Obiecte în Java Script

Java Script este un limbaj bazat pe obiecte şi pune la dispoziţie un set predefinit de obiecte. Fiecare obiect este identificat prin nume.
O **proprietate** a obiectului este apelată prin **numeobiect.proprietate (**document.bgColor = ‘‘red”). O **metodă** este adresată prin **numeobiect.metodă(argument).** Obiectele JavaScript sunt ierarhizate. Un obiect poate fi derivat, inclus în cadrul unui alt obiect. Ierarhia de obiecte JS are 4 ramuri principale, reprezenţate de obiectele: **window**, **navigator**, **screen**, **language.**
Pe lângă acestea, JS pune la dispoziţie obiecte predefinite referitoare în general la tipuri de date: **String**, **Math**, **Date**.

Obiectul String
Încapsulează un şir de caractere. Este creat automat atunci când unei variabile i se asociază o valoare de tip şir de caractere.
Dintre metodele lui **String** întâlnim:

Metoda	Efect
anchor()	returneaza un sir ca şi "anchor"
big()	returneaza un sir cu text mare
blink()	returneaza un sir care clipeste
bold()	returneaza un sir cu litere ingrosate
charAt()	returneaza un caracter de la pozitia care este specificata
charCodeAt()	returneaza codul ascii al unui caracter de la o pozitie specificata
concat()	returneaza doua siruri concatenate
fixed()	returneaza un sir cu caractere tip
fontcolor()	returneaza un sir cu o culoare specificata
fontsize()	returneaza un sir cu litere de o anume marime
fromCharCode()	returneaza valoare unicode a unui caracater
indexOf()	returneaza pozitia primei aparitii a unui subsir intr-un sir
italics()	returneaza un sir in italic (scris aplecat)
lastIndexOf()	returneaza pozitia primei aparitii a unui subsir in un sir
link()	returneaza un sir ca hyperlink
match()	similar cu indexof şi lastindexof, dar aceasta metodă returneaza sirul specificat sir, sau "null", in locul unor valori numerice
replace()	inlocuieste unele caractere specificate cu altele noi specificate.
search()	returneaza un numar intreg daca sirul contine caracterele specificate
slice()	returneaza un sir incepand de la pozitia index specificata

Metoda	Efect
small()	returneaza un sir cu caractere mai mici
split()	imparte un sir in mai multe siruri, in functie de caracterele specificate
strike()	returneaza un sir taiat cu o linie la mijloc
sub()	returneaza un sir ca indice
substr()	returneaza un subsir specificat
toLowerCase()	converteste un sir in litere mici
toUpperCase()	converteste un sir in litere mari

Proprietatea acestui obiect **string este length** care returneaza numarul de caractere dintr-un sir.

Obiectul Date

Obiectul **Date** este folosit pentru a obţine informaţii referitoare la ceasul sistem de pe calculatorul vizitatorului paginii *Web*. Prin intermediul acestui obiect pot fi determinate data şi ora curentă, pot fi efectuate diferite operaţii cu date calendaristice sau momente ale zilei sau poate fi controlat modul în care este afişată pagina *Web* în funcţie de informaţiile furnizate de metodele acestui obiect.
Pentru a crea un obiect care să conţină data şi ora curentă secvenţa *JavaScript* corespunzătoare este nume_variabila_**data=new Date();**

Metodele obiectelor de tip sunt grupate în trei categorii: metode pentru preluare de informaţii, metode pentru setarea anumitor caracteristici şi metode pentru conversie.

Principalele metode ale obiectului Date() sunt:

Date()	returneaza un obiect Date
getDate()	returneaza data (ziua) din luna (între 131)
getDay()	returneaza ziua dintr un obiect Date (între 0 şi 6; 0=Duminica, 1=Luni, etc.)
getMonth()	returneaza luna dintr un obiect Date
getFullYear()	returneaza anul cu 4 cifre
getYear()	returneaza anul dintr un obiect Date
getHours()	returneaza ora
getMinutes()	returneaza minutele
getSeconds()	returneaza secunda
setTimeout(”funcţie”,timp)	timp se exprimă în ms

