptiuni id=MenuCuOptiuni2>
    <A class=MenuOrizantal onmouseover="return !FMenuVertical('2',
event);">Optiune 2</A>
</div>
<div id=ContinutDiv>Pagina principala. </div>
</body>
</html>

```

## Link

Obiectul Link este un șir de caractere, o imagine sau o porțiune dintr-o imagine prin intermediul căreia se poate crea o legătură către un alt document sau URL.

Proprietăți: *hash, host, hostname, href, pathname, port, protocol, search, target, text, x, y.*

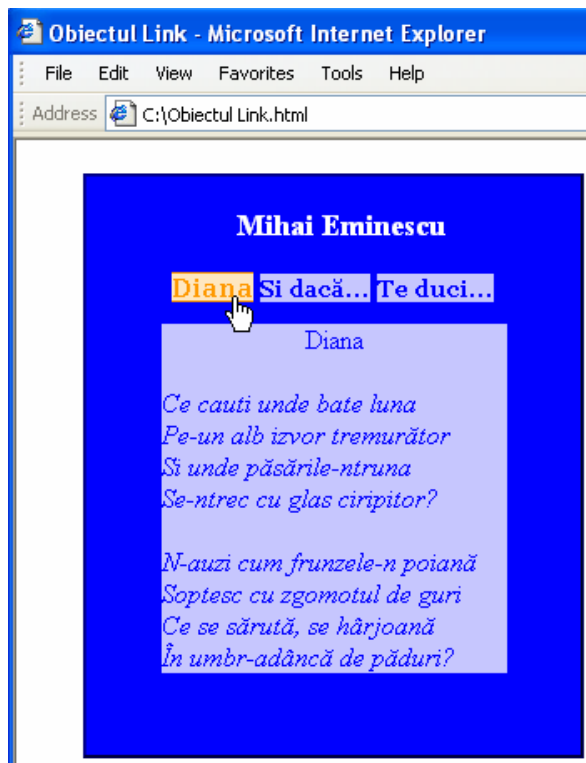
Metode: *handleEvent()*.

Evenimente: *onClick, onDbClick, onKeyDown, onKeyPress, onKeyUp, onMouseDown, onMouseOut, onMouseOver, onMouseUp.*

| Proprietate | Descriere  |
|-------------|--|
| hash        | Reprezintă un șir de caractere ce începe cu caracterul # (hash) ce specifică numele unei ancore.   |
| host        | Este un șir de caractere ce reprezintă denumirea host, numele de domeniu sau adresa IP a unui host dintr-o rețea.                            |
| hostname    | Specifică porțiunea <i>host:port</i> a unui URL.   |
| href        | Reprezintă întregul URL.   |
| pathname    | Este un șir de caractere ce reprezintă o porțiune dintr-un URL ce specifică modalitatea în care pot fi accesate diverse resurse particulare. |
| port        | Este un șir de caractere ce specifică portul pe care îl utilizează server-ul.  |
| protocol    | Este un șir de caractere ce specifică începutul unui URL și include prima coloană (:), ce reprezintă metoda de accesare a URL-ului.          |
| search      | Este un șir de caractere ce conține o întrebare.   |
| target      | Este un șir de caractere ce specifică locul unde se va deschide fereastra în urma acționării unui link.                                      |
| text        | Este un șir de caractere ce conține textul ce corespunde etichetei « A ».  |
| x           | Reprezintă poziția pe orizontală, dinspre stânga, în pixeli, relativ la partea din stânga a documentului.                                    |
| y           | Reprezintă poziția pe verticală, dinspre partea de sus a paginii, în pixeli, relativ la partea de sus a documentului.                        |

| Metodă        | Descriere                                      |
|---------------|--|
| handleEvent() | Invocă handler pentru un eveniment specificat. |

| Eveniment   | Descriere  |
|-------------|--|
| onClick     | Acțiune ce va fi declanșată când se dă click stânga pe mouse.  |
| onDbClick   | Acțiune ce va fi declanșată când se dă dublu click stânga pe mouse.  |
| 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 este eliberată o tastă.   |
| onMouseDown | Acțiune ce va fi declanșată când un buton al mouse-ului este apăsat.   |
| 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.   |



**Exemplu** În acest exemplu, am utilizat obiectul Link pentru a realiza meniul de mai sus.

```

<html>
<head>
  <title> Obiectul Link</title>
<style type="text/css">
a.one:link
{
  color: #0000FF;
  text-decoration: none;
  background: #C6C6FF;
  font-family: "Georgia", "Times New Roman", "Times";
  font-weight: bold;
  font-size: 14px;
}
a.one:visited
{
  color: #0000CE;
  text-decoration: none;
  background: #C6C6FF;
  font-family: "Courier New", "Courier";
  font-size: 14px;
  text-align: center;
}
a.one:hover
{
  color: #FF9900;
  text-decoration:underline overline;
  font-size: 16px;
  background: #FFE7C1;
  text-align: center;
}
.stil
{
  background-color: #0000FF;

```

```

    text-align: center;
    border: 2px outset #0000A6;
    visibility: visible;
    height: 350px;
    width: 300px;
    position: absolute;
    left: 40px;
    top: 20px;
}
.as
{
    color: #0000FF;
    background-color: #C6C6FF;
    visibility: hidden;
    position: absolute;
    left: 45px;
    top: 88px;
    width: 207px;
    height: 135px;
}
.titlul
{
    font-size: 18px;
    font-weight: bolder;
    color: #FFFFFF;
    text-align: center;
    visibility: visible;
    position: absolute;
    left: 46px;
    top: 19px;
    width: 213px;
    height: 20px;
}
.s
{
    text-align: left;
    font-style: italic;
}
</style>
<script type="text/javascript" language="JavaScript">
var layer = new String();
var style = new String();
function VerifBrowser()
{
    if(navigator.userAgent.indexOf("MSIE") != -1)
    {
        layer = ".all";
        style = ".style";
    }
    else
        if(navigator.userAgent.indexOf("Nav") != -1)
        {
            layer = ".layers";
            style = "";
        }
}
function f(strat,viz)
{
    eval("document" + layer + "[" + strat + "]" + style + ".visibility =

```

```

'" + viz + "'");
}
</script>
</head>
<body onload="VerifBrowser()">
<div class="stil"><br />
  <div class="titlul">Mihai Eminescu</div><br /><br />
  <div class="as" id="unu"> Diana<br /><br />
    <div class="s">Ce cauti unde bate luna<br />
    Pe-un alb izvor tremur&#259;tor<br />
    Si unde p&#259;s&#259;rile-ntruna<br />
    Se-ntrec cu glas ciripitor?<br />
    <br />
    N-auzi cum frunzele-n poian&#259;<br />
    Soptesc cu zgomotul de guri<br />
    Ce se s&#259;rut&#259;, se h&acirc;rjoan&#259;<br />
    &Icirc;n umbr-ad&acirc;nc&#259; de p&#259;duri?<br />
  </div> </div>
  <a class="one" href="diana.html" target="_blank" onMouseOver=
    "f('unu','visible')" onMouseOut="f('unu','hidden')"> Diana</a>
  <div class="as" id="doi"> Si dac&#259;... <br /><br />
    <div class="s">Si dac&#259; ramuri bat &icirc;n geam<br />
    Si se cutremur plopii,<br />
    E ca &icirc;n minte s&#259; te am<br />
    Si-ncet s&#259; te apropii.<br /><br />
    Si dac&#259; stele bat &icirc;n lac<br />
    Ad&acirc;ncu-i lumin&acirc;ndu-l,<br />
    E ca durerea mea s-o-mpac<br />
    &Icirc;nsein&acirc;ndu-mi g&acirc;ndul.<br />
  </div></div>
  <a class="one" href="SiDaca.html" target="_blank" onMouseOver=
    "f('doi','visible')" onMouseOut="f('doi','hidden')">Si dac&#259;...
    </a>
  <div class="as" id="trei"> Te duci... <br /><br />
    <div class="s">Te duci si ani de suferint&#259;<br />
    N-or sa te vaz&#259; ochi-mi tristi,<br />
    &Icirc;namorati de-a ta fiint&#259;,<br />
    De cum z&acirc;mbesti, de cum te misti.<br /><br />
    Si nu e bl&acirc;nd ca o poveste<br />
    Amorul meu cel dureros,<br />
    Un demon sufletul t&#259;u este<br />
    Cu chip de marmur&#259; frumos.<br />
  </div></div>
  <a class="one" href="TeDuci.html" target="_blank" onMouseOver=
    "f('trei','visible')" onMouseOut="f('trei','hidden')">Te duci...
    </a>
  </div>
</body>
</html>

```

## Location

Obiectul Location este o proprietate a obiectului *Window* și conține URL-ul complet al obiectului *Window* sau, dacă nu este specificat, al obiectului *Window* curent. Poate fi utilizat pentru a controla afișarea paginii web de către browser.

Proprietăți: *hash*, *host*, *hostname*, *href*, *pathname*, *port*, *protocol*, *search*.

Metode: *reload()*, *replace()*.

| Proprietate | Descriere  |
|-------------|--|
| hash        | Reprezintă un șir de caractere ce începe cu caracterul # (hash) ce specifică numele unei ancore.   |
| host        | Este un șir de caractere ce reprezintă denumirea host-ului, numele de domeniu sau adresa IP a unui host dintr-o rețea.                       |
| hostname    | Specifică porțiunea <i>host:port</i> a unui URL.   |
| href        | Reprezintă întregul URL.   |
| pathname    | Este un șir de caractere ce reprezintă o porțiune dintr-un URL ce specifică modalitatea în care pot fi accesate diverse resurse particulare. |
| port        | Este un șir de caractere ce specifică portul pe care îl utilizează server-ul.  |
| protocol    | Este un șir de caractere ce specifică începutul unui URL și include prima coloană ( :), ce reprezintă metoda de accesare a URL-ului.         |
| search      | Este un șir de caractere ce conține o întrebare.   |

| Metodă       | Descriere   |
|--------------|---|
| reload()     | Specifică faptul că fereastra curentă este reîncărcată.   |
| replace(URL) | Metoda replace are nevoie de URL ca parametru și încarcă documentul de la URL-ul specificat în partea de sus a documentului curent. |

**Zambile frumoase**



**Zambilă roz**

[Inapoi Inainte](#)

[Reincarcati pagina](#)

**Zambile frumoase**



**Zambilă albastră**

[Inapoi Inainte](#)

[Reincarcati pagina](#)

**Zambile frumoase**



**Zambilă albastră**

[Inapoi Inainte](#)

[Reincarcati pagina](#)

**Exemplu** În acest exemplu, am utilizat obiectul *Location*, care a permis reîncărcarea paginii documentului de mai sus.

```
<html>
<head>
  <title>Obiectul Location</title>
<style type="text/css">
body
{
  background-color: #D7D7FF;
}
a.stil:link
{
  color: #0000FF;
}
```

```

    background-color: #AEAEFF;
    font-size: 16px;
    text-decoration: none;
    font-weight: bolder;
    height: 18px;
    width: 30px;
}
a.stil:visited
{
    color: #0066FF;
    background-color: #AACCFE;
    font-size: 16px;
    text-decoration: none;
    font-weight: bolder;
    height: 18px;
    width: 30px;
}
a.stil:hover
{
    font-size: 18px;
    color: #000066;
    background-color: #7D7DFF;
    text-decoration: blink;
    font-weight: bolder;
    height: 21px;
    width: 33px;
}
.as
{
    color: #0000FF;
    background-color: #C6C6FF;
    visibility: hidden;
    position: absolute;
    left: 444px;
    top: 213px;
    width: 116px;
    height: 46px;
    font-size: 16px;
    font-weight: bolder;
    text-align: center;
    border: 1px groove #7171FF;
}
.titlul
{
    font-size: 20px;
    font-weight: bolder;
    color: #0000FF;
    text-align: center;
}
.buton
{
    color: #0000FF;
    background-color: #C6C6FF;
    font-size: 14px;
    font-weight: bolder;
    text-align: center;
    border: 1px groove #7171FF;
    width: 180px;
}

```

```

</style>
<script type="text/javascript" language="JavaScript">
var layer = new String();
var style = new String();
function VerifBrowser()
{
    if(navigator.userAgent.indexOf("MSIE") != -1)
        {
            layer = ".all";
            style = ".style";
        }
    else
        if(navigator.userAgent.indexOf("Nav") != -1)
            {
                layer = ".layers";
                style = "";
            }
}
function f(strat,viz)
{
    eval("document" + layer + "[" + strat + "]" + style + ".visibility =
        '" + viz + "'");
}
var imagini = new Array ("zambila_alba.jpg", "zambila_albastra.jpg",
    "zambila_roz.jpg");
var nr = 0;
var nrTot = imagini.length;
function inapoi()
{
    if (document.images)
    {
        if (nr > 0) nr--
        else
            nr = nrTot-1
        document.images.slides.src = imagini[nr]
        switch(nr)
        { case 0: {f('unu','visible');
            f('doi','hidden');
            f('trei','hidden');}break;
        case 1: {f('unu','hidden');
            f('doi','visible');
            f('trei','hidden');}break;
        case 2: {f('unu','hidden');
            f('doi','hidden');
            f('trei','visible');}break;
        }
    }
}
function inainte()
{
    if (document.images){
        if (nr < nrTot-1)
            nr++
        else
            nr = 0
        document.images.slides.src = imagini[nr]
        switch(nr)
        { case 0: {f('unu','visible');
            f('doi','hidden');

```



```

        f('trei','hidden');
    }break;
    case 1: {f('unu','hidden');
        f('doi','visible');
        f('trei','hidden');
        }break;
    case 2: {f('unu','hidden');
        f('doi','hidden');
        f('trei','visible');
        }break;
    }
}
}
</script>
</head>
<body onload="VerifBrowser()">
<div class="titlul">Zambile frumoase<br /><br />
<div class="as" id="unu"> <br />Zambil&#259; alb&#259;
</div>
<div class="as" id="doi"> <br />Zambil&#259; albastr&#259;
</div>
<div class="as" id="trei"> <br />Zambil&#259; roz
</div>
<IMG SRC="zambila_alba.jpg" NAME="slides"> <br /><br /><br /><br />
<A HREF="javascript:inapoi()" class="stil">Inapoi</A>
<A HREF="javascript:inainte()" class="stil">Inainte</A>
<br /> <br />
<input type="button" value="Reincarcati pagina" onClick=
"window.location.reload()" class="buton">
</div>
</body>
</html>

```

## Math

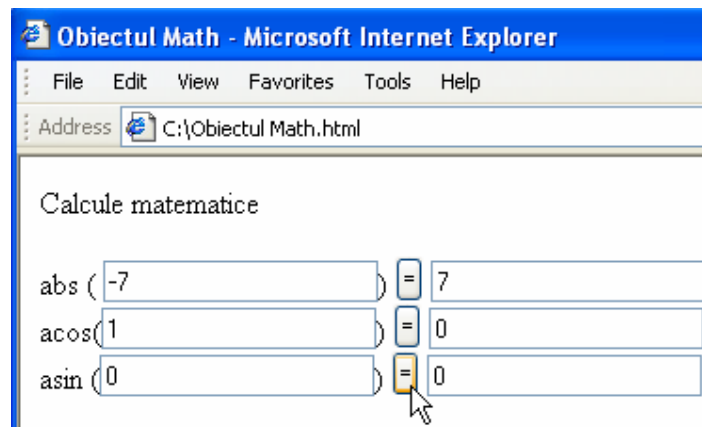
Obiectul Math este printre cele mai importante obiecte JavaScript și permite efectuarea unor calcule matematice cu ajutorul unor proprietăți și metode JavaScript.

Proprietăți: *E*, *LN10*, *LN2*, *LOG10E*, *LOG2E*, *PI*, *SQRT1\_2*, *SQRT2*.

Metode: *abs()*, *acos()*, *asin()*, *atan()*, *atan2()*, *ceil()*, *cos()*, *exp()*, *floor()*, *log()*, *max()*, *min()*, *pow()*, *random()*, *round()*, *sin()*, *sqrt()*, *tan()*.

| Proprietate | Descriere   |
|-------------|---|
| E           | Reprezintă constanta lui Euler, care este aproximativ 2.7182.   |
| LN10        | Este o constantă ce reprezintă logaritm natural din 10, care este aproximativ 2.3025.   |
| LN2         | Este o constantă ce reprezintă logaritm natural din 2, care este aproximativ 0.6931.  |
| LOG10E      | Este o constantă ce reprezintă logaritm în baza 10 din e, care este aproximativ 0.4342.   |
| LOG2E       | Este o constantă ce reprezintă logaritm în baza 2 din e, care este aproximativ 1.4426.  |
| PI          | Este o constantă denumită <i>pi</i> ce reprezintă rezultatul împărțirii lungimii cercului la diametru, care este aproximativ 3.1415.. |
| SQRT1_2     | Este o constantă ce reprezintă radical pătrat din ½, care este aproximativ 0.7071.  |
| SQRT2       | Este o constantă ce reprezintă radical pătrat din 2, care este aproximativ 1.4142.  |

| Metodă   | Descriere   |
|----------|---|
| abs()    | Returnează valoarea absolută dintr-un număr.  |
| acos()   | Returnează valoarea arccosinusului în radiani a unui număr.   |
| asin()   | Returnează valoarea arcsinusului în radiani a unui număr.   |
| atan()   | Returnează valoarea arctangentei în radiani a unui număr.   |
| atan2()  | Returnează o valoare numerică (cuprinsă între $-\pi$ și $\pi$ ce reprezintă unghiul dintre axa OX și punctul (x,y)), ce reprezintă arctangentul unui număr. |
| ceil()   | Returnează cel mai apropiat întreg mai mare sau egal cu numărul respectiv.  |
| cos()    | Returnează valoarea cosinusului unui număr, adică un număr cuprins între -1 și 1.   |
| exp()    | Returnează $e^{\text{număr}}$ , unde <i>număr</i> este argument și <i>e</i> este constanta lui Euler, baza logaritmilor naturali.                           |
| floor()  | Returnează cel mai mare întreg care e mai mic sau egal cu un număr dat.   |
| log()    | Returnează logaritm natural (în baza e) a unui număr.   |
| max()    | Returnează cel mai mare număr dintre două numere date.  |
| min()    | Returnează cel mai mic număr dintre două numere date.   |
| pow()    | Returnează un număr ridicat la o putere, adică baza <sup>exponent</sup>   |
| random() | Returnează un număr aleator între zero și unu.  |
| round()  | Returnează o valoare numerică rotunjită către cel mai apropiat întreg.  |
| sin()    | Returnează sinusul unui număr.  |
| sqrt()   | Returnează o valoare numerică pozitivă ce reprezintă rădăcina pătrată (radical) dintr-o valoare numerică.   |
| tan()    | Returnează tangenta unui număr.   |



**Exemplu** În acest exemplu, am utilizat obiectul Math pentru a efectua câteva calcule matematice, cum ar fi: calcularea valorii absolute, a arccosinusului și a arcsinusului unui număr introdus de către utilizator.

