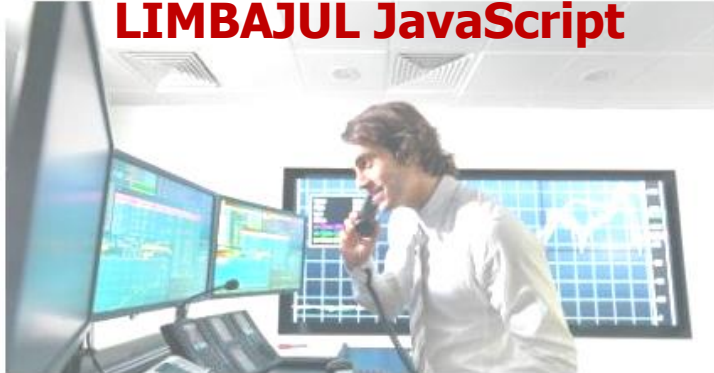




Cap.8

IMPLEMENTARE INTERFETE GRAFICE. LIMBAJUL JavaScript



Limbajul JavaScript

JavaScript/JS: limbaj scripting client-side (ruleaza la nivel browser)

Caracteristici:

- dezvoltat pentru a prelucra/valida informatii din formulare si a adauga interactivitate paginilor web
- este interpretat de browser si este incorporat in paginile HTML
- dezvoltat de Netscape , denumire initiala : LiveScript
- nu e Java , sintaxa similara cu limbajului C/Java
- ruleaza pe orice browser: Firefox, Opera, Internet Explorer, Safari, Chrome, Edge, etc.

<https://www.w3schools.com/js/>



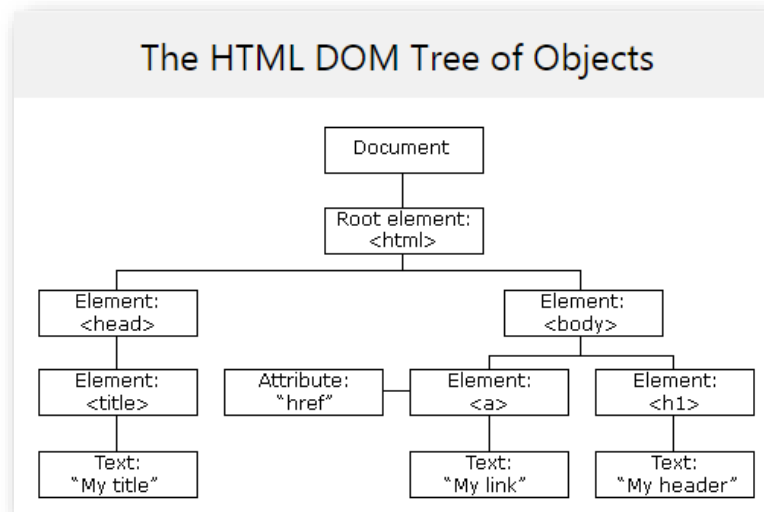
Caracteristici JavaScript

- ❑ poate reactiona la evenimente – (ex. userul a facut click pe un element HTML)
- ❑ poate citi si schimba continutul unui element HTML
- ❑ poate valida date din formulare inainte de a fi trimise catre server
- ❑ poate detecta tipul browserului - in functie de acesta se incarca o pagina sau alta



Pagina **HTML-DOM (Document Object Model)**

- ❑ Orice pagina web incarcata in browser ->DOM, graf /arbore de obiecte





Dynamic HTML

Dynamic HTML: JavaScript poate

- Schimba toate elementele HTML din pagina
- Schimba toate atributele HTML din pagina
- Schimba toate stylurile CSS din pagina
- Sterge elemente si atribute HTML din pagina
- Aadauga elemente si atribute HTML in pagina
- Poate reactiona la orice eveniment HTML din pagina
- Poate crea evenimente noi HTML in pagina



Inserarea codului JavaScript in pagina HTML

- Prin tag-ul <script>

```
<script type="text/javascript">
```

```
...
```

```
  instructiuni
```

```
...
```

```
</script>
```

sau

```
<script>
```

```
...
```

```
</script>
```

unde: Atributul type stabileste limbajul de scripting utilizat (implicit JavaScript)

Codul Java Script poate fi inserat in
<head>..</head> sau <body>...</body>



Limbajul JavaScript: afisare text , expresii

Afisare in pagina HTML: `document.write ("...")`

```
<!DOCTYPE html>
<html>
<head>
<title>Cod JavaScript</title>
</head>
<body>
  <script type="text/javascript">
    document.write("Primul test JavaScript");
  </script>
</body>
</html>
```

Primul test JavaScript

<https://et.utcluj.ro/PIGMC/JavaScript/afisare.html>



Limbajul JavaScript: afisare text , expresii

Afisare intr-o fereastra: `alert("...")`

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript </h2>

<script>
alert("Hello");
</script>

</body>
</html>
```

JavaScript

Hello

OK

<https://et.utcluj.ro/PIGMC/JavaScript/alert.html>



Inserarea codului JavaScript in pagina

Reguli: JavaScript se poate insera

a) **Intern** in documentul HTML

- atât in <head> cât și în <body>
- funcțiile JavaScript se definesc de regulă în <head>
- codul JavaScript din <body> se va executa în momentul în care este încarcata pagina

b) **Extern** într-un fisier cu extensia .js

Avantaj: fisierul poate fi inclus in mai multe pagini HTML

Ex. Includere in document HTML in sectiunea <head> sau <body>:

```
<script type="text/javascript" src="/cale/numefiser.js"></script>
```



Inserarea codului JavaScript in HTML

```
<!DOCTYPE html>
<html>
<head>
<title>Cod JavaScript</title>
</head>
<body>
  <script type="text/javascript">
    document.write("Primul test JavaScript");
  </script>
</body>
</html>
```

Primul test JavaScript

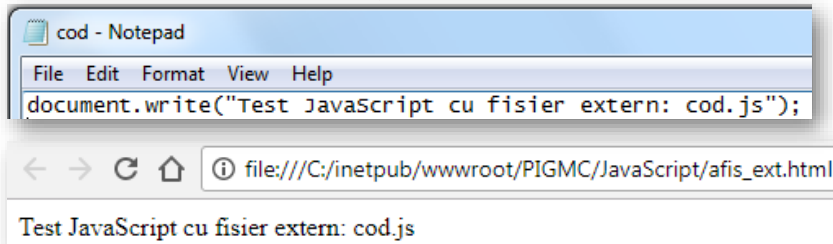
<https://et.utcluj.ro/PIGMC/JavaScript/afisare.html>



Inserarea fisier extern JavaScript

