

Ministerul Educației și Cercetării



**Program postuniversitar de conversie profesională
pentru cadrele didactice din mediul rural**

*Specializarea TEHNOLOGIA INFORMAȚIEI
Forma de învățământ ID - semestrul III*

REALIZAREA PAGINILOR WEB

Ana Magdalena ANGHEL

Adriana OLTEANU

Radu PIETRARU

Ministerul Educației și Cercetării
Proiectul pentru Învățământul Rural

TEHNOLOGIA INFORMAȚIEI

Realizarea paginilor WEB

Ana Magdalena ANGHEL

Adriana OLTEANU

Radu PIETRARU

2006

© 2006 Ministerul Educației și Cercetării
Proiectul pentru Învățământul Rural

Nici o parte a acestei lucrări
nu poate fi reprodusă fără
acordul scris al Ministerului Educației și Cercetării

ISBN 10 973-0-04551-8;
ISBN 13 978-973-0-04551-2.

Proiectul pentru Învățământul Rural

REALIZAREA PAGINILOR WEB

CUPRINS

Unitate de învățare	Titlu	Pagină
	INTRODUCERE	1
1	INTERNET ȘI WORD WIDE WEB	4
	Obiectivele Unității de învățare nr.1	5
	1.1 Introducere în Internet	5
	1.2 Arhitectura Word Wide Web	7
	1.3 Editarea și vizualizarea unei pagini WEB	8
	Răspunsuri și comentarii la întrebările din testele de autoevaluare	11
	Bibliografie Unitate de învățare nr.1	11
2	LIMBAJUL HTML	12
	Obiectivele Unității de învățare nr.2	13
	2.1 Introducere în HTML	13
	2.2 Sintaxa Directivelor HTML	13
	2.3 Structura de bază a unui document HTML	19
	Răspunsuri și comentarii la întrebările din testele de autoevaluare	22
	Bibliografie Unitate de învățare nr.2	22
3	FORMATAREA TEXTELOR IN LIMBAJUL HTML	23
	Obiectivele Unității de învățare nr.3	24
	3.1 Limbajul HTML și formatarea textelor	24
	3.2 Titluri și paragrafe HTML	24
	3.3 Definirea caracteristicilor fontului	29
	3.4 Formatarea fixică a textelor	32
	3.5 Formatarea logica a textelor	33
	3.6 Preformatarea textului cu ajutorul directivei <PRE>	34
	Lucrare de verificare Unitate de învățare nr.1, 2, și 3	35
	Răspunsuri și comentarii la întrebările din testele de autoevaluare	37
	Bibliografie Unitate de învățare nr.3	37
4	LISTE ÎN LIMBAJUL HTML	38
	Obiectivele Unității de învățare nr.4	39
	4.1 Tipuri de liste în limbajul HTML	39
	4.2 Liste HTML neordonate	40
	4.3 Liste HTML ordonate	42
	4.4 Liste definite	44
	Răspunsuri și comentarii la întrebările din testele de autoevaluare	47
	Bibliografie Unitate de învățare nr.4	48

5	HYPERLINKS ÎN LIMBAJUL HTML	49
	Obiectivele Unității de învățare nr.5	50
	5.1 Hypertext și hyperlink	50
	5.2 URL – identificator unic de resurse web	51
	5.3 Realizarea legaturilor în HTML	51
	5.4 Folosirea imaginilor pentru legături	55
	Răspunsuri și comentarii la întrebările din testele de autoevaluare	57
	Bibliografie Unitate de învățare nr.5	57
6	FOLOSIREA IMAGINILOR ÎN LIMBAJUL HTML	58
	Obiectivele Unității de învățare nr.6	59
	6.1 Înțelegerea formatelor grafice folosite în WEB	59
	6.2 Folosirea imaginilor în cadrul paginilor WEB	60
	6.3 Imagini cu arii sensibile (Image Maps)	65
	Lucrare de verificare Unitate de învățare nr.4, 5, și 6	67
	Răspunsuri și comentarii la întrebările din testele de autoevaluare	70
	Bibliografie Unitate de învățare nr.6	71
7	REALIZAREA TABELELOR	72
	Obiectivele Unității de învățare nr.7	73
	7.1 Realizarea unui tabel simplu în limbajul HTML	73
	7.2 Definirea proprietăților globale ale unui tabel HTML	75
	7.3 Definirea rândurilor unui tabel	78
	7.4 Definirea celulelor unui tabel	79
	Răspunsuri și comentarii la întrebările din testele de autoevaluare	82
	Bibliografie Unitate de învățare nr.7	82
8	FORMULARE ÎN PAGINA WEB	83
	Obiectivele Unității de învățare nr.8	84
	8.1 Ce sunt formularele?	84
	8.2 Introducerea unui formular în pagina WEB	86
	8.3 Elementele HTML folosite în formulare	88
	8.4 Alte tipuri de elemente folosite în formulare WEB	90
	Răspunsuri și comentarii la întrebările din testele de autoevaluare	93
	Bibliografie Unitate de învățare nr.8	93
9	SCRIPT, JAVASCRIPT	94
	Obiectivele Unității de învățare nr.9	95
	9.1 Ce este JavaScript?	95
	9.2 Inserarea unui JavaScript într-un document HTML	95
	9.3 Cum și când se execută un script într-o pagina WEB	97
	9.4 Atribute de tip Event Handler	99
	Lucrare de verificare Unitate de învățare nr. 7, 8 și 9	100
	Răspunsuri și comentarii la întrebările din testele de autoevaluare	102
	Bibliografie Unitate de învățare nr.9	102
10	EXECUTABILE ȘI MULTIMEDIA ÎN PAGINA WEB	103
	Obiectivele Unității de învățare nr.10	104
	10.1 Java și Java APPLET	104
	10.2 Obiecte ACTIVE X	106

	10.3 Fișiere multimedia în pagina WEB	107
	10.4 Adăugarea clipurilor multimedia la o pagina Web	108
	Răspunsuri și comentarii la întrebările din testele de autoevaluare	110
	Bibliografie Unitate de învățare nr.10	110
11	XML ȘI FOLOSIREA LUI ÎN PAGINA WEB	111
	Obiectivele Unității de învățare nr.11	112
	11.1 Introducere în XML	112
	11.2 Caracteristici ale XML	113
	11.3 Sintaxa XML	114
	11.4 Modul de folosire a XML în pagina Web	116
	Răspunsuri și comentarii la întrebările din testele de autoevaluare	119
	Bibliografie Unitate de învățare nr.11	119
12	FOI DE STIL	120
	Obiectivele Unității de învățare nr.12	121
	12.1 Foi de stil în HTML-CSS	121
	12.2 Cum funcționează stilurile în HTML	122
	12.3 Adăugarea foilor de stil în documentele HTML	123
	12.4 Proprietăți CSS	126
	Lucrare de verificare Unitate de învățare nr. 10, 11 și 12	127
	Răspunsuri și comentarii la întrebările din testele de autoevaluare	129
	Bibliografie Unitate de învățare nr.12	129
	BIBLIOGRAFIE	130
	ANEXA 1 – Lucrări de laborator	131

Proiectul pentru Învățământul Rural

REALIZAREA PAGINILOR WEB

INTRODUCERE

Stimate cursant,

Încă de la început doresc să îți urez bun venit la studiul modulului, destinat deprinderii utilizării unor unelte pentru realizarea paginilor WEB. Acest modul se adresează în principal personalului didactic din aria învățământului obligatoriu, ce activează în mediul rural în vederea obținerii calificării necesare prin intermediul Programului de Educație la Distanță. Sper că acest modul va fi util personalului didactic care dispune de cunoștințe despre universul calculatoarelor cât și oricărei persoane interesate în dobândirea de cunoștințe specifice de introducere în acest domeniu.

Există totuși anumite cunoștințe specifice necesare parcurgerii acestui modul:

- Cunoașterea modalității de organizare și manipulare a informației în format electronic (sistem de fișiere, directoare, copierea, mutarea și deschiderea de fișiere electronice).
- Operații de bază utilizând sistemul de operare Microsoft Windows (pornirea unei sesiuni de lucru, deschiderea unei aplicații, comutarea între mai multe aplicații ce rulează simultan, închiderea în mod corespunzător a sistemului),

Obiectivele modulului:

După studiul acestui modul veți fi suficient de pregătit pentru a fi capabil să:

- Descrieți arhitectura World Wide Web
- Descrieți structura de bază a unui document HTML
- Explicați sintaxa directivelor HTML
- Descrieți modul de specificare a culorilor pentru elementele HTML
- Definiți titluri și paragrafe în cadrul unui document HTML
- Modificați tipurile de caractere, mărimea sau culoarea acestora într-un document HTML
- Creați liste ordonate, neordonate și imbricate cu ajutorul directivelor HTML specializate
- Creați hyperlink-uri în documente HTML
- Folosiți imagini ca hyperlink-uri
- Adăugați imagini la o pagină HTML
- Specificați modul de aliniere al imaginii în pagină și al textului din jurul ei ; dimensiunea imaginii
- Creați un tabel HTML și să modificați proprietățile acestuia
- Creați și să adăugați elemente la un formular
- Ce este un script pe partea de client

- Folosiți un script într-o pagină WEB
- Folosiți un applet Java într-o pagină WEB
- Folosiți un obiect Active X într-o pagină WEB
- Folosiți fișierele multimedia într-o pagină Web
- Folosiți un document XML într-o pagină HTML
- Folosiți sintaxa pentru definirea unei foi de stil

Prezentul modul este în format tipărit fiind conceput pentru educația prin corespondență. Modulul este de lungime medie: 14 ore pentru SI (studiu individual), 7 ore pentru AT (activități tutoriale), 7 ore pentru TC (teme de casă) și 28 de ore pentru AA (activități asistate). Activitățile tutoriale au ca scop stabilirea unui dialog între cursant și tutore în vederea discutării rezultatelor obținute în urma evaluării temelor de casă și nu în ultimul rând pentru lămurirea eventualelor neclarități sau probleme întâlnite de cursant. Temele de casă constau în rezolvarea lucrărilor de verificare care vor fi trimise tutorelui. Cele 28 ore alocate activităților asistate vor fi destinate efectuării celor șapte lucrări de laborator prezente în Anexa 1. Timpul de învățare poate varia în funcție de cunoștințele anterioare ale cursantului despre realizarea paginilor WEB și de cantitatea de muncă dedicată subiectului în studiu, pe care cursantul este dispus să o aloce.

Manualul de față este organizat în 12 unități de învățare, fiecare dintre aceste unități conținând o parte de prezentare teoretică a subiectului tratat, o parte de exerciții și rezolvările acestora. Cele 12 unități de învățare își propun să te învețe ce este și cum poți să construiești o pagină WEB. Prima unitate face o introducere în Internet, World Wide Web, și prezintă pașii și uneltele necesare pentru realizarea primei tale pagini de WEB. Începând cu unitatea de învățare 2 până la unitatea de învățare 8 manualul te va ghida pas cu pas în tainele limbajului HTML. În unitatea de învățare 9 și 10 sunt prezentate tehnologii folosite în World Wide Web pentru a realiza o pagină capabilă să interacționeze cu utilizatorul. La finalul manualului în unitățile de învățare 11 și 12 vom studia tehnici avansate pentru structurarea într-un mod cât mai eficient și mai flexibil a informațiilor ce trebuiesc prezentate în pagina Web.

Instrucțiuni de transmitere a lucrărilor de verificare:

Modulul **Realizarea paginilor WEB** conține patru lucrări de verificare (LV). Fiecare LV va fi transmisă spre corectare tutorelui, la care ai fost alocat, într-un fișier separat, astfel încât să intre în posesia acestuia înainte sau cel târziu la data specificată de calendarul modului. Prima LV trebuie predată după ce încheiați studiul Unității de învățare nr. 3, a doua după încheierea studiului Unității de învățare nr. 6; a treia după încheierea studiului Unității de învățare nr. 9; iar ultima, la sfârșitul modului.

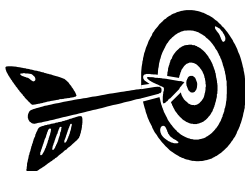
Instrucțiuni de întocmire a lucrărilor de verificare:

LV-urile sunt o componentă importantă a modulului, care îngăduie o corectă evaluare a experienței pe care o căpătați studiind modulul și a capacității dvs. de a-i aplica ideile în practică. Rezolvarea problemelor propuse din lucrările de verificare este asemănătoare cu cea din exemplele din cadrul unității de învățare respective. Ceea ce trebuie să faceți este să prezentați cât mai concis ideile cele mai importante, încercând să nu depășiți limita de **3000 de cuvinte**. Suma maximă a punctelor care vi se acordă pentru tratarea subiectelor unei lucrări de verificare este 25.

Criteriile de evaluare și ponderile evaluării continue și finale:

Notele pe care le veți obține la sfârșitul acestui modul se calculează în funcție de nota pe care o veți primi la examen și de media obținută la LV-uri (evaluare pe parcurs). Cele două componente participă la nota finală cu ponderi de 60% (pentru LV), respectiv 40% (pentru evaluarea finală realizată prin examen). În notarea evaluării pe parcurs, notele celor patru LV-uri intră cu ponderi egale. De aceea, vă recomand insistent să predați **toate** cele patru LV-uri, deoarece pentru o lucrare pe care nu o realizați veți primi nota 0.

Testele de autoevaluare (TA) reprezintă o formă de autoevaluare a cursantului și face parte din tehnologia ID de parcurgere a materialului de studiu. Testele de autoevaluare sunt incluse în manual pentru a te ajuta să îți testezi cunoștințele și felul în care ai înțeles materialul deja parcurs dintr-o unitate de învățare. TA sunt concepute astfel încât să nu îți consume mai mult de câteva minute. Răspunsurile la testele de autoevaluare se vor completa în spațiile libere din chenar, acestea încadrându-se strict în spațiul rezervat.



La începutul fiecărei Unități de învățare vor fi detaliate obiectivele propuse, această secțiune fiind indicată de imaginea alăturată (o săgeată). Modulul nu integrează alte materiale suplimentare de studiu individual, dar recomandă la finalul fiecărei Unități de învățare un decupaj minimal din bibliografia manualului, decupaj necesar pentru aprofundarea și înțelegerea completă a noțiunilor expuse pe durata Unității de învățare.



Pe tot parcursul prezentărilor teoretice, importanța anumitor paragrafe va fi semnalizată în partea stângă a textului prin imaginea unei goarne.

Paragrafele care conțin testele de autoevaluare vor fi semnalizate prin folosirea imaginii unei pene și vor fi încadrate într-un chenar.



În cazul în care nu veți reuși să rezolvați problemele propuse trebuie recitate zonele de text care apar înainte de lucrarea de verificare. În speranța că nu vor exista probleme vă urăm:

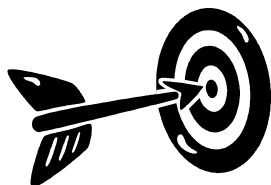
Spor la treabă!

Unitatea de învățare Nr. 1

INTERNET ȘI WORLD WIDE WEB

Obiectivele Unității de învățare Nr.1	5
1.1 Introducere în Internet	5
1.2 Arhitectura World Wide Web	7
1.3 Editarea și vizualizarea unei paginii WEB	8
Răspunsuri și comentarii la întrebările din testele de evaluare	11
Bibliografie	11

Obiectivele Unității de învățare Nr. 1:



Principalele obiective ale Unității de învățare 1 sunt:

- Însușirea unor noțiuni de bază din domeniul rețelelor de calculatoare
- Însușirea unor noțiuni de bază din domeniul Internet
- Descrierea arhitecturii World Wide Web
- Însușirea pașilor necesari într-un ciclu de creare-vizualizare rezultat, pentru o pagină Web

1.1 Introducere în Internet

Înainte de a putea înțelege ce înseamnă și cum funcționează World Wide Web, va trebui să clarificăm anumite noțiuni, definiții, tehnologii pe care se bazează.

Ce este o rețea de calculatoare?

O **Rețea de calculatoare** este o colecție de calculatoare (zeci sau sute) interconectate între ele prin cabluri speciale cu scopul de a putea interschimba sau folosi în comun anumite resurse (fișiere, imprimante, etc.).

Pentru a comunica între ele calculatoarele folosesc un set de reguli care definesc noțiunea de **protocol de comunicație**.

Ce este Internetul?

Rețelele de calculatoare locale pot fi la rândul lor interconectate, formând rețele globale de calculatoare, adică inter-rețele. Cea mai mare inter-rețea cu access public este rețeaua **Internet**.

Definiția de mai sus este foarte generală și mai necesită câteva completări:

- Internetul este un mijloc de comunicare – este un mediu foarte eficient de expunere a ideilor unei audiențe foarte mari.
- Internetul este o resursă de informare – este un imens depozit de informații. Oricând ai nevoie de o informație legată de orice domeniu vei putea găsi undeva publicat pe Internet o lucrare care să te ajute.
- Internetul este o comunitate – face posibilă și foarte eficientă comunicarea între oameni cu aceleași preocupări.

Care sunt serviciile Internet?

Pentru toate facilitățile enumerate mai sus Internetul oferă mai multe metode de access – **servicii Internet** :

- World Wide Web – serviciul ce permite accesul la informația stocată pe un calculator aflat oriunde în lume,

- E-Mail – este un serviciu de mesagerie electronică. Permite schimbul de mesaje între utilizatorii Internet,
- FTP – este prescurtarea de la File Transfer Protocol care înseamnă în limba română protocol pentru transferul fișierelor,
- Telnet – serviciul ce permite accesul la resursele altui calculator din Internet.

TCP/IP protocolul folosit în Internet

Protocolul de comunicație folosit în Internet pentru oricare din serviciile de mai sus este **TCP/IP** (și pentru altele). Acesta definește:

- modul în care calculatoarele ar trebui să fie conectate în Internet,
- modul în care se stabilește o legătură de la un calculator la altul din Internet,
- modul în care sunt transmise date între calculatoarele din Internet.

Ce este o adresă IP?

Pentru a se putea conecta și a fi identificat în rețea un calculator trebuie să aibă o adresă de rețea unică. În Internet această adresă se numește **adresa IP**. O adresă de IP este formată din 4 numere cuprinse între 0 și 255 separate prin caracterul “.”.

66.249.85.99 este un exemplu de adresă IP

Deoarece acest format de adrese este greu de reținut de către om există posibilitatea asocierii unei forme mai prietenoase de adresare fiecărei adrese IP. Această formă de adresare poartă denumirea de **Nume de Domeniu**.

De exemplu pentru adresa IP de mai sus este asociat următorul nume de domeniu:

www.google.com



Test de autoevaluare

1.1 Ce este un protocol de comunicație?

1.2 Cum este identificat un calculator în Internet?

Răspunsurile corecte și comentarii asupra acestora se găsesc la pagina 11.

1.2 Arhitectura World Wide Web

Ce este World Wide Web?

World Wide Web sau pe scurt Web este un sistem de calculatoare în Internet care poate efectua schimb de fișiere într-un anumit format, denumit **HTML** care suportă existența de legături către alte documente sau către fișiere grafice sau audio.

Din ce este format Web-ul?

Acest schimb de fișiere se face folosind tehnologia *client-server* care presupune existența următoarelor entități:

- **Pagini Web** – Fișiere cu un anumit format ce permite organizarea asociativă a informațiilor – HTML,
- **Web Site** – Este o mulțime organizată de pagini Web
- **Server** – Un calculator conectat la Internet pe care sunt stocate paginile Web și pe care rulează un program - **Web server** - care poate servi aceste pagini Web la cerere unui alt calculator din Internet,
- **Client** - Un calculator conectat la Internet pe care rulează un program - **Web Browser** - ce permite comunicația cu HTTP server la care face cerere pentru a primi o resursă Web specificată de un **URL** introdus de utilizator pe care este apoi capabil să îl interpreteze și să îl afișeze.
- **URL** - Fiecare pagină sau resursă WEB are asociată o adresă unică în Internet cunoscută sub acronimul de **URL** (*Uniform Resource Locator*).

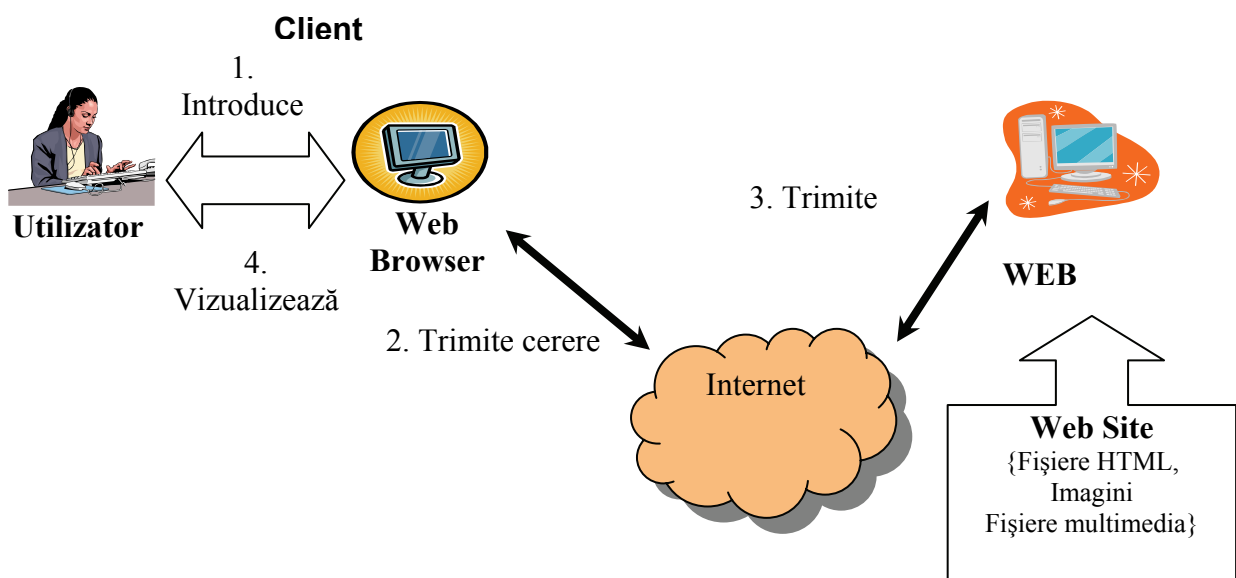


Figura 1.1 Arhitectura World Wide Web

Sensul de circulație al datelor în Web

În Figura 1.1 este prezentat “locul” fiecărei entități descrise mai sus ca fiind parte din World Wide Web, și de asemenea ordinea etapelor pentru accesul la resursele Web:

1. Utilizatorul introduce URL-ul care identifică resursa Web dorită în browserul care rulează pe calculatorul client,

2. Clientul conectat la Internet formulează o cerere de resursă către Web-Serverul identificat de URL,
3. Web Serverul primește, analizează cererea și întoarce ca răspuns la client resursa cerută dacă aceasta există. In caz contrar întoarce un răspuns ce conține un mesaj de eroare.
4. Web Browserul de pe calculatorul client primește răspunsul de la Web-server și îl afișează.



Test de autoevaluare

1.3 Un Web Browser poate rula pe același calculator cu un Web Server?

Răspunsul corect și comentarii asupra acestuia se găsesc la pagina 11.

1.3 Editarea și vizualizarea unei paginii WEB

Un document HTML implementează o pagină Web. Documentele HTML sunt simple fișiere text ceea ce permite ca editarea lor să fie posibilă prin folosirea oricărui editor de texte.

Cum editez o pagină Web?

În prezent există o serie de editoare specializate pentru editarea paginilor HTML. Acestea permit realizarea de documente HTML rapid și ușor doar prin apăsarea a câtorva butoane fără ca utilizatorul să aibă cunoștințe de HTML. Această facilitate este foarte bună pentru utilizatorii începători, însă este foarte importantă cunoașterea temeinică și înțelegerea limbajului. deoarece dezavantajul principal al acestor unelte este că generează adesea cod redundant și uneori chiar incorect. În aceste cazuri fiind necesară intervenția directă in codul generat pentru a corecta aceste neajunsuri.

Exemple de astfel de editoare: Microsoft FrontPage, Macromedia DreamWaver, Eclipse, Bluefish etc.

În acest modul se va folosi editorul de text standard care este livrat cu sistemul de operare Windows: **Notepad**. Acesta precum bănuiești nu are nici o facilitate specială pentru HTML ci este un simplu editor de text.

Dacă nu folosești sistemul de operare Windows poți folosi în mod asemănător orice editor de text disponibil pe sistemul tău:

1. Pentru a porni programul Notepad pe un sistem Windows XP:

- Click pe butonul Start din colțul stânga jos al ecranului
- Click pe meniul All Programs -> Accessories
- Localizați iconița NotePad din acest meniu și faceți click pe ea

2. Odată ce ai pornit aplicația poți începe să scrii documentul HTML. Încearcă să-l scrii pe cel de mai jos:

```
<HTML>
<HEAD>
  <TITLE>Prima mea pagina</TITLE>
</HEAD>
<BODY>
  <H1>Prima mea pagina Web</H1>
  Curand am sa devin un
  <STRONG>expert</STRONG> HTML
  <P>
  Pagina realizata de:
  <CITE>numele tau aici</CITE>
</BODY>
</HTML>
```

3. În momentul în care documentul este complet și dorești să-ti salvezi munca următoarele operații sunt necesare:

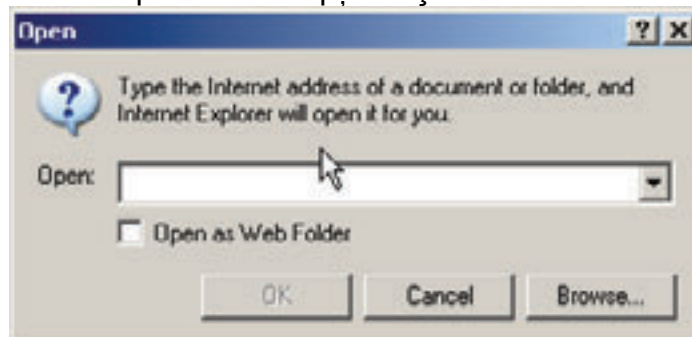
- Localizează în fereastra programului NotePad meniul “File”
- De aici selectează opțiunea “Save As...”
- În fereastra dialog care se deschide introdu numele sub care vrei să salvezi fișierul urmat de extensia .htm sau .html. Spre exemplu alege numele *primapagina.html*.

Atenție este important să specifici extensia deoarece altfel programul NotePad va adăuga automat extensia .txt la numele fișierului.

Cum vizualizez pagina Web creată de mine?

4. Pentru a vizualiza fișierul .html proaspăt creat de tine într-un browser, următoarele operații sunt necesare:

- Pornește programul Microsoft Internet Explorer
- În meniul “File” localizează opțiunea “Open...”
- Click pe această opțiune și se va deschide următorul dialog:



- Click pe butonul “Browse...” și navighează până în folderul în care ai salvat fișierul primapagina.html, selectează-l și. apasă butonul Open

Felicitări! Ai creat prima pagină HTML. Rezultatul ar trebui să fie asemănător cu imaginea de mai jos :

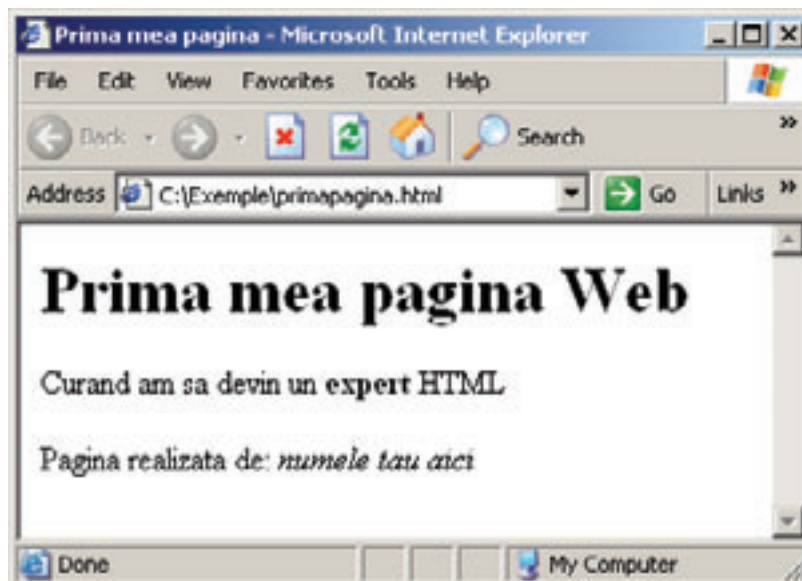


Figura 1.2 – Primul meu document HTML

Dacă rezultatul nu este asemănător cu cel din figura de mai sus înseamnă că nu ai introdus corect codul HTML sau unul dintre pași nu au fost executați corect.

Mai încercă odată!

Răspunsuri și comentarii la întrebările din testele de evaluare



Întrebarea 1.1.

Un protocol de comunicație este un set de reguli folosite de calculatoarele într-o rețea ca să comunice între ele. A se revedea secțiunea 1.1.



Întrebarea 1.2.

Un calculator este identificat în internet cu ajutorul unei **adrese de IP** unică în Internet. Adresele de IP au formă numerică și sunt greu de reținut de către oameni. Din acest motiv o adresă IP poate avea asociat un nume de domeniu cu ajutorul căruia se poate identifica de asemenea un calculator în Internet. A se revedea secțiunea 1.1.



Întrebarea 1.3.

Bineînțeles că da! Nu este cea mai des întâlnită situație însă este posibil. A se revedea secțiunea 1.2.

