



LABORATOR 5

<https://www.w3schools.com/>
<https://developer.mozilla.org/>
<https://www.w3.org/standards/webdesign/>

JS – cod ce poate fi integrat in HTML; limbajul este bazat pe obiecte si trateaza tot ca pe un obiect.

[JavaScript](#) (JS) este unul dintre cele mai populare limbaje de programare din lume, fiind utilizat atat pe partea de client ([browser](#)) cat si pe partea de server ([NodeJs](#)).

Pana acum am lucrat HTML si CSS pentru a construi site-uri web, dar avem o singura problema: sunt pagini web statice. Orice pagina web trebuie sa aiba o interactiune cu utilizatorul si acest lucru il vom face folosind JS. Astfel, utilizatorii vor putea sa deschida si sa inchida meniuri, sa extinda si sa restranga secțiuni din pagina, sa navigheze prin fotografii, sa foloseasca filtre pe paginile produselor și multe altele.

JS – Unde scriem codul?

Codul Javascript poate fi scris in etichetele **<script>**. Acestea arata la fel ca elementele HTML normale si pot fi plasate in <head> sau in secțiunea <body>. In majoritatea cazurilor, cel mai bun loc pentru a include etichetele scriptului va fi la capatul elementului <body>:

```
<!DOCTYPE HTML>

<html>

  <head>
    <script>

      // Javascript code can be written here

    </script>
  </head>

  <body>
    <!-- page content -->

    <script>

      // Javascript code can be written here

    </script>
  </body>
```



</html>

Cu toate acestea, cea mai buna varianta de scris cod JS este intr-un fisier separat:

<body>

<!-- html page -->

<script src = 'js/scripts.js'></script>

</body>

JS – Sintaxa

Declarațiile sunt instrucțiuni care trebuie executate de browserul web. Fiecare declarație trebuie terminata cu punct si virgula (;). Codul Javascript este compus din mai multe instrucțiuni care vor fi executate in ordine secventiala, **de sus in jos**. Fiecare instructiune este executata de catre un *JS engine*, care difera in functie de browserul folosit (Google Chrome - [V8](#), Mozilla Firefox - [Gecko](#), [SpiderMonkey](#) si [Rhino](#), Safari – Nitro etc).

- afisare in consola

```
console.log ('statement 1');
```

```
console.log ('statement 2');
```

- case sensitive

```
console.log('something'); ≠ Console.log('something');
```

JS – VARIABLE – declarare: “var” sau “let”

```
var a = 'hello'; //global
```

```
let b = 'world'; //locala blocului in care a fost declarata
```

JS – NULL / UNDEFINED – *primitive data types*

- undefined – variabila creata, dar fara valoare asignata.

```
var noValue;
```

```
console.log (noValue); // undefined
```

```
console.log (typeof noValue); // undefined
```

- null – nu este socotit ca facand parte dintre tipurile de date (Strings, Numbers, Booleans), dar este folosit pentru a reseta o valoare deja data unei variabile.

```
var temperature = 35;
```

```
console.log (temperature);
```



```
temperature = null;  
console.log (temperature); // consola arata null  
console.log (typeof temperature); // consola arata ca e obiect
```

Cu toate acestea, urmatoarea linie de cod: `console.log (undefined == null)`; va afisa valoarea true.

EXERCITII – 1

1. Afisati in consola: " Hi, my name is _____, I'm _____ years old and I'm learning Javascript ". (folosim anul current pentru calcularea varstei)
2. Validarea numerelor de telefon:

```
var phone1 = "988866552";  
var phone2 = "99087612366";  
var phone3 = 876543123;
```

3. Afisati in consola rezultatul: 32^6 ([Math Object](#))
4. Ce se afiseaza in urma operatiilor:

```
var quantity = "25";  
var number = 6;  
var pressure;  
var temperature = null;  
console.log (quantity += quantity);  
console.log ((7+5) / number + 2);  
console.log (pressure);  
console.log (temperature);  
console.log (typeof pressure);  
console.log (typeof temperature);
```



JS – FUNCTII – bloc de cod care are ca scop indeplinirea unei cerinte/unui task – ajuta la reorganizarea codului si reduce complexitatea acestuia.