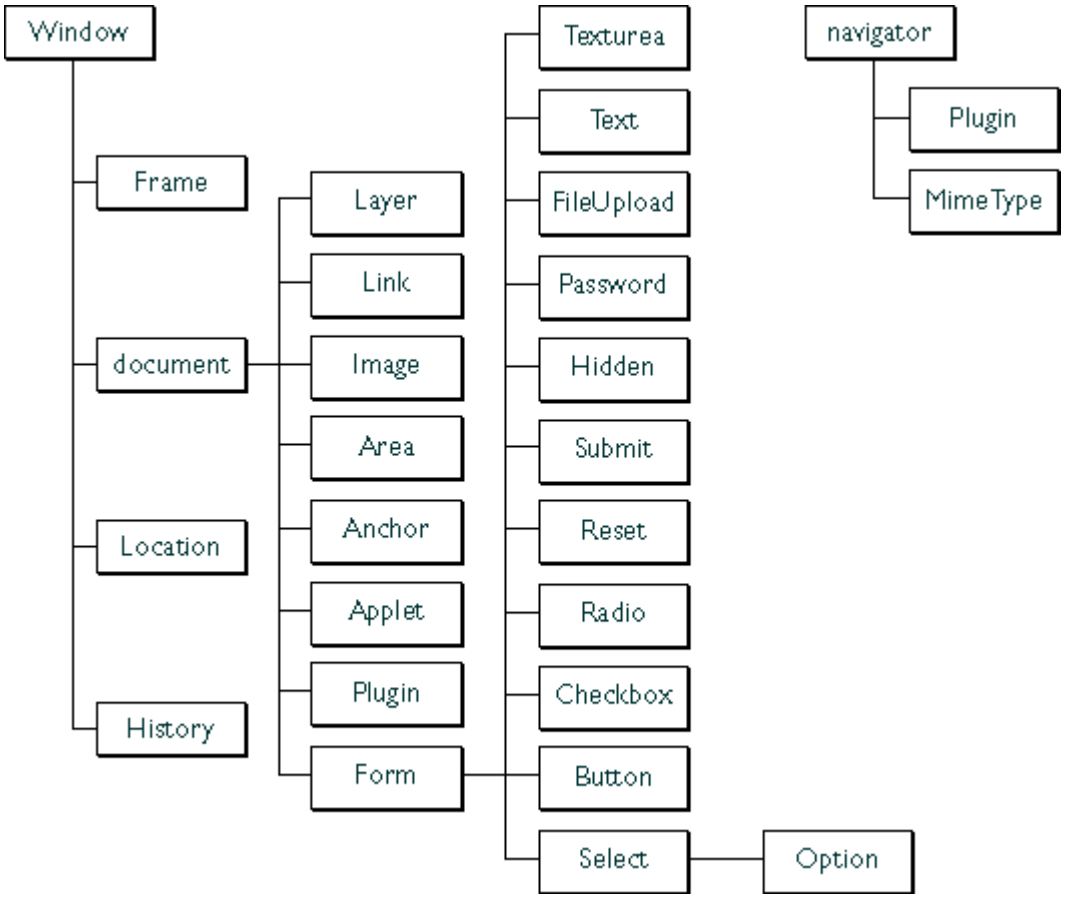
Script care să afişeze ‘‘Aţi intrat pe pagina astăzi: … ora …
<script type=‘‘text/javascript”>
data=new Date();
zi=data.getDate();
luna=data.getMonth()+1;
an=data.getYear()
ora=data.getHours()
min=data.getMinutes()
sec=data.getSeconds()

```
document.write("Ati intrat pe pagina astazi: ''+zi+"-"+luna+"-"+an+"
ora"+ora+":"+min+":"+sec)
</script>
```

Script care afișează un ceas în timp real în linia de stare:

```
<script type=''text/javascript">
function timp()
{
ceas=new Date()
ora=ceas.getHours()
min=ceas.getMinutes()
sec=ceas.getSeconds()
window.status=ora+":"+min+":"+sec
setTimeout("timp()",100)
}
setTimeout("timp()",100)
</script>
```

Ierarhia Window Object



In aceasta ierarhie, descendentii unui obiect sunt de fapt proprietati pentru acel obiect. In general, daca un obiect obj1 are un fiu obj2 care la randul lui este parinte pentru obj3, atunci exista constructia:

obj1.obj2.obj3.prop

Fiecare pagina are urmatoarele obiecte:

- *navigator* : proprietati despre numele si versiunea navigatorului, tipuri MIME, plug-inuri instalate pe client
- *window* : proprietăţi aplicabile întregii ferestre
- *document* : proprietăţi ale documentului (titlu, background, links, forms)
- *location* : proprietăţi bazate pe URI-ul curent
- *history* : proprietati reprezentand URI-urile pe care le-a vizitat clientul

Multe dintre obiectele Navigator au proprietati care sunt vectori. Iterarea in acesti vectori se poate face fie folosind indici numerici ordinali sau folosind numele obiectelor din vector specificate in interiorul operatorului de indexare, “[]”.