Bibliografie

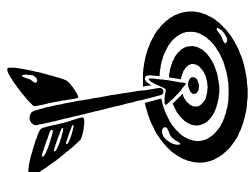
1. Dave Taylor - Crearea paginilor WEB, Editura Teora 1999, pg.21-40
2. Dorin Cârstoiu, Ecaterina Oltean – Introducere în HTML și XML, Editura Printech București 2004, pg.1-7
3. Sabin Buraga – Proiectarea siturilor Web. Design și funcționalitate, Ediția a II-a, Editura Polirom 2002, pg. 13-22

Unitatea de învățare Nr. 2

LIMBAJUL HTML

Obiectivele Unității de învățare Nr. 2	13
2.1 Introducere în HTML	13
2.2 Sintaxa Directivelor HTML	13
2.3 Structura de bază a unui document HTML	19
Răspunsuri și comentarii la întrebările din testele de evaluare	22
Bibliografie	22

Obiectivele Unității de învățare Nr. 2:



Principalele obiective ale Unității de învățare nr. 2 sunt:

- Definirea noțiunii HTML
- Descrierea structurii de bază a unui document
- Explicarea sintaxei directivelor HTML
- Descrierea modului de specificare a culorilor pentru elementele HTML

2.1 Introducere în HTML

Definiție

HTML este limbajul folosit pentru a structura informația în documentele World Wide Web. Numele său reprezintă inițialele următoarelor cuvinte în limba engleză: “*HyperText Markup Language*”.

Limbajul HTML este compus din instrucțiuni de afișare, care vor fi folosite de către browser pentru a determina cum anume să afișeze informația utilă din document. Aceste instrucțiuni de afișare poartă denumirea de **directive HTML** sau, **tags** în limba engleză.

Definiție

Un **element** este o componentă fundamentală din structura unui document. O pereche de directive delimitează un element HTML, iar informația cuprinsă între directiva de început și cea de sfârșit delimitează conținutul elementului. Un element poate conține text simplu sau alte elemente.

2.2 Sintaxa Directivelor HTML

În cazul documentelor HTML mai mult de jumătate din textul sursă nu este afișat de către browser. Mai precis ceea ce lipsește este textul cuprins înăuntrul caracterelor pereche “<” și “>”, aceasta datorită faptului că în limbajul HTML ceea ce este cuprins între aceste caractere sunt interpretate ca fiind **directive HTML** (tags).

O directivă HTML constă dintr-un **nume** care, opțional, este urmat de o **listă de attribute** ale directivei HTML, toate acestea fiind plasate între perechea de caractere “<” și “>”. Attributele unei directive HTML pot lua diferite valori și permit autorului documentului să-i modifice comportamentul. Iată un exemplu:

Dacă directivei <BODY> i se adaugă atributul “BGCOLOR” cu o valoare dorită atunci browserul va afișa pagina respectivă pe un fundal de culoarea specificată de valoarea atributului. Pentru a afișa pagina pe fond roșu vom scrie:

```
<BODY BGCOLOR="red" >
```

De obicei numele directivei este intuitiv fiind reprezentat de un cuvânt sugestiv pentru funcția pe care o are.

Directive HTML pereche

Directivele HTML apar de regulă în perechi, de exemplu <html></html> sau <body></body>, prima directivă fiind de început iar cea de-a doua de sfârșit. Un text aflat între aceste directive va respecta funcționalitatea directivei respective.

```
<B>Text afișat cu caractere Bold</B>
```

Figura 2.1 – Mod de folosire al directivelor HTML pereche.

Exemplu de mai sus prezintă modul în care se pot folosi directivele HTML pentru afișarea textului cu caractere de tip Bold.

O directivă de sfârșit are același nume cu directiva corespunzătoare de început, dar este precedată de caracterul "/". Tagurile de sfârșit nu conțin niciodată atribute.

Directive HTML fără pereche

O parte din directivele HTML nu au directive pereche de sfârșit. Acestea poartă denumirea de *stand-alone*. Un exemplu de directivă stand-alone este . Aceasta face ca browserul să plaseze o imagine (specificată de valoarea atributelor directivei) în pagină.

Atributele

Atributele sunt adăugate directivelor HTML pentru a extinde sau pentru a modifica comportamentul acestora. Atributele apar întotdeauna în perechi de forma nume/valoare. Poți adăuga mai multe atribute la același tag, separând fiecare tag printr-unul sau mai multe spații. Ordinea de apariție nu este relevantă. Valorile atributelor au lungimea limitată la 1024 caractere.

```
<IMG SRC="images/smile.gif" ALT="vesel">
```



Valoarea unui atribut trebuie scrisă de regulă între caracterele ‘ ‘ și ‘ ‘, iar în cazurile speciale în care însuși valoarea atributului conține

caracterul ghilimele se folosesc apostroafe, ca în exemplu de mai jos:

```
<MAP NAME='nume cu caracterul " ' >
```

Dacă valoarea este un singur număr sau cuvânt și conține numai litere (a-z) sau cifre (0-9) sau caracterele "." și "-", nu este obligatorie încadrarea între ghilimele sau apostroafe a valorii atributului, ea putând fi plasată imediat după semnul "=".



Totuși atunci când nu ești sigur că regula de mai sus este respectată este o idee bună folosirea încadrării între ghilimele a valorii atributelor.

Mai jos sunt prezentate câteva exemple de directive care conțin atribute. Fii atent la faptul că prima directivă conține atât atribute a căror valoare este încadrată de ghilimele cât și atribute care nu necesită această încadrare:

```
<IMG SRC="img/myimg.gif" ALIGN=right WIDTH=45
HEIGHT=60>
<BODY BGCOLOR="#000000">
<FONT FACE="Arial, Helvetica" SIZE=4>
```

În afară de textele care conțin caractere obișnuite, HTML oferă posibilitatea de a insera și afișa caractere care în mod normal nu ar putea fi incluse în document sau care au un scop predefinit în limbajul HTML – cum ar fi, spre exemplu, caracterul "<" care este folosit în limbajul HTML pentru a semnală începutul unei directive.



Test de autoevaluare

2.1 Ce este un element HTML?

2.2 Ce este o directivă HTML?

2.3 Ce rol au atributele unui element?

Răspunsurile corecte și comentarii asupra acestora se găsesc la pagina 22.

Caractere speciale

În limbajul HTML, caracterul ampersand ("&") instruește browserul să insereze un caracter special în funcție de codul sau identificatorul

ce îl urmează. O astfel de construcție poartă denumirea de “*character entities*” și este formată din trei părți:

- Caracterul ampersand &
- Un nume predefinit al caracterului sau semnul # urmat de codul caracterului
- Caracterul “;” (punct și virgulă)

Spre exemplu pentru a afișa caracterul “<” se va scrie în text următoarea construcție: <. Similar > introduce caracterul “>” iar & introduce caracterul ampersand.



Numele predefinite pentru caracterele speciale trebuie scrise întotdeauna cu litere mici.

Se pot de asemenea introduce caractere speciale pentru care nu există identificatori predefiniți, folosindu-se semnul “#” urmat de codul ASCII al caracterului. Spre exemplu următoarea construcție < are același efect ca < , adică introducerea caracterului “<”.

Specificarea culorilor în HTML

Culoarea anumitor elemente din cadrul documentului HTML se poate specifica folosindu-se anumite directive HTML sau attribute ale acestora. Există 2 metode de specificare a culorilor în cadrul paginilor WEB:

- prin valoarea codului RGB
- folosind numele culorii

Cel mai adesea folosită și mai flexibilă modalitate este folosirea codului RGB. Fiecare culoare RGB conține 3 valori corespunzătoare concentrației fiecărei componente roșu (**RED**), verde (**GREEN**) sau albastru (**BLUE**) din cadrul culorii alese. Aceste valori se încadrează între 0 și 255. Specificarea concentrației maxime (R=255, G=255, B=255) în fiecare componentă rezultă în culoarea albă, iar specificarea valorii 0 pentru toate componentele rezultă în culoarea neagră.

Majoritatea utilităților de grafică au dialog care permite în mod grafic alegerea culorii dorite furnizând pentru aceasta codul RGB corespunzător. În figura de mai jos este prezentat un astfel de dialog:



Se observă în imaginea de mai sus că în funcție de culoarea aleasă se generează concentrația corespunzătoare fiecărei componente de culori în cazul nostru (R=237, G=15, B=99).

Odată identificate aceste concentrații pentru culoarea dorită vor trebui transformate în valori hexazecimale echivalente pentru a le putea folosi ca valoare pentru un atribut în cadrul unei directive HTML. Aceasta este sintaxa:

```
"#RRGGBB"
```

Cu ajutorul acestor valori se poate specifica orice culoare din spațiu de culori "true color".

Cea de-a doua metodă este a identifica culorile după nume. Există un set de 140 de culori predefinite.

Comentarii HTML

Comentariile sunt folosite în cadrul unui document HTML pentru a documenta codul HTML scris în pagina respectivă. Comentariile nu sunt afișate de către browser. În procesul de creare sau de mentenanță a paginilor de Web aceste comentarii se pot dovedi foarte utile oferind persoanei care creează sau modifică pagina informații suplimentare despre ce anume s-a dorit a fi realizat în acea pagină și eventual informații suplimentare despre modul în care se realizează acel obiectiv.

Pentru a defini un comentariu HTML se folosește o directivă specială cu următoarea sintaxă:

```
<!--Textul comentariului -->
```

sau

```
<!--Textul comentariului  
pe mai multe randuri -->
```

Trebuie să existe obligatoriu un spațiu după `<!--` și un spațiu înainte de `-->`, în rest se poate folosi aproape orice caracter sau combinație de caractere în interiorul comentariului fără a influența modul în care pagina va fi afișată de către browser.

Imbricarea directivelor HTML

O directivă HTML poate fi folosită în interiorul unei alte directive HTML cu scopul de a putea aplica efectul ambelor directive asupra unui anumit element. Spre exemplu dacă se dorește ca un anumit cuvânt din text să fie și în format italic și bold în același timp pentru a-l scoate în evidență se folosește această metodă de imbricare a directivelor după cum este arătat în exemplul de mai jos:

```
...  
Numele meu este: <B><I>Ana</I><B>.  
...
```

Va avea ca rezultat:

Numele meu este: **Ana**



Informații ignorete de browser

Totuși această imbricare trebuie făcută având mereu grijă ca ordinea de închidere a tagurilor să fie inversă cu ordinea de deschidere – **ultimul tag deschis să fie primul închis.**

Există anumite informații, caractere sau chiar directive pe care browserele nu le afișează. Acestea sunt:

- Sfârșitul de linie (CR/LF)
Caracterele de sfârșit de linie folosite în interiorul documentului HTML sunt ignorate și browserul va organiza textul în funcție de dimensiunea ferestrei. Trecerea la linie nouă se va face explicit prin folosirea directivelor `<P>` sau `
`
- Taburile sau spațiile multiple
Dacă browserul întâlnește un caracter Tab sau mai multe caractere “spațiu liber” consecutive îl va afișa ca un singur caracter spațiu liber. Pentru a introduce spații suplimentare se poate folosi caracterul special: ` `.
Spre exemplu:

```
Text,    cu    multe    spatii  
Va fi afișat ca:  
Text, cu multe spatii
```

- Directivele `<p>` care apar de mai multe ori.
Dacă directiva `<P>` este folosită de mai multe ori consecutiv fără conținut, browserul va interpreta această construcție ca un singur paragraf și îl va afișa ca atare.
- Comentariile
Browserele nu vor afișa textul cuprins între caracterele `<!--` și `-->`
- Directivele necunoscute
Browserele ignoră directivele pe care nu le suportă sau pe cele incorect specificate. În funcție de tipul browserului comportamentul în cazul acestor directive este fie de a nu afișa nimic când sunt întâlnite, fie de a afișa conținutul directivei ca simplu text.



Test de autoevaluare

2.4 Ce sunt caracterele speciale și cum pot fi acestea folosite în cadrul unui document HTML?

2.5 Identificați care din codurile de mai jos reprezintă codul RGB care specifică culoarea roșie :

- a> (255, 0, 0) sau #FF0000
- b> (0, 0, 0) sau #000000
- c> (0, 255, 0) sau #00FF00
- d> (255, 255, 0) sau #FFFF00

Răspunsurile corecte și comentarii asupra acestora se găsesc la pagina 22.

2.3 Structura de bază a unui document HTML

Documentele realizate cu ajutorul limbajului HTML au extensia htm sau html și sunt simple fișiere de tip ASCII (text). Un document HTML conține text util (conținutul efectiv al paginii) și directive HTML care sunt folosite pentru a defini structura, modul de afișare și comportamentul conținutului.

Structura unui document HTML

Fiecare document HTML începe cu directiva `<HTML>` care specifică browserului faptul că informația care urmează în fișier este în format HTML. Ultima directivă este `</HTML>` și marchează sfârșitul documentului HTML.

După această primă directivă urmează un bloc marcat de directivele `<HEAD>` și `</HEAD>` care reprezintă antetul documentului, informațiile din cadrul acestui bloc nefiind afișate de către browser.

Următoarea secțiune, marcată de directivele `<BODY>` și `</body>`, conține informația afișată de către browser. Această secțiune include informația utilă, și directive HTML care specifică browserului modul în care acesta o va afișa. Directivele HTML pot de asemenea să facă referire la diverse fișiere externe cum ar fi fișiere de imagine sau alte documente HTML.

Structura de bază a unui document HTML este următoarea:

```
<HTML>
```

```
<HEAD>
  Informații generale despre documentul HTML
</HEAD>
<BODY>
  Corpul principal al paginii
</BODY>
</HTML>
```

Directiva HEAD

Antetul unui document HTML conține informații generale despre conținutul și structura documentului. Directiva `<HEAD>` nu conține atribute ci servește drept cadru pentru alte directive: `<BASE>`, `<ISINDEX>`, `<LINK>`, `<META>`, `<NEXTID>`, `<OBJECT>`, `<SCRIPT>`, `<STYLE>` și `<TITLE>`. Le vom descrie pe scurt pe cele mai des folosite.

Directiva `<TITLE>` - reprezintă titlul documentului. Textul definit în interiorul ei va apărea ca titlu pentru fereastra de browser care afișează pagina și de asemenea va fi folosit atunci când pagina este adăugată la în meniul "Favorites" sau "Bookmarks". De asemenea acest text va fi folosit de către motoarele de căutare atunci când își adaugă pagina în baza de date. Pentru toate aceste motive este important ca acest text să fie cât mai sugestiv.

Directiva `<BASE>` - stabilește calea de bază pe care serverul de WEB o va folosi pentru toate legăturile definite în interiorul documentului. Despre aceasta și despre legături vom vorbi în unitatea de învățare 5

Directiva `<SCRIPT>` - conține cod Java Script sau VB Script despre care vom vorbi în unitate de învățare 9 (

Directiva `<STYLE>` - conține informații despre stilurile folosite de către tabela de stiluri (CSS) despre care vom vorbi în unitate de învățare 12.

Directiva BODY

Corpul documentului HTML este definit de către directiva pereche `<BODY></BODY>`. Conținutul lui poate fi un singur paragraf, o imagine sau o combinație complexă de imagini, tabele, obiecte multimedia, text.

Elementul BODY are un rol foarte important în ceea ce privește imaginea de ansamblu a pagini HTML, deoarece permite definirea unor parametrii globali cum ar fi: culoarea sau imaginea de fundal a paginii sau culoarea textului și a legaturilor din pagină. Acești parametrii globali se definesc cu ajutorul atributelor directivei `<BODY>`.

Atributul BGCOLOR permite stabilirea culorii de fundal a paginii. În exemplul de mai jos se definește culoarea albastru ca fundal:

```
<BODY BGCOLOR="blue" ></BODY>
```

Atributul TEXT este folosit pentru a defini culoarea textului normal din cadrul documentului. Culoare implicită a textului este negru. În exemplul de mai jos este setată culoarea de fundal alb și culoarea textului roșu.

```
<BODY BGCOLOR="#FFFFFF" TEXT="#FF0000" >
```



La alegerea culorii textului trebuie avut în vedere ca acesta să poată fi distins ușor de fundal.

Atributele LINK, ALINK, VLINK permit controlul culorii legăturilor în funcție de starea acestora după cum este descris mai jos:

- LINK – stabilește culoarea cu care vor fi afișate inițial legăturile și implicit este Albastru
- VLINK – stabilește culoarea cu care vor fi afișate legăturile care au mai fost vizitate implicit este Purpuriu
- ALINK – stabilește culoarea cu care este afișată legătura deja vizitată.

În exemplul de mai jos se va stabili următoarea schemă de culori pentru pagină: culoarea de fundal alb, culoarea textului va fi roșu, culoarea legăturilor albastru, culoarea legăturilor vizitate va fi magenta, iar a legăturilor active va fi verde

```
<BODY      BGCOLOR="#FFFFFF" TEXT="#FF0000"
           LINK="#0000FF" VLINK="#FF00FF"
           ALINK="#00FF00" >
</BODY>
```

Atributul BACKGROUND – cu ajutorul acestui atribut al elementului BODY se poate seta ca fundal pentru documentul HTML o imagine. Imaginea specificată ca valoare a atributului trebuie să fie în format .gif sau .jpg, și va fi poziționată astfel încât să acopere întreaga arie a paginii. Dacă o singură imagine nu este destul de mare pentru a satisface această condiție atunci aceasta va fi replicată și spațiul rămas neocupat va fi umplut la dreapta și în jos cu aceste replici până când fereastra browserului este complet ocupată de imagine.

```
<BODY      BGCOLOR="#FFFFFF"
           BACKGROUND="logo.jpg" >
```

În exemplul de mai sus s-a specificat imaginea care va fi folosită ca fundal și de asemenea culoarea de fundal utilizată de browser până în momentul încărcării imaginii.

Răspunsuri și comentarii la întrebările din testele de evaluare



Întrebarea 2.1.

Un element este o componentă fundamentală din structura unui document. O pereche de directive delimitează un element HTML, iar informația cuprinsă între directiva de început și cea de sfârșit delimitează conținutul elementului. Revedeți secțiunea 2.1.



Întrebarea 2.2.

Directivele HTML sunt instrucțiuni folosite de către browser pentru a determina cum anume să afișeze informația utilă din document. Revedeți secțiunea 2.1.



Întrebarea 2.3.

Atributele sunt adăugate directivelor HTML pentru a extinde sau pentru a modifica comportamentul acestora. Revedeți secțiunea 2.2.



Întrebarea 2.4.

În afară de textul normal HTML oferă posibilitate afișării unor caractere care în mod normal nu sunt afișate de către browser deoarece acestea sunt caractere cheie folosite pentru identificarea unor construcții HTML. Spre exemplu: caracterele “<” sau “>”. Acestea se pot specifica în cadrul unui document HTML cu ajutorul caracterelor speciale definite printr-o construcție de tipul: caracterul “&” + cod caracter + caracter “;”. Revedeți secțiunea 2.2.



Întrebarea 2.5.

Codul RGB este : (255, 0, 0). Varianta corectă de răspuns : a). Revedeți secțiunea 2.2.

Indicații la problemele propuse

Problemele propuse în lucrările de verificare se fac după modelele de exemple prezentate în unitatea de învățare sau după tipicul acestora.

Bibliografie

1. Dave Taylor - Crearea paginilor WEB, Editura Teora 1999, pg.53-70
2. Dorin Cârstoiu, Ecaterina Oltean – Introducere în HTML și XML, Editura Printech București 2004, pg.17-20, pg.28-29, pg.48-49

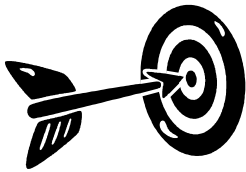
Unitatea de învățare Nr. 3

FORMATAREA TEXTELOR IN LIMBAJUL HTML

Obiectivele Unității de învățare Nr.3	24
3.1 Limbajul HTML și formatarea textelor	24
3.2 Titluri și paragrafe HTML	24
3.3 Definirea caracteristicilor fontului	29
3.4 Formatarea fizică a textelor	32
3.5 Formatarea logică a textelor	33
3.6 Preformatarea textului cu ajutorul directivei <PRE>	34
Lucrare de verificare a Unităților de învățare nr. 1, 2 și 3	35
Răspunsuri și comentarii la întrebările din testele de evaluare	37
Bibliografie	37

Obiectivele Unității de învățare Nr. 3:

Principalele obiective ale Unității de învățare Nr. 3 sunt:



După parcurgerea acestei Unității de învățare veți fi capabili:

- Să definiți titluri și paragrafe în cadrul unui document HTML
- Să modificați tipurile de caractere, mărimea sau culoarea acestora într-un document HTML
- Să folosiți directivele HTML pentru formatarea logică a textului
- Să folosiți directivele HTML pentru formatarea fizică a textului

3.1 Limbajul HTML si formatarea textelor

Pentru ca informațiile dintr-un document să fie asimilate cât mai ușor de către cititorii săi și pentru ca prezentarea lor să aibă succes maxim este foarte important ca textul să fie organizat într-o formă cât mai atractivă. HTML oferă mijloace eficiente pentru a structura și a înfrumuseța un text.

Ca autor al unui document HTML ai două opțiuni în ceea ce privește textul pe care dorești să îl afișezi. Prima este să îl scrii așa cum este și a doua să în incluzi între anumite directive HTML. Spre exemplu dacă vrei să afișezi textul "La mulți ani!" poți pur și simplu să îl tastezi în cadrul documentului și va fi afișat fără nici o problemă. Însă dacă vrei ca acest text să fie afișat într-un anumit mod – spre exemplu vrei ca textul să fie scris cu caractere roșii de dimensiunea mai mare decât cea normală a textului - va trebui să specifici explicit browserului modul în care vrei ca textul tău să fie afișat cu ajutorul directivei HTML pentru formatarea a textelor.

3.2 Titluri și paragrafe HTML

**Directiva
<Hn></H>**

În cadrul oricărui document este necesară definirea unui titlu principal și a mai multor titluri pentru fiecare din subsecțiunile documentului. Limbajul HTML oferă o modalitate ușoară de definire a titlurilor de diferite dimensiuni. Titlurile se definesc folosindu-se directiva <Hn>, unde *n* este un număr cuprins între 1 și 6. Valoarea 1 indică realizarea unui titlu cu cea mai mare dimensiune în timp ce 6 va crea un titlu cu cea mai mică dimensiune.

Directivile pentru titlu au un atribut opțional care poate schimba modul de aliniere. Acest atribut este "align" și poate lua una din valorile: "left" - stânga, "right" - dreapta sau "center" - centru. Folosirea valorii "left" pentru atributul align este însă redundantă deoarece alinierea implicită folosită pentru cazurile în care nu este specificată este "left".

Pentru a vedea modul în care este poziționat textul din interiorul directivelor de titlu HTML și pentru a compara diferite dimensiuni ale acestora vom considera codul HTML de mai jos:

```
<HTML>
<HEAD>
  <TITLE>
    Exemplu titluri
  </TITLE>
</HEAD>
<BODY>
  <H1 ALIGN="CENTER">
    Titlul H1 poziționat central
  </H1>
  <H2>
    Titlul de tip H2.
  </H2>
  <P>
    Acesta nu este un titlu ci este text normal
  </P>
  <H3>
    Titlul de tip H3.
  </H3>
  <H4 ALIGN="RIGHT">
    Titlul H4 aliniat la dreapta
  </H4>
  <H5>
    Titlul de tip H5.
  </H5>
  <H6>
    Titlul de tip H6.
  </H6>
</BODY>
</HTML>
```

Rezultatul va fi în browser de forma:

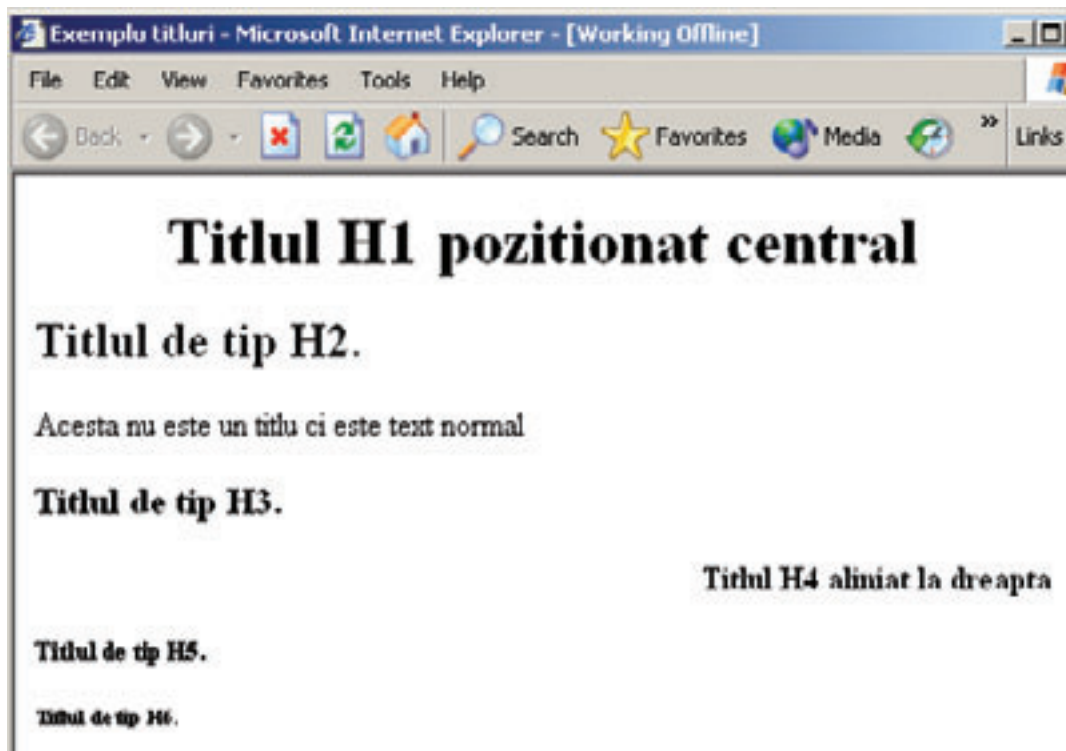


Figura 3. 1 Exemple titluri HTML



- Nu uita să folosești directiva de sfârșit pentru titlu deoarece browserul nu va include implicit una.
- Un document HTML bun ar trebui să folosească titlurile în ordinea ierarhică. H1 ar trebui folosit pentru primul nivel de titlu, H2 pentru al doilea ș.a.m.d
- Se va evita omiterea unui nivel ierarhic la un moment dat, adică H3 nu ar trebui să apară imediat după H1, ci se va folosi H2 în loc.

Directiva <P></P>

O metoda eficientă de a face o pagină cât mai ușor de citit este despărțirea acesteia în paragrafe. Spre deosebire însă de documentele scrise cu majoritatea procesoarelor de text existente, caracterele de linie nouă nu sunt luate în considerare. În fapt, orice tip de spațiere –linie nouă, taburi sau spații – vor fi transformate într-un singur spațiu liber în momentul afișării documentului de către browser.

Pentru a indica începerea sau terminarea unui paragraf și deci trecerea la o linie nouă se folosește directiva HTML <P></P>.

Directiva de încheiere </P> poate fi omisă. Aceasta deoarece majoritatea browserelor la întâlnirea unei noi directive <P> consideră automat că paragraful anterior s-a terminat.

Elementele de tip paragraf permit adăugarea de text la un document iar dimensiunea liniei va fi ajustată în funcție de dimensiunea ferestrei browserului. Această ajustare se face automat de către browser în momentul vizualizării paginii.



- Inserarea mai multor elemente `<P>` fără conținut succesiv va avea ca rezultat în browser afișarea unei singure linii libere în locul lor.
- Pentru a introduce mai multe rânduri libere consecutiv se poate folosi directiva `
` descrisă mai jos.
- Deoarece textul este reformatat de câte ori utilizatorul își dimensionează fereastra se va evita introducerea manuală a rândurilor libere.

Directiva `
`

Adeseori este necesară trecerea forțată la o linie nouă fără a termina însă paragraful curent. Deoarece caracterele de linie nouă sunt ignorate în HTML această trecere se face folosind directiva HTML `
`. Elementele `
` nu au directivă de sfârșit obligatorie deoarece elementul marchează o poziție și nu are nici un conținut care să trebuiască delimitat.

În exemplul de mai jos vom urmări modul de folosire a directivelor HTML `<P>` și `
` precum și diferențele de poziționare în cazul folosirii fiecăreia:

```
<HTML>
<HEAD>
  <TITLE>
    Exemplu Paragraf si Break line
  </TITLE>
</HEAD>
<BODY>
  <H1>Titlul</H1>
  <P> Textul paragrafului 1 </P>
  <P> Textul paragrafului 2
  <BR>
    Linie noua in cadrul paragrafului 2
  <BR> A doua linie noua in cadrul paragrafului 2
  </P>
  <P> Textul paragrafului 3 </P>
</BODY>
</HTML>
```

Va avea ca rezultat în browser:

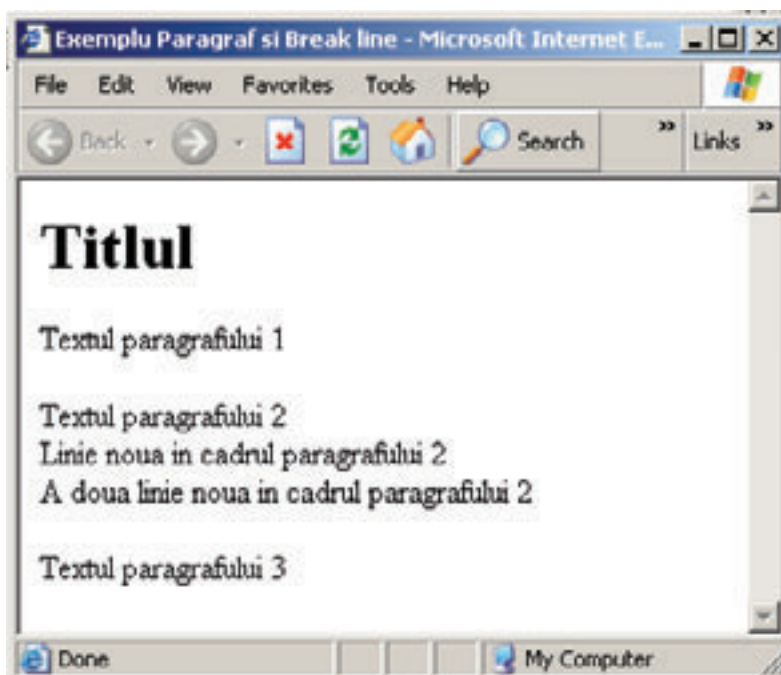


Figura 3.2. Folosirea directivelor <P> și

Se observă faptul că în cazul directivei <P> se inserează automat o linie liberă înainte și după terminarea paragrafului, iar în cazul folosirii directivei
 nu este introdus nici un spațiu suplimentar.