```
<html>
<head>
  <title>Obiectul Math</title>
<script type="text/javascript" language="JavaScript">
function Ec_abs(x)
  {
    var x
    document.formular.val2.value=Math.abs(x)
  }
function Ec_acos(x)
```

```

{
  var x
  document.formular.val4.value=Math.acos(x)
}
function Ec_asin(x)
{
  var x
  document.formular.val6.value=Math.asin(x)
}
</script>
</head>
<body>
Calcule matematice<br />
<form name="formular" method="post" action="">
  abs ( <input type="text" name="val1" value="">
  <input type="Button" name="ex1" value="" onClick="Ec_abs
    (this.form.val1.value)">
  <input type="text" name="val2" value=""><br />
  acos(<input type="text" name="val3" value="">
  <input type="Button" name="ex2" value="" onClick="Ec_acos
    (this.form.val3.value)">
  <input type="text" name="val4" value=""> <br />
  asin (<input type="text" name="val5" value="">
  <input type="Button" name="ex3" value="" onClick="Ec_asin
    (this.form.val5.value)">
  <input type="text" name="val6" value="">
</form>
</body>
</html>

```

## MimeType

Obiectul MimeType este o proprietate a obiectului navigator și permite accesarea unor informații despre tipurile MIME (Multipart Internet Mail Extension) suportate de client. Proprietăți: *description*, *enabledPlugin*, *suffixes*, *type*.

| Proprietate   | Descriere                                      |
|---------------|--|
| description   | Reprezintă descrierea tipului MIME.            |
| enabledPlugin | Reprezintă obiectul Plugin pentru tipul MIME.  |
| suffixes      | Reprezintă extensia numelui pentru tipul MIME. |
| type          | Reprezintă tipul MIME.                         |

Informații despre tipul MIME al computerului client.

```

tip=text/html
descriere=HTML
sufix=*
enabledPlugin=Trident Plugin for Netscape

```

**Exemplu** În acest exemplu, am utilizat browser-ul Netscape și obiectul *MimeType* pentru a afișa informații despre tipul MIME al computerului client.

```

<html>
<head>
  <title>Obiectul MimeType</title>
</head>
<body>
Informatii despre tipul MIME al computerului client.<br /><br />

```

```

<script type="text/javascript" language="JavaScript">
document.writeln("tip=",navigator.mimeTypes[0].type,
    "<br />descriere=",navigator.mimeTypes[0].description,
    "<br />sufix=",navigator.mimeTypes[0].suffixes,
    "<br />enabledPlugin=",navigator.mimeTypes[0].enabledPlugin.name);
</script>
</body>
</html>

```

## Navigator

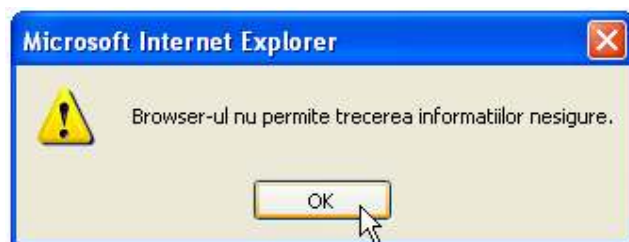
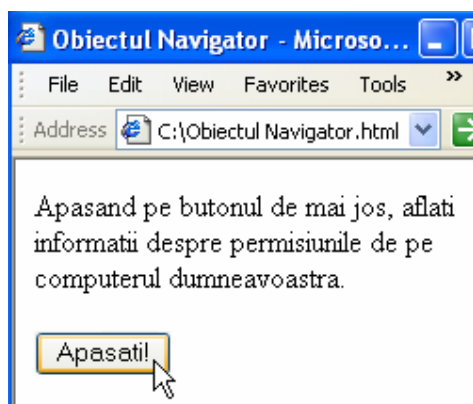
Obiectul Navigator conține informații cu privire la versiunea browser-ului pe care îl folosește utilizatorul.

Proprietăți: *appCodeName*, *appName*, *appVersion*, *language*, *mimeTypes*, *platform*, *plugins*, *userAgent*.

Metode: *javaEnabled()*, *plugins.refresh*, *preference*, *savePreference*, *taintEnabled*.

| Proprietate | Descriere  |
|-------------|--|
| appCodeName | Este un șir de caractere ce reprezintă numele de cod al browser-ului.  |
| appName     | Este un șir de caractere ce returnează numele browser-ului.  |
| appVersion  | Este un șir de caractere ce returnează informații cu privire la versiunea browser-ului.                          |
| language    | Este un șir de caractere format din două litere, ce conține informații cu privire la limba utilizată de browser. |
| mimeTypes   | Reprezintă un array ce conține toate tipurile MIME suportate de către client.                                    |
| platform    | Este un șir ce specifică tipul computerului pe care browser-ul a fost compilat.                                  |
| plugins     | Reprezintă un array ce conține toate plugins instalate pe computerul client.                                     |
| userAgent   | Este un șir ce specifică valoarea <i>user-agent header</i> trimis de client către server prin protocolul HTTP.   |

| Metodă          | Descriere   |
|-----------------|---|
| javaEnabled()   | Testează dacă pe computerul client sunt sau nu permise applet-uri Java. |
| plugins.refresh | Face să fie disponibilă instalarea celor mai noi plugins.               |
| preference      | Permite scripturilor să seteze preferințele browser-ului.               |
| savePreference  | Salvează preferințele browser-ului într-un fișier local prefs.js.       |
| taintEnabled    | Specifică dacă sunt permise date nesigure (posibil infectate).          |



**Exemplu** În acest exemplu, am utilizat obiectul Navigator pentru a afișa informații despre permisiunile setate pe computerul client.

```
<html>
<head>
  <title>Obiectul Navigator</title>
<script type="text/javascript" language="JavaScript">
function informatii()
{
  if (navigator.taintEnabled())
  {
    alert("Browser-ul permite trecerea informatiilor nesigure."); }
  else
    alert("Browser-ul nu permite trecerea informatiilor nesigure.");
}
</script>
</head>
<body>
<form name="formular" method="post" action="">
Apasand pe butonul de mai jos, aflati informatii despre permisiunile de pe
computerul dumneavoastra.<br /><br />
  <input type="submit" name="Submit" value="Apasati!" onClick=
    "informatii()">
</form>
</body>
</html>
```

## Netscape

Obiectul Netscape este utilizat pentru a accesa clase Java din pachetul netscape.

## Number

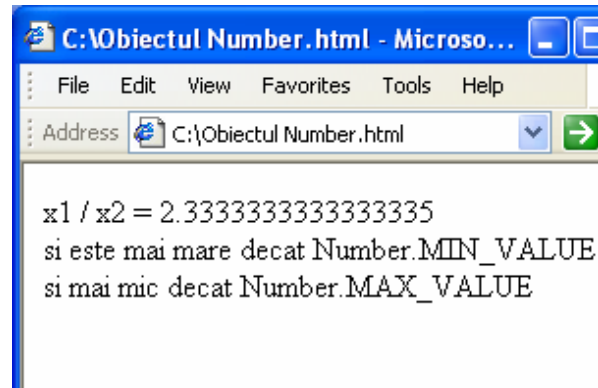
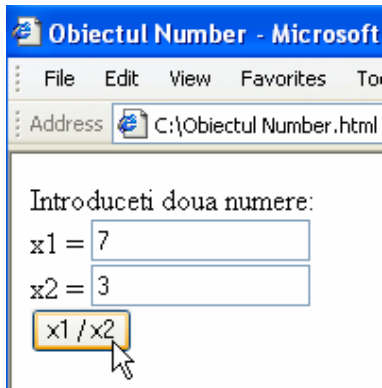
Obiectul Number permite lucrul cu valori numerice.

Proprietăți: *constructor*, *MAX\_VALUE*, *MIN\_VALUE*, *NaN*, *NEGATIVE\_INFINITY*, *POSITIVE\_INFINITY*, *prototype*.

Metode: *toSource()*, *toString()*, *valueOf()*.

| Proprietate       | Descriere  |
|-------------------|--|
| constructor       | Reprezintă funcția ce creează prototype unui obiect.   |
| MAX_VALUE         | Specifică cea mai mare valoare numerică reprezentabilă în JavaScript.  |
| MIN_VALUE         | Specifică cea mai mică valoare numerică reprezentabilă în JavaScript.  |
| NaN               | Specifică o valoare numerică ce reprezintă NaN (nici un număr).  |
| NEGATIVE_INFINITY | Specifică o valoare ce reprezintă minus infinit (-Infinity).   |
| POSITIVE_INFINITY | Specifică o valoare ce reprezintă plus infinit (Infinity).   |
| prototype         | Specifică prototipul pentru un obiect și permite adăugarea unor proprietăți sau metode tuturor instanțelor unei clase. |

| Metodă     | Descriere  |
|------------|--|
| toSource() | Returnează un șir de caractere ce reprezintă codul sursă al obiectului.                      |
| toString() | Returnează un șir de caractere ce reprezintă obiectul <i>Number</i> specificat.              |
| valueOf(). | Returnează un șir de caractere ce reprezintă valoarea primitivă a obiectului <i>Number</i> . |



**Exemplu** În acest exemplu, am utilizat obiectul *Number* pentru a face o comparație între un număr obținut prin împărțirea a două numere introduse de către utilizator și elementele *MIN\_VALUE*, respectiv *MAX\_VALUE*.

```
<html>
<head>
  <title>Obiectul Number</title>
<script type="text/javascript" language="JavaScript">
function Calcul ()
{
  y1=formular.x1.value;
  y2=formular.x2.value;
  var s=y1/y2;
  if (s<Number.MIN_VALUE)
    document.writeln(" x1 / x2 = ",s,"<br />si este mai mic decat
      Number.MIN_VALUE.");
  else
    if ((s>Number.MIN_VALUE) && (s<Number.MAX_VALUE))
      document.writeln(" x1 / x2 = ",s,"<br />si este mai mare decat
        Number.MIN_VALUE si mai mic decat Number.MAX_VALUE");
    else
      document.writeln(" x1 / x2 = ",s,"<br />si este mai mare decat
        Number.MAX_VALUE");
}
</script></head>
<body>
<form name="formular">
Introduceti doua numere:<br>
x1 = <input type="text" name="x1" value="" size="15"><br>
x2 = <input type="text" name="x2" value="" size="15"><br>
<input type="button" name="buton" value="x1 / x2"
  onClick="Calcul(this.form)">
</form>
</body></html>
```

## Object

Obiectul *Object* este un tip de obiect primitiv, din care se obțin toate celelalte obiecte. Proprietăți: *constructor*, *prototype*. Metode: *eval()*, *toSource()*, *toString()*, *unwatch()*, *valueOf()*, *watch()*.

| Proprietate | Descriere  |
|-------------|--|
| constructor | Reprezintă funcția ce creează prototype unui obiect și este moștenit de către toate obiectele. |
| prototype   | Specifică prototipul pentru un obiect și permite adăugarea unor                                |

proprietăți sau metode oricărui obiect.

| Metodă     | Descriere  |
|------------|--|
| eval()     | Această metodă este utilizată pentru a evalua un șir de caractere dintr-un cod JavaScript, în cadrul unui obiect specificat. |
| toSource() | Poate fi utilizat pentru a crea un nou obiect.   |
| toString() | Returnează un șir de caractere ce reprezintă un obiect specificat.   |
| unwatch()  | Înlătură watchpoint din proprietatea obiectului, cu ajutorul metodei <i>watch</i> .  |
| valueOf()  | Returnează o valoare primitivă a obiectului specificat.  |
| watch()    | Adaugă un watchpoint proprietății obiectului.  |

**Exemplu** În acest exemplu am creat un obiect ce reține diverse informații despre mai multe persoane.

```
<html>
<head>
  <title>Object</title>
<style type="text/css">
body
{
  font-size: 14px;
  color: #5B5BFF;
  background-color: #D2D2FF;
}
.titlul
{
  font-family: "Georgia", "Times";
  color: #0000FF;
  font-size: 18px;
  text-align: center;
  font-weight: bolder;
}
</style>
<script language="JavaScript" type=
"text/javascript">
function DatePersonale()
{
x1 = "<b>Nume: </b>" + this.nume + "<br>\n";
x2 = "<b>Prenume: </b>" + this.prenume +
  "<br>\n";
x3 = "<b>Telefon: </b>" + this.telefon +
  "<br>\n";
x4 = "<b>Oras: </b>" + this.oras + "<br>\n";
x5 = "<b>Tara: </b>" + this.tara + "<hr>\n";
document.write(x1, x2, x3, x4, x5);
}
function
  Structura (nume, prenume, telefon, oras, tara)
{
  this.nume = nume;
  this.prenume = prenume;
  this.telefon = telefon;
  this.oras = oras;
  this.tara = tara;
```

**Continutul obiectului**

---

Nume: Chivu  
Prenume: Ingrid  
Telefon: 0212111111  
Oras: Bucuresti  
Tara: Romania

---

Nume: Nemtanu  
Prenume: Mihaela  
Telefon: 0327111111  
Oras: Pascani  
Tara: Romania

---

Nume: Diaconu  
Prenume: Inda  
Telefon: 0245111111  
Oras: Targoviste  
Tara: Romania

---

```

        this.DatePersonale = DatePersonale;
    }
</script>
</head>
<body>
<div          class="titlul">Continutul
    obiectului</div><br />
<br />
<script      language="JavaScript"      type=
    "text/javascript">
chivu = new Structura("Chivu", "Ingrid",
    "0212111111", "Bucuresti", "Romania");
nemtanu = new Structura("Nemtanu", "Mihaela",
    "0327111111", "Pascani", "Romania");
diaconu = new Structura("Diaconu", "Inda",
    "0245111111", "Targoviste", "Romania");
chivu.DatePersonale();
nemtanu.DatePersonale();
diaconu.DatePersonale();
</script>
</body>
</html>

```

## Option

Obiectul Option este o opțiune dintr-o listă de selecție.

Proprietăți: *defaultSelected*, *index*, *length*, *selected*, *text*, *value*.

| Proprietate     | Descriere   |
|-----------------|---|
| defaultSelected | Este o valoare de tip <i>Boolean</i> , ce arată când o opțiune a fost declarată cu atributul HTML SELECTED.                         |
| index           | Este o valoare de tip <i>Boolean</i> , ce arată când o opțiune este selectată, returnând true dacă este selectată și false dacă nu. |
| length          | Reprezintă numărul de elemente din select.options array.  |
| selected        | Specifică starea curentă a selecției unei opțiuni.  |
| text            | Specifică textul pentru opțiune.  |
| value           | Specifică valoarea care este returnată către server când opțiunea este selectată și este trimis formularul.                         |

**Exemplu** În acest exemplu am utilizat obiectul *Option* pentru a afișa o listă de selecție ce conține mai multe opțiuni. Exemplu permite dezactivarea listei de selecție și reactivarea lui, înlăturarea unui element din listă și afișarea unui element selectat din listă, într-un câmp de tip text.

