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.

### Toate elementele HTML sunt proprietati ale obiectului document, iar acestea sunt la randul lor. Pentru a desemna obiectul sau proprietatea curenta pe care dorim sa o folosim, adaugam cuvantul this urmat de caracterul punct (.) si numele proprietatii

### this.nume\_proprietate

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:

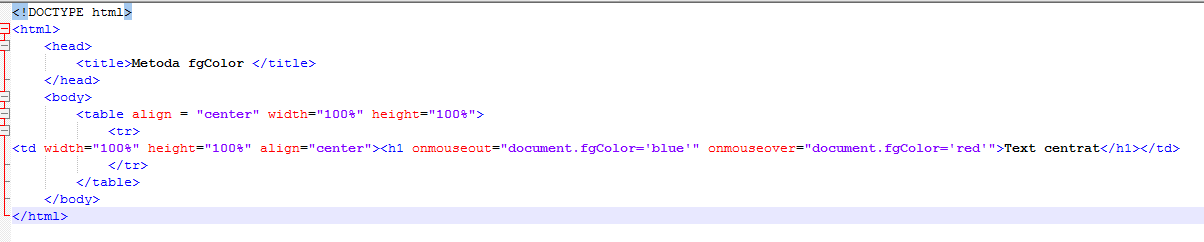
## Properties and Methods

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

|  |  |
| --- | --- |
| **Property / Method** | **Description** |
| [*element*.accessKey](http://www.w3schools.com/jsref/prop_html_accesskey.asp) | Sets or returns the accesskey for an element |
| [*element*.appendChild()](http://www.w3schools.com/jsref/met_node_appendchild.asp) | Adds a new child node, to an element, as the last child node |
| [*element*.attributes](http://www.w3schools.com/jsref/prop_node_attributes.asp) | Returns a NamedNodeMap of an element's attributes |
| [*element*.childNodes](http://www.w3schools.com/jsref/prop_node_childnodes.asp) | Returns a NodeList of child nodes for an element |
| [*element*.className](http://www.w3schools.com/jsref/prop_html_classname.asp) | 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()](http://www.w3schools.com/jsref/met_node_clonenode.asp) | Clones an element |
| [*element*.compareDocumentPosition()](http://www.w3schools.com/jsref/met_node_comparedocumentposition.asp) | Compares the document position of two elements |
| [*element*.dir](http://www.w3schools.com/jsref/prop_html_dir.asp) | Sets or returns the text direction of an element |
| [*element*.firstChild](http://www.w3schools.com/jsref/prop_node_firstchild.asp) | Returns the first child of an element |
| [*element*.getAttribute()](http://www.w3schools.com/jsref/met_element_getattribute.asp) | Returns the specified attribute value of an element node |
| [*element*.getAttributeNode()](http://www.w3schools.com/jsref/met_element_getattributenode.asp) | Returns the specified attribute node |
| [*element*.getElementsByTagName()](http://www.w3schools.com/jsref/met_element_getelementsbytagname.asp) | 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()](http://www.w3schools.com/jsref/met_element_hasattribute.asp) | Returns true if an element has the specified attribute, otherwise false |
| [*element*.hasAttributes()](http://www.w3schools.com/jsref/met_node_hasattributes.asp) | Returns true if an element has any attributes, otherwise false |
| [*element*.hasChildNodes()](http://www.w3schools.com/jsref/met_node_haschildnodes.asp) | Returns true if an element has any child nodes, otherwise false |
| [*element*.id](http://www.w3schools.com/jsref/prop_html_id.asp) | Sets or returns the id of an element |
| [*element*.innerHTML](http://www.w3schools.com/jsref/prop_html_innerhtml.asp) | Sets or returns the content of an element |
| [*element*.insertBefore()](http://www.w3schools.com/jsref/met_node_insertbefore.asp) | Inserts a new child node before a specified, existing, child node |
| [*element*.isDefaultNamespace()](http://www.w3schools.com/jsref/met_node_isdefaultnamespace.asp) | Returns true if a specified namespaceURI is the default, otherwise false |
| [*element*.isEqualNode()](http://www.w3schools.com/jsref/met_node_isequalnode.asp) | Checks if two elements are equal |
| [*element*.isSameNode()](http://www.w3schools.com/jsref/met_node_issamenode.asp) | Checks if two elements are the same node |
| [*element*.isSupported()](http://www.w3schools.com/jsref/met_node_issupported.asp) | Returns true if a specified feature is supported on the element |
| [*element*.lang](http://www.w3schools.com/jsref/prop_html_lang.asp) | Sets or returns the language code for an element |
| [*element*.lastChild](http://www.w3schools.com/jsref/prop_node_lastchild.asp) | Returns the last child of an element |
| [*element*.namespaceURI](http://www.w3schools.com/jsref/prop_node_namespaceuri.asp) | Returns the namespace URI of an element |
| [*element*.nextSibling](http://www.w3schools.com/jsref/prop_node_nextsibling.asp) | Returns the next node at the same node tree level |
| [*element*.nodeName](http://www.w3schools.com/jsref/prop_node_nodename.asp) | Returns the name of an element |
| [*element*.nodeType](http://www.w3schools.com/jsref/prop_node_nodetype.asp) | Returns the node type of an element |
| [*element*.nodeValue](http://www.w3schools.com/jsref/prop_node_nodevalue.asp) | Sets or returns the value of an element |
| [*element*.normalize()](http://www.w3schools.com/jsref/met_node_normalize.asp) | 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](http://www.w3schools.com/jsref/prop_node_ownerdocument.asp) | Returns the root element (document object) for an element |
| [*element*.parentNode](http://www.w3schools.com/jsref/prop_node_parentnode.asp) | Returns the parent node of an element |
| [*element*.previousSibling](http://www.w3schools.com/jsref/prop_node_previoussibling.asp) | Returns the previous element at the same node tree level |
| [*element*.removeAttribute()](http://www.w3schools.com/jsref/met_element_removeattribute.asp) | Removes a specified attribute from an element |
| [*element*.removeAttributeNode()](http://www.w3schools.com/jsref/met_element_removeattributenode.asp) | Removes a specified attribute node, and returns the removed node |
| [*element*.removeChild()](http://www.w3schools.com/jsref/met_node_removechild.asp) | Removes a child node from an element |
| [*element*.replaceChild()](http://www.w3schools.com/jsref/met_node_replacechild.asp) | 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()](http://www.w3schools.com/jsref/met_element_setattribute.asp) | Sets or changes the specified attribute, to the specified value |
| [*element*.setAttributeNode()](http://www.w3schools.com/jsref/met_element_setattributenode.asp) | 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](http://www.w3schools.com/jsref/prop_html_tabindex.asp) | Sets or returns the tab order of an element |
| [*element*.tagName](http://www.w3schools.com/jsref/prop_element_tagname.asp) | Returns the tag name of an element |
| [*element*.textContent](http://www.w3schools.com/jsref/prop_node_textcontent.asp) | Sets or returns the textual content of a node and its descendants |
| [*element*.title](http://www.w3schools.com/jsref/prop_html_title.asp) | Sets or returns the title attribute of an element |
| *element*.toString() | Converts an element to a string |
|  |  |
| [nodelist.item()](http://www.w3schools.com/jsref/met_nodelist_item.asp) | Returns the node at the specified index in a NodeList |
| [nodelist.length](http://www.w3schools.com/jsref/prop_nodelist_length.asp) | Returns the number of nodes in a NodeList |



