TEHNOLOGII WEB - Îndrumar de laborator

LUCRAREA NR. 7 :   
Limbajul JAVASCRIPT – Instrucţiuni, funcţii predefinite şi funcţii definite de utilizator

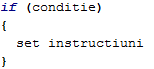
1. **Tematica lucrării**
2. Instrucţiuni în JavaScript
3. Funcţii predefinite ale limbajului JavaScript
4. Funcţii definite de utilizator
5. **Teoria lucrării de laborator**
6. Instrucţiuni în JavaScript

Categoriile de instrucţiuni ale limbajului JavaScript sunt: condiţionale, de ciclare şi de salt.

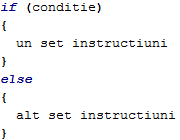
1. *instrucţiuni condiţionale:*

Instrucţiunile condiţionale JavaScript sunt:

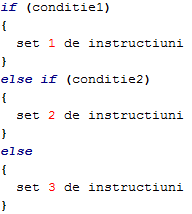
* **instrucţiunea IF** – se execută un set de instrucţiuni numai *dacă* este îndeplinită (adevărată) o condiţie:



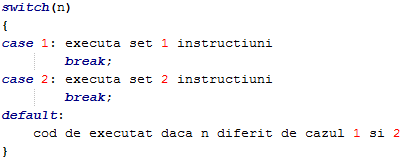
* **instrucţiunea IF...ELSE** – se execută un prim set de instrucţiuni *dacă* condiţia este adevărată şi un al doilea set de instrucţiuni *dacă* condiţia este falsă.



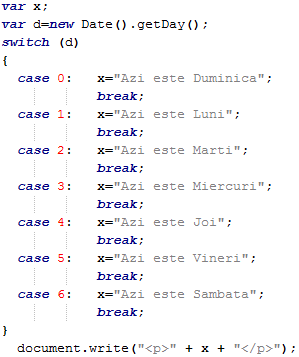
* **instrucţiunea IF...ELSE IF...ELSE** – dacă condiţia este adevărată se execută un set de instrucţiuni din mai multe posibile.



* **instrucţiunea SWITCH** – dacă condiţia este adevărată se execută un set de instrucţiuni din mai multe posibile. Diferenţa faţă de instrucţiunea de mai sus constă în sintaxa de scriere.



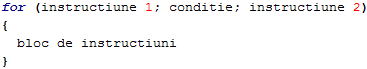
Exemplu de script care afişează în pagina web ziua curentă:



1. *instrucţiuni de ciclare:*

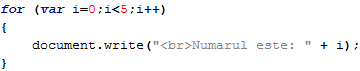
Instrucţiunile de ciclare prezente în JavaScript sunt:

* **instrucţiunea FOR** – ciclează un bloc de instrucţiuni de un număr de ori specificat atât timp cât o condiţie este adevărată:

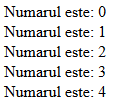


* **instrucţiunea 1** se execută înaintea blocului de instrucţiuni.
* **conditie** defineşte condiţia de rulare a blocului de instrucţiuni.
* **instrucţiunea 2** se execută după fiecare rulare a blocului de instrucţiuni.

Exemplu:

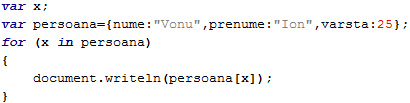


Rezultatul afişat în navigatorul web este:



* **instrucţiunea FOR/IN** – ciclează prin proprietăţile unui obiect:

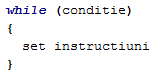
Exemplu:



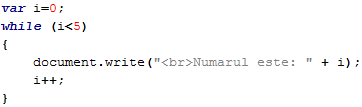
Afişare în browser:



* **instrucţiunea WHILE** – ciclează (executând) un set de instrucţiuni atât timp cât o condiţie este adevărată:

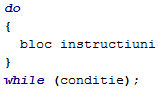


Exemplu:

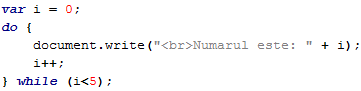


Afişare în navigatorul web este aceeaşi ca şi la ciclul FOR.

