Introducere în JavaScript

Ce este JavaScript?

- JavaScript a fost proiectat pentru a adăuga interactivitate paginilor HTML
- JavaScript este un limbaj pentru scripturi. Un limbaj pentru scripturi este un limbaj de programare simplificat
- JavaScript este, în general, înglobat direct în paginile HTML
- JavaScript este un limbaj interpretat (adică scriptul este executat direct, fără compilare prealabilă)
- JavaScript poate fi folosit fără licenţă

Ce poate face JavaScript?

- **JavaScript oferă proiectanţilor HTML un instrument de programare –** în general proiectanţii paginilor HTML nu au cunostinte mari legate de programare, dar JavaScript este un limbaj cu o sintaxă foarte simplă şi multi utilizataori pot insera mici secvenţe de cod în paginile HTML
- **JavaScript poate insera în mod dinamic text într-o pagină HTML –** O instrucţiune JavaScript ca aceasta: **document.write("<h1>" + name + "</h1>")** poate scrie un text variabil în pagina HTML
- **JavaScript poate reacţiona la evenimente –** Un cod JavaScript poate fi proiectat să se execute când se întâmplă ceva, spre exemplu când pagina s-a încărcat complet sau utilizatorul acţionează un element HTML
- **JavaScript poate citi şi scrie elementele HTML –** Un cod JavaScript poate citi şi modifica conţinutul unui element HTML
- **JavaScript poate fi folosit pentru a valida datele –** Un cod JavaScript poate fi folosit pentru a valida datele înainte de a fi trimise către server. În acest fel serverul nu mai face procesări suplimentare.
- **JavaScript poate fi folosit pentru a detecta browserul utilizatorului** – Un cod JavaScript poate detecta tipul browserului şi poate încărca o pagină proiectată special pentru tipul respectiv de browser
- **JavaScript poate fi folosit pentru a crea cookies** – Un cod JavaScript poate fi utilizat pentru a stoca şi extrage informaţii pe calculatorul vizitatorului paginii HTML

Cum se inserează JavaScript într-o pagină HTML

Pentru a insera JavaScript într-o pagină HTML se utilizează tagul **<script>**.
În exemplul următor, JavaScript este utilizat pentru a scrie un text într-o pagină web:

```
<html>
<body>
<h3>Afisarea unui mesaj cu JavaScript</h3> <hr/>
<script type="text/javascript">
document.write("Bine ati venit!");
</script>
</body>
</html>
```

Exemplul următor ilustrează cum pot fi adăugate taguri HTML pentru a formata textul afişat cu JavaScript:

```
<html>
<body>
<h3>Utilizarea tagurilor HTML in mesajul afisat cu
JS</h3> <hr/>
<script type="text/javascript">
document.write("<h1>Bine ati venit!</h1>");
</script>
</body>
</html>
```

Explicaţii:
Pentru a insera JavaScript într-o pagină HTML, folosim tagul **<script>** şi în interiorul acestui tag folosim atributul type pentru a defini limbajul în care este scris scriptul. Astfel, tagurile **<script type="text/javascript">** şi **</script>** marchează locul în care începe, respectiv se sfârşeşte scriptul:

```
<html>
<body>
<script type="text/javascript">
...
</script>
</body>
</html>
```

Comanda **document.write** reprezintă modalitatea JS standard pentru a scrie un text într-o pagină. Deoarece această comandă este inclusă între tagurile **<script>** şi **</script>**, browserul o va recunoaşte drept comandă JS şi va executa respectiva linie de cod. Pentru exemplul considerat, browserul va scrie în pagină textul **Bine ati venit!**

Obs: Dacă comanda **document.write** nu este inclusă între tagurile de script, browserul o va interpreta ca text obişnuit şi va afişa pe ecran linia de cod.

Inserarea scripturilor JS

Dacă scriptul este inclus în secţiunea **body**, el va fi executat cât timp se încarcă pagina. Dacă scriptul este inclus în secţiunea **head**, el va fi executat numai când este apelat.