Crearea functiei		
<pre>function sum_numbers () { var num1 = 5; var num2 = 2; var sum = x + y; console.log(sum); }</pre>	<pre>function sum_args (num1, num2) { var sum = num1 + num2; console.log(sum); }</pre>	<pre>function sum_args (num1, num2) { var sum = num1 + num2; return sum; }</pre>
functia este executata doar dupa ce a fost invocata		
<pre>sum_numbers (); // consola afiseaza 7</pre>	<pre>sum_args (10,25); // consola afiseaza 35 sum_args (1000,375); // consola afiseaza 1375 sum_args (-2,47); //consola afiseaza 45</pre>	<pre>var average = sum_args (6,10) / 2; console.log (sum_args (13,21));</pre>

JS – ARRAYS – *data types* – secvente de valori: var array1 = [val1, val2, val3, ...]

Operatii:

Push – adauga elemente la **sfarsitul** secventei

Pop – sterge **ultimul** element al secventei

Shift – sterge **primul** element al secventei

Unshift – adauga elemente la **inceputul** secventei



UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ



Slice – extrage parti din secventa: **slice (index_start, index_end)**

- Exemple:

1. var courses = ["HTML", "Python", "PHP"];	
courses.push("Javascript"); console.log(courses);	["HTML", "Python", "PHP", "Javascript"]
courses.unshift("Bootstrap"); console.log(courses);	["Bootstrap", "HTML", "Python", "PHP", "Javascript"]
courses.pop (); console.log(courses);	["Bootstrap", "HTML", "Python", "PHP"]
courses.shift (); console.log(courses);	["HTML", "Python", "PHP"]

2. var ingredients = ["bread", "cheese", "ham"];	
ingredients[0] = "whole bread"; console.log(ingredients);	["whole bread", "cheese", "ham"]

3. var students = ["Peter", "Joana", "Andrew", "Julio", "Kate", "Marie"];	
console.log (students.slice(0,3));	["Peter", "Joana", "Andrew"]

4. var students = ["Peter", "Joana", "Andrew", "Julio", "Kate", "Marie"];	
console.log (students.slice(-3,));	["Julio", "Kate", "Marie"]

JS – OBIECTE – *data types* – ca si *array*-urile, sunt liste de elemente. Dar diferenta consta in faptul ca nu sunt ordonate: aici, identificam elementele dupa **cheie**, nu dupa **index**.

Cheia reprezinta proprietatea obiectului. Deci obiectul este o lista **cheie/valoare**, separate prin ":" si declarant intre "{}":



UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ



```
var objectExample = {  
    'key1': value1,  
    'key2': value2,  
    'key3': value3,  
    etc};
```

- De exemplu, pentru:

```
var employee = {  
    'name': 'James Taylor',  
    'yearOfBirth': 1948,  
    'ID': 'SBJ0001',  
    'role': 'IT Analyst'  
};  
  
console.log( employee['ID'] ); // VAR 1 - consola arata 'SBJ0001'  
console.log( employee.ID ); // VAR 2  
employee.role = 'IT Manager'; // modificarea unei valori  
employee.passport = 'KV09888'; // adaugare de proprietate  
  
console.log(employee);  
/* consola afiseaza:  
{  
    'name': 'James Taylor',  
    'yearOfBirth': 1948,  
    'ID': 'SBJ0001',  
    'role': 'IT Manager',  
    'passport': 'KV09888'  
};  
*/
```



Arrays se identifica prin [].

Arrays sunt liste **ordonate** de valori, separate prin virgula.

Obiectele se identifica prin { }.