* **instrucţiunea DO/WHILE** – la fel ca şi la instrucţiunea WHILE, diferenţa constând în faptul că la instrucţiunea DO/WHILE blocul de instrucţiuni se execută cel puţin o dată:



Exemplu:



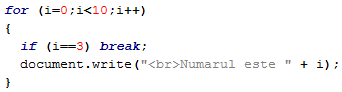
Afişare în navigatorul web este aceeaşi ca şi la ciclul FOR sau WHILE.

1. *instrucţiuni de salt:*

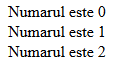
Instrucţiunile de salt utilizate în limbajul JavaScript sunt:

* **instrucţiunea BREAK** – întrerupe o buclă (ciclu) în execuţie, transferând execuţia către prima instrucţiune de după buclă:

Exemplu:

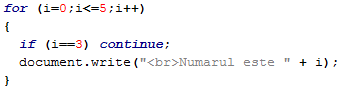


Afişarea se va opri la pasul în care i=2, astfel rezultând:

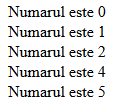


* **instrucţiunea CONTINUE** – întrerupe o singură iteraţie din ciclu, pentru o condiţie specificată şi *continuă* execuţia buclei cu următoarea iteraţie:

Exemplu:



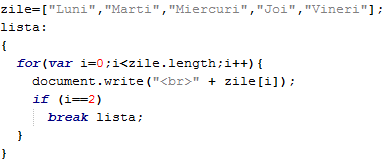
Se vor afişa toate valorile cu excepţia pasului pentru care i=3, rezultând:



Notă:

* instrucţiunea CONTINUE (cu sau fără referinţă la o etichetă) poate fi utilizată numai în cadrul unei bucle condiţionale.
* instrucţiunea BREAK, fără referinţă la o etichetă, poate fi utilizată numai într-un ciclu condiţional sau SWITCH. Cu referinţă la o etichetă, instrucţiunea BREAK poate fi folosită pentru a ieşi din execuţia oricărui bloc de instrucţiuni JavaScript.

Exemplu:



Din tot vectorul „zile” se vor afişa numai primele trei, deoarece instrucţiunea *break lista* va întrerupe blocul reprezentat de eticheta *lista* , după pasul în care i=2, rezultând:



Comentarii în JavaScript

Comentariile pot fi adăugate pentru a explica codul JavaScript, sau pentru a-l face mai lizibil. Comentariile se pot defini în două moduri:

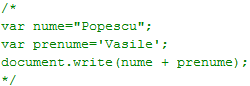
* comentariu pe o linie, utilizând setul de caractere **//** (dublu slash):

Acesta va ignora linia de cod comentată, în procesul de execuţie. Exemplu:



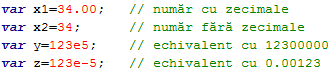
* comentariu pe mai multe linii, utilizând perechea **/\* , \*/** :

În acest caz, se vor ignora la execuţie toate liniile de cod din interiorul grupului de caractere /\*, \*/. Exemplu:



În editoarele moderne, care permit colorarea codului, de obicei, comentariile sunt de culoare verde.

Cel mai des, comentariile sunt utilizate pentru explicarea codului scris. Exemplu:



1. Proprietăţi şi funcţii (metode) predefinite ale limbajului JavaScript

Proprietăţile şi funcţiile predefinite din JavaScript sunt cele conţinute de obiectele predefinite ale limbajului. Aceste obiecte sunt încadrate în trei categorii:

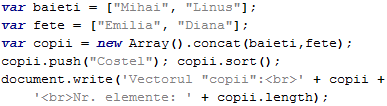
1. obiectele JavaScript
2. obiectele navigatorului (browser-ului)
3. Obiectele JavaScript sunt următoarele:

* obiectul **Array**

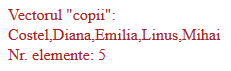
Proprietăţi: *constructor, length, prototype*

Metode: *concat, indexOf, join, lastIndexOf, pop, push, reverse, shift, slice, sort, splice, toString, unshift, valueOf*

Exemplu de utilizare a funcţiilor *concat(), push(), sort()* şi a proprietăţii *length*:



Afişare în browser:

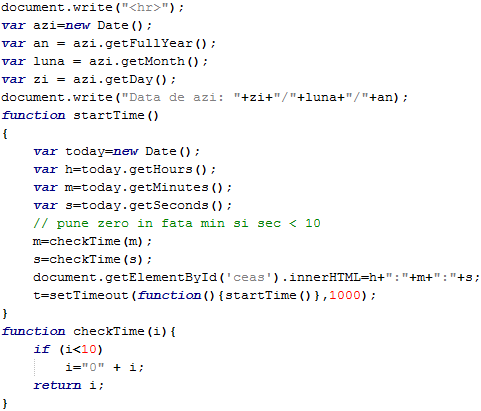


* obiectul **Date**

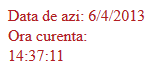
Proprietăţi: *constructor, prototype*

Metode: *getDate, getDay, getFullYear, getHours, getMilliseconds, getMinutes, getMonth, getSeconds, getTime, getTimezoneOffset, getUTCDate, getUTCDay, getUTCFullYear, getUTCHours, getUTCMilliseconds, getUTCMinutes, getUTCMonth, getUTCSeconds, parse, setDate, setFullYear, setHours, setMilliseconds, setMinutes, setMonth, setSeconds, setTime, setUTCDate, setUTCFullYear, setUTCHours, setUTCMilliseconds, setUTCMinutes, setUTCMonth, setUTCSeconds, toDateString, toISOString, toJSON, toLocaleDateString, toLocaleTimeString, toLocaleString, toString, toTimeString, toUTCString, UTC, valueOf*

Exemplu de afisare a datei curente si a ceasului calculatorului în timp real:



Afişare în browser:

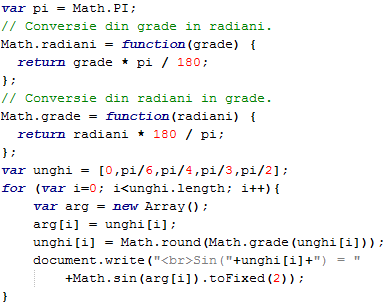


* obiectul **Math**

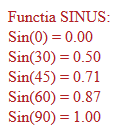
Proprietăţi: *E, LN2, LN10, LOG2E, LOG10E, PI, SQRT1\_2, SQRT2*

Metode: *abs, acos, asin, atan, atan2, ceil, cos, exp, floor, log, max, min, pow, random, round, sin, sqrt, tan*

Exemplu de calculare a valorilor funcţiei *Sinus* pentru unghiurile 0, 300, 450, 600 şi 900:



Afişare în browser:

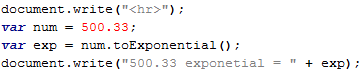


* obiectul **Number**

Proprietăţi: *constructor, MAX\_VALUE, MIN\_VALUE, NEGATIVE\_INFINITY, NaN, POSITIVE\_INFINITY, prototype*

Metode: *toExponential, toFixed, toPrecision, toString, valueOf*

Un exemplu în care un număr este convertit în notaţie exponenţială:



Rezultatul afişat în browser:



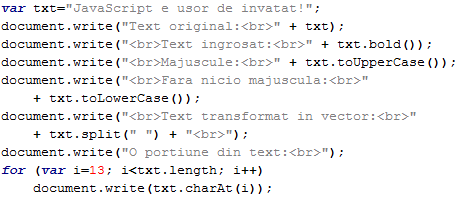
* obiectul **String**

Proprietăţi: *constructor, length, prototype*

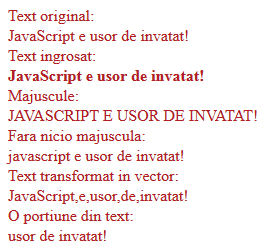
Metode: *charAt, charCodeAt, concat, fromCharCode, indexOf, lastIndexOf, match, replace, search, slice, split, substr, substring, toLowerCase, toUpperCase, valueOf*