Directiva <HR>

Un alt mijloc de a separa diferitele secțiuni ale unui document este folosirea directivei <HR>. La întâlnirea acestei directive browserul trece automat la linie nouă și desenează o linie orizontală. Caracteristicile linie orizontale pot fi controlate de către atributele directivei.

Atributul **WIDTH**. Specifică lungimea liniei fie ca dimensiune absolută în pixeli, fie ca procent din lungimea ferestrei browserului.

Atributul **ALIGN**. Poate lua una din valorile *left*, *center* sau *right* și specifică modul de aliniere a liniei. Acest atribut nu are semnificație fără existența atributului WIDTH.

Atributul **NOSHADE**. Implicit linia este desenată în relief având umbră. Prezența acestui atribut indică faptul că linia va fi desenată fără umbră.

Atributul **SIZE**. Specifică grosimea – numărul de pixeli - cu care va fi desenată linia.

În exemplu de mai jos vom specifica o desenare a unei linii alinate la dreapta desenată fără umbră și cu dimensiunea 20% din fereastra browserului și grosimea de 10 pixeli:

```
<HR
  NOSHADE
  ALIGH="RIGHT"
  SIZE="10"
  WIDTH="20%" >
```



Test de autoevaluare

3.1. Directiva
 se folosește :

- a> pentru a forța trecerea la o linie nouă
- b> forțează începerea unui paragraf nou
- c> pentru introducerea mai multor linii libere succesive
- d> introducerea unei linii orizontale

3.2. Cum formatează textul elementele de tip paragraf?

3.3. Cum se desenează în HTML o linie orizontală centrată cu dimensiunea jumătate din lungimea ferestrei browserului

Răspunsurile corecte și comentarii asupra acestora se găsesc la pagina 37.

3.3 Definirea caracteristicilor fontului

Directiva <BASEFONT>

Directiva <BASEFONT> definește dimensiunea implicită a fontului pentru întreg textul folosit în interiorul elementului <BODY>. Titlurile H1-H6 nu sunt afectate de această directivă. Directiva BASEFONT are un singur atribut: **SIZE** care definește dimensiunea implicită a fontului folosit în cadrul documentului. Valoarea acestui atribut este un întreg între 1 și 7. Dacă nu este specificată acest atribut are valoarea 3.

Pentru a schimba dimensiunea implicită a fontului la 5 scriem:

```
<BASEFONT SIZE="5">
```



- Schimbarea fontului de bază ar trebui să fie făcută cu grijă și având un motiv clar deoarece utilizatorul are posibilitatea oricum să definească dimensiunea preferată a textului din opțiunile browserului.
- Acest element nu are directivă de încheiere
- Directiva BASEFONT afectează textul din cadrul elementului BODY dar nu și titlurile. Alegerea unei dimensiuni prea mari pentru fontul de bază poate duce la situația în care fontul

pentru titlu are dimensiuni mai mici decât fontul textului propriu-zis producând astfel confuzie.

- Directiva BASEFONT afectează întreg textul ce urmează după ea și până la încheierea elementului BODY.
- Nu este recomandat folosirea ei pentru schimbarea dimensiunii fontului unui bloc de text a caracterelor individuale, pentru aceasta fiind disponibile alte directive precum FONT, BIG sau altele despre care vom discuta mai târziu în această unitate de învățare

Directiva

Pentru a stabili caracteristici ale fontului precum: dimensiune, culoare, tip pentru un bloc de text se folosește directiva ****. Schimbarea caracteristicilor fontului pentru o parte din caractere sau pentru anumite cuvinte este o metodă bună pentru a scoate în evidență ceva important sau pentru a crea efecte interesante. Directiva de sfârșit este obligatorie.

Atributul **SIZE**. determină noua dimensiune a fontului pentru textul elementului. Valoarea atributului este un întreg între 1 și 7 specificând astfel dimensiunea absolută a fontului. Prin adăugarea prefixului "+" sau "-" este specificată dimensiunea relativă față de fontul de bază.

Pentru specificarea dimensiunii în valoarea absolută vom scrie:

```
<FONT SIZE="7">A </FONT>fost odata ca niciodata...
```

care va avea care rezultat în browser:

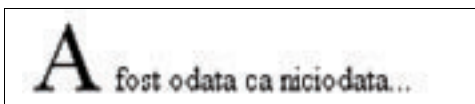


Figura 3. 3. Exemplu specificare dimensiune font în valoare absolută



- În exemplu de mai jos vom încerca să creem un efect de perspectivă pentru exclamația "URAAA" folosind atributul SIZE pentru directiva font și specificând dimensiunea în format relativ:

```
<P>
U
<FONT SIZE="+1">R
<FONT SIZE="+2">A
<FONT SIZE="+3">A
<FONT SIZE="+4">A
</FONT>
</FONT>
</FONT>
</FONT>
```

Va avea ca rezultat în browser:



Figura 3. 3. Exemplu specificare dimensiune font în valoare relativă

În cazul folosirii intercalate a directivelor cu valoarea atributului SIZE specificată în valoare relativă, efectul nu este cumulativ ci referința este dimensiunea fontului de bază.

Atributul **COLOR**. Definește culoarea textului din interiorul elementului FONT. Specificarea culorii se face folosindu-se codul RGB¹ sau numele predefinit al culorii dorite.

Pentru a afișa un text cu culoarea galbenă și dimensiunea absolută 5 vom scrie:

```
...
<P>
  Ultimul cuvânt are culoarea
  <FONT COLOR="#0000FF" SIZE="5">
    Albastra
  <FONT>.
</P>
```

Și rezultatul în browser va fi:

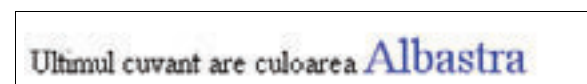


Figura 3. 4. Exemplu specificare culoare font

Atributul **FACE**. Permite schimbarea tipului de font folosit pentru afișarea textului. Dacă fontul nu este suportat de către browser va fi folosit tipul implicit de font. Se pot specifica mai multe tipuri de font despărțite prin virgulă și un spațiu.

```
<FONT FACE="arial, courier, garamond">
```

În acest caz dacă primul tip de font nu este suportat de către browser se va încerca cu cel de al doilea ș.a.m.d.

În exemplu prezentăm o listă de fonturi uzuale suportate de către majoritatea browserelor:

¹ Vezi unitatea de învățare numărul 2

Acesta este font de tip "Arial"
Acesta este font de tip "Algerian"
Acesta este font de tip "Courier"
Acesta este font de tip "Desdemona"
Acesta este font de tip "Garamond"
Acesta este font de tip "Modern"

Figura 3. 5. Tipuri de font uzuale



- Folosește directiva FONT pentru a schimba caracteristicile fontului pentru un număr oarecare de cuvinte sau paragrafe, iar pentru schimbarea întregii pagini folosește BASEFONT.
- Este indicat să eviți folosirea schimbării dimensiunii fontului folosind valori extreme deoarece aceasta poate face documentul greu de citit.



Test de autoevaluare

3.4 În ce cazuri este recomandată a se folosi directiva `<BASEFONT>` în locul directivei ``

Răspunsul corect și comentarii asupra acestuia se găsesc la pagina 37.

3.4 Formatarea fizică a textelor

Directivile pentru formatare fizică forțează browserul să afișeze textul elementului respectiv într-un anumit format dorit de creatorul documentului HTML respectiv neținând cont de setările din browser ale utilizatorului respectiv.

Dacă una din aceste directive nu este suportată de către browser ea va fi ignorată deoarece browserul nu are nici o altă alternativă de afișare. În cele ce urmează este prezentată o listă cu aceste directive și este pe scurt descrisă funcționalitatea fiecăreia.

Directivă	Funcționalitate
	Afișează textul îngroșat
<BIG>	Afișează textul cu o unitate mai mare decât dimensiunea fontului de bază
<SMALL>	Afișează textul cu o unitate mai mic decât dimensiunea fontului de bază
<I>	Afișează textul înclinat.
<U>	Afișează textul subliniat
<SUB>	Afișează textul sub formă de indice inferior
<SUP>	Afișează textul sub formă de indice superior
<TT>	Afișează textul cu caractere teleprinter
<BLINK>	Afișează textul pâlpâind

Toate directivele de formatare fizică de mai sus necesită specificarea explicită a directivei pereche de sfârșit.

3.5 Formatarea logică a textelor

Formatarea logică presupune, spre deosebire de cea fizică, faptul că fiecare browser va formata textul afectat de directivele logice în funcție de posibilitățile platformei pe care rulează. Prin urmare formatarea propriu-zisă nu va fi neapărat la fel de la un browser la altul, însă efectul va fi același. Spre exemplu pentru unele browsere textul din cadrul directivei va apărea îngroșat în timp ce pentru altele va apărea italic. Creatorul paginii se va concentra în acest caz mai mult pe definirea semnificației textului decât asupra modului în care acesta va fi formatat în browser.

Să discutăm despre fiecare element în parte:

<ACRONYM> indică faptul că textul inclus este un acronim, adică un cuvânt format din inițialele unor cuvinte care fac parte dintr-o expresie sau un nume.

<CITE> indică faptul că textul inclus este un citat bibliografic. Prin convenție acest text este afișat înclinat.

<CODE> este folosit pentru a afișa exemple de cod sursă. Textul acestui element este afișat cu font de tip teleprinter precum este fontul "courier".

 este folosit pentru a afișa un text ce trebuie scos în evidență, sau un termen nou introdus în document. Majoritatea browserelor vor afișa textul acestui element în format îngroșat sau înclinat.

 efectul este asemănător cu cazul folosirii directivei dar mai puternic.

Toate directivele de formatare logică de mai sus necesită specificarea explicită a directivei pereche de sfârșit.

3.6 Preformatarea textului cu ajutorul directivei <PRE>

Așa cum am mai arătat în unitatea de învățare precedentă în momentul în care un document HTML este afișat de către browser, spațiile aflate între două cuvinte adiacente, în caz ca sunt mai multe, vor fi transformate automat într-un singur spațiu. Există, bineînțeles metode de a defini spații suplimentare între cuvinte dacă acest lucru este necesar, iar una din aceste metode este folosirea directivei HTML: <PRE>.

Directiva <PRE> împreună cu directiva sa pereche de sfârșit, creează un spațiu în interiorul căruia textul va fi afișat de către browser exact în formatul din codul sursă HTML, păstrându-se numărul de spații libere sau numărul de linii libere succesive. Lungimea liniei nu mai este ajustată în acest caz în funcție de dimensiunea ferestrei browserului.

Textul din interiorul elementului <PRE> este afișat folosindu-se un font cu lungimea constantă a caracterului (e.g. courier).

Conținutul elementului <PRE> poate include orice directivă de formatare fizică sau logică, imagini sau legături². Directivele care implică terminarea unui paragraf – adică <P> sau <Hn> - nu sunt recomandate spre a fi folosite în interiorul elementului deoarece rezultatul folosirii acestora în acest caz nu este consistentă pentru toate tipurile de browser.

Directiva <PRE> are un atribut opțional - **WIDTH** care determină numărul de caractere conținute într-o linie a blocului preformatat. Dacă o linie are lungimea mai mare de cea specificată de acest atribut nu înseamnă ca aceasta va fi automat ajustată la această dimensiune ci mai degrabă va fi extinsă în afara regiunii vizibile a ferestrei browserului.

Folosirea acestor elemente este în general utilă pentru realizarea de tabele sau atunci când se dorește păstrarea integrității unor coloane sau rânduri. Pentru aceasta HTML pune la dispoziție însă și alte directive speciale despre care vom vorbi în lecțiile următoare. Însă avantajul folosirii acestei tehnici în locul directivei specializate este că nu toate browserele suportă utilizarea acestora.

² Vom discuta despre acestea în unitatea de învățare 5 și 6

Lucrare de verificare a Unităților de învățare Nr. 1, 2 și 3



1. Creați un document HTML care să conțină doar structura de bază a unui document HTML și care să aibă

- culoare de fundal: galbenă
- Să afișeze textul: "Am făcut prima Lucrare!" cu culoarea roșie

Predați fișierul HTML în format electronic ca rezultat al rezolvării lucrării de verificare.

Pe lângă informațiile prezentate în manual se poate utiliza reperul bibliografic numărul 2 din bibliografia unității de învățare.

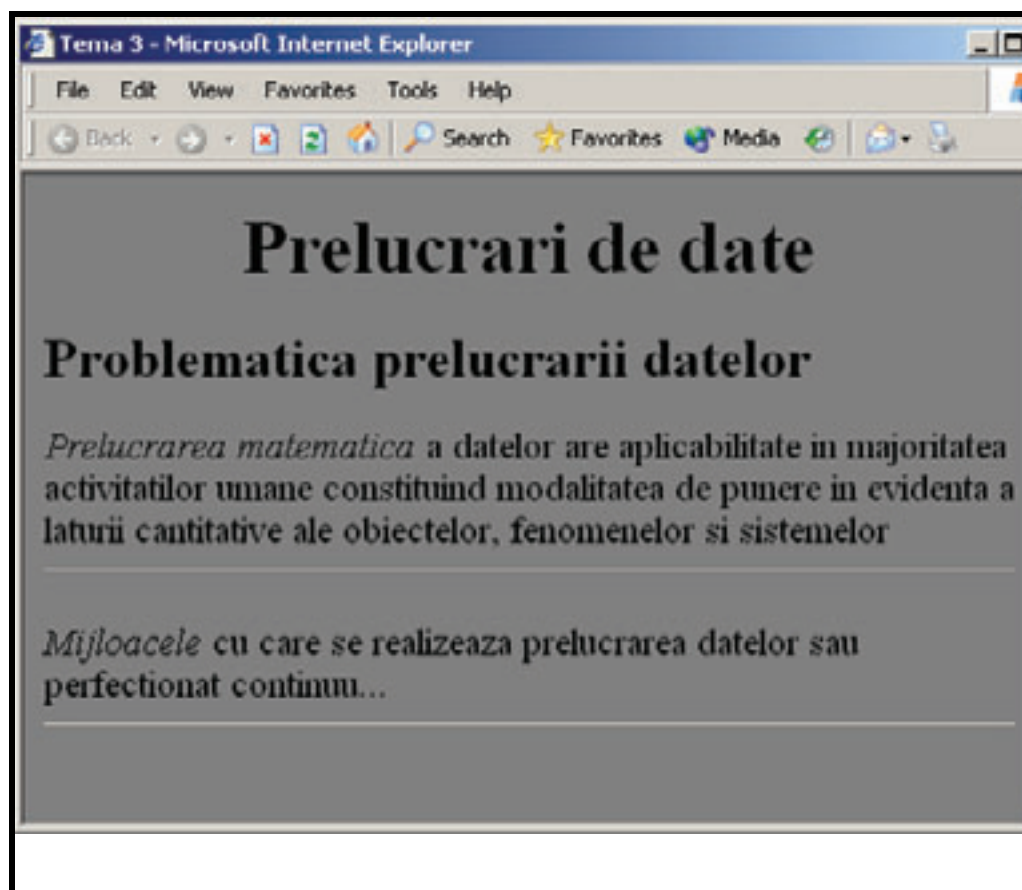
Nr. de puncte **15** (7 puncte definire culoare fundal, 8 puncte afișarea textului cerut folosind culoarea roșie)

2. Scrieți documentul HTML pentru o pagină WEB ca cea din figura de mai jos. Aceasta trebuie să conțină: 1 titlu de tip H1 poziționat central, 1 titlu de tip H2 aliniat la stânga, corpul textului va fi format din 2 paragrafe în care primul cuvânt din text are format italic.

Predați fișierul HTML în format electronic ca rezultat al rezolvării lucrării de verificare.

Ca ajutor suplimentar utilizați reperele bibliografice 2 și 3 din bibliografia unității de învățare.

Nr. de puncte **10** (5p - folosirea directivelor specifice pentru titluri și paragrafe, 2p - împărțirea în 2 paragrafe a textului (la fel ca în figură), 3p - formatarea tip italic a primului cuvânt din fiecare paragraf.



Răspunsuri și comentarii la întrebările din testele de autoevaluare



Întrebarea 3.1.

Directiva
 se folosește pentru a forța trecerea la o linie nouă. De asemenea se poate folosi în cazul în care se dorește introducerea mai multor linii libere succesive. Vezi secțiunea 3.2.



Întrebarea 3.2.

Elementul de tip paragraf va formata textul conținut astfel încât dimensiunea liniei să nu depășească lungimea ferestrei. La începutul și sfârșitul unui paragraf se va adăuga automat o linie liberă. Vezi secțiunea 3.2.



Întrebarea 3.3.

Directiva HTML pentru a desena această linie se scrie în felul următor:

```
<HR  
    ALIGH="CENTER"  
    WIDTH="50%" >
```

Vezi secțiunea 3.2.



Întrebarea 3.4.

Directiva <BASEFONT> se folosește în cazul în care se dorește schimbarea caracteristicilor fontului pentru întreg textul documentului. Directiva FONT se folosește pentru a schimba caracteristicile pentru un număr oarecare de cuvinte din text. Vezi secțiunea 3.4.

Indicații la problemele propuse

Problemele propuse în lucrările de verificare se fac după modelele de exemple prezentate în unitatea de învățare sau după tipicul acestora.

Bibliografie

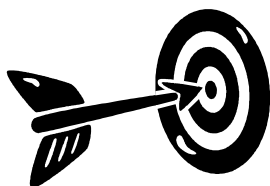
1. Dave Taylor - Crearea paginilor WEB, Editura Teora 1999, pg.71-82
2. Dorin Cârstoiu, Ecaterina Oltean – Introducere în HTML și XML, Editura Printech București 2004, pg.20-27
3. T.Gugoiu – HTML prin exemple, Editura Teora 2000

Unitatea de învățare Nr. 4

LISTE ÎN LIMBAJUL HTML

Obiectivele Unității de învățare Nr. 4	39
4.1 Tipuri de liste în limbajul HTML	39
4.2 Liste HTML neordonate	40
4.3 Liste HTML ordonate	44
4.4 Liste de definiție	44
Răspunsuri și comentarii la întrebările din testele de evaluare	47
Bibliografie	48

Obiectivele Unității de învățare Nr. 4:



După parcurgerea acestei Unități de învățare vei fi capabil:

- Să creezi liste neordonate cu ajutorul directivelor HTML specializate
- Să creezi liste ordonate cu ajutorul directivelor HTML specializate
- Să creezi liste de definiții
- Să creezi liste imbricate.

4.1 Tipuri de liste în limbajul HTML

Clasificare Liste

Listele reprezintă un excelent mijloc pentru a sistematiza informația și de a scoate în evidență anumite aspecte importante dintr-un anumit context. Limbajul HTML oferă un suport bogat, pentru definirea listelor. Se pot crea trei tipuri de liste:

- Neordonate (*unordered lists*) – Listă fără numere de ordine.
- Ordonate (*ordered lists*) – Listă cu numere de ordine.
- Listă de definiții (*definition lists*) – Este o listă compusă din termeni și definiții ale acestora.

Cu excepția listelor de definiții, toate tipurile de listă au aceeași structură de bază, fiecare listă constând dintr-o secvență de elemente marcate de directiva .

```
<SPECIFICATOR_TIP_LISTA>
  <LI> ELEMENT 1 text element 1 </LI>
  <LI> ELEMENT 2 text element 2 </LI>
  ...
</SPECIFICATOR_TIP_LISTA>
```

În cadrul unui element al unei liste se pot folosi următoarele construcții HTML: paragrafe, imagini, legături, alte liste, directive de formatare text.

<p>Funcționalitate:</p> <ul style="list-style-type: none"> • Definește un element dintr-o listă <p>Atribute:</p> <ul style="list-style-type: none"> • TYPE • VALUE <p>Directiva de sfârșit:</p> <ul style="list-style-type: none"> • este opțională

Directiva este folosită atât în cadrul listelor ordonate cât și neordonate după cum vom vedea în secțiunile următoare.

4.2 Liste HTML neordonate

Listele neordonate sunt folosite pentru enumerarea unor elemente pentru care ordinea de apariție nu este importantă, cum ar fi, spre exemplu o listă de cumpărături, sau lista de obiective ale acestei secțiuni de învățare.

Cum
definim
lista

Elementele unei liste neordonate vor fi afișate intențat și fiind precedate de un marcaj. Acest marcaj este introdus automat de către browser și prin urmare nu trebuie specificată în codul sursă HTML.

<p>Funcționalitate:</p> <ul style="list-style-type: none"> • Definește o listă neordonată <p>Atribute:</p> <ul style="list-style-type: none"> • TYPE <p>Directiva de sfârșit:</p> <ul style="list-style-type: none"> • este OBLIGATORIE

O listă neordonată se definește folosind directiva HTML , iar directiva de sfârșit corespunzătoare este obligatorie. Înăuntrul acestui element fiecare element se specifică folosind directiva . Directiva de sfârșit poate fi omisă însă este recomandat ca aceasta să fie totuși specificată deoarece este utilă în cazul folosirii stilurilor CSS despre care vom vorbi în unitatea de învățare 12.

<P> Acesta este un exemplu de lista neordonata:


```

<LI>Primul element al listei</LI>
<LI>Al doilea element al listei</LI>
<LI>Al treilea element al listei</LI>
</UL>
</P>

```

Va avea ca rezultat în browser:

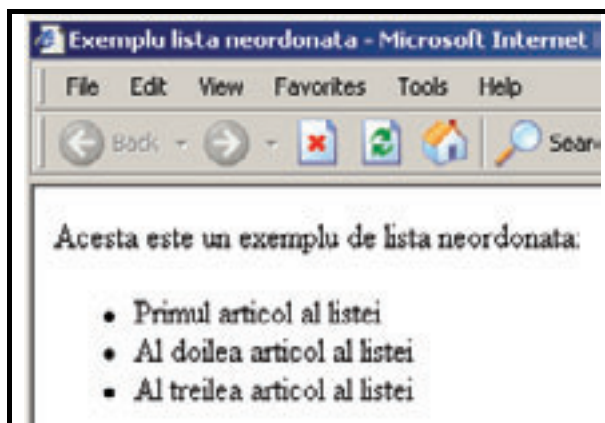


Figura 4.1 Exemplu lista neordonata

O listă neordonată introduce o linie liberă între ea și textul de deasupra și încă una între ea și textul ce urmează ei.

Atributul **TYPE**. Forma marcajului pentru fiecare element se poate schimba folosind atributul TYPE, care permite specificarea a 3 forme pentru marcaj:

- Valoarea "DISC" pentru forma de cerc plin – valoarea implicită
- Valoare "CIRCLE" pentru forma de cerc gol
- Valoarea "SQARE" pentru forma de pătrat gol

Atributul type poate fi de asemenea aplicat și fiecărei directive în parte, în cazul în care se dorește o formă diferită pentru fiecare din elementele listei.



Test de autoevaluare:

4.1 De câte tipuri pot fi listele HTML

4.2 Definiți o listă neordonată care să conțină 3 elemente. Primul și ultimul element să aibă marcajul de tip pătrat, iar al doilea element să aibă marcajul de tip disc.

Răspunsul corect și comentarii asupra acestuia se găsesc la pagina 47.

4.3 Liste HTML ordonate

Cum definim lista ordonata?

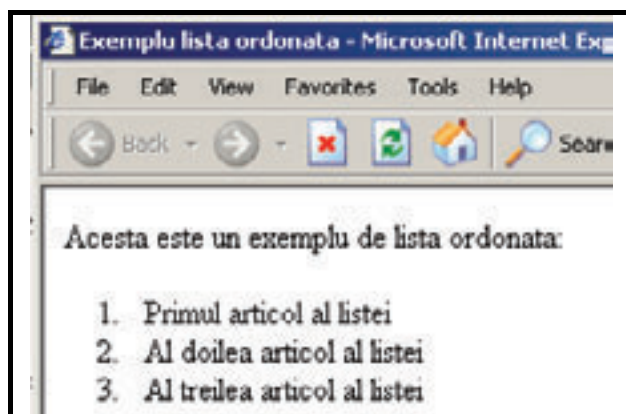
Listele ordonate sunt folosite în cazul în care ordinea elementelor este importantă să fie respectată, cum ar fi spre exemplu cuprinsul unei lucrări sau o listă de instrucțiuni. La fel ca în cazul listelor neordonate elementele sunt afișate intenționate dar în loc de un marcaj grafic browserul va afișa automat în fața fiecărui element un număr de ordine.

Funcționalitate: <ul style="list-style-type: none">• Definește o listă ordonată
Atribute: <ul style="list-style-type: none">• START• TYPE
Directiva de sfârșit: <ul style="list-style-type: none">• - OBLIGATORIE

O listă ordonată are aceeași structură de bază ca o listă neordonată. Pentru a defini o listă ordonată se folosește directiva înăuntrul căreia fiecare element se specifică folosind directiva .

```
<P> Acesta este un exemplu de lista ordonata :  
<OL>  
  <LI>Primul element al listei</LI>  
  <LI>Al doilea element al listei</LI>  
  <LI>Al treilea element al listei</LI>  
</OL>  
</P>
```

Va avea ca rezultat în browser:



Exemplu 4.2 – Exemplu listă ordonată

Atributul **TYPE**. Implicat numerotarea elementelor dintr-o listă ordonată se face automat folosindu-se cifre de tip arabice. Pentru a modifica stilul de numerotare se poate folosi atributul TYPE în cadrul directivei . Există 5 tipuri de numerotare suportate. În funcție de stilul de numerotare dorit se va specifica valoarea atributului TYPE așa cum este specificat în tabelul de mai jos.

Valoare atribut TYPE	Stilul generat	Exemplu
A	Litere majuscule	A, B, C, D...
a	Litere mici	a, b, c, d...
I	Cifre romane majuscule	I, II, III, IV...
i	Cifre romane mici	i, ii, iii, iv...
1	Cifre arabe	1, 2, 3, 4

Spre exemplu pentru a defini o listă care folosește stilul de numerotare cu litere majuscule vom scrie astfel:

```
<OL TYPE="A">
  <LI> Primul element </LI>
  <LI> Elementul 2 din lista</LI>
  <LI> ultimul element din lista</LI>
</OL>
```

Atributul **START**. Permite specificarea valorii cu care va începe numerotarea elementelor listei. Dacă nu este folosit atributul start numerotarea va începe cu 1 sau respectiv cu primul caracter din setul specificat de către atributul TYPE. Spre exemplu pentru a specifica o listă ordonată care folosește stilul de numerotare cu cifre romane și pentru care primul element începe de la valoarea "VI" (6) vom scrie astfel:

```
<OL TYPE="I" START="6">
  <LI>
    Primul element este numerotat cu VI
  </LI>
  <LI>
    Elementul următor va fi numerotat cu VII
  </LI>
  <LI>
    Si asa mai departe.....
  </LI>
</OL>
```

Listele ordonate precum și cele neordonate pot avea diferite nivele, în sensul că fiecare din elementele unei liste poate consta dintr-o altă sublistă. Fiecare din aceste subnivele va fi intențată corespunzător de către browser, efectul acestor intențări fiind cumulativ. De acest

fapt trebuie ținut cont atunci când se folosesc mai multe nivele de imbricare pentru liste.



Test de autoevaluare:

4.3 Când se folosesc listele ordonate în locul celor neordonate? Dați câteva exemple de folosire.

Răspunsul corect și comentarii asupra acestuia se găsesc la pagina 47.

4.4 Liste de definiție

Cum definim lista de definiție?

Listele de definiții permit enumerarea unei liste de elemente fiecare din acestea fiind urmate de explicația lor. Spre exemplu Glosarul unei cărți poate fi considerat o listă de definiții. Pentru definirea acestor tipuri de listă se folosesc trei directive HTML: `<DD>` , `<DL>` și `<DT>`.

<code><DL></code>
<p>Funcționalitate:</p> <ul style="list-style-type: none"> • Creează o listă de definiții
<p>Atribute:</p> <ul style="list-style-type: none"> • COMPACT
<p>Directiva de sfârșit:</p> <ul style="list-style-type: none"> • <code></DL></code> - OBLIGATORIE

O listă de definiții este încadrată de directiva pereche `<DL>`. În interiorul acestor directive fiecare element al unei liste de definiții este compus din 2 părți. Prima parte este un termen urmată în partea a doua de definiția acestuia. Pentru definirea primei părți se folosește directiva HTML `<DT>`, urmată de definiția acestuia care se face cu ajutorul directivei `<DD>`.

În mod normal browserul plasează definiția termenului pe un rând nou, însă dacă această definiție este foarte scurtă (3 caractere) o va plasa pe același rând cu termenul.

