TEHNOLOGII WEB - Îndrumar de laborator

LUCRAREA NR. 6 :   
Limbajul JAVASCRIPT – Introducere, variabile, tipuri de date şi operatori

1. **Tematica lucrării**
2. Ce este JAVASCRIPT?
3. Variabile şi tipuri de date în JavaScript
4. Operatori în JavaScript
5. Exemplu de program JavaScript
6. **Teoria lucrării de laborator**
7. Ce este JAVASCRIPT?

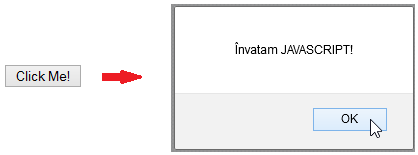
* JAVASCRIPT reprezintă cel mai utilizat limbaj de scripting pe partea de client din reţeaua Web.
* Un limbaj de scripting este caracterizat prin simplitate în sintaxă şi descriere, fiind un limbaj de categorie uşoară.
* Limbajul JavaScript este utilizat în miliarde de pagini web, adăugând dinamică în funcţionalitate, validarea formularelor, comunicarea cu servere şi multe altele.
* JavaScript este un limbaj uşor de învăţat, programele fiind inserate în interiorul paginilor HTML şi sunt interpretate şi executate de către navigatorul web (browser).

Un simplu exemplu constă în afişarea unui mesaj de alertă într-o fereastră de dialog, utilizând funcţia *alert( )*.

*Codul programului:*



*Afişarea în browser:*



1. Variabile şi tipuri de date în JavaScript

În limbajul JavaScript, variabilele se declară simplu, utilizând cuvântul rezervat *„var”*. Pentru a iniţializa o variabilă, aceasta trebuie urmată de simbolul ’=’ şi valoarea dorită pentru iniţializare.

Exemplu: 

De obicei, atât variabilele cât şi instrucţiunile programului sunt urmate de simbolul ’**;**’ . Dacă o variabilă nu este iniţializată, aceasta va primi valoarea *undefined* (nedefinit).

Tipuri de date

Principalele tipuri de date definite de JavaScript sunt:

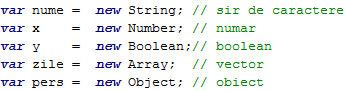
* STRING, şir de caractere;
* NUMBER, număr;
* BOOLEAN, bolean (valori *adevărat* sau *fals*);
* ARRAY, vector de elemente de orice tip;
* OBJECT, obiect cu proprietăţi de orice tip;
* NULL, tipul de date nul;
* UNDEFINED, tipul de date nedefinit.

În tabelul 6. 1 sunt descrise aceste tipuri de date, alături de exemple corespunzătoare.

*Tabelul 6.1.* Tipuri de date definite implicit în JavaScript, cu exemple.

|  |  |
| --- | --- |
| ***Tipul de date*** | ***Exemple*** |
| STRING |  |
| NUMBER |  |
| BOOLEAN |  |
| ARRAY |  |
| OBJECT |  |
| NULL |  |
| UNDEFINED |  |

În limbajul JavaScript se poate declara tipul unei variabile o dată cu declararea acesteia, utilizând cuvântul rezervat *„new”*, ca în exemplul următor:



Toate variabilele JavaScript reprezintă obiecte. Când se declară o variabilă, în fapt se creează un nou obiect.

OBIECTELE limbajului JavaScript

Obiectul este o structură de date având proprietăţi şi metode. Proprietăţile sunt trăsături ale obiectului, asociindu-li-se valori, iar metodele reprezintă acţiuni ale obiectului.

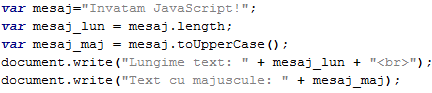
Exemplu, pentru obiectul *persoana* se pot defini următoarele proprietăţi şi metode:

|  |  |
| --- | --- |
| *Proprietăţi* | *Metode* |
|  |  |

Accesul la proprietăţile şi metodele unui obiect JavaScript se realizează prin operatorul de referinţă, simbolul punct ’ **.** ’, astfel:

* *numeObiect* ***.*** *numeProprietate*
* *numeObiect* ***.*** *numeMetoda( )*

Exemplu de folosire şi afişare în browser:





Obiectul *DOCUMENT*

Este unul din cele mai importante obiecte utilizate în JavaScript.

Orice pagină web HTML încărcată într-un navigator web devine un obiect de tip *document*. Proprietăţile şi metodele obiectului *document* sunt listate în tabelul 6.2 şi 6.3, respectiv, de mai jos.

Obiectul de tip *document* oferă accesul la toate elementele HTML conţinute în pagina web respectivă, utilizând un script.