Alte metode ce returnează *string-*uri formatate din diverse elemente HTML: *anchor, big, blink, bold, fixed, fontcolor, fontsize, italics, link, small, strike, sub, sup*

În exemplul următor aplicăm prin cod JavaScript diverse formatări asupra unui text:



Rezultatul afişat în navigatorul web:



* obiectul **RegExp**

O expresie regulată este un obiect care descrie un şablon (pattern) de caractere. Expresiile regulate sunt utilizate la efectuarea de funcţii cum ar fi *pattern-matching* (potrivire pe şablon) şi *search-and-replace* (căutare şi înlocuire).

**Sintaxa:**

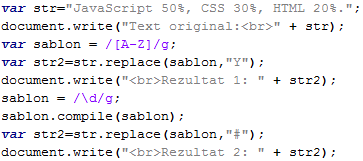


* **şablonul** (pattern) indică modelul, tiparul unei expresii
* **modificatorii** (modifiers) specifică dacă o căutare trebuie făcută global, case-sensitive, etc.

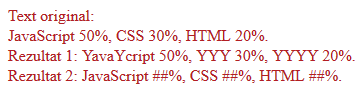
Proprietăţi: *global, ignoreCase, lastIndex, multiline, source*

Metode: *compile, exec, test*

Exemplul următor caută mai întâi litere majuscule şi le înlocuieşte cu litera Y, iar apoi caută cifrele şi le înlocuieşte cu caracterul #:



Rezultatul afişat în browser:

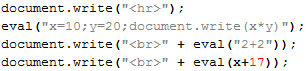


Pe lângă acestea, mai există **proprietăţi şi metode JavaScript globale**.

Proprietăţi globale: *Infinity, NaN, undefined*

Metode globale: *decodeURI, decodeURIComponent, encodeURI, encodeURIComponent, escape, eval, isFinite, isNaN, Number, parseFloat, parseInt, String, unescape*

În exemplul care urmează, testăm funcţia EVAL:



Iar rezultatul execuţiei codului în browser-ul web este:



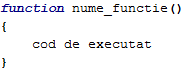
În afară de toate aceste obiecte, JavaScript mai conţine:

1. obiectele navigatorului (browser-ului) sunt următoarele:

* Window
* Navigator
* Screen
* History
* Location

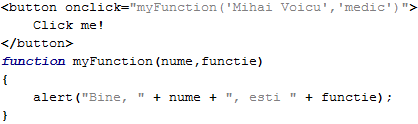
1. Funcţii definite de utilizator

Pentru definirea unei funcţii de către utilizator, sintaxa este următoarea:



Apelul se poate face fie în cadrul codului JavaScript, fie utilizând evenimente de tipul *onclick , onmouseover* etc.

Exemplu de funcţie care afişează un mesaj de alertă într-o fereastră:



Rezultatul afişat în browser este următorul:

|  |  |
| --- | --- |
|  |  |

1. **Modul de lucru**

Utilizând cele învăţate în lucrarea de laborator, să se creeze următoarele script-uri.

1. Calculul sumei numerelor de la 1 la 100.
2. Calculul primelor 10 numere prime.
3. Calculul primelor 10 puteri ale lui 2 .
4. Calculul vârstei dvs., aproape exacte scăzând din data curentă data naşterii.
5. Afişarea denumirii zilei curente.
6. Afişarea unui text în deplasare orizontală sau verticală utilizând strict metode ale obiectelor STRING sau ARRAY.
7. Căutarea în cadrul unui text a literelor *d*, *f* şi *t*.