Scripturi în <head>
Scripturile care trebuie executate când sunt apelate sau când are loc un eveniment, trebuie scrise în secţiunea **head**. În acest fel, scriptul va fi sigur încărcat înainte de a fi utilizat.

```
<html>
<head>
<script type="text/javascript">
function message()
{
alert("Aceasta caseta de alertare este apelata si afisata cand are loc evenimentul onload");
}
</script>
</head>
<body onload="message()">
<h3>Casetele de alertare</h3> <hr/>
</body>
</html>
```

Scripturi în <body>
Scripturile care trebuie executate când pagina se încarcă trebuie scrise în secţiunea body şi vor genera conţinutul paginii:

```
<html>
<head>
</head>
<body>
<h3>Afisarea textului cu JavaScript</h3> <hr/>
<script type="text/javascript">
document.write("Acest mesaj este scris cu JavaScript");
</script>
</body>
</html>
```

Scripturi în <head> şi <body>
Puteţi include un număr nelimitat de scripturi JS în document, deci puteţi avea scripturi şi în **head** şi în **body**:

```
<html>
<head>
<script type="text/javascript">
....
```

```
</script>
</head>
<body>
<script type="text/javascript">
....
</script>
</body>
```

Folosirea unui script extern

Dacă doriţi să utilizaţi acelaşi script în mai multe pagini web fără a rescrie codul, trebuie să scrieţi scriptul JS într-un fişier extern. Fişierul trebuie să aibă extensia **.js** şi nu poate conţine tagul **<script>**. Pentru a utiliza fişierul extern, trebuie sa il includeţi în atributul **src** al tagului **<script>**:

```
<html>
<head>
<script type="text/javascript" src="…….js"></script>
</head>
<body>
</body>
</html>
```

Obs: Scriptul trebuie plasat în locul în care ar fi fost scris în mod normal.

Instrucţiuni JavaScript

JavaScript este o secvenţă de declaraţii, instrucţiuni şi comenzi care vor fi executate de către browser. Spre deosebire de HTML, Java Script este casesensitive, deci aveţi grijă când scrieţi instrucţiuni, declaraţi variabile sau apelaţi funcţii. O instrucţiune JavaScript este o comandă către browser şi are rolul de a spune browserului ce trebuie să facă. Următoarea instrucţiune JS transmite browserului să scrie în pagină textul "Buna ziua":

```
document.write("Buna ziua");
```

Fiecare instrucţiune se încheie cu punct şi virgulă (;).

Codul JavaScript este o secvenţă de instrucţiuni JS. Fiecare instrucţiune este executată de browser în ordinea în care a fost scrisă.

Exemplul următor va scrie un titlu şi două paragrafe într-o pagină web:

```
<html>
<body>
<h3>Utilizarea tagurilor HTML in mesajele afisate cu JS</h3> <hr/>
```

```
<script type="text/javascript">
document.write("<h1>Acesta este un titlu</h1>");
document.write("<p>Acesta este un paragraf.</p>");
document.write("<p>Acesta este un alt paragraf.</p>");
</script>
</body>
</html>
```

Blocuri JavaScript

Instrucţiunile JavaScript pot fi grupate în blocuri care se scriu între acolade.

Instrucţiunile dintr-un bloc vor fi executate împreună.

În acest exemplu, instrucţiunile care scriu un titlu şi două paragrafe, au fost grupate împreună într-un bloc:

```
<html>
<body>
<script type="text/javascript">
{
document.write("<h1>Acesta este un titlu</h1>");
document.write("<p>Acesta este un paragraf.</p>");
document.write("<p>Acesta este un alt paragraf.</p>");
}
</script>
</body>
</html>
```

În mod normal, un bloc este folosit pentru a grupa un grup de instrucţiuni într-o funcţie sau într-o condiţie (blocul va fi executat dacă o anumită condiţie este satisfăcută).

Variabile JavaScript

În JS, variabilele sunt folosite pentru a păstra valori sau expresii. O variabilă poate avea un nume scurt, de exemplu x, sau mai descriptiv, de exemplu prenume.