```

<html>
<head>
  <title>
    Obiectul Option
  </title>
</head>
<style type="text/css">
body
{
  color: #006600;
  background-color: #E6FFE6;
  font-size: 18px;
  font-weight: bolder;

```





## Packages

Obiectul Packages permite accesarea metodelor și claselor Java din interiorul unui cod JavaScript.

Proprietăți: *className*, *java*, *netscape*, *sun*.

| Proprietate      | Descriere   |
|------------------|---|
| <i>className</i> | Reprezintă numele clasei Java din packages.                 |
| <i>java</i>      | Reprezintă orice clasă din Java package <i>java</i> .*.     |
| <i>netscape</i>  | Reprezintă orice clasă din Java package <i>netscape</i> .*. |
| <i>sun</i>       | Reprezintă orice clasă din Java package <i>sun</i> .*.      |

## Password

Obiectul Password reprezintă un câmp de tip password dintr-un formular HTML. Caracterele pe care utilizatorul le introduce în acest câmp sunt ascunse prin afișarea unor caractere sub forma unor buline •.

Proprietăți: *defaultValue*, *form*, *name*, *type*, *value*.

Metode: *blur()*, *focus()*, *handleEvent()*, *select()*.

Evenimente: *onBlur*, *onFocus*.

| Proprietate         | Descriere   |
|---------------------|---|
| <i>defaultValue</i> | Setează și returnează valoarea inițială a câmpului password.  |
| <i>form</i>         | Aceasta este o referință către formularul din care face parte obiectul <i>Password</i> .                  |
| <i>name</i>         | Reprezintă atributul <i>name</i> al obiectului <i>Password</i> .  |
| <i>type</i>         | Reflectă tipul oricărui obiect din formular. În cazul obiectului <i>Password</i> <i>type</i> ="password". |
| <i>value</i>        | Reprezintă valoarea introdusă în câmpul password de către utilizator.                                     |

| Metodă               | Descriere  |
|----------------------|--|
| <i>blur()</i>        | Înlătură focalizarea de pe un obiect.                                    |
| <i>focus()</i>       | Dă focalizare asupra unui obiect.  |
| <i>handleEvent()</i> | Apelează handler pentru un eveniment specificat.                         |
| <i>select()</i>      | Permite selectarea cu ajutorul cursorului a obiectului <i>Password</i> . |

| Eveniment      | Descriere   |
|----------------|---|
| <i>onBlur</i>  | Acțiune ce va fi declanșată când se pierde focalizarea de pe o legătură de tip hipertext. |
| <i>onFocus</i> | Acțiune ce va fi declanșată de focalizarea asupra unei legături de tip hipertext.         |

**Exemplu** În acest exemplu am utilizat obiectul *Password* pentru a introduce și a verifica datele de autentificare introduse de către utilizator. Dacă utilizatorul nu introduce corect parola, apare un mesaj de atenționare în acest sens.

```
<html>
<head>
<title>Obiectul Password</title>
<style type="text/css">
body
{
  font-size: 18px;
  font-weight: bolder;
  color: #BF2600;
  background-color: #FFD5CA;
}
```

```

}
.buton
{
  color: #BF2600;
  background-color: #FFD5CA;
  text-align: center;
}
.x1
{
  position: absolute;
  height: 30px;
  width: 164px;
  left: 102px;
  top: 10px;
}
.x2
{
  position: absolute;
  height: 30px;
  width: 164px;
  left: 102px;
  top: 46px;
}
.t
{
  font-family: Geneva, Arial, Helvetica, sans-serif;
  color: #BF2600;
  font-size: 14px;
  background-color: #FF8A6C;
  border: 1px outset #FF3300;
  position: absolute;
  height: 159px;
  width: 283px;
  left: 9px;
  top: 57px;
}
</style>
<script type="text/javascript" language="JavaScript">
function Autenticare()
{
  var utiliz=document.Aut.utiliz.value;
  var parola=document.Aut.parola.value;
  utiliz=utiliz.toLowerCase();
  parola=parola;
if (utiliz=="diana")
  if (parola=="pARola")
    {
      window.location="diana.html";
    }
  else
    alert("Nu ati introdus corect\n parola!");
else
  alert("Nu ati introdus corect\n numele de utilizator!");
}
</script>
</head>
<body>
<form name="Aut" method="post" >
Introduceti datele de autentificare!<br />

```



|       |   |
|-------|---|
| form  | Reprezintă formularul din care face parte grupul de butoane radio ce au același nume.                               |
| name  | Reprezintă atributul name al unui singur buton radio.   |
| type  | Reprezintă tipul butonului, în cazul nostru fiind vorba despre "radio".   |
| value | Reprezintă valoarea care va fi trimisă de către formular, în cazul în care a fost selectat butonul radio respectiv. |

| Metodă        | Descriere   |
|---------------|---|
| blur()        | Înlătură focalizarea de pe un buton radio din lista de selecție.                            |
| click()       | Acționează asupra unui buton radio, în momentul în care se declanșează evenimentul onClick. |
| focus()       | Dă focalizare asupra unui buton radio din lista de selecție.                                |
| handleEvent() | Invocă handler pentru un eveniment specificat.  |

| Eveniment | Descriere   |
|-----------|---|
| onBlur    | Acțiune ce va fi declanșată când se pierde focalizarea de pe o legătură de tip hipertext. |
| onClick   | Acțiune ce va fi declanșată când se dă click stânga pe mouse.                             |
| onFocus   | Acțiune ce va fi declanșată de focalizarea asupra unei legături de tip hipertext.         |

### Alegeti culori pentru fundal si text!

#### Fundal

Culoarea aleasa:

Valoarea:

- aquamarine
- azure
- black
- blue
- brown
- chocolate
- coral
- cyan
- gold
- gray
- green
- indigo
- ivory
- khaki
- magenta
- maroon
- navy
- orange
- pink
- purple
- red
- royalblue
- silver
- skyblue
- snow
- tomato
- turquoise
- violet
- white
- yellow

#### Text

Culoarea aleasa:

Valoarea:

- aquamarine
- azure
- black
- blue
- brown
- chocolate
- coral
- cyan
- gold
- gray
- green
- indigo
- ivory
- khaki
- magenta
- maroon
- navy
- orange
- pink
- purple
- red
- royalblue
- silver
- skyblue
- snow
- tomato
- turquoise
- violet
- white
- yellow

**Exemplu** În acest exemplu am utilizat obiectul *Radio* pentru a selecta culori pentru fundalul și textul unei pagini Web.

```
<html>
<head>
  <title>Obiectul Radio</title>
</style>
```

```

.titlul
{ font-size: 20px;
  font-weight: bolder;
  text-align: center; }

.s_t
{ font-size: 18px;
  font-weight: bolder;
  text-align: center; }

.still
{ font-size: 14px;
  position: absolute;
  visibility: visible;
  height: 400px;
  width: 275px;
  left: 160px;
  top: 60px; }

.still1
{ font-size: 14px;
  position: absolute;
  visibility: visible;
  height: 277px;
  width: 130px;
  left: 0px;
  top: 108px; }

.still2
{ font-size: 14px;
  position: absolute;
  visibility: visible;
  height: 267px;
  width: 130px;
  left: 144px;
  top: 108px; }

.stil2
{ font-size: 14px;
  position: absolute;
  visibility: visible;
  height: 400px;
  width: 275px;
  left: 550px;
  top: 60px; }

.stil21
{ font-size: 14px;
  position: absolute;
  visibility: visible;
  height: 277px;
  width: 130px;
  left: 0px;
  top: 108px; }

.stil22
{ font-size: 14px;
  position: absolute;
  visibility: visible;
  height: 267px;
  width: 130px;
  left: 144px;
  top: 108px; }

</style>
<script type="text/javascript" language="JavaScript">
function CuloareF(form, CFundal)

```

```

{
var ValCuloare = " ";
if (CFundal == 'aquamarine') ValCuloare = "#7FFFD4";
if (CFundal == 'azure') ValCuloare = "#F0FFFF";
if (CFundal == 'black') ValCuloare = "#000000";
if (CFundal == 'blue') ValCuloare = "#0000FF";
if (CFundal == 'brown') ValCuloare = "#A52A2A";
if (CFundal == 'chocolate') ValCuloare = "#D2691E";
if (CFundal == 'coral') ValCuloare = "#FF7F50";
if (CFundal == 'cyan') ValCuloare = "#00FFFF";
if (CFundal == 'gold') ValCuloare = "#FFD700";
if (CFundal == 'gray') ValCuloare = "#808080";
if (CFundal == 'green') ValCuloare = "#008000";
if (CFundal == 'indigo') ValCuloare = "#4B0082";
if (CFundal == 'ivory') ValCuloare = "#FFFFFF0";
if (CFundal == 'khaki') ValCuloare = "#F0E68C";
if (CFundal == 'magenta') ValCuloare = "#FF00FF";
if (CFundal == 'maroon') ValCuloare = "#800000";
if (CFundal == 'navy') ValCuloare = "#000080";
if (CFundal == 'orange') ValCuloare = "#FFA500";
if (CFundal == 'pink') ValCuloare = "#FFC0CB";
if (CFundal == 'purple') ValCuloare = "#800080";
if (CFundal == 'red') ValCuloare = "#FF0000";
if (CFundal == 'royalblue') ValCuloare = "#4169E1";
if (CFundal == 'silver') ValCuloare = "#C0C0C0";
if (CFundal == 'skyblue') ValCuloare = "#87CEEB";
if (CFundal == 'snow') ValCuloare = "#FFFAFA";
if (CFundal == 'tomato') ValCuloare = "#FF6347";
if (CFundal == 'turquoise') ValCuloare = "#40E0D0";
if (CFundal == 'violet') ValCuloare = "#EE82EE";
if (CFundal == 'white') ValCuloare = "#FFFFFF";
if (CFundal == 'yellow') ValCuloare = "#FFFF00";
document.bgColor = CFundal;
form.CName.value = CFundal;
form.CValue.value = ValCuloare;
}
function CuloareT(form, CText)
{
var ValCuloare = " ";
if (CText == 'aquamarine') ValCuloare = "#7FFFD4";
if (CText == 'azure') ValCuloare = "#F0FFFF";
if (CText == 'black') ValCuloare = "#000000";
if (CText == 'blue') ValCuloare = "#0000FF";
if (CText == 'brown') ValCuloare = "#A52A2A";
if (CText == 'chocolate') ValCuloare = "#D2691E";
if (CText == 'coral') ValCuloare = "#FF7F50";
if (CText == 'cyan') ValCuloare = "#00FFFF";
if (CText == 'gold') ValCuloare = "#FFD700";
if (CText == 'gray') ValCuloare = "#808080";
if (CText == 'green') ValCuloare = "#008000";
if (CText == 'indigo') ValCuloare = "#4B0082";
if (CText == 'ivory') ValCuloare = "#FFFFFF0";
if (CText == 'khaki') ValCuloare = "#F0E68C";
if (CText == 'lime') ValCuloare = "#00FF00";
if (CText == 'magenta') ValCuloare = "#FF00FF";
if (CText == 'maroon') ValCuloare = "#800000";
if (CText == 'navy') ValCuloare = "#000080";
if (CText == 'orange') ValCuloare = "#FFA500";
if (CText == 'pink') ValCuloare = "#FFC0CB";

```

```

    if (CText == 'purple') ValCuloare = "#800080";
    if (CText == 'red') ValCuloare = "#FF0000";
    if (CText == 'royalblue') ValCuloare = "#4169E1";
    if (CText == 'silver') ValCuloare = "#C0C0C0";
    if (CText == 'skyblue') ValCuloare = "#87CEEB";
    if (CText == 'snow') ValCuloare = "#FFFAFA";
    if (CText == 'tomato') ValCuloare = "#FF6347";
    if (CText == 'turquoise') ValCuloare = "#40E0D0";
    if (CText == 'violet') ValCuloare = "#EE82EE";
    if (CText == 'white') ValCuloare = "#FFFFFF";
    if (CText == 'yellow') ValCuloare = "#FFFF00";
    document.fgColor=CText;
    form.TName.value = CText;
    form.TValue.value = ValCuloare;
}
</script>
</head>
<body>
<div class="titlul">Alegeti o culoare pentru fundal!</div>
<FORM name=Fundal method=post class="still">
    <span class="s_t">Fundal</span><br /><br />
    Culoarea aleasa: <INPUT size=21 value="aquamarine" name=CName>
Valoarea: <INPUT size=12 value=#7FFFD4 name=CValue><br />
<div class="still1">
    <INPUT onclick="CuloareF(this.form, 'aquamarine')" type="radio"
        name="Fundal">aquamarine<br />
    <INPUT onclick="CuloareF(this.form, 'azure')" type="radio"
        name="Fundal">azure<br />
    <INPUT onclick="CuloareF(this.form, 'black')" type="radio"
        name="Fundal">black<br />
    <INPUT onclick="CuloareF(this.form, 'blue')" type="radio"
        name="Fundal">blue<br />
    <INPUT onclick="CuloareF(this.form, 'brown')" type="radio"
        name="Fundal">brown<br />
    <INPUT onclick="CuloareF(this.form, 'chocolate')" type="radio"
        name="Fundal">chocolate<br />
    <INPUT onclick="CuloareF(this.form, 'coral')" type="radio"
        name="Fundal">coral<br />
    <INPUT onclick="CuloareF(this.form, 'cyan')" type="radio"
        name="Fundal">cyan<br />
    <INPUT onclick="CuloareF(this.form, 'gold')" type="radio"
        name="Fundal">gold<br />
    <INPUT onclick="CuloareF(this.form, 'gray')" type="radio"
        name="Fundal">gray<br />
    <INPUT onclick="CuloareF(this.form, 'green')" type="radio"
        name="Fundal">green <br />
    <INPUT onclick="CuloareF(this.form, 'indigo')" type="radio"
        name="Fundal">indigo<br />
    <INPUT onclick="CuloareF(this.form, 'ivory')" type="radio"
        name="Fundal">ivory<br />
    <INPUT onclick="CuloareF(this.form, 'khaki')" type="radio"
        name="Fundal">khaki <br />
    <INPUT onclick="CuloareF(this.form, 'magenta')" type="radio"
        name="Fundal">magenta<br>
</div>
<div class="still2">
    <INPUT onclick="CuloareF(this.form, 'maroon')" type="radio"
        name="Fundal">maroon<br />
    <INPUT onclick="CuloareF(this.form, 'navy')" type="radio"

```