```
<!DOCTYPE html>
<head>
<title>Cod JavaScript</title>
</head>
<body>
  <script src="cod.js" type="text/javascript">
  </script>
</body>
</html>
```

https://et.utcluj.ro/PIGMC/JavaScript/afis_ext.html



Metode HTML DOM

Metode DOM adaugare /stergere/inlocuire/afisare elemente

Method	Description
<code>document.createElement(element)</code>	Create an HTML element
<code>document.removeChild(element)</code>	Remove an HTML element
<code>document.appendChild(element)</code>	Add an HTML element
<code>document.replaceChild(element)</code>	Replace an HTML element
<code>document.write(text)</code>	Write into the HTML output stream

Metode DOM adaugare functii gestionare evenimente HTML

Method	Description
<code>document.getElementById(id).onclick = function(){code}</code>	Adding event handler code to an onclick event



Metode HTML DOM

Metode DOM gasire elemente

Method	Description
<code>document.getElementById(id)</code>	Find an element by element id
<code>document.getElementsByTagName(name)</code>	Find elements by tag name
<code>document.getElementsByClassName(name)</code>	Find elements by class name

Metode DOM modificare elemente

Method	Description
<code>element.innerHTML = new html content</code>	Change the inner HTML of an element
<code>element.attribute = new value</code>	Change the attribute value of an HTML element
<code>element.setAttribute(attribute, value)</code>	Change the attribute value of an HTML element
<code>element.style.property = new style</code>	Change the style of an HTML element



Proprietati HTML DOM

Proprietati DOM selectare elemente HTML

Property	Description
<code>document.implementation</code>	Returns the DOM implementation
<code>document.anchors</code>	Returns all <a> elements that have a name attribute
<code>document.applets</code>	Returns all <applet> elements (Deprecated in HTML5)
<code>document.baseURI</code>	Returns the absolute base URI of the document
<code>document.body</code>	Returns the <body> element
<code>document.cookie</code>	Returns the document's cookie
<code>document.doctype</code>	Returns the document's doctype
<code>document.documentElement</code>	Returns the <html> element
<code>document.documentMode</code>	Returns the mode used by the browser
<code>document.documentURI</code>	Returns the URI of the document
<code>document.domain</code>	Returns the domain name of the document server
<code>document.domConfig</code>	Obsolete. Returns the DOM configuration
<code>document.embeds</code>	Returns all <embed> elements
<code>document.forms</code>	Returns all <form> elements
<code>document.head</code>	Returns the <head> element
<code>document.images</code>	Returns all elements
<code>document.inputEncoding</code>	Returns the document's encoding (character set)
<code>document.lastModified</code>	Returns the date and time the document was updated
<code>document.links</code>	Returns all <area> and <a> elements that have href attribute
<code>document.readyState</code>	Returns the (loading) status of the document
<code>document.referrer</code>	Returns the URI of the referrer (the linking document)
<code>document.scripts</code>	Returns all <script> elements
<code>document.strictErrorChecking</code>	Returns if error checking is enforced
<code>document.title</code>	Returns the <title> element
<code>document.URL</code>	Returns the complete URL of the document



Sintaxa limbaj JavaScript

Conventii de sintaxa

1. Limbaj Case-sensitive
2. Instructiunile se termina cu punct si virgula: (;)
3. Spatii libere: JavaScript ignora spatiile libere, tab-urile si liniile libere care apar în instructiuni,
4. Ghilimelele: ' si " delimitaaza siruri de caractere (string).
5. Caractere speciale:
 - \b - backspace
 - \f - indica o pagina noua
 - \n - linie noua
 - \r - indica un carriage return
 - \t - indica TAB
 - \\ - caracter backslash
 - \' - indica un apostrof (ghilimele simple)
 - \" - indica ghilimele duble



Limbajul JavaScript: afisare cu metoda write

```
document.write("Hello World!");
```

```
document.write(exp1, exp2, exp3, ...)
```

```
document.write("<h1>Hello World!</h1><p>Have a nice day!</p>");
```

```
document.write(Date());
```




Limbajul JavaScript: Variabile

Sintaxa :

```
var idVar; idVar = val_initala;  
var idVar = val_initala;
```

- Nume variabile=case-sensitive** , obligatoriu incep cu o litera
- Variabilele nu au tip** (ele pot retine orice valoare)
- Cuvantul var este optional (daca nu se specifica automat variabila e considerata globala)
- Variabilele declarate intr-o functie sunt locale acelei functii
- La declararea variabilelor locale trebuie obligatoriu var
- Variabilele declarate in afara oricarei functii sunt globale (sunt accesibile oriunde in pagina)



Limbajul JavaScript: Variabile

```
<script>  
var length = 16; // Number  
var lastName = "Johnson"; // String  
var cars = ["Saab", "Volvo", "BMW"]; // Array  
document.write('length=',length,' ','lastName=', lastName,' ','cars=', cars);  
</script>
```

```
length=16 ;lastName=Johnson ;cars=Saab,Volvo,BMW
```

```
<script>  
length = 16; // Number  
lastName = "Johnson"; // String  
cars = ["Saab", "Volvo", "BMW"]; // Array  
document.write('length=',length,' ','lastName=', lastName,' ','cars=', cars);  
</script>
```



Limbajul JavaScript: Tipuri de baza

JavaScript are 3 tipuri de baza: number, string, si boolean
(Orice altceva este obiect)

- ❑ **Numerele** sunt memorate intodeauna in virgula flotanta
 - ❑ Numerele **hexazecimale** incep cu 0x
 - ❑ Numerele in **baza 8** incep cu 0 (nu toate browserele suporta acest format)
- ❑ **Sirurile (string)**:secvente de caractere cuprinse “ ” sau ‘ ’
 - ❑ pot contine \n (newline), \" (ghilimele), etc.
- ❑ **Valorile logice (boolean)**: true sau false



Limbajul JavaScript: Variabile numerice

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Variables</h2>
<p>In acest exemplu, variabilele x=5, y=6,
se cunosc.Calculam variabila z=x+y.</p>
<p id="demo"></p>
<script>
var x = 5;
var y = 6;
var z = x + y;
document.write("Rezultatul este z: " + z);
</script>
</body>
</html>
```

JavaScript Variables

In acest exemplu, variabilele x=5, y=6, se cunosc.Calculam variabila z=x+y.