Elementele obiectelor sunt **cheie/valoare** separate prin ":".

Elementele obiectelor sunt separate prin virgula.

JS – METODE – valorile obiectelor pot fi functii = metode => metoda = functie + proprietatea obiectului in js SAU metoda = functia care foloseste cuvantul cheie "this".

- **Exemplu:**

```
var student = {  
  'firstName': 'Marie',  
  'lastName': 'Smith',  
  'fullName': function () {  
    return this.firstName + ' ' + this.lastName;  
  }  
}  
  
console.log (student.fullName() ); // consola afiseaza 'Marie Smith'
```

- **Important:**

Funcțiile declarate in interiorul obiectului nu mai au nevoie de nume dupa cuvantul cheie **function**. Pentru a le apela, vom folosi numele proprietatii (in exemplu: fullName). In acest caz putem folosi doar notatia cu punct, neputand apela functia in felul urmator:
student[fullName()].

EXERCITII – 2

1. Creati o functie de conversie a gradelor Celsius in Fahrenheit. Formula: $F = (9 * C / 5) + 32$. Temperaturile: 38°C, 24°C, -1°C si rezultatele 100.4°F, 75.2°F, 30.2°F.
2. Afisati titlul cursului si principala categorie. Implementati o metoda care sa calculeze procentul review-urilor de 5 stele, rotunjit, fara zecimale.



UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ



```
var course = {
  'title': "Learn to Code in Python 3",
  'categories': ['programming', 'technology', 'python'],
  '5_stars_reviews': 420,
  '4_stars_reviews': 80,
  '3_stars_reviews': 33,
  '2_stars_reviews': 20,
  '1_stars_reviews': 4
}
```

3. Avem următoarea lista de cumpărături. Luați ultimul element și plasati-l la începutul listei; adăugați încă 2 elemente la sfârșitul listei: Cheese, Eggs.

```
var shoppingList = ["Milk", "Butter", "Juice", "Bread", "Beer",
"Rice", "Potatoes", "Chocolate"];
```

4. Scrieți cod JavaScript pentru eliminarea repetițiilor într-un array.

De exemplu, pentru `array1 = ['a', 'b', 'a', 'b', 'c', 'e', 'e', 'c', 'd', 'd', 'd', 'd']` va rezulta `array2=['a','b','c','e','d']` (folosiți metode ale Array: `reduce`, `push`, `indexOf`).

5. Scrieți cod JavaScript care transformă un array de array-uri într-un singur array conținând toate elementele din array-urile continute.

De exemplu, pentru `array1 = [[1,2],[3,4,5],[6,7]]` va rezulta `array2=[1,2,3,4,5,6,7]` (folosiți metode ale Array: `reduce`, `concat`)

6. Creați un array de obiecte numit `persoane` care să conțină informații despre persoane, fiecare obiect trebuie să conțină numele și vârsta unei persoane (ex: `persoane = [{nume: "Maria", varsta: 20}, {nume: "Andrei", varsta: 10}, ...]`).

- Folosind o fereastră prompt cereți utilizatorului să introducă un număr întreg.



UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ



- Creați un vector cu toate persoanele preluate din array-ul "persoane", care au vârsta mai mică sau egală cu numărul introdus în fereastra prompt și afișați într-o fereastră alert numărul lor.

7. Exersați metodele [prompt](#), [alert](#) și [confirm](#) din JavaScript.

Creați un obiect literal care să conțină un dicționar (exemplu pentru dicționar: dicționar = {"acesta": "this", "este": "is", "un": "a"}).

- Folosind o fereastră prompt cereți utilizatorului să introducă un cuvânt.
- Căutați cuvântul în dicționar și afișați traducerea (sau un mesaj în cazul în care nu îl găsiți) folosind o fereastră alert.
- Afișați apoi o fereastră confirm în care întrebați utilizatorul dacă vrea să continue. În caz afirmativ luați 1.