```

        name="Fundal">navy<br />
<INPUT onclick="CuloareF(this.form, 'orange')" type="radio"
    name="Fundal">orange<br />
<INPUT onclick="CuloareF(this.form, 'pink')" type="radio"
    name="Fundal">pink<br />
<INPUT onclick="CuloareF(this.form, 'purple')" type="radio"
    name="Fundal">purple<br />
<INPUT onclick="CuloareF(this.form, 'red')" type="radio"
    name="Fundal">red<br />
<INPUT onclick="CuloareF(this.form, 'royalblue')" type="radio"
    name="Fundal">royalblue<br />
<INPUT onclick="CuloareF(this.form, 'silver')" type="radio"
    name="Fundal">silver<br />
<INPUT onclick="CuloareF(this.form, 'skyblue')" type="radio"
    name="Fundal">skyblue<br />
<INPUT onclick="CuloareF(this.form, 'snow')" type="radio"
    name="Fundal">snow<br />
<INPUT onclick="CuloareF(this.form, 'tomato')" type="radio"
    name="Fundal">tomato<br />
<INPUT onclick="CuloareF(this.form, 'turquoise')" type="radio"
    name="Fundal">turquoise<br />
<INPUT onclick="CuloareF(this.form, 'violet')" type="radio"
    name="Fundal">violet<br />
<INPUT onclick="CuloareF(this.form, 'white')" type="radio"
    name="Fundal">white<br />
<INPUT onclick="CuloareF(this.form, 'yellow')" type="radio"
    name="Fundal">yellow
</div>
</FORM><br />
<FORM name=Text method=post class="stil2">
    <span class="s_t">Text</span><br /><br />
    Culoarea aleasă: <INPUT size=21 value="aquamarine" name=TName>
    Valoarea: <INPUT size=12 value=#7FFFD4 name=TValue><br />
<div class="stil21">
    <INPUT onclick="CuloareT(this.form, 'aquamarine')" type="radio"
        name="Text">aquamarine<br />
    <INPUT onclick="CuloareT(this.form, 'azure')" type="radio"
        name="Text">azure<br />
    <INPUT onclick="CuloareT(this.form, 'black')" type="radio"
        name="Text">black<br />
    <INPUT onclick="CuloareT(this.form, 'blue')" type="radio"
        name="Text">blue<br />
    <INPUT onclick="CuloareT(this.form, 'brown')" type="radio"
        name="Text">brown<br />
    <INPUT onclick="CuloareT(this.form, 'chocolate')" type="radio"
        name="Text">chocolate<br />
    <INPUT onclick="CuloareT(this.form, 'coral')" type="radio"
        name="Text">coral<br />
    <INPUT onclick="CuloareT(this.form, 'cyan')" type="radio"
        name="Text">cyan<br />
    <INPUT onclick="CuloareT(this.form, 'gold')" type="radio"
        name="Text">gold<br />
    <INPUT onclick="CuloareT(this.form, 'gray')" type="radio"
        name="Text">gray<br />
    <INPUT onclick="CuloareT(this.form, 'green')" type="radio"
        name="Text">green <br />
    <INPUT onclick="CuloareT(this.form, 'indigo')" type="radio"
        name="Text">indigo<br />
    <INPUT onclick="CuloareT(this.form, 'ivory')" type="radio"

```

```

        name="Text">ivory<br />
<INPUT onclick="CuloareT(this.form, 'khaki')" type="radio"
        name="Text">khaki <br />
<INPUT onclick="CuloareT(this.form, 'magenta')" type="radio"
        name="Text">magenta<br />
</div>
<div class="stil22">
<INPUT onclick="CuloareT(this.form, 'maroon')" type="radio"
        name="Text">maroon<br />
<INPUT onclick="CuloareT(this.form, 'navy')" type="radio"
        name="Text">navy<br />
<INPUT onclick="CuloareT(this.form, 'orange')" type="radio"
        name="Text">orange<br />
<INPUT onclick="CuloareT(this.form, 'pink')" type="radio"
        name="Text">pink<br />
<INPUT onclick="CuloareT(this.form, 'purple')" type="radio"
        name="Text">purple<br />
<INPUT onclick="CuloareT(this.form, 'red')" type="radio"
        name="Text">red<br />
<INPUT onclick="CuloareT(this.form, 'royalblue')" type="radio"
        name="Text">royalblue<br />
<INPUT onclick="CuloareT(this.form, 'silver')" type="radio"
        name="Text">silver<br />
<INPUT onclick="CuloareT(this.form, 'skyblue')" type="radio"
        name="Text">skyblue<br />
<INPUT onclick="CuloareT(this.form, 'snow')" type="radio"
        name="Text">snow<br />
<INPUT onclick="CuloareT(this.form, 'tomato')" type="radio"
        name="Text">tomato<br />
<INPUT onclick="CuloareT(this.form, 'turquoise')" type="radio"
        name="Text">turquoise<br />
<INPUT onclick="CuloareT(this.form, 'violet')" type="radio"
        name="Text">violet<br />
<INPUT onclick="CuloareT(this.form, 'white')" type="radio"
        name="Text">white<br />
<INPUT onclick="CuloareT(this.form, 'yellow')" type="radio"
        name="Text">yellow
</div>
</FORM>
</body>
</html>

```

## RegExp

Obiectul RegExp conține structura unei expresii regulate și este utilizat pentru a găsi și pentru a înlocui șiruri de caractere.

Proprietăți: *constructor*, *global*, *ignoreCase*, *input*, *lastIndex*, *lastMatch*, *lastParen*, *multiline*, *prototype*, *source*.

Metode: *compile()*, *exec()*, *test()*, *toSource()*, *toString()*, *valueOf()*.

| Proprietate | Descriere   |
|-------------|---|
| constructor | Specifică funcția ce creează prototype unui obiect.   |
| global      | Se utilizează pentru a testa dacă un caracter se găsește sau nu într-un șir de caractere.                                 |
| ignoreCase  | Returnează <i>true</i> dacă se utilizează în cadrul unei expresii și <i>false</i> în cazul în care nu se utilizează.      |
| input       | Este un șir de caractere care specifică dacă un caracter se găsește sau nu la sfârșitul unui șir de caractere. De exemplu |

|           |  |
|-----------|--|
|           | /r\$/ găsește caracterul r în șirul « mar ».   |
| lastIndex | Returnează un întreg care specifică poziția care urmează imediat după un subșir pe care îl căutăm în cadrul unui șir de caractere dat. |
| lastMatch | Returnează ultima potrivire a caracterelor.  |
| lastParen | Returnează ultima potrivire a caracterelor care se află între paranteze.   |
| multiline | Returnează <i>true</i> dacă facem o căutare într-un șir cu mai multe linii și <i>false</i> dacă avem o singură linie.                  |
| prototype | Reprezintă prototipul pentru această clasă și permite adăugarea propriilor proprietăți și metode tuturor obiectelor.                   |
| source    | Conține codul sursă al unei expresii obișnuite.  |

| Metodă            | Descriere  |
|-------------------|--|
| <i>compile()</i>  | Compilează o expresie în timpul execuției unui script.   |
| <i>exec()</i>     | Execută și caută un subșir într-un șir de caractere.   |
| <i>test()</i>     | Testează dacă se găsește un subșir într-un șir de caractere și returnează <i>true</i> sau <i>false</i> . |
| <i>toSource()</i> | Returnează un șir de caractere ce reprezintă codul sursă al obiectului <i>RegExp</i> .                   |
| <i>toString()</i> | Returnează un șir de caractere ce reprezintă șirul obiectului specificat.                                |
| <i>valueOf()</i>  | Returnează un șir de caractere ce reprezintă valoarea primitivă a obiectului <i>RegExp</i> .             |

**Exemplu** În acest exemplu am utilizat obiectul *RegExp* pentru a afișa poziția imediat următoare subșirului "ci" în cadrul șirului inițial.

```
<html>
<head>
  <title>Obiectul RegExp</title>
<script type="text/javascript" language=
"JavaScript">
str = "unu_doi_trei_patru_cinci";
expr=/ci*/g;
masiv=expr.exec(str);
document.writeln("<b>Sirul      initial
este:</b> "+str+"<br />");
document.writeln("Subsirul      <b>"+
masiv[0]+</b> se regaseste in sirul
initial      la      pozitia:      <b>"      +
expr.lastIndex+"</b>");
</script>
</head>
<body>
</body></html>
```

**Sirul initial este:** unu\_doi\_trei\_patru\_cinci  
Subsirul ci se regaseste in sirul initial la pozitia: 21

**Exemplu** În acest exemplu am utilizat obiectul *RegExp* pentru a testa dacă am utilizat **g** pentru o căutare globală a caracterului "a" în șirul de caractere dat. Dacă rezultatul căutării este pozitiv, metoda **global** returnează *true*, altfel returnează *false*.

```
<html>
<head>
  <title>Obiectul RegExp</title>
<script type="text/javascript" language=
"JavaScript">
expr = /a*/g;
```

**Sirul initial este:** unu\_doi\_trei\_patru\_cinci  
Se utilizeaza g pentru prelucrarea sirului de caractere.

```

str = "unu_doi_trei_patru_cinci";
masiv=expr.exec(str);
document.writeln("<b>Sirul initial este:
</b>"+str+"<br />");
if (expr.global)
document.writeln("Se utilizeaza g pentru
prelucrarea sirului de caractere.<br
/>");
else
document.writeln("Nu se utilizeaza g.")
</script>
</head>
<body>
</body>
</html>

```

## Reset

Obiectul Reset este un buton de tipul "reset", creat cu ajutorul etichetei <INPUT> din limbajul HTML. Cu acest buton, se aduc toate elementele din formular la valoarea inițială.

Proprietăți: *form, name, type, value*.

Metode: *blur(), click(), focus(), handleEvent()*.

Evenimente: *onBlur, onClick, onFocus*.

| Proprietate | Descriere   |
|-------------|---|
| form        | Reprezintă formularul din care face parte butonul <i>reset</i> .  |
| name        | Reprezintă atributul name al butonului <i>reset</i> .   |
| type        | Reprezintă tipul butonului, în cazul nostru fiind vorba despre "reset".   |
| value       | Reprezintă valoarea care va fi trimisă de către formular, în cazul în care a fost selectat butonul reset respectiv. |

| Metodă        | Descriere  |
|---------------|--|
| blur()        | Înlătură focalizarea de pe butonul <i>reset</i> .  |
| click()       | Acționează asupra butonului <i>reset</i> , în momentul în care se declanșează evenimentul <i>onClick</i> . |
| focus()       | Se focalizează asupra butonului <i>reset</i> .   |
| handleEvent() | Invocă handler pentru un eveniment specificat.   |

| Eveniment | Descriere  |
|-----------|--|
| onBlur    | Acțiune ce va fi declanșată când se pierde focalizarea de pe butonul <i>reset</i> .    |
| onClick   | Acțiune ce va fi declanșată când se dă click stânga de mouse pe butonul <i>reset</i> . |
| onFocus   | Acțiune ce va fi declanșată de focalizarea asupra unui buton <i>reset</i> .            |

**Exemplu** În acest exemplu am utilizat obiectul *Reset* în cadrul unui formular pentru a putea anula, dacă dorim informațiile scrise în formular, înainte de a le trimite.

```

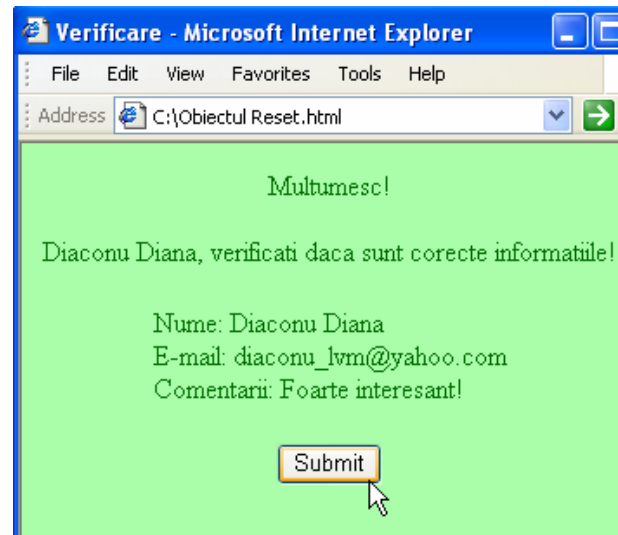
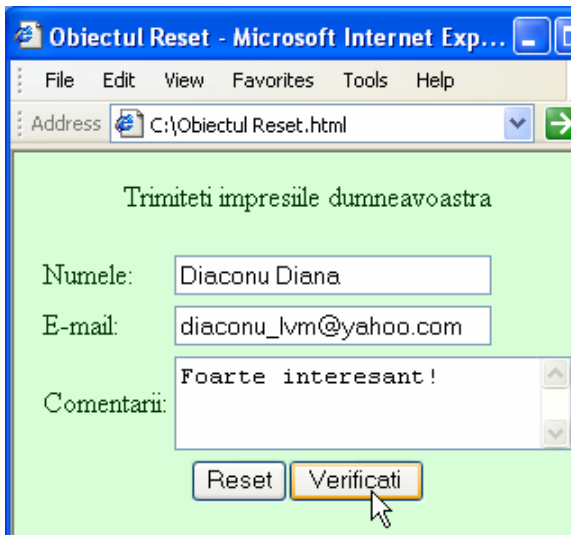
<html>
<head>
  <title>Obiectul Reset</title>
<style>
body
{
  background-color: #D7FFD7;

```

```

    color: #003300;
    text-align: center;
}
</style>
<script type="text/javascript" language="JavaScript">
function Verifica()
{
    fereastra = window.document;
    nume = fereastra.formular.nume.value
    mail = fereastra.formular.mail.value
    comentariu = fereastra.formular.comentariu.value
    fereastra.open();
    fereastra.write("<html><head><title>Verificare</title><style>body{
        background-color: #AAFFAA; color:#006C00;        text-align:
        center;}</style></head><body>");
    fereastra.write("Multumesc!<br /><br />");
    fereastra.write("" + nume + ", verificati daca sunt corecte
        informatiile! <br /><br />");
    fereastra.write("<table border=0><tr><td>Nume: " + nume + "<br>E-mail: "
        + mail + "<br>Comentarii: " + comentariu + "</td></tr></table><br
        />");
    fereastra.write("<form                method=post                action='mailto:
        enctype='text/plain'><input                type=submit
        value=Submit></body></html>");
}
</script>
</head>
<body>
    Trimiteti impresiile dumneavoastra
<FORM name=formular>
<table>
    <tr>
        <td> Numele: </td>
        <td><INPUT size=25 name="nume"></td>
    </tr>
    <tr>
        <td> E-mail:</td>
        <td><INPUT size=25 name="mail"></td>
    </tr>
    <tr>
        <td vAlign=center> Comentarii:</td>
        <td><TEXTAREA name="comentariu" rows=3 cols=25>
            </TEXTAREA></td>
    </tr>
    <tr align="center">
        <td align=middle colSpan=2 >
            <INPUT type="reset" value="Reset">
            <INPUT onclick=Verifica() type="button" value="Verificati"></td>
    </tr>
</table>
</FORM>
</body>
</html>

```



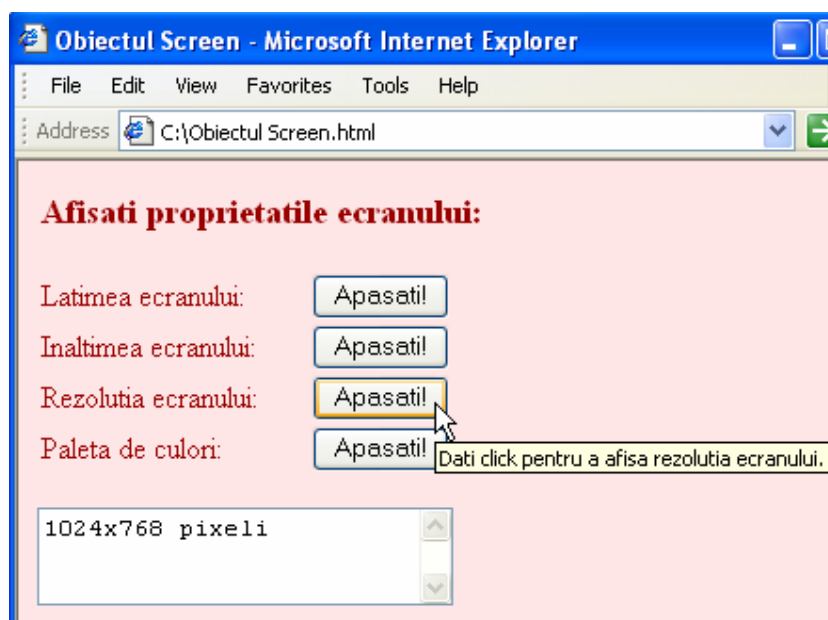
## Screen

Obiectul Screen permite ajustarea culorilor și dimensiunii ecranului.

Metode: *availHeight()*, *availWidth()*, *colorDepth()*, *height()*, *pixelDepth()*, *width()*.

| Metodă               | Descriere  |
|----------------------|--|
| <i>availHeight()</i> | Returnează înălțimea ecranului, în pixeli, din care se scade dimensiunea componentelor interfeței sistemului de operare. |
| <i>availWidth()</i>  | Returnează lățimea ecranului, în pixeli, din care se scade dimensiunea componentelor interfeței sistemului de operare.   |
| <i>colorDepth()</i>  | Returnează calitatea culorilor, în bit (ex: 16 bit, 32 bit..).   |
| <i>height()</i>      | Returnează înălțimea ecranului, în pixeli.   |
| <i>pixelDepth()</i>  | Returnează rezoluția culorilor ecranului, în bit / pixel.  |
| <i>width()</i>       | Returnează lățimea ecranului, în pixeli.   |

**Exemplu** În acest exemplu am utilizat obiectul *Screen* în cadrul unui formular pentru a putea afișa câteva din proprietățile ecranului.