Atributul **COMPACT**. Specifică browserului să afișeze lista ocupând cât mai puțin spațiu posibil.

<DT>	<DD>
<p>Funcționalitate:</p> <ul style="list-style-type: none"> • Creează un termen într-o listă de definiții <p>Atribute:</p> <ul style="list-style-type: none"> • Nu are <p>Directiva de sfârșit:</p> <ul style="list-style-type: none"> • </DT> - opțională 	<p>Funcționalitate:</p> <ul style="list-style-type: none"> • Creează o definiție pentru un termen <p>Atribute:</p> <ul style="list-style-type: none"> • Nu are <p>Directiva de sfârșit:</p> <ul style="list-style-type: none"> • </DD> - opțională

În interiorul elementului <DD> se poate folosi aproape orice directivă HTML, inclusiv alte liste, imagini, directive pentru formatarea textului etc. Totuși dacă nu este folosit nici o directivă de formatare, textul din cadrul elementului <DD> este afișat intențat.

Mai jos este exemplificat modul de folosire a unei liste de definiții prin realizarea unui glosar de termeni din domeniul WEB:

```

...
<H3> Glosar termeni WEB </H3>
  <DL>
    <DT>Browser</DT>
    <DD>Aplicatie software utilizat pentru
      vizualizarea paginilor WEB</DD>
    <DT>GIF</DT>
    <DD>Format de imagine comprimat utilizat
      frecvent in Internet</DD>
    <DT>HTML</DT>
    <DD>Limbaj pentru realizarea paginilor
      WEB</DD>
  </DL>
...

```

Va avea ca rezultat în browser:

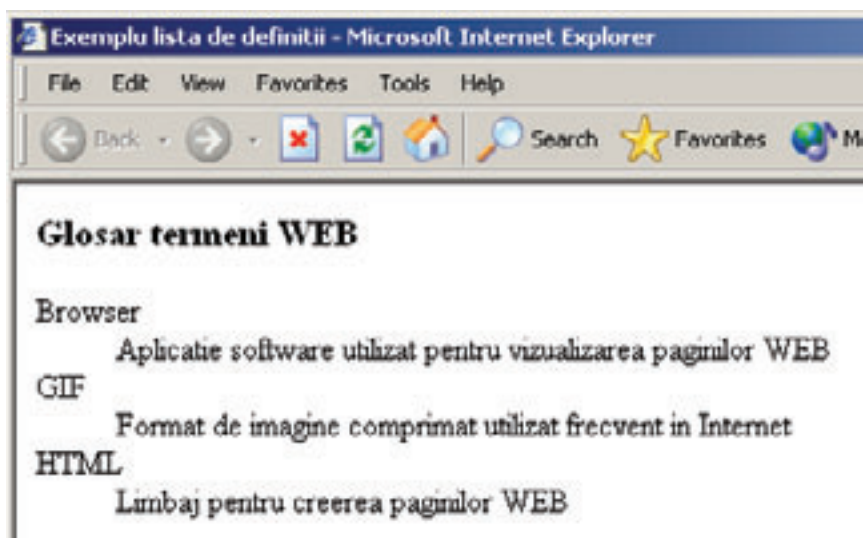


Figura 4.3 Exemplu lista definiții



Test de autoevaluare:

4.4 . Construiți o listă de definiții care să conțină tipurile de liste disponibile în HTML și definiția lor.

Răspunsul corect și comentarii asupra acestuia se găsesc la pagina 47.

Răspunsuri și comentarii la întrebările din testele de evaluare



Întrebarea 4.1.

Limbajul HTML oferă suport pentru realizarea a trei tipuri de liste: neordonate, ordonate, liste de definiții. Consultați secțiunea 4.1.



Întrebarea 4.2.

Codul HTML pentru realizarea acestei liste este:

```
<UL>
  <LI TYPE="SQARE">Element 1</LI>
  <LI>Element 2 </LI>
  <LI TYPE="SQARE"><LI>Element 3</LI>
</UL>
```

Consultați secțiunea 4.2.



Întrebarea 4.3.

Listele ordonate sunt folosite în cazul în care ordinea elementelor este relevantă. Exemple de folosire pentru listele ordonate: Cuprinsul lucrărilor, lista de pași a unui algoritm etc. Consultați secțiunea 4.3.



Întrebarea 4.4.

Codul HTML pentru realizarea acestei liste este:

```
<DL>
  <DT>Neordonate</DT>
  <DD>
    Lista fara numere de ordine
  </DD>
  <DT>Ordonate</DT>
  <DD>
    Lista cu numere de ordine
  </DD>
  <DT>Lista de definitii</DT>
  <DD>
    Lista compusa din termeni si definitii ale acestora
  </DD>
</DL>
```

Consultați secțiunea 4.4.

Indicații la problemele propuse

Problemele propuse în lucrările de verificare se fac după modelele de exemple prezentate în unitatea de învățare sau după tipicul acestora.

Bibliografie

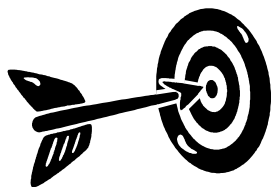
1. Dave Taylor - Crearea paginilor WEB, Editura Teora 1999, pg.85-99
2. Dorin Cârstoiu, Ecaterina Oltean – Introducere în HTML și XML, Editura Printech București 2004, pg.36-38
3. T.Gugoiu – HTML prin exemple, Editura Teora 2000, pg.33-38

Unitatea de învățare Nr. 5

HYPERLINKS ÎN LIMBAJUL HTML

Obiectivele Unității de învățare Nr.5	50
5.1 Hypertext și hyperlink	50
5.2 URL – identificator unic de resurse WEB	51
5.3 Realizarea legăturilor în HTML	51
5.4 Folosirea imaginilor pentru legături	55
Răspunsuri și comentarii la întrebările din testele de evaluare	57
Bibliografie	57

Obiectivele Unității de învățare Nr. 5:



După parcurgerea acestei Unității de învățare veți fi capabili:

- Să creați hyperlink-uri în documente HTML
- Să definiți tipurile de legături
- Să folosiți imagini ca hyperlink-uri
- Să înțelegeți definiția și structura unui URL

5.1 Hypertext și Hyperlink

Hypertext, Hyperlinks

Termenul de *Hypertext* este format dintr-o asociere de alți doi termeni: *Hyper* - care înseamnă “mai încolo de” și *text*. și se referă la o modalitate de organizare a unui document diferită de organizarea liniară folosită în cazul documentelor tipărite. Mai concret cititorul unui document Hypertext are posibilitatea să-l parcurgă într-o altă ordine decât cea predefinită de către autor. Acest lucru este posibil prin existența legăturilor – *Hyperlinks* – între diferite componente ale documentului.

Adevărata putere a limbajului HTML constă tocmai în abilitatea sa de a lega texte sau imagini din cadrul unui document HTML de alt document HTML sau secțiune a acestuia. Aceste legături au scopul de a permite cititorului să “sară” la un moment dat direct la locul unde va găsi mai multe informații despre o chestiune în discuție. Implementând astfel conceptul de Hypertext.

Definiție

În esență un sistem Hypertext este o rețea semantică ale cărei noduri sunt fragmente de text. Dacă nodurile conțin și imagini sau sunete, putem vorbi de un sistem Hypermedia Într-un sistem Hypertext se disting următoarele elemente fundamentale:

- **nodurile** – reprezintă paginile unui document Hypertext
- **ancorele** (țintele)– reprezintă fragmente unitare de noduri cum ar fi spre exemplu cuvinte evidențiate în text, imagini, butoane etc.
- **legăturile** între noduri, având drept punct de plecare fie întreg nodul sursă fie o ancoră a sa

5.2 URL – Identificator unic de resurse WEB

Fiecare pagină sau resursă WEB are asociată o adresă unică în internet cunoscută sub acronimul de **URL** (*Uniform Resource Locator*).

Sintaxa unui URL este concepută a fi cât mai generic posibilă, un URL fiind compus din următoarele componente:

Componentele URL-ului

1. **Numele** resursei, precedat de:
2. **Ierarhia de directoare** unde se află resursa respectivă, precedat de:
3. **Adresa de internet** sau **numele de domeniu** al serverului care găzduiește resursa, precedat de:
4. **Protocolul** folosit de către browser și serverul care găzduiește resursa pentru a o transmite

Nu este obligatoriu pentru toate componentele descrise mai sus să apară într-un URL acestea fiind (în măsura posibilităților) completate automat de către browser sau server. Aceasta este sintaxa unui URL:

protocol://adresă_server/cale_resursă/nume_resursă

Și iată câteva exemple de URL-uri:

http://www.hotnews.ro/revista_presei.htm

<http://www.desprecopii.com>

<ftp://ftp.myftp.ro/pub>

informatii_contact.html

Primul URL din exemplul de mai sus este un URL în format *absolut* sau *complet*. Browserele oferă de asemenea posibilitatea folosirii URL-urilor *relative* sau incomplet specificate, completând automat părțile lipsă ale URL-ului specificat atât cât este posibil.

Cum ar fi în exemplul de mai sus pentru cel de-al patrulea URL browserul îl va completa automat presupunând că pagina "*informatii_contact*" se află pe același server și în același director cu documentul curent.

5.3 Realizarea legăturilor în HTML

În cadrul unei pagini HTML unul sau mai multe cuvinte consecutive pot să aibă asociate o anumită resursă WEB adică, o altă pagină HTML, o imagine, un film etc. Directiva `<A>` face posibilă definirea acestei asocieri sau legăturii folosind în acest scop următoarea sintaxă:

```
<A HREF="URL" >  
    Textul afișat cu care se face asocierea  
</A>
```

Legăturile vor fi afișate de către browser cu altă culoare și vor fi scoase în evidență față de restul textului, de obicei fiind subliniate.

Iată un exemplu de definiție a unei legături HTML:

```
...  
<P>Definirea  
<A HREF="http://www.invathtml.ro/despre_legaturi.html">  
legaturilor HTML  
</A>  
este simpla!  
</P>  
...
```

Va avea ca rezulta în browser:

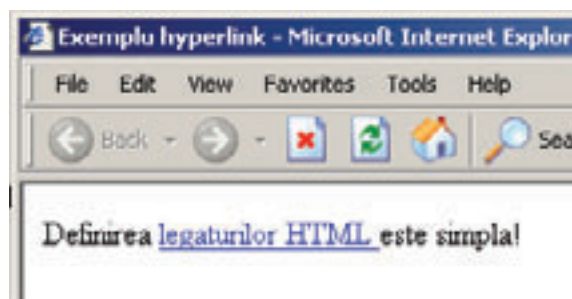


Figura 5.1 Exemplu legătură HTML

Dacă utilizatorul va selecta zona legăturii cu mouse-ul browserul va încărca pagina identificată de URL-ul specificat ca valoare a atributului HREF.

<A>
Funcționalitate: <ul style="list-style-type: none">• Creează o legătură sau o ancoră
Atribute: <ul style="list-style-type: none">• HREF• NAME• TITLE• TARGET
Directiva de sfârșit: <ul style="list-style-type: none">• este OBLIGATORIE

În concluzie, directiva `<A>` este folosită pentru a crea o legătură (hyperlink) către un document, atributul `HREF` având rolul de a specifica adresa acelui document, iar cuvintele dintre directivele ancoră de deschidere `<A>` și respectiv de închidere vor fi afișate ca hyperlink.

Tipuri de legaturi

Există trei tipuri principale de legături:

- **Legături interne** folosite în cadrul documentelor de dimensiuni mari pentru o mai bună structurare a conținutului.
- **Legături locale** sunt legături către alte documente aflate pe același server. Legăturile locale se pot specifica folosind fie URL-ul complet, fie un URL relativ care conține doar calea către resursa respectivă relativ la directorul curent.
- **Legături externe** sunt legături către pagini găzduite pe alt server de WEB. Pentru legăturile externe se folosește întotdeauna URL-ul specificat complet.

Atributul NAME

Așa cum am arătat mai sus o ancoră este o etichetă folosită pentru a identifica o anumită secțiune a unui document HTML. Pentru a defini o ancoră directiva `<A>` este folosită împreună cu atributul său **NAME**.

Spre exemplu, presupunând că acest document este un document HTML și vrem să definim o ancoră pe titlul acestei subunității de învățare pentru a putea face referire la el din altă parte a documentului, vom scrie:

```
<A NAME="legaturi_html" >
  Realizarea legăturilor HTML
</A>
```

Atributul HREF

Mai departe, în momentul în care se dorește referirea acestei ancore se va folosi tot directiva `<A>` dar de această dată împreună cu atributul **HREF**. În cazul în care referința se face din aceeași pagină în care a fost definită ancora este de ajuns specificarea numelui ancorei precedat de caracterul `#`. Ca în exemplul de mai jos:

```
...
<P>Definirea
  <A HREF="#legaturi_html" >
    legaturilor HTML
  </A>
este simpla.</P>
...
```

Accesarea acestei legături va spune browserului să deruleze pagina până la începutul Unității de învățare "Realizarea legăturilor HTML", unde este definită ancora "legaturi_html".

Dacă referirea se face din altă pagină HTML dar care se află pe același server și în același director se poate de asemenea folosi de

asemenea un URL relativ. Presupunând că fișierul ce conține Unitatea noastră de învățare se numește “despre_legaturi.html” vom scrie:

```
...
<P>Definirea
  <A HREF="despre_legaturi.html#legaturi_html">
    legaturilor HTML
  </A>
este simpla.</P>
...
```

În cazul în care pagina noastră se află pe alt server este necesară folosirea URL-ului complet specificat.



Test de autoevaluare

5.1 Scrieți directiva ce realizează hyperlink la pagina și la ancora definite mai sus în cazul în care aceasta se află pe un alt server de WEB decât pagina din care se face referire.

Răspunsul corect și comentarii asupra acestuia se găsesc la pagina 57.

Atributul Target

Atributul **TARGET**. În mod obișnuit la accesarea unui hyperlink browserul va încărca resursa specificată în aceeași pagină a browserului. Pentru a specifica o locație diferită de încărcare a resursei se folosește atributul target. Dacă este specifică valoarea “_blank” resursa va fi încărcată într-o fereastră pe care browserul o va crea special pentru afișarea noii resurse.

```
<A
  HREF="http://www.yahoo.com"
  TARGET="_blank"
>
  Yahoooooo!
</A>
```

La accesarea legăturii browserul va deschide o nouă fereastră în care va afișa pagina cunoscutului portal Yahoo.

Dacă veți folosi pentru toate legăturile din pagină atributul TARGET specificând valoarea “_blank”, fiecare din aceasta va fi lansată într-o

nouă fereastra de browser ceea ce nu este întotdeauna elegant deoarece este foarte probabil ca utilizatorul să piardă numărul ferestrelor deschise și să nu mai fie capabil să le gestioneze eficient.

Atributul TITLE

Atributul **TITLE** Permite specificarea unui text descriptiv pentru resursa la care se referă legătura. Acest text va fi afișat de către browser în momentul în care cursorul mouse-ului este deplasat deasupra legăturii.

De obicei acest text este afișat sub formă de fereastră *tool tip* precum în exemplul de mai jos.

```
<A HREF="http://www.yahoo.com"
  TITLE="Aceasta legatura ne transporta direct la
        pagina Yahoo"
>
  Yahoooooo!
</A>
```

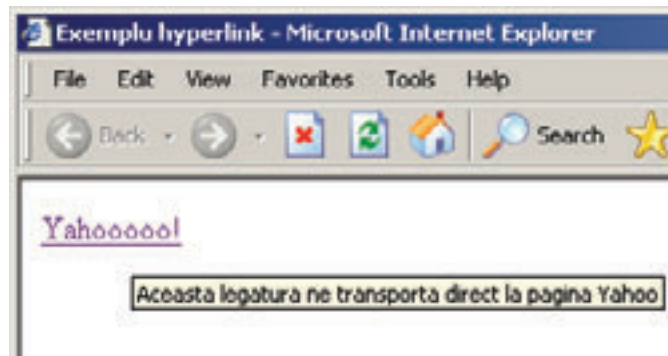


Figura 5.2 Exemplu utilizare atribut TITLE

Atributul TABINDEX

Atributul **TABINDEX**. În mod obișnuit o legătură este accesată în momentul în care utilizatorul se află cu cursorul deasupra legăturii și apasă butonul mouse-ului. O altă modalitate de accesare a unei legături este prin apăsarea succesivă a tastei <TAB> până în momentul în care cursorul ajunge deasupra legăturii urmată de apăsarea tastei <ENTER>. Ordinea în care se face trecerea de la o legătură la alta la apăsarea tastei <TAB> poate fi stabilită de către valoarea atributului **TABINDEX**. Valoarea atributului **TABINDEX** poate fi orice întreg mai mare decât zero.

Atributul ACCESSKEY

Atributul **ACCESSKEY**. Valoarea acestui atribut constă într-un identificator al unei taste la apăsarea căreia se va accesa legătura.

5.4 Folosirea imaginilor pentru legături

Pentru a face o pagină cât mai dinamică, mai interactivă și mai ușor de folosit pentru utilizator o tehnică adesea utilizată este cea a folosirii imaginilor ca hyperlink-uri în locul textelor. Spre exemplu pentru legătura către pagina WEB principală se poate folosi în locul

textului HOME o imagine ce înfățișează o casă precum imaginea de

mai jos: .

Realizarea unei legături care să folosească imagini în locul textului este simplă și se face cu ajutorul directivelor `<A>` și ``, directiva `IMG` fiind specificată în interiorul perechii `<A>` ca în exemplu de mai jos:

```
<A HREF="Home.html">  
  <IMG SRC="home.jpg">Acasa  
</A>
```

Codul HTML de mai sus va crea o imagine însoțită de un text ("Acasa") care se va comporta ca un hyperlink.

Atunci când este folosită ca hyperlink o imagine este în mod normal încadrată într-un pătrat de culoarea setată pentru hyperlink în pagina respectivă. În felul acesta este indicat faptul că hyperlinkul este activ



- Afișarea imaginilor în pagină poate fi dezactivată în anumite browsere de aceea este recomandat în cazul folosirii imaginilor drept hyperlink ca directiva `` să conțină și atributul `ALT` cu o descriere sugestivă pentru imaginea sau hyperlink-ul respectiv.



Test de autoevaluare

5.2 Definiți un hyperlink de tip imagine care să afișeze un text explicativ în locul imaginii în cazul în care imaginea nu poate fi încărcată în browser.

Răspunsul corect și comentarii asupra acestuia se găsesc la pagina 57.

Răspunsuri și comentarii la întrebările din testele de evaluare



Întrebarea 5.1.

...

```
<P>Definirea
<A
  HREF="http://www.adresa_server.ro/cale_fisiere_html
/despre_legaturi.html#legaturi_html">
  legaturilor HTML
</A>
este simpla.</P>
```

...

Pentru nelămuriri revedeți secțiunea 5.3.



Întrebarea 5.2.

Codul HTML este următorul:

```
<A HREF="Home.html">
  <IMG SRC="home.jpg" ALT="ACASA">
</A>
```

Pentru nelămuriri revedeți secțiunea 5.4.

Indicații la problemele propuse

Problemele propuse în lucrările de verificare se fac după modelele de exemple prezentate în unitatea de învățare sau după tipicul acestora.

Bibliografie

1. Dave Taylor - Crearea paginilor WEB, Editura Teora 1999, pg.100-108
2. Dorin Cârstoiu, Ecaterina Oltean – Introducere în HTML și XML, Editura Printech București 2004, pg.29-32
3. T.Gugoiu – HTML prin exemple, Editura Teora 2000, pg.65-74
4. Sabin Buraga – Proiectarea siturilor Web. Design și funcționalitate, Ediția a II-a, Editura Polirom 2002, pg. 23-30

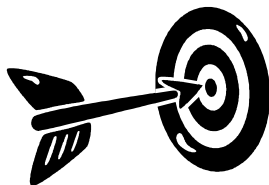
Unitatea de învățare Nr. 6

FOLOSIREA IMAGINILOR ÎN LIMBAJUL HTML

Obiectivele Unității de învățare Nr.6	59
6.1 Înțelegerea formatelor grafice folosite în WEB	59
6.2 Folosirea imaginilor în cadrul paginilor WEB	69
6.3 Imagini cu arii sensibile (Image Maps)	65
Lucrare de verificare a Unităților de învățare Nr. 4, 5 și 6	67
Răspunsuri și comentarii la întrebările din testele de evaluare	70
Bibliografie	71

Obiectivele Unității de învățare Nr. 6:

Principalele obiective ale Unității de învățare Nr. 6 sunt:



După parcurgerea acestei Unității de învățare vei fi capabil:

- Să identifici diferite formate de imagini suportate de browsere
- Să adaugi imagini la o pagină HTML
- Să specifici modul de aliniere al imaginii în pagină și al textului din jurul ei
- Să specifici dimensiunea imaginii

6.1 Înțelegerea formatelor grafice folosite în WEB

Limbajul HTML nu impune folosirea unui anumit format de imagine ci oferă în schimb o modalitate flexibilă de a include orice tip de imagine în pagina HTML. Afișarea unui anumit tip de imagine depinde însă de tipul browserului, de aceea există totuși un număr limitat de formate grafice folosite în WEB., și dintre acestea doar două sunt populare și foarte des folosite: **GIF** și **JPEG**.

GIF

Formatul grafic folosit prima oară în cadrul paginilor WEB și care a rămas cel mai popular este formatul GIF. Denumirea de GIF reprezintă inițialele expresiei *Graphic Interchange Format* care în limba română se traduce prin “*Format grafic pentru transfer*”.

Avantajele formatului grafic GIF sunt următoarele:

- Este independent de platformă. Aceasta înseamnă că dacă o imagine este creată pe o mașină care rulează LINUX aceasta se va vedea la fel pe o mașină care rulează alt sistem de operare Windows sau Macintosh.
- Stocarea datelor se face comprimat ceea ce duce la fișiere de dimensiuni mici care se pot transfera rapid în rețea.
- Comprimarea datelor în cazul formatului grafic GIF se face folosindu-se un algoritm fără pierdere de date. Aceasta înseamnă că dintr-o imagine în format GIF se poate restaura în orice moment imaginea originală necomprimită.

JPEG

Dezavantajul formatului GIF este acela că nu permite folosirea unui număr mai mare de 256 de culori pentru o imagine. Acest neajuns este înlăturat în cazul formatului **JPEG**.

Formatul grafic JPEG a fost creat de către "*Joint Photographic Experts Group*" care înseamnă în limba română "Grupul experților fotografi", și este ca și formatul GIF independent de platformă. Spre deosebire însă de GIF acest format suportă zeci de mii de culori pentru a putea reprezenta imagini fotografice cât mai aproape de realitate. Un alt avantaj și dezavantaj în același timp al formatului JPEG față de GIF este acela că formatul JPEG folosește un algoritm cu ajutorul căruia se obține o rată mai bună de compresie a datelor. Spre exemplu este un lucru obișnuit ca un fișier JPEG să aibă dimensiuni de până la 7-8 ori mai mici decât un fișier în format GIF care reprezintă aceeași imagine. Am spus dezavantaj în același timp deoarece această compresie se face spre deosebire de GIF cu pierdere de date. Aceasta înseamnă că o imagine transformată din format JPEG în format necomprimat nu va corespunde exact cu originalul ci vor fi anumite diferențe. Aceste diferențe însă nu se pot observa în mod normal cu ochiul liber.

Cum alegem între GIF și JPEG? Ei bine, ambele formate sunt universal suportate de către majoritatea browserelor, și prin urmare criteriul compatibilității nu este unul puternic. Prin urmare vom încerca să folosim avantajele fiecărui format în parte.

6.2 Folosirea imaginilor în cadrul paginilor WEB

Pentru a introduce imagini într-o pagină HTML se folosește directiva HTML .

Funcționalitate: <ul style="list-style-type: none">• Introduce o imagine în pagină
Atribute: <ul style="list-style-type: none">• SRC• ALT• ALIGN• BORDER• HEIGHT• WIDTH• HSPACE• VSPACE
Directiva de sfârșit: <ul style="list-style-type: none">• este OPTIONALĂ

Directiva HTML IMG folosește valoarea atributului SRC pentru a identifica locul de unde va prelua imaginea care se dorește a fi adăugată în pagină. Iată mai jos un exemplu cu cel mai simplu mod de utilizare al directivei IMG:

```
<P>
Acesta este poza iepurasului roz:
<IMG SRC="iepuras.gif" ALT="iepuras roz">
</IMG>
</P>
```

va avea ca rezultat în browser:

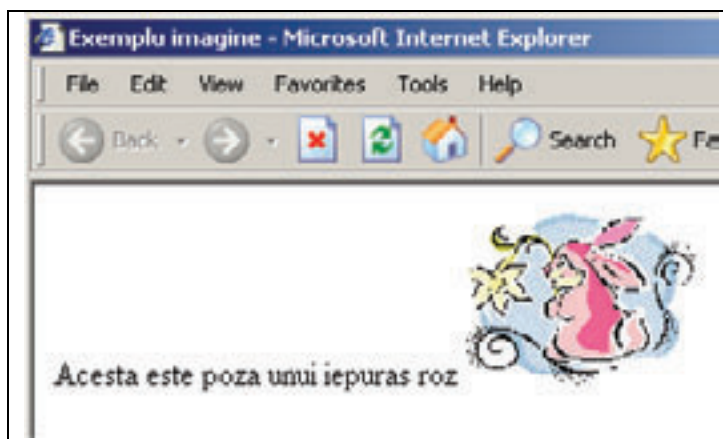


Figura 6.1 Exemplu folosire directiva



Folosirea atributului ALT nu este obligatorie însă este recomandată deoarece browserul se poate configura astfel încât să nu încarce imaginile din pagină, caz în care în locul imaginii este afișat textul oferit de atributul ALT. În mod ideal pagina ar trebui construită astfel încât să fie lizibilă chiar dacă imaginile nu pot fi încărcate.

Folosirea directivei nu va introduce imaginea pe o linie nouă ci în continuarea textului. Implicit imaginea va fi aliniată astfel încât marginea inferioară a imaginii să corespundă cu marginea inferioară a textului precum se vede în Figura 6.1. Dacă se dorește o altfel de aliniere verticală a imaginii sau a textului se vor folosi atributele directivei după cum vom vedea mai jos.

Atributul SRC. Acest atribut este obligatoriu în cazul directivei . Valoarea este un URL și reprezintă locul în care se află fișierul ce conține imaginea ce se dorește a fi inclusă în pagina.

Etichetare imagine

Atributul ALT. Dacă imaginea specificată de către atributul SRC nu poate fi încărcată din diferite motive: fișier imagine incomplet, conexiune întreruptă, URL specificat greșit sau browserul este configurat pentru a nu încărca imaginile, atunci vor fi afișate implicit în locul imaginii o altă imagine generică care să indice faptul că imaginea specificată nu a fost încărcată. Acest lucru nu este întotdeauna de dorit deoarece utilizatorul nu ar putea avea nici o

informație în legătură cu imaginea care nu s-a încărcat. În aceste cazuri textul specificat de atributul ALT va fi afișat în locul imaginii generice.



Teste autoevaluare

6.1. Cum se poate face într-o pagină HTML pentru ca imaginea *margareta.jpg* din directorul curent să fie afișată la începutul paginii?

6.2 Cum se poate face ca browserul să afișeze cuvântul "margareta" în locul imaginii dacă aceasta nu a putut fi încărcată de către browser?

- a> margareta
 b> margareta
 c>
 d> <P>margareta</P>

Răspunsurile corecte și comentarii asupra acestora se găsesc la pagina 70.

Atributul **ALIGN**. Cu ajutorul acestui atribut se poate specifica modalitatea de aliniere pe orizontală sau pe verticală a unei imagini față de textul sau altă imagine vecină.

Alinierea pe orizontală

Valorile disponibile pentru alinierea pe orizontală sunt :

- **ALIGN = "LEFT"** va avea ca efect poziționarea imaginii în stânga paginii, iar textul va încadra imaginea prin partea dreaptă a acesteia ca în figura 6.2.
- **ALIGN = "RIGHT"** va avea ca efect poziționarea imaginii în dreapta paginii, iar textul va încadra imaginea prin partea stângă a acesteia ca în figura 6.2.