Reguli pentru numele variabilelor JavaScript:

- numele este case-sensitive (y şi Y sunt două variabile diferite)
- numele trebuie să înceapă cu o literă sau cu liniuţa de subliniere (underscore)

Exemplu

Valoarea unei variabile se poate modifica în timpul execuţiei scriptului. Puteţi referi variabila prin nume pentru a-i afişa sau modifica conţinutul, ca în exemplul următor:

```
<html>
<body>
<h3>Declararea, initializarea, atribuirea si afisarea unei variabile</h3> <hr/>
<script type="text/javascript">
var prenume;
prenume="Mihai";
document.write("<b>Numele variabilei</b>: prenume");
document.write("<br/>");
document.write("<b>Valoare initiala</b>: "+prenume);
document.write("<br/>");
prenume="Adrian";
document.write("<b>Valoare dupa atribuire</b>: "+prenume);
</script>
</body>
</html>
```

Declararea variabilelor JavaScript

Puteţi crea variabile cu sintaxa:

var *nume_variabila;*

După declarare, variabila nu conţine valori (este vidă). Puteţi să iniţializaţi o variabilă chiar în momentul declarării:

var x=8;

var prenume="Matei";

Obs: Când atribuiţi unei variabile o valoare de tip text, textul trebuie scris între ghilimele.

Dacă atribuiţi valori unei variabile care nu a fost încă declarată, ea va fi declarată automat.

Declaraţiile:

x=8;

prenume="Matei";

au acelaşi efect cu declaraţiile:

var x=8;

var prenume="Matei";

Dacă redeclaraţi o variabilă JavaScript, ea va păstra valoarea iniţială:

var x=7;

var x;

După execuţia instrucţiunilor de mai sus, variabila x are valoarea 7 care nu a fost resetată la redeclarare.

Operatorii JavaScript

Operatorii aritmetici

Sunt folosiţi pentru a efectua operaţii aritmetice cu variabile şi/sau valori.

+ adunare

- scădere

* înmulţire

/ împărţire

% modulo (restul împărţirii întregi)

++ incrementare

-- decrementare

Operatorii de atribuire

Sunt folosiţi pentru a atribui valori variabilelor JavaScript.

=, += , -=, *= , /= , %=

Operatorul + utilizat pentru şiruri de caractere

Acest operator poate fi utilizat şi pentru a concatena variabile tip şir de caractere (string sau text).

Exemplu:

t1="Ce mai";

t2="faci azi?";

t3=t1+t2;

După execuţie, variabila t3 conţine şirul „Ce maifaci azi?”.

Adunarea şirurilor şi a numerelor

Regulă: Dacă se aduna un număr cu un şir de caractere, se va obţine un şir de caractere.

```
<html>
<body>
<h3>Adunarea sirurilor de caractere si a numerelor cu siruri de caractere</h3> <hr/>
<script type="text/javascript">
x=6+7;
document.write("6+7="+x);
document.write("<br />");
x="6"+"7";
document.write('"6"+"7"='+x);
document.write("<br />");
x=6+"7";
document.write('6+"7"='+x);
document.write("<br />");
x="6"+7;
document.write('"6"+7='+x);
document.write("<br />");
</script>
</body>
</html>
```

Operatorii de comparare şi operatorii logici

Operatorii de comparare sunt utilizaţi în construcţii logice pentru a verifica egalitatea sau diferenţa dintre două variabile sau valori.

== egal

=== este egal exact (valoare şi tip). Exemplu: x==="5" este fals

!= Diferit

> Mai mare decat

< Mai mic decât

>= Mai mare sau egal

<= Mai mic sau egal

Operatorii logici sunt utilizaţi pentru a determina relaţia logică dintre variabile sau valori.

&& şi (x < 10 && y > 1) este adevărat

|| sau (x==5 || y==5) este fals

! not !(x==y) este adevărat

Operatorul condiţional

Acest operator atribuie o valoare unei variabile în funcţie de o anumită condiţie.