Obiectul Document

Obiectul *Document* contine informatii despre documentul curent si ofera metode pentru afisarea codului html in fereastra browserului. Acest obiect este creat de catre navigator cand se incarca o pagina si este asociat tag-ului <BODY>. Evenimentele pe care obiectul le poate receptiona sunt: onClick, onDblClick, onKeyDown, onKeyPress, onKeyUp, onMouseDown, onMouseUp. Principalele proprietati sunt urmatoarele:

- *alinkColor* : specifica culoarea unui link activ (valoarea atributului ALINK)
- *anchors* : un vector care contine cate o intrare pentru fiecare link (tag <A>); elementele sunt de tipul *anchor*
- *applets* : vector ce contine cate o intrare pentru fiecare applet din document; elementele sunt de tipul *applet*
- *bgColor* : string ce specifica culoarea de backgroud (atributul BGCOLOR)
- *cookie* :string ce reprezinta toate cookie-ele asociate cu documentul curent
- *embeds* : un vector ce contine toate obiectele (plug-in-uri) incluse in documentul curent (cu tag-ul EMBED)
- *fgColor* : string ce specifica culoarea de foreground
- *formName* : obiectul document contine cate o intrare de tipul Form pentru fiecare formular din document, iar numele proprietatii este chiar numele formularului (valoarea atributului NAME din tagul <FRAME>)
- *forms* : un vector ce contine elemente de tipul *form* pentru fiecare formular din document
- *height* : inaltimea documentului in pixeli
- *width* : latimea documentului in pixeli
- *ids* : creeaza un obiect *style* care poate fi folosit pentru tag-uril html
- *images* : un vector ce contine cate o intrare pentru fiecare imagine (<IMG>) din document; elementele sunt de tipul *image*
- *layers* : un vector cu elemente de tip *layer* si cu cate o intrare pentru fiecare layer definit in document
- *linkColor* : string ce specifica culoarea link-urilor
- *links* : un vector ce contine cate o intrare (de tip *link*) pentru fiecare link din document
- *plugins* : un vector cu toate plugin-urile din codul html; sunt de tip *plugin*
- *referrer* : contine URL-ul de la care s-a ajuns pe acesta pagina
- *title* : titlul documentului
- *URL* : URL-ul documentului
- *vlinkColor* : culoarea link-urilor vizitate

Toate campurile vector de mai sus pot sa le indexez fie cu indecsi intregi plecand de la 0, fie cu indecsi de tip nume (care apar ca valori ale atributului NUME din html). Astfel *document.frames[1]* este echivalent cu *document.frames["frm1"]*, daca frm1 este frame-ul cu indice 1 din frameset.
Iar metodele principale sunt:

- *close* : inchide un stream de iesire deschis cu metoda open si afisaza documentul html utilizatorului
- *open* : deschide un stream pentru scriere cu metodele *write* si *writeln*
- *write*, *writeln* : scrie una sau mai multe expresii separate prin virgula in documentul din fereastra specificata

exemplu 1:

```
<html>
<head>
<title>Javascript</title>
</head>
<body>
<p>
<b>verde</b><br />
[<b onmouseover="document.bgColor='green'">Green</b>]<br /><br /><br />
[<b onmouseover="document.bgColor='blue'">Blue</b>]<br /><br /><br />
[<b onmouseover="document.bgColor='yellow'">Yellow</b>]<br /><br /><br />
[<b onmouseover="document.bgColor='red'">Red</b>]<br /><br /><br />
</body>
</html>
```

Exemplu 2:

```
<html>
        <head>
        <title>Javascript</title>
        </head>
        <body>
        <center><h1>Schimbare culoare </h1>
        <form name="f">
        scrie culoare text:<input type="text" name="nume_culoare" size=20>
        culoare fundal<select name=lista_culori
onChange="document.bgColor=lista_culori.value">
        <option value=green>green</option>
        <option value=antiquewhite>antiquewhite</option>
        <option value=#ffff00>galben</option>
        </select>
        <input type="button" value="Schimba culoarea"
onclick="document.fgColor=nume_culoare.value">
        </form></center>
        <table border=2 cellpadding=5 cellspacing=3 width="90%">
        <tr><td>
        <font>
        mesaj 1<br>
        mesaj 2<br></font>
        </td>
        </tr></table>
        </body>
        </html>
```