Rezultatul este z: 11

<https://et.utcluj.ro/PIGMC/JavaScript/var1.html>



Limbajul JavaScript: Variabile numerice

<pre><!DOCTYPE html> <html> <body> <h2>JavaScript </h2> <script> document.write(5 + 6); </script> </body> </html></pre>	<h3>JavaScript</h3> <p>11</p>
---	-------------------------------



Limbajul JavaScript: Variabile siruri

<pre><!DOCTYPE html> <html> <body> <h2>JavaScript Variables</h2> <p>In acest exemplu, sunt x,y,z,sunt variabile sir Calculam variabila z=x+y (concatenare). </p> <p id="demo"></p> <script> var x ="Buna seara studenti Master."; var y = "Bine ati venit la PIGMC"; var z = x + y; document.write("Rezultatul este z: " + z); </script> </body> </html></pre>	<h3>JavaScript Variables</h3> <p>In acest exemplu, sunt x,y,z,sunt variabile sir Calculam variabila z=x+y (concatenare).</p> <p>Rezultatul este z: Buna seara studenti Master.Bine ati venit la PIGMC</p>
--	---

<https://et.utcluj.ro/PIGMC/JavaScript/var2.html>



Limbajul JavaScript: Operatori

- ❑ Aritmetici: + - * / % ++ --
- ❑ Comparatie: < <= == != >= >
- ❑ Logici: && || !
- ❑ Biti: & | ^ ~ << >> >>>
- ❑ Atribuire: = += -= *= /= %= < <= > >= >>> &= ^= |=
- ❑ Concatenarea siruri: +
- ❑ Conditional : conditie ? val_if_true : val_if_false n
- ❑ Testare egalitate: == si !=
- ❑ Alti operatori : new typeof delete



Limbajul JavaScript: operator ?:

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript operator ?:</h2>
<p>In acest exemplu testam o conditie
si afisam rezultatul in functie de
conditie |adevarat sau fals</p>
<script>
var vizitator="barbat";
mesaj=(vizitator=="barbat")?"Stimate
Domn":"Stimate Doamna";
document.write(mesaj);
</script>
</body>
</html>
```

JavaScript operator ?:

In acest exemplu testam o conditie si afisam rezultatul in functie de conditie adevarat sau fals

Stimate Domn

<https://et.utcluj.ro/PIGMC/JavaScript/operator1.html>



Limbajul JavaScript: operator +=

<pre><!DOCTYPE html> <html> <body> <h2>Operatorul +=</h2> <p>x=10, x+=5= <script> var x = 10; x += 5; document.write(x); </script> </body> </html></pre>	<h3>Operatorul +=</h3> <p>x=10, x+=5= 15</p>
--	--

<https://et.utcluj.ro/PIGMC/JavaScript/operator2.html>



Limbajul JavaScript: operator concatenare +

<pre><!DOCTYPE html> <html> <body> <h2>JavaScript </h2> <script> t1 = "Astazi este o zi" t2 = " frumoasa" t3 = t1+t2 document.write(t1 + t2); </script> </body> </html></pre>	<h3>JavaScript</h3> <p>Astazi este o zi frumoasa</p>
---	--



Limbajul JavaScript: operator typeof

```
<!DOCTYPE html>
<html>
<body>
<h2>typeof in JavaScript</h2>
<script>
var x = "JavaScript";
document.write(typeof(x));
</script>
</body>
</html>
```

typeof in JavaScript

string

<https://et.utcluj.ro/PIGMC/JavaScript/typeof.html>



Limbajul JavaScript: operator **

```
<!DOCTYPE html>
<html>
<body>

<h2>Operatorul putere:** </h2>
<script>
var x = 2;
document.write("x^10="+ x ** 10);
</script>

</body>
</html>
```

Operatorul putere:**

$x^{10}=1024$

<https://et.utcluj.ro/PIGMC/JavaScript/power.html>



Limbajul JavaScript: Instructiuni

- ❑ **Atribuire:** `idVar = expresie;`
- ❑ **Instructiunea compusa**

```
{ instructiune1;  
  instructiune2;  
  ...  
  instructiune n; }
```
- ❑ **Comentarii: (similar C++ sau Java)**

```
// comentariu pe o singura linie  
/* comentarii pe mai multe linii */
```



Limbajul JavaScript: Instructiuni conditionale

- ❑ **Selectie simpla:**

```
if (conditie) instructiune;  
if (conditie) instructiune1; else instructiune2;
```
- ❑ **Selectie multipla:**

```
switch(n) {  
  case constanta1: bloc instructiuni 1 break;  
  ...  
  case constantaN: bloc instructiuni N break;  
  default: bloc instructiuni n+1 }
```



Limbajul JavaScript: **If**

```
<!DOCTYPE html>
<html>
<body>

<h2>Instructiune if </h2>
<script>
var x = "PIGMC";
if (x=="PIGMC")document.write("Corect");
else document.write ("Gresit");
</script>

</body>
</html>
```

Instructiune if

Corect

<https://et.utcluj.ro/PIGMC/JavaScript/if.html>



Limbajul JavaScript: **switch**

```
<!DOCTYPE html>
<html>
<body>
<h2>switch in JavaScript</h2>
<script>
var nume = "Marius";
switch (nume) {
  case "Cristi":
    document.write("Coleg");
    break
  case "Marius":
    document.write("Frate");
    break
  case "Maria":
    document.write("Sora");
    break
  default:
    document.write("Altceineva");
}
</script>
</body>
</html>
```

switch in JavaScript

Frate



Limbajul JavaScript: Instructiuni ciclare

❑ Instructiunea for

for (expr_iniciala;conditie; expr_reiterare) instructiune

<pre><!DOCTYPE html> <html> <body> <h2>Instructiune for, i=1,...,5 </h2> <script> var i=1,s=0,p=1; for (i=1;i<5;i++) {s+=i;p*=i;} document.write("s="+s+" p=" + p); </script> </body> </html></pre>	<p>Instructiune for, i=1,...,5</p> <p>s=10 p=24</p>
--	--

<https://et.utcluj.ro/PIGMC/JavaScript/for.html>



Limbajul JavaScript: Instructiuni ciclare

❑ Instructiunea while : while (conditie) instructiune

<pre><!DOCTYPE html> <html> <body> <h2>Instructiune while i=1,...,5 </h2> <script> var i=1,s=0,p=1; while(i<5) {s+=i;p*=i;i++;} document.write("s="+s+" p=" + p); </script> </body> </html></pre>	<p>Instructiune while i=1,...,5</p> <p>s=15 p=120</p>
---	--

<https://et.utcluj.ro/PIGMC/JavaScript/while.html>



Limbajul JavaScript: Instructiuni ciclare

- Instructiunea do ... while

do { instructiuni } while (conditie);

<pre><!DOCTYPE html> <html> <body> <h2>Instructiune do while i=1,...,5 </h2> <script> var i=1,s=0,p=1; do {s+=i;p*=i;i++;} while(i<=5) document.write("s="+s+" p=" + p); </script> </body> </html></pre>	<p>Instructiune do while i=1,...,5</p> <p>s=15 p=120</p>
--	---

<https://et.utcluj.ro/PIGMC/JavaScript/dowhile.html>



Limbajul JavaScript: Functii

- se definesc de regula in <head>

- Sintaxa definire functie:

```
function numeFunctie(p1, ..., pN)
    { //declaratii de variabile locale (var)
    instructiuni }
```

- O functie poate returna o valoare cu **return valoare**;

- Sintaxa apel functie: **numeFunctie(vp1, ..., vpN)**

- Parametri simpli sunt transmisi prin valoare, obiectele prin referinta



Limbajul JavaScript: Functie afisare

<pre><!DOCTYPE html> <html> <head> <script> function myFunction() { document.write("Buna seara Masteranzi!!"); } </script> </head> <body> <p>Apasati pe butonul"Try it", pentru apelul functiei</p> <p>Functia va afisa un mesaj.</p> <button onClick="myFunction()">Try it</button> </body> </html></pre>	<p>Apasati pe butonul"Try it", pentru apelul functiei</p> <p>Functia va afisa un mesaj.</p> <p><input type="button" value="Try it"/></p>
---	--

← → ↻ 🏠 ⓘ file:///C:/inetpub/wwwroot/PIGMC/JavaScript/buton.html

Buna seara Masteranzi!!

<https://et.utcluj.ro/PIGMC/JavaScript/buton.html>



Limbajul JavaScript: Functie care returneaza valoare

<pre><!DOCTYPE html> <html> <body> <h2>Functie JavaScript</h2> <p>Functia calculeaza produsul a doua variabile si returneaza rezultatul</p> <script> function myFunction(a, b) { return a * b; } document.write(myFunction(10,20)); </script> </body> </html></pre>	<h3>Functie JavaScript</h3> <p>Functia calculeaza produsul a doua variabile si returneaza rezultatul</p> <p>200</p>
--	---

<https://et.utcluj.ro/PIGMC/JavaScript/functie.html>



Limbajul JavaScript: **Funcție care returnează valoare**

```
<!DOCTYPE html>
<html>
<body>

<h2>Funcție JavaScript</h2>

<p>Conversie Fahrenheit in Celsius:</p>
<script>
function toCelsius(f) {
    return (5/9) * (f-32);
}
document.write(toCelsius(77));
</script>

</body>
</html>
```

Funcție JavaScript

Conversie Fahrenheit in Celsius:

25



Limbajul JavaScript: **Tablouri (arrays)**

Tablourile sunt **obiecte** in JavaScript

Sintaxa definire tablou:

```
var nume=[v1,v2,...vN]; //indicii pornesc de la 0
```

Tiparire al 2-lea element din tablou:

```
<!DOCTYPE html>
<html>
<body>

<h2>Tablouri JavaScript</h2>
<script>
var cars = ["Saab","Volvo","BMW"];
document.write(cars[1]);
</script>
</body>
</html>
```

Tablouri JavaScript

Volvo



Limbajul JavaScript: Tablouri (arrays)

- Tiparire intreg tabloul

<pre><!DOCTYPE html> <html> <body> <h2>JavaScript Arrays</h2> <script> var cars = ["Saab", "Volvo", "BMW"]; document.write(cars); </script> </body> </html></pre>	<h3>JavaScript Arrays</h3> <p>Saab,Volvo,BMW</p>
---	--

<file:///C:/inetpub/wwwroot/PIGMC/JavaScript/siruri.html>



Limbajul JavaScript: Tablouri (arrays)

- Tiparire intreg tabloul specificand fiecare element

<pre><!DOCTYPE html> <html> <body> <h2>JavaScript Arrays (enumerari)</h2> <p id="demo"></p> <script> var person = ["Popescu", "Anca", 28]; document.write(person[0]+" "+person[1]+" "+ person[2]); </script> </body> </html></pre>	<h3>JavaScript Arrays (enumerari)</h3> <p>Popescu Anca 28</p>
---	---



Limbajul JavaScript: Tablouri (arrays)

Sortare alfabetica tablou

<pre><!DOCTYPE html> <html> <body> <h2>Sortare alfabetica siruri JavaScript</h2> <button onclick="myFunction()">Sorteaza</button> <script> var fruits = ["Banane", "Portocale", "Mere", "Clementine"]; document.write(fruits); function myFunction() { fruits.sort(); document.write(fruits); } </script> </body> </html></pre>	Banane,Clementine,Mere,Portocale
---	----------------------------------

Sortare alfabetica siruri JavaScript

Sorteaza Banane,Portocale,Mere,Clementine

<https://et.utcluj.ro/PIGMC/JavaScript/sortaresir.html>



Limbajul JavaScript: Tablouri (arrays)

Lungime / dimensiune tablou (**length**)

<pre><!DOCTYPE html> <html> <body> <h2>Lungime tablou JavaScript</h2> <script> var fruits = ["Banane", "Portocale", "Mere", "Clementine"]; document.write(fruits.length); </script> </body> </html></pre>	Lungime tablou JavaScript 4
--	---------------------------------------



Limbajul JavaScript: Tablouri (arrays)

☐ Metode : functii specifice siruri

<code>reduce()</code>	Reduce the values of an array to a single value (going left-to-right)
<code>reduceRight()</code>	Reduce the values of an array to a single value (going right-to-left)
<code>reverse()</code>	Reverses the order of the elements in an array
<code>shift()</code>	Removes the first element of an array, and returns that element
<code>slice()</code>	Selects a part of an array, and returns the new array
<code>some()</code>	Checks if any of the elements in an array pass a test
<code>sort()</code>	Sorts the elements of an array
<code>splice()</code>	Adds/Removes elements from an array
<code>toString()</code>	Converts an array to a string, and returns the result
<code>unshift()</code>	Adds new elements to the beginning of an array, and returns the new length
<code>valueOf()</code>	Returns the primitive value of an array
<code>map()</code>	Creates a new array with the result of calling a function for each array element
<code>pop()</code>	Removes the last element of an array, and returns that element
<code>push()</code>	Adds new elements to the end of an array, and returns the new length



Limbajul JavaScript: Obiecte

Metode creare obiecte in JavaScript:

☐ Utilizand obiecte literal:

```
var curs = {"D02", "PIGMC"}
```

☐ Cream un obiect "gol" cu ajutorul operatorului new, apoi adaugam proprietati:

```
var curs = new Object();  
curs.id = " D02 "; curs.nume = "PIGMC";
```

☐ Cu ajutorul constructorilor:

```
function Curs(id, nume)  
{ this.id = id;  
  this.nume = nume; }  
var curs =new Curs("D02", "PIGMC");
```



Limbajul JavaScript: **creare obiecte**

Exemplu

<pre><!DOCTYPE html> <html> <body> <p>Creare obiect JavaScript</p> <p id="demo"></p> <script> var car = {type:"Fiat", model:"500", color:"white"}; document.write(car.type); </script> </body> </html></pre>	<p>Creare obiect JavaScript</p> <p>Fiat</p>
--	---



Limbajul JavaScript: **creare obiecte**

Exemplu

<pre><!DOCTYPE html> <html> <body> <p>Creare obiect persoana JavaScript </p> <p id="demo"></p> <script> var person = {firstName:"Grindei", lastName:"Laura", age:30, eyeColor:"albastru"}; document.write(person.lastName + " are " + person.age + " ani."); </script> </body> </html></pre>	<p>Creare obiect persoana JavaScript</p> <p>Laura are 30 ani.</p>
---	---



Limbajul JavaScript: creare obiecte mai multe linii

Exemplu

<pre><!DOCTYPE html> <html> <body> <p>Creating a JavaScript Object.</p> <p id="demo"></p> <script> var person = { firstName : "Grindei", lastName : "Laura", age : 30, eyeColor : "albastru" }; document.write(person.lastName + " are " + person.age); </script> </body> </html></pre>	<p>Creating a JavaScript Object.</p> <p>Laura are 30</p>
--	--



Obiecte predefinite: Math

Proprietati: constante matematice

```
Math.E      // returns Euler's number
Math.PI     // returns PI
Math.SQRT2  // returns the square root of 2
Math.SQRT1_2 // returns the square root of 1/2
Math.LN2    // returns the natural logarithm of 2
Math.LN10   // returns the natural logarithm of 10
Math.LOG2E  // returns base 2 logarithm of E
Math.LOG10E // returns base 10 logarithm of E
```



Obiecte predefinite: Math

Proprietati: constante matematice

```

<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Math Constants</h2>
<script>
document.write(
"<p><b>Math.E:</b> " + Math.E + "</p>" +
"<p><b>Math.PI:</b> " + Math.PI + "</p>" +
"<p><b>Math.SQRT2:</b> " + Math.SQRT2 + "</p>" +
"<p><b>Math.SQRT1_2:</b> " + Math.SQRT1_2 + "</p>" +
"<p><b>Math.LN2:</b> " + Math.LN2 + "</p>" +
"<p><b>Math.LN10:</b> " + Math.LN10 + "</p>" +
"<p><b>Math.LOG2E:</b> " + Math.LOG2E + "</p>" +
"<p><b>Math.Log10E:</b> " + Math.Log10E + "</p>");
</script>
</body>
</html>

```

JavaScript Math Constants

- Math.E:** 2.718281828459045
- Math.PI:** 3.141592653589793
- Math.SQRT2:** 1.4142135623730951
- Math.SQRT1_2:** 0.7071067811865476
- Math.LN2:** 0.6931471805599453
- Math.LN10:** 2.302585092994046
- Math.LOG2E:** 1.4426950408889634
- Math.Log10E:** 0.4342944819032518

<https://et.utcluj.ro/PIGMC/JavaScript/math1.html>



Metode

Method	Description
abs(x)	Returns the absolute value of x
acos(x)	Returns the arccosine of x, in radians
asin(x)	Returns the arcsine of x, in radians
atan(x)	Returns the arctangent of x as a numeric value between -PI/2 and PI/2 radians
atan2(y, x)	Returns the arctangent of the quotient of its arguments
ceil(x)	Returns the value of x rounded up to its nearest integer
cos(x)	Returns the cosine of x (x is in radians)
exp(x)	Returns the value of E ^x
floor(x)	Returns the value of x rounded down to its nearest integer
log(x)	Returns the natural logarithm (base E) of x
max(x, y, z, ..., n)	Returns the number with the highest value
min(x, y, z, ..., n)	Returns the number with the lowest value
pow(x, y)	Returns the value of x to the power of y
random()	Returns a random number between 0 and 1
round(x)	Returns the value of x rounded to its nearest integer
sin(x)	Returns the sine of x (x is in radians)
sqrt(x)	Returns the square root of x
tan(x)	Returns the tangent of an angle



Obiecte predefinite: **Math**

Metode: functii matematice

```
<!DOCTYPE html>
<html>
<body>
<h2>Functii matematice in JavaScript</h2>
<script>
document.write ("Exp(1)="+Math.exp(1));
document.write ("<br>"+"max(10,55,22,11,100="
+Math.max(10,55,22,11,100));
document.write ("<br>"+"pow(10,2)="+Math.pow(10,2));
document.write ("<br>"+"sqrt(9)="+Math.sqrt(9));
document.write ("<br>"+"sin(pi/2)="+Math.sin(Math.PI/2));
</script>
</body>
</html>
```

Functii matematice in JavaScript

```
Exp(1)=2.718281828459045
max(10,55,22,11,100)=100
pow(10,2)=100
sqrt(9)=3
sin(pi/2)=1
```

<https://et.utcluj.ro/PIGMC/JavaScript/math2.html>



Functii predefinite: data curenta **Date()**

Afisare data curenta: Date

```
<!DOCTYPE html>
<html>
<body>

<script>
document.write(Date());
</script>

</body>
</html>
```

```
Wed Nov 27 2019 03:23:50 GMT+0200
(Ora standard a Europei de Est)
```

<https://et.utcluj.ro/PIGMC/JavaScript/date.html>



Exemplu data curenta cu buton

Afisare data curenta buton: Date

<pre><!DOCTYPE html> <html> <body> <h1>Buton Data de azi</h1> <button type="button" onClick="document.write(Date())"> Click pentru a afla Data si Ora acum. </button> </body> </html></pre>	<h3>Buton Data de azi</h3> <p>Click pentru a afla Data si Ora acum.</p>
---	---

https://et.utcluj.ro/PIGMC/JavaScript/date_but.html



Metode pentru date calendaristice

Method	Description
getDate()	Get the day as a number (1-31)
getDay()	Get the weekday as a number (0-6)
getFullYear()	Get the four digit year (yyyy)
getHours()	Get the hour (0-23)
getMilliseconds()	Get the milliseconds (0-999)
getMinutes()	Get the minutes (0-59)
getMonth()	Get the month (0-11)
getSeconds()	Get the seconds (0-59)
getTime()	Get the time (milliseconds since January 1, 1970)



Metode pentru date calendaristice

Exemplu : pentru data 27.11.2019, getDate() preia ziua ca nr.

```
<!DOCTYPE html>
<html>
<body>
<p>Afiseaza ziua curenta</p>
<button onclick="myFunction()">Try it</button>
<script>
function myFunction() {
  var d = new Date();
  var n = d.getDate();
  document.write(n);
}
</script>
</body>
</html>
```

Afiseaza ziua curenta

Try it

27



Metode pentru date calendaristice

Exemplu : pentru data 27.11.2019, getDate() preia ziua ca nr.