Sintaxă:

variabila=(conditie)?valoare1:valoare2

Exemplu:

salut=(visitor=="FEM")?"Doamna ":"Domnul";

Instrucţiunile condiţionale

Adesea, când scrieţi cod JS, trebuie să realizaţi operaţii diferite în funcţie de decizii diferite. Pentru a realiza acest lucru, folosiţi în cod instrucţiunile condiţionale.

În JavaScript există următoarele instrucţiuni condiţionale:

if, if...else, switch

Exemplul 1:

```
<html>
<body>
<h3>Scriptul va afisa un mesaj daca ora<10 folosind instructiunea if</h3> <hr/>
<script type="text/javascript">
var d = new Date();
var time = d.getHours();
if (time < 10)
{
document.write("<b>Buna dimineata</b>");
}
</script>
</body>
```

```
</html>
```

Exemplul 2:

```
<html>
<body>
<h3>Scriptul va afisa un mesaj sau altul in functie de ora, cu instructiunea if..else</h3> <hr/>
<script type="text/javascript">
var d = new Date();
var time = d.getHours();
if (time < 10)
{
document.write("<b>Buna dimineata</b>");
}
else
{
document.write("<b>Buna ziua</b>");
}
</script>
</body>
</html>
</html>
```

Exemplul 3:

```
<html>
<body>
<h3>Scriptul va afisa unul din trei mesaje in functie de ora, cu instructiunea if-else-if-else</h3> <hr/>
<script type="text/javascript">
var d = new Date();
var time = d.getHours();
if (time<10)
{
document.write("<b>Buna dimineata</b>");
}
else if (time>=10 && time<17)
{
document.write("<b>Buna ziua</b>");
}
else
{
document.write("<b>Buna seara</b>");
}
</script>
</body>
</html>
```

Exemplul 4:

Link-ul din exemplul următor va deschide Google sau Yahoo.

```
<html>
<body>
<h3>Scriptul afiseaza in mod aleator unul din doua
link-uri, folosind if..else</h3> <hr/>
<script type="text/javascript">
var r=Math.random();
if (r>0.5)
{
document.write("<a href='http://www.google.com'>Google!</a>");
}
else
{
document.write("<a href='http://www.yahoo.com’>Yahoo!</a>");
}
</script>
</body>
</html>
```

Exemplul 5:

```
<html>
<body>
<h3>Scriptul utilizeaza instructiunea switch</h3>
<hr/>
<script type="text/javascript">
var d = new Date();
theDay=d.getDay();
switch (theDay)
{
case 5:
document.write("<b>Vineri</b>");
break;
case 6:
document.write("<b>Sambata</b>");
break;
case 0:
document.write("<b>Duminica</b>");
break;
default:
document.write("<b>Astept weekend-ul!</b>");
}
</script>
</body>
```

```
</html>
```

Casete popup

JavaScript are trei tipuri de casete popup: caseta Alert, caseta Confirm şi caseta Prompt.

Caseta Alert

O casetă de alertă se utilizează atunci doriţi să fiţi siguri că o anumită informaţie ajunge în atenţia utilizatorului. Când o casetă de alertă este afişată, utilizatorul va trebui să acţioneze butonul "OK" pentru a putea continua.

Sintaxă:

alert("...un text....");

Exemplu:

```
<html>
<head>
<script type="text/javascript">
function afiseaza_alert()
{
alert("Sunt o caseta de alertare!");
}
</script>
</head>
<body>
<h3>La apasarea butonului va fi apelata o functie care afiseaza caseta alert</h3> <hr/>
<input type="button" onclick="afiseaza_alert()" value="Apasa" />
</body>
</html>
```

Caseta Confirm

O casetă de confirmare se utilizează atunci când doriţi ca utilizatorul să verifice sau să accepte ceva. Când caseta de confirmare este afişată, utilizatorul va trebui să acţioneze butonul "OK" sau butonul "Cancel" pentru a putea continua. Dacă utilizatorul acţionează butonul "OK", caseta returnează valoarea true, dacă acţionează butonul "Cancel", caseta returnează valoarea false.

