Universitatea Tehnică a Moldovei

Facultatea “Calculatoare, Informatică și Microelectronică”

**Raport**

Sisteme multimedia

Lucrarea de laborator nr.5

Tema: **Limbaje client-side script**

A efectuat : st. gr. FAF-131, Adauge Marius

A verificat : lect.sup., Sava Nina

Chișinău 2016

**Sarcina:**

Familiarizarea cu limbajele JavaScript si ActionScript.

**Teorie:**

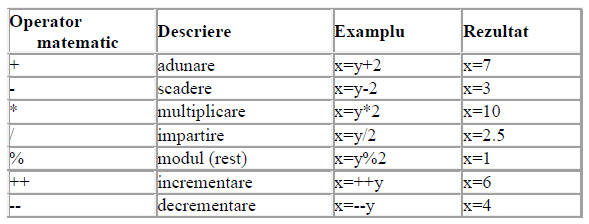
1. **Care sunt avantajele limbajului JavaScript?**

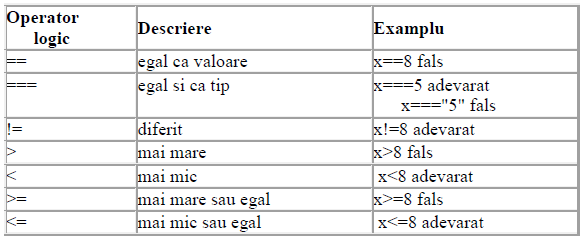
* nu necesită instalarea unor aplicații suplimentare pentru a rula.
* este simplu de învățat (sintaxa este asemănătoare cu cea a limbajelor C și Java)
* permite implementarea unor aplicații complexe într-o manieră simplă
* din el se poate accesa și controla direct toate elementele unei pagini HTML (text, imagini, formulare, frame-uri etc)
* sprijină validarea datelor din formularele HTML
* permite programarea bazată pe evenimente

1. **Variabile JavaScript, operatori, instrucțiuni.**

**Variabile**: variabilele JavaScript sunt *case sensitive*. Varibilele nu au asociat un tip de date de la declarare ci pot să-și schimbe tipul funcție de valoarea stocată. Numele poate începe cu o literă sau cu caracterul "\_". Pentru declararea unor variabile se va folosi construcția *var*. Ex: var x=2, y;

**Operatori:**



Oparatorii de asignare (=, +=, -=, \*=, /=, %=). Operatorul de concatenare șiruri (+). Dacă se concatenează un șir cu un număr rezultă un șir.

**Instrucțiuni:**

* **IF**

<**script** type="text/javascript">

**var** d = new Date();

**var** time = d.getHours();

**if** (time < 10) {

document.write("Buna Dimineata!");

} else document.write("Buna Ziua!");

</**script**>

* **SWITCH**

<**script** type="text/javascript">

**var** d=new Date();

theDay=d.getDay();

**switch** (theDay) {

**case** 0: document.write("duminica"); **break**;

**case** 1: document.write("luni");

...

**case** 6: document.write("sambata"); **break** ;

**default**: document.write("zi gresita!"); **break**;

}

</**script**>

* **FOR**

<**script** type="text/javascript">

**var** n=1,i;

document.write("<h2>n factorial </h2><br>");

**for**(i=1;i<=10;i++) {

n=n\*i; document.write(i"! = "+n+"<br>");

}

</**script**>

* **WHILE / DO WHILE**

<**script** type="text/javascript">

**var** n=1, i=1;

document.write("<h2>numere impare </h2><br>");

**while**(i<=10) {

document.write(n,", ");

n+=2;

}

</**script**>

* **FOR IN**

for (variable in object)

{

//code de executat

}

1. **Tipuri de ferestre de avertizare :**

* Alert box : alert("mesaj");
* Confirm box : confirm("mesaj");
* Prompt box : prompt("mesaj","valoare implicita");

1. **Care este sintaxa de declarare a funcțiilor. Care sunt evenimentele captate de JavaScript?**

**Funcțiile** sunt secvențe de cod care se execută fie prin apelarea explicită fie prin legarea de un eveniment. De obicei funcțiile sunt plasate în secțiunea <head>. Funcțiile au un nume, o listă de parametrii de apel și un bloc de cod. Dacă codul funcției calculează o valoare atunci aceasta poate fi returnată prin aplelul return val.