Obiectul Window

Este plasat la cel mai înalt nivel, fiind părintele tuturor obiectelor dintr-o pagină.
Metode fundamentale:
open(), close()
alert(), prompt(), confirm(),moveTo(x,y),moveBy(x,y)
blur(), focus()
setTimeout()
Obiectul window este important pentru gestionarea ferestrelor în care sunt încărcate documentele html.
Folosind metodele **open**() şi **close**() putem deschide ferestre, putem încărca anumite documente în ferestrele respective şi le putem stabili anumite proprietăţi.
Deschiderea unei ferestre
window.open ("URL", "_Parrent","opţiuni")
- **URL** indică adresa documentului care va fi încărcat în fereastra respectivă
- **nume** poate fi folosit pentru a referi fereastra respectivă cu ajutorul atributului target
- **opţiuni** reprezintă o listă de elemente pentru stabilirea aspectului ferestrei

În cadrul scriptului fereastra deschisă poate fi identificată prin variabilă:
f=window.open ()
Are sens:

Metodă open permite specificarea unor parametri legaţi de aspectul ferestrei deschise. Aceştia se introduc separaţi prin virgulă în cadrul secţiunii *opţiuni* din construcţia metodei **open**:

- **width** - lăţimea în pixeli a suprafeţei ferestrei
- **height** - înălţimea în pixeli a suprafeţei ferestrei
- **toolbar** = yes/no - afişează sau nu bara de instrumente
- **menubar** = yes/no - afişează sau nu bara de meniuri
- **scrollbars** = yes/no/auto - afişează sau nu bara de scroll
- **left** = nr de pixeli faţă de marginea din stânga a ecranului
- **top** = nr de pixeli faţă de marginea sus a ecranului

Închiderea ferestrelor se poate realiza prin metodă **close()**
Poate fi apelată prin:
window.close() - pentru fereasta curentă
f.close() - pentru un obiect de tip fereastra
Metodă **moveTo(i,j)** mută o fereastră cu i pixeli spre dreapta şi j pixeli în jos.
Obiectul document
Este derivat din obiectul window şi este folosit ca metodă de acces la toate elementele html.
Proprietăţile obiectului document setează în primul rând proprietăţile şi atributele html incluse în marcajul HEAD sau în marcajul BODY. Principalele proprietăţi sunt:

- **document.title** = text în bara de titlu
- **document.bgColor** = culoarea fundalului
- **document.fgColor** = culoarea textului
- **document.alinkColor** = culoarea legăturilor active
- **document.vlinkColor** = culoarea legăturilor vizitate
- **document.linkColor** = culoarea legăturilor nevizitate
- **document.cookie** = fisier cookie asociat cu documentul

Metodele obiectului document permit generarea paginilor html în mod dinamic. Principalele metode sunt:

- ***close()*** *- inchide fluxul datelor de iesire spre document*
- ***contextual()*** *- permite să aplicam in mod selectiv un stil unui element HTML care apare într-un anumit context specific*
- ***getSelection()*** *- intoarce textul selectat*
- ***handleEvent()*** *- apeleaza handlerul pentru evenimentul specificat*
- ***open()*** *- deschide fluxul datelor de iesire spre document*
- ***write()*** *- adauga text in document*
- ***writeln()*** *- adauga text şi un caracter linie noua in document (textul pe linia lui)*

Gestionarea cadrelor

Atunci când introducem cadre este creat automat un obiect **parent,** care include un şir de obiecte, fiecare înglobând un cadru din interiorul ferestrei:
parent.frames[0] – referă primul cadru
parent.frames[1] – referă al doilea cadru s.a.m.d.
De exemplu, **parent.frames[0].document.write()** - scrie în primul cadru.
Dacă atunci când au fost introduse cadrele acestea au primit şi un nume putem referi un cadru prin construcţia: **parent.numecadru**
De exemplu, **parent.cadru.document.write()** - scrie în cadrul cu numele ‘‘cadru”.

Obiectul location

Încapsulează adresa URL a paginii curente. Permite deplasarea la o altă adresă URL sau permite extragerea unor elemente din cadrul URL curent. Cea mai folosită proprietate este **href** - specifică întreaga adresă URL De exemplu, încărcarea unei alte pagini web se poate face prin construcţia: **window.location.href = ‘‘URL”**
Exemplu de buton cu funcţie de legătură:
<input type=button value=‘‘Student”
onclick=‘‘window.location.href=’http://www.microsoft.com’”>
Încărcarea unui fişier într-un cadru se poate face prin **parent.cadru.location.href=‘‘URL”**

Construirea unui banner

La încărcarea unui pagini se deschide o fereastră nouă în care, din 2 în 2 secunde se succed 3 imagini.