```
<!DOCTYPE html>
<html>
<body>

<p>Afiseaza anul curent</p>

<button onclick="myFunction()">Try it</button>
<script>
function myFunction() {
  var d = new Date();
  var n = d.getFullYear();
  document.write(n);
}
</script>

</body>
</html>
```

Afiseaza anul curent

Try it

2019



Limbajul JavaScript: creare ferestre

name	Sets or returns the name of a window
navigator	Returns the Navigator object for the window (See Navigator object)
opener	Returns a reference to the window that created the window
outerHeight	Returns the outer height of a window, including toolbars/scrollbars
outerWidth	Returns the outer width of a window, including toolbars/scrollbars
pageXOffset	Returns the pixels the current document has been scrolled (horizontally) from the upper left corner of the window
pageYOffset	Returns the pixels the current document has been scrolled (vertically) from the upper left corner of the window
parent	Returns the parent window of the current window
screen	Returns the Screen object for the window (See Screen object)
screenLeft	Returns the horizontal coordinate of the window relative to the screen
screenTop	Returns the vertical coordinate of the window relative to the screen
screenX	Returns the horizontal coordinate of the window relative to the screen
screenY	Returns the vertical coordinate of the window relative to the screen
sessionStorage	Allows to save key/value pairs in a web browser. Stores the data for one session
scrollX	An alias of pageXOffset
scrollY	An alias of pageYOffset
self	Returns the current window
status	Sets or returns the text in the statusbar of a window
top	Returns the topmost browser window



Limbajul JavaScript: creare ferestre

Metode:

Method	Description
alert()	Displays an alert box with a message and an OK button
atob()	Decodes a base-64 encoded string
blur()	Removes focus from the current window
getComputedStyle()	Gets the current computed CSS styles applied to an element
getSelection()	Returns a Selection object representing the range of text selected by the user
matchMedia()	Returns a MediaQueryList object representing the specified CSS media query string
moveBy()	Moves a window relative to its current position
moveTo()	Moves a window to the specified position
open()	Opens a new browser window
print()	Prints the content of the current window
prompt()	Displays a dialog box that prompts the visitor for input
resizeBy()	Resizes the window by the specified pixels
resizeTo()	Resizes the window to the specified width and height
scroll()	Deprecated. This method has been replaced by the scrollTo() method.
scrollBy()	Scrolls the document by the specified number of pixels
scrollTo()	Scrolls the document to the specified coordinates
setInterval()	Calls a function or evaluates an expression at specified intervals (in milliseconds)
setTimeout()	Calls a function or evaluates an expression after a specified number of milliseconds
stop()	Stops the window from loading

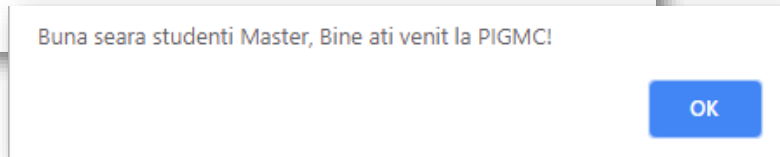


Limbajul JavaScript: creare ferestre Alert window: alert()

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Alert window</h2>
<p>Fereastra afiseaza o fereastră tip Alert window</p>
<script>
alert("Buna seara studenti Master, Bine ati venit la PIGMC!" );
</script>

</body>
</html>
```



La incarcarea paginii in browser, automat se deschide o fereastră alert cu textul "Buna seara.."

<https://et.utcluj.ro/PIGMC/JavaScript/alert.html>



Limbajul JavaScript: creare ferestre Alert window

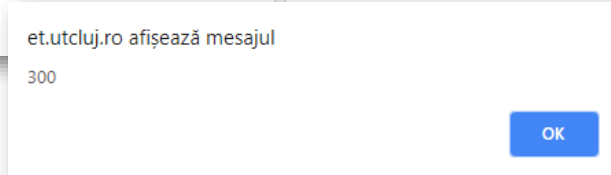
```
<!DOCTYPE html>
<html>
<body>

<h2>Rezultatul expresiei 100+120+80 va fi
afisat intr-o fereastră tip alert</h2>
<p>Fereastra afiseaza o fereastră tip Alert
window</p>
<script>
alert(100+120+80 );
</script>

</body>
</html>
```

Rezultatul expresiei 100+120+80 va fi afisat intr-o fereastră tip alert

Fereastra afiseaza o fereastră tip Alert window



https://et.utcluj.ro/PIGMC/JavaScript/alert_expresie.html



Limbajul JavaScript: creare ferestre Confirm window

```
<!DOCTYPE html>
<html>
<body>

<p>Este corect 2^10=1024??</p>|
<button onclick="myFunction()">Verifica</button>
<script>
function myFunction() {
  confirm("Corect");
}
</script>
</body>
</html>
```

Este corect 2¹⁰=1024??

Verifica

O pagină încorporată de pe această pagină afișează mesajul
Corect

OK

Anulează

<https://et.utcluj.ro/PIGMC/JavaScript/confirmwindow.html>



Limbajul JavaScript: creare ferestre Prompt window

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript </h2>
<script>
x=prompt("Varsta", " ");
document.write("Varsta ta <br>",+x);
</script>
</body>
</html>
```

JavaScript

Varsta ta
35

et.utcluj.ro afișează mesajul

Varsta

35

OK

Anulează

<https://et.utcluj.ro/PIGMC/JavaScript/prompt.html>



Metode HTML DOM

Metode DOM adaugare /stergere/inlocuire/afisare elemente

Method	Description
<code>document.createElement(<i>element</i>)</code>	Create an HTML element
<code>document.removeChild(<i>element</i>)</code>	Remove an HTML element
<code>document.appendChild(<i>element</i>)</code>	Add an HTML element
<code>document.replaceChild(<i>element</i>)</code>	Replace an HTML element
<code>document.write(<i>text</i>)</code>	Write into the HTML output stream

Metode DOM adaugare functii gestionare evenimente HTML

Method	Description
<code>document.getElementById(<i>id</i>).onclick = function(){<i>code</i>}</code>	Adding event handler code to an onclick event



Metode HTML DOM

Metode DOM gasire elemente

Method	Description
<code>document.getElementById(<i>id</i>)</code>	Find an element by element id
<code>document.getElementsByTagName(<i>name</i>)</code>	Find elements by tag name
<code>document.getElementsByClassName(<i>name</i>)</code>	Find elements by class name

Metode DOM modificare elemente

Method	Description
<code><i>element</i>.innerHTML = <i>new html content</i></code>	Change the inner HTML of an element
<code><i>element</i>.attribute = <i>new value</i></code>	Change the attribute value of an HTML element
<code><i>element</i>.setAttribute(<i>attribute</i>, <i>value</i>)</code>	Change the attribute value of an HTML element
<code><i>element</i>.style.<i>property</i> = <i>new style</i></code>	Change the style of an HTML element



Metode DOM gasire elemente: **getElementById**

