**Vectori în JavaScript**

Spre deosebire de varibilele simple, trebuie alocaţi (să-i rezervăm loc în memorie).

Acest lucru se face prin:

vector = new Array();

(! atenţie la litere mari/mici !)

După această alocare, elementele se folosesc ca atare (a[1], a[2], ...)

Pentru început vom citi valorile unui vector una câte una, fiecare printr-un prompt:

n=parseInt(prom**p**t("Nr. de elemente",""));

a=new Array();

for(i=1;i<=n;i++)

a[i]=parseInt(prom**p**t("Elementul al "+i+"lea:",""));

Vezi apl01

**Stringuri în JavaScript**

(http://www.w3schools.com/jsref/jsref\_obj\_string.asp)

Lucrul cu stringuri în JavaScript este MULT mai intuitiv decât în C++.

Am văzut deja că, pentru a concatena, utilizăm operatorul +.

În rest, o serie de funcţii specifice stringurilor se accesează ca în C#, adică stringurile sunt privite ca şi variabile de tip obiect, funcţiile specifice fiind membri ai acestor obiecte.

Iată-le pe cele mai importante:

.length – numărul de caractere din string.

Ex:

s="Both";

alert(s.length);//afişează 4

.charAt(indice) – întoarce caracterul de la indicele specificat. Ca şi în C++, într-un string indicii încep de la 0.

Ex:

s="Both";

for(i=0;i<s.length;i++)

alert(s.charAt(i));

.[indexOf(subshir)](http://www.w3schools.com/jsref/jsref_indexof.asp) – intoarce indicele primei aparitzii ale lui subshir in stringul dat.

Dacă nu se găseşte, întoarce -1

.lastI[ndexOf(subshir)](http://www.w3schools.com/jsref/jsref_indexof.asp) – intoarce indicele ultimei aparitzii ale lui subshir in stringul dat.

Dacă nu se găseşte, întoarce -1

Ex:

s="Ana are mere";

alert(s.indexOf("are")); //afişează 4

.substr(start,length) → întoarce substringul care începe de la indicele start şi are lungimea length

s="Ana are mere";

alert(s.substr(4,3)); //afişează "are"

.toLowerCase();

.toUpperCase(); → convertesc stringul la litere mari/mici

.charCodeAt(indice) → codul ASCII al caracterului de la indicele indice

String.fromCharCode(n1, n2, ..., nX) → formează un string din caracterele

obţinute din codurile date. Se pot da oricâte coduri (înclusiv

doar unul singur)

Ex:

alert(String.fromCharCode(65,78,65)); → afişează "Ana"

alert(String.fromCharCode(65)); → afişează "A"

.split("separator") → formează un Array, din substringurile obţuinte prin separarea în funcţie de separatorul specificat. Indicierea începe de la 0 (funcţia îl şi alocă)

Ex:

s="Ana are mere";

a=s.split(" ");

nc=a.length; //ne dă numărul de cuvinte din stringul a

for(i=0;i<nc;i++)

alert(a[i]);

Aplicaţii:

1) Citiţi un string. Număraţi câte vocale are (fie mici, fie mari)

2) Citiţi un string. Număraţi câte perechi de caractere vecine au coduri ASCII consecutive crescătiare

3) Citiţi un string format din multe numere separate prin spaţii.

Formaţi un vector ce are ca elemente, începând de la indicele 1, valorile din string.

Sortaţi valorile descrescător şi reafişaţi vectorul