<script type=‘‘text/javascript”>

```
imagini=new Array();
imagini[0]=‘‘x.gif”;
imagini[1]=‘‘y.gif”;
imagini[2]=‘‘z.gif”;
function deschide() {
f=window.open(”‘‘,””,”width=200, height=100, scrollbars=no, menubar=no,
toolbar=no”);
 }
i=-1;
function banner()
{
i++;
f.location.href=imagini[i]
if(i==2)
i=-1;
setTimeout(”banner()”,2000)
}
</script>
<body onload=‘‘deschide();banner()”>
```

Obiectul Image

Cuprinde proprietăţile şi metodele necesare pentru lucrul cu imagini. Se creează cu **new Image()**

poza=new Image()

Proprietăţi: **poza.src**=sursa imaginii; **poza.width**=lăţimea imaginii în pixeli; **poza.height**=înălţimea imaginii în pixeli; **poza.border**=grosimea chenarului în pixeli; **poza.name**=numele imaginii;

La introducerea imaginilor într-o pagină web se creează în mod automat un şir de obiecte imagine. Fiecărui obiect de tip imagine i se pot aplica toate proprietăţile imaginilor. Este identificat prin **document.images[i],** unde i este numărul de ordine al imaginii din cadrul documentului.

Realizarea unui efect RollOver

Un efect RollOver înseamnă schimbarea fişierului sursă al imaginii de fiecare dată când se plasează mouse-ul deasupra imaginii. Se revine la imaginea iniţială atunci când mouse-ul este scos de pe imagine.

Exemplu

```
<html>
<head>
<title>vector images</title>
</head>
<body>
<table>
<tr>
<td>
<img src="imagini/p1.jpg" width="200" height="200"

onmouseover=document.images[0].border=8
onmouseout=document.images[0].border=0
onclick=document.images[0].src="imagini/p2.jpg" />
```

```
<br />
<b onmouseover="document.bgColor='yellow'">mesaj</b>
</td>
<td>
<img src="imagini/p3.jpg" width="200" height="200"

onmouseover=document.images[1].border=20
onmouseout=document.images[1].border=0
onclick=document.images[1].src="imagini/p4.jpg" />
<br />
<b onmouseover="document.fgColor='blue'">mesaj</b>
</td>
</tr>
</table>
</body>
</html>
```

Obiectul **Date** - pentru a lucra cu data zilei si timpul.

Pentru a crea o instanta a obiectului "Date" se foloseste operatorul new astel: :

var data = new Date();

➔ **se memoreaza data curenta in variabila "data".**
Dupa ce a fost creata instanta, se pot folosi metodele obiectului astfel: data.getDate()

- Metodele obiectului Date sunt:

Date()	Returneaza un obiect Date
getDate()	Returneaza data (ziua) din luna (intre 1-31)
getDay()	Returneaza ziua dintr-un obiect Date (intre 0-6; 0=Duminica, 1=Luni, etc.)
getMonth()	Returneaza luna dintr-un obiect Date (intre 0-11. 0=January, 1=February, etc.)
getFullYear()	Returneaza anul dintr-un obiect Date (patru cifre)
getHours()	Returneaza ora dintr-un obiect Date (intre 0-23)
getMinutes())	Returneaza minutele dintr-un obiect Date (intre 0-59
getSeconds()	Returneaza secunda dintr-un obiect Date (intre 0-59)
getMilliseconds()	Returneaza milisecunda dintr-un obiect Date (intre 0-999)
getTime()	Returneaza numarul de milisecunde pana la miezul noptii
getTimezoneOffset()	Returneaza diferenta de timp intre computer si GMT
getUTCDate()	Returneaza data dintr-un obiect Date in (UTC) timp universal
getUTCDay()	Returneaza ziua dintr-un obiect Date in timp universal
getUTCMonth()	Returneaza luna dintr-un obiect Date in timp universal
getUTCFullYear()	Returneaza anul (4 cifre) dintr-un obiect Date in timp universal
getUTCHours()	Returneaza ora dintr-un obiect Date in timp universal
getUTCMinutes()	Returneaza minutele dintr-un obiect Date in timp universal
getUTCSeconds()	Returneaza secundele dintr-un obiect Date in timp universal
getUTCMilliseconds	Returneaza millisecundele dintr-un obiect Date in timp universal
parse()	Returneaza un sir ce are ca valoare numarul de millisecunde pana in January 01 1970 00:00:00
setDate	Seteaza luna in un Obiect Date (intre 1-31)
setFullYear()	Seteaza anul in un Obiect Date (four digits)
setHours()	Seteaza ora in un Obiect Date (intre 0-23)
setMilliseconds())	Seteaza millisecundele in un Obiect Date (intre 0-999
setMinutes()	Seteaza minutele in un Obiect Date (intre 0-59)
setMonth()	Seteaza luna in un Obiect Date (intre 0-11. 0=January, 1=February)
setSeconds()	Seteaza secunda in un Obiect Date (intre 0-59)
setTime()	Seteaza millisecundele dupa 1/1-1970
setUTCDate()	Seteaza data in un Obiect Date, in timp universal (intre 1-31)
setUTCMonth()	Seteaza luna in un Obiect Date, in timp universal (intre 0-11. 0=January, 1=February)
setUTCFullYear() -	Seteaza anul in un Obiect Date, in timp universal (four digits)
setUTCHour() -	Seteaza ora in un Obiect Date, in timp universal (intre 0-23)
setUTCMinutes() -	Seteaza minutele in un Obiect Date, in timp universal (intre 0-59)