```

<html>
<head>
  <title>Obiectul Screen</title>
<style type="text/css">
body
{
  font-size: 14px;
  color: #990000;
  background-color: #FFE6E6;
}
.titlul
{
  font-size: 18px;
  font-weight: bolder;
  text-align: center;
}
</style>
<script language="JavaScript">
function Proprietati(x)
{
  switch(x)
  {
    case 1: formular.text.value=screen.width+" pixeli"; break;
    case 2: formular.text.value=screen.height+" pixeli"; break;
    case 3: formular.text.value=screen.width+"x"+ screen.height+" pixeli";
            break;
    case 4: formular.text.value=screen.colorDepth+" bit"; break;
  }
}
</script>
</head>
<body>
<form name="formular" method="post" action="">
<table border="0" cellspacing="1" cellpadding="1">
<tr>
  <td colspan="2" class="titlul">Afisati proprietatile ecranului:<br /><br
  /></td>
</tr>
<tr>
  <td>Latimea ecranului:</td>
  <td><input type="Button" name="ex1" value="Apasati!"
    onClick="Proprietati(1)" title="Dati click pentru a afisa latimea
    ecranului."></td>
</tr>
<tr>
  <td>Inaltimea ecranului:</td>
  <td><input type="Button" name="ex2" value="Apasati!"
    onClick="Proprietati(2)" title="Dati click pentru a afisa inaltimea
    ecranului."></td>
</tr>
<tr>
  <td>Rezolutia ecranului:</td>
  <td><input type="Button" name="ex3" value="Apasati!"
    onClick="Proprietati(3)" title="Dati click pentru a afisa rezolutia
    ecranului."></td>
</tr>
<tr>
  <td>Paleta de culori:</td>
  <td><input type="Button" name="ex4" value="Apasati!"

```

```

        onClick="Proprietati(4)" title="Dati click pentru a afisa paleta de
        culori."></td>
</tr>
</table><br />
    <textarea name="text" cols="25" rows="3"></textarea>
</form>
</body>
</html>

```

## Select

Obiectul Select permite selectarea unor elemente ce se află într-o listă de selecție, creată cu ajutorul etichetei <SELECT> din limbajul HTML.

Proprietăți: *form*, *length*, *name*, *options*, *selectedIndex*, *type*.

Metode: *blur()*, *focus()*, *handleEvent()*.

Evenimente: *onBlur*, *onChange*, *onFocus*.

| Proprietate   | Descriere   |
|---------------|---|
| form          | Reprezintă formularul din care face parte lista de selecție.  |
| length        | Reprezintă numărul de opțiuni din lista de selecție.  |
| name          | Reprezintă atributul <i>name</i> al listei de selecție.   |
| options       | Reprezintă un element al unui masiv ce conține toate opțiunile listei de selecție și se încadrează între etichetele <option> și </option>.  |
| selectedIndex | Reprezintă un întreg ce corespunde elementului sau elementelor selectate dintr-o listă de selecție. Elementele din lista de selecție sunt numerotate începând de la zero (0).                             |
| type          | Reprezintă tipul obiectului Select, în cazul nostru fiind vorba despre "select-one", atunci când poate fi selectat un singur element sau "select-multiple", atunci când putem selecta mai multe elemente. |

| Metodă        | Descriere   |
|---------------|---|
| blur()        | Înlătură focalizarea de pe lista de selecție.             |
| focus()       | Se focalizează asupra unui element din lista de selecție. |
| handleEvent() | Invocă handler pentru un eveniment specificat.            |

| Eveniment | Descriere   |
|-----------|---|
| onBlur    | Acțiune ce va fi declanșată când se pierde focalizarea de pe o listă de selecție. |
| onChange  | Acțiune ce va fi declanșată la schimbarea unui element dintr-un formular.         |
| onFocus   | Acțiune ce va fi declanșată de focalizarea asupra unei liste de selecție.         |

```

<html>
<head>
  <title>Obiectul Select</title>
<style type="text/css">
.titlul
{
  font-family: Georgia, "Times New Roman", Times, serif;
  font-size: 18px;
  font-weight: bolder;
  color: #006600;

```



```

    background-color: #DFFFDF;
    text-align: center;
}
</style>
<script type="text/javascript" language="JavaScript">
function f(i)
{
    var s=1;
    var x1=formular.test1.selectedIndex;
    var x2=formular.test2.selectedIndex;
    var x3=formular.test3.selectedIndex;
    var x4=formular.test4.selectedIndex;
    var x5=formular.test5.selectedIndex;
    var x6=formular.test6.selectedIndex;
    var x7=formular.test7.selectedIndex;
    var x8=formular.test8.selectedIndex;
    var x9=formular.test9.selectedIndex;
    if (x1==1) s=s+1;
    if (x2==0) s=s+1;
    if (x3==1) s=s+1;
    if (x4==3) s=s+1;
    if (x5==2) s=s+1;
    if (x6==2) s=s+1;
    if (x7==0) s=s+1;
    if (x8==3) s=s+1;
    if (x9==1) s=s+1;
formular.text.value=s
}
</script>
</head>
<body>
<form name="formular" method="post" action="" >
    <table border="0" cellspacing="1" cellpadding="1">
        <tr>
            <td colspan="2" class="titlul">Alegeti raspunsurile corecte la
intrebarile
                de mai jos<br /> </td>
            </tr>
            <tr>
                <td>In interiorul caror etichete se incadreaza codul sursa
JavaScript?</td>
                <td><select name="test1" id="s1">
                    <option value="o1" selected>< js >< /js ></option>
                    <option value="o2">< script >< /script ></option>
                    <option value="o3">< JavaScript >< /JavaScript ></option>
                    <option value="o4">< java >< /java ></option>
                </select></td>
            </tr>
            <tr>
                <td>Unde este corect sa positionam codul JavaScript?</td>
                <td><select name="test2" id="select">
                    <option value="o5" selected>In sectiunile: < head > sau < body
></option>
                    <option value="o6">In sectiunea < title ></option>
                    <option value="o7">In sectiunea < head ></option>
                    <option value="o8">In sectiunea < body ></option>
                </select></td>
            </tr>
        </tr>
    </table>

```

```

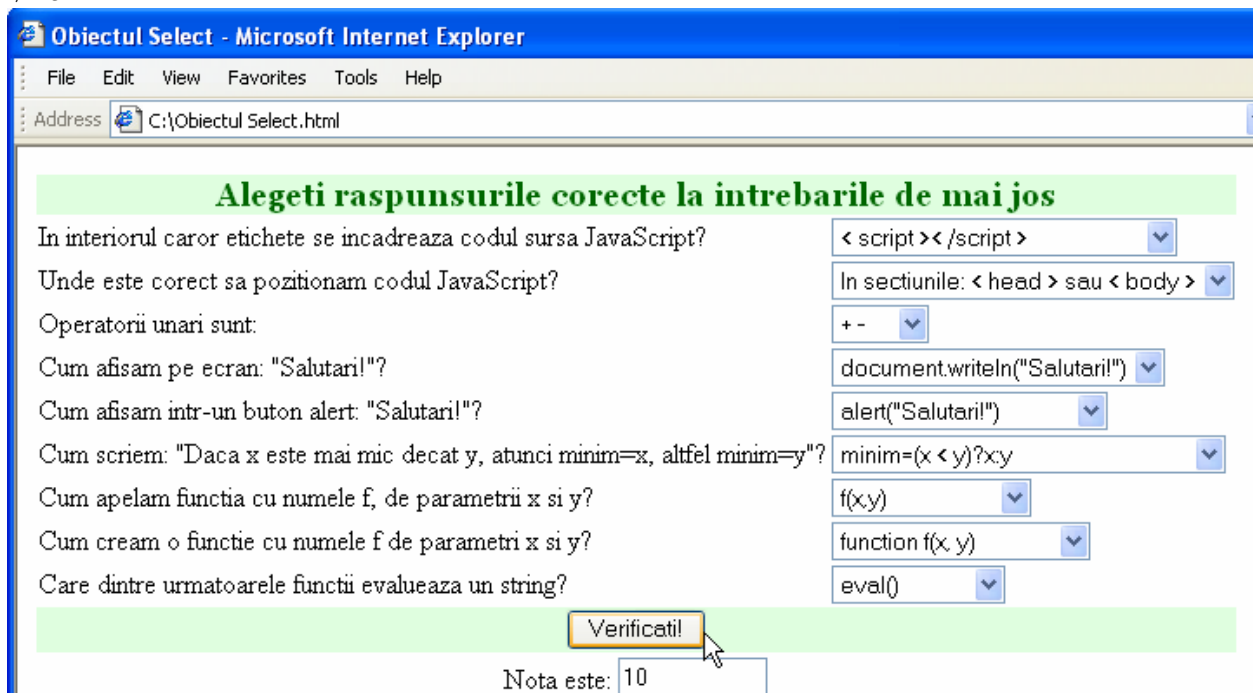
<td>Operatorii unari sunt: </td>
<td><select name="test3" id="select2">
  <option value="o9" selected> + / %</option>
  <option value="o10"> + -</option>
  <option value="o11"> * /</option>
  <option value="o12"> + - %</option>
</select></td>
</tr>
<tr>
  <td>Cum afisam pe ecran: &quot;Salutari!&quot;;? </td>
  <td><select name="test4" id="select3">
    <option value="o13" selected>writeln("Salutari!")</option>
    <option value="o14">document.writeln(Salutari!)</option>
    <option value="o15">alert("Salutari!")</option>
    <option value="o16">document.writeln("Salutari!")</option>
  </select></td>
</tr>
<tr>
  <td>Cum afisam intr-un buton alert: &quot;Salutari!&quot;;?</td>
  <td><select name="test5" id="select4">
    <option value="o17" selected>writeln.alert("Salutari!")</option>
    <option value="o18">msgBox("Salutari!")</option>
    <option value="o19">alert("Salutari!")</option>
    <option value="o20">alert.writeln("Salutari!")</option>
  </select></td>
</tr>
<tr>
  <td>Cum scriem: &quot;Daca x este mai mic decat y, atunci minim=x, altfel minim=y&quot;;?</td>
  <td><select name="test6" id="select5">
    <option value="o21" selected>if(x < y) then minim=x else minim=y</option>
    <option value="o22">(x < y)?minim=x:minim=y</option>
    <option value="o23">minim=(x < y)?x:y</option>
    <option value="o24">if x< y minim=x else minim=y</option>
  </select></td>
</tr>
<tr>
  <td>Cum apelam functia cu numele f, de parametrii x si y?</td>
  <td><select name="test7" id="select6">
    <option value="o25" selected>f(x,y)</option>
    <option value="o26">f parametrii x, y</option>
    <option value="o27">function f(x,y)</option>
    <option value="o28">f(x and y)</option>
  </select></td>
</tr>
<tr>
  <td>Cum cream o functie cu numele f de parametri x si y?</td>
  <td><select name="test8" id="select7">
    <option value="o29" selected>function:x,y</option>
    <option value="o30">function=f(x, y)</option>
    <option value="o31">function f(var x, var y)</option>
    <option value="o32">function f(x, y)</option>
  </select></td>
</tr>
<tr>
  <td>Care dintre urmatoarele functii evalueaza un string?</td>
  <td><select name="test9" id="select8">

```

```

        <option value="o33" selected>String()</option>
        <option value="o34">eval()</option>
        <option value="o35">Number()</option>
        <option value="o36">parseFloat()</option>
    </select></td>
</tr>
<tr>
    <td colspan="2" class="titlul"><input name="button" type="button"
onClick="f(this.form)" value="Verificati!" checked></td>
</tr>
<tr>
    <td colspan="2" align="center">Nota este: <input type="text"
name="text" value="" size="10"></td>
</tr>
</table>
</form>
</div>
</body>
</html>

```



## String

Obiectul String reprezintă o serie de caractere dintr-un string pe care îl încadrăm între ghilimele simple sau duble.

Proprietăți: *constructor*, *length*, *prototype*.

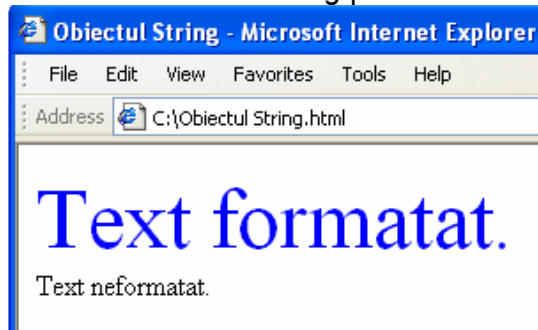
Metode: *anchor()*, *big()*, *blink()*, *bold()*, *charAt()*, *charCodeAt()*, *concat()*, *fixed()*, *fontcolor()*, *fontSize()*, *fromCharCode()*, *indexOf()*, *italics()*, *lastIndexOf()*, *link()*, *match()*, *replace()*, *search()*, *slice()*, *small()*, *split()*, *strike()*, *sub()*, *substr()*, *substring()*, *sup()*, *toLowerCase()*, *toSource()*, *toString()*, *toUpperCase()*, *valueOf()*.

| Proprietate | Descriere   |
|-------------|---|
| constructor | Specifică funcția care creează prototype unui obiect.     |
| length      | Reprezintă lungimea unui șir de caractere.                |
| prototype   | Permite adăugarea unor proprietăți și metode unui obiect. |

| Metodă         | Descriere   |
|----------------|---|
| anchor()       | Este utilizată pentru a crea o ancoră HTML.   |
| big()          | Returnează șiruri de caractere cu font mare, ca și etichetele <big> și </big> din HTML.                                       |
| blink()        | Returnează șiruri de caractere care clipesc, ca și etichetele <blink> și </blink> din HTML.                                   |
| bold()         | Returnează șiruri de caractere îngroșate, ca și etichetele <b> și </b> din HTML.  |
| charAt()       | Returnează un caracter din indexul specificat.  |
| charCodeAt()   | Returnează un număr care indică valoarea <i>Unicode</i> a unui caracter din indexul specificat.                               |
| concat()       | Concatenează textul conținut în două șiruri de caractere, returnând un alt șir de caractere.                                  |
| fixed()        | Returnează șiruri de caractere într-un format diferit, ca și etichetele <tt> și </tt> din HTML.                               |
| fontcolor()    | Afișează șiruri de caractere într-o culoare specificată, ca și etichetele <FONT COLOR="culoare"> și </FONT> din HTML.         |
| fontSize()     | Afișează șiruri de caractere de o dimensiune specificată, ca și etichetele <FONT SIZE="dimensiune font"> și </FONT> din HTML. |
| fromCharCode() | Returnează un șir de caractere format în urma transformării valorilor Unicode.  |
| indexOf()      | Returnează indicele primei apariții a unui caracter într-un șir de caractere sau -1 dacă nu este găsit caracterul respectiv.  |
| italics()      | Returnează șiruri de caractere înclinate, ca și etichetele <i> și </i> din HTML.  |
| lastIndexOf()  | Returnează indicele ultimei apariții a unui șir de caractere într-un șir de caractere sau 1 dacă nu este găsit șirul căutat.  |
| link()         | Creează o legătură de tip hipertext către un alt URL.   |
| match()        | Caută o expresie într-un șir de caractere și returnează expresia respectivă sau "null" dacă nu este găsită.                   |
| replace()      | Caută un subșir în cadrul unui șir. După ce acesta a fost găsit, îl înlocuiește cu un alt subșir.                             |
| search()       | Caută o expresie într-un șir de caractere și returnează poziția în cadrul șirului.  |
| slice()        | Extrage un subșir dintr-un șir de caractere și returnează noul șir.   |
| small()        | Returnează șiruri de caractere cu fonturi mici, ca și etichetele <small> și </small> din HTML.                                |
| split()        | Împarte un șir de caractere într-un array de șiruri prin separarea șirului inițial în subșiruri.                              |
| strike()       | Returnează șiruri de caractere tăiate, ca și etichetele <strike> și </strike> din HTML.                                       |
| sub()          | Returnează șiruri de caractere sub formă de indice (subscript), ca și etichetele <sub> și </sub> din HTML.                    |
| substr()       | Returnează caracterele dintr-un șir, începând dintr-un anumit loc, până la un număr specificat de caractere.                  |
| substring()    | Returnează caracterele dintr-un șir de caractere aflate între doi indici transmiși ca parametri.                              |
| sup()          | Returnează șiruri de caractere sub formă de putere (superscript), ca și etichetele <sup> și </sup> din HTML.                  |
| toLowerCase()  | Returnează șirul apelat scris cu litere mici (Lower Case).  |

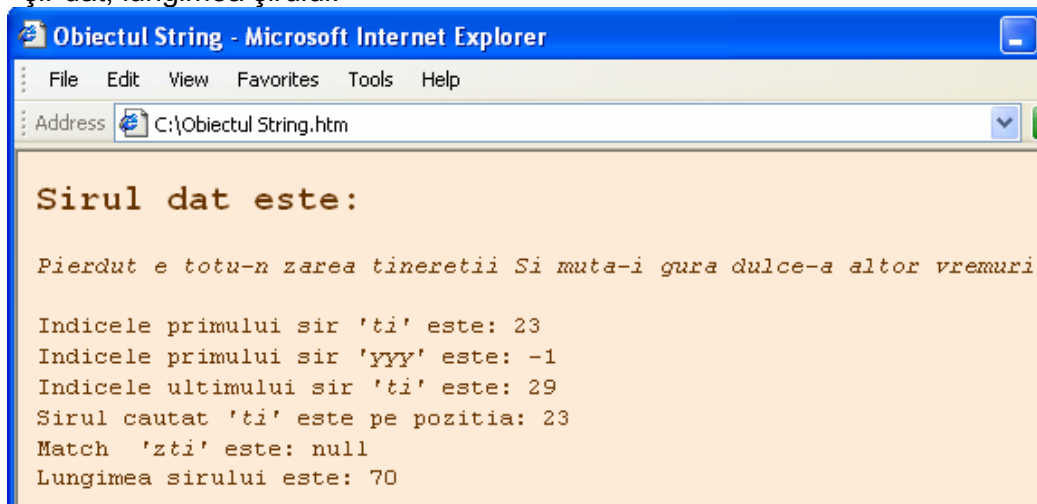
|               |  |
|---------------|--|
| toSource()    | Returnează un șir care reprezintă codul sursă al obiectului String.                  |
| toString()    | Returnează un șir care reprezintă obiectul specificat.                               |
| toUpperCase() | Returnează șirul apelat scris cu litere mari (Upper Case).                           |
| valueOf()     | Returnează un șir de caractere ce reprezintă valoarea primitivă a obiectului String. |

**Exemplu** În acest exemplu am utilizat obiectul *String* pentru a formata un text.