Dacă se dorește la un moment dat întreruperea încadrării imaginii de către text se folosește directiva
 împreună cu atributul său CLEAR.

```
...
<P>
<IMG SRC="iepuras.gif"
      ALT="Bugs Bunny"
      ALIGN="left">
Un iepuras ... 9 x 9 x 19 h
<BR CLEAR="LEFT">
...
```

```
<IMG SRC="iepuras.gif"
      ALT="Bugs Bunny"
      ALIGN="RIGHT">
Un iepuras ... 9 x 9 x 19 h
<BR CLEAR=RIGHT>
...
```

Va avea ca rezultat în browser:



Figura 6.2 Exemplu aliniere orizontală la stânga și la dreapta

Alinierea pe verticală

Pentru specificarea modului de aliniere pe verticală a imaginii față de textul vecin sau față de alte imagini pe aceeași linie se folosește de asemenea atributul ALIGN următoarele valori fiind disponibile:

- **ALIGN="TOP"** – partea superioară a imaginii va fi la același nivel cu partea superioară a altei imagini sau cu a celei mai înalte litere din textul de pe aceeași linie.
- **ALIGN="MIDDLE"** – mijlocul imaginii va corespunde cu mijlocul altei imagini sau cu mijlocul textului de pe aceeași linie.
- **ALIGN="BOTTOM"** – partea inferioară a imaginii va corespunde cu partea inferioară a textului de pe aceeași linie.

```
...
Iepuras sus
<IMG SRC="iepuras.gif"
      ALIGN="TOP">. Ce dragut este!
<BR CLEAR="ALL">
Iepuras la mijloc
<IMG SRC="iepuras.gif"
```

```
        ALIGN="MIDDLE">Ce dragut este!  
<BR CLEAR="ALL">  
Iepuras jos  
<IMG SRC="iepuras.gif"  
        ALIGN="BOTTOM">Ce dragut este!  
...
```

Va avea ca rezultat în browser:

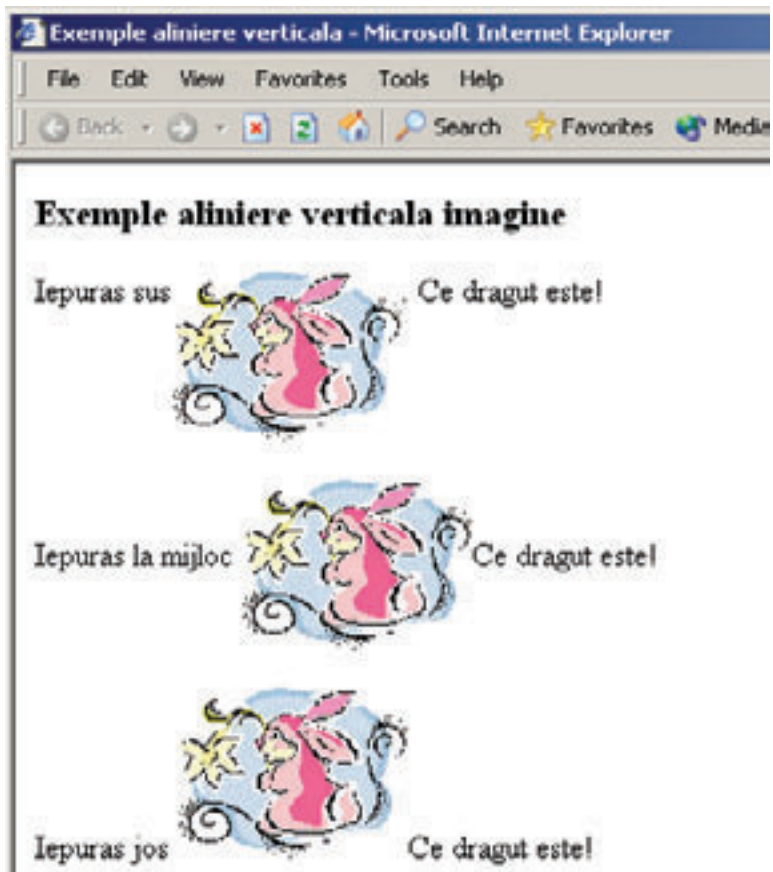


Figura 6.3 Exemple aliniere verticală.



Teste autoevaluare

6.3 Cum se scrie codul HTML care să afișeze o imagine centrată fără text la stânga și la dreapta ei ?

Răspunsul corect și comentarii asupra acestuia se găsesc la pagina 70.

Dimensiunea imaginilor

Atributele **HEIGHT** și **WIDTH** în mod normal browserul determină dimensiunea imaginii și în funcție de acesta rezervă un spațiu corespunzător pentru imagine în pagină pentru fiecare imagine în parte. Totuși aceste informații nu pot fi determinate decât în momentul în care imaginea a fost complet adusă de pe server. Acest lucru poate face ca timpul necesar pentru afișarea paginii în browser să crească. Dacă aceste informații legate de dimensiunile imaginii sunt specificate prin atributele **HEIGHT** și **WIDTH** timpul necesar pentru ca browserul să afișeze pagina va fi mult mai mic (chiar dacă imaginile vor fi afișate în pagină mai târziu, în momentul în care au fost complet încărcate).

Valoarea atributelor **HEIGHT** și **WIDTH** poate fi un întreg pozitiv și reprezintă numărul de pixeli pe verticală respectiv pe diagonală pe care îi ocupă imaginea.

Dacă valorile acestor atribute nu corespund cu dimensiunile imaginii browserul va redimensiona imaginea afișată astfel încât să fie conform cu valorile specificate de atribute.



Dimensiunea spațiilor

Redimensionarea imaginilor trebuie făcută cu grijă deoarece specificarea unor valori foarte îndepărtate de valoarea originală a imaginii poate rezulta în afișarea unor imagini distorsionate sau cu o rezoluție nepotrivită.

Atributele **HSPACE** și **VSPACE**. Permite definirea spațiilor între imagine și textul sau obiectele vecine. Valoarea acestor atribute poate fi un întreg pozitiv și reprezintă numărul de pixeli ce va fi lăsat între imagine și textul ce o înconjoară la stânga sau dreapta ei respectiv deasupra sau dedesubtul ei.

6.3 Imagini cu arii sensibile (Image Maps)

Așa cum am văzut în secțiunea anterioară imaginile pot fi folosite pentru hyperlink prin simpla adăugare a directivei `` în interiorul directivei `<A>`. Dacă utilizatorul va selecta oricare din punctele imaginii respective se va deschide pagina de la URL-ul specificat de valoarea atributului **HREF**. HTML oferă în plus o modalitate specială de folosire a imaginilor ca hyperlink permițând definirea unor suprafețe în cadrul imaginii care să aibă fiecare asociat câte un URL separat. Definirea acestor suprafețe se face cu ajutorul directivei `<MAP>` și `<AREA>`.

<MAP>	<AREA>
<p>Funcționalitate:</p> <ul style="list-style-type: none"> • Definește o hartă de suprafețe sensibile <p>Atribute:</p> <ul style="list-style-type: none"> • NAME <p>Directiva de sfârșit:</p> <ul style="list-style-type: none"> • </MAP> este OBLIGATORIE 	<p>Funcționalitate:</p> <ul style="list-style-type: none"> • Definește o suprafață sensibilă în cadrul unei hărți <p>Atribute:</p> <ul style="list-style-type: none"> • SHAPE • COORDS • HREF • NOHREF <p>Directiva de sfârșit:</p> <ul style="list-style-type: none"> • </AREA> este OPTIONALĂ

ată pașii pentru definirea suprafețelor sensibile pentru o anume imagine:

- Utilizarea în cadrul directivei IMG a atributelor ISMAP și USEMAP

```
<IMG SRC="imagine.gif"
      ISMAP
      USEMAP="#harta">
```

- Definirea hărții specificate ca valoarea a atributului MAP cu ajutorul directivelor MAP și AREA

```
<map name="harta">
  <area coords="0,25,50,50"
        href="link1.html">
  <area coords="0,0,25,25"
        href="link2.html"
</map>
```

Rezultatul va fi o imagine care va avea 2 arii sensibile una în partea stângă și una în partea dreaptă a imaginii. Dacă utilizatorul va selecta partea stângă browserul va încărca pagina "link1.html", dacă se va selecta aria dreaptă browserul va încărca pagina "link2.html".

Directiva MAP

Directiva MAP are un singur atribut, NAME, cu ajutorul căruia această hartă poate fi identificată și folosită pentru o anumită imagine. Definirea suprafețelor sensibile pentru fiecare hartă în parte se face folosind o succesiune de directive <AREA> și a atributelor acestora după cum vom vedea mai jos.

Directiva AREA

Directiva AREA definește fiecare din suprafețele sensibile ale unei hărți. Cu ajutorul atributelor sale se pot defini suprafețe cu diferite

forme și dimensiuni și se pot asocia hyperlinkuri pentru fiecare în parte.

Atributul **COORDS** permite definirea limitelor suprafeței, valoarea sa fiind reprezentată de o serie de numere întregi separate prin virgulă. Numărul și semnificația acestor valori este în funcție de valoarea atributului **SHAPE**.

Atributul **SHAPE**. Permite definirea formei pentru zona sensibilă. Mai jos sunt prezentate formele ce pot fi specificate de către acest atribut și valoarea corespunzătoare și de asemenea este explicat pentru fiecare în parte cum este interpretat :

- **RECTANGLE** – definește o zonă dreptunghiulară.
Coordonatele se specifică sub forma: **COORDS="x,y,z"** unde
- **CIRCLE** – definește o zonă circulară
- **POLYGON** – definește un poligon neregulat și închis cu un număr oarecare de laturi.

Lucrare de verificare a Unităților de învățare Nr. 4, 5 și 6



1. Creați o listă precum cea de mai jos din Figura 4.4 . Lista conține două nivele imbricate. La primul nivel avem o listă de definiții. Definiția pentru primul termen este o listă neordonată, iar definiția celui de-al doilea termen este o listă ordonată. Termenii listei de definiție trebuie să fie afișați îngroșat.

Predați fișierul HTML în format electronic ca rezultat al rezolvării lucrării de verificare

Pe lângă informațiile prezentate în manual se poate utiliza reperul bibliografic numărul 2 din bibliografia unității de învățare.

Nr. de puncte **8** (4p - realizarea listei de definiție conform cerințelor
2p – realizarea listei neordonate ca definiție pentru primul termen
2p – realizarea listei ordonate ca definiție pentru cel de-al doilea termen)

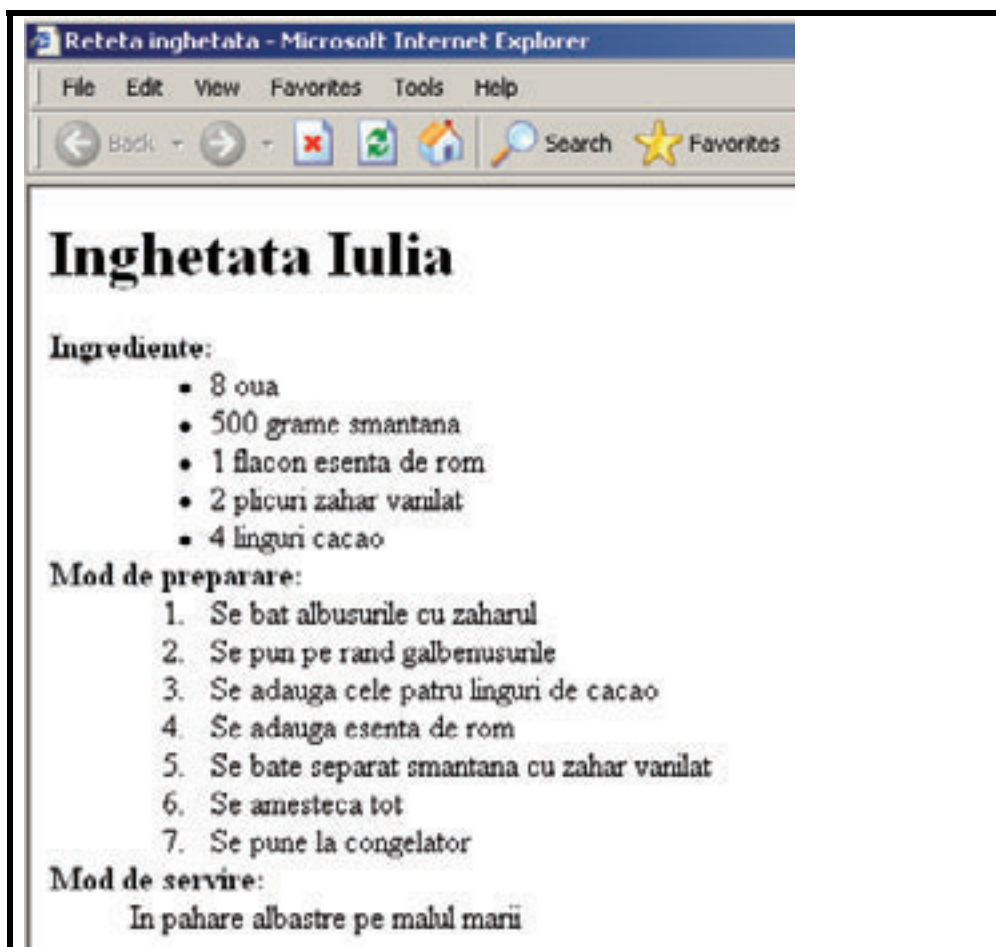


Figura 4.4 Lucrarea de verificare

2. Realizați un meniu pentru un site Web de prezentare a unei firme. Site-ul are următoarele pagini: *acasa.html*, *produse.html*, *servicii.html* și *contact.html*. În fiecare din acestea trebuie să fie o legătură la oricare altă pagină a site-ului. Încercați să îl faceți cât mai atractiv cu puțință!

Predați cele 4 fișiere HTML în format electronic ca rezultat al rezolvării lucrării de verificare.

Nr. de puncte **9** (6p – definirea corectă, în fiecare pagină a legăturilor către celelalte pagini html, 3p - pentru aspectul paginii și ingeniozitate.

Ca ajutor suplimentar utilizați reperele bibliografice 2 din bibliografia unității de învățare.

3. Realizați o listă HTML neordonată care să aibă în locul marcajelor standard imaginea conținută în fișierul *bulina.jpg*
Sugestie: Folosiți o listă de definiție pentru a simula lista neordonată cerută.

Predați fișierul HTML în format electronic ca rezultat al rezolvării lucrării de verificare

Pe lângă informațiile prezentate în manual se poate utiliza reperul bibliografic numărul 2 și 3 din bibliografia unității de învățare.

Nr. de puncte **8** (4p - realizarea listei, 4p – folosirea imaginii cerute în locul semnelor predefinite pentru listele neordonate)

Răspunsuri și comentarii la întrebările din testele de evaluare



Întrebarea 6.1.

Codul HTML care realizează acest lucru este următorul:

```
...  
<BODY>  
<P >  
  <IMG SRC="margareta.jpg"> </IMG>  
</P>
```

Directiva IMG este pusă imediat după directiva BODY pentru ca imaginea să fie afișată la începutul paginii. Revedeți indicațiile din secțiunea 6.2.



Întrebarea 6.2.

Varianta corectă de răspuns este **c**. Pentru aceasta se folosește atributul ALT pentru a specifica textul înlocuitor ca în exemplul de mai jos:

```
<IMG SRC="margareta.jpg"  
      ALT="margareta"> </IMG>
```

Revedeți indicațiile din secțiunea 6.2.



Întrebarea 6.3.

Pentru a realiza acest lucru folosim directiva BR împreună cu atributul CLEAR pentru a evita înconurarea imaginii cu text și directiva <P> împreună cu atributul ALIGN pentru a centra imaginea în pagină ca în exemplul de mai jos:

```
<BR CLEAR="ALL">  
<P ALIGN="CENTER">  
  <IMG SRC="margareta.jpg"  
        ALT="margareta"> </IMG>  
</P>
```

Revedeți indicațiile din secțiunea 6.2.

Indicații la problemele propuse

Problemele propuse în lucrările de verificare se fac după modelele de exemple prezentate în unitatea de învățare sau după tipicul acestora.

Bibliografie

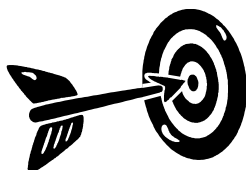
1. Dave Taylor - Crearea paginilor WEB, Editura Teora 1999, pg.127-169
2. Dorin Cârstoiu, Ecaterina Oltean – Introducere în HTML și XML, Editura Printech București 2004, pg.46-47
3. T.Gugoiu – HTML prin exemple, Editura Teora 2000, pg.51-59

Unitatea de învățare Nr. 7

REALIZAREA TABELELOR

Obiectivele Unității de învățare Nr.7	73
7.1 Realizarea unui tabel simplu în limbajul HTML	73
7.2 Definirea proprietăților globale ale unui tabel HTML	75
7.3 Definirea rândurilor unui tabel	78
7.4 Definirea celulelor unui tabel	79
Răspunsuri și comentarii la întrebările din testele de evaluare	82
Bibliografie	82

Obiectivele Unității de învățare Nr.7 :



După parcurgerea acestei Unității de învățare vei ști:

- Care este structura unui tabel HTML
- Care sunt elementele unui tabel HTML
- Să creezi un tabel HTML
- Să schimbi proprietățile unui tabel

Scopul inițial al tabelelor în limbajul HTML a fost pentru prezentarea anumitor date organizate în format tabular. Suportul oferit de către HTML pentru tabele s-a dovedit însă foarte eficient la aranjarea în pagină a diferitelor elemente HTML. Cu ajutorul directivelor oferite de către limbajul HTML este posibilă poziționarea a practic oricărui tip de element HTML la poziția dorită în pagină.

Componentele unui tabel

Hai să enumerăm și să descriem fiecare componentă a unui tabel în parte:

- **ROW** – Se referă la rândul unui tabel
- **COLUMN** – Se referă la coloana unui tabel
- **CELL** – Se referă la intersecția dintre o linie și o coloană
- **CAPTION** – Se referă la un text explicativ cu privire la conținutul tabelului care apare deasupra tabelului
- **HEADERS** – Se referă la primul rând al tabelului sau antetul tabelului
- **BORDERS** – Se referă la liniile de delimitare care înconjoară o celulă a tabelului sau întreg tabelul.

7.1 Realizarea unui tabel simplu în limbajul HTML

În limbajul HTML, un tabel simplu se poate defini cu ajutorul directivei `<TABLE>`; cu ajutorul directivei `<TR>` tabelul este împărțit în linii, iar cu ajutorul directivei `<TD>` fiecare linie este împărțită în celule.

Orice tabel în limbajul HTML va începe întotdeauna cu următoarea directivă pereche cu rol de container:

```
<TABLE>
</TABLE>
```


Pași

După directiva `<TABLE>` următorii pași sunt necesari pentru construirea tabelului:

Pasul 1. Adăugarea unui rând. – Se face folosind directiva pereche `<TR>` `</TR>` în interiorul elementului `<TABLE>`

```
<TABLE>
  <TR>
  </TR>
</TABLE>
```

Pasul 2. Împărțirea rândului într-un număr de coloane – Se face folosind directiva pereche `<TD>``</TD>` în interiorul unui element `<TR>`. Fiecare combinație `<TD>``</TD>` reprezintă o coloană/celulă a tabelului. Spre exemplu dacă tabelul are 3 coloane vom scrie:

```
<TABLE>
  <TR>
    <TD></TD>
    <TD></TD>
    <TD></TD>
  </TR>
</TABLE>
```

Pasul 3. Introducerea datelor în fiecare celulă a tabelului. În fiecare celulă a tabelului definită mai sus se introduce textul sau elementele HTML pe care vrem să le afișăm în tabel, precum în exemplul de mai jos:

```
<TABLE>
  <TR>
    <TD>Iepure</TD>
    <TD>25 Kg</TD>
    <TD><IMG SRC=" iepuras.gif" ></TD>
  </TR>
</TABLE>
```

Într-o celulă a unui tabel se poate pune aproape orice element HTML: text formatat, imagini, liste, hyperlink etc. În exemplul de mai sus ultima coloană conține imagine.

Pasul 4. Repetarea pașilor 1,2, 3 până ce tabelul este complet



Test de autoevaluare

7.1 Introduceți un nou rând în tabelul de mai sus

Răspunsul corect și comentarii asupra acestuia se găsesc la pagina 82.

7.2 Definirea proprietăților globale ale unui tabel HTML

Directiva HTML `<TABLE>` semnalează browserului faptul că urmează definiția unui tabel. La întâlnirea acesteia browserul va trece automat la o linie nouă unde va poziționa tabelul definit urmând ca la întâlnirea directivei de sfârșit `</TABLE>` să sară din nou la o linie nouă unde va afișa textul ce urmează tabelului.

Caracteristicile globale ale tabelului precum: modul de aliniere, dimensiuni, culori pot fi specificate folosind atributele acestei directive după cum vom vedea mai jos.

<TABLE>
<p>Funcționalitate:</p> <ul style="list-style-type: none"> • Definește un tabel <p>Atribute:</p> <ul style="list-style-type: none"> • ALIGN • BORDER • BGCOLOR • WIDTH • HEIGHT <p>Directiva de sfârșit:</p> <ul style="list-style-type: none"> • <code></TABLE></code> este OBLIGATORIE

Atributul ALIGN

Atributul **ALIGN**. Specifică unde anume va fi poziționat tabelul în pagină: aliniat la stânga (“LEFT”), la dreapta (“RIGHT”) sau pe centru (“CENTER”). Dacă acest atribut nu este specificat tabelul va fi aliniat la stânga.

Atributul VALIGN

Atributul **VALIGN**. Specifică modul de aliniere al textului din cadrul celulelor tabelului. Valorile posibile pentru atribut sunt “TOP” pentru

aliniere la partea superioară a celulei, "BOTTOM" aliniere la partea inferioară a celulei și "CENTER" aliniere la centru celulei.

**Atributul
BORDER**

Atributul **BORDER**. Specifică dimensiunea liniilor de delimitare care încadrează celulele tabelului. Valoarea atributului este un întreg între 0 și 15 și reprezintă grosimea în pixeli a liniei. Specificarea valorii "0" va face ca aceste linii de delimitare să fie invizibile în browser. Dacă atributul nu este specificat dimensiunea implicită este de 1 pixel.

**Atributul
CELLSPACING**

Atributul **CELLSPACING**. Specifică spațiul dintre două celule adiacente ale tabelului. Dacă nu este specificat valoarea implicită a acestui atribut este de 2 pixeli.

**Atributul
CELLPADDING**

Atributul **CELLPADDING** – Specifică spațiul minim în număr de pixeli dintre marginea unei celule și conținutul său. Implicit acest atribut are valoarea de un pixel.

Toate atributele de mai sus se pot folosi în același timp pentru a obține aspectul dorit pentru tabel.

Iată mai jos un exemplu de folosirea a acestor atribute. S-a folosit o valoare intenționat exagerată pentru a identifica în figura rezultată rolul fiecăruia în parte. De asemenea în aceeași pagină a fost definit un tabel pentru care aceste atribute nu au fost specificate și deci pentru care browserul a folosit la afișarea lui valorile implicite.

```

...
<TABLE BORDER="1">
  <TR>
    <TD>celula11</TD>
    <TD>celula12</TD>
  </TR>
  <TR>
    <TD>celula21</TD>
    <TD>celula22</TD>
  </TR>
</TABLE>
...

<TABLE
BORDER="15"
CELLPADDING="15"
CELLSPACING="15"
>
  <TR>
    <TD>celula11</TD>
    <TD>celula12</TD>
  </TR>
  <TR>
    <TD>celula21</TD>
    <TD>celula22</TD>
  </TR>
</TABLE>
...

```

The image shows a screenshot of a Microsoft Internet Explorer browser window. The browser title is "Exemplu Tabele - Microsoft Internet Explo...". The browser's menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help". The address bar shows "Back" and "Search" buttons. The main content area displays two tables. The first table is titled "Tabel cu dimensiunile pentru margini implicite" and contains a 2x2 grid of cells labeled "celula11", "celula12", "celula21", and "celula22". The second table is titled "Tabel cu dimensiunile pentru margine modificate" and also contains a 2x2 grid of cells with the same labels. The second table has a thick border and significant padding around the cells. Arrows point from the labels "BORDER", "CELLPADDING", and "CELLSPACING" to the corresponding visual elements in the second table.

Figura 7.1 Exemplu folosire atribute pentru directiva TABLE

Atributele **WIDTH** și **HEIGHT**. În mod implicit browserul va dimensiona tabelul în funcție de dimensiunea ferestrei browserului și în funcție de dimensiunea conținutului tabelului. Dacă este necesar se poate specifica o dimensiune explicită a tabelului cu ajutorul atributului **WIDTH**. Această dimensiune poate fi fie o valoare absolută, adică un întreg reprezentând numărul de pixeli ocupați de tabel pe orizontală, fie o valoare procentuală caz în care browserul va afișa tabelul ocupând procentul specificat din dimensiunea ferestrei browserului.

Spre exemplu dacă se vrea ca dimensiunea tabelului să fie de 100 pixeli vom scrie:

```
<TABLE WIDTH="100" >
```

sau dacă dorim ca dimensiunea tabelului să fie jumătate din dimensiunea browserului vom scrie:

```
<TABLE WIDTH="50%" >
```



Dacă dimensiunea tabelului este mai mare decât valoarea specificată de către atributul WIDTH browserul va ignora acest atribut și va dimensiona tabelul.

Similar atributul HEIGHT poate fi folosit pentru a specifica înălțimea tabelului. Browserul va afișa tabelul astfel încât înălțimea lui să nu fie mai mică decât valoarea acestui atribut.

Atributul NOWRAP

Atributul **NOWRAP**. Textul dintr-o celulă a unui tabel este "rupt" și se trece la linie nouă³ în momentul în care dimensiunea acestuia este mai mare decât lungimea celulei. Dacă este specificat atributul NOWRAP atunci browserul va dimensiona celulele astfel încât să nu fie nevoie ca textul conținut în nici una din ele să nu fie "rupt". Dacă se dorește trecerea la o linie nouă în cadrul unei celule se poate folosi una din directivele HTML
 sau <P>.



Test de autoevaluare

7.2 Cum se definește un tabel care să aibă lățimea 80% din fereastra browserului și care să fie afișat centrat în pagină?

Răspunsul corect și comentarii asupra acestuia se găsesc la pagina 82.

7.3 Definirea rândurilor unui tabel

Pentru a insera un nou rând într-un tabel se folosește directiva <TR> a cărei denumire vine de la "Table Row".⁴

³ Operațiunea se numește în limba engleză WRAP

⁴ Rând de tabel

<TR >
<p>Funcționalitate:</p> <ul style="list-style-type: none"> • Definește un rând într-un tabel <p>Atribute:</p> <ul style="list-style-type: none"> • ALIGN • VALIGN • BGCOLOR • NOWRAP <p>Directiva de sfârșit:</p> <ul style="list-style-type: none"> • </TR> este opțională

Atributul ALIGN

Atributul **ALIGN** . Permite configurarea modului de aliniere a textului pentru toate celulele dintr-un rând. Valoarea acestuia nu afectează însă modul de aliniere pentru celulele din alt rând decât cel curent. Valorile posibile sunt "LEFT" pentru stânga, "RIGHT" pentru dreapta și "CENTER" pentru aliniere la centru.

Restul atributelor au aceeași efect ca a atributelor directivei TABLE despre care tocmai am discutat.

7.4 Definirea celulelor unui tabel

Directiva <TD> este folosită în cadrul directivei <TR> pentru a defini o celulă a tabelului și conținutul acesteia. Există cazuri în care într-un tabel primul rând este mai special deoarece conține informații despre tipul datelor conținute în tabel Acest rând poartă denumirea de antet. În HTML acest rând se poate defini cu ajutorul directivei <TH>

Cele două directive acționează asemănător și au aceleași atribute diferența între ele fiind faptul că în cazul directivei <TH> textul este afișat bold și centrat (dacă nu este specificat altfel) iar în cazul <TD> textul este afișat aliniat la stânga (dacă nu este specificat altfel);

<TD > sau <TH>**Funcționalitate:**

- Definește o celulă a unui tabel

Atribute:

- ALIGN
- VALIGN
- COLSPAN
- ROWSPAN
- WIDTH
- HEIGHT
- BGCOLOR
- BACKGROUND

Directiva de sfârșit:

- </TD> respectiv </TH> sunt opționale

Atributul COLSPAN

Atributul **COLSPAN** este folosit pentru a unui celulele învecinate ale unui rând. Valoarea acestui atribut indică numărul de celule de pe rândul curent care vor fi unite astfel încât să formeze o singură celulă.

Spre exemplu dacă vrei ca pe primul rând într-un tabel cu 4 coloane să fie titlul tabelului care să ocupe toată lungimea tabelului și nu numai o celulă se poate folosi atribut COLSPAN atât pentru <TH> cât și pentru <TD> astfel:

```
<TABEL>
<TR>
  <TH COLSPAN="4">
    Titlul principal
  </TH>
</TR>
<TR>
  <TD COLSPAN="4">
    Subtitlul tabelului
  </TD>
</TR>
<TR>
  <TD>Col1</TD>
  <TD>Col2</TD>
  <TD>Col3</TD>
  <TD>Col4</TD>
</TR>
</TABEL>
```

Rezultatul în browser va fi de forma:



Figura 7.3 Exemplu utilizare COLSPAN

Atributul ROWSPAN

Atributul **ROWSPAN** este folosit pentru a unii mai multe celule învecinate de pe aceeași coloană. Se folosește similar cu atributul COLSPAN. Prin urmare dacă dorim să extindem o celulă pe mai multe rânduri vom scrie:

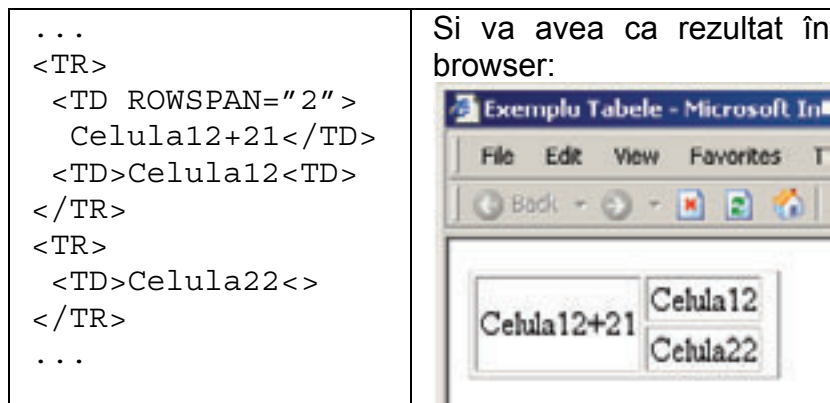


Figura 7.4 Exemplu utilizare ROWSPAN

O celulă se poate extinde pe mai multe celule învecinate de pe același rând și în același timp pe mai multe celule învecinate de pe mai multe rânduri. Acest efect se obține combinând cele două atribute în cadrul aceleiași directive <TD> sau <TH>.