setUTCSeconds() -	Seteaza secundele in un Obiect Date, in timp universal (intre 0-59)
setUTCMilliseconds() -	Seteaza millisecundele in un Obiect Date, in timp universal (intre 0-999)
toLocaleString() -	Converteste un Obiect Date la un sir, ce contine ora curenta
toString()	Converteste un Obiect Date la un sir
Date()	Returneaza un obiect Date

Obiectul Math include constante matematice si functii si nu este nevoie sa fie creat (instantat) un **obiect Math** inainte de a fi folosit.

Metodele obiectului Math: **abs(x), acos(x), asin(x), atan(x), atan2(y,x), cos(x), exp(x), log(x), max(x,y), min(x,y), pow(x,y), random(),round(x), sin(x), sqrt(x), tan(x) etc.**

Obiectul document, responsabil de continutul afisat pe o pagina si se foloseste pentru afisarea de paginii HTML dinamice.

Principalele proprietati ale obiectul Document sunt:

alinkColor - culoarea unei legaturi active
all - tabloul tuturor etichetelor HTML
applets - tabloul de obiecte "Applet"
bgcolor - culoarea de fundal a documentului
classes - tabloul claselor paginilor cu stiluri
cookie - fisier cookie asociat cu documentul
embeds - tablou de obiecte inglobate
fgcolor - culoarea textului in document
forms[] - tablou de obiecte "Form" (formular)

exemplu: **name = document.forms[0].elements[0].value;**

alert(document.forms[1].elements[1].value);

height - specifica inaltimea documentului in pixeli
ids - tabloul identificatorilor paginii cu stiluri
images - tablou de obiecte "Image"
linkColor - culoarea legaturilor
title - titlul documentului
URL - adresa URL a documentului curent
vlinkColor - culoarea legaturilor vizitate
width - specifica latimea documentului in pixeli

Obiecte pe partea de client:

Properties and Methods

The following properties and methods can be used on all HTML elements:

Property / Method	Description
element.accessKey	Sets or returns the accesskey for an element
element.appendChild()	Adds a new child node, to an element, as the last child node
element.attributes	Returns a NamedNodeMap of an element's attributes
element.childNodes	Returns a NodeList of child nodes for an element
element.className	Sets or returns the class attribute of an element
element.clientHeight	Returns the viewable height of an element
element.clientWidth	Returns the viewable width of an element
element.cloneNode()	Clones an element
element.compareDocumentPosition()	Compares the document position of two elements
element.dir	Sets or returns the text direction of an element
element.firstChild	Returns the first child of an element
element.getAttribute()	Returns the specified attribute value of an element node
element.getAttributeNode()	Returns the specified attribute node
element.getElementsByTagName()	Returns a collection of all child elements with the specified tagname
element.getFeature()	Returns an object which implements the APIs of a specified feature
element.getUserData()	Returns the object associated to a key on an element
element.hasAttribute()	Returns true if an element has the specified attribute, otherwise false
element.hasAttributes()	Returns true if an element has any attributes, otherwise false
element.hasChildNodes()	Returns true if an element has any child nodes, otherwise false
element.id	Sets or returns the id of an element
element.innerHTML	Sets or returns the content of an element
element.insertBefore()	Inserts a new child node before a specified, existing, child node
element.isDefaultNamespace()	Returns true if a specified namespaceURI is the default, otherwise false
element.isEqualNode()	Checks if two elements are equal