*Tabelul 6.2*. Proprietăţile obiectului *DOCUMENT*

|  |  |
| --- | --- |
| *Proprietatea* | *Descrierea* |
| **anchors** | Returnează o listă cu toate ancorele din document |
| **applets** | Returnează o colecţie cu toate applet-urile |
| **body** | Returnează elementul *body* al documentului |
| **cookie** | Returnează toate perechile *nume/valoare* ale cookie-urilor din document |
| **domain** | Returnează numele de domeniu al serverului care a încărcat documentul |
| **forms** | Returnează lista cu toate formularele din document |
| **images** | Returnează o colecţie cu toate imaginile din document |
| **links** | Returnează lista tuturor link-urilor din document |
| **referrer** | Returnează adresa URL a documentului care a încărcat pagina curentă |
| **title** | Setează sau returnează titlul documentului |
| **URL** | Returnează adresa întreagă URL a documentului |

*Tabelul 6.3*. Metodele obiectului *DOCUMENT*

|  |  |
| --- | --- |
| *Metoda* | *Descrierea* |
| **open()** | Deschide un flux de ieşire (*output stream*) pentru a culege ieşirea generată de metodele *document.write()* sau *document.writeln()* |
| **close()** | Închide fluxul de ieşire deschis anterior prin metoda *document.open()* |
| **getElementsByName()** | Accesează toate elementele cu un nume specificat în atributul *„name”* |
| **write()** | Afişează într-un document expresii HTML sau cod Javascript executat |
| **writeln()** | La fel ca la metoda *write()*, în plus adăugându-se un caracter de linie nouă după fiecare instrucţiune |

Limbajul JavaScript deţine o colecţie mare de obiecte implicite, descrierea lor putând fi accesată prin consultarea referinţelor bibliografice de la finalul îndrumarului de laborator.

1. Operatori în JavaScript

În limbajul JavaScript, operatorii se împart în următoarele categorii: aritmetici, de atribuire, de comparaţie, logici şi condiţionali.

* operatori aritmetici:

*Tabelul 6.3.* Operatorii aritmitici în JavaScript.

|  |  |
| --- | --- |
| *Operator* | *Descriere* |
| + | adunare |
| - | scădere |
| \* | înmulţire |
| / | împărţire |
| % | modulo (restul împărţirii) |
| ++ | incrementare |
| -- | decrementare |

* operatori de atribuire, combinaţi cu cei aritmiteci:

*Tabelul 6.4.* Operatorii de atribuire în JavaScript.

|  |  |  |
| --- | --- | --- |
| *Operator* | *Exemplu* | *Efect* |
| = | x=y |  |
| += | x+=y | x=x+y |
| -= | x-=y | x=x-y |
| \*= | x\*=y | x=x\*y |
| /= | x/=y | x=x/y |
| %= | x%=y | x=x%y |

Notă

Operatorul + aplicat asupra şirurilor de caractere (string) are funcţia de concatenare. Adunarea între un număr şi un şir de caractere are ca rezultat tot un şir de caractere.

* operatori de comparaţie:

*Tabelul 6.5.* Operatorii de comparaţie în Javascript.

|  |  |
| --- | --- |
| *Operator* | *Descriere* |
| == | egalitate |
| === | egalitate exactă (valoare şi tip de date) |
| != | inegalitate |
| !== | inegalitate (valoare şi tip de date) |
| > | mai mare |
| < | mai mic |
| >= | mai mare sau egal |
| <= | mai mic sau egal |

* operatori logici:

*Tabelul 6.6.* Operatorii logici în JavaScript

|  |  |  |
| --- | --- | --- |
| *Operator* | *Descriere* | *Example* |
| && | ŞI logic | (x<10 && y>1) |
| || | SAU logic | (x==5 || y==5) |
| ! | NEGAŢIE logică | !(x==y) |

* operatorul condiţional (operatorul ternar):

Limbajul JavaScript conţine, de asemenea, un operator condiţional, numit şi ternar, care atribuie o valoare unei variabile în baza îndeplinirii unei condiţii.

Sintaxă: 

Exemplu: 

1. Exemplu de program JavaScript

Inserarea de coduri JavaScript în interiorul documentelor HTML se realizează prin utilizarea elementului <SCRIPT>, astfel:



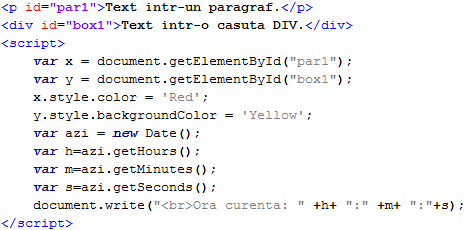
Ca şi la limbajul CSS, scripturile JavaScript se pot încărca şi din exterior, prin fişiere cu extensia „**.js**”, precizându-se sursa fişierului în atributul *„src”*, în modul următor:



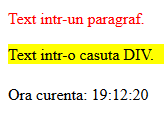
În exemplul următor se foloseşte JavaScript pentru:

* setarea culorii textului unui paragraf şi a fundalului unei căsuţe DIV;
* afişarea ceasului calculatorului, utilizând un obiect de tip *Date()*.

Codul HTML şi JavaScript:



Afişarea documentului în browser:



1. **Modul de lucru**
2. Să se creeze o pagină HTML care să conţină elementele următoare:
   * 2 paragrafe, o imagine şi un tabel de 3x3.
3. Să se creeze un script JavaScript care să realizeze următoarele:
   * să interschimbe textele din cele două paragrafe;
   * să poziţioneze imaginea la dreapta paginii;
   * să seteze bordura tabelului cu grosimea de 1px, culoare verde şi fundalul celulelor de culoare portocalie;
   * să afişeze ziua, data şi ora curentă.
4. Creaţi un script care să execute următoarele operaţii aritmetice:
   * adunare, scădere, înmulţire, împărţire, modulo, incrementare şi decrementare;
   * calculul minimului dintre 3 numere oarecare. Numerele pot fi introduse în căsuţe text.