



Laborator 6

Metode de implementare a unei interfețe web grafice- utilizand JavaScript



Limbajul JavaScript

JavaScript: limbaj de scripting client-side (rulează la nivelul browser-ului)

Avantaje :

- dezvoltat pentru a prelucra informațiile din formulare și a adăuga dinamism paginilor web
- este interpretat de browser și este încorporat în paginile HTML
- dezvoltat de Netscape , denumire inițială : LiveScript
- nu e Java , sintaxa similară cu limbajului C/Java
- rulează pe orice browser: Firefox, Opera, Netscape Navigator, Internet Explorer, Safari, etc.



Caracteristici JavaScript

- ofera posibilitatea de a executa scripturi la nivelul browserului
- poate reacționa la evenimente – (ex. utilizatorul a făcut clic pe un element HTML)
- poate citi și scrie elemente HTML – poate citi și schimba conținutul unui element HTML
- poate fi utilizat pentru a valida datele din formulare înainte de a fi trimise către server
- Poate fi utilizat pentru a detecta tipul browserului - în funcție de acesta putem încărca o pagină sau alta
- JavaScript poate fi utilizat pentru a manipula cookie-uri



Inserarea codului JavaScript în pagină

- Prin tag-ul `<script>`

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

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



Inserarea codului JavaScript in pagina

Exemplu:

```
1. <html>
2. <head>
3. <title>Hello</title>
4. </head>
5. <body>
6. <script type="text/javascript">
7.   document.write("<h1>Hello World! </h1>");
8. </script>
9. </body>
10. </html>
```



Inserarea codului JavaScript in pagina

Exemplu: browser-ul nu poate interpreta Javascript

```
1. <html>
2. <head>
3. <title>Hello</title>
4. </head>
5. <body>
6. <script type="text/javascript">
7.   <!--
8.   document.write("<h1>Hello World!</h1>");
9.   //-->
10. </script>
11. <noscript>Limbajul JavaScript nu e disponibil</noscript>
12. </body>
13. </html>
```



Inserarea codului JavaScript in pagina

Reguli: JavaScript se poate insera

a) **Intern** in documentul HTML

- atât <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 încărcată pagina

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

Avantaj: fișierul poate fi inclus în mai multe pagini HTML

Ex. Incluziune în document HTML în secțiunea <head>:

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



Limbajul JavaScript: **Variabile**

Sintaxa :

```
var idVar;  
var idVar = val_iniciala;  
idVar = val_iniciala;
```

- Numele variabilelor sunt case-sensitive și trebuie să înceapă cu o literă
- Variabilele nu au tip (ele pot reține orice valoare)
- Cuvântul var este opțional (dacă nu se specifică automat variabila este considerată globală)
- Variabilele declarate într-o funcție sunt locale acelei funcții
- La declararea variabilelor locale trebuie obligatoriu var n
Variabilele declarate în afara oricărei funcții sunt globale (sunt accesibile oriunde în pagina)



Limbajul JavaScript: Tipuri de baza

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

- Numerele** sunt memorate întotdeauna în virgulă flotantă
 - Numerele **hexazecimale** încep cu 0x
 - Numerele în **baza 8** încep cu 0 (nu toate browserele suportă)
- Sirurile (string)**: secvențe de caractere cuprinse " " sau "
 - pot conține \n (newline), \" (ghilimele), etc.
- Valorile logice (boolean)**: true sau false



Limbajul JavaScript: Operatori

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



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: Instrucțiuni ciclare

- ❑ Instrucțiunea for

for (expr_i;conditie; expresie_reinitializare) instructiune

Exemplu

```
<script type="text/javascript">
var i=0, s=0;
for (i=0;i<=5;i++)
    { s+=i;
    }
document.write(" Suma este " + s);
```



Limbajul JavaScript: Instrucțiuni ciclare

- ❑ Instrucțiunea for ...in

for (variabila in object) { instructiuni }

Exemplu

```
<script type="text/javascript">
var persoana={nume:"Popa", prenume:"Mihai", varsta:25};
for (x in persoana)
    { document.write(persoana[x] + " ");
    }
</script
```



Limbajul JavaScript: Instrucțiuni ciclare

- ❑ Instrucțiunea while

while (condiție) instrucțiune

Exemplu

```
<script type="text/javascript">
var i=0, s=0;
while (i<=5)
    { s+=i;
      i++ }
document.write(" Suma este " + s);
</script>
```



Limbajul JavaScript: Instrucțiuni ciclare

- ❑ Instrucțiunea do ... while

```
do
    { instrucțiuni }
while (condiție);
```

Exemplu

```
<script type="text/javascript">
var i=0, s=0;
do {
    s+=i;
    i++ }
while (i<=5);
document.write(" Suma este " + s);
```




Limbajul JavaScript: Funcții

❑ se definesc de regula în <head>

❑ Sintaxa definiției funcției:

```
function numeFuncție(p1, ..., pN)
    { //declarații de variabile locale (var)
      instrucțiuni }