element.isSameNode()	Checks if two elements are the same node
element.isSupported()	Returns true if a specified feature is supported on the element
element.lang	Sets or returns the language code for an element
element.lastChild	Returns the last child of an element
element.namespaceURI	Returns the namespace URI of an element
element.nextSibling	Returns the next node at the same node tree level
element.nodeName	Returns the name of an element
element.nodeType	Returns the node type of an element
element.nodeValue	Sets or returns the value of an element
element.normalize()	Joins adjacent text nodes and removes empty text nodes in an element
element.offsetHeight	Returns the height of an element
element.offsetWidth	Returns the width of an element
element.offsetLeft	Returns the horizontal offset position of an element
element.offsetParent	Returns the offset container of an element
element.offsetTop	Returns the vertical offset position of an element
element.ownerDocument	Returns the root element (document object) for an element
element.parentNode	Returns the parent node of an element
element.previousSibling	Returns the previous element at the same node tree level
element.removeAttribute()	Removes a specified attribute from an element
element.removeAttributeNode()	Removes a specified attribute node, and returns the removed node
element.removeChild()	Removes a child node from an element
element.replaceChild()	Replaces a child node in an element
element.scrollHeight	Returns the entire height of an element
element.scrollLeft	Returns the distance between the left edge of an element and the view
element.scrollTop	Returns the distance between the top edge of an element and the view
element.scrollWidth	Returns the entire width of an element
element.setAttribute()	Sets or changes the specified attribute, to the specified value
element.setAttributeNode()	Sets or changes the specified attribute node
element.setIdAttribute()	

element.setIdAttributeNode()	
element.setUserData()	Associates an object to a key on an element
element.style	Sets or returns the style attribute of an element
element.tabIndex	Sets or returns the tab order of an element
element.tagName	Returns the tag name of an element
element.textContent	Sets or returns the textual content of a node and its descendants
element.title	Sets or returns the title attribute of an element
element.toString()	Converts an element to a string
nodelist.item()	Returns the node at the specified index in a NodeList
nodelist.length	Returns the number of nodes in a NodeList

```
<!DOCTYPE html>
<html>
    <head>
        <title> Prima pagina </title>
        <script type="text/javascript">
            var azi= new Date();
            var ziua = azi.getDate();
            if(ziua<10) ziua="0"+ziua;
            var luna = azi.getMonth()+1;
            if(luna<10) luna="0"+luna;
            var anul = azi.getFullYear();
            var ora = azi.getHours();
            var minutul = azi.getMinutes();
            var secunda = azi.getSeconds();
            document.write("Astazi suntem in " + ziua +"/"+ luna+"/"+ anul+", ");
            document.write("ora este " + ora +":"+ minutul+":"+ secunda);
            document.write('<style type="text/css"');

        </script>
    </head>
    <body>
    Scrie de culoare

    </body>
</html>
```

Metoda fgColor,bgColor exemple.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Metoda fgColor </title>
    </head>
    <body>
        <table align = "center" width="100%" height="100%">
            <tr>
<td width="100%" height="100%" align="center"><h1 onmouseout="document.fgColor='blue'" onmouseover="document.fgColor='red'">Text centrat</h1></td>
            </tr>
        </table>
    </body>
</html>
```