**Metoda fgColor,bgColor exemple.**





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>");

}

Aplicatie cu metodele obiectului window

<html>

<head>

<title>Obiecte</title>

</head>

<script language="javascript">

function change\_page() {

document.write("O zi buna!!!");

document.bgColor="yellow";

document.fgColor="blue";

document.title="Titlu pagina";

}

function change\_status()

{

window.status="Noul status";

}

</script>

<body onload="change\_status();">

<input type="button" value="Schimba proprietatiile pajini" onClick="change\_page();">

<hr>

<script>

function deschide\_fereastra()

{

fer=window.open("ex1.html","\_blank","top=400,left=600,width=200, height=200,menubar=yes,toolbar=yes,scrollbars=yes,resizable=no")

}

function window\_confirm()

{//fer este un obiect de tip fereastra

f=confirm("Deschizi un popup?");

///fer este tip

if(f) window.open("ex1.html","\_blank","");

else //alert("Bine!");

fer.close();

}

</script>

<input type="button" value="deschide o fereastra noua" onClick="deschide\_fereastra();"><br><br>

<input type="button" value="w blur" onClick="fer.blur();">

<input type="button" value="w focus" onClick="fer.focus();"><br><br>

<input type="button" value="inchide fereastra deschisa" onClick="fer.close();"><br><br>

<input type="button" value="confirm" onClick="window\_confirm();">

</body>

</html>