Sintaxă:

confirm("....un text....");

Exemplu:

```
<html>
<head>
<script type="text/javascript">
function afiseaza_confirm()
{
var r=confirm("Apasati un buton");
if (r==true)
{
document.write("Ati apasat butonul OK!");
}
else
{
document.write("Ati apasat butonul Cancel!");
}
}
</script>
</head>
<body>
<h3>La apasarea butonului va fi apelata o functie care afiseaza caseta confirm si verifica ce buton ati apasat</h3> <hr/>
<input type="button" onclick="afiseaza_confirm()"
value="Apasa" />
</body>
</html>
```

Caseta Prompt

Această casetă se utilizează atunci când doriţi ca utilizatorul să introducă o anumită valoare înainte de a accesa pagina. Când caseta prompt este afişată, utilizatorul va trebui să acţioneze butonul "OK" sau butonul "Cancel" pentru a putea continua după ce introduce valoarea solicitată. Dacă utilizatorul acţionează butonul "OK", caseta returnează valoarea true, dacă acţionează butonul "Cancel", caseta returnează valoarea false.

Sintaxă:

prompt("....un text....","valoare_implicita");

Exemplu:

```
<html>
<head>
<script type="text/javascript">
function afiseaza_prompt()
{
var name=prompt("Va rog sa va introduceti numele","");
if (name!=null && name!="")
{
document.write("Buna ziua " + name + "! Ce mai
faci?");
}
}
</script>
</head>
<body>
<h3>La apasarea butonului va fi apelata o functie care afiseaza caseta prompt</h3> <hr/>
<input type="button" onclick="afiseaza_prompt()"
value="Apasa" />
</body>
</html>
```

Obs. Dacă doriţi ca textul dintr-o casetă să fie afişat pe mai multe linii, trebuie procedat ca în exemplul următor:

```
<html>
<head>
<script type="text/javascript">
function afiseaza_alert()
{
alert("Buna! Asa se adauga" + '\n' + "o intrerupere de linie" + '\n' + "intr-o caseta de alertare!");
}
</script>
</head>
<body>
<h3>Caseta alert cu textul scris pe mai multe linii</h3> <hr/>
<input type="button" onclick="afiseaza_alert()" value="Apasa" />
</body>
</html>
```

Funcţii

O funcţie va fi executată când are loc un eveniment sau când este apelată. Dacă doriţi ca browserul să nu execute un script atunci când pagina se încarcă, puteţi scrie scriptul într-o

funcţie. O funcţie poate fi apelată din orice punct al paginii, sau chiar din alte pagini, dacă funcţia este scris într-un fişier JS extern.

Funcţiile JS pot fi scrise în secţiunea <head> sau în secţiunea <body> a documentului. Totuşi, pentru a fi siguri că funcţia este încărcată în browser înainte de a fi apelată, este recomandat să o scrieţi în secţiunea <head>.

Definirea unei funcţii

Sintaxă:

function *nume_functie*(*var1,var2,...,varX*)

{

codul functiei

}

Parametrii var1, var2, etc. sunt variabile sau valori transmise funcţiei. Acoladele marchează începutul şi sfârşitul corpului funcţiei.

Exemplu:

```
<html>
<head>

<script type="text/javascript">
function afiseaza_mesaj()
{
alert("Bine ati venit!");
}
</script>

</head>
<body>
<h3>La apasarea butonului este apelata o functie JS care afiseaza caseta alert</h3> <hr/>
<form>
<input type="button" value="Apasati!" onclick="afiseaza_mesaj()" />
</form>
</body>
</html>
```

Dacă linia de cod alert("Bine ati venit!") din exemplul anterior nu ar fi fost scrisă în corpul unei funcţii, codul ar fi fost executat imediat ce linia respectivă ar fi fost încărcată în browser.

Deoarece codul a fost inclus într-o funcţie, el nu va fi executat decât atunci când utilizatorul acţionează butonul şi este apelată funcţia afiseaza_mesaj().