```
<html>
<head> <title>Obiectul String</title>
<script type="text/javascript" language="JavaScript">
function Efect ()
  { culoarea=" color='#0000FF'";
    dimensiunea=" size='10'";
    fontul="font" + culoarea+dimensiunea;
    text = this.toString();
    start = "<" + fontul + ">";
    stop = "</" + fontul + ">";
    return start + text + stop;
  }
String.prototype.formatare = Efect;
document.write ("Text formatat.".formatare());
</script>
</head>
<body>
<br />Text neformatat.
</body>
</html>
```

**Exemplu** În acest exemplu am utilizat obiectul *String* pentru a afișa indicele unui subșir într-un șir dat, lungimea șirului.



```
<html>
```

```

<head>
  <title>Obiectul String</title>
<style type="text/css">
body
  {
    color: #663300;
    background-color: #FFEBD7;
  }
.titlul
  {
    font-size: 20px;
    font-weight: bolder;
  }
</style>
</head>
<body>
<pre>
<div class="titlul">Sirul dat este:</div>
<script type="text/javascript" language="JavaScript">
var sir="Pierdut e totu-n zarea tineretii Si muta-i gura dulce-a altor
vremuri.";
document.writeln("<I>" + sir + "</I><br /><br />");
document.writeln("Indicele primului sir <I>'ti'</I> este: "+
sir.indexOf("ti"));
document.writeln("Indicele primului sir <I>'yyy'</I> este: "+
sir.indexOf("yyy"));
document.writeln("Indicele ultimului sir <I>'ti'</I> este: "+
sir.lastIndexOf("ti"));
document.writeln("Sirul cautat <I>'ti'</I> este pe pozitia: "+
sir.search("ti"));
document.writeln("Match <I>'zti'</I> este: "+sir.match("zti"));
document.writeln("Lungimea sirului este: "+sir.length);
</script></pre>
</body></html>

```

## Style

Obiectul Style specifică stilul elementelor HTML.

Proprietăți: *align, backgroundColor, backgroundImage, borderBottomWidth, borderColor, borderLeftWidth, borderRightWidth, borderStyle, borderTopWidth, clear, color, display, fontFamily, fontSize, fontStyle, fontWeight, lineHeight, listStyleType, marginBottom, marginLeft, marginRight, marginTop, paddingBottom, paddingLeft, paddingRight, paddingTop, textAlign, textDecoratation, textIndent, whiteSpace, width.*

Metode: *borderWidths(), margins(), paddings().*

| Proprietate       | Descriere   |
|-------------------|---|
| align             | Specifică alinierea unui element în HTML.                                   |
| backgroundColor   | Specifică o culoare pentru fundalul documentului.                           |
| backgroundImage   | Specifică imaginea pentru fundalul documentului.                            |
| borderBottomWidth | Specifică dimensiunea părții de jos a bordurii elementului specificat.      |
| borderColor       | Specifică o culoare pentru bordura elementului specificat.                  |
| borderLeftWidth   | Specifică dimensiunea părții din stânga a bordurii elementului specificat.  |
| borderRightWidth  | Specifică dimensiunea părții din dreapta a bordurii elementului specificat. |

|  |   |
|--|---|
| borderStyle  | Specifică stilul bordurii elementului specificat.   |
| borderTopWidth                                       | Specifică dimensiunea părții de sus a bordurii elementului specificat.  |
| clear  | Specifică latura pe care nu sunt permise <i>layer</i> -e.   |
| color  | Specifică o culoare pentru textul elementului.  |
| display  | Specifică dacă un element va fi afișat sau nu, iar dacă va fi afișat, care este modalitatea de afișare a lui.   |
| fontFamily   | Specifică fontul cu care va fi afișat șirul de caractere.   |
| fontSize   | Specifică dimensiunea cu care va fi afișat șirul de caractere.  |
| fontStyle  | Specifică stilul cu care va fi afișat șirul de caractere.   |
| fontWeight   | Specifică grosimea cu care va fi afișat un șir de caractere, ținând cont că grosimea normală este 400 și Bold este 700.   |
| lineHeight   | Specifică înălțimea liniei pe care este poziționat textul.  |
| listStyleType  | Specifică tipul listei cu care va fi formatat textul, cum ar fi: disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, none.   |
| marginBottom, marginLeft, marginRight, marginTop     | Specifică dimensiunea marginilor de jos, stânga, dreapta și sus dintre bordura elementului setat și alt element.  |
| paddingBottom, paddingLeft, paddingRight, paddingTop | Specifică dimensiunea marginilor de jos, stânga, dreapta și sus dintre conținutul elementului setat și bordura lui (sau marginea lui, în cazul în care nu are bordură).   |
| textAlign  | Specifică tipul de aliniere a șirului de caractere specificat.  |
| textDecoration                                       | Specifică adăugarea de anumite decorațiuni unui șir de caractere, cum ar fi: underline (subliniat), overline (o linie deasupra textului), line-through (o linie peste text), blink (clipește textul) sau none. Șirul de caractere obișnuit este setat inițial None, iar legăturile de tip hipertext, sunt setate Underline. |
| textIndent   | Specifică valoarea cu care va fi indentată prima linie din șirul de caractere.  |
| whiteSpace   | Specifică spațiul alb din element, dacă elementul se află între etichetele <pre> și </pre>.   |
| Width și Height                                      | Specifică lățimea și înălțimea elementului respectiv.   |

| Metodă         | Descriere  |
|----------------|--|
| borderWidths() | Returnează dimensiunea bordurii elementului respectiv.   |
| margins()      | Returnează dimensiunea marginilor de jos, stânga, dreapta și sus dintre bordura elementului setat și alt element.  |
| padding()      | Returnează dimensiunea marginilor de jos, stânga, dreapta și sus dintre conținutul elementului setat și bordura lui (sau marginea lui, în cazul în care nu are bordură). |

**Exemplu** În acest exemplu am utilizat obiectul Style pentru a afișa o imagine care are în fundal altă imagine, cu borduri de dimensiuni și culori diferite. Ca fundal al paginii, avem un gradient de culoare de la galben la roșu.

```
<html>
<head>
<title>Obiectul Style</title>
<style type="text/css">
.imagine
{
font-size: 20px;
font-weight: bolder;
```

```

background-image: url(trandafir1.gif);
border-top: 1px outset #0000FF;
border-right: 3px outset #FFFF00;
border-bottom: 5px outset #00CC00;
border-left: 7px outset #FF0000;
text-align: center;
clear: left;
height: 200px;
width: 350px;
vertical-align: middle;
display: ;
text-decoration: overline;
margin: 1px;
}
.strat
{
background-color: #FF0000;
height: 15px;
width: 140px;
left: 0px;
top: 0px;
font-size: 18px;
color: #FFFF00;
margin: 50px 10px;
text-decoration: blink;
white-space: nowrap;
font-weight: bolder;
font-family: Georgia, "Times New Roman", "Times", "serif";
font-style: normal;
text-align: center;
display: block;
list-style-type: square;
text-transform: lowercase;
}
</style>
</head>
<body style="filter:progid:DXImageTransform.Microsoft.Gradient(endColorstr='red',
startColorstr= 'yellow', gradientType='0');">
<div class="imagine"><br /><br /><br />

<pre>
<div class="strat">trandafiri rosii</div>
</pre>
</div>
</body>
</html>

```





## Submit

Obiectul Submit este un buton creat cu eticheta `<INPUT type="SUBMIT">` și poate fi acționat de către evenimentele asociate din formularul HTML care execută un anumit script specificat.

Proprietăți: *form, name, type, value*.

Metode: *blur(), click(), focus(), handleEvent()*.

Evenimente: *onBlur, onClick, onFocus*.

| Proprietate | Descriere   |
|-------------|---|
| form        | Obiectul form care include butonul de tip "submit".                                 |
| name        | Numele butonului.   |
| type        | Tipul obiectului, care poate fi: Button, Submit sau Reset, în cazul nostru: Submit. |
| value       | Valoarea butonului, care apare afișată pe buton.                                    |

| Metodă        | Descriere   |
|---------------|---|
| blur()        | Înlătură focalizarea de pe butonul "Submit" specificat.   |
| click()       | Această funcție acționează dacă utilizatorul dă click pe butonul "Submit" specificat dintr-un formular. |
| focus()       | Se focalizează asupra butonului "Submit" specificat.  |
| handleEvent() | Invocă handlerul pentru un anumit eveniment specificat.   |

| Eveniment | Descriere  |
|-----------|--|
| onBlur    | Execută codul JavaScript în momentul în care se înlătură focalizarea de pe butonul "Submit".   |
| onClick   | Execută codul JavaScript în momentul în care efectuăm click de mouse pe butonul "Submit" dintr-un formular.  |
| onFocus   | Execută codul JavaScript în momentul în care are loc evenimentul focus, adică atunci când o fereastră, un frame sau un frameset primește un focus. |

**Exemplu** În acest exemplu, am utilizat obiectul *Submit*, pentru a realiza un test cu mai multe variante de răspuns corecte.

Test - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address C:\Test.html

## Răspundeți prin alegerea variantei sau variantelor corecte

1. Un site Web este:

- o mulțime de pagini Web
- mai multe pagini Web legate între ele prin legături de tip hipertext sau hipermedia
- mai multe pagini Web care au o temă comună
- mai multe pagini Web legate între ele numai prin legături de tip hipertext

2. Organizarea liniară a informațiilor dintr-un site presupune:

- scrierea lor într-o anumită ordine stabilită de către autor
- structurarea informațiilor și organizarea lor într-o anumită ordine pe niveluri sau capitole
- scrierea informațiilor similar cu gândirea umană asociativă
- parcurgerea informațiilor în totalitate, de la început până la sfârșit

3. HTML reprezintă:

- un limbaj de marcare a textelor
- Hypertext Markup Language
- Hipermedia Markup Language
- Hypertext Markup Landscape

Apasati!



```
<html>
<head>
  <title>Test</title>
<style type="text/css">
body
{
  font-family: Verdana, Arial, Helvetica, sans-serif;
  background-color: #D2D2FF;
}
.buton
  Pagini Web cu JavaScript
```

Diana Elena Diaconu



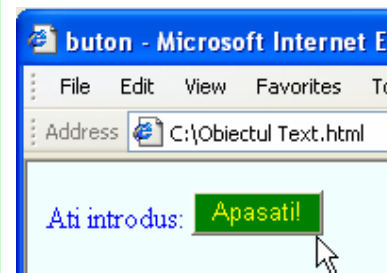
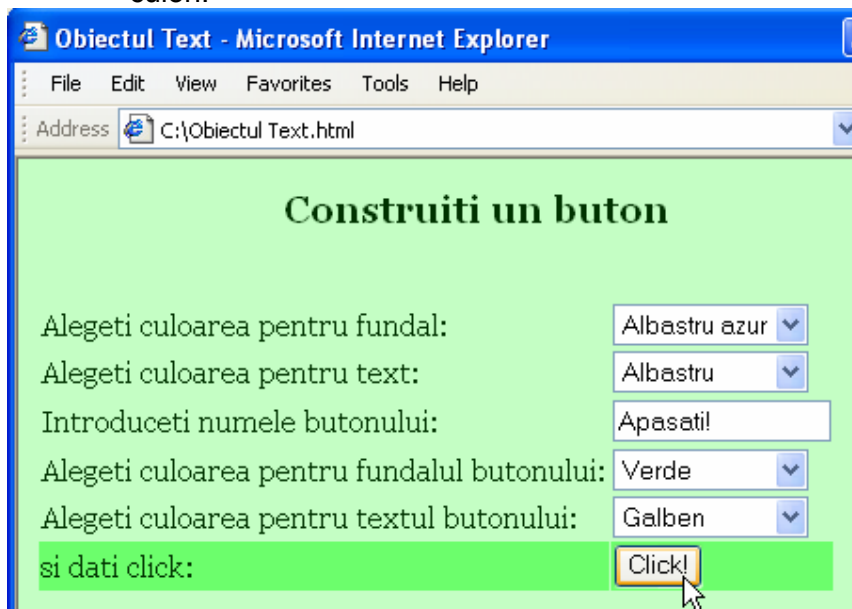


| Proprietate  | Descriere   |
|--------------|---|
| defaultValue | Setează și returnează valoarea inițială a câmpului de tip text. |
| form         | Obiectul form care include câmpul de tip text.                  |
| name         | Numele butonului.   |
| type         | Tipul obiectului, care în cazul nostru este: Text.              |
| value        | Valoarea obiectului Text.                                       |

| Metodă        | Descriere   |
|---------------|---|
| blur()        | Înlătură focalizarea de pe câmpul de tip text specificat.   |
| focus()       | Dă focalizare asupra câmpului de tip text specificat.   |
| handleEvent() | Invocă handlerul pentru un anumit eveniment specificat.   |
| select()      | Permite selectarea cu ajutorul cursorului a întregului șir de caractere care a fost introdus în câmpul de tip text. |

| Eveniment | Descriere  |
|-----------|--|
| onBlur    | Execută codul JavaScript în momentul în care se înlătură focalizarea de pe câmpul de tip text.   |
| onChange  | A acțiune ce va fi declanșată la schimbarea unui element dintr-un formular.  |
| onFocus   | Execută codul JavaScript în momentul în care are loc evenimentul <i>focus</i> , adică atunci când o fereastră, un frame sau un frameset primește un focus. |
| onSelect  | Execută codul JavaScript în momentul în care are loc evenimentul <i>select</i> .   |

**Exemplu** În acest exemplu, am utilizat obiectul *Text*, pentru a realiza o pagină cu un buton, în care utilizatorul scrie textul ce va fi afișat pe buton, va alege culorile pentru: fundalul și textul paginii și al butonului. Exemplul se poate extinde prin adăugarea mai multor culori.