```
<!DOCTYPE html>
<html>
<body>
<h2> identificare element HTML dupa id</h2>
<p id="intro">Limbajul Java Script</p>
<p id="demo"></p>
<script>
var myElement = document.getElementById("intro");
document.getElementById("demo").innerHTML =
"Elementul cu id=demo a fost identificat : " +
myElement.innerHTML;
</script>
</body>
</html>
```

identificare element HTML dupa id

Limbajul Java Script

Elementul cu id=demo a fost identificat : Limbajul Java Script

Daca elementul este identificat in pagina html cu metoda getElementId() , se returneaza elementul ca un obiect in variabila myElement.

Daca elementul nu este gasit , myElement va contine null si nu va afisa nimic. Afisarea elementului "intro" se face in elemental "demo" utilizand innerHTML



Proprietatea **innerHTML**

innerHTML: este o **proprietate a unui obiect HTML**.

HTMLObject.innerHTML

Set innerHTML : atribuim elementului continut nou

HTMLObject.innerHTML = text

unde text=specifica continut nou

```
<p id="intro">Limbajul Java Script</p>
<p id="demo"></p>
<script>
var myElement = document.getElementById("intro");
document.getElementById("demo").innerHTML =
"Elementul cu id=demo a fost identificat : " +
myElement.innerHTML;
```



Metode DOM gasire elemente : **getElementById** cu actiune pe buton

Exemplu : Metoda cauta un element HTML (cu id="demo") si schimba continutul elementului inlocuindu-l cu "Buna ziua studenti Master!":

```
<!DOCTYPE html>
<html>
<body>
<p id="demo">Schimba continut html apasand pe buton</p>

<button type="button" onclick='document.getElementById("demo").innerHTML = "Buna ziua studenti Master!'">Click!</button>

</body>
</html>
```

Schimba continut html apasand pe buton

Click!

Buna ziua studenti Master!

Click!



Prelucrare formulare in JavaScript: **.value**

Afisarea camp text dintr-un formular: se preia valoarea (sirul de caractere) din campul text

```
<!DOCTYPE html>
<html>
<body>
<h3>Acces la un camp de tip text din formular</h3>
<form>
<input type="text" id="myText" ></form>
<p>Click "Afiseaza" pentru a afisa textul din
campul text al formularului</p>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
function myFunction() {
  var x = document.getElementById("myText").value;
  document.getElementById("demo").innerHTML = x;
}
</script>
</body>
</html>
```

Acces la un camp de tip text din formular

Buna ziua

Click "Afiseaza" pentru a afisa textul din campul text al formularului

Try it

Buna ziua



Prelucrare formulare in JavaScript : **.elements.value**

Afisarea tuturor campurilor dintr-un formular

```
<!DOCTYPE html>
<html>
<body>
<form id="myForm" >
  Nume: <input type="text" name="fname" value="Popa"><br>
  Prenume: <input type="text" name="lname" value="Anca"><br>
  <input type="submit" value="Submit">
</form>
<p>Click "Afiseaza" fiecare element din formular</p>
<button onclick="myFunction()">Afiseaza</button>
<p id="demo"></p>
<script>
function myFunction() {
  var x = document.getElementById("myForm");
  var txt = "Buna ziua ";
  var i = 0;
  for (i = 0; i < x.length-1; i++) {
    txt = txt + x.elements[i].value + " ";
  }
  document.getElementById("demo").innerHTML = txt;
}
</script>
</body>
</html>
```

Nume: Popa
Prenume: Anca
Submit

Click "Afiseaza" fiecare element din formular

Afiseaza

Nume: Popa
Prenume: Anca
Submit

Click "Afiseaza" fiecare element din formular

Afiseaza

Buna ziua Popa Anca



Prelucrare formulare in JavaScript : **validare input**

```
<!DOCTYPE html>
<html>
<body>
<h2>Validare numar din formular cu JavaScript</h2>

<p>Introduceti un numar intre 1 si 10:</p>
<input id="numb">
<button type="button"
onclick="myFunction()">Submit</button>
<script>
function myFunction() {
  var x, text;
  // Get the value of the input field with id="numb"
  x = document.getElementById("numb").value;
  // If x is Not a Number or less than one or greater
  than 10
  if (isNaN(x) || x < 1 || x > 10) {
    text = "Input invalid";
  } else {
    text = "Input OK";
  }
  document.write(text);
}
</script>
</body>
</html>
```

Validare numar din formular cu JavaScript

Introduceti un numar intre 1 si 10:

Submit

Funcția `isNaN()` determina daca o valoare este nr sau nu (Not-a-Number=NaN).



Limbajul JavaScript: Global Properties

Java Script contine proprietatile globale predefinite:

Property	Description
<code>Infinity</code>	A numeric value that represents positive/negative infinity
<code>NaN</code>	"Not-a-Number" value
<code>undefined</code>	Indicates that a variable has not been assigned a value

Exemplu: NaN se utilizeaza cu functia globala `isNaN`



Limbajul JavaScript: Global Functions `isNaN()`

Exemplu: verifica daca urmatoarele valori nu sunt numere(NaN)

```
<!DOCTYPE html>
<html>
<body>
<p>Functia isNaN() returneaza true daca valoarea din
paranteze nu este numar, si false altfel.</p>
<p>Click pe buton pentru a verifica daca o variabila este
numar.</p>
<button onclick="myFunction()">Verifica</button>
<p id="demo"></p>
<script>
function myFunction() {
  var res = "";
  res = res + isNaN(123) + ": 123<br>";
  res = res + isNaN(-1.23) + ": -1.23<br>";
  res = res + isNaN(5-2) + ": 5-2<br>";
  res = res + isNaN(0) + ": 0<br>";
  res = res + isNaN('Hello') + ": 'Hello'<br>";
  res = res + isNaN('2005/12/12') + ": '2005/12/12'<br>";
  res = res + isNaN('NaN') + ": 'NaN'<br>";
  res = res + isNaN(0 / 0) + ": 0 / 0<br>";
  document.getElementById("demo").innerHTML = res;
}
</script>
</body>
</html>
```

Functia `isNaN()` returneaza true daca valoarea din paranteze nu este numar, si false altfel.

Click pe buton pentru a verifica daca o variabila este numar.

