

1 Aproape toate limbajele de programare folosesc noțiuni fundamentale, cum ar fi **variabile**, **operatori**, **bucle repetitive (FOR, WHILE)**, **structuri condiționale (IF)**, **funcții**, **obiecte**, **proprietăți**, **metode**. Fără acestea programarea calculatoarelor ar de neimaginat.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Test_JS_variabile </title>
5   </head>
6   <body>
7     <script>
8   </script>
9   </body>
10  </html>
```

Crează acum acest fișier html simplu pe care îl vom folosi doar pentru exemplificarea acestor concepte. Vom scrie cod doar între rândurile 7 și 8 și vom testa imediat în browser.

2 Variabile și operatori

Scrie în secțiunea de cod Javascript și testează:

```
var cuvinte = "Aceasta este "
var alte_cuvinte = " o propozitie"
var prop = cuvinte + alte_cuvinte;
document.write(prop);
```

Apoi testează acest cod:

```
var num1 = 6;
var num2 = 5;
var total = num1 + num2;
document.write(total);
```

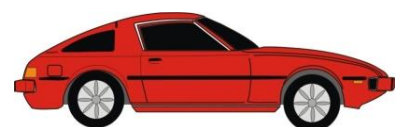


Modifică operatorul din rândul al treilea în : -, /, *. Apoi încearcă `num1%num2`
Acest operator (%) îți dă restul împărțirii celor două numere.
Apoi scrie: `var total = num1++`;
Acest operator (++) incrementează numărul cu valoarea 1.

3 Obiecte, proprietăți, metode

Ceea ce scriem noi ca un text este de fapt un șir de caractere (String). Șirurile sunt de fapt **OBIECTE**. Ele au anumite proprietăți. Lungimea este una din ele.

```
var alpha = "ABCDEFGH";
var lun = alpha.length;
document.write(lun);
```



Pentru obiecte mai avem și metode. Diferența între proprietate și metodă este că proprietatea este doar o bucată de informație, este o valoare care îți spune ceva despre obiect, în timp ce metodele primesc date de intrare, fac anumite calcule și oferă niște date de ieșire.

Aici vom apela o metodă (substring) care extrage dintr-un șir de caractere doar o porțiune:

```
var alpha = "ABCDEFGG";  
var alphaSmall = alpha.substring(3,5);  
document.write(alphaSmall);
```

Testează și cu **alpha.substring(0,2)**
Primul argument al acestei metode arată de unde începe subșirul (primul element este 0) iar al doilea element arată unde se termină (exclusiv acela).

4 Arrays (siruri, matrici, tablouri)

Arrays conțin mai multe valori într-o singură variabilă. Arrays sunt utile atunci când vrei să lucrezi cu mai multe date (obiecte) și vrei să le faci cam același operațiuni tuturor.

```
var a = new Array(5);  
a[0] = "apple";  
a[1] = "orange";  
a[2] = "banana";  
a[3] = 15;  
a[4] = true;  
document.write(a[1]);
```



Aici am declarat un array de 5 elemente și i-am atribuit valori.
Apoi am afișat al doilea element, adică a[1] care are valoarea "orange".

Array-urile se pot define în mai multe feluri:

```
var a = new Array("apple", "orange", 15, true);
```

```
var a = ["apple", "orange", 15, true];
```

Retine!

Array-urile sunt obiecte mai speciale. Ele utilizează numere pentru a accesa un anumit element. Întotdeauna primul element are indicele 0. **De ex:** a[0] va returna valoarea "apple". Acesta este primul element din array.

Cel mai ușor mod de a adăuga elemente unui array este prin metoda push()

```
a.push("kiwi")  
document.write(a.length);  
document.write(a);
```

Aici am adăugat un nou element, și apoi am verificat lungimea array-ului.
Apoi am cerut să ne afișeze întregul array.

Nu-i așa că rezultatele afișate sunt cam înghesuite? Asta se întâmplă deoarece html nu ține cont de spațieri decât dacă le declarăm explicit. De aceea vom folosi tagul
. De exemplu așa:

```
document.write(a.length+"<br>");
```