```
<html>
<head>
<title>Obiectul Text</title>
<style type="text/css">
body
{
font-family: "Georgia", "Times New Roman";
```

```

        color: #003300;
        background-color: #C4FFC4;
    }
    .titlul
    {
        font-weight: bold;
        font-size: 20px;
        text-align: center;
    }
</style>
<script type="text/javascript" language="JavaScript">
function f()
{
    document.writeln("<html><head><title>buton</title></head><body    text="
formular.culoril.value    +"    bgcolor='"+formular.culori.value+"'>Ati
introduc:    <input    type='button'    name='buton'
value='"+formular.text.value+"'
style='color='"+formular.culori3.value+";background-
color='"+formular.culori2.value+"'></body></html>");
}
</script>
</head>
<body>
<form name="formular" method="post" action="">
<div class="titlul">Construiti un buton</div><br /><br />
<table border="0" cellspacing="1" cellpadding="1">
<tr>
<td>Alegeti culoarea pentru fundal:</td>
<td><select name="culori" >
<option value="azure" selected>Albastru azur</option>
<option value="red">Rosu</option>
<option value="yellow">Galben</option>
<option value="blue">Albastru</option>
<option value="green">Verde</option>
</select></td>
</tr>
<tr>
<td>Alegeti culoarea pentru text:</td>
<td><select name="culori1" >
<option value="azure" selected>Albastru azur</option>
<option value="red">Rosu</option>
<option value="yellow">Galben</option>
<option value="blue">Albastru</option>
<option value="green">Verde</option>
</select></td>
</tr>
<tr>
<td>Introduceti numele butonului:</td>
<td><input type="text" name="text" size="15"></td>
</tr>
<tr>
<td>Alegeti culoarea pentru fundalul butonului:</td>
<td><select name="culori2" >
<option value="azure" selected>Albastru azur</option>
<option value="red">Rosu</option>
<option value="yellow">Galben</option>
<option value="blue">Albastru</option>
<option value="green">Verde</option>
</select></td>

```

```

</tr>
<tr>
  <td>Alegeti culoarea pentru textul butonului:</td>
  <td><select name="culori3" >
    <option value="azure" selected>Albastru azur</option>
    <option value="red">Rosu</option>
    <option value="yellow">Galben</option>
    <option value="blue">Albastru</option>
    <option value="green">Verde</option>
  </select></td>
</tr>
<tr bgcolor="#6CFF6C">
  <td >si dati click: </td>
  <td><input type="submit" name="Submit" value="Click!"
onClick="f()" "></td>
</tr>
</table>
<br /></p>
</form>
</body>
</html>

```

## Textarea

Obiectul Textarea este un câmp de tip text ce poate avea mai multe linii, în care utilizatorul poate introduce date și este creat cu eticheta <TEXTAREA>, care aparține unui formular HTML.

Proprietăți: *defaultValue*, *form*, *name*, *type*, *value*.

Metode: *blur()*, *focus()*, *handleEvent()*, *select()*.

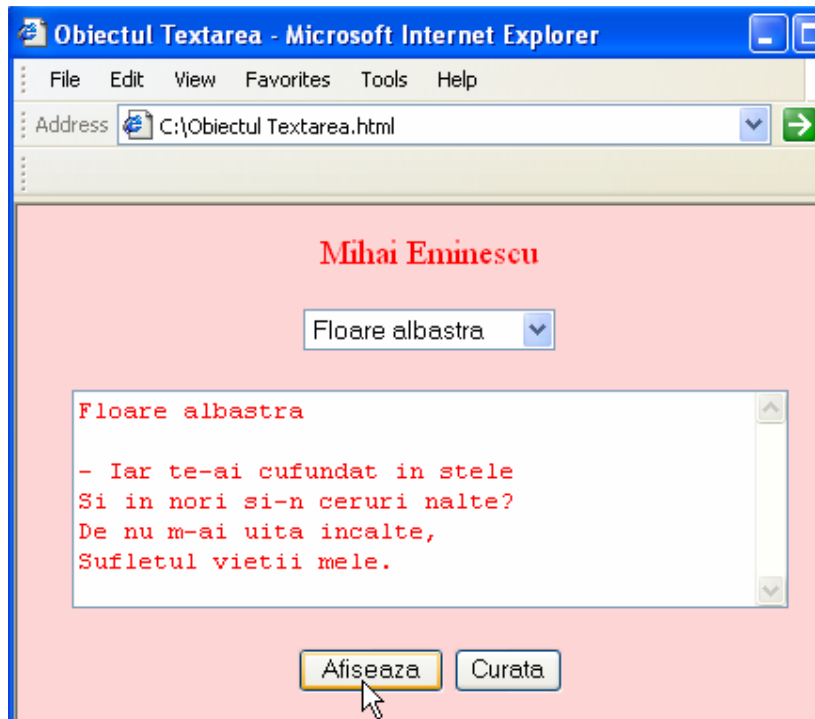
Evenimente: *onBlur*, *onChange*, *onFocus*, *onKeyDown*, *onKeyPress*, *onKeyUp*, *onSelect*.

| Proprietate  | Descriere   |
|--------------|---|
| defaultValue | Setează și returnează valoarea inițială a câmpului de tip textarea. |
| form         | Obiectul form, care include câmpul de tip textarea.                 |
| name         | Numele butonului.   |
| type         | Tipul obiectului, care în cazul nostru este: Textarea.              |
| value        | Valoarea obiectului Textarea.                                       |

| Metodă        | Descriere   |
|---------------|---|
| blur()        | Înlătură focalizarea de pe câmpul de tip textarea specificat.   |
| focus()       | Dă focalizare asupra câmpului de tip textarea specificat.   |
| handleEvent() | Invocă handlerul pentru un anumit eveniment specificat.   |
| select()      | Permite selectarea cu ajutorul cursorului a întregului șir de caractere care a fost introdus în câmpul de tip textarea. |

| Eveniment | Descriere  |
|-----------|--|
| onBlur    | Execută codul JavaScript în momentul în care se înlătură focalizarea de pe câmpul de tip textarea.   |
| onChange  | Acțiune ce va fi declanșată la schimbarea unui element dintr-un formular.  |
| onFocus   | Execută codul JavaScript în momentul în care are loc evenimentul <i>focus</i> , adică atunci când o fereastră, un frame sau un frameset primește un focus. |

|            |   |
|------------|---|
| onKeyDown  | Execută codul JavaScript în momentul în care are loc evenimentul KeyDown.                       |
| onKeyPress | Execută codul JavaScript în momentul în care are loc evenimentul KeyPress (se apasă o tastă).   |
| onKeyUp    | Execută codul JavaScript în momentul în care are loc evenimentul KeyUp (se eliberează o tastă). |
| onSelect   | Execută codul JavaScript în momentul în care are loc evenimentul <i>select</i> .                |



```

<html>
<head>
  <title>Obiectul Textarea</title>
<style type="text/css">
body
{
  font-size: 18px;
  color: #FF0000;
  background-color: #FFD5D5;
  text-align: center;
}
.vers
{
  color: #FF0000;
}
</style>
<script type="text/javascript" language="JavaScript">

```

```

function vector(n)
{
  var i;
  this.length=n;
  for(i=1; i<=n; i++)
  {
    this[i]=null;
  }
}

```



```

return this }

function strofe(Mpoezie)
{
    Mpoezie.pz.value="Poezii.";
    Mpoezie.optiune.selectedIndex="0";
}
var poezie=new vector(4);
var ax1="Floare albastra\r\n\r\n- Iar te-ai cufundat in stele\r\n\r\n";
var ax2="Si in nori si-n ceruri nalte?\r\n\r\n";
var ax3="De nu m-ai uita incalte,\r\n\r\n";
var ax4="Sufletul vietii mele." ;
var bx1="Singuratate\r\n\r\nCu perdelele lasate,\r\n\r\n";
var bx2="Sed la masa mea de brad,\r\n\r\n";
var bx3="Focul pilpiie in soba,\r\n\r\n";
var bx4="Iara eu pe ginduri cad." ;
var cx1="Sonete\r\n\r\nAfara-i toamna, frunza-mprastiata,\r\n\r\n";
var cx2="Iar vintul zvirle-n geamuri grele picuri;\r\n\r\n";
var cx3="Si tu citesti scrisori din roase plicuri\r\n\r\n";
var cx4="Si intr-un ceas gindesti la viata toata.";
poezie[0]="Selectati o poezie.";
poezie[1]=ax1+ax2+ax3+ax4;
poezie[2]=bx1+bx2+bx3+bx4;
poezie[3]=cx1+cx2+cx3+cx4;

function F(Mpoezie)
{
    if (document.Mpoezie.optiune.selectedIndex==0)
    {
        document.Mpoezie.pz.value="Selectati o poezie.";
        return
    }
}
var nr=document.Mpoezie.optiune.selectedIndex
document.Mpoezie.pz.value=poezie[nr]
}
</script>
</head>
<body>
Mihai Eminescu<br /><br />
<FORM name="Mpoezie" size="4">
    <SELECT name="optiune">
        <option selected>Selectati o poezie</option>
        <option>Floare albastra</option>
        <option>Singuratate</option>
        <option>Sonete</option>
    </SELECT>
    <br /><br />
<TEXTAREA name=pz rows=7 wrap=physical cols=45 class="vers">Poezii
</TEXTAREA><br /><br />
<INPUT onclick=F(this.form) type="button" value="Afiseaza" name="Afiseaza">
<INPUT onclick=strofe(this.form) type="button" value="Curata">
</FORM>
</body>
</html>

```

## Window

Obiectul Window poate fi fereastra unui browser sau frame, creat cu etichetele <body> sau <frameset> din HTML.

Proprietăți: *closed*, *crypto*, *defaultStatus*, *document*, *frames*, *history*, *innerHeight*, *innerWidth*, *length*, *location*, *locationbar*, *menubar*, *name*, *offscreenBuffering*, *opener*, *outerHeight*, *outerWidth*, *pageXOffset*, *pageYOffset*, *parent*, *personalbar*, *screenX*, *screenY*, *scrollbars*, *self*, *status*, *statusbar*, *toolbar*, *top*, *window*.

Metode: *alert()*, *atob()*, *back()*, *blur()*, *btoa()*, *captureEvents()*, *clearInterval()*, *clearTimeout()*, *close()*, *confirm()*, *crypto.random()*, *crypto.signText()*, *disableExternalCapture()*, *enableExternalCapture()*, *find()*, *focus()*, *forward()*, *handleEvent()*, *home()*, *moveBy()*, *moveTo()*, *open()*, *print()*, *prompt()*, *releaseEvents()*, *resizeBy()*, *resieTo()*, *routeEvent()*, *scroll()*, *scrollBy()*, *scrollTo()*, *setHotKeys()*, *setInterval()*, *setResizable()*, *setTimeout()*, *stop()*.

Evenimente: *onBlur*, *onDragDrop*, *onError*, *onFocus*, *onLoad*, *onMove*, *onResize*, *onUnload*.

| Proprietate                             | Descriere  |
|---|--|
| <i>closed</i>                           | Returnează o valoare de tip Bolean, care pentru valoarea true, determină dacă fereastra a fost închisă, iar pentru false dacă este deschisă.   |
| <i>crypto</i>                           | Returnează browser-ului obiectul <i>crypto</i> , ce poate fi utilizat pentru a accesa diverse servicii criptografice, care pot proteja utilizatorii împotriva site-urilor web răuvoitoare. |
| <i>defaultStatus</i>                    | Reprezintă mesajul care este afișat inițial în bara de jos a ferestrei.  |
| <i>document</i>                         | Reprezintă obiectul <i>document</i> conținut de fereastră.   |
| <i>frames</i>                           | Reprezintă un array ce conține referințe către toate <i>frame</i> -urile conținute de fereastră curentă.   |
| <i>history</i>                          | Conține detalii despre URL-urile vizitate din fereastră respectivă.  |
| <i>innerHeight</i> , <i>innerWidth</i>  | Reprezintă dimensiunile (înălțime și lățime) interioare ale unei suprafețe din fereastră.  |
| <i>length</i>                           | Reprezintă numărul de <i>frame</i> -uri conținute de către fereastră.  |
| <i>location</i>                         | Returnează detalii cu privire la URL-ul curent al ferestrei.   |
| <i>locationbar</i>                      | Returnează detalii cu privire la URL sau bookmark.   |
| <i>menubar</i>                          | Returnează detalii cu privire la meniuri (File, Edit, View).   |
| <i>name</i>                             | Returnează numele ferestrei.   |
| <i>offscreenBuffering</i>               | Returnează o valoare de tip boolean, care specifică momentul când fereastra este închisă, înainte de a fi vizibilă utilizatorului.   |
| <i>opener</i>                           | Returnează detalii cu privire la fereastră sursă, când este utilizată în fereastră destinație.   |
| <i>outerHeight</i> , <i>outerWidth</i>  | Determină dimensiunile în pixeli, ale marginilor exterioare, incluzând toate elementele de interfață ale ferestrei.  |
| <i>pageXOffset</i> , <i>pageYOffset</i> | Returnează pozițiile X și Y ale paginii curente relativ la colțul din stânga al suprafeței afișate ale ferestrei.  |
| <i>parent</i>                           | Reprezintă o referință către fereastră sau <i>frame</i> -ul care apelează alte <i>frame</i> -uri.  |
| <i>personalbar</i>                      | Reprezintă personal bar al browser-ului.   |
| <i>screenX</i> , <i>screenY</i>         | Specifică coordonatele ferestrei: x și y relativ la ecranul monitorului utilizatorului.  |
| <i>scrollbars</i>                       | Reprezintă scrollbar orizontal și vertical al browser-ului.  |
| <i>self</i>                             | Reprezintă o referință pentru fereastră curentă activă sau pentru <i>frame</i> .   |

|           |   |
|-----------|---|
| status    | Definește mesajul care poate fi afișat în status bar al browser-ului.   |
| statusbar | Reprezintă status bar al browser-ului.  |
| toolbar   | Returnează o valoare de tip Boolean care determină când bara de instrumente a browser-ului este vizibilă și când nu este. |
| top       | Reprezintă o referință pentru fereastra browser-ului.   |
| window    | Reprezintă o referință pentru fereastra curentă sau pentru frame.   |

| Metodă  | Descriere   |
|---|---|
| alert()   | Afișează o fereastră de dialog cu un mesaj și cu un buton OK.   |
| atob()  | Decodifică un șir de caractere ce a fost codificat utilizând baze-64 (cu metoda btoa()).                                  |
| back()  | Încarcă documentul (URL-ul) anterior din lista History.   |
| blur()  | Înlătură focalizarea de pe fereastra curentă.   |
| btoa()  | Creează un cod ASCII base-64 dintr-un șir de caractere.   |
| captureEvents()                                       | Capturează toate evenimentele de un anumit tip.   |
| clearInterval()                                       | Anulează timpul setat cu metoda setInterval.  |
| clearTimeout()  | Anulează timpul setat cu metoda setTimeout.   |
| close()   | Închide fereastra specificată.  |
| confirm()   | Afișează o fereastră de dialog Confirm ce conține butoanele: OK și Cancel.  |
| crypto.random(),                                      | Returnează un șir de caractere aproximativ aleator, a cărui lungime are un anumit număr de byte specificați.              |
| crypto.signText(),                                    | Returnează un șir de caractere cu date codificate.  |
| disableExternalCapture(),<br>enableExternalCapture(), | Dezactivează sau activează capturarea evenimentelor externe.  |
| find()  | Descoperă un șir de caractere în conținutul unei ferestre specificate.  |
| focus()   | Dă focalizare asupra ferestrei specificate.   |
| forward()   | Puntează browser-ul către URL-ul următor din lista History, similar cu butonul Forward al browser-ului.                   |
| handleEvent()   | Invocă handlerul pentru un anumit eveniment specificat.   |
| home()  | Puntează browser-ul către URL-ul setat de către utilizator ca <i>home page</i> , similar cu butonul Home al browser-ului. |
| moveBy()  | Mută fereastra cu un anumit număr de pixeli.  |
| moveTo()  | Mută colțul din stânga-sus al ferestrei la numite coordonate specificate ale ecranului.                                   |
| open()  | Deschide o nouă fereastră a browser-ului web.   |
| print()   | Printează conținutul ferestrei.   |
| prompt()  | Afișează o fereastră de dialog Prompt ce conține un mesaj și un câmp în care utilizatorii pot introduce date.             |
| releaseEvents()                                       | Setează fereastra sau documentul pentru a elibera evenimentele capturate, de un anumit tip.                               |
| resizeBy()  | Redimensionează fereastra mutând colțul din dreapta la o anumită poziție specificată, în pixeli.                          |
| resieTo()   | Redimensionează fereastra la o anumită dimensiune specificată, în pixeli.   |
| routeEvent()  | Trimite un eveniment capturat de-a lungul ierarhiei obișnuite a evenimentelor.  |
| scroll()  | Permite utilizarea scroll într-o fereastră, la anumite coordonate specificate.  |

|                |   |
|----------------|---|
| scrollBy()     | Permite utilizarea scroll într-o anumită suprafață specificată dintr-o fereastră.             |
| scrollTo()     | Permite utilizarea scroll într-o fereastră, între anumite coordonate specificate.             |
| setHotKeys()   | Activează sau dezactivează anumite taste dintr-o fereastră care nu are meniuri.               |
| setInterval()  | Evaluează o expresie sau apelează o funcție la anumite intervale specificate, în milisecunde. |
| setResizable() | Specifică momentul în care unui utilizator i se permite să redimensioneze o fereastră.        |
| setTimeout()   | Evaluează o expresie sau apelează o funcție după un anumit număr de milisecunde specificat.   |
| stop()         | Permite oprirea acțiunii de download curente, similar cu butonul Stop al browser-ului.        |

| Eveniment  | Descriere   |
|------------|---|
| onBlur     | Execută codul JavaScript în momentul în care se înlătură focalizarea de pe fereastra specificată.   |
| onDragDrop | Execută codul JavaScript în momentul în care are loc evenimentul DragDrop.  |
| onError    | Execută codul JavaScript în momentul în care are loc evenimentul Error, adică în urma eșecului încărcării unei ferestre.                        |
| onFocus    | Execută codul JavaScript în momentul în care are loc evenimentul <i>focus</i> , adică atunci când o fereastră primește un focus.                |
| onLoad     | Execută codul JavaScript în momentul în care are loc evenimentul <i>Load</i> , adică la încărcarea unei ferestre.                               |
| onMove     | Execută codul JavaScript în momentul în care are loc evenimentul <i>Move</i> , adică la mișcarea cursorului pe suprafața ferestrei specificate. |
| onResize   | Execută codul JavaScript în momentul în care are loc evenimentul <i>Resize</i> , adică la redimensionarea ferestrei specificate.                |
| onUnload   | Execută codul JavaScript în momentul în care are loc evenimentul <i>Unload</i> , adică atunci când o fereastră nu se încarcă.                   |

**Exemplu** În acest exemplu, am utilizat obiectul Window, pentru a afișa o fereastră cu dimensiunile alese de către utilizator.

Stabiliti dimensiunile si pozitia unei ferestre

Distanța de la colțul din stanga:  pixeli

Distanța de la colțul de sus:  pixeli

Latimea ferestrei:  pixeli

Înălțimea ferestrei:  pixeli

<html>