```
<!DOCTYPE html>
<html>
    <head>
        <title> Prima pagina </title>
    </head>
    <body>
        <center>
            <h1>Schimbarea culorii background-ului</h1>
            <form>
                scrie culoare:<input type="text" name="nume_culoare" size=18>
                <select name="lista_culori" onChange="document.bgColor=lista_culori.value">
                    <option  value = "aliceblue">aliceblue</option>
                    <option  value = "antiquewhite">antiquewhite</option>
                    <option  value = "#ffff00">bisque</option>
                </select>
                <input type="button" value="Schimba culoarea" onclick="document.fgColor=nume_culoare.value">
            </form>
        </center>
        <center>
        <table border=2 cellpadding=5 cellspacing=3 width="90%">
            <tr>
                <td>
                    <font color="navy">aliceblue<br/>antiquewhite<br/></font>
                </td>
            </tr>
        </table>Alt text
        </center>
    </body>
</html>
```

ID-ul este un atribut care poate fi adaugat in etichetele (sau tag-urile) HTML. Prin valoarea data acestui atribut se atribuie un nume unic etichetei respective. Acest "id" poate fi folosit in stilurile CSS pentru a defini aspectul grafic si aranjarea in pagina a tag-ului respectiv, dar poate fi folosit si in scripturi JavaScript pentru a lucra cu elementele si continutul etichetei HTML.

Id-ul poate face o legatura intre JavaScript si orice eticheta HTML din document.

Pentru a face referire intr-un script JS la o eticeta HTML, prin intermediul id-ului acesteia, se foloseste urmatoarea sintaxa:

document.getElementById("id")

- **getElementById("id")** este o metoda pentru obiectului "document" si intoare elementul care are id-ul specificat intre paranteze.
- *se pot folosi si ghilimele simple (' ') pentru numele id-ului din paranteze*.

Aceasta sintaxa "*document.getElementById("id")*" returneaza o referinta la intreg elementul HTML care are acest "id".

Pentru a face referire la anumite parti din acest obiect (element HTML), de exemplu la continut sau la un atribut "style", se folosesc proprietati specifice acestui obiect de nivel 2.
Proprietati folosite cu "*document.getElementById("id")*"

- **attributes[]** - contine intr-o matrice (*cu index de la 0*) toate atributele etichetei HTML apelate. Acestea se adauga in matrice incepand de la dreapta spre stanga. Are 2 proprietati:
  - **name** - returneaza numele atributului apelat
  - **value** - returneaza valoarea atributului apelat
- **getAttribute("atribut")** - obtine valoarea atributului specificat intre paranteze
- **setAttribute("atribut", "valoare")** - modifica valoarea atributului specificat cu valoarea data
- **removeAttribute("atribut")** - elimina existenta atributului specificat intre paranteze
- **href** - defineste sau obtine valoarea atributului "href" (*adresa URL*) din etichetele pentru link-uri
- **innerHTML** - returneaza sau schimba continutul, inclusiv cod HTML, incadrat de o eticheta HTML
- **src** - defineste sau obtine valoarea atributului "src" din etichetele <img>
- **style** - defineste valori ale atributului "style", folosit pentru elemente de stil CSS. Aceasta proprietate este urmata de o proprietatea tip CSS.
- **value** - se foloseste pentru elemente de formular (*din <form>*), obtine sau defineste valoarea unei casete (cu un anume id) din formular

Metodele getElementById si innerHTML ale obiectului document

```
<html>
<head><title>text</title>
<script type="text/javascript">
function mes1(){
  document.getElementById('diverse').innerHTML = 'Succes la JV!';
}
function mes2(){
```

```
  document.getElementById('diverse').innerHTML = 'Alt mesaj';
  document.getElementById('diverse1').innerHTML = 'Mesaj pentru H2';

        div3.value="alt mesaj sau variabila de memorie";

}


</script></head><body>


<input type="button" onclick="mes1()" value="Mesaj 1" />
<input type="button"  onclick="mes2()" value="Mesaj 2" />
<p id="diverse"></p> <hr/>
<h2 id="diverse1"></h2> <hr/>
<input type="text" size="40" id="div3"  name="div3"/>
</body>
</html>
```

Utilizare functii in fisiere externe

```
<!DOCTYPE html><html>
        <head>                  <title> Utilizare functii in fisiere externe </title>
                <link rel = "stylesheet" href = "style/style.css" type = "text/css">

                <script type="text/javascript" src="functii.js">
                </script>
                </head>
                <body>
                <table border="1" align="center">
                <tr>
                <td colspan="3" align="center">head</td>
                </tr>
                <td width="200"><br />
                </td>
                <td width="600"><br /></td>
                <td width="200">
                Spectacole recente
                <script>
                scrie();
                </script>
                </td>
                <tr>
```

```
<td colspan="3" align="center">subsol</td>
</tr>
</table>
</body></html>
```

Functie:scrie.js

```
function scrie()
{
document.write("<ol>");
document.write("<li>Nou spectacol....</li>");
document.write("<li>spectacol a....</li>");
document.write("<li>Spectocol b....</li>");
document.write("<li>Spectocol .c...</li>");
document.write("<li>Spectocol d....</li>");
document.write("<li>Spectocol ....</li>");
document.write("<li>Spectocol ....</li>");
document.write("<li>Spectocol ....</li>");
document.write("</ol>");
}
```