Răspunsuri și comentarii la întrebările din testele de evaluare



Întrebarea 7.1.

Codul HTML pentru tabelul cu un nou rând adăugat va arăta ca cel de mai jos

```
<TABLE>
  <TR>
    <TD>Iepure</TD>
    <TD>25 Kg</TD>
    <TD><IMG SRC="iepuras.gif"></TD>
  </TR>
  <TR>
    <TD>Pisica</TD>
    <TD>10 Kg</TD>
    <TD><IMG SRC="pisica.gif"></TD>
  </TR>
</TABLE>
```

Pentru nelămuriri revedeți secțiunea 7.1.



Întrebarea 7.2.

Pentru a afișa tabelul cerut se pot folosi atributele directivei TABLE ca în exemplul de mai jos:

```
<TABLE ALIGN="CENTER" WIDTH="80%">
```

A se revedea secțiunea 7.2.

Indicații la problemele propuse

Problemele propuse în lucrările de verificare se fac după modelele de exemple prezentate în unitatea de învățare sau după tipicul acestora.

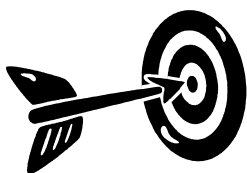
Bibliografie

1. Dave Taylor - Crearea paginilor WEB, Editura Teora 1999, pg.171-198
2. Dorin Cârstoiu, Ecaterina Oltean – Introducere în HTML și XML, Editura Printech București 2004, pg.32-36
3. T.Gugoiu – HTML prin exemple, Editura Teora 2000, pg.39-50

Unitatea de învățare Nr. 8
FORMULARE ÎN PAGINA WEB

Obiectivele Unității de învățare Nr.8	84
8.1 Ce sunt formularele?	84
8.2 Introducerea unui formular în pagina WEB	86
8.3 Elementele HTML folosite în formulare	88
8.4 Alte tipuri de elemente folosite în formulare WEB	90
Răspunsuri și comentarii la întrebările din testele de evaluare	93
Bibliografie	93

Obiectivele Unității de învățare Nr. 8:



După parcurgerea acestei Unității de învățare veți fi capabili:

- Să creați un formular
- Să cunoașteți elementele unui formular
- Să adăugați elemente la un formular
- Să specificați o acțiune pentru un formular
- Să înțelegeți ce înseamnă CGI

8.1 Ce sunt formularele?

Formularele sunt o metodă de colectare a datelor de la utilizator cu scopul prelucrării lor sau a stocării într-o bază de date pentru o prelucrare viitoare.

Formularele în WEB sunt compuse din obiecte ce permit introducerea de text, selectarea unor opțiuni, liste de selecție, harți de imagine sau butoane. Vom discuta despre fiecare în parte în cadrul acestei unități de învățare.

Să vedem care este fluxul de operații și de date pentru o pagină WEB care conține un formular (Vezi figura 8.1):

Fluxul de operații/date

1. Utilizatorul va introduce informații în cadrul formularului
2. Utilizatorul va apăsa un buton special în cadrul formularului
3. Formularul va fi trimis la server
4. Serverul va primi informația și o va prelucra
5. Serverul va trimite înapoi la browser o pagină de răspuns ce poate conține un rezultat al prelucrării datelor trimise.

Prelucrarea informației și alcătuirea răspunsului pe partea de server este o problemă relativ complexă și nu vom discuta despre ea în cadrul acestui modul. Această prelucrare se face folosindu-se un program ce poartă denumirea de **CGI**. CGI reprezintă inițialele de la *Common Gateway Interface* care se poate traduce în limba română prin **interfață comună pentru schimb de date**. Un program sau script CGI poate fi un program scris în orice limbaj de programare. Cele mai folosite limbaje sunt: C/C++, Perl, Python, sau anumite limbaje specializate pentru prelucrarea informațiilor WEB cum ar fi PHP, ASP, JSP sau altele.

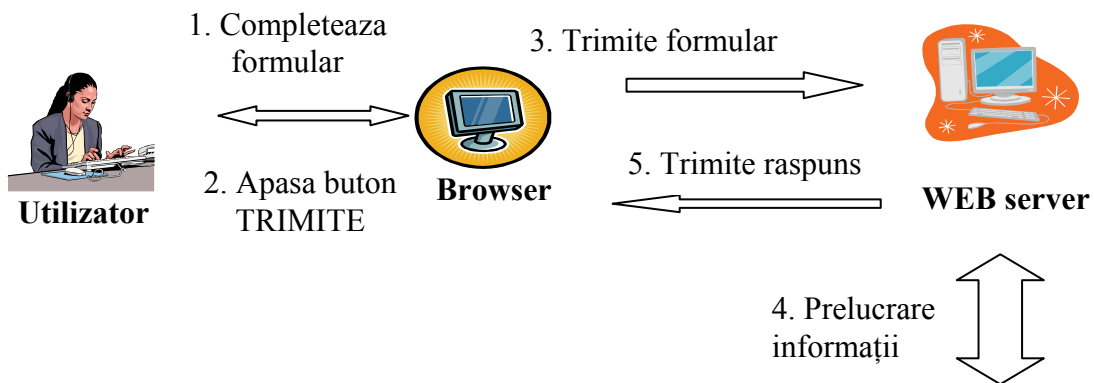


Figura 8.1 Prelucrare formular la server



CGI

În scenariu descris mai sus informația din formular este trimisă la un server spre a fi prelucrată, există însă și posibilitatea ca această informație să fie prelucrată de către browser. Prelucrarea se face în acest caz cu ajutorul unui limbaj de script pe partea de client despre care vei învăța în unitatea de învățare numărul 9. Fluxul operațiilor pentru acest caz este prezentată în figura 8.2.

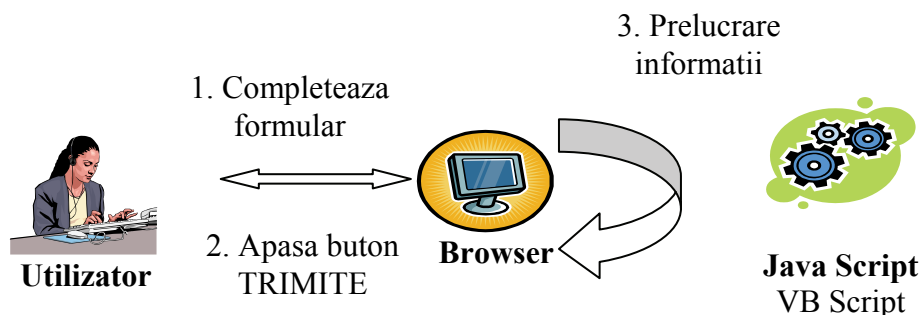


Figura 8.2 Formular prelucrat la client



Test de autoevaluare

8.1 Ce se întâmplă cu datele introduse de către utilizator într-un formular?

Răspunsul se va da în spațiul gol de mai sus. Răspunsurile se găsesc la pagina 93.

8.2 Introducerea unui formular în pagina WEB

Pentru a introduce un formular într-o pagină WEB se folosește directiva pereche `<FORM></FORM>`. Rolul acestei directive este de a delimita formularul și de a defini cu ajutorul atributelor sale modul și de către cine vor fi prelucrate informațiile introduse de către utilizator.

<FORM>
<p>Funcționalitate:</p> <ul style="list-style-type: none"> • Definește un formular <p>Atribute:</p> <ul style="list-style-type: none"> • ACTION • METHOD • NAME • TARGET <p>Directiva de sfârșit:</p> <ul style="list-style-type: none"> • <code></FORM></code> - OBLIGATORIE



- Toate elementele unui formular trebuie incluse în cadrul unui element de tip `<FORM>`.
- Elementele unui formular vor fi afișate în fereastra browserului chiar dacă nu sunt incluse într-un formular însă în acest caz informațiile introduse NU pot fi prelucrate.
- **Un formular NU poate fi definit în interiorul altui formular**
- O pagină WEB poate conține mai multe formulare în același timp.

În figura de mai jos (Figura 8.3) avem un formular simplu așa cum este afișat de către browser. Vom învăța în această secțiune cum se poate realiza un astfel de formular și chiar altele mai complexe:

The screenshot shows a web browser window titled "Exemplu Formular - Microsoft Internet Explor...". The browser's menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help". The address bar contains "Back", "Forward", "Home", and "Search" buttons. The main content area displays a form with the following elements:

- A text input field labeled "Nume:".
- Radio buttons for "Sex:" with "Barbat" selected and "Femeie" unselected.
- A checkbox labeled "Casatorit:" which is currently unchecked.
- A "Trimite" button at the bottom.

Figura 8.3 Exemplu simplu formular

Atributul ACTION **Atributul ACTION.** Serverul de WEB va alege scriptul CGI care va prelucra informația din formular în funcție de valoarea acestui atribut.

Atributul METHOD **Atributul METHOD** se referă la modul în care valorile elementelor din formular vor fi transmise la browser. Există două metode:

- **POST** – Această metodă are 2 etape de transmitere a informațiilor. Prima etapă constă în stabilirea unei conexiuni cu URL-ul specificat de către atributul ACTION. Odată stabilită conexiunea a doua etapă constă în transmiterea informațiilor din formular la server
- **GET** – informațiile din formular sunt adăugate la URL-ul specificat de către atributul ACTION. Scriptul CGI va prelua aceste informații din interiorul URL-ului.

Când se folosește metoda GET și când metoda POST? Iată câteva reguli:



- Dacă este importantă viteza de transmisie a informațiilor atunci va fi folosită metoda GET
- În cazul metodei GET preluarea informațiilor de către aplicația server de prelucrare se face mai ușor
- Dacă securitatea este o problemă atunci este de preferat folosirea metodei POST deoarece în cazul metodei GET informațiile pot fi citite din URL de către persoane neautorizate.

În exemplul de mai jos definim un formular care va transmite informațiile la URL-ul "http://www.despremine.ro" folosind metoda GET

```
<FORM ACTION="http://www.despremine.ro"
      METHOD="GET"
>
...
</FORM>
```

Atributul TARGET Atributul **TARGET** Cu ajutorul acestui atribut se poate redirecționa rezultatul prelucrării formularului într-o altă fereastră de browser.



Test de autoevaluare

8.2 Există mai multe metode de transmitere a datelor dintr-un formular la server? Dacă răspunsul este "DA" care este cea mai nesigură metodă?

Răspunsul corect și comentarii asupra acestuia se găsesc la pagina 93.

8.3 Elementele HTML folosite în formulare

Elementele unui formular se introduc folosind directiva HTML `<INPUT>`. Tipul câmpului din formular și proprietățile acestuia sunt specificate cu ajutorul atributelor directivei după cum vom vedea mai jos.

<INPUT>
<p>Funcționalitate:</p> <ul style="list-style-type: none"> • Definește un câmp într-un formular <p>Atribute:</p> <ul style="list-style-type: none"> • TYPE • NAME • ALIGN • MAXLENGTH • SIZE • CHECKED <p>Directiva de sfârșit:</p> <ul style="list-style-type: none"> • <code></INPUT></code> este opțională

Câmpul TEXTBOX

TEXTBOX – sunt câmpuri de tip text care permit utilizatorului introducerea unui text pe un singur rând. Este util în colectarea informațiilor de tip nume, adresă, dată, telefon, e-mail și multe altele

```
<INPUT TYPE="TEXTBOX" >
```

Browserul va afișa:



Câmpurile TEXTBOX folosesc următoarele atribute:

- **NAME:** Numele variabilei ce conține textul introdus de utilizator ce va fi trimis către scriptul CGI
- **SIZE** specifică lungimea vizibilă a câmpului text. Dimensiunea implicită, în cazul în care acest câmp nu este specificat este 20
- **VALUE** textul implicit ce va fi afișat în acest câmp atunci când acesta va fi prima oară afișat de către browser.
- **MAXLENGTH** specifică numărul maxim de caractere pe care le poate accepta câmpul.

Câmpul PASSWORD

PASSWORD – Sunt câmpuri folosite pentru introducerea de parole. Textul introdus de către utilizator nu va fi vizibil în interiorul acestui câmp. Pentru acest câmp se pot folosi aceleași atribute cu aceeași semnificație ca pentru câmpul Textbox

```
<INPUT TYPE="PASSWORD" >
```

Browserul va afișa:



Se observă că pentru fiecare caracter introdus de utilizator se afișează un caracter “•” sau “*” în loc. Aceasta cu scopul de a proteja informația introdusă în acest câmp de alte persoane neautorizate care văd ecranul utilizatorului în timp ce acesta introduce parola.

Câmpul CHECKBOX

CHECKBOX Este un câmp ce poate avea două stări “*Selectat*” – “*CHECK*” sau “*Neselectat*”. Se folosește atunci când este necesară obținerea unei informații de tip adevărat/fals da/nu de la utilizator

```
<INPUT TYPE="CHECKBOX" >
```

Browserul va afișa:



- neselectat



- selectat

Câmpurile CHECKBOX folosesc următoarele atribute:

- **CHECKED** dacă acest atribut este prezent atunci la încărcarea paginii acest câmp va fi selectat implicit
- **NAME** –numele variabilei cu valoarea corespunzătoare acestui câmp ce va fi trimisă la scriptul CGI.

Câmpul RADIOBUTTON

RADIOBUTTON Permite utilizatorului să selecteze la un moment dat doar o singură opțiune dintr-un grup de opțiuni disponibile. Dacă un grup de câmpuri radiobutton au același nume numai unul dintre ele va putea fi selectat la un moment dat restul fiind automat deselectate.

...

```
<INPUT TYPE="RADIO" NAME="RADIO1" CHECKED >
```

...

```
<INPUT TYPE="RADIO" NAME="RADIO1" >
```

...

Browserul va afișa: Sex: Barbat: Femeie:

Câmpurile CHECKBOX folosesc următoarele atribute:

- **CHECKED** dacă acest atribut este prezent atunci acest câmp va fi selectat implicit la încărcarea pagii.
- **NAME** specifică numele variabilei care va fi transmisă la CGI. Se folosește aceeași valoare pentru butoanele din același grup.

Butonul TRIMITE

BUTTON SUBMIT Creează un buton care odată apăsat de către utilizator va declanșa trimiterea informației din formular la server pentru a fi prelucrată

```
<INPUT TYPE="SUBMIT" VALUE="Trimite" >
```

Browserul va afișa:



Butonul SUBMIT folosește următoarele atribute:

- **VALUE** Definește textul care va fi afișat pe buton
- **NAME** specifică numele variabilei care va fi transmisă la CGI.

Butonul ANULEAZĂ

BUTTON RESET Creează un buton care odată apăsat de către utilizator va șterge informația introdusă până la momentul respectiv în formular de către utilizator. Nici un fel de informație nu va fi trimisă la server.

După apăsarea acestui buton formularul va fi afișat ca și cum ar fi proaspăt reîncărcat. Utilizatorul poate reîncepe introducerea de date în formular imediat după apăsarea acestuia.

```
<INPUT TYPE="SUBMIT" VALUE="Anulează" >
```

Browserul va afișa:



Butonul RESET folosește următorul atribut:

- **VALUE** pentru a defini textul care va fi afișat în browser pe acest buton

8.4 Alte tipuri de elemente folosite în formulare WEB

Liste de selecție

Câmpurile de tip RADIOBUTTON și CHECKBOX oferă posibilitatea utilizatorului să aleagă o anumită opțiune dintr-un set. Au însă un mic dezavantaj: În cazul în care există multe opțiuni posibile spațiul fizic necesar pentru prezentarea tuturor acestor opțiuni este foarte mare.

Pentru aceste cazuri există elementele de tip listă de selecție care se creează în cadrul formularului cu ajutorul directivelor SELECT și OPTION.

<SELECT>	<OPTION>
Funcționalitate: <ul style="list-style-type: none"> • Definește o listă de selecție într-un formular Atribute: <ul style="list-style-type: none"> • NAME • SIZE • MULTIPLE Directiva de sfârșit: <ul style="list-style-type: none"> • </SELECT> Obligatorie 	Funcționalitate: <ul style="list-style-type: none"> • Definește o opțiune într-o listă de selecție Atribute: <ul style="list-style-type: none"> • SELECTED • VALUE Directiva de sfârșit: <ul style="list-style-type: none"> • </OPTION> Obligatorie

Există două tipuri de liste de selecție:

1. Dropdown List – care inițial afișează un singur element pe un singur rând iar în momentul în care utilizatorul selectează această listă se va extinde afișând toate opțiunile disponibile. Dacă valoarea atributului SIZE este 1 atunci lista de opțiuni va fi afișată în această formă.



Figura 8. 4 Exemplu listă selecție de tip Dropdown

2. List Box – Elementele listei sunt afișate într-o zonă rectangulară unele sub altele. Dimensiunea vizibilă a acestei zone se definește cu ajutorul atributului SIZE. Lista de selecție va fi afișată în această formă dacă valoarea atributului SIZE este diferită de 1



Figura 8.5 Exemplu listă de selecție de tip Listbox

Pașii pentru realizarea unei liste de selecție:

1. Introducerea directivei pereche `<SELECT>`
`</SELECT>`
2. Stabilirea cu ajutorul atributului `SIZE` a tipului de listă de selecție
 - i. `SIZE = "1"` – Dropdown List
 - ii. `SIZE > 1` – ListBox
3. Adăugarea elementelor la listă cu ajutorul directivei `<OPTION>`. Un element din listă este introdus de o directivă `<OPTION>` ca în exemplul de mai jos:

```
<SELECT SIZE="1" MULTIPLE>  
<OPTION>Margareta</OPTION>  
<OPTION SELECTED>Trandafir</OPTION>  
<OPTION >Garoafa</OPTION>  
...  
</SELECT>
```

Atributul MULTIPLE

Atributul **MULTIPLE** al directivei `<SELECT>` setează modul de selecție multiplă permițând selectarea mai multor elemente din listă odată. În mod implicit nu se poate selecta decât un singur element din listă la un moment dat.

Atributul SELECTED

Atributul **SELECTED** al directivei `<OPTION>` definește care din elementele listei este setat implicit la încărcarea paginii.

Răspunsuri și comentarii la întrebările din testele de evaluare



Întrebarea 8.1.

La apăsarea butonului Trimite (submit) datele vor fi trimise spre a fi prelucrate în funcție de modul în care a fost specificat, fie către un server fie către un script care rulează în browser. Pentru nelămuriri revedeți secțiunea 8.1.



Întrebarea 8.2.

Există două metode de transmitere a datelor dintr-un formular către server: GET și POST. Cea mai nesigură metodă este GET deoarece aceasta trimite datele din formular ca parte din URL care poate fi vizualizat de către utilizatori neautorizați. Consultați secțiunea 8.2.

Indicații la problemele propuse

Problemele propuse în lucrările de verificare se fac după modelele de exemple prezentate în unitatea de învățare sau după tipicul acestora.

Bibliografie

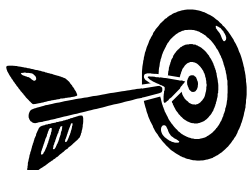
1. Dave Taylor - Crearea paginilor WEB, Editura Teora 1999, pg.253-277
2. Dorin Cârstoiu, Ecaterina Oltean – Introducere în HTML și XML, Editura Printech București 2004, pg.38-46
3. T.Gușoiu – HTML prin exemple, Editura Teora 2000, pg.82-95

Unitatea de învățare Nr. 9

SCRIPT, JAVASCRIPT...

Obiectivele Unității de învățare Nr.9	95
9.1 Ce este JavaScript?	95
9.2 Inserarea unui JavaScript într-un document HTML	95
9.3 Cum și când se execută un script într-o pagină WEB	97
9.4 Atribute de tip Event Handler	99
Lucrare de verificare a Unităților de învățare Nr. 7, 8 și 9	100
Răspunsuri și comentarii la întrebările din testele de evaluare	102
Bibliografie	102

Obiectivele Unității de învățare Nr. 9:



După parcurgerea acestei Unității de învățare vei ști:

- Ce este un script pe partea de client
- Ce este JavaScript
- Ce este un atribut Event Handler
- Să folosești un script într-o pagină WEB

9.1 Ce este JavaScript?

JavaScript a fost inventat pentru a adăuga paginilor WEB capabilitatea de a prelucra informații introduse de utilizator sau de a executa operații fără a fi nevoie de intervenția sau ajutorul serverului de WEB.

Iată câteva exemple practice de întrebuițare a JavaScript într-o pagină WEB:

- Animații
- Prelucrarea unor informații introduse de utilizator fără a fi nevoie de trimiterea datelor la server pentru a fi prelucrate de un CGI
- Realizarea de meniuri dinamice în pagina WEB.

Spre deosebire de limbajul Java care este scris exclusiv pentru programatori, JavaScript este un limbaj simplu scris cu scopul de a fi ușor de asimilat și de folosit de către persoane care nu au experiență anterioară în programare.

9.2 Inserarea unui JavaScript într-un document HTML

O aplicație JavaScript poate fi adăugată într-un document HTML utilizând directiva pereche `<SCRIPT>`. `</SCRIPT>`. Se pot include oricâte directive `<SCRIPT>` într-o pagină WEB în oricare din secțiunile documentului `<HEAD>` sau `<BODY>`.

Singura restricție ar fi că în interiorul acestei directive nu se pot introduce alte directive HTML. Introducerea de directive HTML aici va fi semnalată ca eroare de către browser în momentul în care va afișa pagina.

```
<HEAD>
...
<SCRIPT LANGUAGE="JavaScript">
```

Aici intră scriptul tău

```
</SCRIPT>
...
</HEAD>
<BODY>
...
<SCRIPT LANGUAGE="JavaScript">
    Aici intră scriptul tău
</SCRIPT>
...
</BODY>
```

Exemplul 9.1 Inserare script în document HTML



Browsersle care nu suportă JavaScript vor trata scriptul din interiorul acestei directive ca pe un text normal și prin urmare îl va afișa în pagină. Acest lucru nu este de dorit adesea. Pentru a nu ajunge într-o astfel de situație se recomandă ca scriptul din interiorul acestei directive să fie inclus într-un comentariu HTML, ca în exemplul de mai jos:

```
<SCRIPT LANGUAGE="JavaScript">
<!
    Aici intră scriptul tău
-->
</SCRIPT>
```

<SCRIPT>
<p>Funcționalitate:</p> <ul style="list-style-type: none"> • Inserează un script în documentul HTML
<p>Atribute:</p> <ul style="list-style-type: none"> • LANGUAGE • SRC • TYPE
<p>Directiva de sfârșit:</p> <ul style="list-style-type: none"> • </SCRIPT> OBLIGATORIE

Atributele LANGUAGE Si TYPE

Atributele **LANGUAGE** și **TYPE** JavaScript sunt unele dintre cele mai populare limbaje de script folosite în WEB, însă există și altele precum VBScript. Rolul acestor atribute este de a specifica browserului ce tip și ce versiune de script este inclus în interiorul directivei.

Este de ajuns folosirea fie a atributului LANGUAGE, fie a atributului TYPE. NU este nevoie a fi folosite ambele în același timp.

Cele mai utilizate valori pentru atributul LANGUAGE sunt "JavaScript" și "VBScript". Același lucru se poate specifica utilizând valorile "text/javascript" respectiv "text/vbscript" pentru atributul TYPE.

Atributul SRC

Atributul **SRC**. Pentru cazurile în care un anumit script are dimensiuni foarte mari sau este folosit de către mai multe pagini WEB acesta poate fi scris într-un fișier separat. Includerea acestor scripturi pentru a putea fi referite în pagina HTML curentă se face folosind atributul **SRC**.

Valoarea atributului este URL-ul la care se găsește fișierul ce conține scriptul. Prin urmare pentru a include un script definit în alt fișier vom scrie:

```
<SCRIPT LANGUAGE="JavaScript"
SRC="scripts/scriptulmeu.js">
```



Test de autoevaluare

9.1 De ce este recomandat plasarea codului unui script în interiorul unui comentariu HTML?

Răspunsul corect și comentarii asupra acestuia se găsesc la pagina 102.

9.3 Cum și când se execută un script într-o pagină WEB

Scripturile se pot executa fie la încărcarea paginii, dacă scriptul este definit în secțiunea <BODY> fie la apariția unui eveniment care are asociat un script.

Una din cele mai importante facilități oferite de JavaScript este posibilitatea de a detecta anumite evenimente care au loc în pagină și de a reacționa la acestea.

Exemple de astfel de evenimente pot fi: trecerea cu cursorul mouse-ului peste un anumit obiect (hyperlink, imagine, buton etc.) , încărcarea paginii, descărcarea paginii, apăsarea unei taste etc.

Majoritatea directivelor pot fi configurate să răspundă la astfel de evenimente prin executarea unui script JavaScript la producerea acestuia.

Hai sa vedem ce scriem pentru ca o imagine să declanșeze execuția unui script JavaScript atunci când utilizatorul "trece" cu cursorul mouse-ului pe deasupra imaginii:

```
<HEAD>
  <SCRIPT LANGUAGE="JavaScript">
    function mesajIepuras()
    {
      alert("Salut! Sunt iepurasul pinki!")
    }
  </SCRIPT>
</HEAD>
<BODY>
...
<IMG SRC="iepuras.gif"
      ALT="Bugs Bunny"
      onMouseOver="mesajIepuras();">
...
```

În browser la încărcarea paginii va fi încărcată imaginea iar în momentul în care cursorul mouse-ului va trece peste imagine va fi afișată o fereastră cu mesaj ca în figura de mai jos:

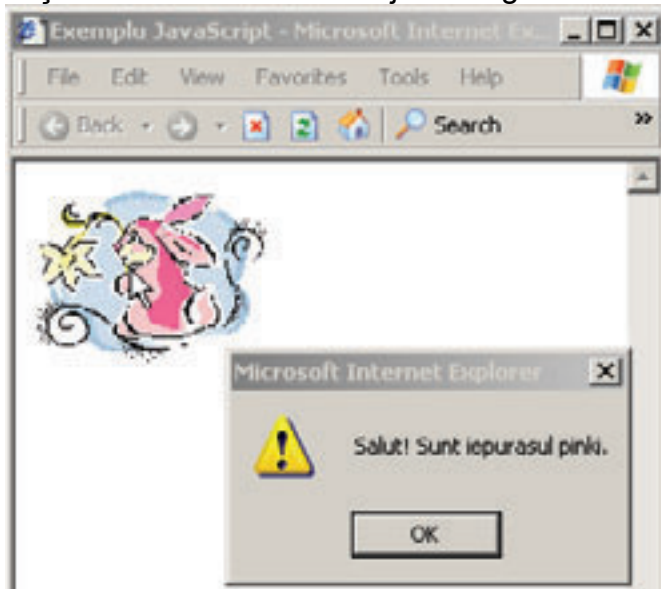


Figura 9.1 Exemplu declanșare execuție script

De ce a apărut mesajul? Deoarece la definirea directivei IMG mai sus am adăugat atributul **onMouseOver** care are ca valoare numele unei funcții JavaScript.

Asemănător pentru fiecare tip de eveniment există un atribut asociat. Dacă unul din aceste atribute este definit în cadrul directivei HTML,

iar evenimentul asociat are loc la un anumit moment dat atunci se va executa funcția JavaScript definită de valoarea atributului.

Atribute Event Handler

Acest tip de atribute se numesc **Atribute Event Handler**. În limba română aceasta se poate traduce prin “atribute pentru procesarea evenimentelor”. Pentru majoritatea directivelor HTML se pot defini astfel de atribute asociind astfel un script care să fie executat la apariția evenimentului. Există doar câteva excepții de directive care nu suportă astfel de atribute, acestea sunt: <HEAD>, <BODY>, <BASEFONT>, ,
, <HTML>, <SCRIPT>, <TITLE>, <APPLET>, <SCRIPT> și <FRAME>

Execuția scriptului declanșat poate consta în simpla afișare a unui mesaj ca mai sus sau poate fi o prelucrare complexă de date introduse de către utilizator.



Test de autoevaluare

9.2 Când se execută un script JavaScript într-o pagină WEB?

- a> La încărcarea paginii
- b> La apariția unui eveniment
- c> la cererea vizitatorului paginii WEB
- d> la o anumita ora stabilita de către vizitatorul paginii WEB

Răspunsul corect și comentarii asupra acestuia se găsesc la pagina 102.

9.4 Atribute de tip Event Handler

Mai jos vom enumera cele mai utilizate atribute Event Handler suportate de către directivele HTML, însoțite de o scurtă explicație pentru fiecare în parte. Pentru o listă completă a acestor atribute consultă una din lucrările din bibliografia acestei Unității de învățare.

Unele evenimente apar mai rar și nu pot fi asociate decât anumitor directive. Pentru acestea vom enumera directivele care le suportă.

Directive suportate

- **onClick** Evenimentul apare atunci când utilizatorul execută un click de mouse pe elementul respectiv. Este suportat de majoritatea directivelor.
- **onDbIcIck** Evenimentul apare atunci când utilizatorul execută un dublu click de mouse pe elementul respectiv. Este suportat de majoritatea directivelor.
- **onMouseOver** Evenimentul apare atunci când utilizatorul trece cu cursorul mouseului pe deasupra elementului. Este suportat de majoritatea directivelor.

- **onSubmit** Evenimentul apare atunci când există un formular în pagină în momentul în care utilizatorul apasă un buton de tip “**submit**” (Trimite). Directivele care suportă acest atribut sunt <FORM> și <BODY>.
- **onReset** Evenimentul apare atunci când există un formular în pagină în momentul în care utilizatorul apasă un buton de tip “**reset**” (Anulează). Directivele care suportă acest atribut sunt <FORM> și <INPUT>.
- **onKeyPress** Evenimentul apare atunci când utilizatorul apasă o tastă. Este suportat de majoritatea directivei.