```

<head>
  <title>Textarea</title>
<style type="text/css">
body
{
  color: #990000;
  background-color: #FFD2D2;
}
.still
{
  background-color: #FF9D9D;
  text-align: center;
  font-size: 19px;
  font-weight: bolder;
}
</style>
<script type="text/javascript" language="JavaScript">
function f(stanga, sus, latime, inaltime)
{
  var fereastr="location=no,scrollbars=yes,menubars=yes,toolbars=no,
resizable=yes"+",left="+ stanga + ",top=" + sus + ",width=" + latime + ",
height=" + inaltime;
  var URL = "http://www.manuscris.html";
  popup = window.open(URL,"Meniu",fereastr);
}
</script>
</head>
<body>
<FORM name="formular">
<table>
<tr>
<td colspan="2" class="still">
  Stabiliti dimensiunile si pozitia unei ferestre<br /><br /></td>
</tr>
<tr>
<td>Distanta de la coltul din stanga: </td>
<td><input type="text" maxLength=4 size=4 name="stanga"> pixeli</td>
</tr>
<tr>
<td>Distanta de la coltul de sus: </td>
<td><input type="text" maxLength=4 size=4 name="sus"> pixeli</td>
</tr>
<tr>
<td>Latimea ferestrei:</td>
<td><input type="text" maxLength=4 size=4 name="latime"> pixeli</td>
</tr>
<tr>
<td>Inaltimea ferestrei: </td>
<td><input type="text" maxLength=4 size=4 name="inaltime"> pixeli</td>
</tr>
<tr>
<td colspan="2" class="still"><INPUT onclick="f(this.form.stanga.value,
this.form.sus.value, this.form.latime.value, this.form.inaltime.value)"
type="button" value="Apasati!"></td>
</tr>
</table>
</FORM>
</body>
</html>

```

## Evaluare

5. Enumerați și descrieți trei dintre cele mai importante obiecte JavaScript.
6. Care expresie returnează numărul de elemente ale masivului anotimpuri, de mai jos?  
var anotimpuri = new Array()  
anotimpuri[0] = "Primavara"  
anotimpuri[1] = "Vara"  
anotimpuri[2] = "Toamna"  
anotimpuri[3] = "Iarna"
  - a) anotimpuri.length;
  - b) element.anotimpuri.array;
  - c) anotimpuri[3];
  - d) constructor.anotimpuri;
7. Care dintre următoarele metode returnează ziua dintr-o dată specificată, conform timpului local:
  - a) getDay();
  - b) getDate();
  - c) getTime();
  - d) getUTCDate;

# Bibliografie

1. **Diana Elena Diaconu, Cristian Morărescu**, JavaScript prin exemple, Editura Printech, 2005
2. **Marin Vlada**, Informatică: Windows, Word, Excel, Internet, Editura Universității din București, 2000
3. **Marin Vlada**, Birotică. Tehnologii multimedia, Editura Universității din București, 2004
4. **Emanuela Cerchez**, Internet, Editura Polirom, 2000
5. **Tudor Sorin, Vlad Huțanu**, Tehnologia informației și a comunicațiilor, Editura L&S Infomat, 2004
6. **Dan Somnea**, Inițiere în JavaScript și tehnologiile Netscape, Editura Tehnică, 1998
7. **Richard Wagner, R. Allen Wyke**, JavaScript, Editura Teora, 2001
8. **Tom Negrino, Dori Smith**, JavaScript pentru World Wide Web, Editura Corint, 2004
9. **Hakon Wium Lie, Bert Bos**, Cascading Style Sheets, United States of America, Editura Addison-Wesley, 1999
10. **Tom Negrino, Dori Smith**, JavaScript pentru World Wide Web, Editura Corint, 2004
11. **Jim Keogh**, JavaScript fără mistere. Ghid pentru autodidacți, Editura Rosetti Educațional, 2006
12. **Calin Marin Văduva**. Programarea în Java, Editura Albastră, 2002
13. **Teodoru Gugoiu**, HTML, XHTML, CSS și XML prin Exemple. Ghid practic, Editura Teora, 2005
14. **Sabin Burada**, Proiectarea siturilor Web. Design și funcționalitate, Editura Polirom, 2002
15. **Ștefan Tanasă, Cristian Olaru**, Dezvoltarea aplicațiilor Web folosind Java, Editura Polirom, 2005
16. **Mihaela Brut**, Instrumente pentru e-Learning. Ghidul informatic al profesorului modern, Editura Polirom, 2006

# Resurse Web

1. **Denumire organizatie/resursa** - <http://www.darpa.mil>
2. **ISOC, The Internet Society (ISOC)** - <http://www.isoc.org>
3. <http://www.ietf.org>
4. <http://w3c.org>
5. <http://docs.sun.com/app/docs>
6. <http://javascript.internet.com/>
7. <http://www.htmlgoodies.com/>
8. <http://www.w3schools.com>



# Glosar

**Download** – reprezintă procesul prin care un fișier este transferat la distanță de pe un computer pe computerul client.

**MIME** – Multipurpose Internet Mail Extensions

**Pagină Web** – un singur document Web. O pagină Web este tot ceea ce se afișează în fereastra unui browser la un moment dat, inclusiv ceea ce se vede dacă utilizăm scroll.

**URL** – Uniform Resource Locator este o adresă Web. Ea este compusă din patru părți: numele protocolului (Ex: HTTP, scriem http://), locația site-ului (Ex: WWW), numele organizației care ține site-ul (Ex: google), sufixul care identifică tipul organizației (Ex: .com). Pentru exemplul dat, introducem adresa: <http://www.google.com>

**Web Browser** – un program ce rulează pe computerul utilizatorului și permite vizualizarea paginilor Web.

**Web Server** – un computer ce stochează mai multe pagini Web pe care le trimite clienților care le cer. De exemplu, un Web Browser face o cerere către un Web Server pentru a afișa utilizatorului o pagină Web care este stocată în Web Server.

**Web Site** – un set de pagini Web legate între ele prin legături de tip hipertext sau hipermedia. De regulă, paginile Web dintr-un site Web au aceeași structură și aspect asemănător.

# Cuprins

|  |   |
|--|---|
| Introducere.....                         | 2 |
| <b>Capitolul 1</b>                       |   |
| Internet.....                            | 3 |
| 1.1. Scurt istoric .....                 | 3 |
| 1.2. Rețea de calculatoare .....         | 4 |
| 1.3. Ce este Internetul .....            | 5 |
| Evaluare.....                            | 6 |
| <b>Capitolul 2</b>                       |   |
| W.W.W.....                               | 7 |
| 2.1. World Wide Web .....                | 7 |
| 2.2. Cum funcționează paginile Web ..... | 8 |
| 2.3. Construirea site-urilor Web.....    | 9 |
| Evaluare.....                            | 9 |

### Capitolul 3

|  |    |
|--|----|
| HTML.....  | 10 |
| 3.1. Introducere.....                            | 10 |
| 3.2. Cum se realizează o pagină Web cu HTML..... | 11 |
| 3.3. Elementul Meta.....                         | 12 |
| 3.4. Culori.....                                 | 13 |
| 3.5. Elementul BODY.....                         | 13 |
| 3.6. Formatarea textelor.....                    | 14 |
| 3.7. Formatarea paragrafelor.....                | 16 |
| 3.8. Preformatarea textului.....                 | 16 |
| 3.9. Liste.....                                  | 17 |
| 3.10. Tabele.....                                | 19 |
| 3.11. Legături HTML.....                         | 21 |
| 3.12. Cadre HTML.....                            | 23 |
| 3.13. Formulare HTML.....                        | 25 |
| 3.13.1. Elementul form.....                      | 25 |
| 3.13.2. Elementul input.....                     | 26 |
| 3.13.3. Elementul textarea.....                  | 27 |
| 3.13.4. Elementul label.....                     | 27 |
| 3.13.5. Elementul fieldset.....                  | 27 |
| 3.13.6. Elementul legend.....                    | 28 |
| 3.13.7. Elementul select.....                    | 28 |
| 3.13.8. Elementul optgroup.....                  | 29 |
| 3.13.9. Elementul option.....                    | 29 |
| 3.13.10. Elementul button.....                   | 29 |
| 3.14. Imagini HTML.....                          | 30 |
| 3.14.1. Elementul img.....                       | 30 |
| 3.14.2. Elementul map.....                       | 31 |
| 3.14.3. Elementul area.....                      | 31 |
| 3.15. Fundal HTML.....                           | 32 |
| 3.16. Fonturi HTML.....                          | 34 |
| 3.17. Trucuri cu HTML.....                       | 35 |
| Evaluare.....                                    | 35 |

### Capitolul 4

#### CSS

|  |    |
|--|----|
| 4.1. Cascading Style Sheet.....        | 36 |
| 4.1.1. Elementul style.....            | 37 |
| 4.1.2. Elementul link.....             | 40 |
| 4.1.3. Elementul div.....              | 41 |
| 4.1.4. Elementul span.....             | 41 |
| 4.1.5. Elementul font.....             | 42 |
| 4.1.6. Elementul basefont.....         | 42 |
| 4.1.7. Elementul center.....           | 43 |
| 4.2. Attribute CSS.....                | 44 |
| 4.3. Aplicații interesante cu CSS..... | 46 |
| Evaluare.....                          | 49 |

### Capitolul 5

#### JavaScript

|                       |    |
|-----------------------|----|
| 5.1. Introducere..... | 50 |
|-----------------------|----|

|   |     |
|---|-----|
| 5.2. Unde scriem script-urile JavaScript.....   | 51  |
| 5.3. Operatori JavaScript   | 52  |
| 5.3.1. Operatori aritmetici.....  | 52  |
| 5.3.2. Operatori de incrementare / decrementare.....  | 52  |
| 5.3.3. Operatori de comparație.....   | 53  |
| 5.3.4. Operatori logici (booleeni).....   | 54  |
| 5.3.5. Operatori logici pe biți (bitwise).....  | 56  |
| 5.3.6. Operator pentru șiruri de caractere (string).....  | 58  |
| 5.3.7. Operatori de atribuire.....  | 59  |
| 5.3.8. Operatorul condițional.....  | 60  |
| 5.3.9. Operatorul typeof.....   | 60  |
| Evaluare.....   | 61  |
| 5.4. Instrucțiuni.....  | 61  |
| 5.4.1. Instrucțiuni primitive.....  | 61  |
| 5.4.2. Instrucțiuni de decizie.....   | 63  |
| 5.4.3. Instrucțiuni repetitive.....   | 67  |
| Evaluare.....   | 73  |
| <br>  |     |
| <b>Capitolul 6</b>  |     |
| Funcții JavaScript  | 75  |
| 6.1. Introducere.....   | 75  |
| 6.2. Definirea funcțiilor.....  | 75  |
| 6.3. Apelarea funcțiilor.....   | 76  |
| 6.4. Parametrii funcțiilor.....   | 78  |
| 6.5. Funcții predefinite.....   | 80  |
| 6.6. Funcții și stiluri.....  | 87  |
| Evaluare.....   | 93  |
| <br>  |     |
| <b>Capitolul 7</b>  |     |
| Evenimente JavaScript   | 94  |
| 7.1. Introducere.....   | 94  |
| 7.2. Evenimente și tratarea evenimentelor.....  | 94  |
| 7.3. Tratarea evenimentelor – clasificare.....  | 94  |
| 7.3.1. Tratarea evenimentelor de către sistem (onError, onLoad, onUnload)   | 95  |
| 7.3.2. Tratarea evenimentelor de către un formular (onBlur, onChange, onFocus, onReset, onSelect, onSubmit) .....                   | 98  |
| 7.3.3. Tratarea evenimentelor de către tastatură (onKeyDown, onKeyPress, onKeyUp).....  | 111 |
| 7.3.4. Tratarea evenimentelor de către mouse (onClick, onDbClick, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp).    | 115 |
| 7.4. Aplicații cu evenimente (sortare tabel, formular numai cu cifre, contorizare caractere, formular de calcul instantaneu ) ..... | 132 |
| Evaluare.....   |     |
| <br>  |     |
| <b>Capitolul 8</b>  |     |
| Obiecte JavaScript  |     |
| 8.1. Introducere.....   | 133 |
| Anchor.....   | 133 |
| Applet.....   | 138 |
| Area.....   | 138 |
| Array.....  | 140 |
| Boolean.....  | 143 |
| Button.....   | 144 |

|                   |     |
|-------------------|-----|
| Checkbox.....     | 145 |
| Date.....         | 148 |
| Document.....     | 153 |
| Event.....        | 155 |
| FileUpload.....   | 162 |
| Form.....         | 164 |
| Frame.....        | 168 |
| Function.....     | 170 |
| Hidden.....       | 171 |
| History.....      | 172 |
| Image.....        | 174 |
| Java.....         | 176 |
| JSONArray.....    | 176 |
| JavaClass.....    | 176 |
| JavaObject.....   | 176 |
| JavaPackage.....  | 176 |
| Layer.....        | 176 |
| Link.....         | 183 |
| Location.....     | 186 |
| Math.....         | 190 |
| MimeType.....     | 192 |
| Navigator.....    | 193 |
| Netscape.....     | 194 |
| Number.....       | 194 |
| Object.....       | 195 |
| Option.....       | 197 |
| Packages.....     | 199 |
| Password.....     | 199 |
| Plugin.....       | 201 |
| Radio.....        | 201 |
| RegExp.....       | 207 |
| Reset.....        | 209 |
| Screen.....       | 211 |
| Select.....       | 213 |
| String.....       | 216 |
| Style.....        | 219 |
| Submit.....       | 222 |
| Sun.....          | 225 |
| Text.....         | 225 |
| Textarea.....     | 228 |
| Window.....       | 231 |
| Evaluate.....     | 235 |
| <br>              |     |
| Bibliografie..... | 236 |
| Resurse web.....  | 237 |
| Glosar.....       | 238 |
| Cuprins.....      | 239 |