**function** numeFunctie(var1,var2,...,varN)

{

//codul functiei

}

**Evenimentele** sunt acțiuni care sunt detectate de către browser și pot declanșa execuția unor funcții. Evenimentele captate de JavaScript: **onLoad, onUnload, onFocus, onBlur, onChange, onSubmit, onMouseOver, onMouseOut**.

Exemplu:

**function** verificSuma() {

if(document.getElementById("suma").value<0) {

alert("Suma nu poate fi negativa!"); document.getElementById("suma").value=0;

}

}

...

<**input** type="text" size="30" id="suma" onchange="verificSuma()">

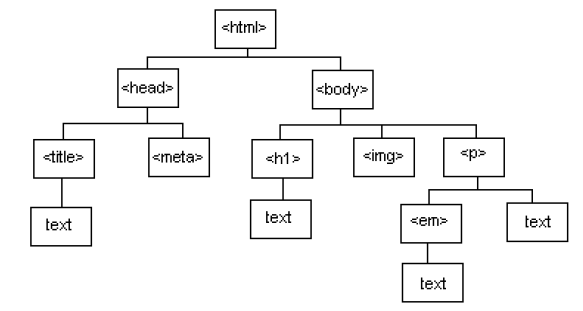
1. **Ce mecanisme JavaScript se utilizează pentru tratarea erorilor, captarea lor?**

Pentru tratarea erorilor se pot folosi contrucțiile **try..cartch** pentru captarea lor, respectiv **throw** pentru generarea lor.

1. **Ce operator se utilizează pentru crearea obiectelor?**

Pentru crearea obiectelor se foloseste operatorul **new**

1. **Dați exemplu de model DOM.**



1. **Ce reprezintă Action Script, care este versiunea actuală?**

ActionScript este un limbaj asemănător cu JavaScript, folosit în principal pentru dezvoltarea de functionalități în cadrul animațiilor Adobe Flash prin generarea fișierelor cu extensia .swf ce rulează în Adobe Flash Player. A fost dezvoltat inițial de compania Macromedia, iar acum aparține Adobe, care a achiziționat Macromedia în 2005. Limbajul de programare are la bază standardul ECMA. El a fost creat pentru desenarea și controlarea animațiilor vectoriale 2D dar de-a lungul vremii s-a dezvoltat pentru a realiza aplicații WEB complexe sau chiar jocuri. ActionScript este *case-sensitive*.(wikipedia). Versiunea actuală: Actionscript 3.0.

1. **Elemente utilizate în Action Script.**

* Tipuri de date: Boolean, Int, Uint, Object, Number, String, Null, Void.
* Variabile:

var a:int = 10, b:int = 20;

var numArray: Array = ["zero", "one", "two"];

* Constante predefinite: null, undefined, true, false.
* Declararea constantelor: const MIN:int = 0;
* Obiecte:

declarare:

var someObj:Object;

instantiere: var myBirthday:Date = new Date();

var someArray:Array = new Array(1,2,3);

* Clase dinamice:

**dynamic** **class** Persoana {

**private var** adresa:String;

**public var** nume:String = "Ion";

**function** Persoana() {

this.adresa="nu are";

}

}

**var** pers:Persoana = new Persoana();

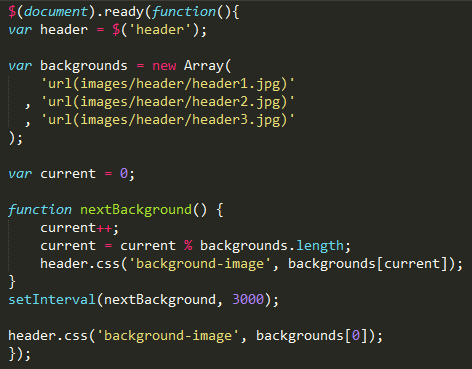
pers.inaltime = 180;

* Instrucțiuni condiționale: if..then..else, switch la fel ca și la JavaScript
* Instrucțiuni ciclice: for, for.. in, for each.. in, while, do while.