Instrucţiunea return

Instrucţiunea return este folosită pentru a specifica valoarea returnată de o funcţie şi trebuie inclusă în orice funcţie care returnează o valoare.

În exemplul următor, funcţia suma returnează suma celor doi parametri de intrare:

```
<html>
<head>
<script type="text/javascript">
function suma(a,b)
{
return a+b;
}
</script>
</head>
<body>
<h3>Suma urmatoare este calculata si returnata de o functie</h3> <hr/>
<script type="text/javascript">
document.write("7+9="+suma(7,9));
</script>
</body>
</html>
```

Durata de viaţă a variabilelor JavaScript

Dacă declaraţi o variabilă în interiorul unei funcţii, ea poate fi accesată numai din interiorul funcţiei. Când funcţia se încheie, variabila este distrusă.

Variabilele declarate în corpul unei funcţii se numesc variabile locale. Puteţi avea variabile locale cu acelaşi nume în funcţii diferite, deoarece fiecare variabilă locală este recunoscută numai în interiorul funcţiei în care este declarată. Dacă declaraţi o variabilă în afara tuturor funcţiilor (variabilă globală), ea poate fi accesată de toate funcţiile din pagină. O variabilă global este distrusă numai atunci când pagina este închisă.

Exemplul 1

Ilustrează cum se poate transmite o variabilă unei funcţii şi cum poate fi folosită respectiva variabilă în corpul funcţiei.

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

```
function functia_1(text)
{
alert(text);
}
</script>
</head>
<body>
<h3>Functii JavaScript</h3> <hr/>
<form>
<input type="button" onclick="functia_1('Bune ati venit!')" value="Apasati">
</form>
<p>Cand apasati butonul, va fi apelata o functie cu textul "Bine ati venit!" drept parametru. Functia va afisa parametrul cu o caseta de alertare.</p>
</body>
</html>
```

Exemplul 2

Ilustrează cum poate fi folosit rezultatul returnat de o funcţie.

```
<html>
<head>
<script type="text/javascript">
function functie_2()
{
return ("Bine ati venit!");
}
</script>
</head>
<body>
<h3>Textul urmator este returnat de o functie apelata direct din document.write()</h3> <hr/>
<script type="text/javascript">
document.write(functie_2())
</script>
</body>
</html>
```

Exemplul 3

```
<html>
<head>
<script type="text/javascript">
function salut(txt)
{
alert(txt);
}
```

```
</script>
</head>
<body>
<h3>Utilizarea functiilor JavaScript</h3> <hr/>
<form>
<input type="button"
onclick="salut('Buna dimineata!')"
value="Dimineata">
<input type="button" onclick="salut('Buna seara!')" value="Seara">
</form>
<p>
Cand apasati unul dintre butoane, va fi apelata o functie care afiseaza mesajul primit ca
parametru.</p>
</body>
</html>
```

Instrucţiunea for

Instrucţiunile repetitive sunt utilizate pentru a executa o secvenţă de cod în mod repetat. În JS sunt două tipuri diferite de instrucţiuni repetitive:

for

while si *do ... while*

Sintaxă:

for(var=val_initiala;var<=val_finala;var=var+increment)

{

codul ce trebuie executat

}

Instrucţiunea while

Sintaxă:

while (*propozitielogica*)

{

codul ce trebuie executat

}

Instrucţiunea do...while

Este o variantă a instrucţiunii while. Secvenţa de instrucţiuni va fi executată în mod sigur o dată, apoi în mod repetat, cât timp condiţia specificată este adevărată.

Sintaxă:

do

{

codul ce trebuie executat

}

while (*propozitielogica*);

Instrucţiunea for...in

Această instrucţiune este utilizată pentru a parcurge elementele unui tablou sau a enumera proprietăţile unui obiect.

Sintaxă:

for (*variabila* in *obiect*)

{

cod ce trebuie executat

}

Obs: Codul din corpul instrucţiunii este executat câte o dată pentru fiecare element din tablou sau proprietate.