Test de autoevaluare

9.3 Ce rol au atributele Event Handler?

Răspunsul corect și comentarii asupra acestuia se găsesc la pagina 102.

Lucrare de verificare a Unităților de învățare Nr. 7, 8 și 9



1. Să se realizeze un tabel HTML cu următoarele caracteristici:

1. tabelul să fie centrat și să ocupe 50% din lungimea ferestrei browserului
2. Să aibă 4 rânduri
3. Să aibă 3 colane
4. Datele pe coloana 1 să fie aliniate la stânga
5. Datele pe coloana 2 să fie centrate
6. Datele pe coloana 3 să fie aliniate la dreapta

Predați fișierul HTML în format electronic ca rezultat al rezolvării lucrării de verificare

Pe lângă informațiile prezentate în manual se poate utiliza reperul bibliografic numărul 2 și 3 din bibliografia unității de învățare nr.7.

Nr. de puncte **9** :

1 – 3p, 2 – 1p, 3 – 1p, 4 – 1p, 5 – 1p, 6 – 1p

2. Realizați un formular HTML care să poată fi folosit la un sondaj de opinie (alege singur subiectul). Acesta trebuie să conțină câmpuri textbox, butoane radio și liste de selecție

Predați fișierul HTML în format electronic ca rezultat al

rezolvării lucrării de verificare

Pe lângă informațiile prezentate în manual se poate utiliza reperul bibliografic numărul 2 și 3 din bibliografia unității de învățare nr.8.

Nr. de puncte **8** (2p – utilizarea directivei FORM, 2p -utilizarea câmpuri de tip textbox, 2p – utilizarea câmpuri de tip buton radio; 2p - utilizarea câmpurilor de tip listă de selecție)

3. Realizați o pagină HTML care să conțină o directivă HTML `<A>` care să aibă următorul comportament: când utilizatorul execută dublu-click pe acest hyperlink să se afișeze mesajul “De doua ori click!” într-o fereastră separată (o fereastră de tip dialog). Pentru aceasta se va utiliza JavaScript și atribute de tip Event Handler.

Predați fișierul HTML în format electronic ca rezultat al rezolvării lucrării de verificare

Pe lângă informațiile prezentate în manual se poate utiliza reperul bibliografic numărul 1 și 2 din bibliografia unității de învățare.

Nr. de puncte **8** (4p – pentru definirea corectă a atributului EventHandler pentru directiva `<A>`, 4p – definirea și folosirea secvenței JavaScript pentru afișarea mesajului)

Răspunsuri și comentarii la întrebările din testele de evaluare



Întrebarea 9.1.

Este recomandat definirea scripturilor în interiorul comentariilor HTML pentru cazurile în care browserul cu care este vizualizată pagina nu suportă JavaScript. În aceste cazuri browserul va afișa textul scriptului în loc să îl execute. Pentru a preveni afișarea scriptului în pagină acesta trebuie introdus în interiorul unui comentariu HTML. A se revedea secțiunea 9.2.



Întrebarea 9.2.

Un script Java Script se execută la încărcarea paginii dacă acesta este definit în interiorul directivei<BODY> sau la apariția unui anumit eveniment. Variante corecte de răspuns: a) și b). A se revedea secțiunea 9.3.



Întrebarea 9.3.

Atributele Event Handler au rolul de a defini acțiunea (scriptul) care se va executa în momentul apariției unui eveniment. Consultați secțiunea 9.4.

Indicații la problemele propuse

Problemele propuse în lucrările de verificare se fac după modelele de exemple prezentate în unitatea de învățare sau după tipicul acestora.

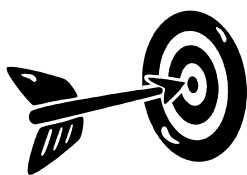
Bibliografie

1. Richard Wagner, R.Allen Wyke – Java Script, Editura Teora 2000, pg.47-59, 64-76
2. Mihaela Brut, Sabin Buraga - Prezentrari multimedia pe Web. Limbajele XHTML + TIME și SMIL Editura Polirom 2003, pg. 13-22, 30-33
3. Călin Ioan Acu – Optimizarea paginilor Web, Editura Polirom 2003, pg.182-267

Unitatea de învățare Nr. 10
EXECUTABILE ȘI MULTIMEDIA ÎN PAGINA WEB

Obiectivele Unității de învățare Nr.10	104
10.1 Java și Java APPLET	104
10.2 Obiecte ACTIVE X	106
10.3 Fișiere multimedia în pagina WEB	107
10.4 Adăugarea clipurilor multimedia la o pagină Web	108
Răspunsuri și comentarii la întrebările din testele de evaluare	110
Bibliografie	110

Obiectivele Unității de învățare Nr.10:



După parcurgerea acestei Unității de învățare vei ști:

- Ce este un Applet Java
- Cum se folosește un applet Java într-o pagină WEB
- Ce este un obiect Active X
- Cum se folosește un obiect Active X într-o pagină WEB
- Ce sunt și cum se pot folosi fișierele multimedia într-o pagină Web

10.1 Java și Java APPLET

Java	Java este un limbaj de programare dezvoltat de compania “Sun Microsystems”. Este un limbaj care oferă întreg suportul pentru programarea orientată pe obiecte și a fost creat cu scopul de putea fi utilizat fără nici un fel de diferență pe orice sistem care suportă Java (<i>cross-platform</i>). Un alt mare avantaj al limbajului Java este faptul că programele scrise în Java și compilate pentru un sistem de operare vor rula fără a fi recompilate pe orice alt sistem de operare care suportă Java. Acest lucru este posibil deoarece programele Java nu rulează folosind direct resursele oferite de sistemul de operare ci rulează într-un alt program numit Java Virtual Machine care se traduce prin Mașina Virtuală Java.
Avantaj	În literatura de specialitate se face referire la ea prin acronimul său - JVM. După cum îi spune și numele acest program creează un mediu virtual în care un program Java poate rula.
APPLET Java	Programele Java pot fi folosite într-o pagină Web sub formă de APPLET Java. Un applet Java este un program executabil într-un JVM. Aceste programe se găsesc în fișiere cu extensia <i>.class</i> , și pot fi incluse direct în pagina Web folosind directiva HTML <code><APPLET></code> împreună cu directiva <code><PARAM></code> după cum vom vedea mai jos. Browsersle care suportă Java creează un astfel de mediu pe care programele Java de tip APPLET pot să îl folosească pentru a rula. Dar atenție nu toate browserele suportă Java! Și deci la realizarea unei pagini de Web trebuie ținut cont de acest lucru.

<APPLET>	<PARAM>
<p>Funcționalitate:</p> <ul style="list-style-type: none"> • Introduce un Applet Java în pagină <p>Atribute:</p> <ul style="list-style-type: none"> • CODE • CODEBASE • ALIGN • WIDTH • HEIGHT <p>Directiva de sfârșit: </APPLET> OBLIGATORIE</p>	<p>Funcționalitate:</p> <ul style="list-style-type: none"> • Permite setarea unui parametru de intrare pentru un applet <p>Atribute:</p> <ul style="list-style-type: none"> • NAME • VALUE <p>Directiva de sfârșit: </PARAM> OPTIONALĂ</p>

Pași folosiți

Pașii pentru a introduce un applet Java într-o pagină Web sunt următorii:

Pasul 1: Fișierul *.class* ce conține appletul Java va trebui copiat în directorul care se află fișierul HTML, sau în directorul specificat de către atributul CODEBASE.

Pasul 2: Appletul Java aflat în fișierul *.class* va fi introdus în pagina Web folosind directiva HTML <APPLET>. Exemplul de mai jos ilustrează modul în care se poate introduce într-o pagină HTML un simplu applet Java fără parametri de intrare:

```
<APPLET CODE="appletulmeu.class"
        WIDTH="300"
        HEIGHT="200">
</APPLET>
```

Dacă appletul are nevoie de anumiți parametri de intrare pentru a rula aceștia pot fi specificați cu ajutorul unui set de directive <PARAM> în interiorul elementului <APPLET> ca în exemplul de mai jos unde appletul primește ca parametri de intrare un nume și o parolă:

```
<APPLET CODE="appletulmeu.class"
        WIDTH="300"
        HEIGHT="200">
  <PARAM NAME="nume" VALUE="ana">
  <PARAM NAME="parola" VALUE="bu12cc">
</APPLET>
```

Hai să vedem ce semnificație are și unde poate fi folosit fiecare atribut în parte:

**Atributul
CODE**

Atributul **CODE** – specifică numele fișierului *.class* sau *.jar* care conține codul executabil al applet-ului Java folosit. Acest parametru este obligatoriu.

**Atributul
CODEBASE**

Atributul **CODEBASE** – specifică locația unde poate fi găsit fișierul specificat de atributul CODE. Dacă acest atribut lipsește fișierul *.class* va fi căutat în directorul unde se află documentul HTML.

**Atributele
WIDTH si
HEIGHT**

Atributele **WIDTH** și **HEIGHT** – specifică dimensiunea pe orizontală respectiv pe verticală a spațiului rezervat de browser pentru afișarea appletului. Este recomandată specificarea acestor atribute pentru ca appletul să afișeze datele corect.

Pentru directiva <PARAM> atributul **NAME** este folosit pentru a specifica numele, iar atributul **VALUE** pentru a specifica valoarea parametrului de intrare care va fi transmis appletului.

Appleturile Java extind posibilitățile oferite de HTML și/sau browser permițând realizarea unor pagini puternic interactive, cu interfață grafică ce nu ar putea fi în mod normal creată cu facilitățile puse la dispoziție de către limbajul HTML.



Test de autoevaluare

10.1 Un applet Java rulează pe calculatorul client sau rulează pe server și prezintă doar rezultatul în fereastra browserului?

Răspunsul corect și comentarii asupra acestuia se găsesc la pagina 110.

10.2 Obiecte ACTIVE X

Active X este o tehnologie dezvoltată de către compania Microsoft care permite adăugarea unor componente executabile în pagina WEB. Obiectele sau controalele Active X sunt asemănătoare cu un applet Java. Spre deosebire de acestea, însă, pot avea un control mult mai puțin restricționat asupra mașinii pe care rulează, ceea ce poate fi perceput ca o problemă de securitate de către anumiți utilizatori.

Avantaj

Marele avantaj al obiectelor Active X față de appleturile Java este că sunt transferate de la server o singură dată. După ce au fost transferate ele devin parte integrantă din sistemul de operare. Execuțiile ulterioare ale obiectului se vor face astfel rapid. Datorită acestui avantaj și al altor caracteristici, Active X este folosit în special

pentru aplicațiile grafice livrate către utilizator prin intermediul browserului.

Dezavantaj

Marele dezavantaj al obiectelor Active X este că acestea nu pot rula decât pe un calculator care rulează sistemul de operare Windows.

Controalele Active X pot fi dezvoltate folosind diferite limbaje de programare, cum ar fi Microsoft Visual Basic, Microsoft Visual C++, și chiar Java.

Un control Active X poate fi adăugat într-o pagină Web folosind directiva HTML `<OBJECT>` în mod similar cu directiva `APPLET` pentru appleturi Java. Ca și în cazul elementului `APPLET` parametrii de intrare pot fi specificați utilizând directiva `<PARAM>`.

10.3 Fișiere multimedia în pagina WEB

Fișierele multimedia sunt fișiere care conțin sunete, secvențe muzicale, sau chiar clipuri video. Acestea sunt din ce în ce mai folosite în paginile Web.

În momentul în care browserul detectează un fișier multimedia fie va lansa un program extern fie va folosi o facilități internă (plug-in) pentru a vă rula clipul multimedia aflat în fișierul respectiv. Ce program extern va folosi pentru a rula clipul multimedia depinde de mai mulți factori, precum tipul sistemului de operare sau tipul și versiunea browserului. De aceea în calitate de creator al unei pagini web nu poți controla direct programul în care va rula clipul multimedia pe care îl oferi în pagină.

Metode

Există două metode de a livra conținutul fișierelor multimedia către browserul unui utilizator:

- Metoda statică sau ***non-streaming*** caz în care fișierul multimedia trebuie să fie complet descărcat pe calculatorul utilizatorului înainte ca acesta să fie rulat.
- Metoda dinamică sau ***streaming*** caz în care clipul conținut de un fișier este rulat aproape imediat ce a fost referit transferul restului de fișier făcându-se pe măsură ce clipul este rulat.

Avantaj

Avantajul primei metode este că nu necesită existența nici unui program pe partea de server care să facă posibilă rularea clipului.

Dezavantaj

Principalul dezavantaj al acestei metode este că în cazul în care dimensiunea fișierelor este mare, durata de transfer crește foarte mult și deci și timpul până când clipul poate fi rulat. De asemenea datorită faptului că fișierul ajunge direct pe calculatorul utilizatorului face mai dificil pentru creatorii acelor clipuri să își protejeze drepturile de autor asupra unor lucrări.

Streaming

Pentru a înlătura aceste dezavantaje majore a fost dezvoltată metoda dinamică de transfer a fișierelor multimedia referită în literatura de specialitate cu termenul de "**streaming**". Această tehnologie presupune existența următoarelor componente software:

- *Streaming server* este o componentă care gestionează cererile de streaming pentru diferite formate multimedia de la mai mulți utilizatori simultan, folosind în mod eficient resursele calculatorului server.
- *Encoder* – Este o componentă software care convertește un fișier dintr-un anumit format multimedia într-un format potrivit pentru streaming.
- *Player* - este o aplicație software care rulează pe mașina utilizatorului și care are rolul de a realiza la cerere conexiunea cu serverul de streaming și de a rula clipul servit de acesta.

Avantaje si dezavantaje

Avantajele acestei metode sunt clare după cum am descris mai sus deoarece utilizatorul nu trebuie să aștepte transferul complet al clipului multimedia înainte de a putea să-l vizioneze. Dezavantajele însă se leagă de faptul că această metodă necesită existența unui server de streaming care este de obicei scump și destul de dificil de configurat și menținut.

10.4 Adăugarea clipurilor multimedia la o pagină Web

Pentru a include un clip multimedia într-o pagină Web se poate folosi fie directiva <A> fie directiva <EMBED>. Fișierele multimedia audio pot fi de asemenea rulate ca fundal sonor la o pagină Web cu ajutorul unor comenzi JavaScript.

Prin urmare poți folosi o directivă simplă de tip ancoră ca în exemplul de mai jos:

```
<A HREF="audio/clipulmeu.avi">  
No.1 clip (1.3Mo)  
</A>
```

Atunci când utilizatorul selectează hiperlinkul de mai sus browserul va transfera pe hardiskul local clipul respectiv și îl va rula cu ajutorul unei aplicații externe sau folosind facilitățile oferite de către browser în acest sens.

Directiva EMBED

Folosirea directivei <EMBED> oferă mai multe opțiuni pentru rularea clipurilor multimedia decât în cazul folosirii unui simplu hiperlink. Browserul va afișa în pagină în locul acestei directive o interfață grafică ce va permite utilizatorului să controleze derularea clipului – să-l oprească sau să-l repornească, să deruleze înainte și înapoi etc.

Cu ajutorul atributelor directivei <EMBED> se poate configura ca clipul să fie rulat automat, ori să poată fi rulat în buclă sau doar o singură dată. Iată un exemplu de folosire a acestei directive:

```
<EMBED  
SRC="http://www.siteulmeu.ro/clip1.wav" >
```

Singurul atribut obligatoriu pentru această directivă este **SRC**. Valoarea lui reprezintă URL-ul unde se află clipul multimedia care se dorește a fi rulat în pagină.

Atributele **WIDTH** și **HEIGHT** permit definirea dimensiunilor interfeței grafice de control afișate în browser.

Atributul AUTOSTART

Atributul **AUTOSTART** poate lua una din valorile TRUE sau FALSE și specifică browserului să ruleze la încărcarea paginii clipul specificat de atributul SRC.

Atributul LOOP

Atributul **LOOP** poate lua una din valorile TRUE sau FALSE și specifică browserului să ruleze sau nu în buclă clipul specificat de atributul SRC.

Atributul HIDDEN

Atributul **HIDDEN** poate lua una din valorile TRUE sau FALSE și specifică browserului să afișeze sau să nu afișeze interfața grafică de control în pagina Web.



Test de autoevaluare

10.2 Realizați o pagină web care să includă clipul de la URL-ul <http://www.multamuzica.ro/melo1.wav> care să fie rulată automat la încărcarea paginii și care să fie cântată în buclă

Răspunsul corect și comentarii asupra acestuia se găsesc la pagina 110.

Răspunsuri și comentarii la întrebările din testele de evaluare



Întrebarea 10.1.

Un applet Java rulează întotdeauna direct pe mașina client după ce a fost transferat de la server. Acesta fiind unul din avantajele acestei tehnologii pentru că în acest fel serverul este eliberat de anumite sarcini. În același timp, însă poate fi și un dezavantaj pentru cazurile în care mașina client nu este destul de puternică. Consultați secțiunea 10.1.



Întrebarea 10.2.

Codul HTML corespunzător pentru a include acest clip este:

```
<EMBED  
SRC="http://www.multamuzica.ro/mel01.wav"  
LOOP="TRUE"  
AUTOSTART="TRUE"  
>
```

Revedeți indicațiile din secțiunea 10.4.

Indicații la problemele propuse

Problemele propuse în lucrările de verificare se fac după modelele de exemple prezentate în unitatea de învățare sau după tipicul acestora.

Bibliografie

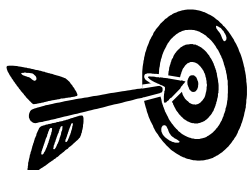
1. Cioata Mihai – ActiveX. Concepte și aplicații Editura Polirom, pg.95-102, 174-175
2. Călin Ioan Acu – Optimizarea paginilor Web, Editura Polirom 2003, pg.95-99, 144-156
3. Mihaela Brut, Sabin Buraga - Prezentați multimedia pe Web. Limbajele XHTML + TIME și SMIL Editura Polirom 2003, pg. 13-22 și 233-247

Unitatea de învățare Nr. 11

XML ȘI FOLOSIREA LUI IN PAGINA WEB.

Obiectivele Unității de învățare Nr.11	112
11.1 Introducere în XML	112
11.2 Caracteristici ale XML	113
11.3 Sintaxa XML	114
11.4 Modul de folosire a XML în pagina Web	116
Răspunsuri și comentarii la întrebările din testele de evaluare	119
Bibliografie	119

Obiectivele Unității de învățare Nr.11:



Principalele obiective ale Unității de învățare Nr. 11 sunt:

- Ce este limbajul XML și la ce poate fi folosit
- Ce este și care este structura unui document XML
- Cum poate fi folosit un document XML într-o pagină HTML

11.1 Introducere în XML

Limbajul XML (**eXtended Markup Language**) este un nou limbaj de adnotare, care este folosit intens de tehnologiile software actuale pentru *schimburi de informații* între diverse aplicații din Internet sau rețele. În cazul în care aceste aplicații funcționează conform unor standarde diferite, XML este limbajul comun în care acestea pot schimba informații. Această facilitate este foarte importantă în cazul comunicării datelor între companii și firme. Dacă este folosit XML nici una din companii nu trebuie să cunoască modul de structurare a datelor în baza de date a partenerului ci doar să știe structura XML de export a acestor date pentru a avea acces la informațiile necesare.



XML nu este un înlocuitor al HTML, ci este mai degrabă un complement al acestuia. În vreme ce în cazul HTML atenția este orientată spre modul în care trebuie să fie afișată informația conținută de directivele HTML, XML este realizat cu unicul scop de a oferi suport pentru structurarea informațiilor.

La realizarea acestui limbaj s-au stabilit următoarele obiective:

- XML va fi compatibil cu SGML (un alt limbaj de adnotare).
- Documentele XML vor putea fi ușor citite și interpretate de către om.
- Documentele XML vor putea fi proiectate și realizate ușor.
- Programele pentru a interpreta documentul XML trebuie să poată fi realizate ușor.
- Să fie destul de flexibil pentru a putea fi utilizat într-o gamă largă de aplicații.

**Limbajul XML
nu poate
executa nici o
operație**

Limbajul XML nu este conceput să ofere instrucțiuni pentru execuția unor operații, ci este conceput pentru a "împacheta" informația cu scopul de a fi trimisă sau stocată. Pentru a face ceva cu aceste date cineva trebuie să scrie un program într-un limbaj de programare oarecare care să interpreteze aceste date și să le folosească într-un anumit scop.

Iată un exemplu de fișier XML:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<articol>
  <titlu>
    Impozitarea tranzactiilor imobiliare
  </titlu>
  <descriere>
    Terenurile sunt din ce in ce mai
    scumpe in Capitala dupa ce ...
  </descriere>
  <URL>
    http://stiri.ro/articol123.html
  </URL>
</articol>
<articol>
...
</articol>
```

În exemplul de mai sus se constată că este vorba despre un document XML care ar putea fi folosit de o aplicație ce afișează revista presei. Documentul XML din exemplul de mai sus conține informații despre articolele disponibile în ziarele de azi

- Nume articol - <titlu>
- O scurtă descriere - <descriere>
- Locația unde poate fi găsit acest articol - <URL>

Se vede că în acest document sunt prezente numai informații cu privire la structura sau tipul datelor și bineînțeles datele propriu-zise. Acestea vor putea fi folosite de către o aplicație care să afișeze revista presei folosind datele din acest document XML care ar putea fi transmise periodic de la un server.

11.2 Caracteristici ale XML

Principalele caracteristici ale XML sunt:

- **XML nu are directive predefinite**, acestea vor fi realizate în momentul proiectării documentului XML în funcție de scopul acestuia și tipul sau structura datelor.
- Pentru descrierea directivelor folosite, XML utilizează un set de reguli denumit **DTD (Document Type Definition)** sau o **schemă XML**. Împreună cu această schemă se spune despre XML că este autodescriptiv.
- Documentul XML trebuie să respecte strict regulile definite în DTD pentru a fi un document **valid**.

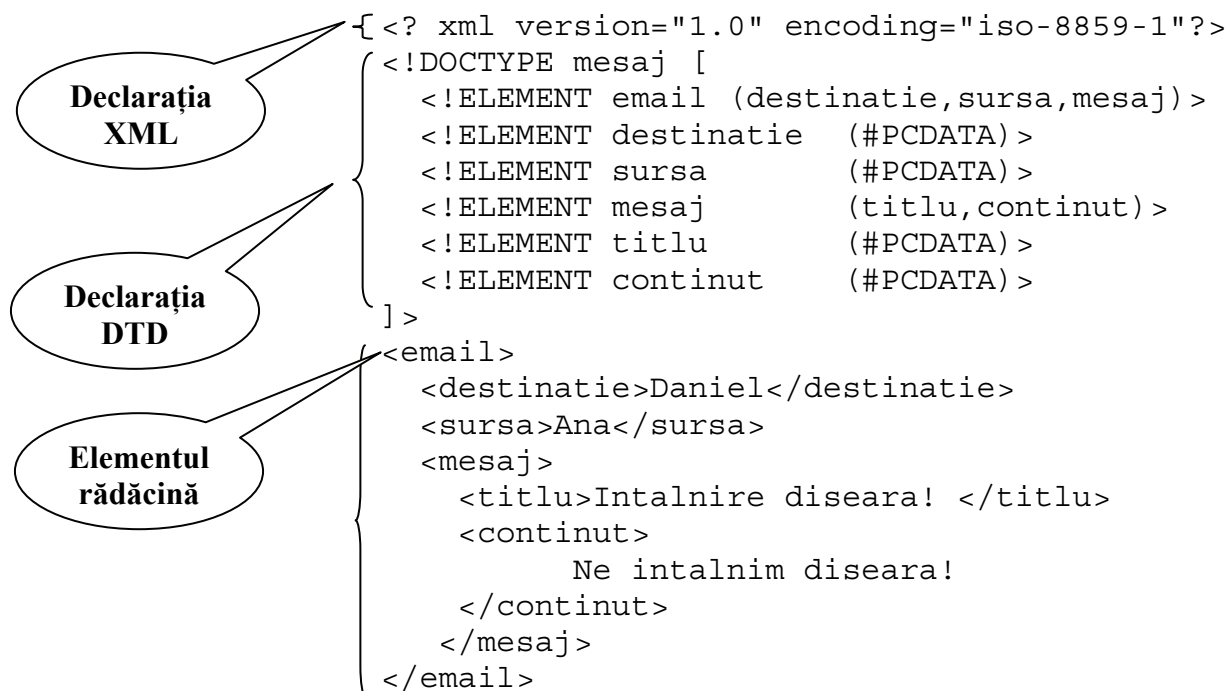
- Documentele XML au o structură strictă (**Well-formed**). Asta înseamnă că documentul trebuie să respecte anumite reguli de sintaxă prezentate în Unitatea de învățare următoare.
- XML este un limbaj de adnotare complementar cu HTML și **NU** înlocuiește HTML. Cele două limbaje, după cum s-a arătat, au scopuri distincte: HTML afișează informații ce apar în pagină Web, în timp ce XML descrie aceste informații.

11.3 Sintaxa XML

Un document XML este format din două blocuri principale astfel:

- **Antetul** documentului care cuprinde la rândul său:
 - **Declarația XML** – Se află întotdeauna pe prima linia a documentului și conține informații despre *versiunea XML și standardul de codificare al caracterelor*.
 - **Declarația DTD** care conține informații despre structura documentului. Documentul este verificat dacă este valid sau nu conform cu această declarație
- **Documentul XML propriu-zis** Un document XML are o structură arborescentă având un singur element rădăcină marcat de o pereche de directive XML. Toate celelalte elemente sunt descendenți și conținute în interiorul elementului rădăcină. Un element poate conține la rândul său alte subelemente.

Un exemplu complet:



În exemplul de mai sus:

- Prima linie reprezintă Declarația XML,

- Este urmată de Declarația DTD care conține reguli despre structura documentului într-un anumit format
- Documentul XML propriu-zis are ca element rădăcină elementul reprezentat cu ajutorul directivei `<email></email>`. Elementele care urmează marcate de directivele `<destinatie>`, `<sursa>` și `<mesaj>` sunt descendenți de rangul 1. Elementul `<mesaj>` are la rândul său descendenți elementele marcate de directivele `<titlu>` și `<conținut>`. Nivelul de imbricare poate continua atât cât este nevoie și nu sunt stabilite limite.

Directive XML

Directivele XML sunt identificatori pentru elementele XML și sunt plasate ca și în cazul directivelor HTML între paranteze ascuțite “<” și “>” la care sunt regulile pentru folosirea directivelor XML:

Orice directivă, ex. `<directivă>`, trebuie să aibă o directivă pereche de încheiere ex `</directivă>`. Omiterea directivei de încheiere nu este permisă ca în cazul unor directive HTML

Directivele XML sunt sensibile la scrierea cu majuscule sau litere mici prin urmare directiva `<Titlu>` este diferită de directiva `<titlu>`.

În cazul în care se folosesc mai multe niveluri de imbricare ordinea închiderii directivelor trebuie să fie inversă decât ordinea în care au fost deschise.

Pentru elementele care nu conțin text se poate folosi o altă modalitate de definire a directivelor. Iată cum:

```
<directiva />
```

Se observă că s-a folosit un caracter “/” după numele elementului. În acest caz NU se mai specifică directiva pereche de încheiere

Atribute XML

Directivele XML pot avea atribute pentru a putea specifica informații suplimentare despre acel element. Atributele sunt definite sub formă de perechi nume-valoare. Valorile atributelor se scriu obligatoriu între ghilimele ca în exemplul de mai jos:

```
...
<mesaj data="01/08/2005">
...
</mesaj>
...
```

În XML se pot folosi comentarii care au o sintaxă similară cu cea din HTML, adică:

```
<!-- Orice text explicativ --!>
```

Acestea pot apărea oriunde în documentul XML.



Test de autoevaluare

11.1 XML poate fi considerat ca fiind o alta versiune de HTML?

Răspunsul corect și comentarii asupra acestuia se găsesc la pagina 119.

11.4 Modul de folosire a XML în pagina Web

Așa cum am spus mai sus XML a fost realizat ca un limbaj folosit pentru a structura datele pentru a fi trimise spre prelucrare în cadrul altor aplicații. HTML are ca principal scop definirea modului în care sunt afișate date.

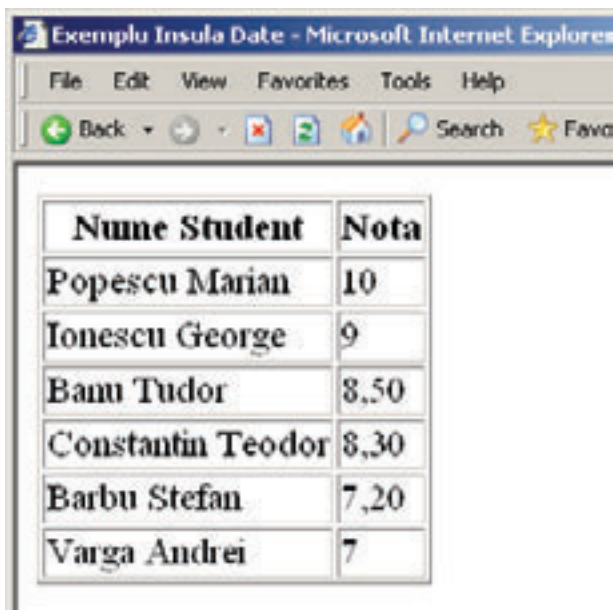
Cele două concepte pot fi folosite împreună pentru a putea separa datele care trebuiesc afișate de instrucțiunile de afișare într-o pagină Web, implementând astfel conceptul de **insulă de date** și permițând afișarea, selectivă sau totală a datelor conținute într-un document XML într-o pagină HTML

Acest lucru este posibil utilizând în cadrul paginii HTML directiva HTML <XML> pentru a o “lega” pe aceasta de datele din fișierul XML ca în exemplul de mai jos:

Presupunând ca avem următorul fișier XML care conține date despre studenții admiși la un anumit examen și nota obținută, ne propunem să realizăm o pagină HTML care să utilizeze datele din fișierul XML pentru a afișa un tabel cu studenții admiși într-o pagină Web.

Fișierul listaadmitere.xml:	Fișierul publicListaAdmitere.html
<pre> <?xml version="1.0" ?> - <listaadmitere> - <student> <nume>Popescu Marian</nume> <nota>10</nota> </student> - <student> <nume>Ionescu George</nume> <nota>9</nota> </student> - <student> <nume>Banu Tudor</nume> <nota>8,50</nota> </student> - <student> <nume>Constantin Teodor</nume> <nota>8,30</nota> </student> - <student> <nume>Barbu Stefan</nume> <nota>7,20</nota> </student> - <student> <nume>Varga Andrei</nume> <nota>7</nota> </student> </listaadmitere> </pre>	<pre> <HTML> <HEAD> <TITLE> Exemplu Insula Date </TITLE> </HEAD> <BODY> <XML ID="admitere" SRC="listaadmitere.xml" async="false"></XML> <TABLE BORDER="1" DATASRC="#admitere"> <THEAD> <TH> Nume Student </TH> <TH> Nota </TH> </THEAD> <TR> <TD> </TD> <TD> </TD> </TR> </TABLE> </BODY> </HTML> </pre>

Deschiderea paginii publicListaAdmitere în browser va avea ca rezultat afișarea datelor din fișierul XML sub forma unui tabel a cărui formă de afișare a fost definit în pagina HTML precum în figura 11.1:



The screenshot shows a browser window titled "Exemplu Insula Date - Microsoft Internet Explorer". The address bar is empty. The menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help". The toolbar contains "Back", "Forward", "Stop", "Home", "Search", and "Favorites" icons. The main content area displays a table with two columns: "Nume Student" and "Nota".

Nume Student	Nota
Popescu Marian	10
Ionescu George	9
Banu Tudor	8,50
Constantin Teodor	8,30
Barbu Stefan	7,20
Varga Andrei	7

Figura 11.1 Exemplu insula de date.



Test de autoevaluare

11.2 Definiți conceptul de insulă de date.

Răspunsul corect și comentarii asupra acestuia se găsesc la pagina 119.

Răspunsuri și comentarii la întrebările din testele de evaluare



Întrebarea 11.1.

NU XML este un limbaj de adnotare complementar cu HTML și **NU** înlocuiește HTML. Cele două limbaje, după cum s-a arătat, au scopuri distincte: HTML afișează informații ce apar în pagină Web, în timp ce XML descrie aceste informații. Revedeți indicațiile din secțiunea 11.1.



Întrebarea 11.2.

Conceptul de *insulă de date* se referă la separarea datelor de afișat de instrucțiunile de afișare dintr-un document HTML. Datele vor fi citite dintr-un document XML și afișate conform instrucțiunilor de afișare din documentul HTML. Pentru nelămuriri revedeți secțiunea 11.4.

Indicații la problemele propuse

Problemele propuse în lucrările de verificare se fac după modelele de exemple prezentate în unitatea de învățare sau după tipicul acestora.

Bibliografie

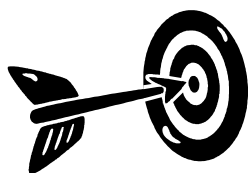
1. Dorin Cârstoiu, Ecaterina Oltean – Introducere în HTML și XML, Editura Printech București 2004, pg.59-82
2. <http://www.w3schools.com/default.asp>

Unitatea de învățare Nr. 12

FOI DE STIL

Obiectivele Unității de învățare Nr.12	121
12.1 Foi de stil în HTML - CSS	121
12.2 Cum funcționează stilurile în HTML	122
12.3 Adăugarea foilor de stil în documentul HTML	123
12.4 Proprietăți CSS	126
Lucrare de verificare a Unităților de învățare Nr. 10, 11 și 12	127
Răspunsuri și comentarii la întrebările din testele de evaluare	129
Bibliografie	129

Obiectivele Unității de învățare Nr. 12:



După parcurgerea acestei Unității de învățare vei ști:

- Care sunt avantajele folosirii foilor de stil
- Sintaxa folosită pentru definirea unei foi de stil
- Metode de folosire a informațiilor de stil într-un document HTML

12.1 Foi de stil în HTML - CSS

Foie de stil

Stilurile CSS au fost recent adăugate la limbajul HTML cu scopul de a adăuga mai multe facilități pentru controlul modului de apariție al paginii. Începând cu HTML 4, toate instrucțiunile de formatare pot fi definite în afara documentului HTML într-o structură denumită **foaie de stil**.

În literatura de specialitate foile de stil sunt referite de acronimul **CSS** care vine de la **Cascading Style Sheets**. Cuvântul “*Cascading*” înseamnă “în cascadă” și se referă la modul de aplicare al acestor stiluri asupra elementelor din pagina WEB .atunci când sunt folosite simultan mai multe foi de stil.

Avantaje

Acestea sunt avantajele folosirii foilor de stil:

- Separarea instrucțiunilor de formatare de structura documentului HTML
- Documentele HTML care folosesc foi de stil pot avea dimensiuni mai mici
- Site-uri mai ușor de menținut. Prin folosirea unui set comun de foi de stil pentru toate paginile HTML se poate schimba înfățișarea unui site care conține sute de pagini WEB prin editarea unui singur fișier.

Dezavantaje

Dezavantajul este că încă există browsere care nu au implementat suport pentru foi de stil. Dar chiar și pentru acest dezavantaj există un remediu. Paginile de web pot fi inițial realizate fără a folosi facilitățile de stil.

În acest fel se poate verifica modul în care acestea sunt afișate de browserele fără suport pentru CSS. Informațiile de stil urmând a fi adăugate pe urmă păstrând în același timp nealterat conținutul documentului HTML.

12.2 Cum funcționează stilurile în HTML

Sintaxa de bază O foaie de stil constă în una sau mai multe reguli care descriu modul în care un element dintr-o pagină HTML va fi afișat. O regulă într-o foaie de stil are următorul format:

```
selector{proprietate:valoare}
```

unde:

Selector **selector** – identifică elementul HTML pentru care se specifică modul de afișare. Practic orice element HTML poate fi un posibil selector CSS. În exemplul de mai jos, **P** este un selector de stil și va defini pentru elementele de tip <P> din documentul care folosește această foaie de stil dimensiunea fontului de 12 puncte

```
P {font-size: 12pt;}
```

Proprietate -valoare **proprietate** și **valoare** definesc stilul care va fi aplicat elementului identificat de selector. Proprietatea este separată de valoare prin caracterul “.” urmată de un spațiu. Această pereche, proprietate-valoare, este încadrată între acolade și reprezintă **declarația** stilului.

Declarație O declarație de stil poate conține mai multe perechi proprietate-valoare separate prin caracterul “;” ca în exemplul de mai jos:

```
P {font-size: 12pt;
    font-face: Arial}
```

Declarație

Gruparea stilurilor Pentru a nu folosi în mod repetat aceeași declarație pentru mai multe tipuri de elemente, CSS oferă posibilitatea grupării selectorilor. Spre exemplu toate titlurile dintr-un document HTML pot fi configurate să aibă anumite proprietăți comune:

```
H1, H2, H3, H4, H5, H6 {
    color: blue;
    font-face: Arial}
```



Test de autoevaluare

12.1 Care este principalul avantaj al folosirii foilor de stil la realizarea paginilor de WEB?

12.2 Ce este un selector într-o foaie de stil?

Răspunsurile corecte și comentarii asupra acestora se găsesc la pagina 129.

12.3 Adăugarea foilor de stil în documentul HTML

Regulile și seturile de reguli pot fi incluse în documentul HTML în trei feluri: ca stiluri inline, ca foaie de stil intern sau ca foaie de stil externă.

Stiluri INLINE

Stilurile inline. sunt adăugate fiecărui element HTML în parte prin folosirea atributului STYLE care este disponibil pentru majoritatea directivelor HTML. Valoarea atributului este reprezentată de una sau mai multe declarații de stil ca în exemplul de mai jos:

```
<H1 STYLE="color: blue">
  Acest titlu este albastru
</H1>
```

sau:

```
<P
  STYLE="font-size: 12pt;
        font-face: Verdana">
  Acesta este un paragraf cu stilul definit
  inline.
```



Foi interne de stil

Deși este o construcție perfect valabilă pentru HTML, nu este recomandată deoarece nici unul din avantajele foilor de stil nu poate fi folosit în acest caz. Dacă se dorește modificarea modului de afișare pentru un set de elemente este necesară intervenția în documentul HTML asupra fiecărui element în parte.

O metodă mult mai compactă și mai eficientă este folosirea foilor interne de stil. O foaie internă de stil se definește în headerul documentului HTML folosind directiva pereche <STYLE> </STYLE>.

În interiorul acestui element se pot defini stilurile elementelor din pagină folosind sintaxa descrisă mai sus, Spre exemplu:

```
<STYLE TYPE="test/css" >
  <!--
    P {color: blue};
    H1, H2, H3, H4, H5{
      color: aqua;
      font-size: 12pt
    }
  -->
</STYLE>
```

Se observă că în interiorul elementului STYLE declarațiile de stil sunt încadrate de comentariu HTML: <!-- și -->. Aceasta este pentru cazul în care pagina este vizualizată într-un browser care nu suportă foi de stil, pentru a evita afișarea informațiilor din cadrul acestui element în pagină.

O foaie internă de stil se poate aplica unui singur document HTML. Aceasta poate modifica aspectul paginii păstrând în același timp nealterat conținutul documentului HTML.

Foi externe de stil

Cea mai puternică metodă de folosire a foilor de stil este de a le defini pe toate în interiorul unui fișier separat la care să se facă referire în toate documentele HTML care vor folosi acest stil. Aceste foi de stil poartă denumirea de foi externe de stil, iar pentru referirea lor există două modalități.

Legarea

Legarea (linking) Este cea mai des folosită metodă și constă în folosirea directivei <LINK>. Această directivă se definește în interiorul headerului .

<LINK>

Funcționalitate:

- Importă o foaie de stil dintr-un fișier

Atribute:

- HREF
- REL
- TYPE

Directiva de sfârșit:**</LINK>** OBLIGATORIE

Iată un exemplu de folosire a acestei directive pentru importarea unei foi de stil definită într-un fișier *stilulmeu.css* extern aflat în același director cu documentul HTML:

```
<HEAD>
<LINK
    REL="STYLESHEET"
    HREF="stilulmeu.css"
    TYPE="text/css">
</HEAD>
```

**Atributul
REL**

Atributul **REL** – Definește relația dintre documentul HTML și fișierul cu care se face legătura – o foaie de stil: "STYLESHEET".

**Atributul
HREF**

Atributul **HREF** – Reprezintă URL-ul fișierului care conține declarațiile de stil.

**Atributul
TYPE**

Atributul **TYPE** – Definește formatul fișierului ce conține informațiile de stil. Se folosește valoarea "text/css".

Import O metodă alternativă pentru folosirea foilor externe de stil este folosirea directivei <STYLE> împreună cu declarația *@import* după cum este arătat în exemplul de mai jos:

```
<STYLE
    TYPE="text/css">
<!--
    @import url(http://stiluri.ro/stilulmeu.css);
    @import url(altstil.css);
    DT { background: yellow;
        color: black }
-->
</STYLE>
```



Declarația *@import* trebuie să fie înainte de orice altă declarație de stil.



Test de autoevaluare

12.3 Enumerați metodele de adăugare a informațiilor de stil într-o pagină HTML.

Răspunsul corect și comentarii asupra acestuia se găsesc la pagina 129.

12.4 Proprietăți CSS

O foaie de stil este compusă dintr-un set de instrucțiuni de afișare. Fiecare instrucțiune este la rândul său compusă dintr-un *selector* care identifică elementul căruia i se aplică stilul și o *declarație* care definește modul în care va fi afișat elementul respectiv. O declarație este o colecție de proprietăți CSS.

O proprietate CSS este o construcție de forma nume/valoare. CSS împarte aceste proprietăți în câteva categorii principale. Vom enumera aceste categorii și vom trece în revistă proprietățile din fiecare categorie în parte. Pentru o listă completă a proprietăților disponibile recomandăm consultarea materialelor din bibliografie.

Proprietati ale fontului

Proprietăți ale fontului - Aceste proprietăți definesc aspectul grafic al textului afișat. Majoritatea modifică caracteristicile fontului cu care este afișat textul. Iată câteva exemple de astfel de proprietăți:

- font-family – definește tipul fontului folosit pentru afișarea textului
- font-size – definește dimensiunea fontului folosit pentru afișarea textului
- font-style – definește modul de afișare a fontului: bold, italic, subliniat

Exemplu de utilizare

```
P {font-family: Arial;
    font-size: 14pt;
    font-style: italic
}
```

Proprietati pentru text

Proprietăți pentru text - Aceste proprietăți definesc modul de aliniere, și spațiere la afișarea unui text. Iată câteva exemple de astfel de proprietăți:

- letter-spacing – definește dimensiunea spațiului între litere
- text-align – definește modul de aliniere al textului

- `text-intend` – definește spațiul folosind pentru intendarea textului

Exemplu de utilizare:

```
P {letter-spacing: 0.1em;
text-align: center
text-intend: 5em
}
```

Proprietati pentru spatierea elementelor

Proprietăți pentru spațierea elementelor - Aceste proprietăți sunt folosite pentru a defini spațiul dintre elemente HTML, marginile acestora și alte aspecte privind poziționarea lor. Iată câteva exemple de astfel de proprietăți:

- `padding-top` – definește dimensiunea spațiului ce va fi pus între conținutul elementului și marginea superioară
- `padding-right` - definește dimensiunea spațiului ce va fi pus între conținutul elementului și marginea dreaptă
- `padding-left` – similar cu `padding-right` pentru stânga.

Proprietati pentru definirea culorilor

Proprietăți pentru definirea culorilor:

- `color` - Această proprietate este folosită pentru a defini culoarea textului din cadrul elementului specificat de către selector. Culoarea poate fi specificată fie prin codul RGB, fie prin numele predefinit.

Proprietati pentru definirea fundalului

Proprietăți pentru definirea fundalului - Aceste proprietăți sunt folosite pentru a defini caracteristicile fundalului pentru elementul specificat de către selector. Iată câteva exemple de astfel de proprietăți:

- `background-color` – definește culoarea fundalului
- `background-image` – definește imaginea ce va fi afișată ca fundal.

Lucrare de verificare a Unităților de învățare Nr. 10, 11, 12



1. Realizați o pagină Web care să includă un applet Java al cărui cod se află în fișierul *film.class*. Appletul are deja scris codul pentru a rula un film de la un anumit URL însă acest URL trebuie transmis ca parametru de intrare. Poziționați appletul în centrul paginii.

Predați fișierul HTML în format electronic ca rezultat al rezolvării lucrării de verificare

Ca ajutor suplimentar în rezolvarea lucrării se recomandă

utilizarea reperului bibliografic numărul 1 din bibliografia unității de învățare nr.10.

Nr. puncte: **9** (4p. – utilizarea corectă a directivei APPLET, 3p - utilizarea atributului PARAM, 2p - poziționarea apletului pe centrul paginii)

2. Realizați un document XML care să descrie și să conțină informații despre produsele unui magazin, precum: Nume produs, descriere, preț, cantitate etc. Realizați pe urmă un document HTML care să afișeze aceste date din fișierul XML sub formă de tabel.

Predați documentele în format electronic (un fișier HTML și un fișier XML) ca rezultat al rezolvării lucrării de verificare.

Ca ajutor suplimentar în rezolvarea lucrării se recomandă utilizarea reperului bibliografic numărul 1 din bibliografia unității de învățare nr.11.

Nr. puncte: **8** (4p – realizarea corectă a fișierului XML, 4p realizarea fișierului HTML care să afișeze informațiile din cadrul documentului XML)

3. Definiți un fișier *stilulmeu.css* pe care să îl folosiți pentru a construi o pagină HTML care să formateze elementele folosind foaia de stil din fișierul *stilulmeu.css*. Stilurile definite în acest fișier trebuie să afecteze următoarele elemente din pagina HTML:

- H1, H2 , H3 să fie scris cu font Veranda cu culoarea galbena
- Textul din documentul HTML să fie scris cu font de dimensiunea 10pt, culoarea textului să fie albă, iar culoarea de fundal să fie neagră.

Predați fișierul *stilulmeu.css* precum și documentul HTML în format electronic ca rezultat al rezolvării lucrării de verificare

Ca ajutor suplimentar în rezolvarea lucrării se recomandă utilizarea reperului bibliografic numărul 1 din bibliografia unității de învățare.

Nr. puncte: **8** (5p – definirea corectă în cadrul foii de stil a atributelor cerute, 3p – folosirea foii de stil in cadrul documentului HTML)

Răspunsuri și comentarii la întrebările din testele de evaluare



Întrebarea 12.1.

Principalul avantaj este separarea informațiilor de formatare de structura documentului HTML și posibilitatea modificării modului de afișare a paginii fără a intervenii în documentul HTML. Pentru nelămuriri revedeți secțiunea 12.1.



Întrebarea 12.2.

Un selector, este acea parte dintr-o regula CSS care identifică elementul HTML pentru care se specifică modul de afișare. A se revedea secțiunea 12.2.



Întrebarea 12.3.

Informațiile de stil pot fi adăugate într-un document HTML folosind una din cele 3 metode: adăugarea informațiilor inline, folosirea foilor interne de stil, folosirea foilor de stil definite într-un fișier extern. Consultați secțiunea 12.3.

Indicații la problemele propuse

Problemele propuse în lucrările de verificare se fac după modelele de exemple prezentate în unitatea de învățare sau după tipicul acestora.

Bibliografie

1. Dorin Cârstoiu, Ecaterina Oltean – Introducere în HTML și XML, Editura Printech București 2004, pg.51-53
2. Sabin Buraga – Proiectarea siturilor Web. Design și funcționalitate, Ediția a II-a, Editura Polirom 2002, pg. 163-195
3. Călin Ioan Acu – Optimizarea paginilor Web, Editura Polirom 2003, pg. 35-73
4. Richard Wagner, R.Allen Wyke – Java Script, Editura Teora 2000, pg.481-495

BIBLIOGRAFIE

1. Călin Ioan Acu – Optimizarea paginilor Web, Editura Polirom 2003
2. Cioata Mihai – ActiveX. Concepte si aplicații Editura: Polirom 2003
3. Dorin Cârstoiu, Ecaterina Oltean – Introducere în HTML și XML, Editura Printech București 2004
4. Mihaela Brut, Sabin Buraga - Prezentați multimedia pe Web. Limbajele XHTML + TIME și SMIL Editura Polirom 2003
5. McFedries Paul, Trad Voin, Doru Sorin – Crearea paginilor WEB, Editura ALL 2003
6. Negrino Tom, Smith Dori - JavaScript pentru World Wide Web. Ghid de învățare rapidă prin imagini, Editura Corint 2004
7. Richard Wagner, R.Allen Wyke – Java Script, Editura Teora 2000
8. Sabin Buraga – Proiectarea siturilor Web. Design și funcționalitate, Ediția a II-a, Editura Polirom 2002
9. T.Gugoiu – HTML prin exemple, Editura Teora 2000
10. <http://www.w3schools.com/default.asp>

ANEXA 1 – Lucrări de laborator

Sesiunea de lucru de laborator L1

Setul de lucrări are ca structură aprofundarea și consolidarea cunoștințelor dobândite în cadrul Unităților de învățare Nr.1 și 2. Sesiunii de laborator îi sunt alocate 4 ore. Se recomandă a se efectua sub supravegherea cadrului didactic.

Pentru efectuarea laboratorului este necesar utilizarea unui editor de texte, spre exemplu: Windows Notepad, și a unui program de navigare internet: Internet Explorer

Obiectivele acestei sesiuni de lucru sunt:

- Însușirea unor noțiuni de bază din domeniul rețelelor de calculatoare
- Însușirea pașilor necesari într-un ciclu de realizare-vizualizare rezultat, pentru o pagină Web
- Însușirea regulilor de sintaxă pentru directivele HTML
- Însușirea modului de definire a culorilor pentru elementele HTML

În cadrul lucrării de laborator numărul 2 se recomandă efectuarea următoarelor tipuri de activități:

- Discuție pe tema noțiunilor introduse în Unitatea de învățare numărul 1 cu scopul de a clarifica nelămuriri în ceea ce privește arhitectura World Wide Web sau Internet
- Efectuarea unui exercițiu la calculator, asistat de tutore pas cu pas care să urmărească clarificarea pașilor necesari într-un ciclu de realizare-vizualizare rezultat, pentru o pagină WEB
- Discuție despre structura de bază a unui document HTML (schelet) și despre reguli ce privesc sintaxa directivelor HTML
- Efectuarea unui exercițiu la calculator care să urmărească definirea și utilizarea corectă a culorilor într-un document HTML

Sesiunea de lucru de laborator L2

Setul de lucrări are ca structură aprofundarea și consolidarea cunoștințelor dobândite în cadrul Unităților de învățare Nr.3 și 4. Sesiunii de laborator îi sunt alocate 4 ore. Se recomandă a se efectua sub supravegherea cadrului didactic.

Pentru efectuarea laboratorului este necesar utilizarea unui editor de texte, spre exemplu: Windows Notepad, și a unui program de navigare internet: Internet Explorer

Obiectivele acestei sesiuni de lucru sunt:

- Definirea și folosirea titluri și paragrafe în HTML
- Definirea și folosirea directivelor HTML pentru formatare logică
- Definirea și folosirea directivelor HTML pentru formatare fizică
- Definirea și folosirea listelor în limbajul HTML

În cadrul lucrării de laborator numărul 2 se recomandă efectuarea următoarelor tipuri de activități:

- Discuție pe tema formătărilor textelor HTML în care se vor clarifica noțiunile însușite în Unitatea de învățare numărul 3 legate de formatare textelor cu ajutorul HTML
- Efectuarea unui exercițiu la calculator care să urmărească însușirea noțiunilor și tehnicilor de formatare a textelor HTML.
- Discuție pe tema celor 3 tipuri de liste HTML în care se vor clarifica noțiunile însușite în Unitatea de învățare numărul 4 legate de utilizarea listelor HTML
- Efectuarea unui exercițiu la calculator care să urmărească însușirea modului de definire a listelor în limbajul HTML.

Sesiunea de lucru de laborator L3

Setul de lucrări are ca structură aprofundarea și consolidarea cunoștințelor dobândite în cadrul Unităților de învățare Nr. 5, și 6. Sesiunii de laborator îi sunt alocate 4 ore. Se recomandă a se efectua sub supravegherea cadrului didactic.

Pentru efectuarea laboratorului este necesar utilizarea unui editor de texte, spre exemplu: Windows Notepad, și a unui program de navigare internet: Internet Explorer

Obiectivele acestei sesiuni de lucru sunt:

- Înțelegerea conceptelor de Hypertext, Hyperlink și URL
- Definirea și folosirea legăturilor în cadrul paginilor Web
- Înțelegerea formatelor grafice folosite în Web
- Înțelegerea modului de adăugare și utilizare a imaginilor într-o pagina Web

În cadrul lucrării de laborator numărul 3 se recomandă efectuarea următoarelor tipuri de activități:

- Discuție teoretică pe marginea conceptelor de Hypertext, Hyperlink și URL. Se va urmări clarificarea acestor concepte,

și înțelegerea modului de folosire al acestor concepte în cadrul paginilor WEB

- Efectuarea unui exercițiu la calculator care să urmărească însușirea noțiunilor și tehnicilor de utilizare a linkurilor în cadrul paginilor HTML.
- Discuție asupra tipurilor de formate grafice folosite pentru realizarea paginilor WEB, avantajele și dezavantajele fiecărui tip de format.
- Discuție teoretică pe tema utilizării imaginilor în cadrul paginilor WEB cu scopul clarificării noțiunilor din cadrul Unității de învățare numărul 6
- Efectuarea unui exercițiu la calculator care să urmărească însușirea modului de folosire a imaginilor, integrare, aliniere, dimensionare în pagina HTML

Sesiunea de lucru de laborator L4

Setul de lucrări are ca structură aprofundarea și consolidarea cunoștințelor dobândite în cadrul Unității de învățare Nr.7

Sesiunii de laborator îi sunt alocate 4 ore. Se recomandă a se efectua sub supravegherea cadrului didactic.

Pentru efectuarea laboratorului este necesar utilizarea unui editor de texte, spre exemplu: Windows Notepad, și a unui program de navigare internet: Internet Explorer

Obiectivele acestei sesiuni de lucru sunt:

- Înțelegerea modului de realizare și de configurare al tabelelor HTML
- Explicarea părților componente ale unui tabel HTML și a modului în care proprietățile acestora pot fi schimbate
- Prezentarea diferitelor moduri de folosire a tabelelor în pagina HTML

În cadrul lucrării de laborator numărul 4 se recomandă efectuarea următoarelor tipuri de activități:

- Explicarea modului de utilizare a tabelelor și clarificarea noțiunilor prezentate în Unitatea de învățare numărul 7.
- Efectuarea unui exercițiu la calculator care să urmărească însușirea modului de definire și folosire a tabelelor HTML

Sesiunea de lucru de laborator L5

Setul de lucrări are ca structură aprofundarea și consolidarea cunoștințelor dobândite în cadrul Unității de învățare Nr. 8.

Sesiunii de laborator îi sunt alocate 4 ore. Se recomandă a se efectua sub supravegherea cadrului didactic.

Pentru efectuarea laboratorului este necesar utilizarea unui editor de texte, spre exemplu: Windows Notepad, și a unui program de navigare internet: Internet Explorer

Obiectivele acestei sesiuni de lucru sunt:

- Clarificarea conceptului de formular HTML
- Explicarea modului de folosire al fiecărui tip de element din cadrul unui formular HTML
- Înțelegerea modului de folosire al formularelor HTML

În cadrul lucrării de laborator numărul 5 se recomandă efectuarea următoarelor tipuri de activități:

- Discuție asupra modului de utilizare a tabelelor și clarificarea noțiunilor prezentate în Unitatea de învățare numărul 8.
- Efectuarea unui exercițiu la calculator care să urmărească însușirea modului de definire și folosire a formularelor HTML. Exercițiul ar trebui să acopere modul de folosire al fiecărui tip de element din cadrul unui formular HTML

Sesiunea de lucru de laborator L6

Setul de lucrări are ca structură aprofundarea și consolidarea cunoștințelor dobândite în cadrul Unităților de învățare Nr. 9 și 10. Sesiunii de laborator îi sunt alocate 4 ore. Se recomandă a se efectua sub supravegherea cadrului didactic.

Pentru efectuarea laboratorului este necesar utilizarea unui editor de texte, spre exemplu: Windows Notepad, și a unui program de navigare internet: Internet Explorer, precum și accesul la componente JavaApplets și/sau ActiveX.

Obiectivele acestei sesiuni de lucru sunt:

- Înțelegerea rolului limbajului JavaScript
- Clarificare noțiunilor de tip script pe partea de client, Evenimente și tratarea acestora
- Înțelegerea modului de folosire al JavaScript în cadrul unui document HTML
- Clarificarea aspectelor legate de folosirea executabilelor și componentelor multimedia în pagina WEB

În cadrul lucrării de laborator numărul 6 se recomandă efectuarea următoarelor tipuri de activități:

- Discuție asupra modului de utilizare a tabelelor și clarificarea noțiunilor prezentate în Unitatea de învățare numărul 8.
- Efectuarea unui exercițiu la calculator care să urmărească însușirea modului de definire și folosire a formularelor HTML. Exercițiul ar trebui să acopere modul de folosire al fiecărui tip de element din cadrul unui formular HTML
- Efectuarea unui exercițiu la calculator care să urmărească însușirea modului de folosire al diferitelor tipuri de componente multimedia și executabile în cadrul unei pagini WEB

Sesiunea de lucru de laborator L7

Setul de lucrări are ca structură aprofundarea și consolidarea cunoștințelor dobândite în cadrul Unităților de învățare Nr. 11 și 12 Sesiunii de laborator îi sunt alocate 4 ore. Se recomandă a se efectua sub supravegherea cadrului didactic.

Pentru efectuarea laboratorului este necesar utilizarea unui editor de texte, spre exemplu: Windows Notepad, și a unui program de navigare internet: Internet Explorer

Obiectivele acestei sesiuni de lucru sunt:

- Clarificarea noțiunilor legate de XML precum definiții, sintaxă, caracteristici
- Clarificarea modului de folosire al XML în cadrul unei pagini WEB, și a conceptului de separare a datelor utile de interfața grafică
- Clarificarea noțiunii de pagină de stil.
- Evidențierea avantajelor folosirii paginilor de stil

În cadrul lucrării de laborator numărul 7 se recomandă efectuarea următoarelor tipuri de activități:

- Discuție asupra conceptelor de bază XML și clarificarea noțiunilor de bază
- Efectuarea unui exercițiu la calculator care să urmărească însușirea noțiunilor fundamentale ale limbajului XML.
- Discuție asupra modului de utilizare al paginilor de stil în cadrul unei pagini WEB
- Efectuarea unui exercițiu la calculator care să urmărească însușirea modului de folosire al paginilor de stil

Felicitări pentru parcurgerea acestui modul! Acesta reprezintă un prim pas în vastul domeniu al Internetului care este într-o continuă evoluție. Și acum vă dorim succes în realizarea paginilor de WEB pe care le veți realiza!