```
false: 123
false: -1.23
false: 5-2
false: 0
true: 'Hello'
true: '2005/12/12'
true: 'NaN'
true: 0 / 0
```



Limbajul JavaScript **Global Functions: eval()**

Sintaxa: `eval(string)` unde `string` este o expresie

Exemplu: functia `eval()` calculeaza expresiile `x+y`, `x*y`

```
<!DOCTYPE html>
<html>
<body>
<p>Click pe buton pentru a calcula expresiile:</p>
<p>x=10, y=20</p>
<button onclick="suma()">Calculeaza: x+y</button>
<button onclick="produs()">Calculeaza: x*y</button>
<p id="demo"></p>
<script>
var x = 10;
var y = 20;
function suma() {
var res = eval("x + y");
document.getElementById("demo").innerHTML = res;
}
function produs() {
var res1 = eval("x * y");
document.getElementById("demo").innerHTML = res1;
}
</script>
</body>
</html>
```

Click pe buton pentru a calcula expresiile:
x=10, y=20
Calculeaza: x+y Calculeaza: x*y
30



Limbajul JavaScript: **Evenimente**

Evenimente: actiuni care pot fi detectate de JavaScript.

Se pot configura actiuni la detectarea de evenimente

Exemple de evenimente:

- S-a efectuat click pe un buton
- S-a terminat de incarcata pagina

Exemplu : eveniment "onClick" mouse

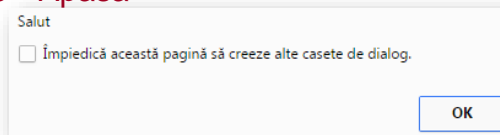
`<form>`

`<input type="button" value="Apasa"`

`onClick="alert('Salut!)" />`

`</form>`

Apasa





Limbajul JavaScript: Evenimente

Evenimente ale ferestrelor

- onload
- onunload - onresize: (Netscape)
- onmove
- onabort
- onerror
- onfocus
- onblur

Evenimente de mouse:

- onmousedown
- onmouseup
- onmousemove
- onmouseover
- onmouseout
- ondblclick
- onclick

Evenimente ale formularelor :

- onsubmit
- onreset
- onchange
- onselect
- onclick
- onblur
- onfocus

Evenimente ale tastelor

- onkeydown
- onkeyup
- onkeypress



Alte exemple JavaScript: schimbare atribut (src) element (img) HTML

```
<!DOCTYPE html>
<html>
<body>

<h1>What Can JavaScript Do?</h1>
<p>JavaScript can change HTML attributes.</p>
<p>In this case JavaScript changes the src (source)
attribute of an image.</p>
<button
onclick="document.getElementById('myImage').src='pic_bulbo
n.gif'">Turn on the light</button>

<button
onclick="document.getElementById('myImage').src='pic_bulbo
ff.gif'">Turn off the light</button>
</body>
</html>
```

What Can JavaScript Do?

JavaScript can change HTML attributes.

In this case JavaScript changes the src (source) attribute of an image.



Turn on the light Turn off the light



Turn on the light

Turn off the light

<https://et.utcluj.ro/PIGMC/JavaScript/image2.html>



Alte exemple JavaScript: Button HTML si eveniment: on click

```
<!DOCTYPE html>
<html>
<body>

<br><br>
<button onclick="document.getElementById('myImg').src='compman.gif'">On</button>
<button onclick="document.getElementById('myImg').src='compman_lowres.gif'">Off</button>
</body>
</html>
```



<https://et.utcluj.ro/PIGMC/JavaScript/image1.html>



Alte exemple JavaScript: Eveniment HTML: Mouse event

```
<!DOCTYPE html>
<html>
<body>

<h1 onmouseover="style.color='red'"
onmouseout="style.color='black'">Mouse over
this text</h1>

</body>
</html>
```

Mouse over this text

<https://et.utcluj.ro/PIGMC/JavaScript/mouseovertext.html>



Alte exemple JavaScript: SO pe care ruleaza Browser-ul: Obiect Navigator Navigator.platform

```
<!DOCTYPE html>
<html>
<body>

<h2>The Navigator Object</h2>

<p>The platform property returns the
browser platform (operating system):
</p>

<p id="demo"></p>

<script>
document.write(
"navigator.platform is " +
navigator.platform);
</script>

</body>
</html>
```

The Navigator Object

The platform property returns the browser platform (operating system):

navigator.platform is Win32



Test JavaScript:

Cu ce tag se introduce cod JavaScript?

- a) <js>
- b) <scripting>
- c) <script>
- d) <javascript>

c) - corect



Test JavaScript:

Cum se afiseaza "Hello World" intr-o fereastră nouă?

- a) `msgBox("Hello World");`
- b) `alertBox("Hello World");`
- c) `alert("Hello World");`
- d) `msg("Hello World");`

c) - corect



Test JavaScript:

Cum se definește corect un șir (array) de culori în JavaScript?

- a) `var colors = "red", "green", "blue"`
- b) `var colors = 1 = ("red"), 2 = ("green"), 3 = ("blue")`
- c) `var colors = (1:"red", 2:"green", 3:"blue")`
- d) `var colors = ["red", "green", "blue"]`

d) - corect



Test JavaScript:

Cum se determina maximul dintre x si y?

- a) ceil(x, y)
- b) Math.max(x, y)
- c) top(x, y)
- d) Math.ceil(x, y)

b) - corect



Validator sintaxa JavaScript:

Esprima Demo Project Documentation

Syntax Validator checks for mistakes and errors

```
1 |
2 | document.getElementById("demo").innerHTML =
3 | "Number of links: " + document.links.length;
4 |
```

Unlike a typical code linter, this syntax validator does **not** care about coding styles and formatting.

If there is a syntax error, the sign ⚠ will be shown in the left-side gutter. Placing the mouse cursor over that sign will reveal the complete error description.

For a command-line usage, check [esvalidate](#) from [Esprima package](#) (for Node.js). There is also a plugin for [Grunt](#) called [grunt-jsvalidate](#). Ant users can take a look at an exemplary [Ant task](#) for syntax validation.

Code is syntactically valid.



Validator sintaxa JavaScript:

BeautifyTools

Home All Tools Samples Contact Us

Javascript Validator

Use Online Javascript Validator to validate javascript code and find errors and warnings of your code that can be fixed. See real time code validation and fix errors and warnings of your code as you type. After coding is complete, format your javascript code.

```
1 document.getElementById("demo").innerHTML =
2 "Number of links: " + document.links.length;
```

Line	Col	Errors
		No syntax errors!

Validate Code Options Format Code Clear



Language	Learners	Lessons	Quizzes
C++	3,905,264	80	324
Java	2,859,896	65	140
Python 3	2,774,052	92	275
JavaScript	1,636,228	51	147
PHP	959,603	46	99
C#	954,689	69	199