Obs: Argumentul variabila poate fi o variabilă, un element de tablou sau o proprietate a unui obiect.

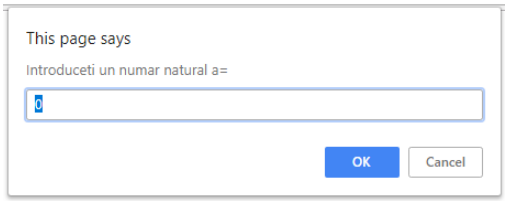
Exemplu

Instrucţiunea for..in este utilizată pentru a parcurge elementele unui tablou:

```
<html>
<body>
<h3>Parcurgerea elementelor unui tablou cu instructiunea for..in</h3> <hr/>
<script type="text/javascript">
var x;
var pets = new Array();
pets[0] = "Pisica";
pets[1] = "Caine";
pets[2] = "Papagal";
pets[3] = "Hamster";
document.write("Valorile memorate in tablou sunt:"+"<br/>");
for (x in pets)
{
document.write(pets[x] + "<br />");
}
</script>
</body>
</html>
```

Citirea si afisarea datelor in JavaScript

Citirea datelor o vom face cu functia prompt, iar afisarea cu functia alert. Ambele functii deschid o fereastra pentru acesta operatie.

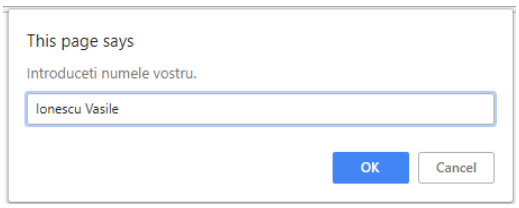


Exemplul 1. Suma a doua numere.

```
<html>

<head>

</head>

<body>

<script type = "text/javascript">

a = parseInt(prompt("Introduceti un numar natural a =",0));

b = parseInt(prompt("Introduceti un numar natural b =",0));

s = a + b;

alert("suma = " + s);
```

```
</script>

</body>

</html>
```

Exemplul 2. Citirea numelui si afisarea lui impreuna cu un text.




```
<html>

<head>

<script type = "text/javascript">


var nume = prompt("Introduceti numele vostru! ",'');

if(nume == '' || nume == null)

        nume = "student";


</script>


</head>


<body>

<script type = "text/javascript">


document.write("<center><h2>Universitatea din Pitesti, " + nume + " ! </h2> </center>");


</script>


</body>
```

```
</html>
```

Exemplul 3. Citirea si afisarea unui vector.

```
<html>
<head>
<script type = "text/javascript">
n = parseInt(prompt("Introduceti numarul de componente al vectorului!",0));
x = new Array();
for(i=1;i<=n;i++)
        x[i] = parseInt(prompt("Introduceti a - " +i+ " -a componenta ",0));
s ="";
for(i=1;i<=n;i++)
        s += x[i] + " ";
alert("Elementele vectorului sunt: " + s);
</script>
<head>


</html>
```

Probleme propuse

1. Cititi trei numere si afisati suma si produsul lor.
2. Cititi a, b numere naturale si rezolvati ecuatia ax+b=0.
3. Cititi a, b, c numere natural nenule si rezolvati ecuatia $ax^2+bx+c=0$.
4. Cititi n numere naturale si afisati aceste numere in ordine crescatoare, respectiv descrescatoare.
5. Cititi numele si prenumele unei presoane si apoi afisati separate numele, respective prenumele pe randuri diferite.
6. Cititi cnp-ul unei persoane si afisati data nasterii, urmata de varsata sa.

7. Cititi n cuvinte si afisati cuvintele de lungime maxima.
8. Cititi un text in care cuvintele sunt separate prin cate un spatiu. Apoi afisati primul si ultimul cuvant.
9. Cititi un text in care cuvintele sunt separate prin cate un spatiu. Apoi afisati cuvintele pe randuri diferite.

Bibliografie

https://www.w3schools.com

http://www.ls-infomat.ro

http://www.e-learn.ro/tutorial/javascript