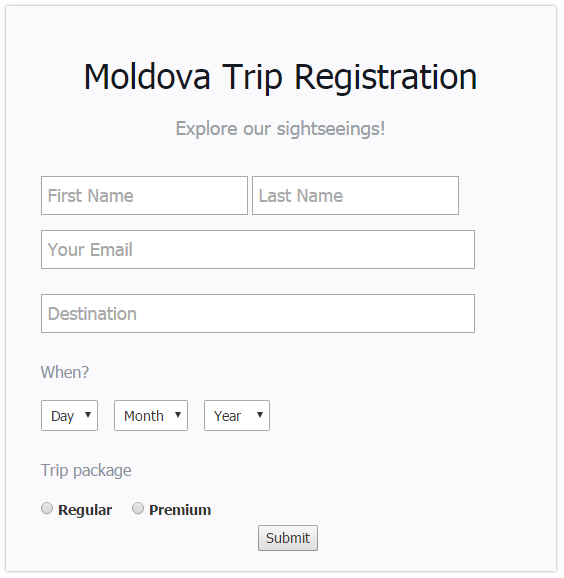
**Realizare:**

În această lucrare de laborator s-a lucrat cu pagina web creată la laboratorul precedent (HTML și CSS), adăugându-i dinamicitate. Câteva elemente dinamice au fost adăugate.

Pe pagina principală s-a creat un slide-show pentru header. Având câteva imagini salvate pe local, acestea sunt plasate într-un array apoi se iterează prin acesta, imaginile schimbându-se fiecare 3 secunde. Mai jos este functia scrisă în JavaScript care va rula automat ce pagina este accesata:

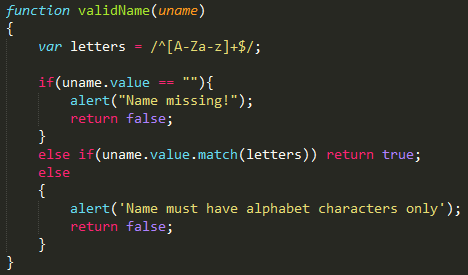


Al doilea element de dinamicitate care a fost implimentat este validarea unei forme de înregistrare. Aceasta este forma de înregistrarea pentru o eventuală excursie prin Molova:



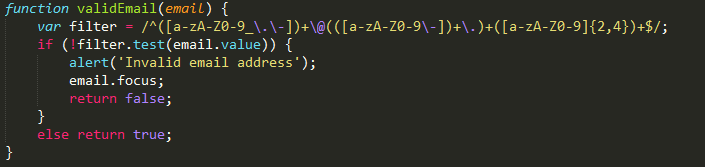
La apăsarea butonului **Submit** este apelată funcția **formValidation(),** care la rândul său conține mai multe funcții ce verifică prezența și corectitudinea datelor introduse în forma de registrare. Mai jos sunt prezentate câteva funcții care verifică anumite câmpuri din formă.

Validarea numelui utilizatorului:



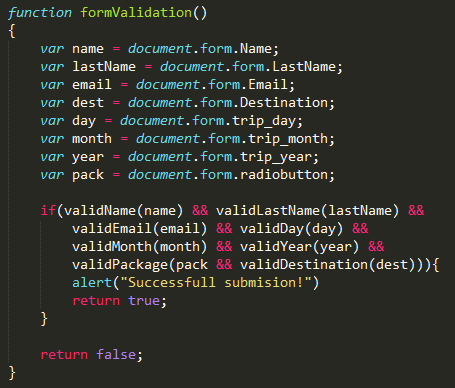
Poate fi observat că în caz că numele n-a fost introdus sau acesta conține caractere non-alphabetice, va apărea o fereastră de avertizare. În caz de succes, funcția returnează **true**.

Validarea email:



Aici se verifică dacă email-ul introdus coincide cu pattern-ul: [nume@website.dom](mailto:nume@website.dom). Pentru aceasta se folosesc expresii regulate care definesc pattern-ul respectiv.

Funcțiile de validare pentru fiecare câmp din forma de înregistrare sunt asamblate spre final într-o singură funcție formValidation(). În caz că fiecare din funcțiile de validare returnează **true** atunci va apărea o fereastră cu mesajul de ”Successful submision”:

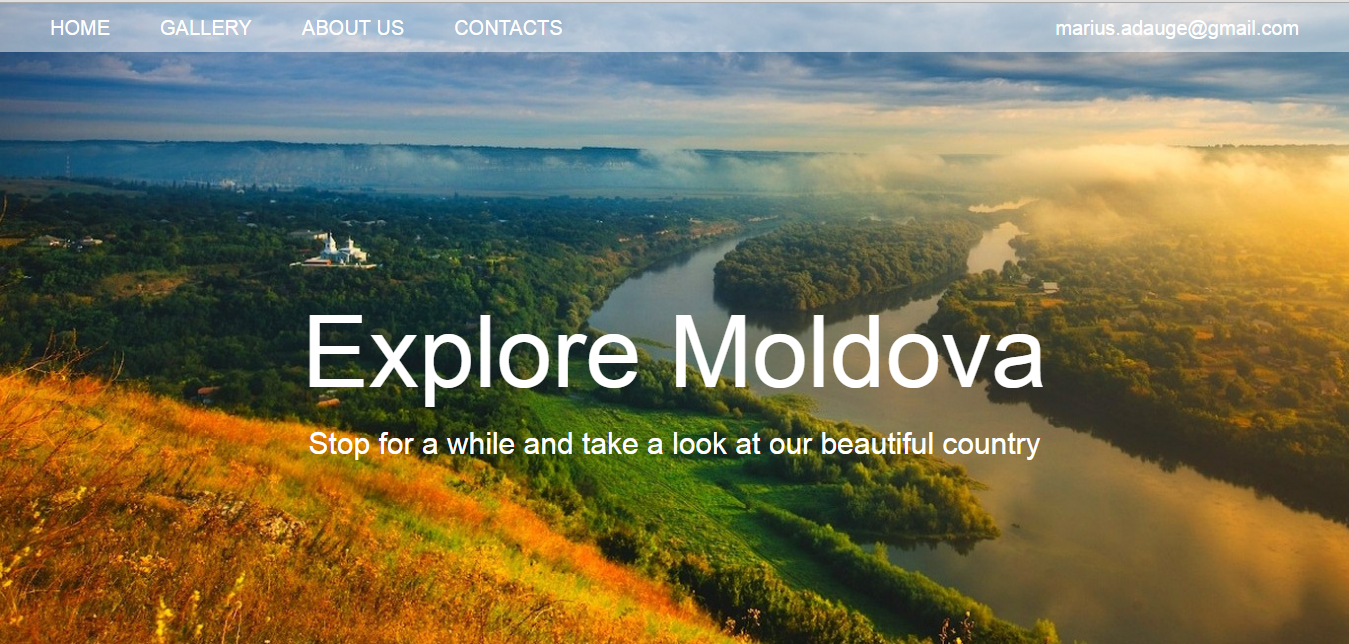


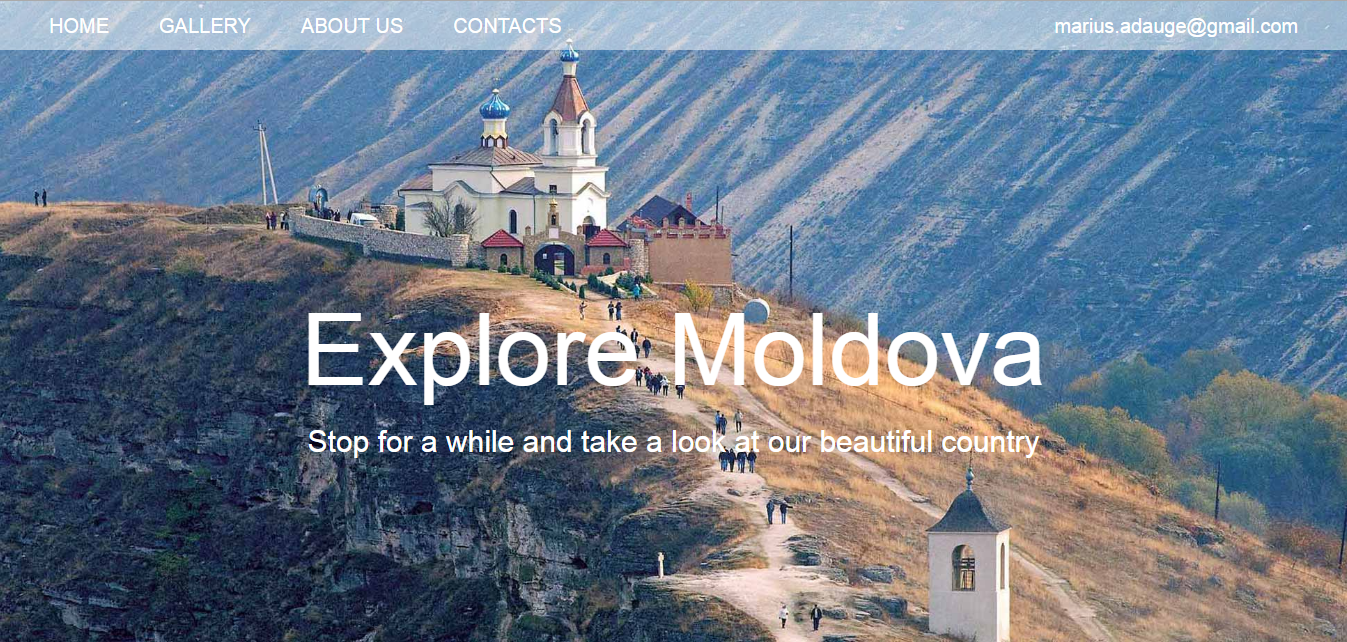
Funcția de mai sus este asociată cu evenimentul de onSubmit al formei în codul HTML:



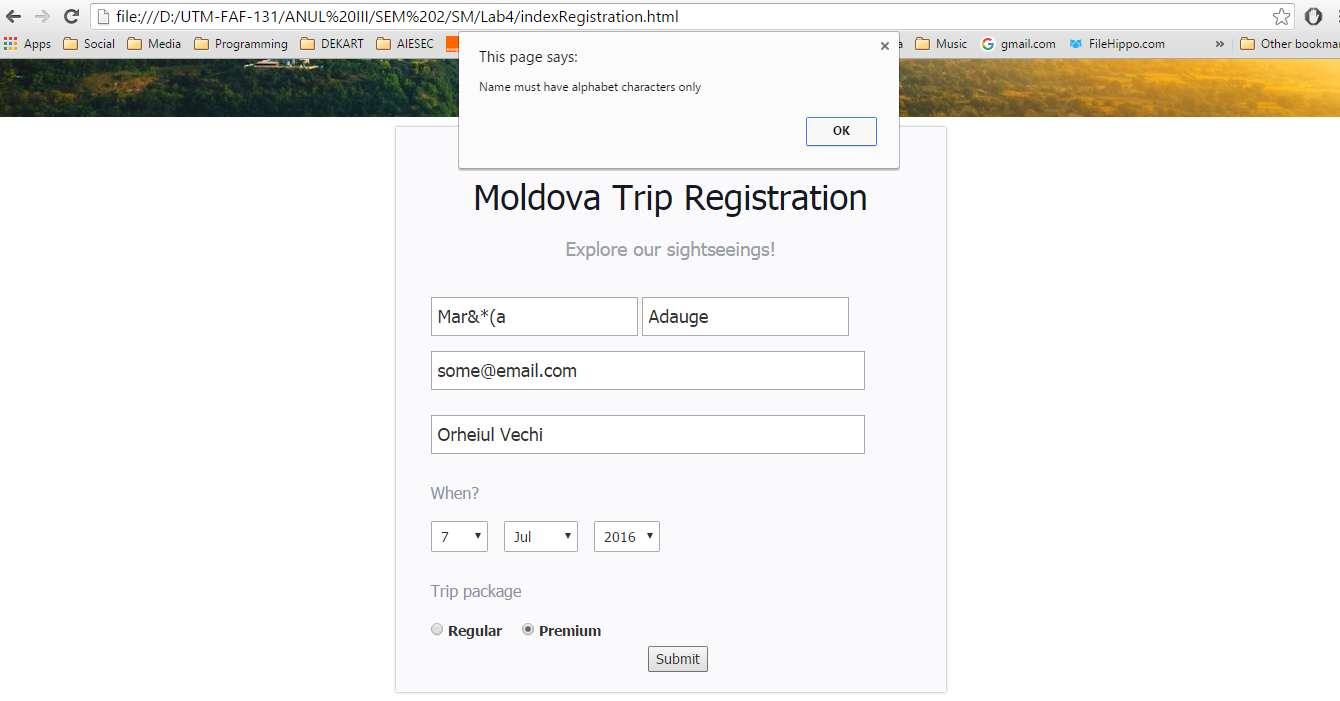
**Rezultate**

Slide-show-ul imaginilior din header:





Introducerea datelor greșite în forma de înregistrare:



**Concluzie:**

Lucrarea data de laborator a fost o recapitulare a cunoștințelor legate de limbajele client-side. Am aflat cum acestea implementează dinamism în pagina web, care-s elementele de bază și avantajele. Am aflat despre ActionScript și cum acesta se utilizează în crearea aplicațiilor web.

**Appendix:**

Codul sursă: <https://github.com/MarcusAdauge/SM-Labs>