```

❑ O funcție poate returna o valoare cu `return valoare;`

❑ Sintaxa apelului funcției

```
numeFuncție(vp1, ..., vpN)
```

❑ Parametri simpli sunt transmiși prin valoare, obiectele prin referință



Limbajul JavaScript: Funcții

Exemplu:

```
<html>
<head>
  <title>Instrucțiuni</title>
  <script type="text/javascript">
    function factorial(n){
      var i, p=1;
      for (i=1;i<=n;i++) {
        p*=i;
      }
      return p;
    }
  </script>
</head>
  <body>
    <h1>Funcții</h1>
    <script type="text/javascript">
      var n=3;
      document.write( n + "!=" +
        factorial(n));
    </script>
  </body>
</html>
```



Limbajul JavaScript: **Tablouri**

Exemplu:

```
var masini= new Array();
masini[0]="Audi";
masini[1]="Logan";
var masini= new Array("Audi", "Logan"); //tablou condensat
var masini= ["Audi", "Logan"]; //tablou de literali
var masini= ["Audi", "Logan", "Ford"]; //tablou de literali
masini.sort();
var i=0;
for(i=0;i<masini.length;i++){
    document.write(masini[i]);
}
masini.reverse();
document.write(masini.toString());
```



Limbajul JavaScript: **Obiecte**

Creare obiecte JavaScript:

Utilizând obiecte literal:

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

Cream un obiect "gol" cu ajutorul operatorului new, apoi adaugăm proprietăți:

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

Cu ajutorul constructorilor:

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



Exemple JavaScript: creare ferestre

Exemplu : Confirm

```
<script type="text/javascript">
  intrebare = window.confirm("Rezultatul lui 0+0 este 0?");
  if (intrebare) alert("Corect");
  else alert("Incorect");
</script>
```



Exemple JavaScript: creare ferestre

Exemplu : interfata calculator

```
script type="text/javascript">
function calculeaza() {
  form.rezultat.value = eval(form.expresie.value);
}
eval("alert('Rezultatul este: " + form.rezultat.value + "')");
</script>
<form>
  <p>Introduceti o expresie matematica (adunare, scadere, inmultire, impartire), de exemplu (7*8 sau 3+8/2): </p>
  <input type="text" name="expresie" size="35" >
  <input type="button" name="calc" value="Calculare"
  onclick="calculeaza(this.form)">
  <br>
  Rezultatul este:
  <input type="text" name="rezultat" size="18">
</form>
```



Exemple JavaScript: creare ferestre

Exemplu : Alert

```
<script type="text/javascript">  
window.alert("Bine ai venit");  
</script>
```



Exemplu : Confirm

```
<script type="text/javascript">  
intrebare = window.confirm("Rezultatul lui 0+0  
este 0?");  
if (intrebare) alert("Corect");  
else alert("Incorect");  
//-->  
</script>
```



Limbajul JavaScript: Evenimente

Evenimente: acțiuni care pot fi detectate de JavaScript.

Se pot configura acțiuni la detectarea de evenimente

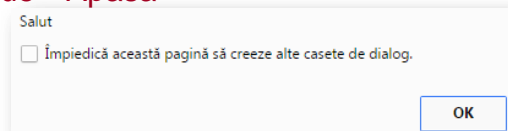
Exemple de evenimente:

- S-a efectuat click pe un buton
- S-a terminat de încărcat 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

```
<!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



Aplicații utilizând Java Script

Implementare cod JavaScript : nu necesită mediu specific de dezvoltare și compilare, se integrează în fișierele HTML.

Instrumente implementare interfețe grafice utilizând JavaScript:

Notepad++, Brackets, Sublime sau programe specifice: Adobe Dreamweaver

Testare și depanare cod: se realizează din browser web



Aplicații utilizând Java Script

Aplicația 1: Implementați o interfață grafică în care creați o fereastră de tip alert, confirm sau prompt

Aplicația 2: Creați o interfață în care:

- inserați un text cărui îi asociați un eveniment onmouseover care va schimba culoarea textului,
- o imagine și un buton, în care imaginea se schimbă la click pe buton
- Inserați o variabilă șir și afișați șirul ordonat alfabetic



Aplicatii utilizand Java Script

Aplicatia 2: Implementati o interfata grafica sub forma unui formular web utilizand HTML5 cu urmatoarele campuri : nume, prenume, adresa, telefon, email si 1 buton Trimite (submit). Implementati in Java Script functionalitatea ca la apasarea butonului Trimite sa se realizeze urmatoarele validari de date:

- Campul nume sa contina cel putin 5 litere
- Campul prenume sa contina cel putin 3 litere
- Campul adresa trebuie sa contina cel putin 5 litere si 1 cifra
- Campul email trebuie sa contina o adresa email corecta